



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ**  
**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**  
**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ**  
**ΥΠΟΛΟΓΙΣΤΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

Ανάπτυξη Web-App με Server/Client Side για ολική διαχείριση, έξυπνη αναζήτηση  
και Online κράτηση σε Pet Grooming Services

**ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΚΟΥΚΗΣ ΝΙΚΟΛΑΟΣ**

**ΑΡΙΘΜΟΣ ΜΗΤΡΩΟΥ: 2851**

**ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΤΖΗΜΑΣ ΙΩΑΝΝΗΣ**

**ΠΑΤΡΑ, 2021**

## ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία πραγματοποιείται στα πλαίσια της φοίτησής μου στο Πανεπιστήμιο Πελοποννήσου, στο τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών και έχει ως σκοπό την ανάπτυξη μίας κύριας διαδικτυακής εφαρμογής για ολική διαχείριση, έξυπνη αναζήτηση και online κράτηση σε Pet Grooming Services. Για το σκοπό αυτό, χρησιμοποιούνται οι τεχνολογίες κατασκευών διαδικτυακών εφαρμογών, εξετάζονται και δοκιμάζονται όλες οι σύγχρονες τεχνολογίες στην πράξη για την κατασκευή της.

## ABSTRACT

This dissertation is carried out during my studies at the University of Peloponnese, in the Department of Electrical and Computer Engineering and aims to develop a major online application for total management, smart search and online booking in Pet Grooming Services. For this purpose, the technologies of construction of web applications are used, all modern technologies are examined and tested in practice for its construction.

## ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου Ιωάννη Τζήμα, για τις οδηγίες και το χρόνο που μου προσέφερε για την ολοκλήρωση της διπλωματικής μου εργασίας. Επίσης, θα ήθελα να ευχαριστήσω όλους τους καθηγητές του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών για τη συμβολή τους στην τεχνολογική μου πορεία στα χρόνια της φοίτησής μου στο Τμήμα. Τέλος, ένα μεγάλο ευχαριστώ στους γονείς μου για την υποστήριξή τους σε όλο το διάστημα των σπουδών μου.

# ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ .....	2
ABSTRACT .....	3
ΕΥΧΑΡΙΣΤΙΕΣ .....	4
ΠΕΡΙΕΧΟΜΕΝΑ .....	5
ΠΡΟΛΟΓΟΣ.....	7
ΚΕΦΑΛΑΙΟ 1 ΘΕΩΡΗΤΙΚΟ ΠΛΑΙΣΙΟ .....	9
1.1 Web App Εφαρμογή.....	9
1.1.1 Πλεονεκτήματα Web App .....	10
1.1.2 Μειονεκτήματα Web App .....	11
1.2 HTML.....	12
1.2.1 Ετικέτες/tags της HTML .....	13
1.3 CSS.....	16
1.3.1 Επιλογείς (Selectors) .....	17
1.4 JavaScript .....	18
1.4.1 Εμφωλευμένο Script.....	19
1.4.2 Εσωτερικό Script.....	20
1.4.3 Εξωτερικό Script .....	21
1.5 Bootstrap .....	22
1.6 jQuery.....	24
1.6.1 Τρόπος λειτουργίας της jQuery .....	25
1.7 Cron Job .....	27
1.8. PHP.....	31
1.9 MySQL.....	33
1.9.1 Τρόπος λειτουργίας MySQL .....	34
ΚΕΦΑΛΑΙΟ 2 ΔΗΜΙΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ/WEB APP .....	37
2.1 Κατηγορίες χρηστών .....	37
2.1.1 Διαχειριστής (Administrator) .....	37
2.1.2 Πελάτης (Client).....	37
2.2 Βάση δεδομένων .....	38
2.2.1 Πίνακες διαχειριστή .....	38
2.2.2 Πίνακες πελάτη .....	41
2.2.3 Κοινοί πίνακες.....	43
2.3 Εφαρμογή τεχνολογιών .....	46

ΚΕΦΑΛΑΙΟ 3 ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ .....	47
3.1 Εγχειρίδιο χρήσης για τον χρήστη – user .....	47
3.2 Εγχειρίδιο χρήσης για τον διαχειριστή - groomer .....	67
ΕΠΙΛΟΓΟΣ/ ΣΥΜΠΕΡΑΣΜΑ .....	91
ΠΑΡΑΡΤΗΜΑ 1 .....	92
ΠΑΡΑΡΤΗΜΑ 2 .....	108
ΠΑΡΑΡΤΗΜΑ 3 .....	113
ΒΙΒΛΙΟΓΡΑΦΙΑ/ΔΙΚΤΥΟΓΡΑΦΙΑ .....	131

## ΠΡΟΛΟΓΟΣ

Η παρούσα διπλωματική εργασία πραγματοποιείται στα πλαίσια της φοίτησής μου στο Πανεπιστήμιο Πελοποννήσου, στο τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, και έχει ως σκοπό την ανάπτυξη μίας κύριας διαδικτυακής εφαρμογής για ολική διαχείριση, έξυπνη αναζήτηση και online κράτηση σε Pet Grooming Services.

Η σχεδίαση της εφαρμογής έγινε βάσει των τελευταίων web τεχνολογιών και των βάσεων δεδομένων, οι οποίες χρησιμοποιούνται για την ανάπτυξη σύγχρονων Web Εφαρμογών και υπηρεσιών. Αυτές που χρησιμοποιήθηκαν στην παρούσα διπλωματική εργασία είναι ονομαστικά η HTML(5), η CSS(3), η JavaScript, η PHP, η MySQL καθώς και οι βιβλιοθήκες Bootstrap και jQuery.

Η εφαρμογή έχει ως στόχο να παρέχει ένα πρακτικό περιβάλλον, μέσω του οποίου θα γίνεται καταχώρηση ενός ραντεβού και επεξεργασία από τον διαχειριστή του συστήματος, καθώς θα υπάρχει και άμεση ενημέρωση του χρήστη – πελάτη μέσω email. Η εφαρμογή κάνει αναφορά για δύο είδη χρηστών, που είναι ο απλός χρήστης – πελάτης και ο διαχειριστής – groomer.

Από την μεριά του πελάτη, ο χρήστης εισέρχεται στην εφαρμογή με τα στοιχεία του, τα οποία του δίνεται η δυνατότητα, αν το επιθυμεί, να τα αλλάξει ή, σε περίπτωση που δεν τα θυμάται, να αποκτήσει νέα. Έπειτα, ο πελάτης μπορεί να δει τις πληροφορίες των μαγαζιών, όλους τους διαθέσιμους groomer, τα διαθέσιμα ραντεβού και να κάνει αίτημα για ραντεβού.

Από την μεριά του διαχειριστή – groomer, ο οποίος και αυτός εισέρχεται στην εφαρμογή με τα στοιχεία του, τα οποία μπορεί να επεξεργαστεί ή και να αλλάξει, επεξεργάζεται το αίτημα του πελάτη και μπορεί είτε να το αποδεχτεί είτε να το απορρίψει. Και στα δύο είδη χρηστών έχουν προστεθεί επιπλέον λειτουργίες και δυνατότητες που έχουν τα σύγχρονα διαχειριστικά συστήματα, όπως επεξεργασία του προφίλ ή το ιστορικό των ραντεβού.

Η εφαρμογή χρησιμοποιεί βάση δεδομένων, με σκοπό να καταχωρούνται όλα τα δεδομένα για τους χρήστες, τους διαχειριστές, τα ραντεβού και τις ώρες λειτουργίας των καταστημάτων. Αξίζει να σημειωθεί πως υπάρχει συχνή αλληλεπίδραση της βάσης δεδομένων με την εφαρμογή, με σκοπό την καταχώρηση, την τροποποίηση και την προβολή των δεδομένων.

Η εργασία αυτή αποτελείται από τρία βασικά μέρη. Το πρώτο μέρος, το θεωρητικό, παρουσιάζει όλα τα εργαλεία που χρησιμοποιήθηκαν για την σχεδίαση και την κατασκευή της εφαρμογής. Στο δεύτερο μέρος, αναφερόμαστε στην δημιουργία της εφαρμογής ενώ στο τρίτο μέρος γίνεται η παρουσίαση της εφαρμογής στην τελειοποιημένη της μορφή, συνοδευόμενη από στιγμιότυπα και την σταδιακή επεξήγηση των πρακτικών που ακολουθήθηκαν, ώστε να φθάσει η εφαρμογή στην μορφή αυτή. Τέλος, πριν το κλείσιμο της εργασίας, θα παρουσιαστούν βασικά στοιχεία του κώδικα, πάνω στον οποίο δημιουργήθηκε η εφαρμογή, και από την πλευρά του πελάτη αλλά και του διαχειριστή.



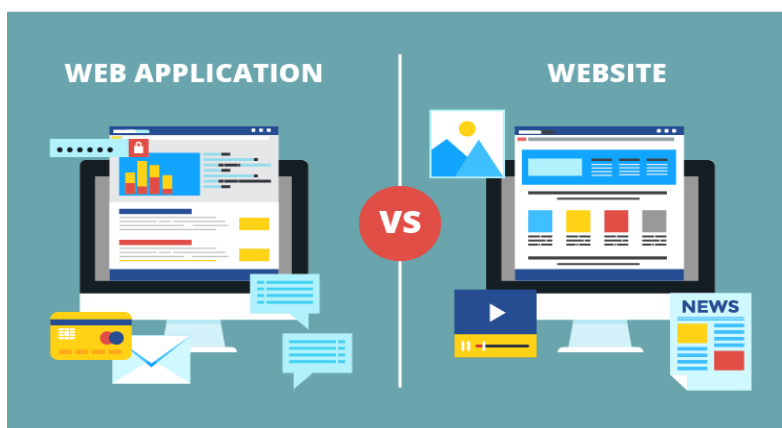
# ΚΕΦΑΛΑΙΟ 1

## ΘΕΩΡΗΤΙΚΟ ΠΛΑΙΣΙΟ

### 1.1 Web App Εφαρμογή

Για να κατανοήσουμε την έννοια της web εφαρμογής θα πρέπει πρώτα να αφομοιώσουμε την έννοια μιας εφαρμογής. Με τον όρο εφαρμογή, αναφερόμαστε στο λογισμικό που εγκαθίσταται σε έναν ηλεκτρονικό υπολογιστή με σκοπό να πραγματοποιεί κάποιες συγκεκριμένες διεργασίες και να εξάγει στον χρήστη την απαραίτητη πληροφορία. Η Web εφαρμογή είναι το ίδιο, αλλά διαφέρει στο ότι δεν εγκαθίσταται σε έναν ή περισσότερους ηλεκτρονικούς υπολογιστές, αλλά μπορείς να έχεις πρόσβαση μέσω ενός φυλλομετρητή (web browser), μέσω του διαδικτύου (internet) ή κάποιου τοπικού δικτύου.

Η μόνη διαφορά ανάμεσα σε μία web εφαρμογή και μία ιστοσελίδα, είναι ότι η βασική λειτουργία μίας ιστοσελίδας είναι να πληροφορεί τον χρήστη με την βοήθεια



κειμένων, εικόνων και video, ενώ η web εφαρμογή προσφέρει στον χρήστη ένα εύχρηστο περιβάλλον στο οποίο ο χρήστης μπορεί να εισάγει δεδομένα, να τα εξάγει αλλά και να τα επεξεργαστεί ώστε να πετύχει κάποιο σκοπό. Ένα παράδειγμα Web εφαρμογής είναι το γνωστό σε όλους «Hotmail», με το οποίο ο χρήστης αλληλεπιδρά με το σύστημα.

Σε καμία περίπτωση η Web εφαρμογή δεν υποβαθμίζει την έννοια της ιστοσελίδας, αλλά προσφέρει περισσότερες λύσεις σε πιο εξειδικευμένες ανάγκες. Η ιστοσελίδα είναι πλέον ανεκτίμητη και αναντικατάστατη. Οι Web εφαρμογές είναι εργαλεία των επιχειρήσεων, καθώς με αυτές προσφέρουν στους πελάτες τους προηγμένες υπηρεσίες. Για να έχει κάποιος πρόσβαση σε μια Web εφαρμογή αρκεί να έχει πρόσβαση στο διαδίκτυο (internet) ή σε κάποιο τοπικό δίκτυο. Παρακάτω θα παρουσιαστούν κάποια πλεονεκτήματα αλλά και μειονεκτήματα των Web εφαρμογών.

### 1.1.1 Πλεονεκτήματα Web App

Η Web App παρουσιάζει τα εξής πλεονεκτήματα:

- **Άμεση πρόσβαση από οποιαδήποτε συσκευή:** Οι χρήστες των Web εφαρμογών έχουν άμεση πρόσβαση στις εφαρμογές από οποιονδήποτε ηλεκτρονικό υπολογιστή ή συσκευή που είναι συνδεδεμένη στο διαδίκτυο (internet), χωρίς να χρειάζεται να γίνει εγκατάσταση επιπρόσθετου λογισμικού. Η μόνη απαραίτητη εφαρμογή είναι ο φυλλομετρητής διαδικτύου (browser), ο οποίος είναι προεγκατεστημένος σε όλα τα λειτουργικά συστήματα των συσκευών, είτε είναι ηλεκτρονικοί υπολογιστές, είτε φορητές είτε σε κινητά τηλέφωνα.
- **Δυνατότητα χρήσης ανεξαρτήτως τοποθεσίας:** Με βάση τα παραπάνω, οι χρήστες των Web εφαρμογών μπορούν να τις χρησιμοποιήσουν ακόμα και εκτός χώρου εργασίας τους. Η δυνατότητα αυτή προσφέρει ευελιξία και άνεση στους χρήστες, ώστε να χρησιμοποιούν τις εφαρμογές από οποιοδήποτε χώρο κι αν βρίσκονται, καθώς και να δουλεύουν εξ αποστάσεως από το σπίτι.
- **Συμβατότητα με όλα τα λειτουργικά συστήματα:** Ένα επιπλέον πλεονέκτημα των Web εφαρμογών είναι ότι είναι συμβατές με όλα τα λειτουργικά συστήματα. Εφόσον η εφαρμογή αυτή εκτελείται μέσω του φυλλομετρητή διαδικτύου (browser) και όχι στον υπολογιστή του χρήστη, της δίνει την δυνατότητα να εκτελείται σε όλα τα λειτουργικά συστήματα. Σε όλα αυτά βέβαια παίζουν και σημαντικό ρόλο οι γλώσσες προγραμματισμού που χρησιμοποιεί η εφαρμογή.
- **Δεν καταναλώνουν πόρους:** Με βάση τα παραπάνω, εφόσον οι Web εφαρμογές δεν εκτελούνται στους υπολογιστές των χρηστών, δεν καταναλώνουν και τους πόρους του συστήματος. Για τον λόγο αυτό οι Web εφαρμογές είναι σχετικά ελαφριές για τον υπολογιστή.
- **Δεν καταλαμβάνουν χώρο:** Με βάση την παραπάνω λογική οι εφαρμογές δεν καταλαμβάνουν σχεδόν καθόλου χώρο στον δίσκο του χρήστη, αφού το συνολικό μέρος της εφαρμογής είναι αποθηκευμένο στον εξυπηρετητή και μόνο όταν η εφαρμογή χρησιμοποιείται μπορεί να υπάρχει μεταφορά δεδομένων προς τον υπολογιστή του χρήστη και μόνο όταν ο χρήστης το επιθυμεί.

- **Γρήγορη αναβάθμιση:** Ένα από τα σημαντικότερα πλεονεκτήματα κάνει αναφορά στην αναβάθμιση των εφαρμογών, είτε αυτές είναι τοπικές είτε είναι διαδικτυακές. Σε μια απλή εφαρμογή η αναβάθμιση της θα πρέπει να γίνει σε κάθε υπολογιστή ξεχωριστά, με σκοπό να δαπανηθεί χρόνος και χρήμα. Σε αντίθεση με μία Web εφαρμογή, η αναβάθμιση πραγματοποιείται στον Web Server που φιλοξενεί την εφαρμογή και ταυτόχρονα το αναβαθμισμένο πρόγραμμα είναι διαθέσιμο για όλους τους χρήστες. Με αυτό τον τρόπο υπάρχει εξοικονόμηση χρόνου, ο οποίος είναι πολύτιμος στις μεγάλες επιχειρήσεις, καθώς και χαμηλό κόστος μιας και απαιτείται λιγότερο εργατικό δυναμικό για την υλοποίηση της αναβάθμισης.
- **Νέο, βελτιωμένο περιβάλλον:** Ένα επιπλέον βασικό πλεονέκτημα των Web εφαρμογών είναι ότι πλέον με την εμφάνιση των καινούργιων τεχνολογιών ανάπτυξης διαδικτυακών εφαρμογών όπως είναι η HTML5, η CSS3 καθώς και η χρήση της JavaScript, είναι δυνατόν ο προγραμματιστής να φτιάξει μία φιλική και εύχρηστη προς το χρήστη εφαρμογή με ευκολότερο τρόπο.

### 1.1.2 Μειονεκτήματα Web App

Η Web App παρουσιάζει τα εξής μειονεκτήματα:

- **Χρήση της εφαρμογής εκτός διαδικτύου:** Ένα μειονέκτημα, που είναι και πλεονέκτημα ταυτόχρονα, είναι η χρήση της εφαρμογής εκτός διαδικτύου (internet) ή το ενδοδίκτυο της εταιρείας. Για να χρησιμοποιηθεί μία Web εφαρμογή εκτός διαδικτύου θα πρέπει να έχει σχεδιαστεί και κατασκευαστεί με ανάλογο τρόπο. Σε περίπτωση που δεν έχει πραγματοποιηθεί κάτι τέτοιο, τότε η εφαρμογή δε θα δουλεύει.
- **Αδυναμία χρήσης χωρίς σύνδεση στο διαδίκτυο:** Με βάση τα παραπάνω, οι εφαρμογές που δεν έχουν υλοποιηθεί με τις τελευταίες τεχνολογίες κατασκευών των Web εφαρμογών, δηλαδή παλαιότερες εφαρμογές δεν είναι δυνατόν να χρησιμοποιηθούν αν δεν υπάρχει σύνδεση στο διαδίκτυο. Αυτό είναι και το κυριότερο μειονέκτημα των παλιών Web εφαρμογών, καθώς αν υπάρξει διακοπή σύνδεσης με το διαδίκτυο, τότε η εφαρμογή δε θα μπορεί να χρησιμοποιηθεί.

- **Μη πλήρης συμβατότητα των περιηγητών:** Ένα ακόμα μειονέκτημα, που αφορά κυρίως την τελευταία έκδοση της HTML, είναι η μη πλήρης συμβατότητα των φυλλομετρητών με την έκδοση αυτή. Αν και παραπάνω έχουμε παρουσιάσει πολλά πλεονεκτήματα των τελευταίων τεχνολογιών ανάπτυξης Web εφαρμογών, αρκετοί φυλλομετρητές δεν είναι ακόμα πλήρως συμβατοί με αυτά. Επίσης, αν ένας προγραμματιστής δεν έχει προβλέψει τη μη λειτουργία κάποιου χαρακτηριστικού της εφαρμογής σε κάποιον φυλλομετρητή του διαδικτύου, αυτό μπορεί να έχει ως αποτέλεσμα να μην λειτουργεί σωστά η εφαρμογή ή ακόμα και να μη λειτουργεί καθόλου.
- **Άμεση αναβάθμιση:** Ένα ακόμα μειονέκτημα, που είναι ταυτόχρονα και πλεονέκτημα, αποτελεί την αναβάθμιση της εφαρμογής. Όταν η εφαρμογή είναι τοπική, τότε η επιχείρηση μπορεί να αναβαθμίσει την εφαρμογή όταν το κρίνει αναγκαίο κρίνοντας το κόστος, την αξιοπιστία αλλά και το χρόνο που θα δαπανηθεί, ώστε το εργατικό δυναμικό της να προσαρμοστεί στην αναβάθμιση αυτή. Αντίθετα, όταν η εφαρμογή είναι διαδικτυακή, η αναβάθμιση γίνεται αυτόματα, χωρίς να ενημερωθούν οι χρήστες. Για παράδειγμα, όταν μια ερχόμενη αναβάθμιση περιέχει σφάλματα, τότε στην τοπική εφαρμογή είναι δυνατόν να μην εφαρμοστεί αναβάθμιση σε αντίθεση με την διαδικτυακή εφαρμογή που δεν μπορεί να αποτραπεί η ενέργεια αυτή.

## 1.2 HTML

Η HTML είναι το ακρωνύμιο των λέξεων HyperText Markup Language, δηλαδή Γλώσσα Σήμανσης Υπερκειμένου, η οποία είναι η βασικότερη γλώσσα προγραμματισμού που χρησιμοποιείται για την κατασκευή σελίδων του Παγκοσμίου Ιστού.

Η HTML αποτελείται από κανόνες που βοηθούν στη διαμόρφωση της εμφάνισης και του περιεχομένου μιας σελίδας του Διαδικτύου. Ουσιαστικά, δεν είναι γλώσσα προγραμματισμού, αλλά γλώσσα με την οποία περιγράφονται οι ιδιότητες των στοιχείων που αποτελούν μια ιστοσελίδα. Η HTML χρησιμοποιεί ειδικές ετικέτες (tags) για να δομήσει το περιεχόμενο ενός εγγράφου html και να το εμφανίσει σε έναν φυλλομετρητή διαδικτύου (browser). Πολλοί προγραμματιστές λαμβάνουν υπόψη τους

τα web standards (<https://www.w3.org/>) που υπάρχουν για να υλοποιήσουν σωστή συγγραφή της HTML.

Τα HTML αρχεία πρέπει να γράφονται σε προγράμματα τα οποία είναι συντάκτες κειμένων που μπορούν να αποθηκεύσουν αρχεία κειμένου σε μορφή ASCII με επέκταση αρχείου .html ή .htm. Ένα από τα βασικά προγράμματα που θα χρησιμοποιήσω στην παρούσα πτυχιακή για να παρουσιάσω κάποια παραδείγματα είναι το πρόγραμμα notepad++. Μπορούμε να γράψουμε απευθείας κώδικα HTML ή μπορεί να δημιουργηθεί αυτόματα από κάποια προγράμματα ή πλατφόρμες κατασκευής ιστοσελίδων όπως είναι το WordPress. Εκεί οι προγραμματιστές χρησιμοποιούν διάφορα εργαλεία που τους παρέχει το WordPress για να σχεδιάσουν την ιστοσελίδα τους.

### 1.2.1 Ετικέτες/tags της HTML

Τα αρχεία HTML αποτελούνται από τα κείμενο της ιστοσελίδας και τα HTML tags, που επισημαίνουν τη μορφοποίηση των ιστοσελίδων. Η μεγαλύτερη πλειοψηφία των HTML tags παρουσιάζονται κατά ζεύγη, όπου το πρώτο αποτελεί την αρχή, ενώ το δεύτερο το τέλος του, περικλείοντας το κείμενο που επηρεάζουν.

Π.χ.: <TagName> κείμενο </TagName >

Εκτός από τις ετικέτες που παρουσιάζονται ως ζεύγη, ορισμένες είναι μονομελείς, ενώ άλλες περιέχουν επιπλέον πληροφορίες και κείμενο μέσα στα σύμβολα <>.

Π.χ. η <hr> η οποία δημιουργεί μια οριζόντια γραμμή.

Στον πίνακα αναφέρονται οι κύριες ετικέτες που χρησιμοποιεί η HTML για την περιγραφή της συνολικής δομής μιας ιστοσελίδας:

Ετικέτα	Περιγραφή
<html></html>	Με την ετικέτα αυτή ορίζουμε την αρχή και το τέλος μίας HTML σελίδας.
<head></head>	Ορίζει το τμήμα εκείνο της ιστοσελίδας που περιέχει πληροφορίες σχετικά με το html έγγραφο. Περιέχει στοιχεία όπως τον τίτλο και τους συνδέσμους για τα css και js αρχεία.

<code>&lt;body&gt;&lt;/body&gt;</code>	Με την ετικέτα αυτή ορίζουμε το περιεχόμενο της HTML σελίδας.
<code>&lt;title&gt;&lt;/title&gt;</code>	Με την ετικέτα αυτή ορίζουμε τον τίτλο της HTML σελίδας.
<code>&lt;p&gt;&lt;/p&gt;</code>	Με την ετικέτα αυτή ορίζουμε την παράγραφο.
<code>&lt;br&gt;</code>	Με την ετικέτα αυτή κάνουμε αλλαγή γραμμής.
<code>&lt;img&gt;</code>	Η ετικέτα αυτή χρησιμοποιείται για να κάνουμε εισαγωγή εικόνας σε μία HTML σελίδα.
<code>&lt;a href=' '&gt;&lt;/a&gt;</code>	Η ετικέτα αυτή χρησιμοποιείται για να συνδέσουμε τις σελίδες μεταξύ τους ή να μεταφερθούμε σε έναν εξωτερικό σύνδεσμο.

Table 1: Βασικές ετικέτες HTML

Παρακάτω θα παρουσιαστούν μερικά παραδείγματα, όπου θα γίνει χρήση των html tags:

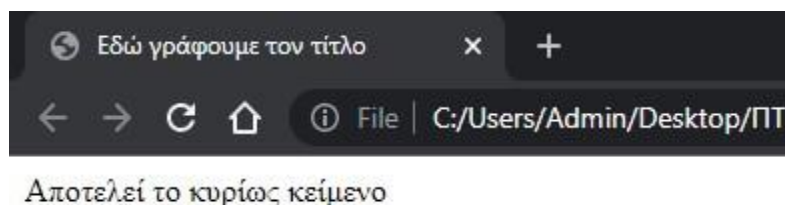
Ο κώδικας του προγράμματος είναι ο εξής:

```

1  <html>
2
3  <head>
4      <title>Εδώ γράφουμε τον τίτλο</title>
5  </head>
6
7  <body>
8      <p>Αποτελεί το κυρίως κείμενο</p>
9  </body>
10
11 </html>
12
13

```

Το αποτέλεσμα του κώδικα είναι ο εξής:



Για τον ορισμό των κεφαλίδων σε ένα HTML έγγραφο χρησιμοποιούνται οι κεφαλίδες (headings). Ένα HTML έγγραφο διαθέτει 6 διαφορετικές κεφαλίδες, οι οποίες παρουσιάζονται παρακάτω:

`<h1>` Παράδειγμα 1 `</h1>`

`<h2>` Παράδειγμα 2 `</h2>`

<h3> Παράδειγμα 3 </h3>

<h4> Παράδειγμα 4 </h4>

<h5> Παράδειγμα 5 </h5>

<h6> Παράδειγμα 6 </h6>

Η HTML ετικέτα <a></a> χρησιμοποιείται για τη δημιουργία ενός συνδέσμου. Τέτοιοι σύνδεσμοι μπορεί να είναι εξωτερικοί, οι οποίοι οδηγούν σε έναν άλλο σύνδεσμο ή μπορεί να είναι και εσωτερικοί, οι οποίοι χρησιμοποιούνται για τα σημεία που βρίσκονται στην ίδια HTML σελίδα. Ένα τέτοιο tag παρουσιάζεται παρακάτω:

```
<a href= 'https://www.google.gr'>Google</a>
```

Μια από τις ιδιότητες που μπορεί να πάρει η <a> ετικέτα είναι η ιδιότητα href. Αυτή η ιδιότητα παίρνει ως όρισμα το υπερκείμενο στο οποίο δείχνει ο σύνδεσμος. Στον χρήστη εμφανίζεται μόνο το δεύτερο μέρος, το «Google» και όταν κάνουμε κλικ επάνω στο κείμενο, τότε ο φυλλομετρητής χρησιμοποιεί το πρώτο μέρος, το «https://www.google.gr» για να μεταφερθεί στον συγκεκριμένο σύνδεσμο.

Μια HTML ετικέτα που δεν έχει τέλος, είναι η ετικέτα <img>. Η ετικέτα αυτή έχει πολλές ιδιότητες. Μία απ' αυτές είναι και η ιδιότητα src. Η ιδιότητα αυτή παίρνει ως όρισμα την εικόνα που θέλουμε να εισάγουμε γραμμένο μέσα σε εισαγωγικά. Το όρισμα αυτό μπορεί να είναι είτε μια διεύθυνση που αντιστοιχεί σε μια φωτογραφία, είτε εάν έχουμε μια εικόνα με όνομα image.jpg στο ίδιο φάκελο με το αρχείο HTML να το συμπεριλάβουμε ως εξής: src= 'image.jpg'

Η εισαγωγή βίντεο και ήχου σε μία HTML σελίδα, πραγματοποιείται με την ετικέτα <iframe></iframe>. Η ετικέτα αυτή υποστηρίζει μεταξύ των άλλων ιδιότητες καθορισμού μεγέθους. Ένας άλλος τρόπος ενσωμάτωσης βίντεο και ήχου είναι με την χρήση της HTML ετικέτας <div> σε συνδυασμό με την γλώσσα προγραμματισμού JavaScript.



## 1.3 CSS

Η CSS είναι το ακρωνύμιο των λέξεων Cascading Style Sheets. Είναι μία γλώσσα η οποία περιέχει κανόνες για τον έλεγχο της εμφάνισης ενός HTML εγγράφου. Συνδυάζεται πάντοτε με την HTML, η οποία δημιουργήθηκε για να δομεί το περιεχόμενο ενός HTML εγγράφου και όχι να περιέχει ετικέτες μορφοποίησης. Στην έκδοση HTML 4.0, όλη η μορφοποίηση αφαιρέθηκε από το HTML έγγραφο και αποθηκεύτηκε σε χωριστό αρχείο CSS. Όλα τα στοιχεία που αφορούν την μορφοποίηση του εγγράφου αποθηκεύονται σε εξωτερικό αρχείο με κατάληξη .css. Με ένα τέτοιο εξωτερικό css αρχείο, είναι δυνατή η μορφοποίηση ολόκληρης της ιστοσελίδας, τροποποιώντας μόνο ένα αρχείο. Για να χρησιμοποιηθεί το σύνολο των κανόνων CSS αρκεί να γίνει η αντιστοίχιση του css αρχείου (π.χ style.css) μέσα στο HTML έγγραφο χρησιμοποιώντας έναν από τους παρακάτω τρόπους:

-κάνοντας χρήση της ετικέτας link, μέσα στην ετικέτα head

```
<link rel='stylesheet' href='style.css'>
```

-με ενσωμάτωση των κανόνων στο έγγραφο HTML με την ετικέτα style

```
<style>
  h1 {
    color: black;
  }
</style>
```

Οι κανόνες CSS έχουν πολύ απλή σύνταξη, χρησιμοποιώντας έναν «επιλογή» (selector) και τους κανόνες που συμπεριλαμβάνονται μέσα σε αγκύλες, οι οποίοι κανόνες διαχωρίζονται από το ελληνικό ερωτηματικό «;» και αποτελούνται από ζεύγη ονομάτων και τιμών χωρισμένα με άνω κάτω τελεία «:».

```
p {
  color: black;
  border-bottom: 1px solid black;
  padding: 10px;
}
```

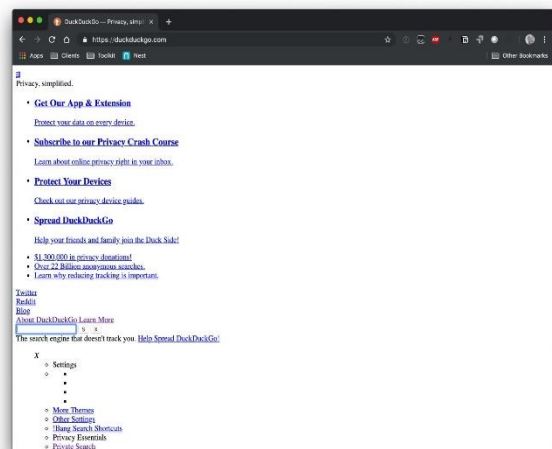
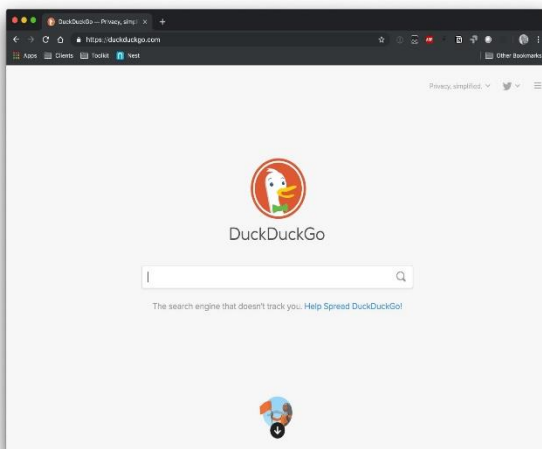
Το παραπάνω παράδειγμα καθορίζει ότι όλες οι παράγραφοι (p) θα έχουν χρώμα κειμένου κόκκινο, κάτω περίγραμμα πάχους 1 pixel και περιθώριο 10 pixel από κάθε πλευρά της παραγράφου.



This is a paragraph

---

Παρακάτω ακολουθεί ένα παράδειγμα με μια σελίδα που είναι μορφοποιημένη με την χρήση CSS (αριστερά) και ακριβώς η ίδια σελίδα χωρίς την χρήση CSS (δεξιά).



### 1.3.1 Επιλογείς (Selectors)

Οι επιλογείς (selectors) είναι ονόματα ετικετών (tags) της γλώσσας HTML (όπως h1, h2, h3...h6, p, a, div) αλλά συνήθως χρησιμοποιούνται σε συνδυασμό με κλάσεις (class) ή κωδικούς αναγνώρισης (id). Οι κλάσεις ομαδοποιούν κάποια στοιχεία, ενώ οι κωδικοί αναγνώρισης θα πρέπει να είναι μοναδικοί για κάθε στοιχείο.

- Παράδειγμα συνδυασμού επιλογέα (selector) με χρήση id:

```
p#title {
  color: red;
  font-size: 25px;
}
```

Πέρα από τους κανόνες για τις παραγράφους, ειδικά για την παράγραφο με κωδικό αναγνώρισης title, το χρώμα γραμμμάτων θα είναι κόκκινο και το μέγεθος 25pixel.

- Παράδειγμα συνδυασμού επιλογέα (selector) με χρήση κλάσης (class):

```
p.test {  
  padding-left: 20px;  
  border-left: 2px;  
}
```

Πέρα από τους κανόνες για τις παραγράφους, για όλες τις παραγράφους με όνομα κλάσης test θα προστεθεί περιθώριο 20pixel αριστερά και περίγραμμα 2pixel αριστερά.

## 1.4 JavaScript

Η JavaScript είναι μία γλώσσα scripting (προγραμματισμού), η οποία χρησιμοποιείται για να εισάγουμε διαδραστικότητα στα html έγγραφα. Είναι εμνηυτική γλώσσα. Αυτό σημαίνει πως εκτελεί τα scripts χωρίς να απαιτείται η διαδικασία της σύνταξης. Πολλοί έχουν την τάση να μπερδεύουν την JavaScript με την Java, αλλά μιλάμε για δύο διαφορετικές γλώσσες προγραμματισμού.



Η JavaScript ως client-side γλώσσα προγραμματισμού, σημαίνει ότι εκτελείται στον φυλλομετρητή του επισκέπτη. Δεν συνίσταται για την κατασκευή μίας εξ' ολοκλήρου ιστοσελίδας, απλά λειτουργεί «συμπληρωματικά». Επιτρέπει στον χρήστη να κάνει δυναμικά πράγματα στην ιστοσελίδα.

Με την JavaScript μπορούμε να εκτελέσουμε κάποιες ενέργειες όταν συμβαίνει ένα γεγονός, όπως για παράδειγμα όταν ο χρήστης κλικάρει ή περνάει το ποντίκι του πάνω από ένα συγκεκριμένο σημείο του HTML εγγράφου, να εκτελείται κάποιο script και ο επισκέπτης να λαμβάνει τα αντίστοιχα αποτελέσματα. Με την χρήση JavaScript μπορούμε να τροποποιήσουμε τα περιεχόμενα ενός HTML στοιχείου. Επίσης, μπορούμε να την χρησιμοποιήσουμε για την επικύρωση μιας φόρμας προτού τα

στοιχεία της υποβληθούν στον server. Με αυτόν τον τρόπο γλιτώνουμε και τον server με επιπλέον διεργασίες.

Έστω ότι σε ένα HTML έγγραφο έχουμε μια φόρμα. Θέλουμε όμως να μην επιτρέπουμε στον χρήστη στο πεδίο με το όνομα χρήστη να εισάγει παραπάνω από 20 χαρακτήρες. Επιπλέον, θέλουμε ο χρήστης να εισάγει ένα έγκυρο email της μορφής [someone@gmail.com](mailto:someone@gmail.com). Έτσι μόλις πατήσουμε το κουμπί «Αποστολή», η JavaScript θα ελέγξει αν ο χρήστης έκανε έγκυρη εισαγωγή δεδομένων με βάση τους παραπάνω περιορισμούς. Με αυτόν τον τρόπο, η διαδικασία είτε θα συνεχιστεί είτε θα του εμφανίσει μήνυμα λάθους, για παράδειγμα: Μη έγκυρο email.

Για να εκτελεστεί η JavaScript είναι απαραίτητη η ύπαρξη των HTML εγγράφων. Επίσης, για να εκτελεστεί θα πρέπει να εισαχθεί μέσα σε ένα HTML έγγραφο. Για να δημιουργηθεί δυναμικό περιεχόμενο σε ένα έγγραφο, η JavaScript χρησιμοποιεί τα HTML στοιχεία. Τέτοιες αλλαγές συνήθως επηρεάζουν τις ιδιότητες ενός στοιχείου, το περιεχόμενό του ακόμα και το style του (css). Δεν χρειαζόμαστε κάποιον Web Server για να εκτελεστεί η JavaScript, αρκεί να ανοίξουμε τοπικά το αρχείο που περιλαμβάνει τον script κώδικα και θα εκτελεστεί αυτόματα κατά την φόρτωση του αρχείου από τον φυλλομετρητή.

Ακολουθούν τρία είδη script που μπορούμε να χρησιμοποιήσουμε μέσα σε ένα έγγραφο html:

1. Εμφωλευμένο script
2. Εσωτερικό script
3. Εξωτερικό script

#### 1.4.1 Εμφωλευμένο Script

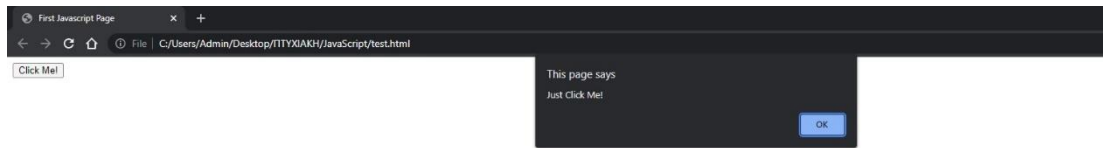
Για να εισάγουμε ένα εμφωλευμένο script μέσα σε ένα html έγγραφο, θα πρέπει να εισάγουμε τον κώδικα JavaScript μέσα σε ένα ήδη υπάρχον html tag/ετικέτα. Με αυτό τον τρόπο δεν χρειάζεται να χρησιμοποιηθούν οι ετικέτες `<script></script>`.

Κώδικας:



```
test.html
1 <html>
2 <head>
3   <meta charset="utf-8">
4   <title>First Javascript Page</title>
5 </head>
6 <body>
7   <input type='button' value='Click Me!' onClick="alert('Just Click Me!')">
8 </body>
9 </html>
10
11
12
```

## Αποτέλεσμα:



Όταν γίνεται χρήση εισαγωγικών μέσα σε εισαγωγικά, όπως στο παραπάνω παράδειγμα, τα εσωτερικά εισαγωγικά θα πρέπει να είναι διαφορετικά από τα εξωτερικά εισαγωγικά. Στο συγκεκριμένο παράδειγμα, χρησιμοποιούμε ως εξωτερικά εισαγωγικά τα διπλά και ως εσωτερικά εισαγωγικά τα μονά. Υπάρχει η δυνατότητα να χρησιμοποιηθούν αντίθετα, αλλά σε καμία περίπτωση τα ίδια, για να μην υπάρξει κάποια σύγχυση στο πρόγραμμα περιήγησης. Στο παραπάνω παράδειγμα, χρησιμοποιείται το γεγονός «onClick», το οποίο ενεργοποιείται όταν κάποιος πατήσει πάνω στο συγκεκριμένο html στοιχείο. Όταν πατηθεί αυτό το στοιχείο, τότε θα εκτελεστεί ο κώδικας που βρίσκεται μέσα στα εισαγωγικά «"», «"».

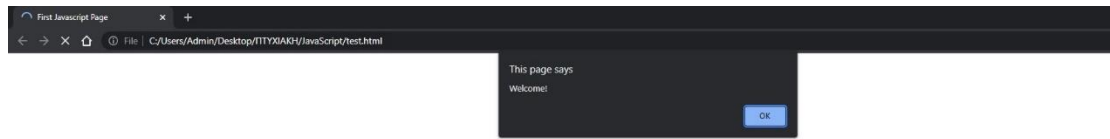
### 1.4.2 Εσωτερικό Script

Για την εισαγωγή ενός ή περισσότερων script σε ένα html αρχείο θα πρέπει να χρησιμοποιηθεί η ετικέτα αρχή και τέλους `<script></script>`. Στο παρακάτω παράδειγμα το script θα εκτελεστεί κατά την φόρτωση της ιστοσελίδας.

Κώδικας:

```
1 <html>
2 <head>
3   <meta charset="utf-8">
4   <title>First Javascript Page</title>
5 </head>
6 <body>
7   <script>
8     alert("Welcome!");
9   </script>
10 </body>
11 </html>
12
13
14
```

## Αποτέλεσμα:



Με την εντολή «alert» επιτρέπουμε στον φυλλομετρητή να εμφανίσει ένα αναδυόμενο παράθυρο με ένα μήνυμα, το οποίο μήνυμα καθορίζεται από το κείμενο που εισάγεται ανάμεσα στις παρενθέσεις.

### 1.4.3 Εξωτερικό Script

Για να είναι ο κώδικας μας πιο οργανωμένος, τότε όλα τα script θα πρέπει να γράφονται σε ξεχωριστό αρχείο με κατάληξη .js. Για να εισάγουμε αυτά τα script στο html αρχείο, θα πρέπει να χρησιμοποιήσουμε την ετικέτα αρχή και τέλους `<script></script>` με την παράμετρο `src` στην οποία τοποθετούμε το μονοπάτι το οποίο βρίσκεται το αρχείο με κατάληξη js.

#### Κώδικας:

```
1 <html>
2 <head>
3   <meta charset="utf-8">
4   <title>First Javascript Page</title>
5   <script type='text/javascript' src='index.js'></script>
6 </head>
7 <body>
8 </body>
9 </html>
10
11
12
13
```

#### Αποτέλεσμα:



## 1.5 Bootstrap

Το Bootstrap δημιουργήθηκε το έτος 2011 από τους προγραμματιστές τους twitter τους Mark Otto και Jacob Thornton. Αποτελεί ένα από τα πιο δημοφιλή Web Design Frameworks και είναι ευρέως χρησιμοποιούμενο σε όλο τον κόσμο.

Τα κύρια χαρακτηριστικά του είναι τα εξής:

1. Παρέχει ποικιλία κλάσεων (CSS classes) τα οποία βοηθούν στην σχεδίαση του ιστότοπου.
2. Προσφέρει έτοιμα συστατικά (ready-made web components) που βοηθούν στην δημιουργία λειτουργικών και σύγχρονων ιστότοπων.
3. Υποστηρίζει τη συνεπή τυπογραφία (consistent typography), η οποία αποτελεί την συνεπή δομή του κειμένου σε μία ιστοσελίδα.
4. Τέλος, έχει σχεδιαστεί για τη παροχή responsive, mobile-first websites.

Ο καθένας μπορεί να χρησιμοποιήσει το Bootstrap και αυτό οφείλεται στην ευκολία χρήσης του. Για να χρησιμοποιηθούν τα αρχεία του bootstrap, αρκεί μια απλή εισαγωγή στον κώδικα του css και js αρχείου. Περιλαμβάνει πολλά έτοιμα στοιχεία για να μπουκ στην ιστοσελίδα καθώς επίσης το συγκεκριμένο framework διανέμεται δωρεάν. Για να χρησιμοποιήσει κανείς το Bootstrap αρκεί να εισάγει τις παρακάτω βιβλιοθήκες του στον κώδικα, προτού φορτώσει τα υπόλοιπα css αρχεία.

```
<link  
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.2/dist/css/boot  
strap.min.css" rel="stylesheet" integrity="sha384-  
uWxY/CJNBR+1zjPwMfnSnVxwRheevXITnMqoEIEG1LJrdI0GLVs/9cVSyPYXdcSF  
" crossorigin="anonymous">
```

Για την JavaScript αρκεί να αντιγράψετε τις παρακάτω βιβλιοθήκες.

```
<script  
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.2/dist/js/bootst  
rap.bundle.min.js" integrity="sha384-  
kQtW33rZJAHjgefvyhyyzCGF3C5TFyBQBA13V1RKPf4uH+bwyzQxZ6CmMZHMNBefJ  
" crossorigin="anonymous"></script>
```

```
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd
/popper.min.js" integrity="sha384-
7+zCNj/IqJ95wo16oMtfsKbZ9ccEh31eOz1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB
" crossorigin="anonymous"></script>
```

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.2/dist/js/bootst
rap.min.js" integrity="sha384-
PsUw7Xwds7x08Ew3exXhgzbuEYmA2xnwc8BuD6SEr+UmEHLX8/MCltYEodzWA4u
" crossorigin="anonymous"></script>
```

Ακολουθεί ένα παράδειγμα template με χρήση Bootstrap.

```
index.html
1 <!doctype html>
2 <html lang="en">
3 <head>
4 <!-- Required meta tags -->
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7
8 <!-- Bootstrap CSS -->
9 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-0kx/CNBB+1z/PNafcoYoVwBewXITuMqoE1eD1Jrd1001Va/90ByFYXoRf" crossorigin="anonymous">
10
11 <title>Hello, world!</title>
12 </head>
13 <body>
14 <h1>Hello, world!</h1>
15
16 <!-- Optional JavaScript: choose one of the two! -->
17
18 <!-- Option 1: Bootstrap Bundle with Popper -->
19 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-kqW3380ARjpeVvya00f3C57FyqB413V1R0Tf401bWysq2400M808R7" crossorigin="anonymous"></script>
20
21 <!-- Option 2: Separate Popper and Bootstrap JS -->
22 <!--
23 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js" integrity="sha384-7+zCNj/IqJ95wo16oMtfsKbZ9ccEh31eOz1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB" crossorigin="anonymous"></script>
24 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.2/dist/js/bootstrap.min.js" integrity="sha384-PsUw7Xwds7x08Ew3exXhgzbuEYmA2xnwc8BuD6SEr+UmEHLX8/MCltYEodzWA4u" crossorigin="anonymous"></script>
25 -->
26 </body>
27 </html>
```

Οι διαφορές των Bootstrap και CSS παρουσιάζονται στον παρακάτω πίνακα:

CSS	BOOTSTRAP
Αντιπροσωπεύει την εμφάνιση του περιεχομένου (γραμματοσειρά, χρώμα, περιθώριο, επένδυση κ.ά)	Είναι ένα δωρεάν και Open Source CSS Framework με σκοπό την ανάπτυξη ανταποκρίσιμου ιστότοπου.
Δεν διαθέτει grid system	Βασίζεται στο grid system
Λειτουργεί στο CSS3 που είναι η τελευταία έκδοση του CSS.	Εργάζεται επί του παρόντος στο Bootstrap 5.1 που είναι η τελευταία έκδοση του Bootstrap.
Δεν παρέχει responsive design	Μπορούμε να σχεδιάσουμε έναν ιστότοπο ή ιστοσελίδες που να είναι responsive.

Είναι περισσότερο πολύπλοκο από το Bootstrap επειδή δεν υπάρχουν προκαθορισμένες κλάσεις.	Είναι εύκολο να κατανοηθεί και είναι προσχεδιασμένο.
Στο CSS, πρέπει να γράψουμε κώδικα από την αρχή.	Στο Bootstrap, μπορούμε να προσθέσουμε προκαθορισμένη κλάση στον κώδικα χωρίς να γράψουμε κώδικα.

Table 2: Διαφορές CSS και Bootstrap

## 1.6 jQuery

Για να απλοποιηθεί η διαδικασία συγγραφής κώδικα σε JavaScript, σχεδιάστηκε η jQuery, η οποία είναι μία από τις πιο δημοφιλείς βιβλιοθήκες JavaScript και χρησιμοποιείται από μεγάλες εταιρείες όπως η Google.

Με την βιβλιοθήκη της jQuery μπορεί κανείς να διαμορφώσει τις HTML σελίδες, καθώς και τα CSS αρχεία. Επίσης μπορεί να δημιουργήσει JavaScript events και animations. Μια ακόμα δυνατότητα είναι η χρήση της τεχνολογίας AJAX που κάνει την jQuery να λειτουργεί ασύγχρονα καθιστώντας την πολύ γρήγορη και ευέλικτη.

Η βιβλιοθήκη της jQuery καθιστά τον προγραμματισμό σε JavaScript αρκετά εύκολο χρησιμοποιώντας εύκολους μηχανισμούς. Επίσης διαθέτει μία μεγάλη κοινότητα υποστήριξης που αποτελεί τον ακρογωνιαίο λίθο για την ανάπτυξη της, μιας και είναι ένα από τα μεγαλύτερα Open Source έργα λογισμικού. Το βασικό πακέτο της jQuery κυμαίνεται μόλις στα 20KB, γεγονός που αποδεικνύει την απλότητά της. Τέλος, είναι εφικτό να χρησιμοποιηθεί το μεγαλύτερο ποσοστό των λειτουργιών που προσφέρει η JavaScript, από βασικές λειτουργίες εμφάνισης ή απόκρυψης HTML αρχείων έως και AJAX αιτήματα.

Για να μπορέσει κανείς να χρησιμοποιήσει την βιβλιοθήκη της jQuery στην εφαρμογή του, μπορεί να ακολουθήσει τα παρακάτω βήματα.:

1. Να κατεβάσει τη βιβλιοθήκη από την επίσημη σελίδα της jQuery, την [jquery.com](http://jquery.com) και πιο συγκεκριμένα να κατεβάσει το minified αρχείο, ή
2. Να χρησιμοποιήσει τη βιβλιοθήκη μέσω CDN. Το CDN είναι τα αρχικά των Content Delivery Network, το οποίο είναι ένα δίκτυο γεωγραφικά κατανεμημένων server που διανέμουν διάφορα scripts και αρχεία όπως είναι το Bootstrap, η jQuery κ.ά. Στο δίκτυο αυτό, τα αρχεία και scripts που μας ενδιαφέρουν τα λαμβάνουμε από τον κόμβο που βρίσκεται πιο κοντά σε εμάς.



Σε περίπτωση που ο χρήστης έχει επισκεφτεί ήδη σελίδες που χρησιμοποιούν το ίδιο CDN, τότε το αρχείο έχει αποθηκευτεί ήδη στο φυλλομετρητή και δε χρειάζεται να φορτωθεί ξανά. Έτσι, υπάρχει μεγαλύτερη ταχύτητα και επιπλέον στέλνουμε λιγότερα δεδομένα από τον δικό μας εξυπηρετητή. Τέτοια δίκτυα CDN διαθέτουν οι μεγάλες εταιρείες όπως είναι η Google και η Microsoft.

Η βιβλιοθήκη της jQuery δεν είναι τίποτα άλλο από ένα αρχείο JavaScript. Όπως είδαμε και παραπάνω, υπάρχουν δύο τρόποι για να χρησιμοποιήσει κανείς την βιβλιοθήκη, είτε κατεβάζοντας την από την επίσημη σελίδα, είτε χρησιμοποιώντας το CDN. Και στις δύο περιπτώσεις η βιβλιοθήκη θα πρέπει να δηλωθεί. Η δήλωση αυτή γίνεται χρησιμοποιώντας την ετικέτα αρχή και τέλους `<script></script>` όπως φαίνεται παρακάτω.

**1ος τρόπος:** όταν το αρχείο της βιβλιοθήκης βρίσκεται στον εξυπηρετητή μας

```
<script src="jquery-3.5.1.min.js"></script>
```

**2ος τρόπος:** όταν το αρχείο λαμβάνεται από CDN

```
<script src="http://ajax.aspnetcdn.com/ajax/jquery/jquery-3.5.1.min.js"></script>
```

Η δήλωση της βιβλιοθήκης θα πρέπει να βρίσκεται εντός της ετικέτας `<head></head>` ή να γράφεται ακριβώς πριν κλείσουμε την ετικέτα `</body>`.

### 1.6.1 Τρόπος λειτουργίας της jQuery

Η γενικότερη φιλοσοφία της jQuery, και γενικότερα της JavaScript, κάνει αναφορά στο ότι πρώτα γίνεται η επιλογή των διάφορων στοιχείων της σελίδας και μετά γίνεται η επεξεργασία μέσω των εντολών. Γενικότερα, με την JavaScript τροποποιούμε διάφορα στοιχεία τα οποία βρίσκονται στο αρχείο στο οποίο χρησιμοποιούμε ή καλούμε JavaScript. Για να λειτουργήσει η βιβλιοθήκη της jQuery θα πρέπει ο κώδικας JavaScript να βρίσκεται κάτω από την ετικέτα που την δηλώνεις. Αφού κάνετε τα παραπάνω ακολουθήστε το παρακάτω παράδειγμα:

```
<script type="text/javascript">
    $(document).ready(function(){
    });
</script>
```

Στον παραπάνω κώδικα, αποτυπώνεται η γνωστή εντολή της jQuery, η `.ready()`, με τη οποία εξασφαλίζεται πως όλα τα στοιχεία του αρχείου έχουν φορτωθεί, προτού ξεκινήσουν οι εντολές επεξεργασίας. Με αυτόν τον τρόπο βεβαιώνετε ότι όλα τα στοιχεία που καλείτε υπάρχουν.

Στη συνέχεια γράψτε την πρώτη σας εντολή:

```
<script type="text/javascript">
    $(document).ready(function(){
        $('p').css('color','red');
    });
</script>
```

Για να επιλέξουμε κάποια στοιχεία του αρχείου με την jQuery χρησιμοποιούμε τον επιλογέα της (\$) . Στον παραπάνω κώδικα, αφού έχει φορτώσει πλήρως η σελίδα, επιλέγουμε όλες τις παραγράφους του αρχείου και τροποποιούμε το χρώμα τους με την εντολή `.css()`, η οποία παίρνει δύο παραμέτρους. Η πρώτη είναι η ιδιότητα της css που θέλουμε να χρησιμοποιήσουμε και η δεύτερη παράμετρος η τιμή. Στο συγκεκριμένο παράδειγμα χρησιμοποιούμε το `color` με χρώμα `red`. Ο επιλογέας της jQuery χρησιμοποιείται με δύο τρόπους. Αν θέλουμε να επιλέξουμε τα στοιχεία του αρχείου χρησιμοποιώντας τις κλάσεις τους τότε θα χρησιμοποιήσουμε την τελεία (.) και μετά το όνομα της κλάσης. Αν θέλουμε να επιλέξουμε τα στοιχεία του αρχείου χρησιμοποιώντας τα ID τους τότε θα χρησιμοποιήσουμε την δέση (#) και μετά το όνομα του ID. Στο παράδειγμα που ακολουθεί χρησιμοποιούμε και τους δύο τρόπους.

```
<script type="text/javascript">

$(document).ready(function(){

    $('p').hide(); // κρύβει όλες τις παραγράφους
    $('p.test').hide(); //κρύβει όλες τις παραγράφους με class="test"
    $('#test').hide(); //κρύβει όλα τα στοιχεία με id="test"

});

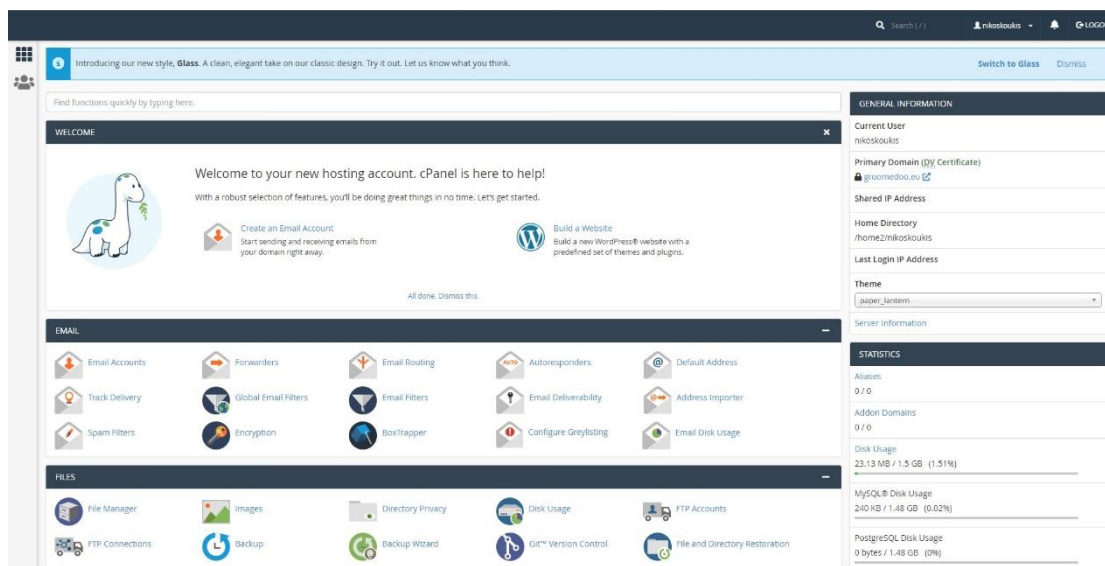
</script>
```

## 1.7 Cron Job

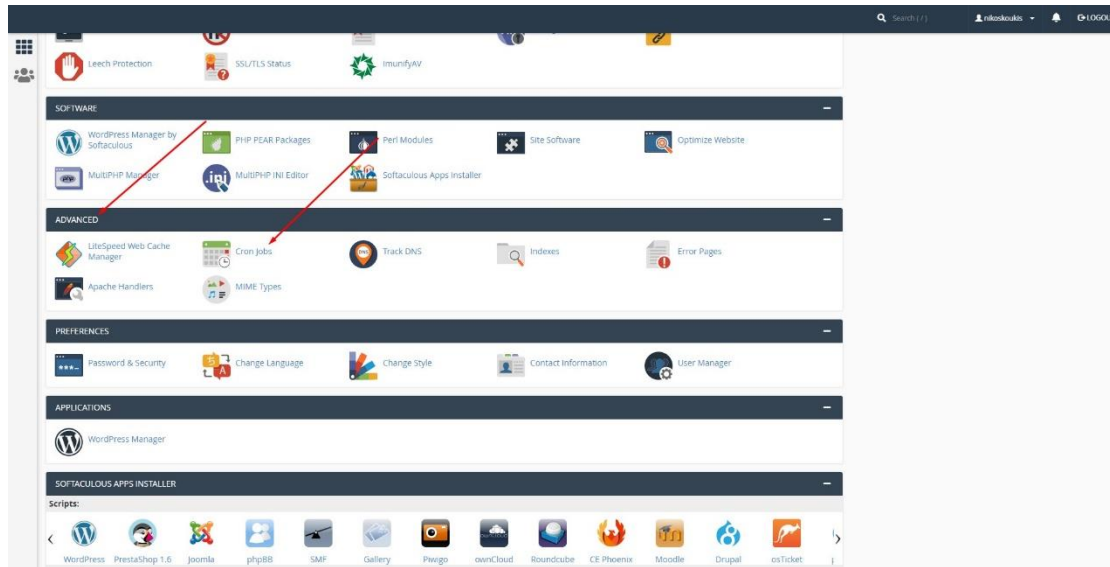
Οι εταιρείες Web Hosting που φιλοξενεί την ιστοσελίδα ή την εφαρμογή μας, μας παρέχει διάφορους πίνακες ελέγχου όπως το cPanel ή το Plesk. Σε αυτούς τους πίνακες ελέγχου μπορούμε να βρούμε την λειτουργία Cron Job. Το cron ουσιαστικά είναι μια εντολή Linux που χρησιμοποιείται για τον προγραμματισμό εργασιών που θα εκτελεστούν αυτόματα, σε κάποια συγκεκριμένη χρονική στιγμή στο μέλλον, χωρίς να χρειάζεται να κάνουμε κάτι εμείς.

Αν κάποιος θέλει να χρησιμοποιήσει το cPanel για να ορίσει ένα Cron Job, αρκεί να ακολουθήσει τα εξής βήματα:

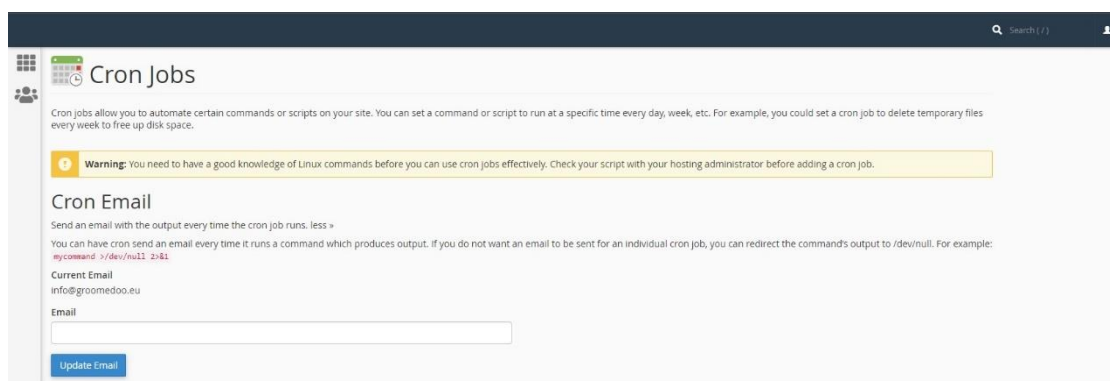
1. Αρχικά, θα πρέπει να συνδεθεί στο cPanel.



2. Από τον πίνακα ελέγχου, μεταβείτε στο μενού Advanced και κάντε κλικ στο Cron Jobs.



3. Στη συνέχεια, θα πρέπει να εισάγετε ένα email για να ενημερώνεστε για το πότε θα εκτελείται το cron και αν κατά τη διάρκεια της εκτέλεσης προκύψει κάποιο σφάλμα. Ένα το τρέχον email που έχει δηλωθεί είναι έγκυρο δε χρειάζεται να κάνετε κάτι. Αν κάποιος επιθυμεί να εισάγει ένα διαφορετικό email, πρέπει να το πληκτρολογήσει στο πεδίο φόρμας Email και να κάνει κλικ στην επιλογή Update Email.



4. Στην ενότητα με την ένδειξη Add New Cron Job, μπορεί κανείς να καθορίσει ποια εντολή θέλει να εκτελεστεί καθώς και τη συχνότητά της. Ξεκινάμε εισάγοντας την επιθυμητή εντολή στο κάτω πεδίο φόρμας με την ένδειξη Command. Μπορούμε να παραθέσουμε οποιαδήποτε εντολή Linux εδώ ή, εάν κάποιος έχει ένα αρχείο που θα ήθελε να τρέξει σε μια συγκεκριμένη στιγμή, απλώς δίνει τη διαδρομή στο αρχείο που θέλει να εκτελεστεί.

Update Email

## Add New Cron Job

PHP command examples:  
 General example:  
`/usr/local/bin/php /home2/nikoskoukis/public_html/path/to/cron/script`  
 Domain-specific example:  
`/usr/local/bin/ea-php99 /home2/nikoskoukis/domain_path/path/to/cron/script`  
 In the above example, replace "ea-php99" with the PHP version assigned to the domain you wish to use. Look in the MultiPHP Manager for the actual PHP version assigned to a domain.

**Common Settings**

Once Per Minute(\* \* \* \* \*)

Minute:  
 \*      Once Per Minute(\*)

Hour:  
 \*      Every Hour (\*)

Day:  
 \*      Every Day (\*)

Month:  
 \*      Every Month (\*)

Weekday:  
 \*      Every Day (\*)

Command:

Add New Cron Job

5. Στη συνέχεια, μεταβαίνουμε στο πεδίο φόρμας με την ένδειξη Common Settings και επιλέγουμε σε ποια συχνότητα θέλουμε να εκτελείται η εντολή από τις διαθέσιμες επιλογές που αναφέρονται στο αναπτυσσόμενο μενού. Αυτό θα ενημερώσει αυτόματα τις άλλες ρυθμίσεις ώρας σύμφωνα με την επιλογή σας.

Update Email

## Add New Cron Job

PHP command examples:  
 General example:  
`/usr/local/bin/php /home2/nikoskoukis/public_html/path/to/cron/script`  
 Domain-specific example:  
`/usr/local/bin/ea-php99 /home2/nikoskoukis/domain_path/path/to/cron/script`  
 In the above example, replace "ea-php99" with the PHP version assigned to the domain you wish to use. Look in the MultiPHP Manager for the actual PHP version assigned to a domain.

**Common Settings**

Once Per Minute(\* \* \* \* \*)

Minute:  
 \*      Once Per Minute(\*)

Hour:  
 \*      Every Hour (\*)

Day:  
 \*      Every Day (\*)

Month:  
 \*      Every Month (\*)

Weekday:  
 \*      Every Day (\*)

Command:

Add New Cron Job

6. Εναλλακτικά, μπορεί κανείς να επιλέξει να εισάγει την επαναλαμβανόμενη συχνότητα με μη αυτόματο τρόπο, προσαρμόζοντας τις μεμονωμένες ρυθμίσεις

ώρας. Τα λεπτά, η ώρα, ο μήνας και η ημέρα της εβδομάδας μπορούν να τροποποιηθούν για να επιτευχθούν τα επιδιωκόμενα αποτελέσματα.

Update Email

### Add New Cron Job

PHP command examples:  
General example:  
`/usr/local/bin/php /home2/nikoskoukis/public_html/path/to/cron/script`  
Domain-specific example:  
`/usr/local/bin/php99 /home2/nikoskoukis/domain_path/path/to/cron/script`  
In the above example, replace "ea-php99" with the PHP version assigned to the domain you wish to use. Look in the [MultiPHP Manager](#) for the actual PHP version assigned to a domain.

**Common Settings**  
Once Per Minute(\* \* \* \* \*)

Minute:  
\*      Once Per Minute(\*)

Hour:  
\*      Every Hour (\*)

Day:  
\*      Every Day (\*)

Month:  
\*      Every Month (\*)

Weekday:  
\*      Every Day (\*)

Command:  
`/usr/local/bin/php /home2/nikoskoukis/public_html/cron_job_update_appointment.php`

Add New Cron Job

7. Όταν ολοκληρώσουμε τις επιλογές μας, απλώς κάνουμε κλικ στο κουμπί Add New Cron Job για να ολοκληρωθεί η διαδικασία.

Update Email

### Add New Cron Job

PHP command examples:  
General example:  
`/usr/local/bin/php /home2/nikoskoukis/public_html/path/to/cron/script`  
Domain-specific example:  
`/usr/local/bin/php99 /home2/nikoskoukis/domain_path/path/to/cron/script`  
In the above example, replace "ea-php99" with the PHP version assigned to the domain you wish to use. Look in the [MultiPHP Manager](#) for the actual PHP version assigned to a domain.

**Common Settings**  
Once Per Minute(\* \* \* \* \*)

Minute:  
\*      Once Per Minute(\*)

Hour:  
\*      Every Hour (\*)

Day:  
\*      Every Day (\*)

Month:  
\*      Every Month (\*)

Weekday:  
\*      Every Day (\*)

Command:  
`/usr/local/bin/php /home2/nikoskoukis/public_html/cron_job_update_appointment.php`

Add New Cron Job

Και τώρα έχουμε προγραμματίσει με επιτυχία μια εντολή χρησιμοποιώντας το εργαλείο Cron Jobs στο cPanel. Εάν υπάρχουν άλλες εντολές που πρέπει να ρυθμιστούν, απλώς επαναλαμβάνεται αυτή τη διαδικασία για τυχόν άλλες εντολές.

## 1.8. PHP

Η PHP είναι μια scripting γλώσσα προγραμματισμού ανοιχτού κώδικα. Η PHP σχεδιάστηκε για την ανάπτυξη Web εφαρμογών αλλά και για δυναμικών ιστοσελίδων. Παρακάτω παρουσιάζεται ένα μικρό παράδειγμα κώδικα σε PHP:

```
<html>
  <head>
    <title>First Example</title>
  </head>
  <body>
    <?php
      echo "Hello World!";
    ?>
  </body>
</html>
```

Στο παραπάνω παράδειγμα διακρίνουμε πως η PHP γράφεται μέσα στην HTML. Ουσιαστικά είναι ένα HTML script με ενσωματωμένο PHP κώδικα που στην συγκεκριμένη περίπτωση, εμφανίζει το κείμενο «Hello World». Για να γράψουμε PHP θα πρέπει να χρησιμοποιήσουμε τις ειδικές ετικέτες αρχής και τέλους. (<?php ... ?>).

Η γλώσσα PHP είναι μια server-side γλώσσα προγραμματισμού. Αν και γλώσσα scripting, εκτελείται στον εξυπηρετητή σε αντίθεση με άλλες γλώσσες scripting όπως η JavaScript, που εκτελείται στη μεριά του client. Για να ξεκινήσει κανείς να γράφει PHP δε χρειάζεται να διαθέτει εξειδικευμένες γνώσεις. Είναι πολύ απλή για αρχάριους προγραμματιστές αλλά ταυτόχρονα μπορεί να προσφέρει προηγμένες δυνατότητες σε επαγγελματίες προγραμματιστές. Επίσης, δεν απαιτείται η εγκατάσταση κάποιου λογισμικού για να γράψει κανείς PHP, αρκεί ένας συντάκτης κειμένου.

Καθώς η PHP είναι server-side scripting γλώσσα προγραμματισμού, μπορεί να συλλέξει δεδομένα, να παράξει δυναμικό περιεχόμενο καθώς και να επεξεργαστεί sessions και cookies. Υπάρχουν τρεις διαφορετικές περιπτώσεις που χρησιμοποιείται ένα PHP script:

1. Server-side scripting: Είναι ένα από τα σημαντικότερα πεδία για την PHP. Για να δουλέψει αυτό χρειάζονται 3 βασικά πράγματα. Έναν PHP μεταγλωττιστή, έναν Webserver και έναν φυλλομετρητή (browser). Αρχικά πρέπει να ξεκινήσει ο webserver και να γίνει σύνδεση με μια εγκατάσταση της PHP. Για να δείτε το

αποτέλεσμα αρκεί ο web browser στον οποίο ο server θα σερβίρει την PHP σελίδα.

2. Command line scripting. Για να τρέξει κανείς PHP script μπορεί να αρκестεί μόνο και σε έναν PHP μεταγλωττιστή, χωρίς να χρειάζεται κάποιος webserver ή web browser. Τέτοια scripts είναι ιδανικά να εκτελούνται με την χρήση cron jobs (σε Linux) ή με τον Task Scheduler (στα Windows).
3. Client-side GUI εφαρμογών: Η PHP ίσως να μην είναι και η καλύτερη επιλογή γλώσσας προγραμματισμού για συγγραφή παραθυριακών εφαρμογών, αλλά προσφέρει προηγμένα χαρακτηριστικά για client-side εφαρμογές, καθώς υπάρχει και η δυνατότητα συγγραφής cross-platform εφαρμογές.

Η PHP μπορεί να χρησιμοποιηθεί σε όλα τα λειτουργικά συστήματα όπως Linux, Windows, Max Os κ.ά. Η PHP επίσης υποστηρίζεται από πολλούς εξυπηρετητές όπως τους Apache HTTP Server, Internet Information Service, LIGHTTPD, NGINX κ.ά. Με αυτόν τον τρόπο έχουμε την ελευθερία να επιλέξουμε ένα λειτουργικό σύστημα και έναν Web Server. Επίσης η PHP δίνει την δυνατότητα χρήσης αντικειμενοστρεφή προγραμματισμού (object oriented).

Με την PHP μπορεί κανείς να εξάγει αρχεία PDF καθώς και XHTML, XML αρχεία. Επίσης ένα από τα σημαντικότερα χαρακτηριστικά της PHP είναι ότι υποστηρίζεται από ένα μεγάλο σύνολο βάσεων δεδομένων. Μερικές από τις βάσεις δεδομένων που υποστηρίζεται η PHP είναι οι εξής:

- |                              |                                 |                                 |                               |
|------------------------------|---------------------------------|---------------------------------|-------------------------------|
| ➤ <a href="#">Access</a>     | ➤ <a href="#">Borland iBase</a> | ➤ <a href="#">OCI805</a>        | ➤ <a href="#">Postgres</a>    |
| ➤ <a href="#">Ado</a>        | ➤ <a href="#">Informix</a>      | ➤ <a href="#">OCI8Po</a>        | ➤ <a href="#">Postgres64</a>  |
| ➤ <a href="#">Ado_Access</a> | ➤ <a href="#">Informix72</a>    | ➤ <a href="#">ODBC</a>          | ➤ <a href="#">Postgres7</a>   |
| ➤ <a href="#">Ado_MSSql</a>  | ➤ <a href="#">LDAP</a>          | ➤ <a href="#">ODBC_MSSql</a>    | ➤ <a href="#">Postgres8</a>   |
| ➤ <a href="#">DB2</a>        | ➤ <a href="#">MSSql</a>         | ➤ <a href="#">ODBC_Oracle</a>   | ➤ <a href="#">Postgres9</a>   |
| ➤ <a href="#">ODBC_DB2</a>   | ➤ <a href="#">MSSqlPO</a>       | ➤ <a href="#">ODBTP</a>         | ➤ <a href="#">SAPDB</a>       |
| ➤ <a href="#">Vfp</a>        | ➤ <a href="#">MySql</a>         | ➤ <a href="#">ODBTP_Unicode</a> | ➤ <a href="#">SqlAnywhere</a> |
| ➤ <a href="#">FBSql</a>      | ➤ <a href="#">MySqli</a>        | ➤ <a href="#">Oracle</a>        | ➤ <a href="#">SQLite</a>      |
| ➤ <a href="#">iBase</a>      | ➤ <a href="#">MySQLt/MaxSql</a> | ➤ <a href="#">Netezza</a>       | ➤ <a href="#">SQLitePO</a>    |
| ➤ <a href="#">Firebird</a>   | ➤ <a href="#">OCI8</a>          | ➤ <a href="#">PDO</a>           | ➤ <a href="#">SyBase</a>      |



## 1.9 MySQL

Τα Συστήματα Διαχείρισης Βάσεων Δεδομένων (DBMS) χρησιμοποιούν την γλώσσα προγραμματισμού MySQL, η οποία είναι ανοικτού κώδικα. Για να υπάρχει ένας κοινός άξονας ανάμεσα σε αυτά τα συστήματα, δημιουργήθηκε η γλώσσα SQL (Structured Query Language), η οποία με διάφορες εντολές μας επιτρέπει να ανακτήσουμε δεδομένα μέσα από ένα DBMS. Η MySQL είναι μία διανομή της γλώσσας SQL και οποιοσδήποτε μπορεί να την κατεβάσει, αφού αποτελεί ελεύθερο λογισμικό και μπορεί να την χρησιμοποιήσει είτε ερασιτεχνικά είτε επαγγελματικά. Επίσης αναγνωρίζεται για την ταχύτητα, την αξιοπιστία αλλά και την ευελιξία της, καθώς είναι συμβατή σε περιβάλλοντα Linux, Mac Os και Windows. Αντίστοιχα με την MySQL υπάρχουν κι άλλα συστήματα δωρεάν ή ακόμα και επί πληρωμή. Μερικά από τα πιο γνωστά DBMS είναι το SQL Server της Microsoft, η Oracle SQL Server κ.ά.

Αξίζει να σημειωθεί πως μια βάση δεδομένων είναι μια συλλογή δεδομένων που είναι αποθηκευμένα σε έναν υπολογιστή με σκοπό να μπορούν να επεξεργαστούν με ευκολία από τους διαχειριστές των συστημάτων. Για παράδειγμα, αν



έχουμε δημιουργήσει μια διαδικτυακή εφαρμογή για μαθητές, τότε στη βάση δεδομένων μας μπορούμε να έχουμε έναν πίνακα που να αναπαριστά τους μαθητές και έναν άλλον πίνακα να αναπαριστά τις τάξεις που ανήκουν οι μαθητές. Οι δύο αυτοί πίνακες θα σχετίζονται με σχέση πολλά προς 1 (N:1), διότι ένας μαθητής ανήκει μόνο σε μία τάξη, ενώ μια τάξη μπορεί να έχει πολλούς μαθητές. Με βάση αυτό το παράδειγμα μπορεί κανείς να πάρει μια πρώτη ιδέα με το πώς μπορεί να γίνει η χρήση της MySQL στις διαδικτυακές εφαρμογές. Οι ενέργειες αυτές επάνω στα δεδομένα γίνονται μέσα από τα DBMS, τα οποία με κατάλληλους μηχανισμούς επεξεργάζονται τα δεδομένα χωρίς κανέναν κίνδυνο για αλλοίωση ή απώλεια δεδομένων.

Τέλος, αξίζει να γίνει αναφορά στο εργαλείο λογισμικού phpMyAdmin, το οποίο είναι ένα δωρεάν λογισμικό γραμμένο σε PHP. Με το phpMyAdmin μπορούμε να διαχειριστούμε εύκολα διακομιστές βάσεων δεδομένων MySQL ή MariaDB. Με το phpMyAdmin μπορείς να εκτελέσεις τις περισσότερες εργασίες διαχείρισης όπως είναι

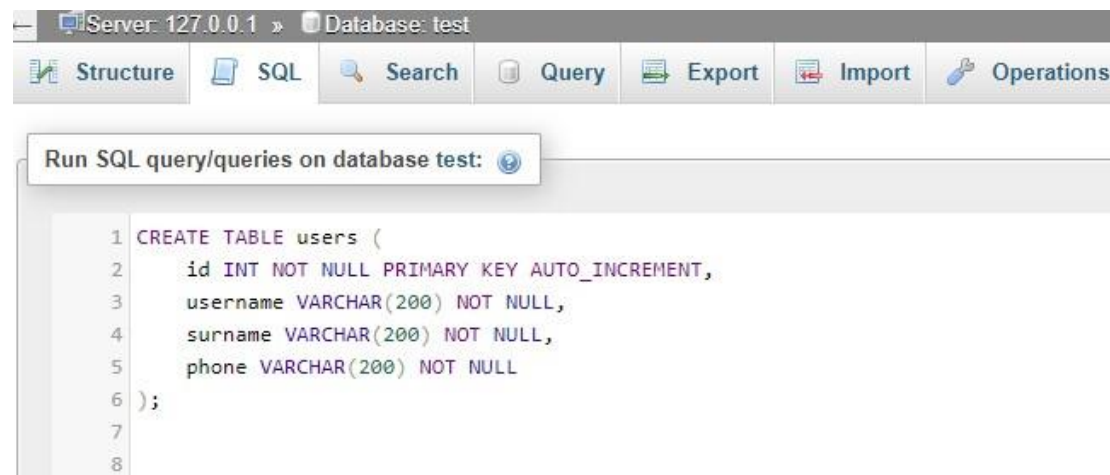
η δημιουργία πινάκων αλλά υποστηρίζει και διαδικασίες προσθήκης λογαριασμών χρηστών. Η εταιρεία Web Hosting που φιλοξενεί την εφαρμογή ή την ιστοσελίδα σου, σου παρέχει τον πίνακα ελέγχου cPanel ή Plesk, ο οποίος πίνακας ελέγχου διαθέτει το λογισμικό phpMyAdmin.

### 1.9.1 Τρόπος λειτουργίας MySQL

Παρακάτω θα παρουσιαστούν μερικά παραδείγματα με τα οποία μπορεί κανείς να δημιουργήσει πίνακα και να εισάγει, να ενημερώσει ή να διαγράψει στοιχεία ενός πίνακα.

#### 1. Δημιουργία πίνακα user με τη χρήση της εντολής CREATE TABLE

Η εντολή δημιουργίας πινάκων απαιτεί το όνομα του πίνακα, τα ονόματα των τομέων και τους ορισμούς για κάθε τομέα.



```
Server: 127.0.0.1 » Database: test
Structure SQL Search Query Export Import Operations
Run SQL query/queries on database test:
1 CREATE TABLE users (
2   id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
3   username VARCHAR(200) NOT NULL,
4   surname VARCHAR(200) NOT NULL,
5   phone VARCHAR(200) NOT NULL
6 );
7
8
```

Ο τομέας ID - ταυτότητας ορίζεται ως πρωτεύον κλειδί. Με τη χρησιμοποίηση του auto\_increment ως ιδιότητα του τομέα, λέμε στην MySQL να προχωρήσει και να προσθέσει τον επόμενο διαθέσιμο αριθμό στον τομέα ταυτότητας για μας.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	username	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
3	surname	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
4	phone	varchar(200)	latin1_swedish_ci		No	None			Change Drop More

## 2. Εισαγωγή δεδομένων στο πίνακα με τη χρήση εντολής INSERT



The screenshot shows the result of the INSERT query in a table view:

Options	id	username	surname	phone
<input type="checkbox"/> Show all   Number of rows: 25   Filter rows: Search this table	1	nikos	koukis	6972534499

Below the table, there are options for actions:  Check all, With selected: Edit, Copy, Delete, Export.

## 3. Συλλογή δεδομένων από το πίνακα με τη χρήση της εντολής SELECT



Select είναι η εντολή της SQL που χρησιμοποιείται για να ανακτήσει τα αρχεία. Αυτή η σύνταξη εντολής μπορεί να είναι απλή ή πολύ περίπλοκη.

The screenshot shows the result of the SELECT query in a table view:

Options	id	username	surname	phone
<input type="checkbox"/> Show all   Number of rows: 25   Filter rows: Search this table	1	nikos	koukis	6972534499

Below the table, there are options for actions:  Check all, With selected: Edit, Copy, Delete, Export.

#### 4. Τροποποίηση στοιχείων πίνακα με την χρήση της εντολής UPDATE

The screenshot shows a database management interface with the following components:

- Header: Server: 127.0.0.1 » Database: test » Table: users
- Navigation tabs: Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations
- Query editor: Run SQL query/queries on table test.users:

```
1 UPDATE users
2 SET surname = 'Antonopoulos'
3 WHERE id=1;
```
- Table view controls: Show all, Number of rows: 25, Filter rows: Search this table
- Table data:

Options	id	username	surname	phone
<input type="checkbox"/> Edit Copy Delete	1	nikos	Antonopoulos	6972534499
- Row actions: Check all, With selected, Edit, Copy, Delete, Export

## ΚΕΦΑΛΑΙΟ 2

### ΔΗΜΙΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ/WEB APP

#### 2.1 Κατηγορίες χρηστών

Η εφαρμογή, όπως αναφέρθηκε και στον πρόλογο, απευθύνεται σε δύο είδη χρηστών, τον διαχειριστή του συστήματος (Administrator) και τον πελάτη (client). Ο καθένας συνδέεται σε διαφορετικό περιβάλλον χρήστη (με βάση το URL) και έχει πρόσβαση σε διαφορετικές σελίδες και λειτουργίες. Δεν υπάρχουν κοινές λειτουργίες. Προφανώς ένας διαχειριστής μπορεί να φτιάξει λογαριασμό πελάτη και να περιηγηθεί στην εφαρμογή ως πελάτης.

##### 2.1.1 Διαχειριστής (Administrator)

Για να μπει κανείς στο διαχειριστικό περιβάλλον αρκεί να πληκτρολογήσει την διεύθυνση <https://groomedoo.eu/groomer/>. Το πρώτο πράγμα που αντικρίζει ο διαχειριστής είναι η φόρμα εισόδου στη πλατφόρμα. Εκεί καταχωρεί τα στοιχεία σύνδεσής του, δηλαδή το email και τον κωδικό πρόσβασής του. Αν έχει πραγματοποιήσει αρχικά την εγγραφή του και τα στοιχεία είναι αληθή, τότε εισέρχεται στην διαχειριστική πλατφόρμα, αλλιώς του «πετάει» μήνυμα σφάλματος.

Εφόσον η σύνδεση πραγματοποιηθεί με επιτυχία, ο διαχειριστής εισέρχεται στην διαχειριστική πλατφόρμα, όπου μπορεί να δει στατιστικά στοιχεία, να τροποποιήσει τα στοιχεία του προφίλ του, να επεξεργαστεί και να ελέγχει τα ραντεβού του, να πραγματοποιεί χειροκίνητα ένα ραντεβού, καθώς και να επεξεργάζεται το ωράριο λειτουργίας του καταστήματός του.

##### 2.1.2 Πελάτης (Client)

Το περιβάλλον του συγκεκριμένου χρήστη είναι και η βασική διεύθυνση (URL) της εφαρμογής. Για να έχει κάποιος πρόσβαση στην εφαρμογή αρκεί να πληκτρολογήσει την διεύθυνση <https://groomedoo.eu/>. Με αφετηρία την αρχική σελίδα ο χρήστης έχει την δυνατότητα με μια έξυπνη αναζήτηση να περιηγηθεί στην εφαρμογή, να δει τους διαθέσιμους groomer ανά περιοχή και ημερομηνία αλλά και να δει τα διαθέσιμα ραντεβού. Παρ' όλα αυτά για να ολοκληρώσει ένα αίτημα για ραντεβού, ο πελάτης θα πρέπει να δημιουργήσει λογαριασμό και να συνδεθεί με τα στοιχεία του, για να ολοκληρώσει την διαδικασία.

Αφού ο χρήστης πραγματοποιήσει σύνδεση, θα έχει την δυνατότητα να επεξεργαστεί τα στοιχεία του προφίλ του, να πραγματοποιήσει αίτημα για ραντεβού και να δει το ιστορικό των ραντεβού του.

## 2.2 Βάση δεδομένων

Η βάση δεδομένων της εφαρμογής αποτελείται από 8 πίνακες. Ακολουθεί παρουσίαση των πινάκων της εφαρμογής. Αρχικά θα παρουσιάσουμε τους πίνακες του διαχειριστή, έπειτα θα παρουσιάσουμε τους πίνακες του πελάτη και τέλος τους κοινούς πίνακες.

### 2.2.1 Πίνακες διαχειριστή

Ο πρώτος πίνακας διαχειριστή ονομάζεται groomer και χρησιμοποιείται για την αποθήκευση των δεδομένων κατά την εγγραφή του χρήστη (groomer) στην πλατφόρμα. Επίσης ο πίνακας χρησιμοποιείται και κατά την είσοδό του στην διαχειριστική πλατφόρμα ως επικύρωση δεδομένων, τόσο για το email όσο και για τον κωδικό πρόσβασης. Τα πεδία του πίνακα είναι τα εξής:

- id: αποτελεί το πρωτεύον κλειδί του πίνακα και για κάθε νέα εγγραφή αυξάνεται μοναδιαία.
- email: περιλαμβάνει τα email των διαχειριστών - groomer.
- company: περιλαμβάνει τα ονόματα των επιχειρήσεων των διαχειριστών - groomer.
- city: περιλαμβάνει τις πόλεις που εδρεύουν οι επιχειρήσεις των διαχειριστών.
- address: περιλαμβάνει τις διευθύνσεις που εδρεύουν οι επιχειρήσεις των διαχειριστών.
- postcode: περιλαμβάνει τους ταχυδρομικούς κώδικες που εδρεύουν οι επιχειρήσεις των διαχειριστών.
- phone: περιλαμβάνει τα κινητά τηλέφωνα των διαχειριστών
- afm: περιλαμβάνει τα ΑΦΜ των επιχειρήσεων των groomer.
- password: περιλαμβάνει τους κωδικούς πρόσβασης των groomer. Για λόγους ασφαλείας, οι κωδικοί πρόσβασης αποθηκεύονται σε κατακερματισμένη μορφή. Αυτό προφυλάσσει από την πιθανότητα ότι κάποιος που αποκτά μη εξουσιοδοτημένη πρόσβαση στη βάση δεδομένων μπορεί να ανακτήσει τους

κωδικούς πρόσβασης κάθε χρήστη στο σύστημα. Ο κατακερματισμός εκτελεί έναν μετασχηματισμό σε έναν κωδικό πρόσβασης, μετατρέποντας τον κωδικό πρόσβασης σε μια άλλη συμβολοσειρά, που ονομάζεται *κατακερματισμένος κωδικός πρόσβασης*.

- **verified**: μπορεί να πάρει είτε την τιμή 0 που σημαίνει ότι ο λογαριασμός δεν έχει επαληθευτεί, είτε την τιμή 1 που σημαίνει ότι ο κωδικός έχει επαληθευτεί με επιτυχία. Σε περίπτωση, που κάποια εγγραφή του πίνακα έχει την τιμή 0 στο συγκεκριμένο πεδίο, τότε το αντίστοιχο email δε θα έχει την δυνατότητα σύνδεσης στην πλατφόρμα.
- **date-registered**: περιλαμβάνει τις ημερομηνίες και ώρες κατά τις οποίες πραγματοποιήσαν εγγραφή οι διαχειριστές – groomer.

	id	email	company	city	address	postcode	phone	afm	password	verified	date_registered
<input type="checkbox"/>	1	groomer1@gmail.com	groomer1	Πάτρα	Patras	26500	6972534499	12345678910	\$2y\$10\$QZP96h3BAIMY6Ywvc52uSH3pFfXSmBmDA0DAUd8y...	1	2021-08-11 17:59:59
<input type="checkbox"/>	2	groomer2@gmail.com	groomer2	Αθήνα	Athens	26500	6972534499	12345678911	\$2y\$10\$NzphB7dByRCfHkMx8XqebN851aUOPhPr5kT2h4...	1	2021-08-11 17:59:59
<input type="checkbox"/>	3	groomer3@gmail.com	groomer3	Θεσσαλονίκη	Thessaloniki	26500	6972534499	12345678912	\$2y\$10\$QZP96h3BAIMY6Ywvc52uSH3pFfXSmBmDA0DAUd8y...	1	2021-08-11 17:59:59
<input type="checkbox"/>	4	groomer4@gmail.com	groomer4	Πάτρα	Patras	26500	6972534499	12345678913	\$2y\$10\$QZP96h3BAIMY6Ywvc52uSH3pFfXSmBmDA0DAUd8y...	1	2021-08-11 14:59:59
<input type="checkbox"/>	5	groomer5@gmail.com	groomer5	Αθήνα	Athens	26500	6972534499	12345678914	\$2y\$10\$QZP96h3BAIMY6Ywvc52uSH3pFfXSmBmDA0DAUd8y...	1	2021-08-11 14:59:59
<input type="checkbox"/>	6	groomer6@gmail.com	groomer6	Θεσσαλονίκη	Thessaloniki	26500	6972534499	12345678915	\$2y\$10\$QZP96h3BAIMY6Ywvc52uSH3pFfXSmBmDA0DAUd8y...	1	2021-08-11 14:59:59
<input type="checkbox"/>	7	groomer7@gmail.com	groomer7	Πάτρα	Patras	26500	6972534499	12345678916	\$2y\$10\$QZP96h3BAIMY6Ywvc52uSH3pFfXSmBmDA0DAUd8y...	1	2021-08-11 14:59:59
<input type="checkbox"/>	8	groomer8@gmail.com	groomer8	Αθήνα	Athens	26500	6972534499	12345678917	\$2y\$10\$QZP96h3BAIMY6Ywvc52uSH3pFfXSmBmDA0DAUd8y...	1	2021-08-11 14:59:59
<input type="checkbox"/>	9	groomer9@gmail.com	groomer9	Θεσσαλονίκη	Thessaloniki	26500	6972534499	12345678918	\$2y\$10\$QZP96h3BAIMY6Ywvc52uSH3pFfXSmBmDA0DAUd8y...	1	2021-08-11 14:59:59
<input type="checkbox"/>	10	groomer10@gmail.com	10company	Πάτρα	Patras	26500	6972534499	12345678919	\$2y\$10\$QZP96h3BAIMY6Ywvc52uSH3pFfXSmBmDA0DAUd8y...	1	2021-08-11 14:59:59
<input type="checkbox"/>	11	groomer11@gmail.com	groomer11	Αθήνα	Athens	26500	6972534499	12345678920	\$2y\$10\$QZP96h3BAIMY6Ywvc52uSH3pFfXSmBmDA0DAUd8y...	1	2021-08-11 14:59:59
<input type="checkbox"/>	12	groomer12@gmail.com	groomer12	Θεσσαλονίκη	Thessaloniki	26500	6972534499	12345678921	\$2y\$10\$QZP96h3BAIMY6Ywvc52uSH3pFfXSmBmDA0DAUd8y...	1	2021-08-11 14:59:59
<input type="checkbox"/>	13	groomer13@gmail.com	groomer13	Πάτρα	Patras	26500	6972534499	12345678922	\$2y\$10\$QZP96h3BAIMY6Ywvc52uSH3pFfXSmBmDA0DAUd8y...	1	2021-08-11 14:59:59
<input type="checkbox"/>	14	groomer14@gmail.com	groomer14	Αθήνα	Athens	26500	6972534499	12345678923	\$2y\$10\$QZP96h3BAIMY6Ywvc52uSH3pFfXSmBmDA0DAUd8y...	1	2021-08-11 14:59:59
<input type="checkbox"/>	15	groomer15@gmail.com	groomer15	Θεσσαλονίκη	Thessaloniki	26500	6972534499	12345678924	\$2y\$10\$QZP96h3BAIMY6Ywvc52uSH3pFfXSmBmDA0DAUd8y...	1	2021-08-11 14:59:59
<input type="checkbox"/>	16	nikoskouk13@hotmail.com	Groomer	Πάτρα	Patras	26500	6972534499	12345678925	\$2y\$10\$P11cFw1WZqT3P88uUwubcchvRjB.1A8QRP6zh...	1	2021-09-13 11:50:11

Στον δεύτερο πίνακα, με τίτλο `groomer_working_days`, αποθηκεύονται οι ημέρες της εβδομάδας, ενώ λειτουργεί ως βοηθητικός πίνακας για τους επόμενους πίνακες της εφαρμογής. Τα πεδία του πίνακα είναι τα εξής:

- **id**: αποτελεί το πρωτεύον κλειδί του πίνακα. Στην συγκεκριμένη περίπτωση λειτουργεί και ως δείκτης για τις ημέρες. Το 0 αναπαριστά την Κυριακή, το 1 την Δευτέρα, το 2 την Τρίτη, το 3 την Τετάρτη, το 4 την Πέμπτη, το 5 την Παρασκευή και το 6 το Σάββατο.
- **name**: αποθηκεύει τις ημέρες της εβδομάδας στα Ελληνικά.
- **name\_engl**: αποθηκεύει τις ημέρες της εβδομάδας στα Αγγλικά αλλά σε σύντομη μορφή (τα 3 αρχικά γράμματα της κάθε ημέρας).

- `name_engl_complete`: αποθηκεύει τις ημέρες της εβδομάδας στα Αγγλικά.

Showing rows 0 - 6 (7 total, Query took 0.0003 seconds.)

```
SELECT * FROM `groomer_working_days`
```

Number of rows: 25 Filter rows: Search this table Sort by key: None

	id	name	name_engl	name_engl_complete
<input type="checkbox"/>	0	Κυριακή	Sun	Sunday
<input type="checkbox"/>	1	Δευτέρα	Mon	Monday
<input type="checkbox"/>	2	Τρίτη	Tue	Tuesday
<input type="checkbox"/>	3	Τετάρτη	Wed	Wednesday
<input type="checkbox"/>	4	Πέμπτη	Thu	Thursday
<input type="checkbox"/>	5	Παρασκευή	Fri	Friday
<input type="checkbox"/>	6	Σάββατο	Sat	Saturday

Query results operations: Print Copy to clipboard Export Display chart Create view

Στον τρίτο πίνακα, `groomer_working_hours`, αποθηκεύονται οι ώρες λειτουργίας των καταστημάτων για κάθε groomer. Τα πεδία του πίνακα είναι τα εξής:

- `id`: αποτελεί το πρωτεύον κλειδί του πίνακα και για κάθε νέα εγγραφή αυξάνεται μοναδιαία.
- `groomer_id`: αποτελεί ξένο κλειδί, το οποίο κάνει αναφορά στο πίνακα `groomer` και δηλώνει το `id` του κάθε groomer.
- `day_id`: αποτελεί ξένο κλειδί, το οποίο κάνει αναφορά στο πίνακα `groomer_working_days` και αναπαριστά το `id` της κάθε ημέρας.
- `from_hour`: αποθηκεύει την ώρα έναρξης λειτουργίας του καταστήματος.
- `until_hour`: αποθηκεύει την ώρα κλεισίματος του καταστήματος.
- `from_hour_extra`: αποθηκεύει την ώρα έναρξης λειτουργίας του καταστήματος σε περίπτωση σπαστού ωραρίου.
- `until_hour_extra`: αποθηκεύει την ώρα κλεισίματος του καταστήματος σε περίπτωση σπαστού ωραρίου.
- `is_closed`: αναπαριστά αν το κατάστημα παραμένει κλειστό την συγκεκριμένη ημέρα και όχι αν το κατάστημα την τωρινή χρονική στιγμή είναι ανοιχτό ή κλειστό. Μπορεί να πάρει 2 τιμές: 0,1.



- status: αναπαριστά την κατάσταση λειτουργίας του καταστήματος, δηλαδή αν την τωρινή χρονική στιγμή είναι ανοιχτό ή κλειστό. Συμπληρώνεται με βάση των προηγούμενων πεδίων του πίνακα. Μπορεί να πάρει 2 τιμές: closed,open.

Showing rows 0 - 24 (35 total, Query took 0.0003 seconds.)

```
SELECT * FROM `groomer_working_hours`
```

1 > >> |  Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	id	groomer_id	day_id	from_hour	until_hour	from_hour_extra	until_hour_extra	is_closed	status
<input type="checkbox"/>	25	1	1	09:30	14:00	0	0	0	open
<input type="checkbox"/>	26	1	2	09:30	14:00	17:00	21:00	0	closed
<input type="checkbox"/>	27	1	3	09:30	14:00	0	0	0	closed
<input type="checkbox"/>	28	1	4	09:30	14:00	17:00	21:00	0	closed
<input type="checkbox"/>	29	1	5	09:30	14:00	17:00	23:30	0	closed
<input type="checkbox"/>	30	1	6	09:30	15:00	0	0	0	closed
<input type="checkbox"/>	32	2	1	09:00	14:00	0	0	0	open
<input type="checkbox"/>	33	2	2	09:00	14:00	17:00	21:00	0	closed
<input type="checkbox"/>	34	2	3	09:00	14:00	0	0	0	closed
<input type="checkbox"/>	35	2	4	09:00	14:00	17:00	21:00	0	closed
<input type="checkbox"/>	36	2	5	09:00	14:00	17:00	21:00	0	closed
<input type="checkbox"/>	37	2	6	09:00	15:00	0	0	0	closed
<input type="checkbox"/>	38	3	1	09:00	14:00	0	0	0	open
<input type="checkbox"/>	39	3	2	09:00	14:00	17:00	21:00	0	closed
<input type="checkbox"/>	40	3	3	09:00	14:00	0	0	0	closed
<input type="checkbox"/>	41	3	4	09:00	14:00	17:00	21:00	0	closed

## 2.2.2 Πίνακες πελάτη

Ο πρώτος πίνακας για τον πελάτη, με τίτλο user, χρησιμοποιείται για την αποθήκευση των δεδομένων κατά την εγγραφή του χρήστη (client) στην πλατφόρμα. Επίσης χρησιμοποιείται και κατά την είσοδό του στην εφαρμογή ως επικύρωση δεδομένων τόσο για το email όσο και για τον κωδικό πρόσβασης. Τα πεδία του πίνακα είναι τα εξής:

- id: αποτελεί το πρωτεύον κλειδί του πίνακα και για κάθε νέα εγγραφή αυξάνεται μοναδιαία.
- email: περιλαμβάνει τα email των χρηστών.
- name: περιλαμβάνει τα ονόματα των χρηστών.
- surname: περιλαμβάνει τα επίθετα των χρηστών.
- phone: περιλαμβάνει τα κινητά τηλέφωνα των χρηστών.
- city: περιλαμβάνει τις πόλεις κατοικίας των χρηστών.

- password: περιλαμβάνει τους κωδικούς πρόσβασης των χρηστών. Για λόγους ασφαλείας, οι κωδικοί πρόσβασης αποθηκεύονται σε κατακερματισμένη μορφή. Αυτό προφυλάσσει από την πιθανότητα ότι κάποιος που αποκτά μη εξουσιοδοτημένη πρόσβαση στη βάση δεδομένων μπορεί να ανακτήσει τους κωδικούς πρόσβασης κάθε χρήστη στο σύστημα. Ο κατακερματισμός εκτελεί έναν μετασχηματισμό σε έναν κωδικό πρόσβασης, μετατρέποντας τον κωδικό πρόσβασης σε μια άλλη συμβολοσειρά, που ονομάζεται *κατακερματισμένος κωδικός πρόσβασης*.
- verified: μπορεί να πάρει είτε την τιμή 0 που σημαίνει ότι ο λογαριασμός δεν έχει επαληθευτεί, είτε την τιμή 1 που σημαίνει ότι ο κωδικός έχει επαληθευτεί με επιτυχία. Σε περίπτωση, που κάποια εγγραφή του πίνακα έχει την τιμή 0 στο συγκεκριμένο πεδίο, τότε το αντίστοιχο email δε θα έχει την δυνατότητα σύνδεσης στην πλατφόρμα.
- is\_customer: μπορεί να πάρει είτε την τιμή 0 που σημαίνει ότι δεν έχει πραγματοποιήσει ποτέ ραντεβού, δηλαδή δεν υπάρχει σαν εγγραφή στον πίνακα customer, είτε την τιμή 1 που σημαίνει ότι έχει πραγματοποιηθεί ραντεβού και υπάρχει σαν εγγραφή στον πίνακα customer.
- date-registered: περιλαμβάνει τις ημερομηνίες και ώρες που πραγματοποιήσαν εγγραφή οι χρήστες.

id	email	name	surname	phone	city	password	verified	is_customer	date_registered
1	user1@gmail.com	user1name	user1surname	6972534499	Πάτρα	\$2y\$10\$9v9d4oVOA0Jvxb/zN9NEau091eiPGGkKz1y3k1RPXq...	1	1	2021-08-11 17:52:38
2	user2@gmail.com	user2name	user2surname	6972534499	Αθήνα	\$2y\$10\$9v9d4oVOA0Jvxb/zN9NEau091eiPGGkKz1y3k1RPXq...	0	1	2021-08-11 17:52:38
3	user3@gmail.com	user3name	user3surname	6972534499	Οικοσολωνική	\$2y\$10\$9v9d4oVOA0Jvxb/zN9NEau091eiPGGkKz1y3k1RPXq...	1	1	2021-08-11 17:52:38
4	user4@gmail.com	user4name	user4surname	6972534499	Πάτρα	\$2y\$10\$9v9d4oVOA0Jvxb/zN9NEau091eiPGGkKz1y3k1RPXq...	1	1	2021-08-11 17:52:38
5	user5@gmail.com	user5name	user5surname	6972534499	Αθήνα	\$2y\$10\$9v9d4oVOA0Jvxb/zN9NEau091eiPGGkKz1y3k1RPXq...	1	1	2021-08-11 17:52:38
6	user6@gmail.com	user6name	user6surname	6972534499	Οικοσολωνική	\$2y\$10\$9v9d4oVOA0Jvxb/zN9NEau091eiPGGkKz1y3k1RPXq...	1	0	2021-08-11 17:52:38
11	nikoskoukis99@gmail.com	nikos	koukis	6969696969	Πάτρα	\$2y\$10\$m.xJWFH67s9Y0KqH9aEieWNGy.e59Fno14vurZXFd...	1	1	2021-08-17 11:15:49
12	test@gmail.com	test	test	6972534499	Πάτρα	\$2y\$10\$HB3o74miv/IQQ08Q6peCSreNgpIil/J.aCPHsIkzUh8Bb...	1	0	2021-11-08 11:17:37

Στον επόμενο πίνακα, με τίτλο customer, αποθηκεύονται όλοι οι χρήστες οι οποίοι έχουν πραγματοποιήσει έστω μια φορά ένα ραντεβού.

- id: αποτελεί το πρωτεύον κλειδί του πίνακα και για κάθε νέα εγγραφή αυξάνεται μοναδιαία.

- email: περιλαμβάνει τα email των χρηστών-πελατών.
- surname: περιλαμβάνει τα επίθετα των χρηστών- πελατών.
- phone: περιλαμβάνει τα κινητά τηλέφωνα των χρηστών- πελατών.
- appointment\_date: περιλαμβάνει τις ημερομηνίες των ραντεβού.
- appointment\_time: περιλαμβάνει τις ώρες των ραντεβού.
- is\_user: παίρνει την τιμή 1, και δηλώνει ότι η συγκεκριμένη εγγραφή είναι user.
- user\_id: αποτελεί ξένο κλειδί που κάνει αναφορά στο πίνακα user και δηλώνει το id του κάθε user.
- entry\_appointment: δηλώνει τις ώρες και τις ημερομηνίες που έγιναν οι αιτήσεις για ραντεβού.

	id	email	name	surname	phone	appointment_date	appointment_time	is_user	user_id	entry_appointment
☐ Edit Copy Delete	14	user2@gmail.com	user2name	user2surname	6972534499	2021-08-18	09:00	1	3	2021-08-14 20:48:48
☐ Edit Copy Delete	15	user2@gmail.com	user2name	user2surname	6972534499	2021-08-18	09:30	1	2	2021-08-14 20:49:04
☐ Edit Copy Delete	16	user2@gmail.com	user2name	user2surname	6972534499	2021-08-18	13:00	1	2	2021-08-18 12:34:10
☐ Edit Copy Delete	17	nikoskoukis99@gmail.com	apostolis	koukakos	2610640582	2021-08-19	09:00	1	11	2021-08-18 14:34:02
☐ Edit Copy Delete	18	nikoskoukis99@gmail.com	apostolis	koukakos	2610640582	2021-08-19	09:30	1	11	2021-08-18 14:34:07
☐ Edit Copy Delete	19	user2@gmail.com	user2name	user2surname	6972534499	2021-08-20	11:00	1	2	2021-08-20 09:51:04
☐ Edit Copy Delete	20	user2@gmail.com	user2name	user2surname	6972534499	2021-08-20	11:30	1	2	2021-08-20 09:51:26
☐ Edit Copy Delete	21	user2@gmail.com	user2name	user2surname	6972534499	2021-08-20	11:00	1	2	2021-08-20 09:51:47
☐ Edit Copy Delete	22	user2@gmail.com	user2name	user2surname	6972534499	2021-08-20	11:00	1	2	2021-08-20 10:31:17
☐ Edit Copy Delete	23	user2@gmail.com	user2name	user2surname	6972534499	2021-08-20	11:00	1	2	2021-08-20 10:32:16
☐ Edit Copy Delete	24	user2@gmail.com	user2name	user2surname	6972534499	2021-08-20	11:00	1	2	2021-08-20 10:47:00
☐ Edit Copy Delete	25	user2@gmail.com	user2name	user2surname	6972534499	2021-08-20	11:00	1	2	2021-08-20 10:47:17
☐ Edit Copy Delete	26	user2@gmail.com	user2name	user2surname	6972534499	2021-08-20	10:30	1	2	2021-08-20 10:47:40
☐ Edit Copy Delete	27	user2@gmail.com	user2name	user2surname	6972534499	2021-08-20	11:30	1	2	2021-08-20 10:51:00
☐ Edit Copy Delete	28	user2@gmail.com	user2name	user2surname	6972534499	2021-08-20	09:30	1	2	2021-08-20 10:51:58
☐ Edit Copy Delete	29	user2@gmail.com	user2name	user2surname	6972534499	2021-08-20	20:00	1	2	2021-08-20 10:58:36
☐ Edit Copy Delete	30	nikoskoukis99@gmail.com	apostolis	koukakos	2610640582	2021-08-20	20:00	1	11	2021-08-20 11:17:05

### 2.2.3 Κοινοί πίνακες

Ο πρώτος κοινός πίνακας λέγεται email\_verification και, όπως είναι λογικό, λειτουργεί τόσο για τον χρήστη όσο και για τον διαχειριστή, ως πίνακας αποθήκευσης δεδομένων για την επιβεβαίωση δημιουργίας λογαριασμού. Εφόσον πραγματοποιηθεί μια εγγραφή στην πλατφόρμα, για να ενεργοποιηθεί ο λογαριασμός, θα πρέπει να ακολουθήσει κανείς τις οδηγίες που λαμβάνει στο email του, ώστε να γίνει η επαλήθευση του λογαριασμού. Τα πεδία του πίνακα είναι τα εξής:

- id: αποτελεί το πρωτεύον κλειδί του πίνακα και για κάθε νέα εγγραφή αυξάνεται μοναδιαία.
- email: περιλαμβάνει το email εγγραφής του χρήστη ή του διαχειριστή.
- token: δημιουργία τυχαίας συμβολοσειράς. Όσο αυτή η συμβολοσειρά είναι διαθέσιμη, θα ισχύει και ο σύνδεσμος επαλήθευσης του λογαριασμού.
- code: αναπαριστά έναν τυχαίο 6ψήφιο κωδικό επαλήθευσης λογαριασμού.

Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)

```
SELECT * FROM `email_verification`
```

Number of rows: 25 | Filter rows: Search this table

+ Options

	id	email	token	code
1		test@gmail.com	ce0bfb48e7bfd1380139389448c05a486188eb359fa47	187246

Number of rows: 25 | Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

Ο επόμενος κοινός πίνακας, `reset_password`, λειτουργεί τόσο για τον χρήστη όσο και για τον διαχειριστή ως πίνακας αποθήκευσης δεδομένων για επαναφορά κωδικού πρόσβασης. Αν κάποιος ξεχάσει τον κωδικό πρόσβασης μπορεί να τον ανακτήσει. Τα πεδία του πίνακα είναι τα εξής:

- `id`: αποτελεί το πρωτεύον κλειδί του πίνακα και για κάθε νέα εγγραφή αυξάνεται μοναδιαία.
- `email`: περιλαμβάνει το email του χρήστη ή του διαχειριστή.
- `token`: δημιουργία τυχαίας συμβολοσειράς. Όσο αυτή η συμβολοσειρά είναι διαθέσιμη, θα ισχύει και ο σύνδεσμος ανάκτησης κωδικού πρόσβασης.

Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)

```
SELECT * FROM `reset_password`
```

Number of rows: 25 | Filter rows: Search this table

+ Options

	id	email	token
1		nikoskoukis99@gmail.com	d78bb19eadaaead7ee369daae5df2ed36188f0ea47582

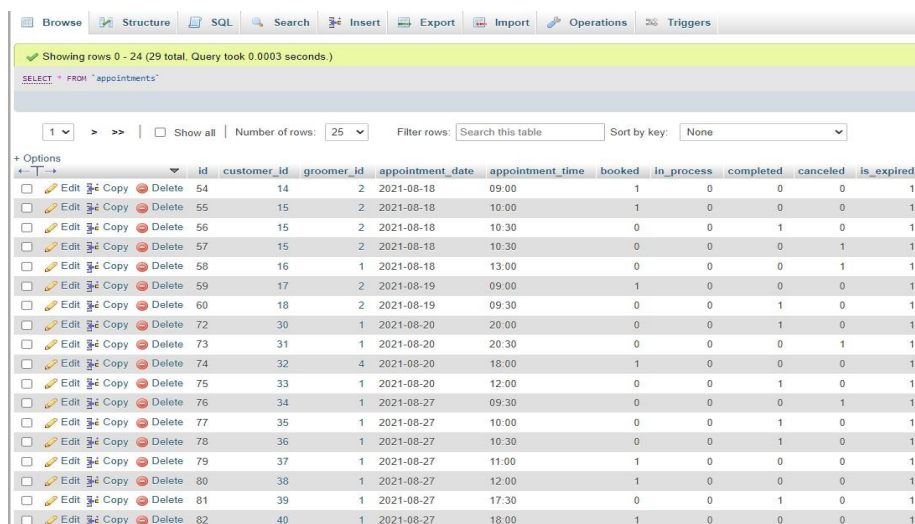
Number of rows: 25 | Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

Ο τελευταίος κοινός πίνακας, με το όνομα appointments, αναπαριστά τα ραντεβού που έχουν πραγματοποιηθεί ή επρόκειτο να πραγματοποιηθούν. Τα πεδία του πίνακα είναι τα εξής:

- id: αποτελεί το πρωτεύον κλειδί του πίνακα και για κάθε νέα εγγραφή αυξάνεται μοναδιαία.
- customer\_id: αποτελεί ξένο κλειδί που κάνει αναφορά στο πίνακα customer και δηλώνει το id του κάθε customer
- groomer\_id: αποτελεί ξένο κλειδί που κάνει αναφορά στο πίνακα groomer και δηλώνει το id του κάθε groomer
- appointment\_date: περιλαμβάνει τις ημερομηνίες των ραντεβού.
- appointment\_time: περιλαμβάνει τις ώρες των ραντεβού.
- booked: μπορεί να πάρει είτε την τιμή 1 που σημαίνει ότι έχει πραγματοποιηθεί με επιτυχία η αίτηση για ραντεβού, είτε την τιμή 0 που σημαίνει ότι η αίτηση ραντεβού δεν έχει πραγματοποιηθεί με επιτυχία.
- in\_process: μπορεί να πάρει είτε την τιμή 1 που σημαίνει ότι η αίτηση για ραντεβού έχει γίνει αποδεκτή από τον groomer και το ραντεβού θα πραγματοποιηθεί σύντομα, είτε την τιμή 0 που σημαίνει ότι δεν έχει γίνει αποδεκτή η αίτηση του ραντεβού από τον groomer.
- completed: μπορεί να πάρει είτε την τιμή 1 που σημαίνει ότι το ραντεβού έχει ολοκληρωθεί, είτε την τιμή 0 που σημαίνει ότι το ραντεβού δεν έχει ολοκληρωθεί.
- is\_expired: μπορεί να πάρει είτε την τιμή 1 που σημαίνει ότι έχει περάσει η ημερομηνία και ώρα διεξαγωγής του ραντεβού, είτε την τιμή 0 που σημαίνει ότι το ραντεβού είναι μέσα στα όρια ημερομηνιών και ώρας.



The screenshot shows a database management interface with a table named 'appointments'. The table has 12 columns: id, customer\_id, groomer\_id, appointment\_date, appointment\_time, booked, in\_process, completed, canceled, and is\_expired. The table contains 25 rows of data, each with a set of action icons (Edit, Copy, Delete) to its left. The data includes appointment dates from 2021-08-18 to 2021-08-27, various appointment times, and status values for booked, in\_process, completed, canceled, and is\_expired.

	id	customer_id	groomer_id	appointment_date	appointment_time	booked	in_process	completed	canceled	is_expired
<input type="checkbox"/> Edit Copy Delete	54	14	2	2021-08-18	09:00	1	0	0	0	1
<input type="checkbox"/> Edit Copy Delete	55	15	2	2021-08-18	10:00	1	0	0	0	1
<input type="checkbox"/> Edit Copy Delete	56	15	2	2021-08-18	10:30	0	0	1	0	1
<input type="checkbox"/> Edit Copy Delete	57	15	2	2021-08-18	10:30	0	0	0	1	1
<input type="checkbox"/> Edit Copy Delete	58	16	1	2021-08-18	13:00	0	0	0	1	1
<input type="checkbox"/> Edit Copy Delete	59	17	2	2021-08-19	09:00	1	0	0	0	1
<input type="checkbox"/> Edit Copy Delete	60	18	2	2021-08-19	09:30	0	0	1	0	1
<input type="checkbox"/> Edit Copy Delete	72	30	1	2021-08-20	20:00	0	0	1	0	1
<input type="checkbox"/> Edit Copy Delete	73	31	1	2021-08-20	20:30	0	0	0	1	1
<input type="checkbox"/> Edit Copy Delete	74	32	4	2021-08-20	18:00	1	0	0	0	1
<input type="checkbox"/> Edit Copy Delete	75	33	1	2021-08-20	12:00	0	0	1	0	1
<input type="checkbox"/> Edit Copy Delete	76	34	1	2021-08-27	09:30	0	0	0	1	1
<input type="checkbox"/> Edit Copy Delete	77	35	1	2021-08-27	10:00	0	0	1	0	1
<input type="checkbox"/> Edit Copy Delete	78	36	1	2021-08-27	10:30	0	0	1	0	1
<input type="checkbox"/> Edit Copy Delete	79	37	1	2021-08-27	11:00	1	0	0	0	1
<input type="checkbox"/> Edit Copy Delete	80	38	1	2021-08-27	12:00	1	0	0	0	1
<input type="checkbox"/> Edit Copy Delete	81	39	1	2021-08-27	17:30	0	0	1	0	1
<input type="checkbox"/> Edit Copy Delete	82	40	1	2021-08-27	18:00	1	0	0	0	1

### 2.3 Εφαρμογή τεχνολογιών

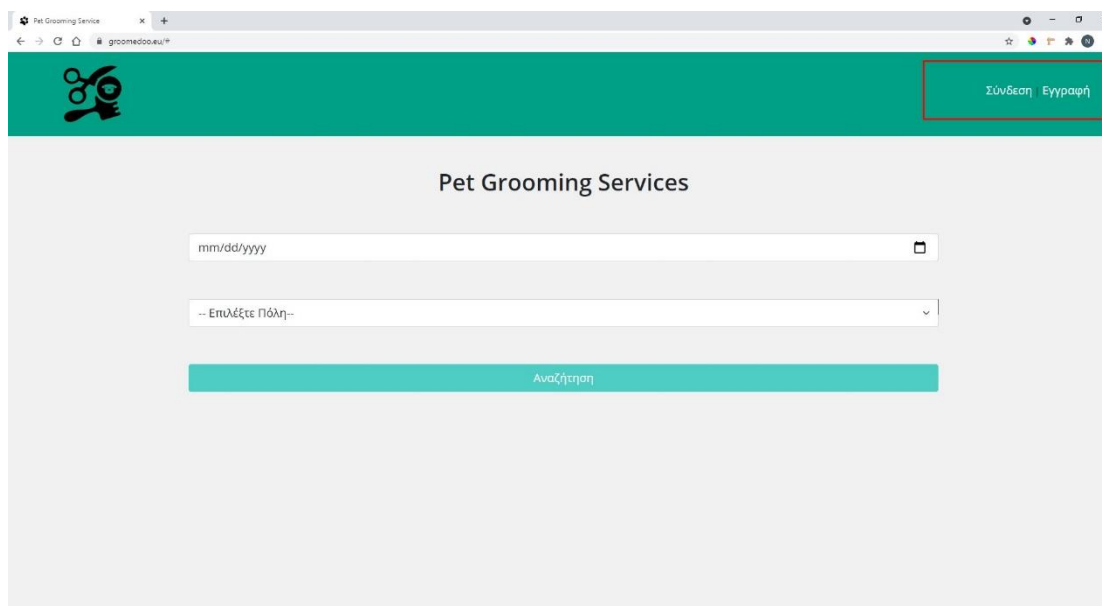
Η σχεδίαση της εφαρμογής έγινε με βάση των τελευταίων web τεχνολογιών και των βάσεων δεδομένων, οι οποίες χρησιμοποιούνται για τον σχεδιασμό σύγχρονων διαδικτυακών εφαρμογών και υπηρεσιών. Στην παρούσα διπλωματική εργασία χρησιμοποιήθηκαν ονομαστικά η HTML(5), η CSS(3), η Bootstrap, η JavaScript, η jQuery , η PHP και η MySQL. Η εφαρμογή όλων αυτών έγινε μέσω του Web Server Apache, που βρίσκεται εγκατεστημένος στον Web Hosting που φιλοξενεί το domain της εφαρμογής ( <https://groomedoo.eu/> ).

## ΚΕΦΑΛΑΙΟ 3

### ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ

#### 3.1 Εγχειρίδιο χρήσης για τον χρήστη – user

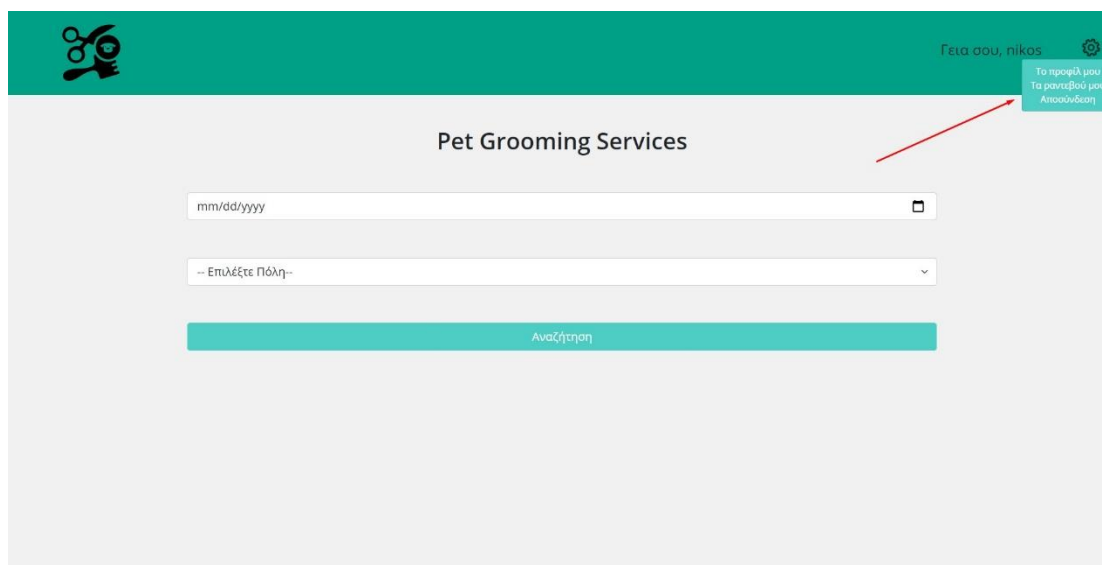
Η εφαρμογή από την πλευρά του χρήστη – πελάτη (user) είναι σχεδιασμένη με τέτοιο τρόπο, ώστε να μπορεί να πλοηγηθεί στην εφαρμογή χωρίς να έχει πραγματοποιήσει κάποια σύνδεση ή εγγραφή. Θα μπορεί να ελέγξει τους διαθέσιμους groomer με βάση τη πόλη του και τις ημερομηνίες που επιθυμεί. Στη συνέχεια ο χρήστης θα μπορεί να δει τα διαθέσιμα ραντεβού που υπάρχουν στον συγκεκριμένο groomer χωρίς να έχει την δυνατότητα να κάνει αίτηση για ραντεβού. Για να πραγματοποιήσει την αίτηση για ραντεβού θα πρέπει να είναι εγγεγραμμένο μέλος στην διαδικτυακή εφαρμογή και να έχει πραγματοποιήσει είσοδο σε αυτήν. Ο χρήστης έχει την δυνατότητα να πραγματοποιήσει εγγραφή ή σύνδεση σε όλες τις σελίδες της διαδικτυακής εφαρμογής, χρησιμοποιώντας τους 2 συνδέσμους που θα βρίσκονται στο επάνω-δεξιά μέρος της εφαρμογής, όπως απεικονίζεται στην παρακάτω φωτογραφία. Αυτοί οι δύο σύνδεσμοι σε οδηγούν αντίστοιχα σε αναδυόμενα παράθυρα είτε για *εγγραφή* (register) είτε για *σύνδεση* (login), όπως θα παρουσιαστούν στη συνέχεια.



Εικόνα 3.1.1: Εγγραφή και σύνδεση χρήστη

Όταν ο χρήστης πραγματοποιήσει είσοδο στην εφαρμογή, θα μπορεί να μεταβεί από οποιαδήποτε σελίδα της εφαρμογής, στο προφίλ του, να δει τα ραντεβού του ή και ακόμα να αποσυνδεθεί χρησιμοποιώντας το μενού που θα του εμφανιστεί πατώντας το

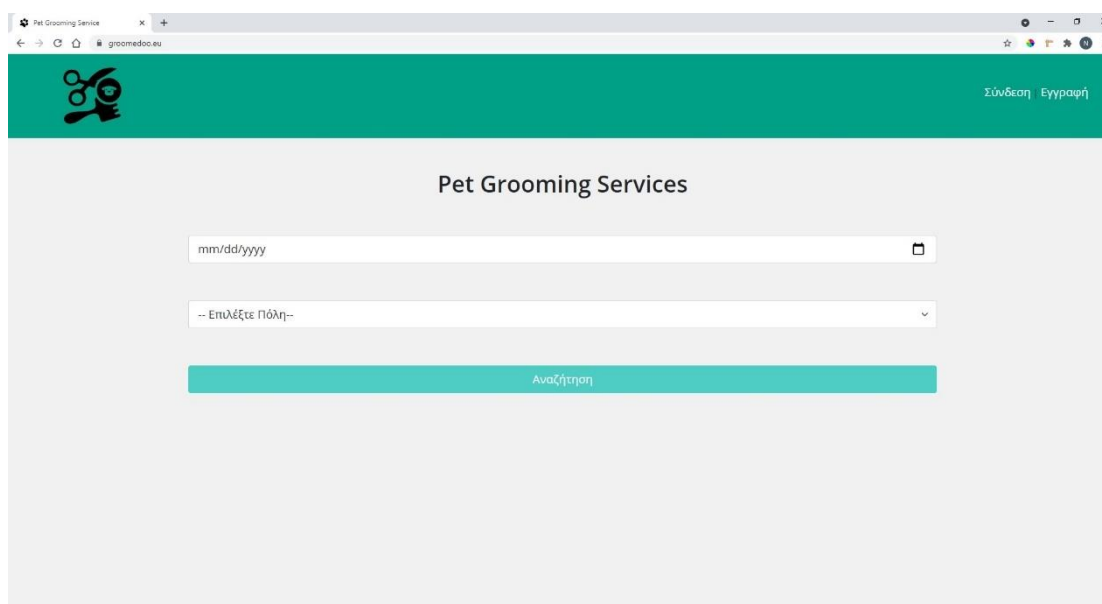
εικονίδιο (γρανάζι) στο επάνω-δεξιά μέρος, όπως παρουσιάζεται στο παρακάτω στιγμιότυπο.



Εικόνα 3.1.2: Ενέργειες χρήστη

### Αρχική Σελίδα Εφαρμογής

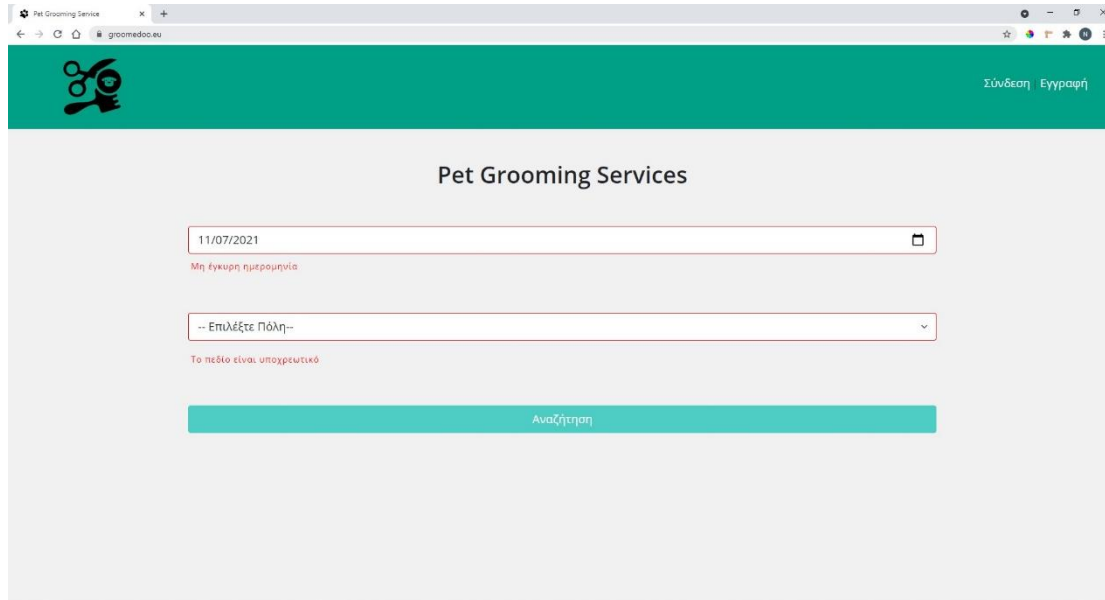
Η αρχική σελίδα της εφαρμογής παρουσιάζεται στην Εικόνα 3.1.3. Για να πλοηγηθεί κάποιος στην εφαρμογή αρκεί να πληκτρολογήσει την διεύθυνση <https://groomedoo.eu/>. Σε αυτήν την σελίδα, ο χρήστης θα μπορεί να πραγματοποιήσει εγγραφή, να κάνει σύνδεση ή να πλοηγηθεί στην εφαρμογή και να αναζητήσει διαθέσιμο groomer με βάση την πόλη και την ημερομηνία.



Εικόνα 3.1.3: Αρχική σελίδα



Στην φόρμα που υπάρχει στην αρχική σελίδα της εφαρμογής, εφαρμόζεται επικύρωση στοιχείων, καθώς ελέγχει αν η φόρμα είναι πλήρως συμπληρωμένη ή κάποιο πεδίο της είναι μη έγκυρο. Η μη επικυρωμένη φόρμα παρουσιάζεται στην Εικόνα 3.1.4.

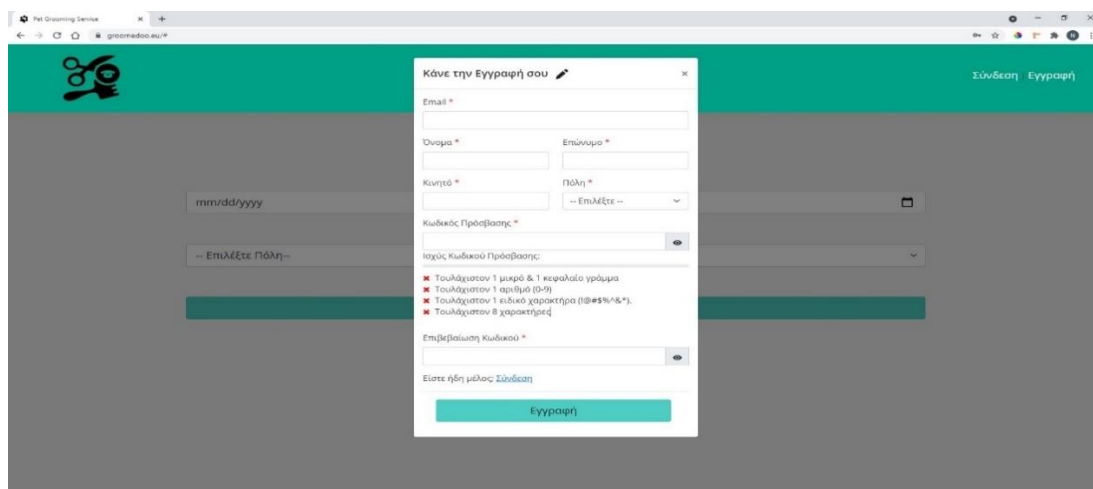


The screenshot shows a web browser window with the URL 'groomedoo.eu'. The page title is 'Pet Grooming Services'. The registration form has two input fields with red error messages below them. The first field contains the date '11/07/2021' and has the error message 'Μη έγκυρη ημερομηνία'. The second field is a dropdown menu with the text '-- Επιλέξτε Πόλη--' and has the error message 'Το πεδίο είναι υποχρεωτικό'. A teal button labeled 'Αναζήτηση' is positioned below the second field.

Εικόνα 3.1.4: Μη επικυρωμένη φόρμα

### Εγγραφή Χρήστη

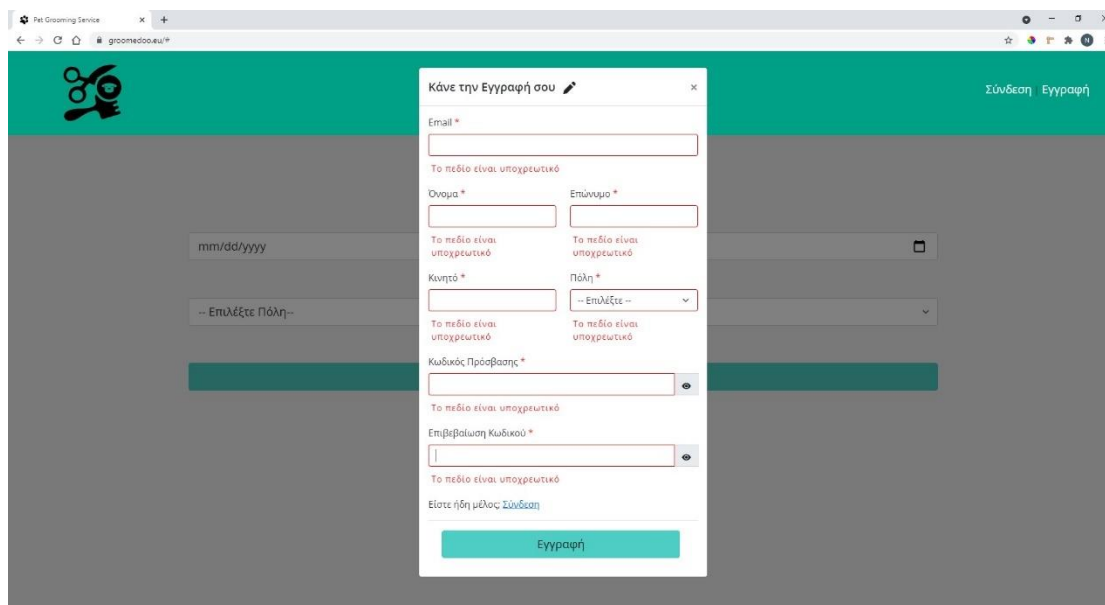
Στην Εικόνα 3.1.5 παρουσιάζεται η φόρμα εγγραφής για τους χρήστες. Εγγραφή μπορεί να πραγματοποιήσει κανείς σε όλες τις σελίδες της διαδικτυακής εφαρμογής. Στην φόρμα εγγραφής έχει σχεδιαστεί ένα real-time password strength checker, το οποίο βοηθάει τον χρήστη να δημιουργήσει έναν ισχυρό κωδικό πρόσβασης.



The screenshot shows the same registration form as in Figure 3.1.4, but with a modal window titled 'Κάνε την Εγγραφή σου' overlaid on top. The modal contains a form for user registration with the following fields: Email \*, Όνομα \*, Επώνυμο \*, Κινητό \*, Πόλη \*, Κωδικός Πρόσβασης \*, and Επικύρωση Κωδικού \*. Below the password field is a 'Ισχύς Κωδικού Πρόσβασης:' section with three red error messages: 'Τουλάχιστον 1 μικρό & 1 κεφαλαίο γράμμα', 'Τουλάχιστον 1 αριθμό (0-9)', and 'Τουλάχιστον 1 ειδικό χαρακτήρα (@#%\$%^&\* )'. Below this is another red error message: 'Τουλάχιστον 8 χαρακτήρες'. At the bottom of the modal is a teal button labeled 'Εγγραφή' and a link 'Είστε ήδη μέλος; Σύνδεση'.

Εικόνα 3.1.5: Φόρμα εγγραφής

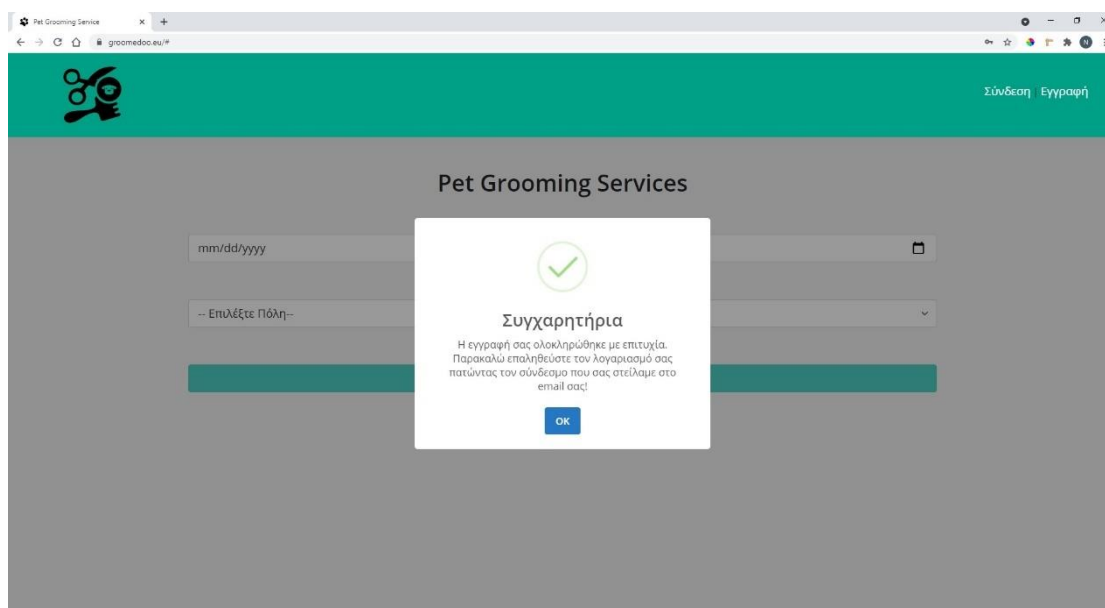
Στην φόρμα εγγραφής έχει εφαρμοστεί επικύρωση στοιχείων, καθώς ελέγχει αν η φόρμα είναι πλήρως συμπληρωμένη ή κάποιο πεδίο της είναι μη έγκυρο. Η μη επικυρωμένη φόρμα εγγραφής παρουσιάζεται στην Εικόνα 3.1.6.



The screenshot shows a web browser window with the URL 'groomedoo.eu'. The page title is 'Pet Grooming Services'. A modal window titled 'Κάνε την Εγγραφή σου' is open, displaying a registration form. The form fields are: Email, Onoma (Name), Epinymon (Surname), Kineto (Phone), Poley (City), Kodikos Prorabashis (Access Code), and Epibebaiosh Kodikou (Access Code Confirmation). Each field has a red border and a message below it stating 'Το πεδίο είναι υποχρεωτικό' (The field is required). The 'Email' field also has a red 'x' icon. The 'Kineto' field has a red 'x' icon. The 'Poley' field is a dropdown menu with the text '-- Επιλέξτε Πόλη --'. The 'Κωδικός Πρόσβασης' field has a red 'x' icon. The 'Επιβεβαίωση Κωδικού' field has a red 'x' icon. At the bottom of the modal, there is a green button labeled 'Εγγραφή'.

Εικόνα 3.1.6: Μη επικυρωμένη φόρμα εγγραφής

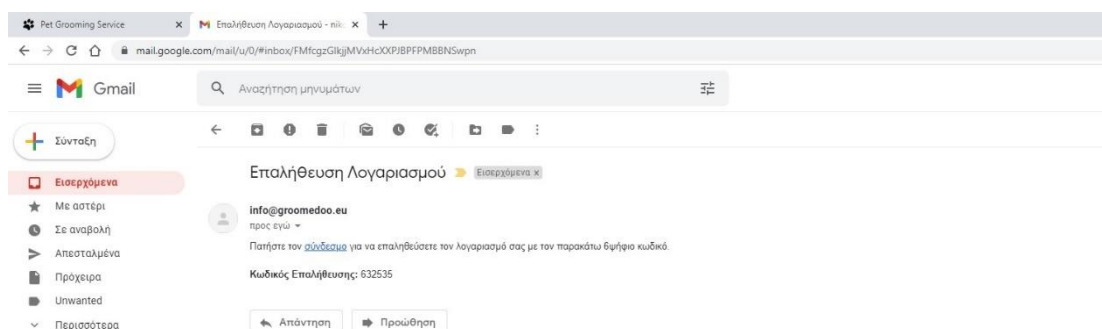
Εφόσον ο χρήστης συμπληρώσει ορθά όλα τα πεδία της φόρμας και πατήσει υποβολή θα του εμφανίσει ένα αναδυόμενο παράθυρο που θα τον ενημερώνει ότι η φόρμα υποβλήθηκε με επιτυχία καθώς θα τον ενημερώνει πως ο λογαριασμός του θα πρέπει να επαληθευτεί, όπως παρουσιάζεται στην Εικόνα 3.1.7.



The screenshot shows the same web browser window as in Figure 3.1.6. The registration form is now displayed in a modal window titled 'Pet Grooming Services'. The form fields are: Email, Onoma (Name), Epinymon (Surname), Kineto (Phone), Poley (City), Kodikos Prorabashis (Access Code), and Epibebaiosh Kodikou (Access Code Confirmation). The form is now filled out. A success message is displayed in a white modal window with a green checkmark icon. The message reads: 'Συγχαρητήρια. Η εγγραφή σας ολοκληρώθηκε με επιτυχία. Παρακαλώ επαληθεύστε τον λογαριασμό σας πατώντας τον σύνδεσμο που σας στείλαμε στο email σας!'. At the bottom of the success message, there is a blue button labeled 'OK'.

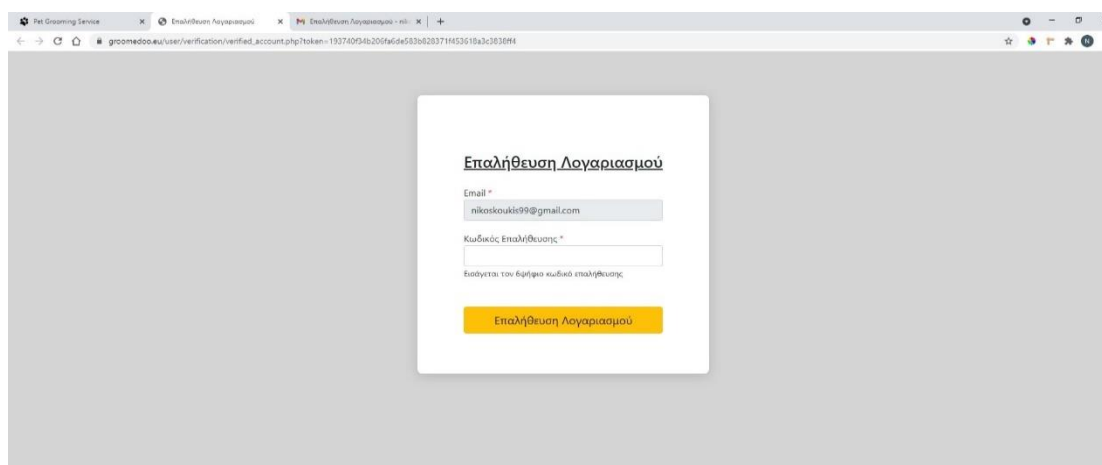
Εικόνα 3.1.7: Αναδυόμενο παράθυρο επιτυχημένης εγγραφής

Στο email που χρησιμοποίησε κατά την εγγραφή του, θα του σταλθεί ένας σύνδεσμος επαλήθευσης λογαριασμού. Θα πρέπει να ακολουθήσει τις οδηγίες του email έτσι ώστε να επαληθεύσει τον λογαριασμό του και να μπορεί να πραγματοποιήσει είσοδο στην εφαρμογή. Εάν ένας λογαριασμός δεν έχει επαληθευτεί, δε θα μπορεί να συνδεθεί στην διαδικτυακή εφαρμογή. Στο email επαλήθευσης υπάρχει ένας σύνδεσμος τον οποίο θα πρέπει να επισκεφτεί ο χρήστης ώστε να καταχωρήσει τον 6ψήφιο κωδικό για να επαληθευτεί ο λογαριασμός. Στην Εικόνα 3.1.8 παρουσιάζεται ένα email επαλήθευσης λογαριασμού.



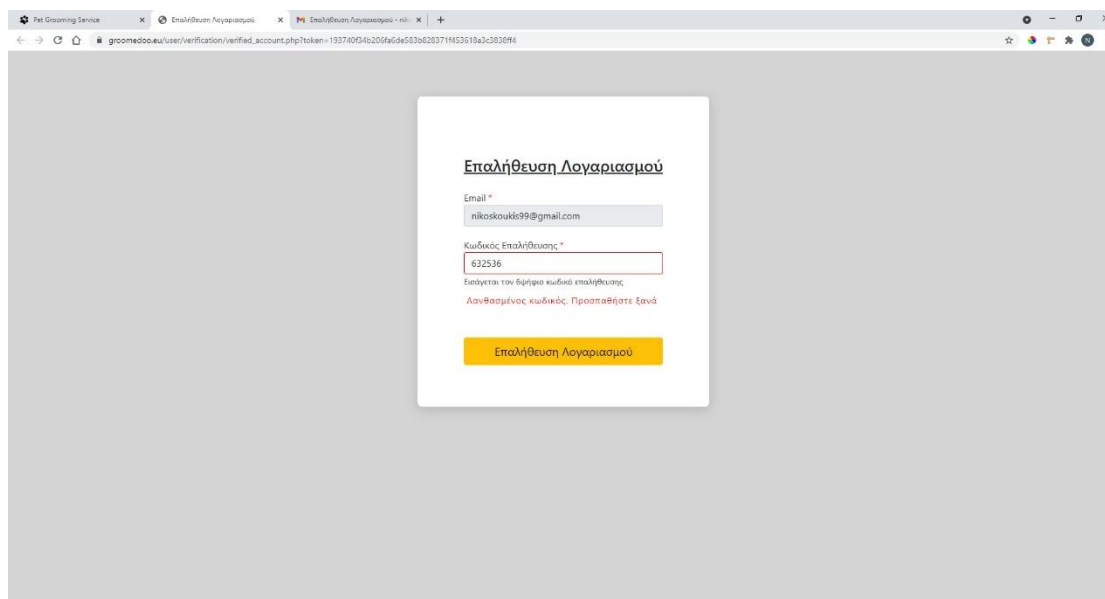
Εικόνα 3.1.8: Email επαλήθευσης λογαριασμού

Στην Εικόνα 3.1.9 παρουσιάζεται η σελίδα που οδηγήθηκε ο χρήστης πατώντας τον σύνδεσμο του email επαλήθευσης. Η σελίδα αυτή περιέχει μια φόρμα με το δηλωθέν email της εγγραφής το οποίο ο χρήστης δε μπορεί να τροποποιήσει και ένα πεδίο ώστε ο χρήστης να τοποθετήσει τον κωδικό επαλήθευσης που του έχει σταλθεί στο email.



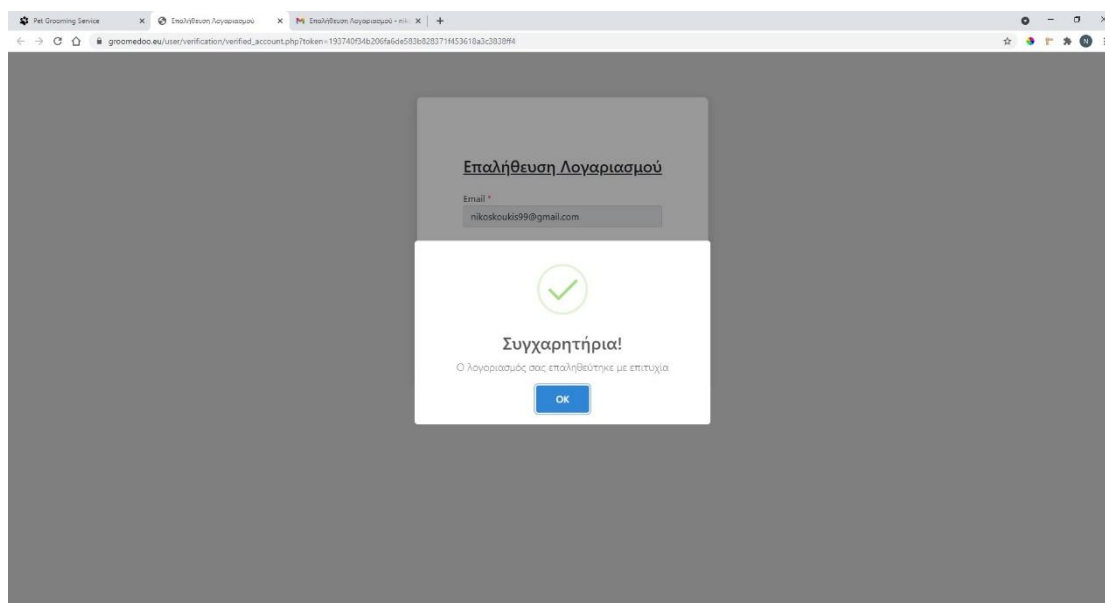
Εικόνα 3.1.9: Επαλήθευση λογαριασμού

Στην παραπάνω φόρμα έχει εφαρμοστεί επικύρωση στοιχείων σε περίπτωση που το πεδίο του κωδικού είναι κενό ή μη έγκυρο. Ένας μη έγκυρος κωδικός επαλήθευσης παρουσιάζεται στο παρακάτω στιγμιότυπο.



Εικόνα 3.1.10: Μη έγκυρος κωδικός επαλήθευσης λογαριασμού

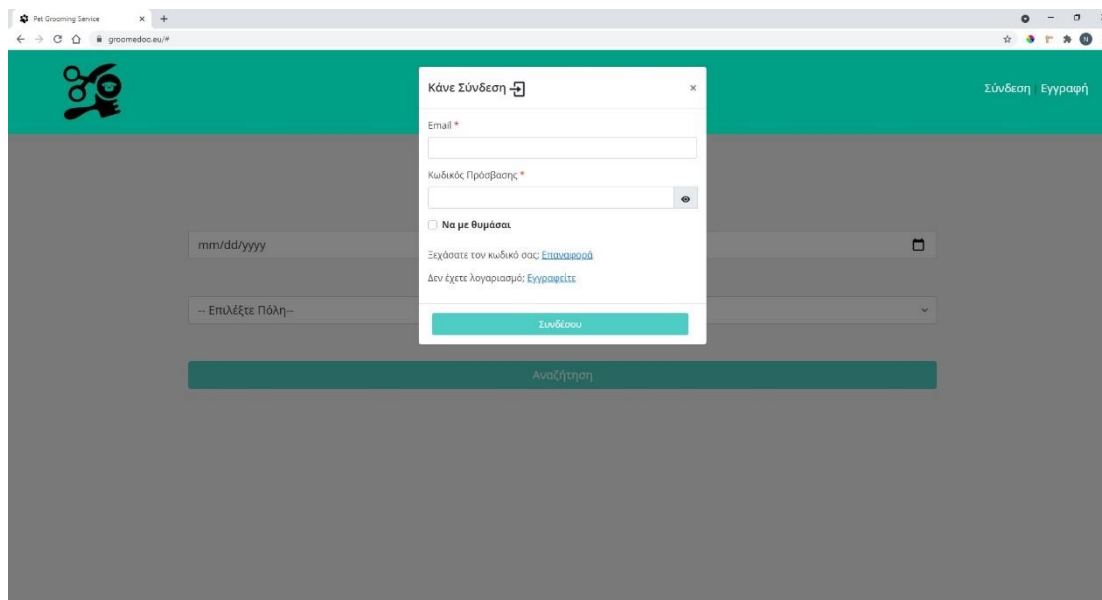
Εφόσον ο χρήστης εισάγει τον σωστό κωδικό επαλήθευσης, θα ενημερωθεί με ένα αναδυόμενο παράθυρο και θα μεταφερθεί στην αρχική σελίδα της εφαρμογής έχοντας τον λογαριασμό του επαληθευμένο, όπως παρουσιάζεται στην Εικόνα 3.1.11.



Εικόνα 3.1.11: Επιτυχημένη επαλήθευση λογαριασμού

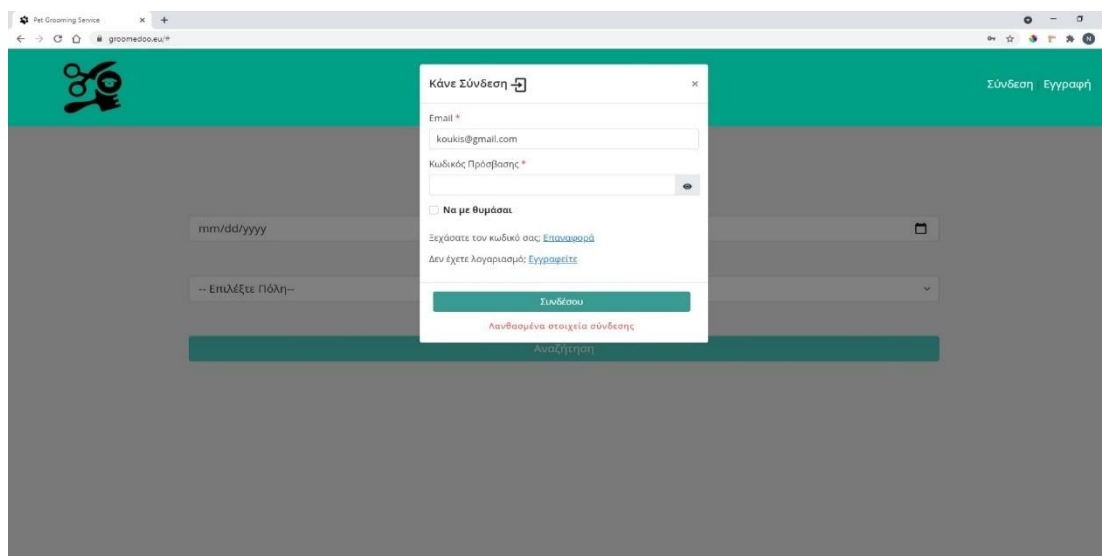
## Σύνδεση Χρήστη

Στην Εικόνα 3.1.12 παρουσιάζεται η φόρμα σύνδεσης για τους χρήστες. Σύνδεση μπορεί να πραγματοποιήσει κανείς σε όλες τις σελίδες της διαδικτυακής εφαρμογής. Μία σύνδεση μπορεί να πραγματοποιηθεί μόνο όταν ο λογαριασμός είναι επαληθευμένος.



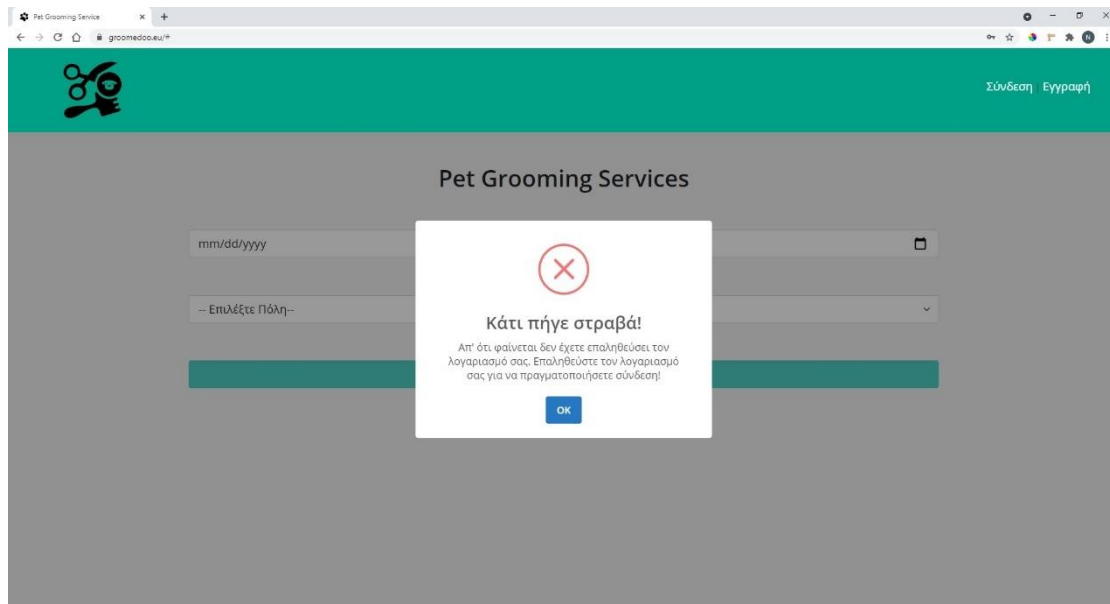
Εικόνα 3.1.12: Φόρμα σύνδεσης χρηστών

Στην φόρμα σύνδεσης έχει εφαρμοστεί επικύρωση στοιχείων, καθώς ελέγχει αν η φόρμα είναι πλήρως συμπληρωμένη, εάν έχει επαληθευτεί ή ακόμα και κάποιο πεδίο της είναι μη έγκυρο. Η μη επικυρωμένη φόρμα σύνδεσης λόγω μη συμπλήρωσης κάποιου πεδίου ή μη έγκυρου πεδίου παρουσιάζεται στην Εικόνα 3.1.13.



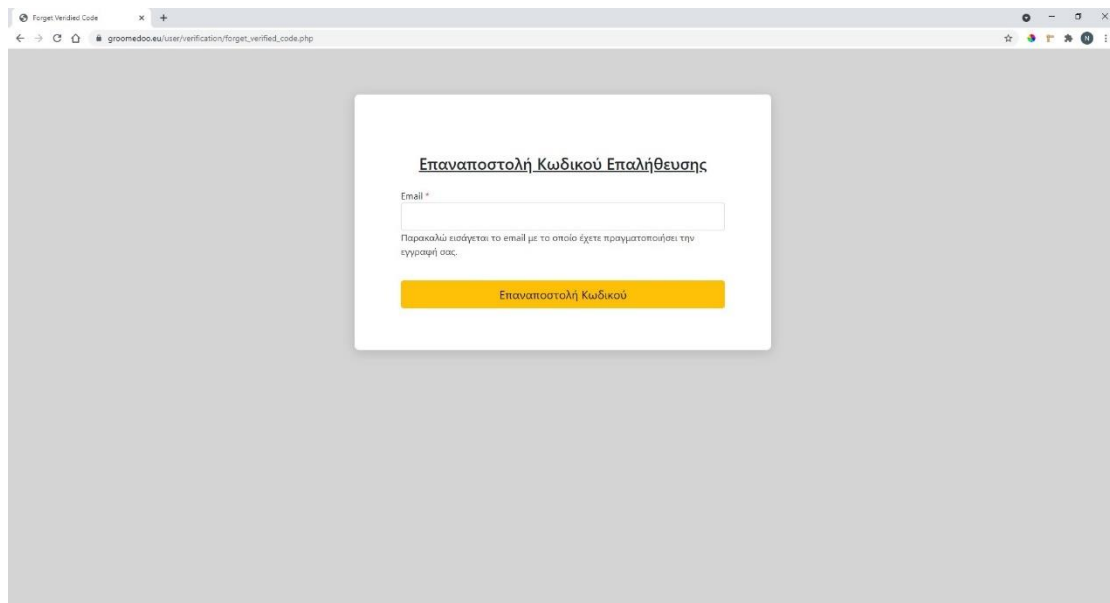
Εικόνα 3.1.13: Μη επικυρωμένη φόρμα σύνδεσης

Όταν μια φόρμα υποβληθεί και ο λογαριασμός δεν έχει επαληθευτεί θα του εμφανίσει ένα αναδυόμενο παράθυρο, όπως παρουσιάζεται παρακάτω.



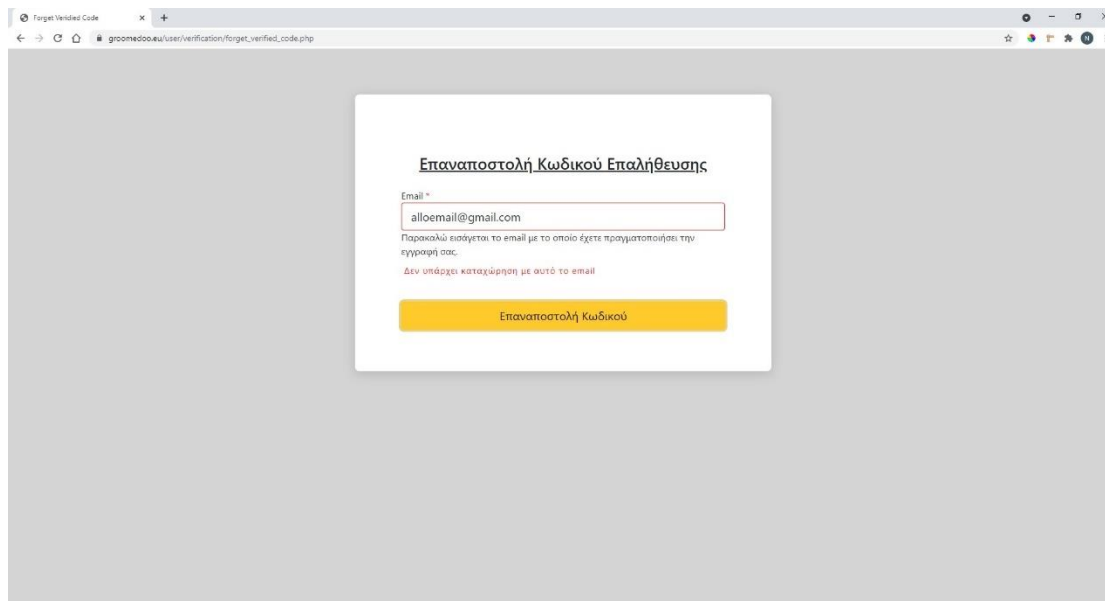
Εικόνα 3.1.14: Σφάλμα υποβληθείσας φόρμας

Όταν ο χρήστης πατήσει «OK» στο αναδυόμενο παράθυρο, θα τον μεταφέρει σε μια καινούργια σελίδα, ώστε να τοποθετήσει το email που έχει κάνει εγγραφή, έτσι ώστε να του σταλθεί στο email του ένας κωδικός επαλήθευσης κάνοντας τα βήματα που έχουμε περιγράψει παραπάνω.



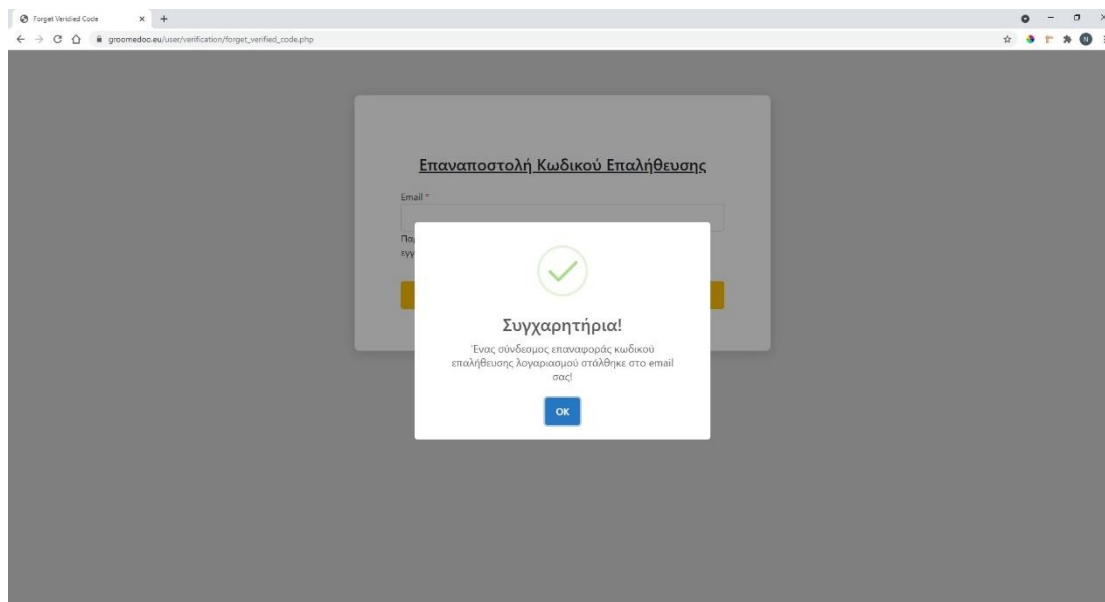
Εικόνα 3.1.15: Επαναποστολή κωδικού επαλήθευσης

Στην παραπάνω φόρμα έχει εφαρμοστεί επικύρωση στοιχείων, ώστε να αποφευχθεί το γεγονός κάποιος χρήστης να τοποθετήσει άλλο email από εκείνο που είχε χρησιμοποιήσει κατά την εγγραφή του.



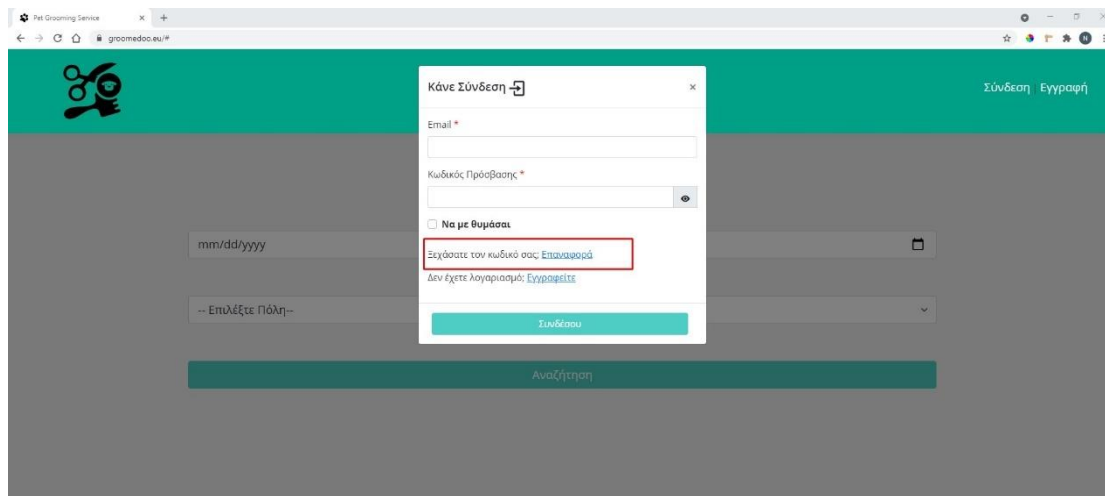
Εικόνα 3.1.16: Επικύρωση στοιχείων

Όταν η φόρμα υποβληθεί με το σωστό email, τότε ένα αναδυόμενο παράθυρο θα δώσει στον χρήστη τις κατάλληλες οδηγίες που περιγράψαμε παραπάνω, ώστε να επαληθεύσει τον λογαριασμό του.



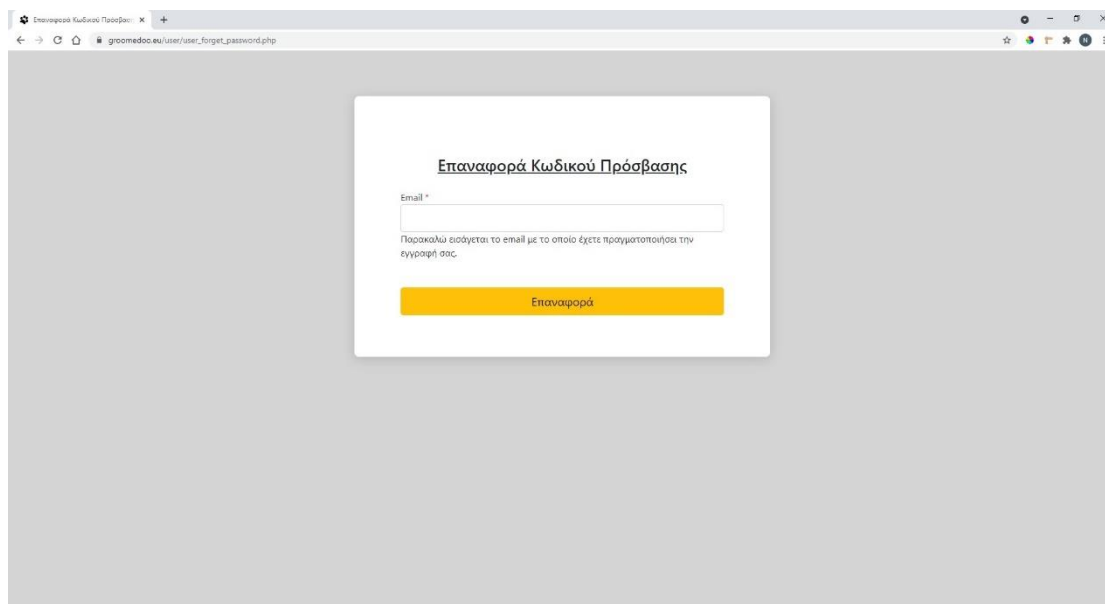
Εικόνα 3.1.17: Οδηγίες επαλήθευσης λογαριασμού

Σε περίπτωση που ο χρήστης ξεχάσει τον κωδικό πρόσβασης μπορεί να κάνει επαναφορά κωδικού πρόσβασης πατώντας τον βοηθητικό σύνδεσμο που υπάρχει στην φόρμα σύνδεσης, όπως παρουσιάζεται στην Εικόνα 3.1.18.



Εικόνα 3.1.18: Βοηθητικός σύνδεσμος για επαναφορά κωδικού πρόσβασης χρήστη

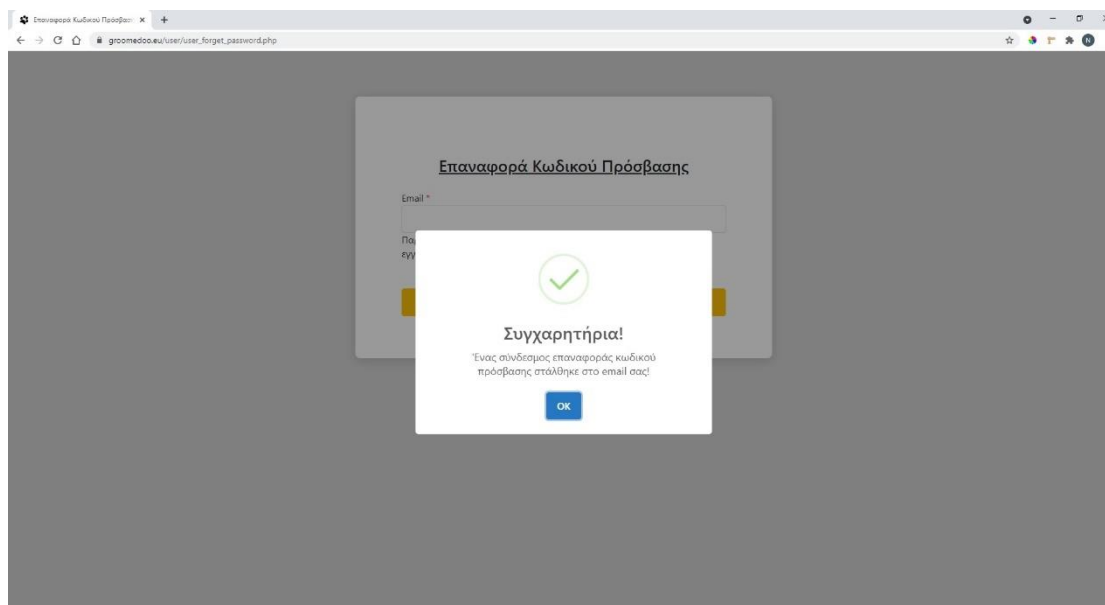
Ο παραπάνω σύνδεσμος, θα τον οδηγήσει σε μια καινούργια σελίδα, όπως παρουσιάζεται παρακάτω, ώστε να τοποθετήσει το email που έχει κάνει εγγραφή έτσι ώστε να του σταλθεί στο email του, οδηγίες ανάκτησης κωδικού πρόσβασης. Στην παρακάτω φόρμα έχει εφαρμοστεί επικύρωση στοιχείων ώστε να αποφευχθεί το γεγονός κάποιος χρήστης να τοποθετήσει άλλο email από εκείνο που είχε χρησιμοποιήσει κατά την εγγραφή του.



Εικόνα 19: Επαναφορά κωδικού πρόσβασης

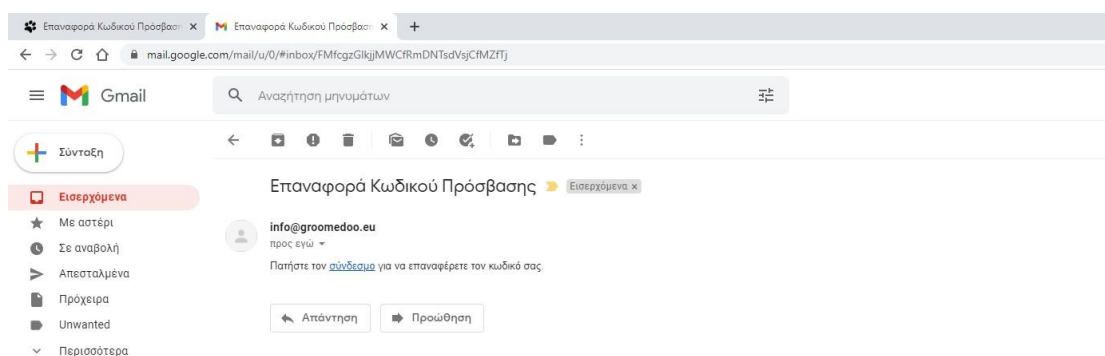


Εφόσον η φόρμα υποβληθεί με επιτυχία ένα αναδυόμενο παράθυρο θα τον ενημερώνει πως ο σύνδεσμος επαναφοράς κωδικού έχει σταλθεί στο email του, όπως παρουσιάζεται παρακάτω.



Εικόνα 3.1.20: Επιτυχημένη επαναφορά κωδικού

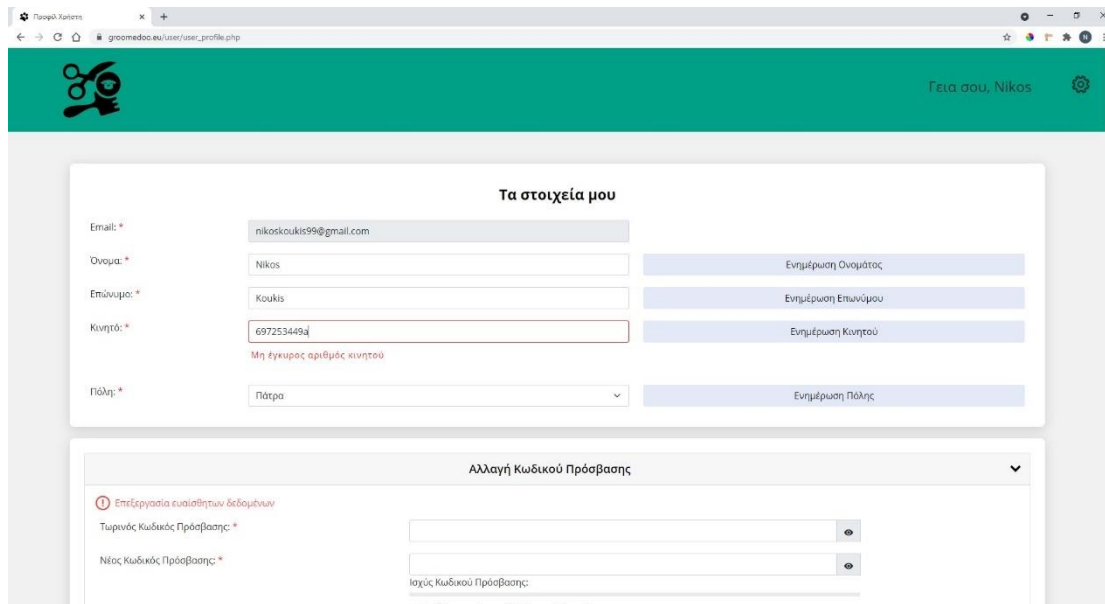
Στην Εικόνα 3.1.21 παρουσιάζεται ένα email επαναφοράς κωδικού πρόσβασης.



Εικόνα 3.1.21: Email επαναφοράς κωδικού πρόσβασης

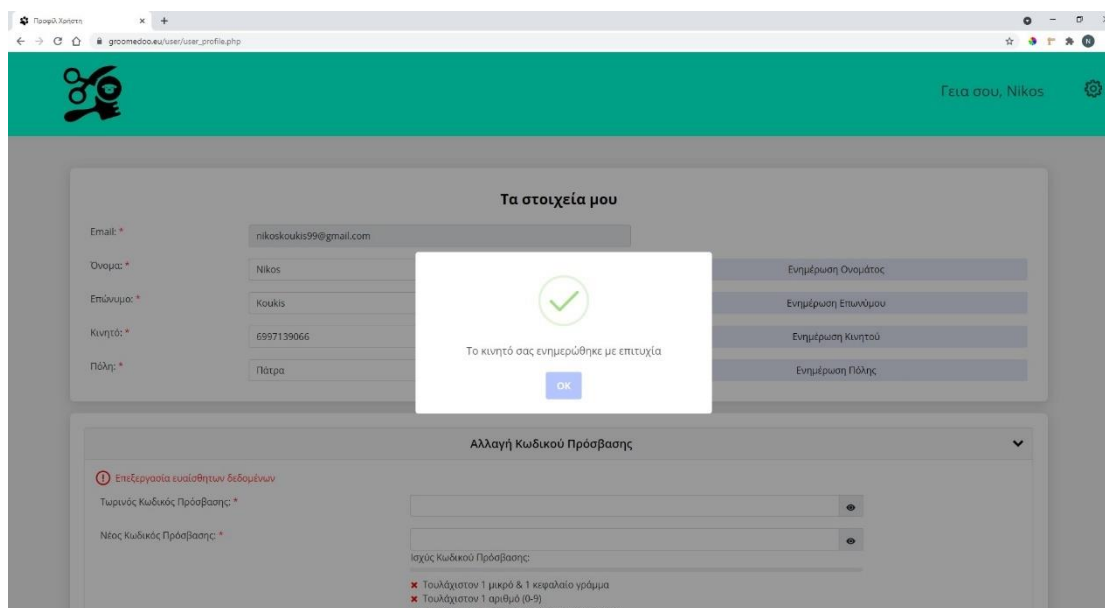
Στην Εικόνα 3.1.22 παρουσιάζεται η σελίδα που οδηγήθηκε ο χρήστης πατώντας τον σύνδεσμο του email για επαναφορά κωδικού πρόσβασης. Η σελίδα αυτή περιέχει μια φόρμα με το δηλωθέν email της εγγραφής το οποίο ο χρήστης δε μπορεί να τροποποιήσει και δύο πεδία για να εισάγει ο χρήστης τον νέο κωδικό πρόσβασης.





Εικόνα 3.1.24: Μη έγκυρες τιμές στοιχείων χρήστη

Όταν ο χρήστης τροποποιήσει κάποιο πεδίο (για παράδειγμα το κινητό) και πατήσει «Ενημέρωση Κινητού» θα του εμφανίσει ένα αναδυόμενο παράθυρο που θα τον ενημερώνει πως το συγκεκριμένο πεδίο ενημερώθηκε με επιτυχία, όπως παρουσιάζεται παρακάτω.



Εικόνα 3.1.25: Επιτυχημένη ενημέρωση στοιχείων χρήστη

Επίσης, ο χρήστης μέσα από την σελίδα του προφίλ μπορεί να αλλάξει τον κωδικό του πρόσβασης. Στην φόρμα αλλαγής κωδικού πρόσβασης έχει σχεδιαστεί ένα real-time password strength checker, το οποίο βοηθάει τον χρήστη να δημιουργήσει έναν ισχυρό κωδικό πρόσβασης.

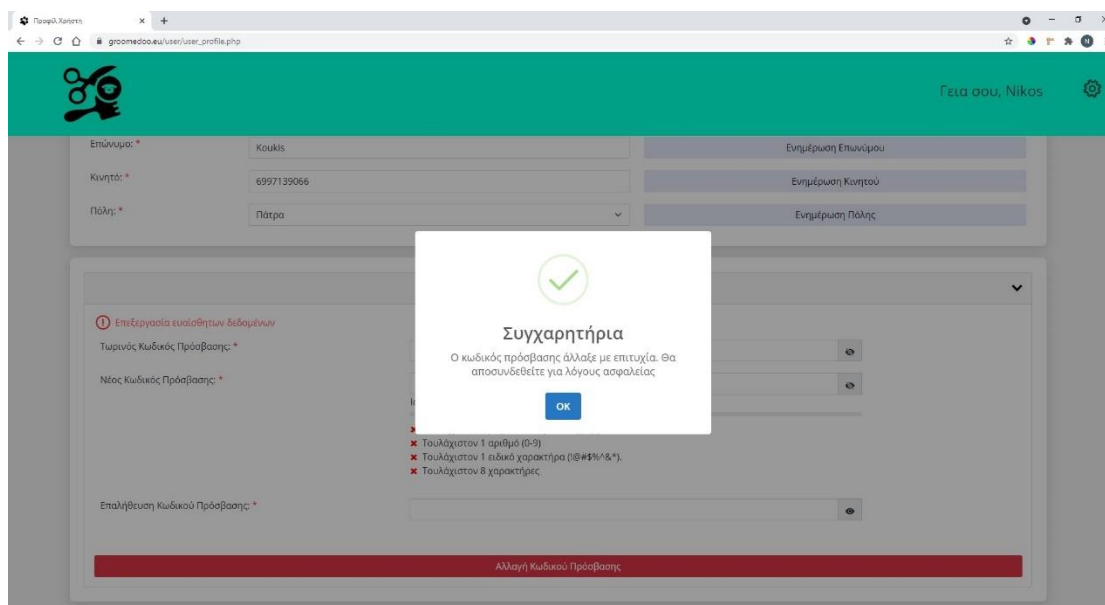
Εικόνα 3.1.26: Αλλαγή κωδικού πρόσβασης

Στην παραπάνω φόρμα, έχει εφαρμοστεί επικύρωση στοιχείων σε περίπτωση που ο χρήστης τοποθετήσει μη έγκυρο τωρινό κωδικό πρόσβασης ή ο νέος κωδικός πρόσβασης είναι αδύναμος ή δεν γίνεται σωστή επαλήθευση κωδικού πρόσβασης. Ο μη έγκυρος τωρινός κωδικός πρόσβασης παρουσιάζεται στην Εικόνα 3.1.27.

Εικόνα 3.1.27: Επικύρωση στοιχείων αλλαγής κωδικού πρόσβασης

Όταν ο χρήστης εισάγει σωστά τα πεδία στη φόρμα και πατήσει «Αλλαγή Κωδικού Πρόσβασης» θα του εμφανίσει ένα αναδυόμενο παράθυρο που θα τον ενημερώνει ότι ο κωδικός άλλαξε με επιτυχία, όπως παρουσιάζεται στην Εικόνα 3.1.28.

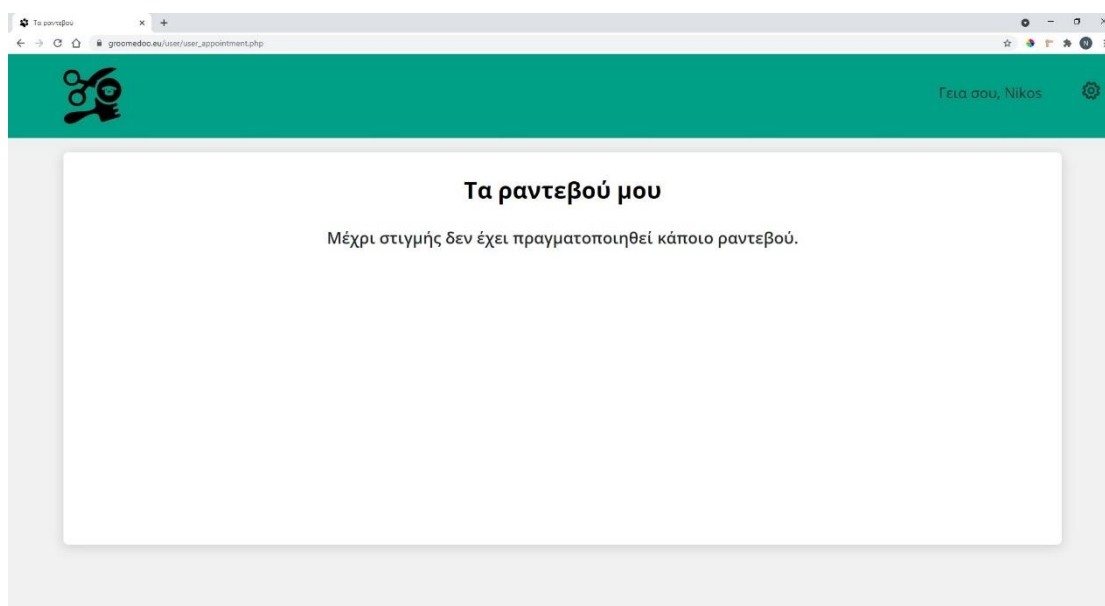
Στη συνέχεια ο χρήστης θα μεταφερθεί στην αρχική σελίδα και θα αποσυνδεθεί από την εφαρμογή ώστε να κάνει ξανά σύνδεση με τον νέο κωδικό πρόσβασης.



Εικόνα 3.1.28: Επιτυχημένη αλλαγή κωδικού πρόσβασης

### Ραντεβού Χρήστη

Ο χρήστης για να δει την σελίδα με τα ραντεβού αρκεί να πατήσει τον σύνδεσμο «Τα ραντεβού μου» από το μενού. Παρακάτω παρουσιάζεται μια σελίδα με κενή τη λίστα των ραντεβού. Αυτό σημαίνει πως ο χρήστης δεν έχει πραγματοποιήσει ακόμα αίτημα για ραντεβού.



Εικόνα 3.1.29: Κενή λίστα ραντεβού

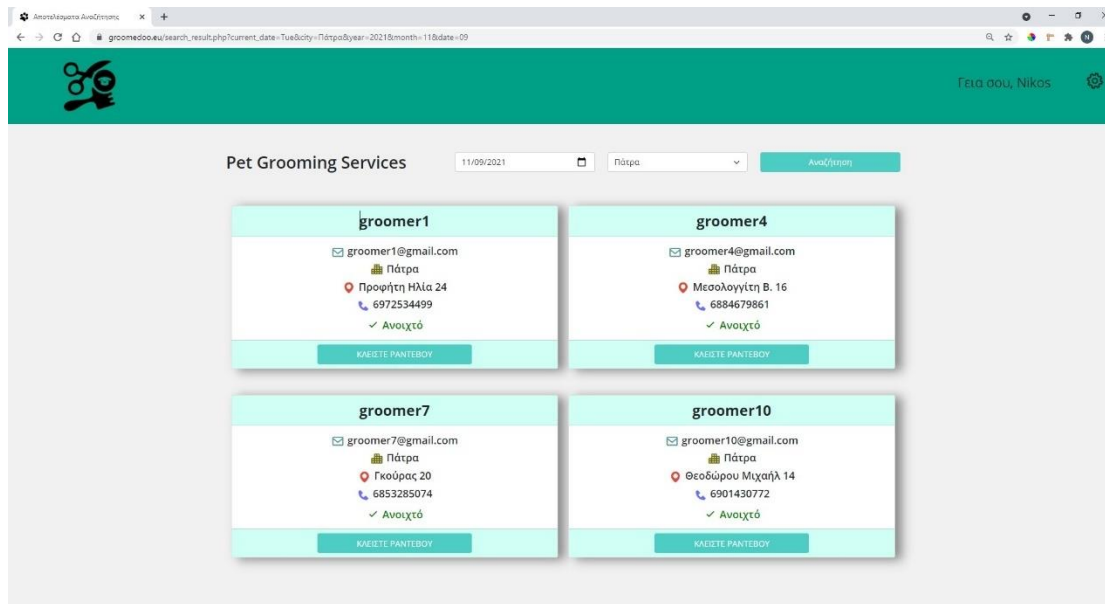
Μια συμπληρωμένη λίστα με τα ραντεβού του χρήστη παρουσιάζεται στην Εικόνα 3.1.30. Η λίστα μας ενημερώνει για την ημερομηνία του κάθε ραντεβού, για την ώρα, για τον groomer και την κατάσταση στην οποία βρίσκονται τα ραντεβού. Όσα ραντεβού έχουν κατάσταση «Εκκρεμεί» σημαίνει ότι το ραντεβού έχει υποβληθεί με επιτυχία και περιμένει έγκριση από τον διαχειριστή – groomer. Όσα έχουν κατάσταση «Σε εξέλιξη» σημαίνει ότι το ραντεβού έχει υποβληθεί με επιτυχία και έχει εγκριθεί και από τον groomer. Όταν η κατάσταση είναι «Ακυρωμένο» σημαίνει ότι το ραντεβού έχει υποβληθεί με επιτυχία αλλά ο διαχειριστής – groomer απέρριψε την αίτηση. Τέλος, όταν η κατάσταση είναι «Ολοκληρωμένο» σημαίνει ότι το ραντεβού έχει πραγματοποιηθεί με επιτυχία.

#	Ημερομηνία	Ώρα	Groomer	Κατάσταση
1	2021-12-16	17:30	groomer10	Ακυρωμένο
2	2021-11-20	19:30	groomer10	Εκκρεμεί
3	2021-11-16	17:30	groomer10	Σε εξέλιξη
4	2021-10-16	17:30	groomer10	Ολοκληρωμένο
5	2021-09-16	17:30	groomer10	Ολοκληρωμένο

Εικόνα 3.1.30: Κατάσταση ραντεβού πελάτη

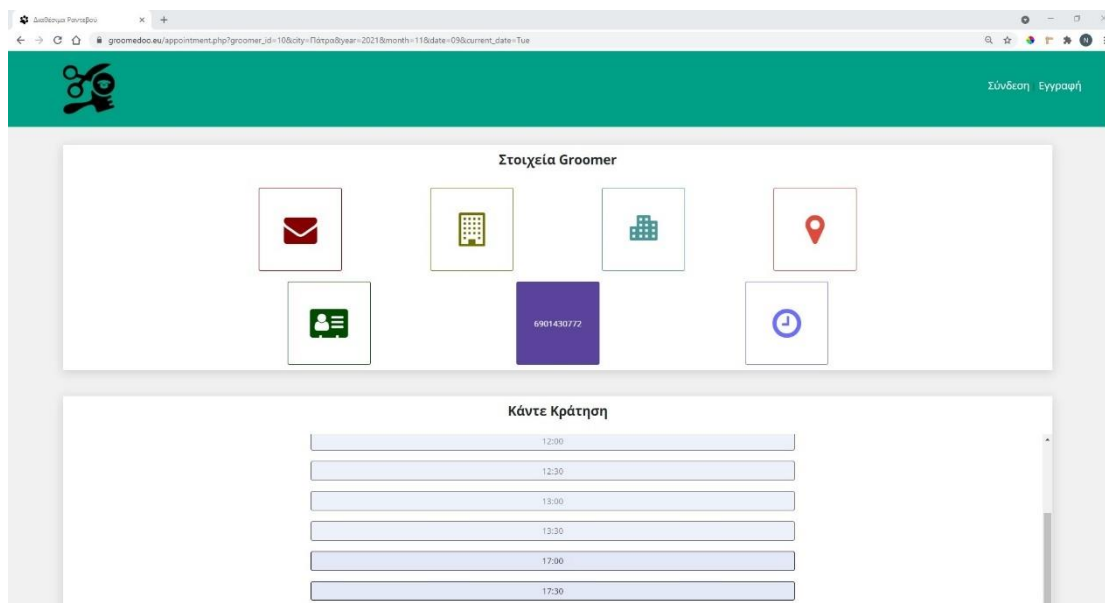
### Αναζήτηση Αποτελεσμάτων με βάση την πόλη και την ημερομηνία

Εφόσον ο χρήστης συμπληρώσει την φόρμα της αρχικής σελίδας με βάση τις επιλογές του, θα οδηγηθεί σε μια σελίδα που θα εμφανίζονται τα αποτελέσματα της αναζήτησης. Αρχικά, θα μπορεί να δει τις επιλογές του όσον αφορά την ημερομηνία και την πόλη και στην συνέχεια μπορεί να δει τους διαθέσιμους groomer με βάση τις επιλογές που περιγράψαμε. Στις καρτέλες των groomer μπορούμε να δούμε μερικές πληροφορίες για τον κάθε groomer, όπως είναι το email, η πόλη, η διεύθυνση και το τηλέφωνο. Η σελίδα αναζήτησης παρουσιάζεται στο παρακάτω στιγμιότυπο.

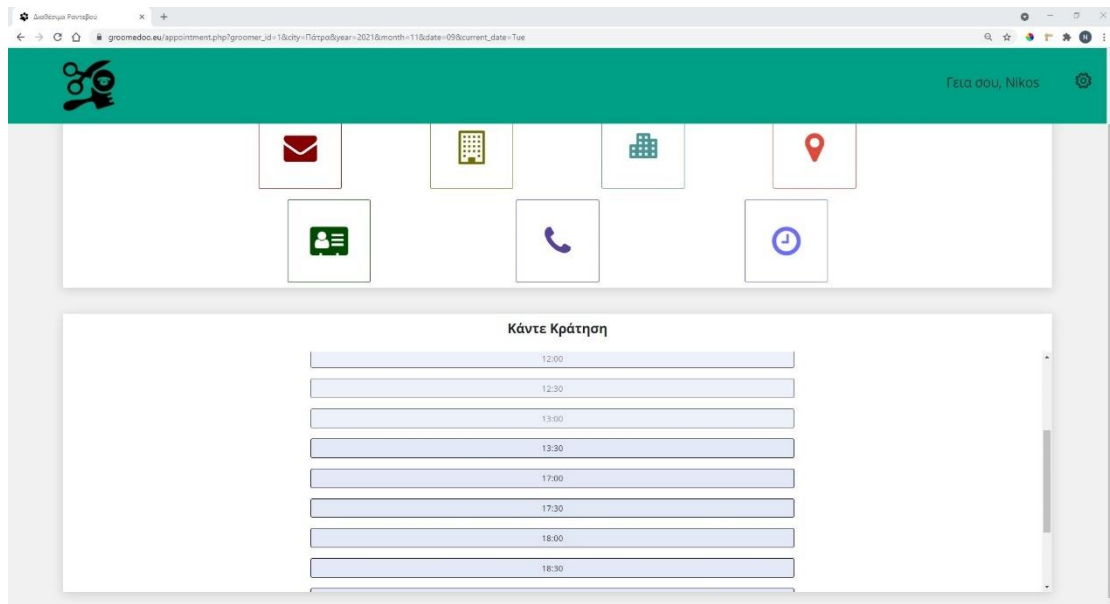


Εικόνα 3.1.31: Σελίδα αναζήτησης για groomer

Εφόσον ο χρήστης επιλέξει τον groomer που επιθυμεί, οδηγείται στην σελίδα όπου θα πρέπει να κάνει αίτηση για ραντεβού. Η σελίδα αυτή παρουσιάζεται στην Εικόνα 3.1.32. Αρχικά μπορούμε να δούμε τα στοιχεία του groomer περνώντας το ποντίκι πάνω από στοιχείο που θέλουμε να δούμε. Έπειτα μπορούμε να δούμε τις διαθέσιμες ώρες για να πραγματοποιηθεί μια αίτηση για ραντεβού. Οι ώρες οι οποίες είναι μη-επιλέξιμες υποδηλώνουν ότι η συγκεκριμένη ώρα είναι μη διαθέσιμη, είτε επειδή η συγκεκριμένη ώρα είναι ήδη κλεισμένη από κάποιο άλλο ραντεβού, είτε επειδή η τωρινή ώρα είναι μεγαλύτερη από την διαθέσιμη ώρα

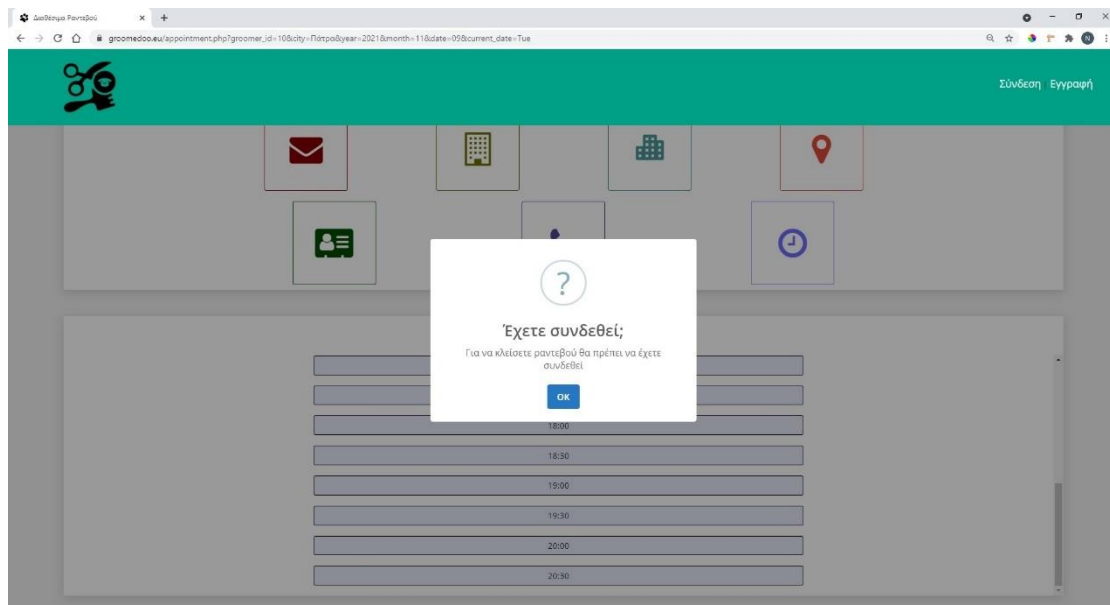


Εικόνα 3.1.32: Σελίδα αιτήσεων ραντεβού



Εικόνα 3.1.33: Σελίδα αιτήσεων ραντεβού (συνέχεια)

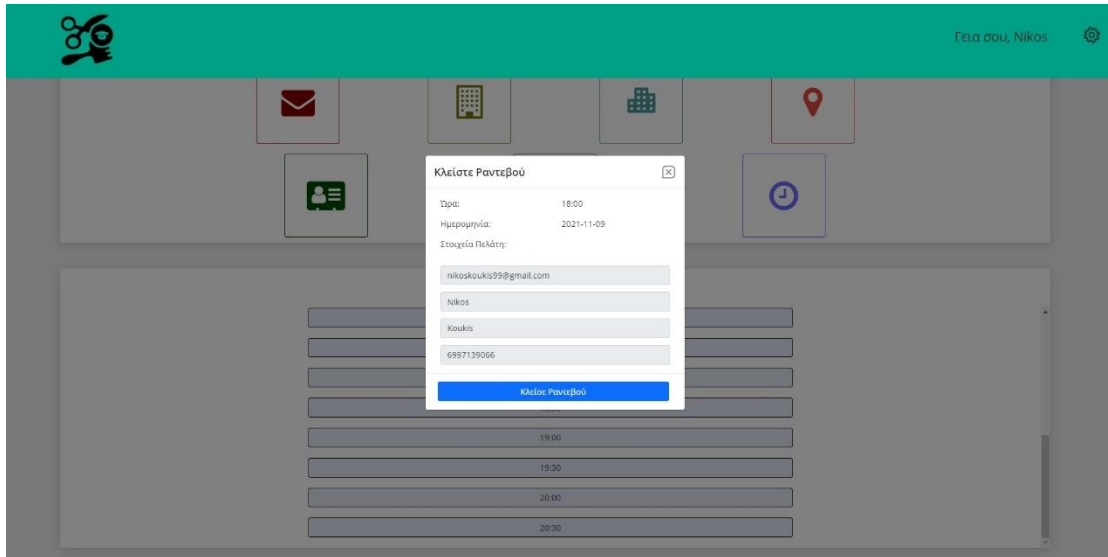
Σε περίπτωση που ο χρήστης πάει να πραγματοποιήσει αίτηση για ραντεβού θα του εμφανιστεί αναδυόμενο παράθυρο με κατάλληλο μήνυμα όπως παρουσιάζεται στην Εικόνα 3.1.34.



Εικόνα 3.1.34: Επιβεβαίωση σύνδεσης για αίτημα ραντεβού

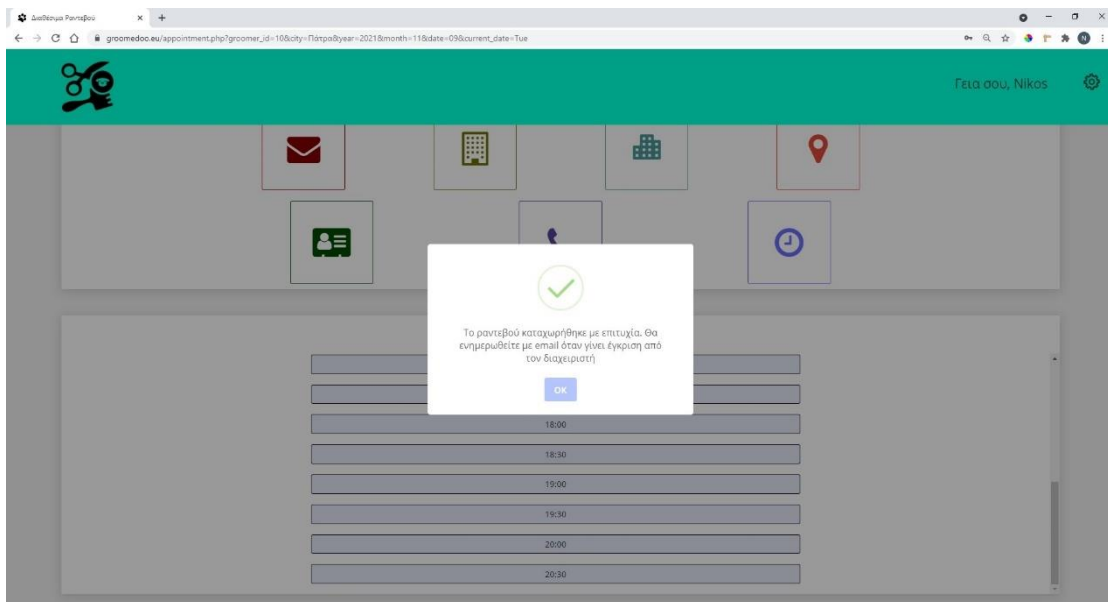
Αφού ο χρήστης είναι συνδεδεμένος στη πλατφόρμα μπορεί να κάνει αίτημα για ραντεβού επιλέγοντας μία από τις διαθέσιμες ώρες. Εφόσον επιλέξει την ώρα που επιθυμεί θα του εμφανιστεί ένα αναδυόμενο παράθυρο με τα στοιχεία του και την ημερομηνία με την ώρα του ραντεβού, όπως παρουσιάζεται στην Εικόνα 3.1.35.





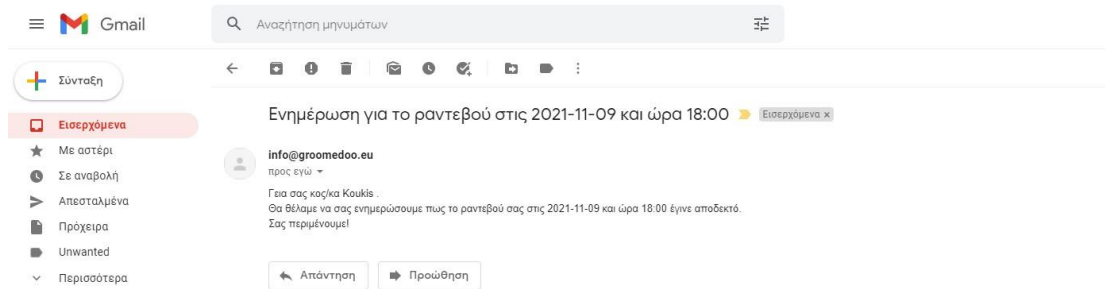
Εικόνα 3.1.35: Επιλογή διαθέσιμων ωρών για ραντεβού

Αφού ελέγξει όλα τα στοιχεία μπορεί να κάνει αίτημα για ραντεβού πατώντας το κουμπί «Κλείσε Ραντεβού». Εφόσον το κάνει, θα του εμφανιστεί ένα αναδυόμενο παράθυρο, όπως παρουσιάζεται παρακάτω, θα γίνει ανανέωση της σελίδας και η συγκεκριμένη ώρα θα γίνει μη διαθέσιμη για τους υπόλοιπους χρήστες. Το αίτημα έχει υποβληθεί με επιτυχία αλλά θα πρέπει να γίνει αποδεκτό από τον groomer. Σε περίπτωση απόρριψης του αιτήματος, η συγκεκριμένη ώρα θα γίνει και πάλι διαθέσιμη.



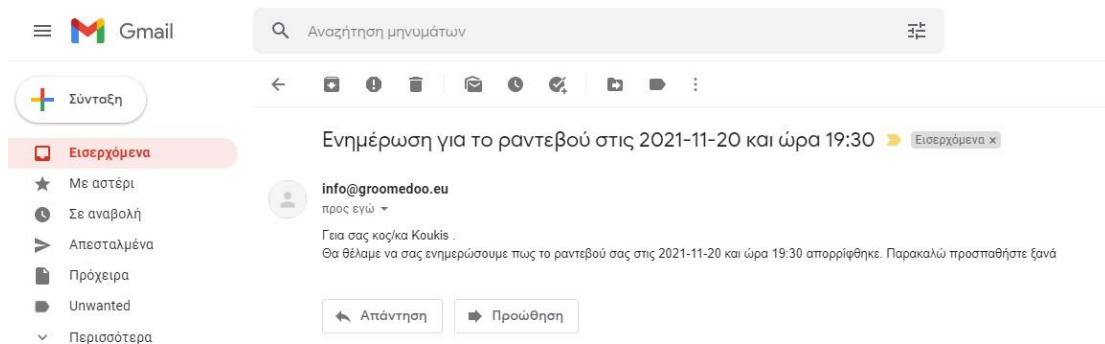
Εικόνα 3.1.36: Επιτυχημένη καταχώρηση ραντεβού

Ακολουθεί email μετά την αποδοχή του αιτήματος για ραντεβού από τον groomer.



Εικόνα 3.1.37: Email επιβεβαίωσης ραντεβού από τον groomer

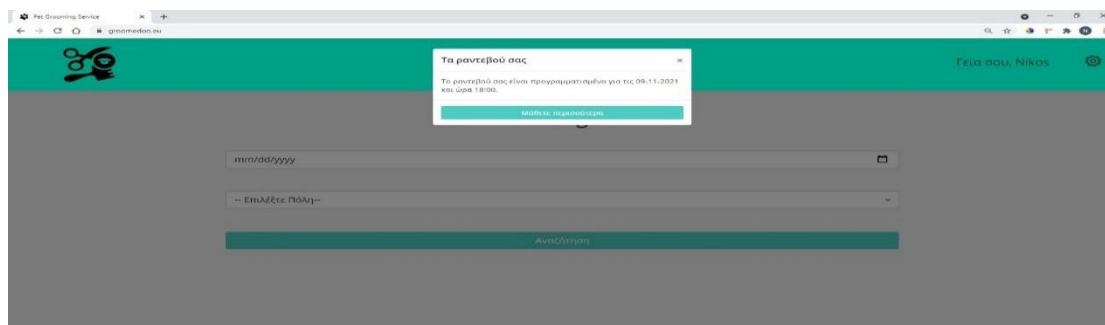
Ακολουθεί email μετά την απόρριψη του αιτήματος για ραντεβού από τον groomer.



Εικόνα 3.1.38: Email απόρριψης ραντεβού από τον groomer

### Ενημερωτικό αναδυόμενο παράθυρο υπενθύμισης ραντεβού

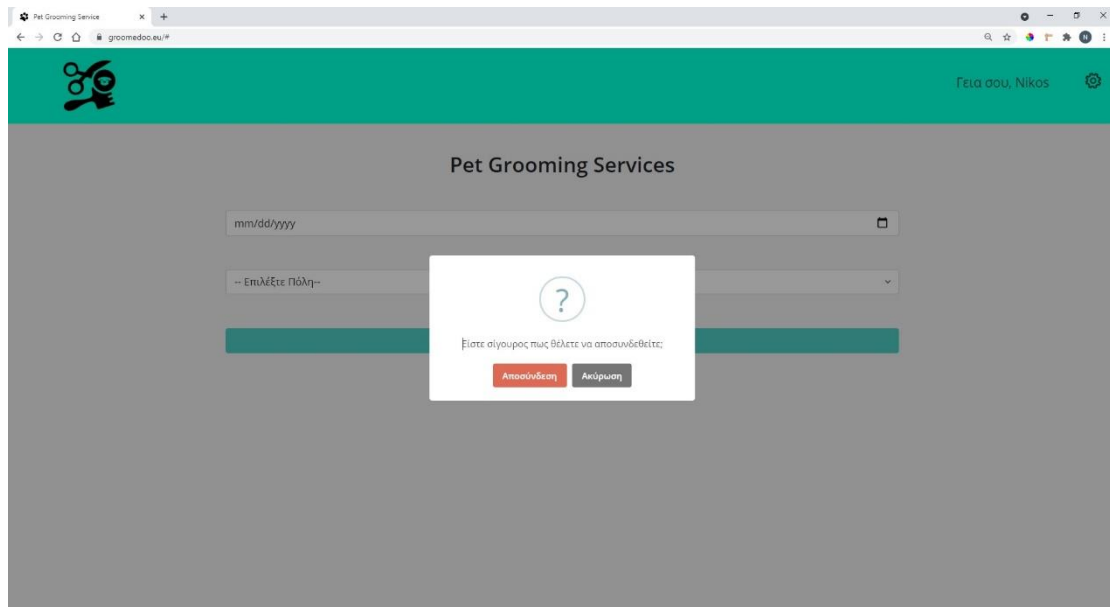
Για κάθε χρήστη, σε όλες τις σελίδες της διαδικτυακής εφαρμογής, σε περίπτωση που κάποιος χρήστης έχει προγραμματισμένο ραντεβού, θα του εμφανίζεται (ανά 5 λεπτά) ένα αναδυόμενο παράθυρο ενημέρωσης για το ραντεβού, όπως παρουσιάζεται στην Εικόνα 3.1.39. Με το κουμπί «Μάθετε περισσότερα» ο χρήστης θα μεταφέρεται στη σελίδα με την λίστα των ραντεβού που περιγράψαμε παραπάνω.



Εικόνα 3.1.39: Ενημερωτικό αναδυόμενο παράθυρο υπενθύμισης ραντεβού

### *Αποσύνδεση User από την εφαρμογή*

Για να αποσυνδεθεί ένας χρήστης από την εφαρμογή αρκεί να πατήσει «Αποσύνδεση» από το μενού και να πατήσει «Αποσύνδεση» από το αναδυόμενο παράθυρο, όπως παρουσιάζεται στην Εικόνα 3.1.40.



*Εικόνα 3.1.40: Αποσύνδεση χρήστη*

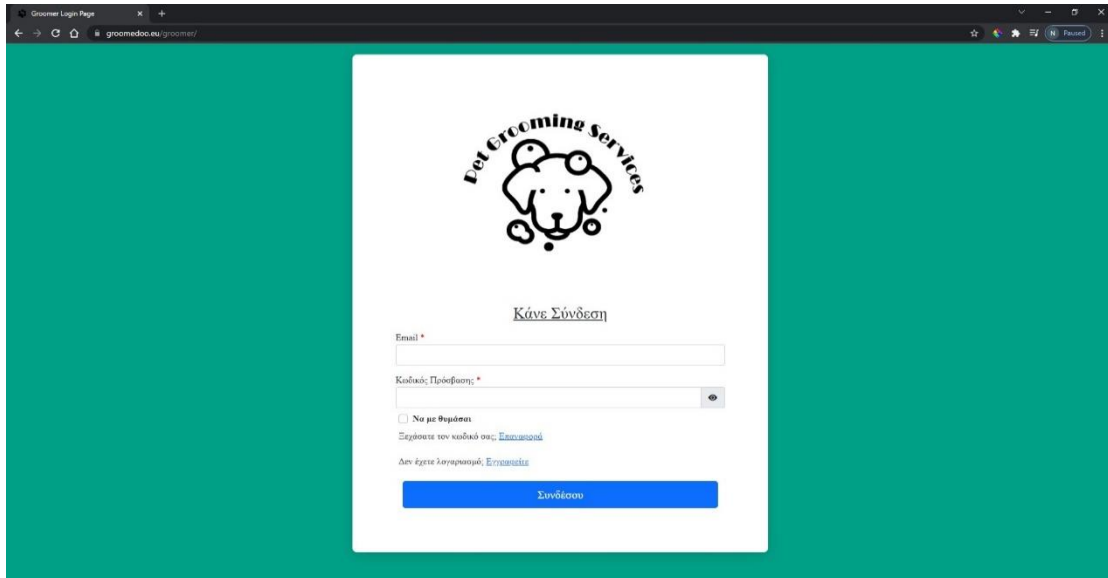
Στο σημείο αυτό αξίζει να σημειωθεί πως χάριν ευκολίας υπάρχει συμπληρωματικό υλικό στο Παράρτημα 2 της εργασίας που περιέχει τα βασικά σημεία του κώδικα από την πλευρά του user- πελάτη.

## 3.2 Εγχειρίδιο χρήσης για τον διαχειριστή - groomer

### *Αρχική Σελίδα Εφαρμογής*

Η αρχική εικόνα της εφαρμογής από πλευράς διαχειριστή – groomer απεικονίζεται στην Εικόνα 3.2.1. Για να μεταφερθείς στην οθόνη αυτή αρκεί να πληκτρολογήσεις τη διεύθυνση <https://groomedoo.eu/groomer/>. Η αρχική οθόνη είναι και η σελίδα όπου ο groomer πραγματοποιεί είσοδο στην εφαρμογή.

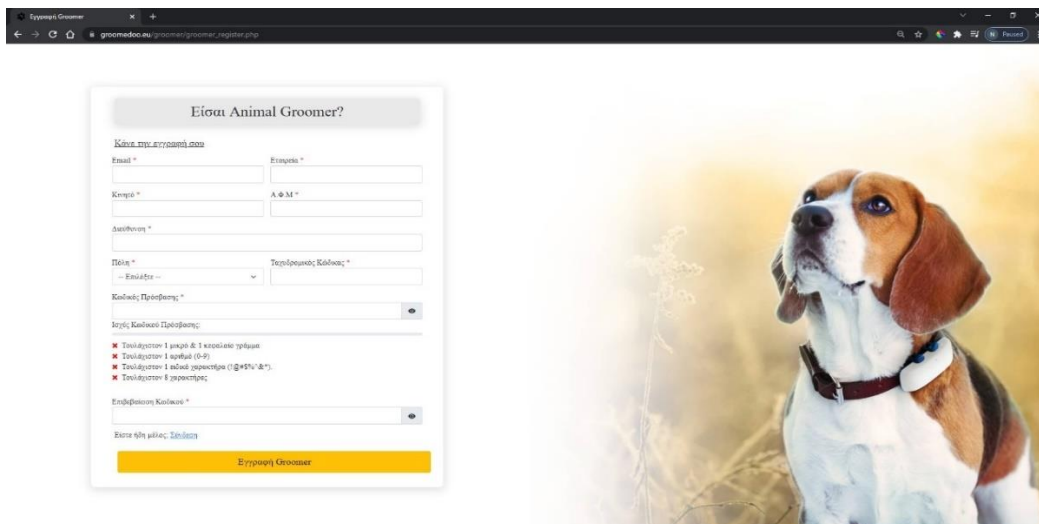
Όπως αναφέρθηκε και προηγουμένως, χάριν ευκολίας υπάρχει συμπληρωματικό υλικό στο Παράρτημα 3 της εργασίας που περιέχει τα βασικά σημεία του κώδικα από την πλευρά του groomer-διαχειριστή.



Εικόνα 3.2.1: Αρχική οθόνη

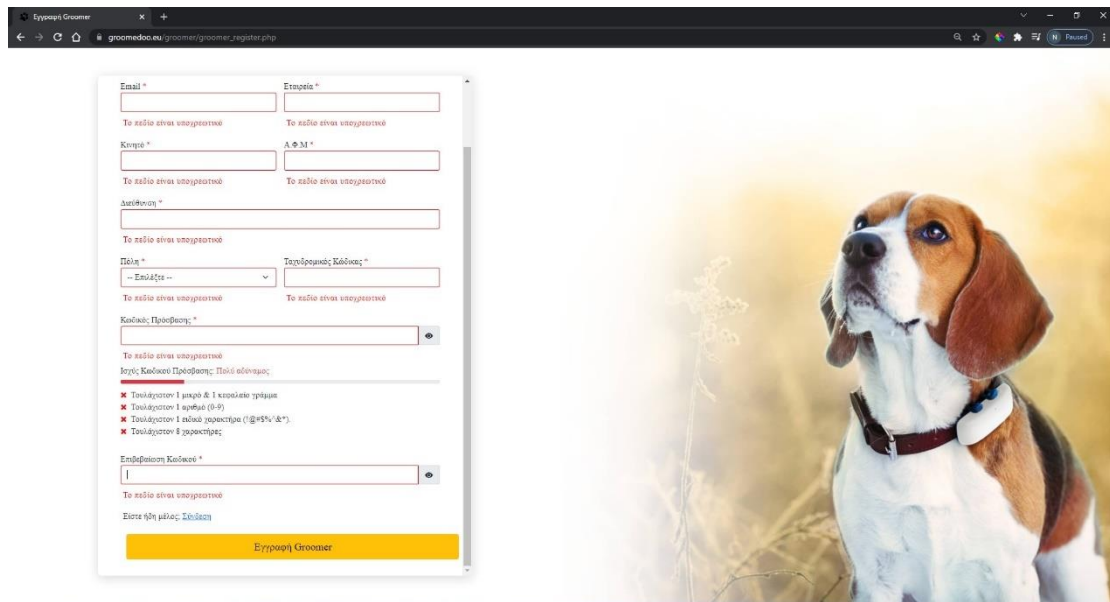
### Εγγραφή Groomer

Στην Εικόνα 3.2.2 παρουσιάζεται η φόρμα εγγραφής για τους διαχειριστές - groomers. Στην φόρμα εγγραφής έχει σχεδιαστεί ένα real-time password strength checker, το οποίο βοηθάει τον χρήστη να δημιουργήσει έναν ισχυρό κωδικό πρόσβασης.



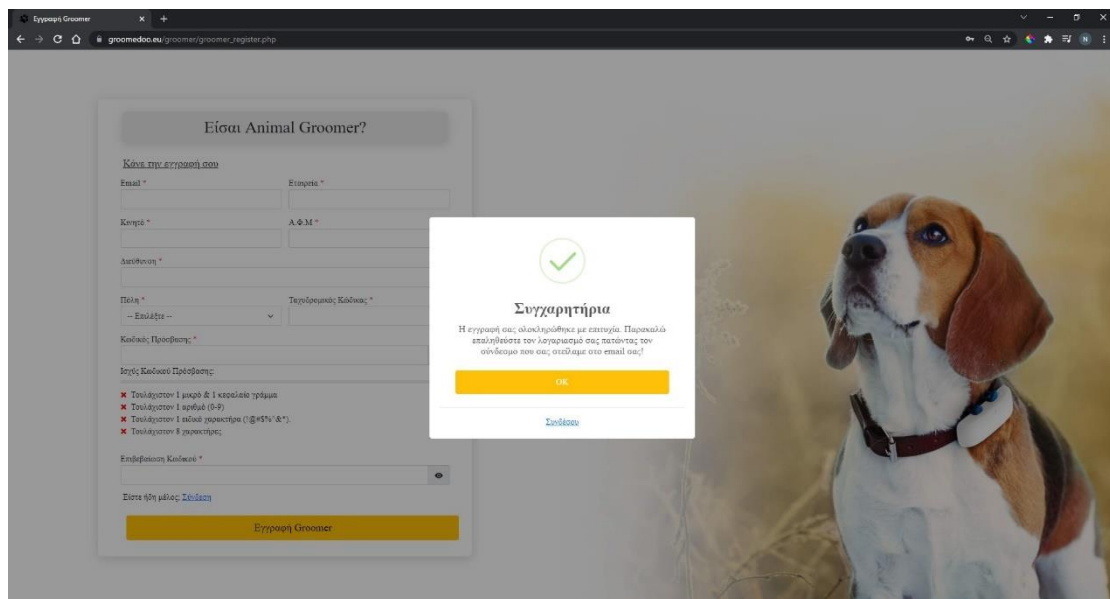
Εικόνα 3.2.2: Φόρμα εγγραφής για groomers

Στην φόρμα εγγραφής έχει εφαρμοστεί επικύρωση στοιχείων, καθώς ελέγχει αν η φόρμα είναι πλήρως συμπληρωμένη ή κάποιο πεδίο της είναι μη έγκυρο. Η μη επικυρωμένη φόρμα εγγραφής παρουσιάζεται στην Εικόνα 3.2.3.



Εικόνα 3.2.3: Μη επικυρωμένη φόρμα εγγραφής

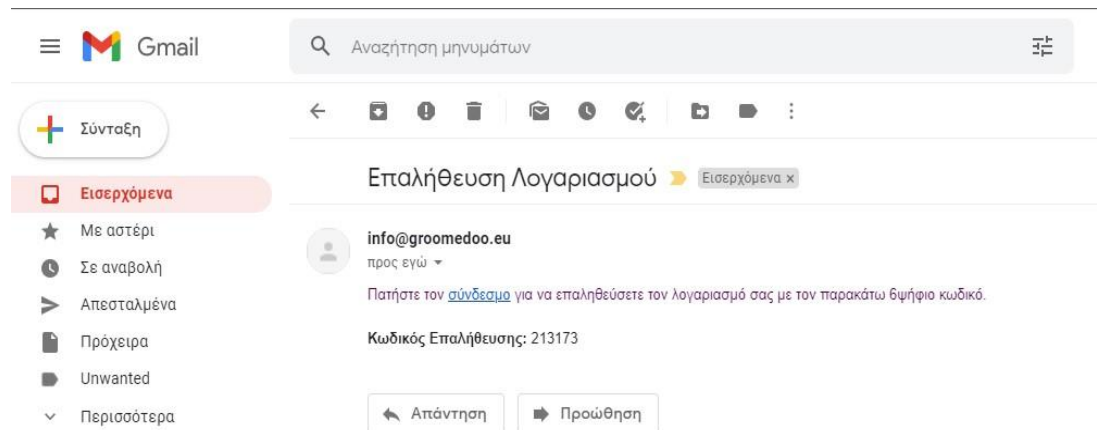
Εφόσον ο διαχειριστής - groomer συμπληρώσει ορθά όλα τα πεδία της φόρμας και πατήσει υποβολή θα του εμφανίσει ένα αναδυόμενο παράθυρο που θα τον ενημερώνει ότι η φόρμα υποβλήθηκε με επιτυχία, ενώ επίσης θα τον ενημερώνει πως ο λογαριασμός του θα πρέπει να επαληθευτεί, όπως παρουσιάζεται στην Εικόνα 3.2.4.



Εικόνα 3.2.4: Επιτυχημένη επικύρωση στοιχείων φόρμας εγγραφής

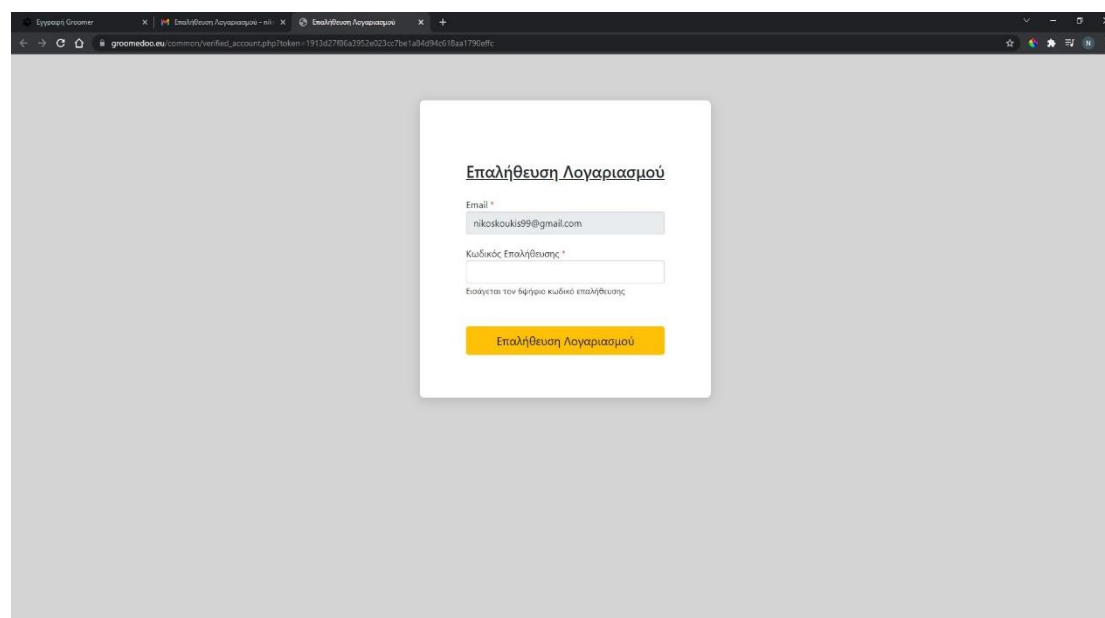
Στο email που χρησιμοποίησε κατά την εγγραφή του, θα του σταλθεί ένας σύνδεσμος επαλήθευσης λογαριασμού. Θα πρέπει να ακολουθήσει τις οδηγίες του email, έτσι ώστε να επαληθεύσει τον λογαριασμό του και να μπορεί να πραγματοποιήσει είσοδο στην εφαρμογή. Εάν ένας λογαριασμός δεν έχει επαληθευτεί,

δε θα μπορεί να συνδεθεί στην διαδικτυακή εφαρμογή. Στο email επαλήθευσης υπάρχει ένας σύνδεσμος τον οποίο θα πρέπει να επισκεφτεί ο διαχειριστής - groomer ώστε να καταχωρήσει τον δψήφιο κωδικό για να επαληθευτεί ο λογαριασμός. Στην Εικόνα 3.2.5 παρουσιάζεται ένα email επαλήθευσης λογαριασμού.



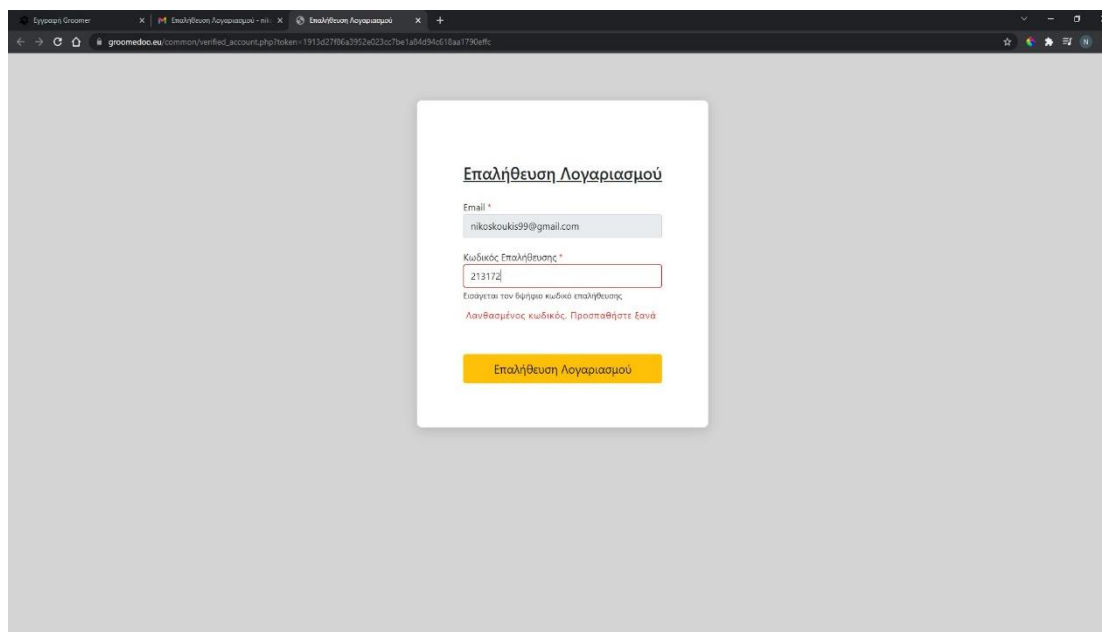
Εικόνα3.2.5: Email επαλήθευσης λογαριασμού

Στην Εικόνα 3.2.6 παρουσιάζεται η σελίδα που οδηγήθηκε ο groomer πατώντας τον σύνδεσμο του email επαλήθευσης. Η σελίδα αυτή περιέχει μια φόρμα με το δηλωθέν email της εγγραφής το οποίο ο groomer δε μπορεί να τροποποιήσει και ένα πεδίο ώστε ο groomer να τοποθετήσει τον κωδικό επαλήθευσης που του έχει σταλθεί στο email.



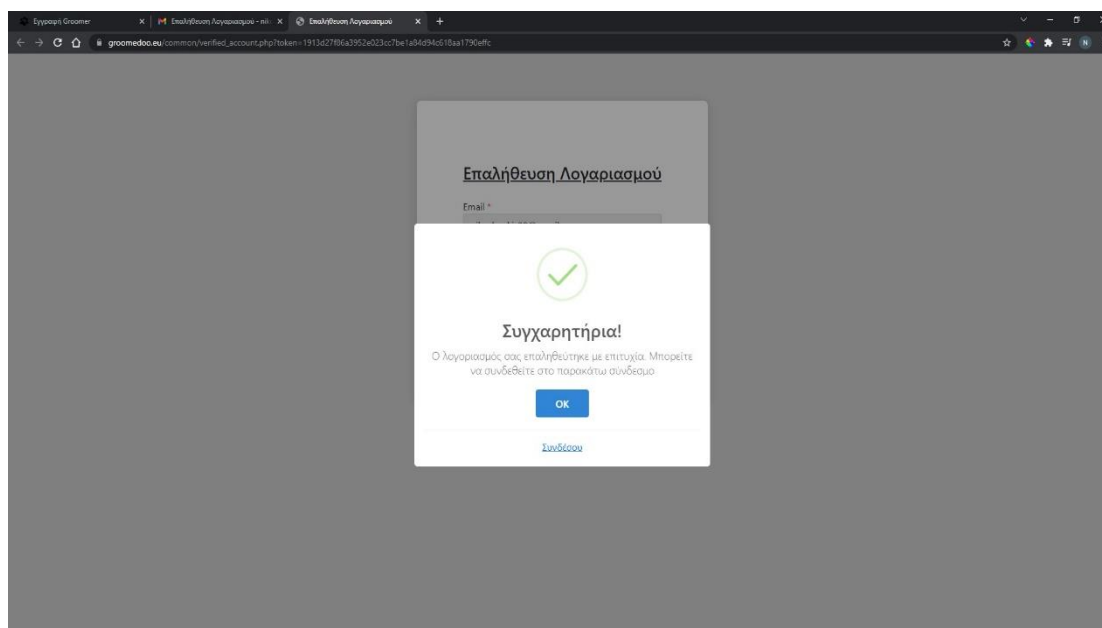
Εικόνα 3.2.6: Επαλήθευση λογαριασμού

Στην παραπάνω φόρμα έχει εφαρμοστεί επικύρωση στοιχείων σε περίπτωση που το πεδίο του κωδικού είναι κενό ή μη έγκυρο. Ένας μη έγκυρος κωδικός επαλήθευσης παρουσιάζεται στην Εικόνα 3.2.7.



Εικόνα 3.2.7: Μη έγκυρος κωδικός επαλήθευσης

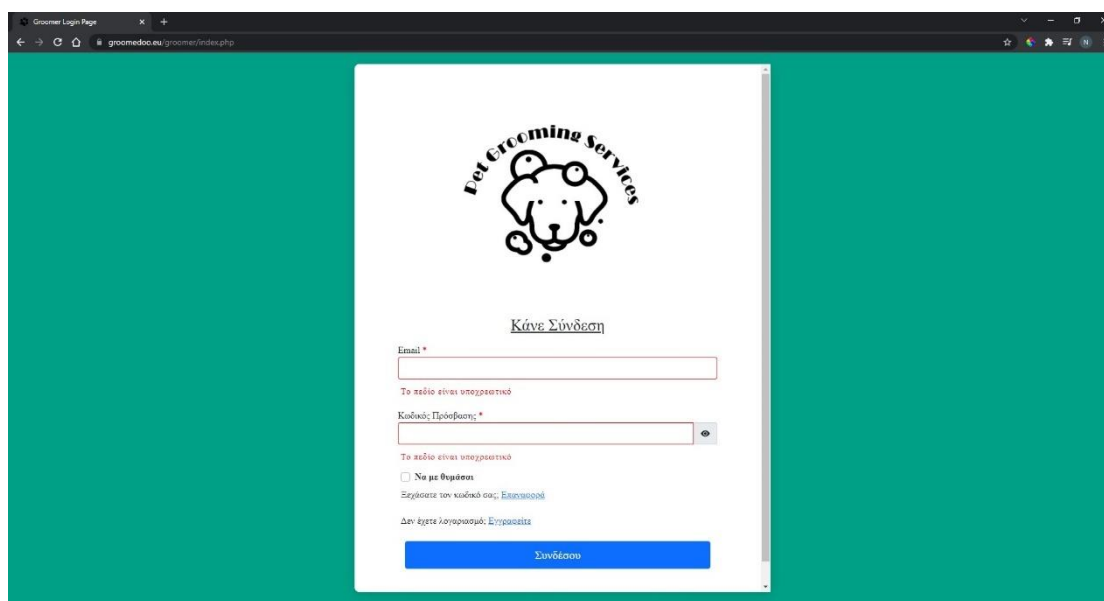
Εφόσον ο χρήστης εισάγει τον σωστό κωδικό επαλήθευσης, θα ενημερωθεί με ένα αναδυόμενο παράθυρο και θα μεταφερθεί στην αρχική σελίδα της εφαρμογής έχοντας τον λογαριασμό του επαληθευμένο, όπως παρουσιάζεται στην Εικόνα 3.2.8.



Εικόνα 3.2.8: Επιτυχημένη επαλήθευση λογαριασμού

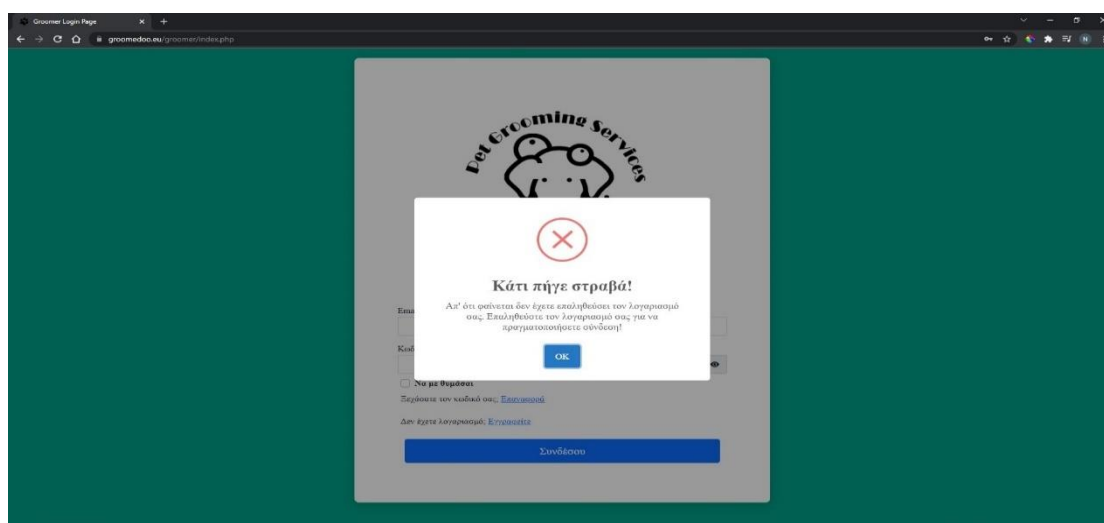
## Σύνδεση Groomer

Στην Εικόνα 3.2.9 παρουσιάζεται η φόρμα σύνδεσης για τους groomers (αρχική οθόνη εφαρμογής). Μία σύνδεση μπορεί να πραγματοποιηθεί μόνο όταν ο λογαριασμός είναι επαληθευμένος. Στην φόρμα σύνδεσης έχει εφαρμοστεί επικύρωση στοιχείων, καθώς ελέγχει αν η φόρμα είναι πλήρως συμπληρωμένη, εάν έχει επαληθευτεί ή ακόμα και κάποιο πεδίο της είναι μη έγκυρο. Η μη επικυρωμένη φόρμα σύνδεσης λόγω μη συμπλήρωσης κάποιου πεδίου ή μη έγκυρου πεδίου παρουσιάζεται στο παρακάτω στιγμιότυπο.



Εικόνα 3.2.9: Μη επικυρωμένη φόρμα σύνδεσης

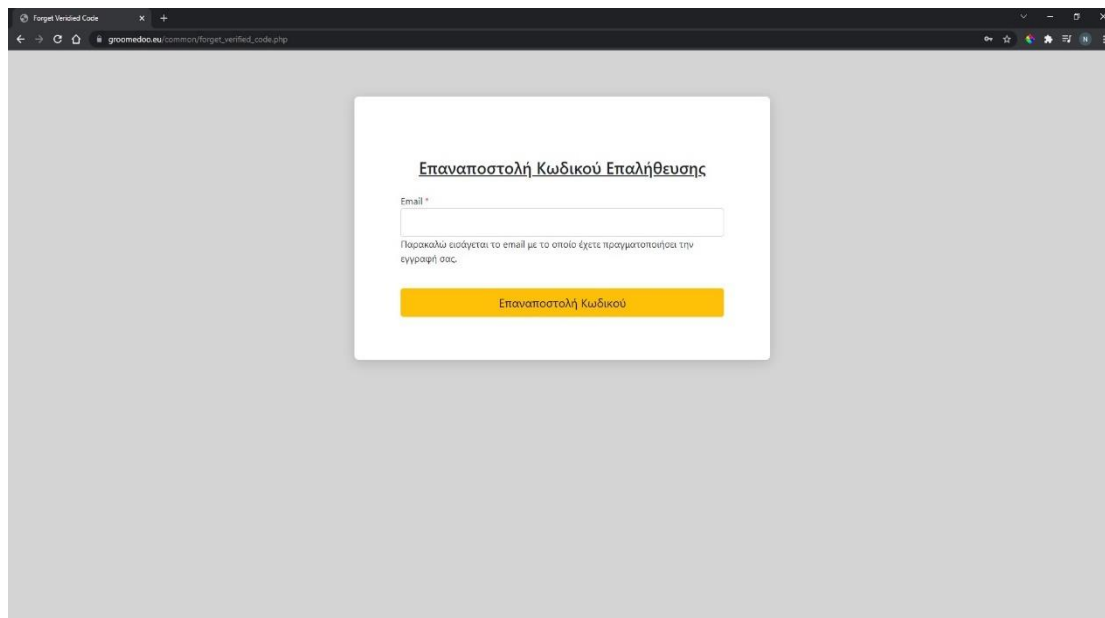
Όταν μια φόρμα υποβληθεί αλλά ο λογαριασμός δεν έχει επαληθευτεί θα του εμφανίσει ένα αναδυόμενο παράθυρο, όπως παρουσιάζεται παρακάτω.



Εικόνα 3.2.10: Σφάλμα υποβληθείσας φόρμας

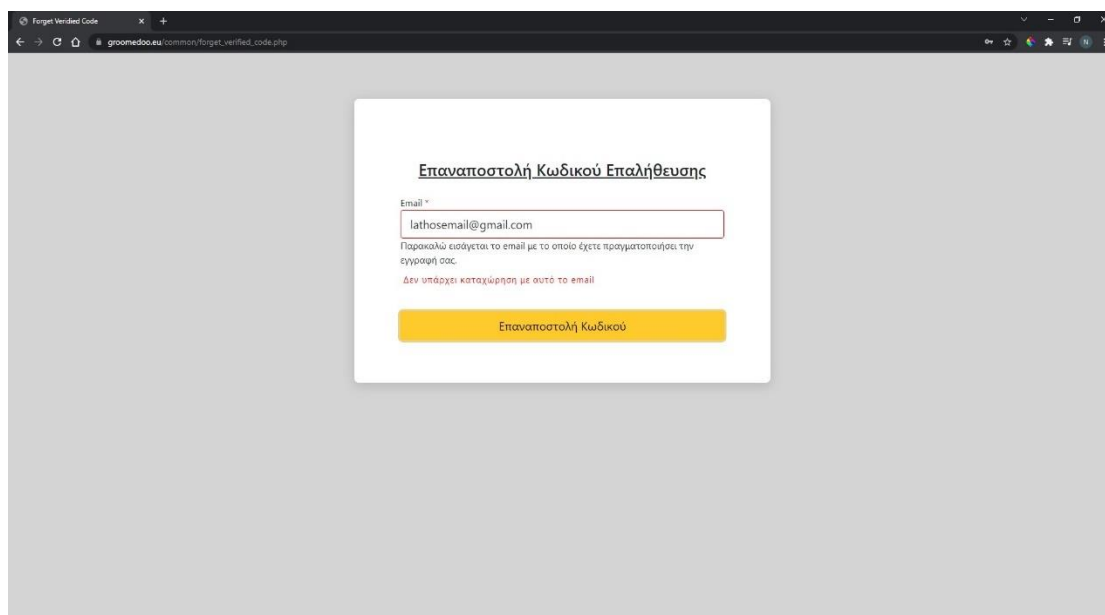


Όταν ο groomer πατήσει «OK» στο αναδυόμενο παράθυρο θα τον μεταφέρει σε μια καινούργια σελίδα ώστε να τοποθετήσει το email που έχει κάνει εγγραφή έτσι ώστε να του σταλθεί στο email του, ένας κωδικός επαλήθευσης κάνοντας τα βήματα που έχουμε περιγράψει παραπάνω.



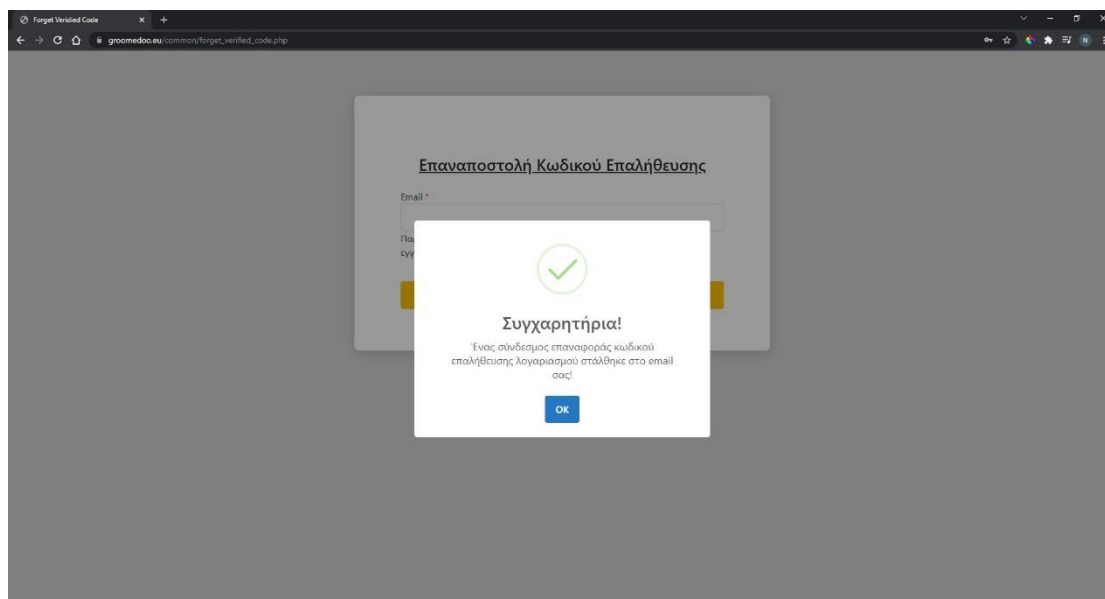
Εικόνα 3.2.11 Επαναποστολή κωδικού επαλήθευσης

Στην παραπάνω φόρμα έχει εφαρμοστεί επικύρωση στοιχείων, ώστε να αποφευχθεί το γεγονός να τοποθετήσει άλλο email από εκείνο που είχε χρησιμοποιήσει κατά την εγγραφή του groomer.



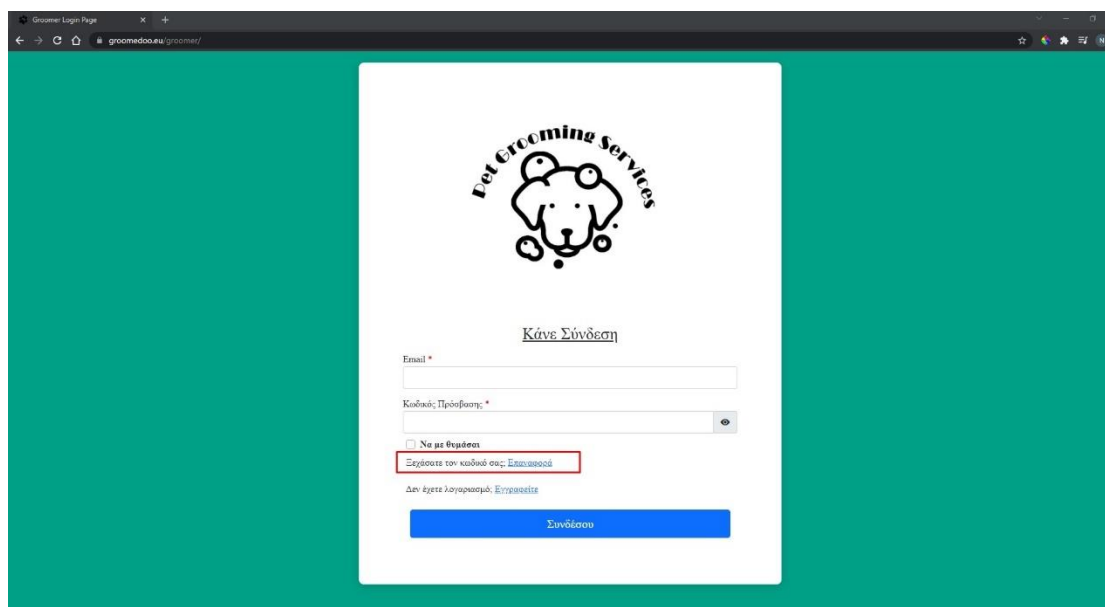
Εικόνα 3.2.12: Επικύρωση στοιχείων

Όταν η φόρμα υποβληθεί με το σωστό email, τότε ένα αναδυόμενο παράθυρο θα δώσει στον groomer τις κατάλληλες οδηγίες που περιγράψαμε παραπάνω ώστε να επαληθεύσει τον λογαριασμό του.



Εικόνα 3.2.13: Οδηγίες επαλήθευσης λογαριασμού

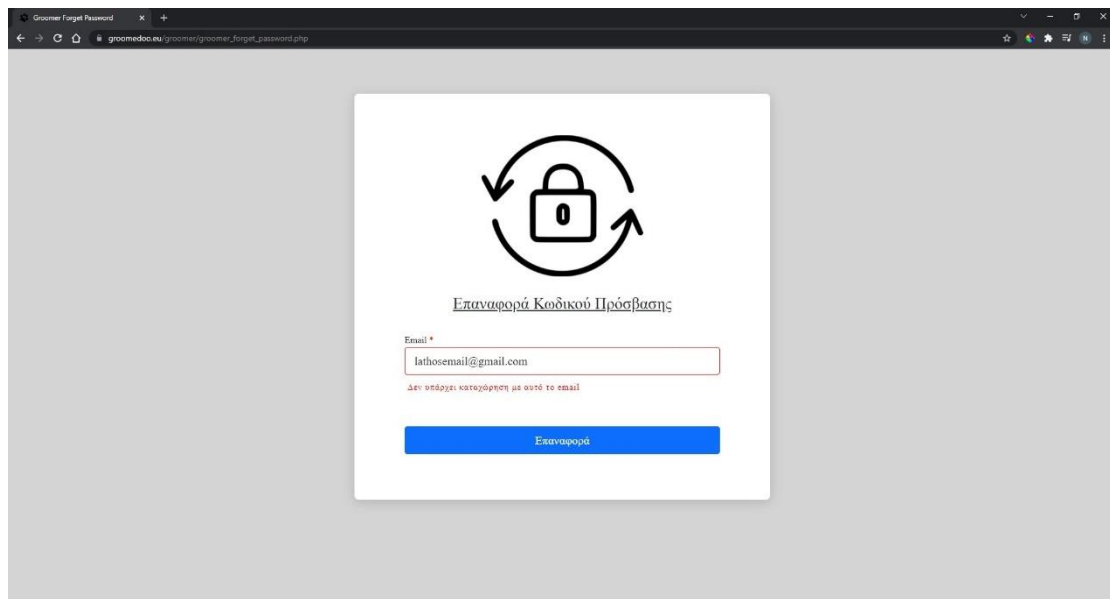
Σε περίπτωση που ο groomer ξεχάσει τον κωδικό πρόσβασης μπορεί να κάνει επαναφορά κωδικού πρόσβασης πατώντας τον βοηθητικό σύνδεσμο που υπάρχει στην φόρμα σύνδεσης, όπως παρουσιάζεται στην Εικόνα 3.2.14.



Εικόνα 3.2.14: Επαναφορά κωδικού πρόσβασης

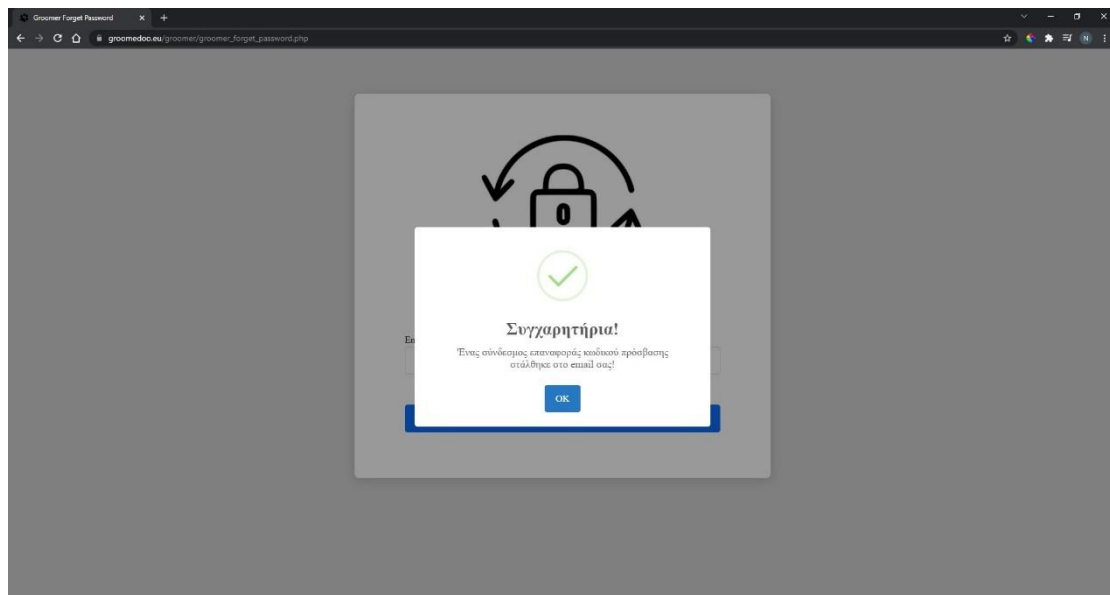
Ο παραπάνω σύνδεσμος, θα τον οδηγήσει σε μια καινούργια σελίδα, όπως παρουσιάζεται παρακάτω, ώστε να τοποθετήσει το email που έχει κάνει εγγραφή, έτσι

ώστε να του σταλούν στο email οδηγίες ανάκτησης κωδικού πρόσβασης. Στην παρακάτω φόρμα έχει εφαρμοστεί επικύρωση στοιχείων, ώστε να αποφευχθεί το να τοποθετήσει άλλο email από εκείνο που είχε χρησιμοποιήσει κατά την εγγραφή του.



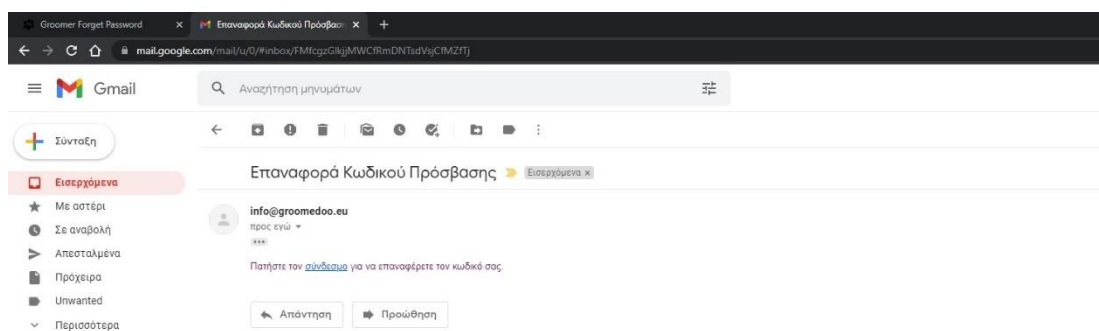
Εικόνα 3.2.15: Επικύρωση στοιχείων επαναφοράς κωδικού πρόσβασης

Εφόσον η φόρμα υποβληθεί με επιτυχία ένα αναδυόμενο παράθυρο θα τον ενημερώσει πως ο σύνδεσμος επαναφοράς κωδικού έχει σταλθεί στο email του, όπως παρουσιάζεται παρακάτω.



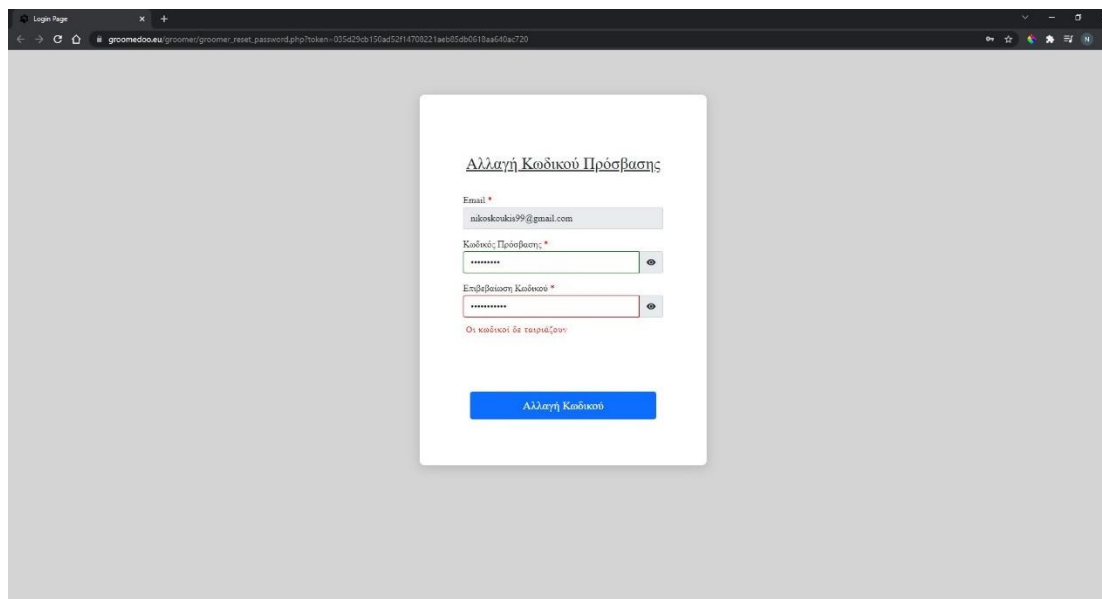
Εικόνα 3.2.16: Επιτυχημένη επαναφορά κωδικού πρόσβασης

Στην Εικόνα 3.2.17 παρουσιάζεται ένα email επαναφοράς κωδικού πρόσβασης.



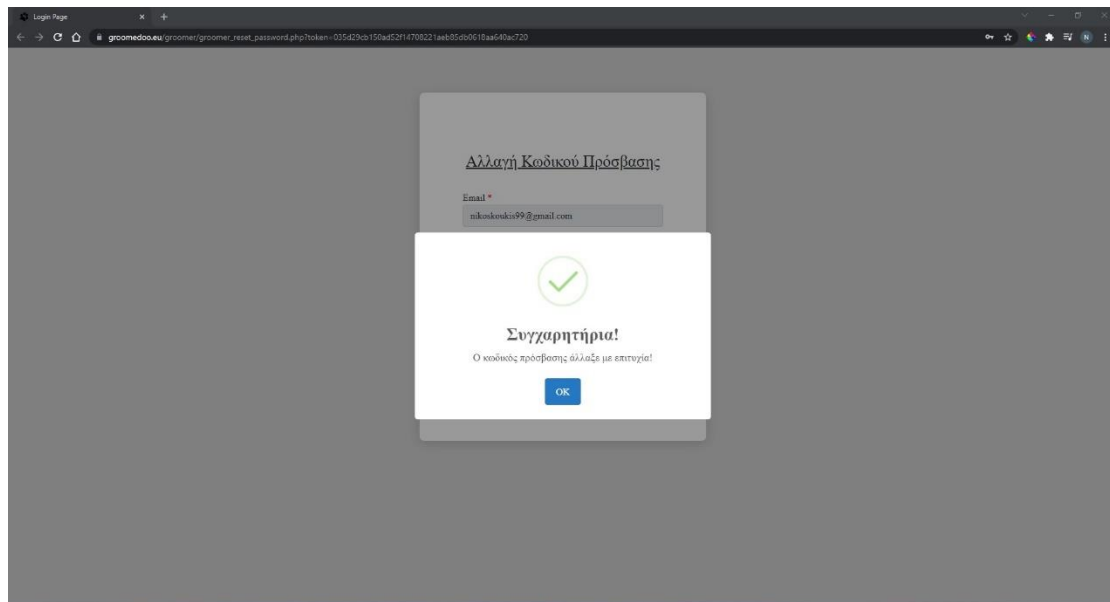
Εικόνα 3.2.17: Email επαναφοράς κωδικού πρόσβασης

Στην Εικόνα 3.2.18 παρουσιάζεται η σελίδα που οδηγήθηκε ο groomer πατώντας τον σύνδεσμο του email για επαναφορά κωδικού πρόσβασης. Η σελίδα αυτή περιέχει μια φόρμα με το δηλωθέν email της εγγραφής, το οποίο ο groomer δε μπορεί να τροποποιήσει, και δύο πεδία για να εισάγει ο χρήστης τον νέο κωδικό πρόσβασης. Στην παρακάτω φόρμα έχει εφαρμοστεί επικύρωση στοιχείων, ώστε να αποφευχθεί το γεγονός οι δύο κωδικοί πρόσβασης να μη ταιριάζουν. Σε αυτή την περίπτωση θα εμφανιστεί μήνυμα λάθους.



Εικόνα 3.2.18: Αλλαγή κωδικού πρόσβασης

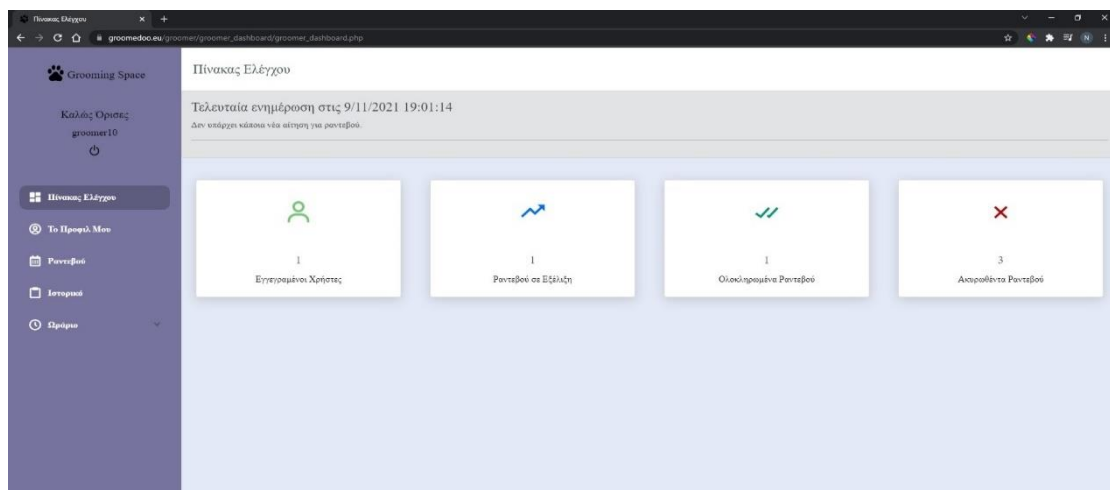
Αν η φόρμα δεν περιέχει κάποιο λάθος, θα υποβληθεί με επιτυχία ενημερώνοντας τον groomer με αναδυόμενο παράθυρο, όπως παρουσιάζεται στην Εικόνα 3.2.19.



Εικόνα 3.2.19: Επιτυχημένη αλλαγή κωδικού πρόσβασης

### Πίνακας Ελέγχου Διαχειριστή – Groomer

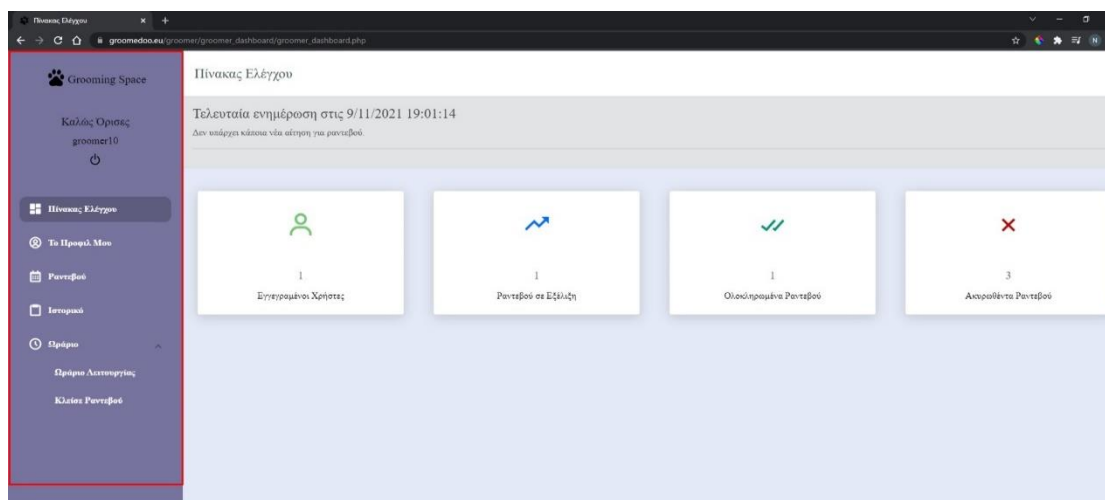
Εφόσον ο διαχειριστής – groomer πραγματοποιήσει σωστά την σύνδεσή του με όσα περιγράψαμε παραπάνω θα μεταφερθεί στο διαχειριστικό κομμάτι της εφαρμογής, τον πίνακα ελέγχου. Όλες οι σελίδες στο διαχειριστικό κομμάτι συνοδεύονται από την αριστερή πλαϊνή μπάρα, όπως παρουσιάζεται στην Εικόνα 3.2.20. Η πλαϊνή αυτή μπάρα, λειτουργεί σαν μενού και βοηθάει τον groomer να πλοηγηθεί στις διάφορες λειτουργίες της εφαρμογής εύκολα και γρήγορα.



Εικόνα 3.2.20: Πίνακας ελέγχου

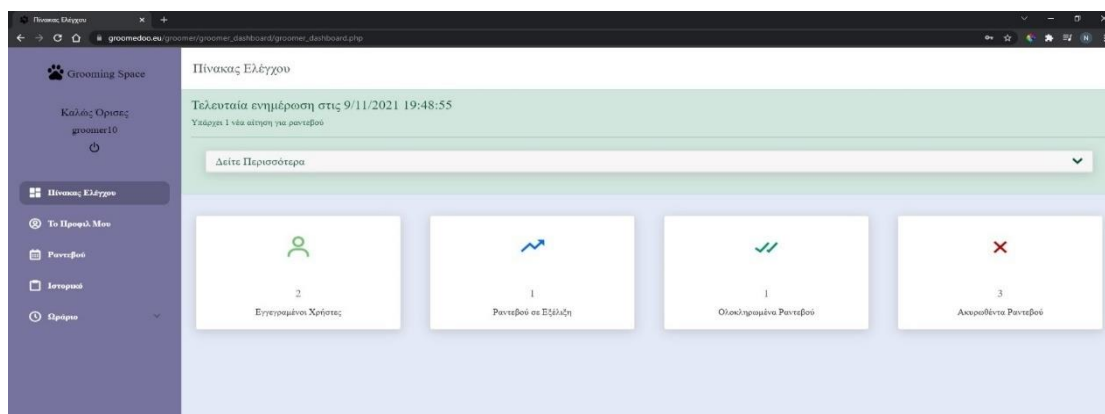
Η πλαϊνή μπάρα αποτελείται από το logo – Grooming Space, το οποίο σε οδηγεί στον πίνακα ελέγχου, από οποιαδήποτε σελίδα κι αν βρίσκεσαι, εμφανίζει την ονομασία της επιχείρησης καθώς και το κουμπί αποσύνδεσης, ώστε ο διαχειριστής να

μπορεί να αποσυνδεθεί από την εφαρμογή οποιαδήποτε στιγμή επιθυμεί. Επίσης, η πλαϊνή μπάρα έχει όλες τις λειτουργίες της εφαρμογής όπως το προφίλ του groomer, την λίστα με τα ραντεβού του, το ιστορικό των ραντεβού του καθώς και το ωράριο λειτουργίας που μπορεί να ορίσει για την επιχείρησή του. Τέλος, ο groomer έχει την δυνατότητα να προσθέσει χειροκίνητα ένα ραντεβού.



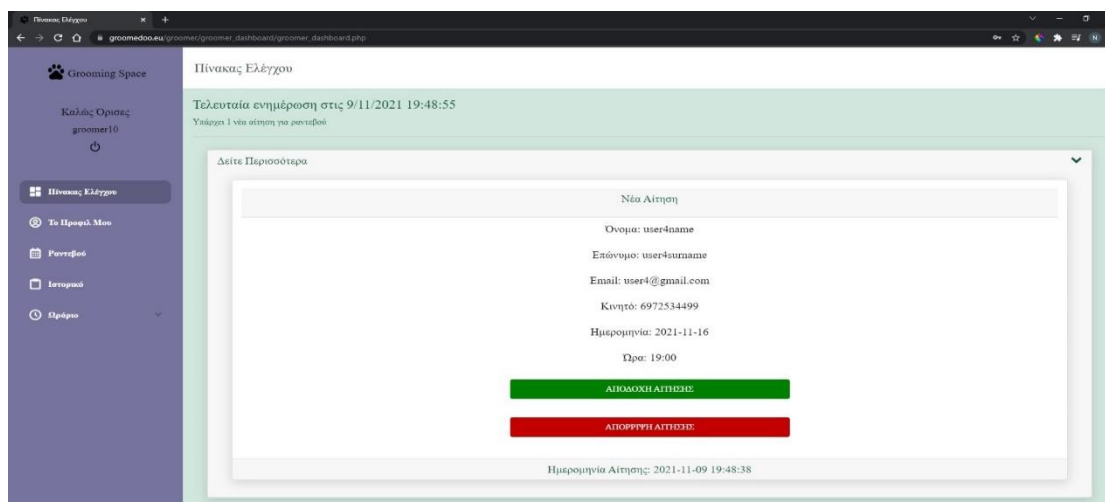
Εικόνα 3.2.21: Μπάρα λειτουργιών

Στον πίνακα ελέγχου (βλ. Εικόνα 3.2.20) υπάρχει ένα σημείο που λειτουργεί σαν ειδοποίηση για τον groomer. Ενημερώνει για την ημερομηνία και την ώρα που έγινε η τελευταία ενημέρωση καθώς και για το πλήθος των αιτήσεων για ραντεβού. Επίσης υπάρχουν και στατιστικά δεδομένα που δείχνουν το πλήθος των πελατών, το πλήθος των ραντεβού που βρίσκονται σε εξέλιξη και δεν έχουν ολοκληρωθεί, τα πλήρως ολοκληρωμένα ραντεβού και τέλος τα ακυρωθέντα ραντεβού του groomer. Στην Εικόνα 3.2.22 παρουσιάζεται ο πίνακας ελέγχου όταν υπάρχει καινούργιο αίτημα για ραντεβού.



Εικόνα 3.2.22: Νέο αίτημα ραντεβού

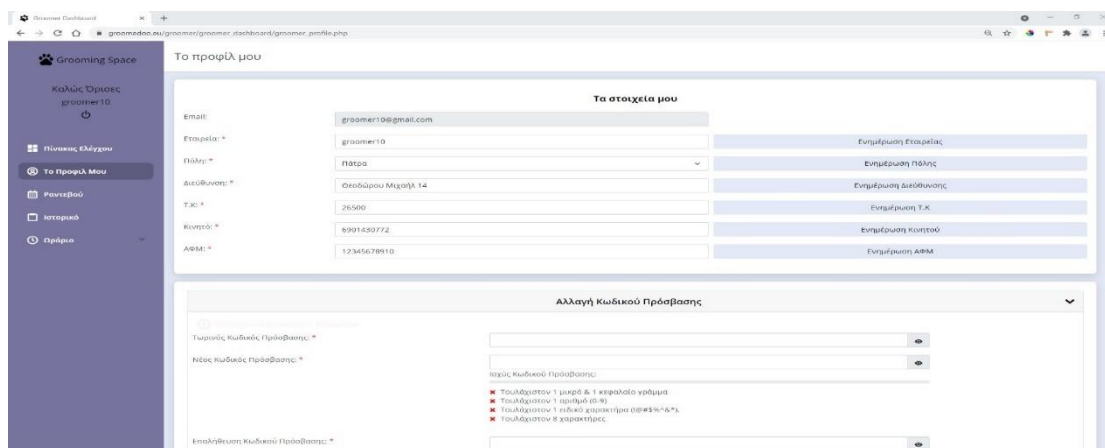
Πατώντας το κουμπί «Δείτε Περισσότερα», όπως παρουσιάζεται στην Εικόνα 3.2.23, θα μπορούμε να δούμε τις λεπτομέρειες του αιτήματος για ραντεβού, όπως το όνομα του πελάτη, το επώνυμό του, το email του, το κινητό του καθώς την ώρα και την ημερομηνία που επιθυμεί να κλείσει ραντεβού. Στη συνέχεια ο groomer θα αποφασίσει για την αποδοχή ή την απόρριψη του αιτήματος για ραντεβού. Και στις δύο περιπτώσεις ο πελάτης θα ενημερωθεί με αντίστοιχο μήνυμα όπως παρουσιάσαμε σε παραπάνω κεφάλαιο («Εγχειρίδιο χρήσης για τον χρήστη – user»).



Εικόνα 3.2.23: Λεπτομέρειες νέου αιτήματος ραντεβού

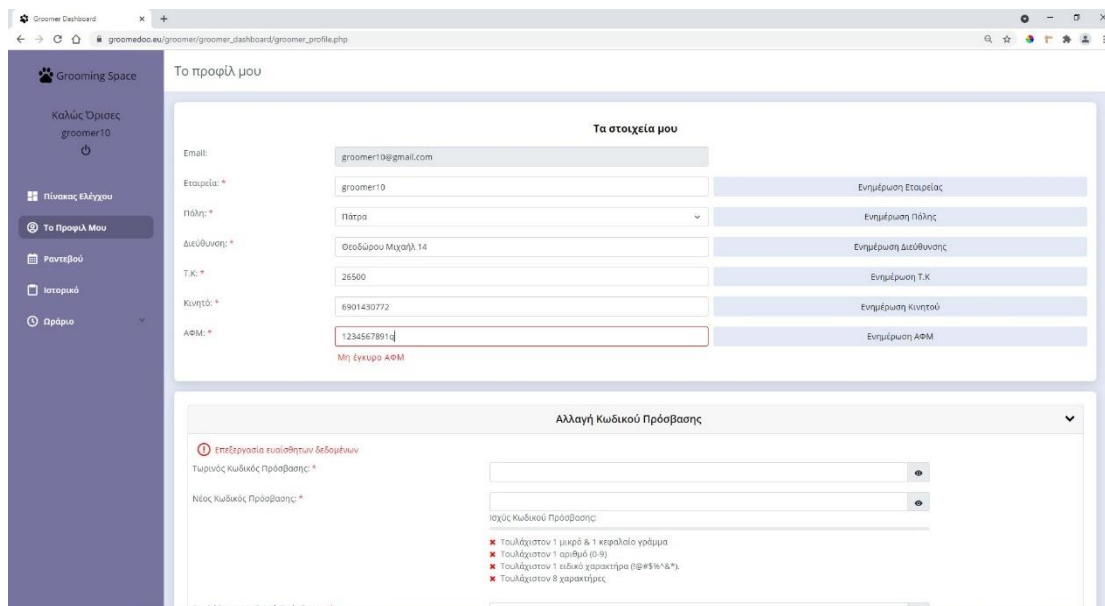
### Προφίλ Διαχειριστή – Groomer

Στην Εικόνα 3.2.24 παρουσιάζεται η σελίδα με το προφίλ του groomer. Αρχικά βλέπουμε τα στοιχεία του groomer, τα οποία μπορεί να επεξεργαστεί εκτός από το email του. Σε όλα τα πεδία έχει εφαρμοστεί επικύρωση στοιχείων για να αποφευχθεί το γεγονός ο χρήστης να αφήσει κενό κάποιο πεδίο είτε να τοποθετήσει μη έγκυρη τιμή σε κάποιο πεδίο.



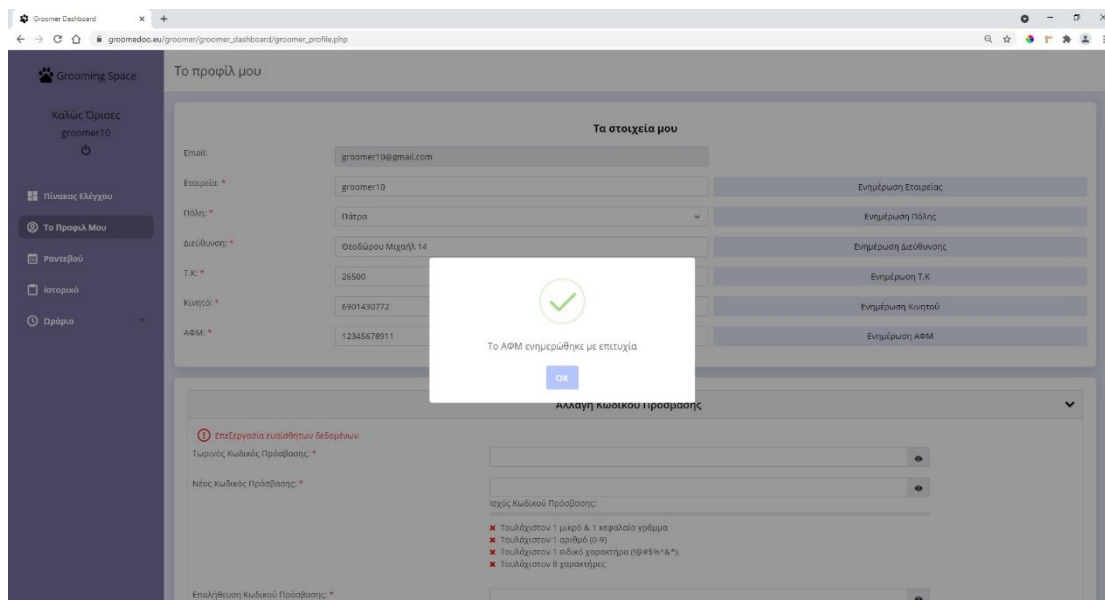
Εικόνα 3.2.24: Προφίλ groomer

Αν ο groomer τοποθετήσει μη έγκυρη τιμή θα του εμφανίσει μήνυμα λάθους, όπως παρουσιάζεται στην Εικόνα 3.2.25.



Εικόνα 3.2.25: Ειδοποίηση για μη έγκυρη τιμή

Όταν ο groomer τροποποιήσει κάποιο πεδίο (για παράδειγμα το ΑΦΜ) και πατήσει «Ενημέρωση ΑΦΜ» θα του εμφανίσει ένα αναδυόμενο παράθυρο που θα τον ενημερώνει πως το συγκεκριμένο πεδίο ενημερώθηκε με επιτυχία, όπως παρουσιάζεται παρακάτω.

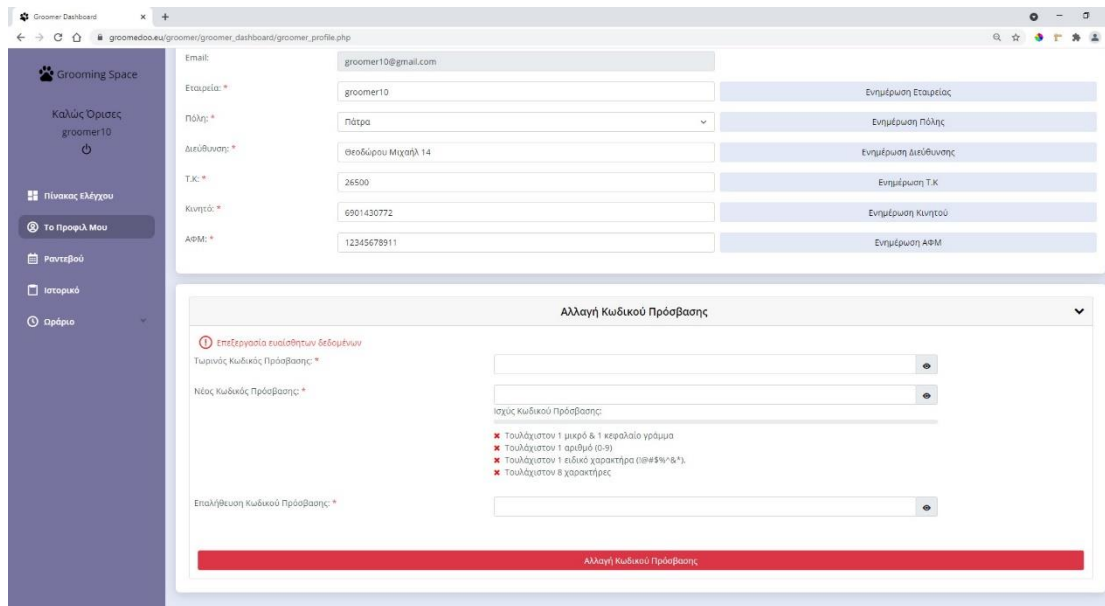


Εικόνα 3.2.26 Επιτυχημένη ενημέρωση πεδίου

Επίσης, ο groomer μέσα από την σελίδα του προφίλ μπορεί να αλλάξει τον κωδικό του πρόσβασης. Στην φόρμα αλλαγής κωδικού πρόσβασης έχει σχεδιαστεί ένα



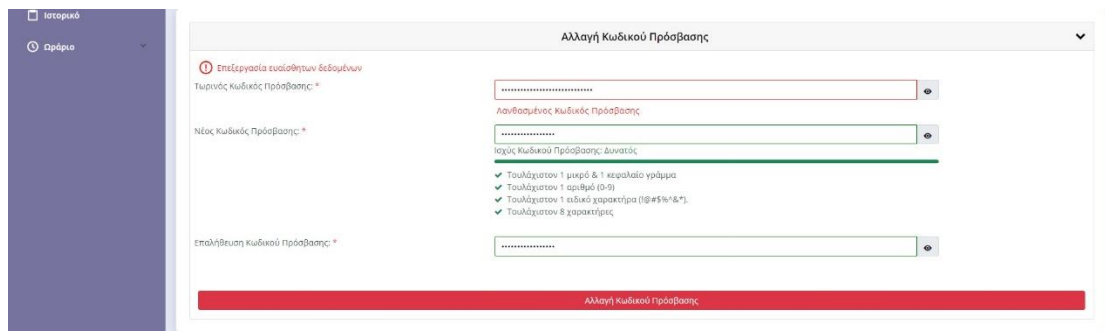
real-time password strength checker, το οποίο βοηθάει τον groomer να δημιουργήσει έναν ισχυρό κωδικό πρόσβασης.



The screenshot shows the 'Groomer Dashboard' profile page. The top section contains personal information: Email (groomer10@gmail.com), Εταιρεία (groomer10), Πόλη (Πάτρα), Διεύθυνση (Θεοδωρου Μιχαήλ 14), Τ.Κ. (26500), Κινητό (6901430772), and ΑΦΜ (12345678911). Below this is the 'Αλλαγή Κωδικού Πρόσβασης' (Change Password) section. It features a warning icon and the text 'Επεξεργασία ευαίσθητων δεδομένων'. The form includes fields for 'Τωρινός Κωδικός Πρόσβασης', 'Νέος Κωδικός Πρόσβασης', and 'Επαλήθευση Κωδικού Πρόσβασης'. A 'Ισχύς Κωδικού Πρόσβασης' section shows a strength indicator and a list of requirements: 'Τουλάχιστον 1 μικρό & 1 κεφαλαίο γράμμα', 'Τουλάχιστον 1 αριθμός (0-9)', 'Τουλάχιστον 1 ειδικό χαρακτήρα (@#%\*~&\*)', and 'Τουλάχιστον 6 χαρακτήρες'. A red button at the bottom is labeled 'Αλλαγή Κωδικού Πρόσβασης'.

Εικόνα 3.2.27: Αλλαγή κωδικού πρόσβασης

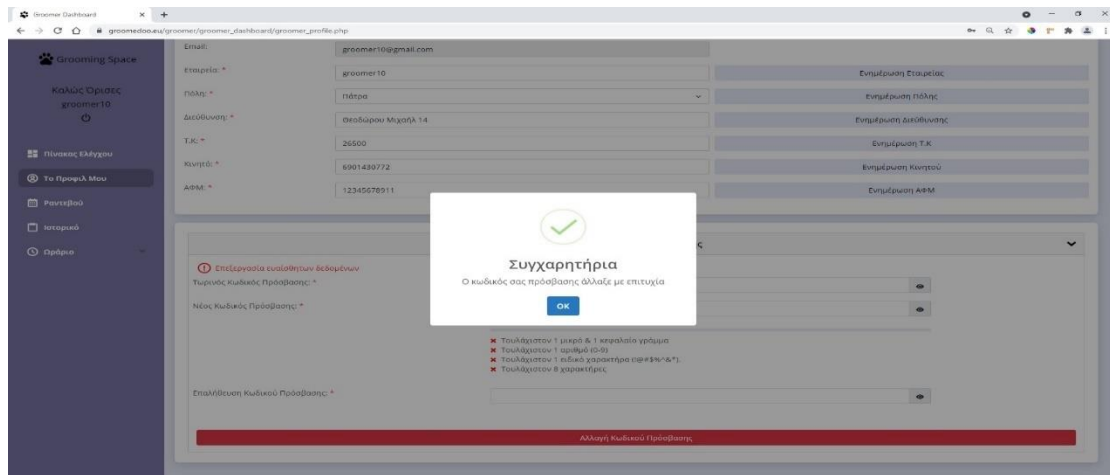
Στην παραπάνω φόρμα, έχει εφαρμοστεί επικύρωση στοιχείων σε περίπτωση που ο groomer τοποθετήσει μη έγκυρο τωρινό κωδικό πρόσβασης ή ο νέος κωδικός πρόσβασης είναι αδύναμος ή δεν γίνεται σωστή επαλήθευση κωδικού πρόσβασης. Ο μη έγκυρος τωρινός κωδικός πρόσβασης παρουσιάζεται στην Εικόνα 3.2.28.



This screenshot shows the same password change form as in Figure 3.2.27, but with an error message. The 'Τωρινός Κωδικός Πρόσβασης' field is highlighted in red, and a red error message 'Λανθασμένος κωδικός πρόσβασης' is displayed below it. The 'Νέος Κωδικός Πρόσβασης' field is highlighted in green, and the 'Επαλήθευση Κωδικού Πρόσβασης' field is also highlighted in green. The strength indicator and requirements list are now green, indicating a strong password. The red button at the bottom is still labeled 'Αλλαγή Κωδικού Πρόσβασης'.

Εικόνα 3.2.28: Ειδοποίηση για μη έγκυρο κωδικό πρόσβασης

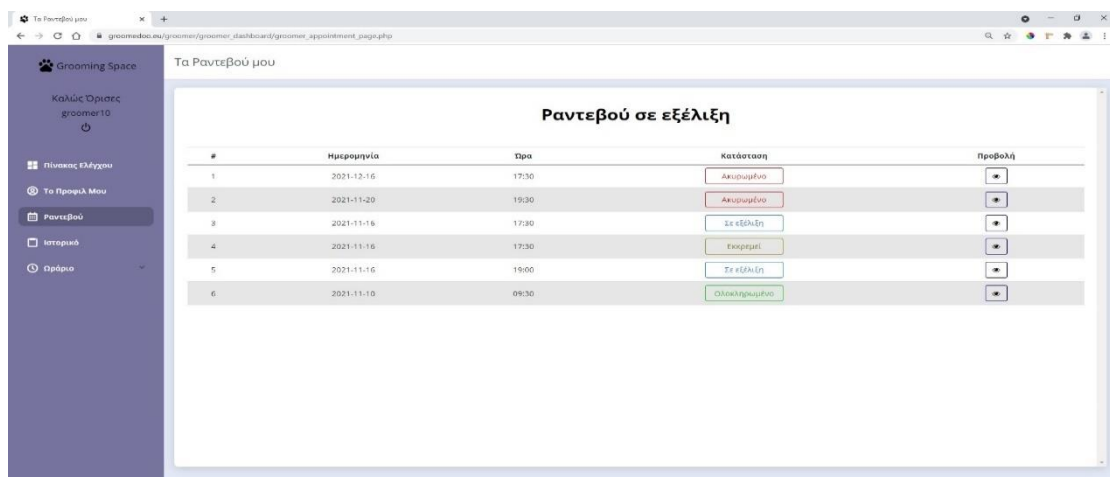
Όταν ο groomer εισάγει σωστά τα πεδία στη φόρμα και πατήσει «Αλλαγή Κωδικού Πρόσβασης» θα του εμφανίσει ένα αναδυόμενο παράθυρο που θα τον ενημερώνει ότι ο κωδικός άλλαξε με επιτυχία, όπως φαίνεται στην Εικόνα 3.2.29.



Εικόνα 3.2.29: Επιτυχημένη αλλαγή κωδικού πρόσβασης

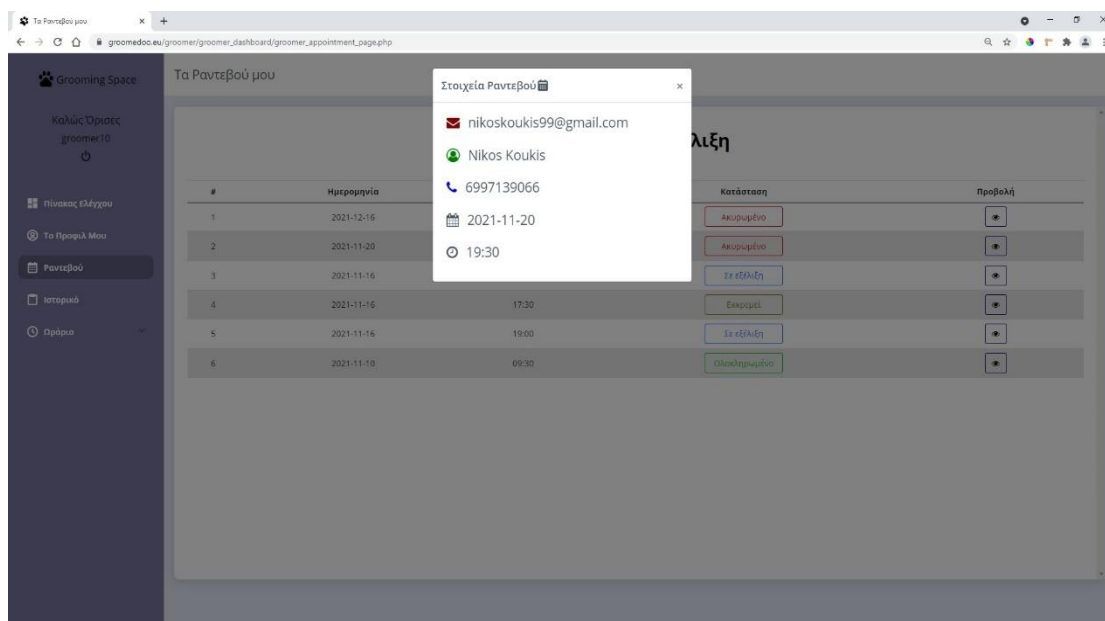
### Σελίδα Ραντεβού Διαχειριστή - Groomer

Ο groomer για να δει την σελίδα με τα ραντεβού αρκεί να πατήσει τον σύνδεσμο «Ραντεβού» που βρίσκεται στην αριστερή πλαϊνή μπάρα. Η συγκεκριμένη σελίδα παρουσιάζει τα ραντεβού εκείνα που η ημερομηνία διεξαγωγής δεν έχει περάσει. Η λίστα με τα ραντεβού του groomer παρουσιάζεται στην Εικόνα 3.2.30. Η λίστα μας ενημερώνει για την ημερομηνία του κάθε ραντεβού, για την ώρα και την κατάσταση στην οποία βρίσκονται τα ραντεβού. Όσα ραντεβού έχουν κατάσταση «Εκκρεμεί» σημαίνει ότι το ραντεβού έχει υποβληθεί με επιτυχία και περιμένει έγκριση από τον διαχειριστή – groomer. Όσα έχουν κατάσταση «Σε εξέλιξη» σημαίνει ότι το ραντεβού έχει υποβληθεί με επιτυχία και έχει εγκριθεί και από τον groomer. Όταν η κατάσταση είναι «Ακυρωμένο» σημαίνει ότι το ραντεβού έχει υποβληθεί με επιτυχία αλλά ο διαχειριστής – groomer απέρριψε την αίτηση. Τέλος, όταν η κατάσταση είναι «Ολοκληρωμένο» σημαίνει ότι το ραντεβού έχει πραγματοποιηθεί με επιτυχία.



Εικόνα 3.2.30: Κατάσταση ραντεβού

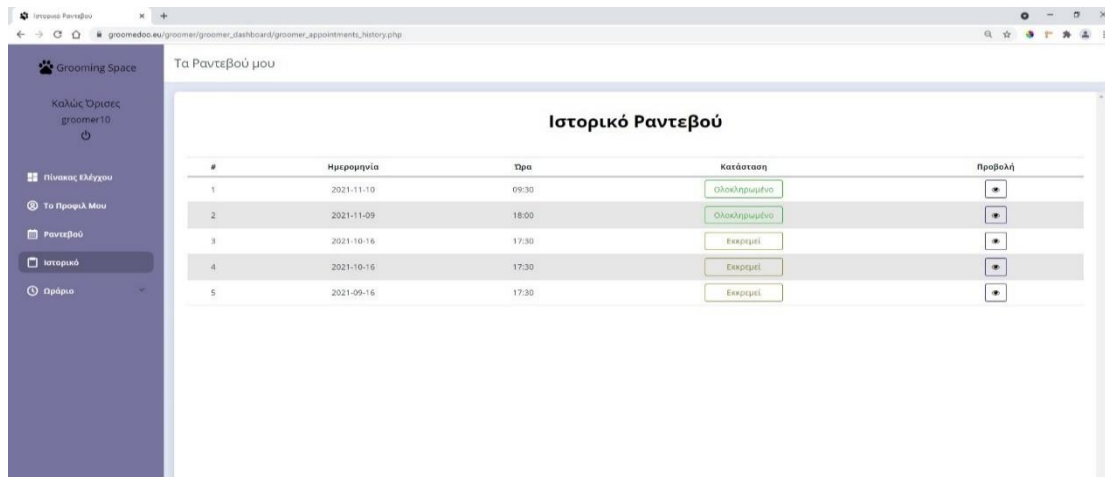
Επίσης, στη λίστα με τα ραντεβού υπάρχει η δυνατότητα «Προβολή». Πατώντας το θα μας εμφανίσει ένα αναδυόμενο παράθυρο με όλες τις πληροφορίες του πελάτη που έκανε αίτηση για ραντεβού. Οι πληροφορίες αυτές αφορούν το email του πελάτη, το ονοματεπώνυμό του, το τηλέφωνο επικοινωνίας του και την ημερομηνία με την ώρα του ραντεβού. Ένα τέτοιο παράδειγμα, παρουσιάζεται στην Εικόνα 3.2.31.



Εικόνα 3.2.31: Στοιχεία πελατών

### Σελίδα Ιστορικό Διαχειριστή - Groomer

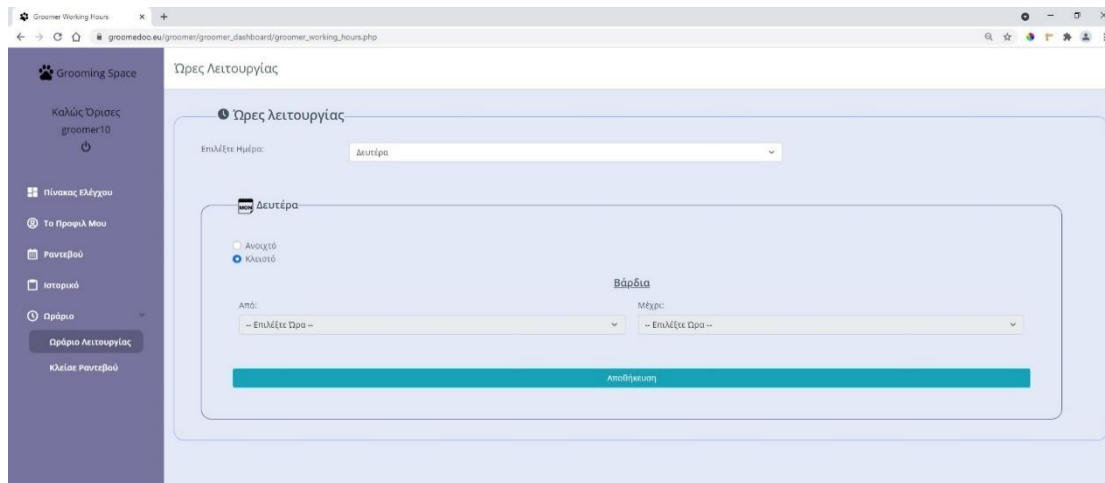
Ο groomer για να δει την σελίδα με το ιστορικό αρκεί να πατήσει τον σύνδεσμο «Ιστορικό» που βρίσκεται στην αριστερή πλαϊνή μπάρα. Το ιστορικό απεικονίζει την ίδια ακριβώς λίστα που περιγράψαμε και παραπάνω (βλ. «Σελίδα Ραντεβού Διαχειριστή – Groomer») με την μόνη διαφορά ότι εδώ υπάρχουν όλα τα ραντεβού ακόμα και αυτά που η ημερομηνία διεξαγωγής έχει περάσει. Και σε αυτή την λίστα, ο groomer έχει την δυνατότητα να δει τις πληροφορίες των πελατών. Η σελίδα με το ιστορικό των ραντεβού παρουσιάζεται στην Εικόνα 3.2.32.



Εικόνα 3.2.32: Ιστορικό ραντεβού

### Ωράριο Λειτουργίας Διαχειριστή – Groomer

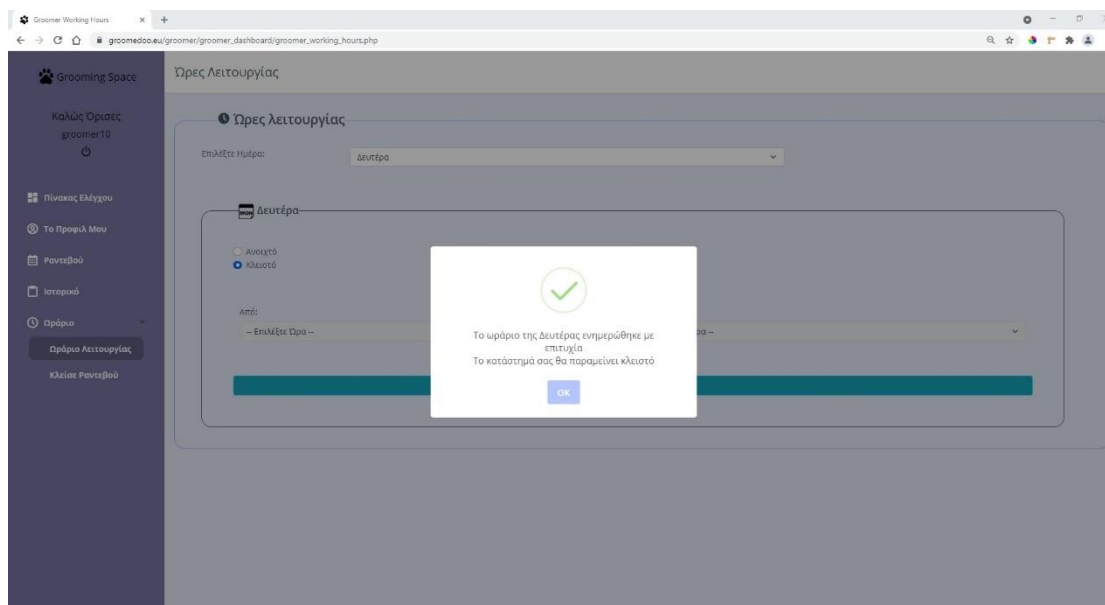
Το σημαντικότερο χαρακτηριστικό της διαδικτυακής εφαρμογής από την πλευρά του διαχειριστή είναι να ορίσει το ωράριο λειτουργίας του καταστήματός του για κάθε ημέρα. Αν δε πραγματοποιήσει αυτή την ενέργεια, τότε το κατάστημά του δεν θα εμφανίζεται στα διαθέσιμα καταστήματα που βλέπει ο πελάτης. Για να ορίσει το ωράριο λειτουργίας, αρκεί να πατήσει τον σύνδεσμο «Ωράριο» στην αριστερή πλαϊνή μπάρα και θα ανοίξει ένα πτυσσόμενο μενού. Στη συνέχεια θα πατήσει «Ωράριο Λειτουργίας». Η σελίδα με το Ωράριο Λειτουργίας απεικονίζεται στην Εικόνα 3.2.33.



Εικόνα 3.2.33: Ωράριο καταστήματος

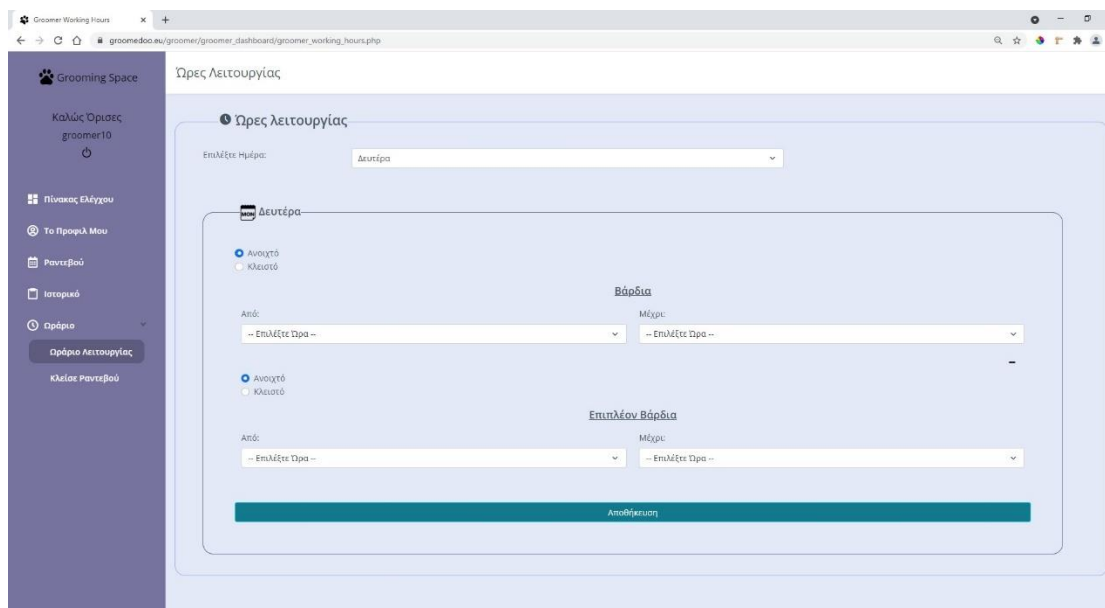
Εδώ ο χρήστης μπορεί να επιλέξει την ημέρα που επιθυμεί για να ορίσει το ωράριο λειτουργίας. Στη συνέχεια, έχοντας δύο επιλογές, μπορεί να ορίσει το κατάστημα είτε ως ανοιχτό είτε ως κλειστό και πατώντας αποθήκευση, θα εμφανιστεί ένα αναδυόμενο παράθυρο που θα τον ενημερώνει για την ενέργεια που μόλις

πραγματοποίησε. Η μόνη διαφορά είναι ότι στην επιλογή «Κλειστό», η επιλογή βάρδια παραμένει απενεργοποιημένη. Ένα τέτοιο παράδειγμα εμφανίζεται στην Εικόνα 3.2.34.



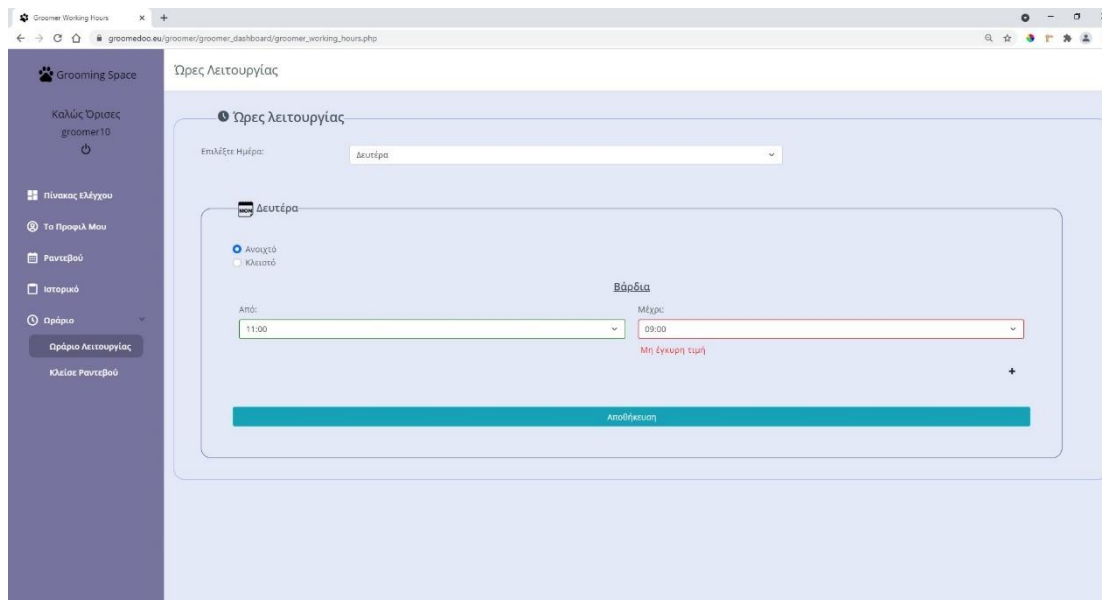
Εικόνα 3.2.34: Ρύθμιση ωραρίου λειτουργίας

Στην περίπτωση που ο groomer επιθυμεί το κατάστημά του να είναι ανοιχτό, τότε επιλέγοντας «Ανοιχτό» θα του ενεργοποιηθεί η δυνατότητα να ορίσει την βάρδια. Μπορεί το ωράριο λειτουργίας του να είναι σπαστό, οπότε να πρέπει να ορίσει δύο βάρδιες. Για να πραγματοποιηθεί αυτή η δυνατότητα, αρκεί ο groomer να πατήσει το σύμβολο «+», όπως απεικονίζεται στην Εικόνα 3.2.35.



Εικόνα 3.2.35 : Ρύθμιση ωραρίου λειτουργίας (συνέχεια)

Στις παραπάνω φόρμες, έχει εφαρμοστεί επικύρωση στοιχείων, καθώς ελέγχει αν οι ώρες που βάζει ο groomer είναι έγκυρες. Μία μη έγκυρη εισαγωγή δεδομένων είναι όταν για παράδειγμα ο groomer ορίσει το ωράριο λειτουργίας από 11:00 π.μ. έως 09:00 π.μ. Ένα τέτοιο παράδειγμα απεικονίζεται στην Εικόνα 3.2.36.

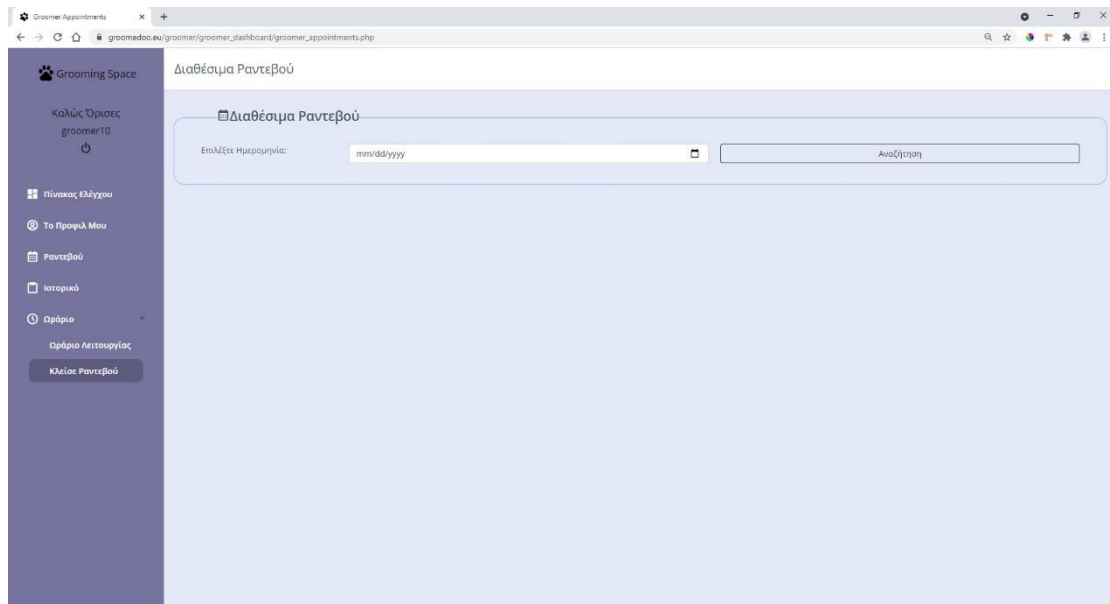


Εικόνα 3.2.36: Εγκυρότητα δεδομένων

Τέλος, όταν ο ιδιοκτήτης ορίσει τις ώρες λειτουργίας αρκεί να πατήσει το κουμπί «Αποθήκευση» και θα ενημερωθεί με ένα αναδυόμενο παράθυρο, όπως παρουσιάστηκε παραπάνω.

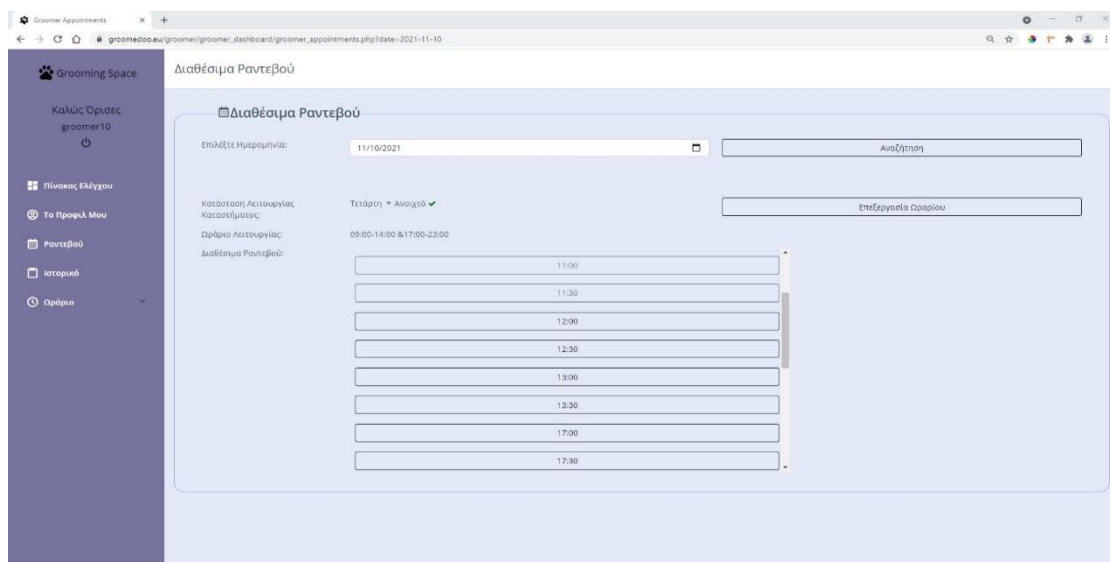
### *Χειροκίνητη Κράτηση Ραντεβού του Διαχειριστή – Groomer*

Ο groomer μέσα από την διαδικτυακή πλατφόρμα έχει την δυνατότητα να κλείσει ραντεβού για κάποιο πελάτη, που πιθανόν να ενημερώθηκε τηλεφωνικά και ο groomer θέλει να τον περάσει στο σύστημά του. Για να πραγματοποιήσει μία τέτοια ενέργεια, αρκεί να πατήσει τον σύνδεσμο «Ωράριο» στην αριστερή πλαϊνή μπάρα και θα ανοίξει ένα πτυσσόμενο μενού. Στη συνέχεια θα πατήσει «Κλείσε Ραντεβού». Η σελίδα μετά από χειροκίνητη κράτηση ραντεβού παρουσιάζεται στην Εικόνα 3.2.37.



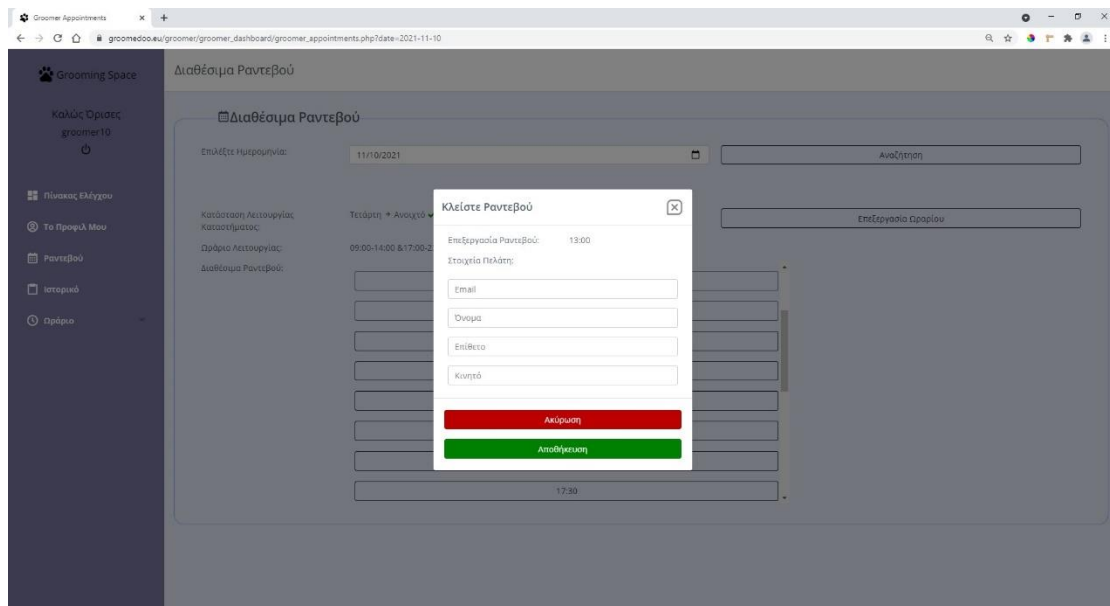
Εικόνα 3.2.37: Χειροκίνητη κράτηση ραντεβού

Αρχικά ο groomer θα πρέπει να επιλέξει την ημερομηνία που επιθυμεί να κάνει την κράτηση. Αφού την επιλέξει θα του εμφανίσει μερικές πληροφορίες που αφορά το κατάστημά του, όπως την κατάσταση λειτουργίας του, δηλαδή αν είναι ανοιχτό ή κλειστό και το ωράριο λειτουργίας του. Επίσης, εμφανίζεται και μία λίστα με τα διαθέσιμα ραντεβού. Όσα ραντεβού είναι απενεργοποιημένα σημαίνει ότι είτε έχει γίνει κάποια άλλη κράτηση εκείνη την ώρα είτε η συγκεκριμένη ώρα έχει περάσει με βάση την τωρινή ώρα. Ένα τέτοιο παράδειγμα απεικονίζεται στην Εικόνα 3.2.38.



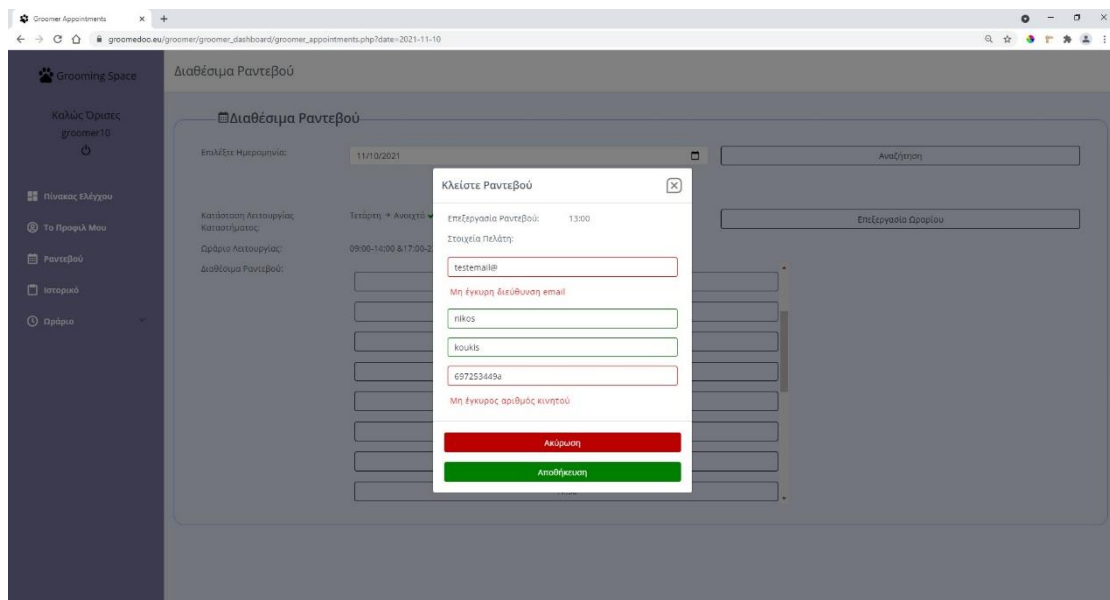
Εικόνα 3.2.38: Επιλογή ώρας χειροκίνητου ραντεβού

Αφού ο groomer επιλέξει μία από τις διαθέσιμες ώρες, θα του εμφανιστεί ένα αναδυόμενο παράθυρο ώστε να εισάγει τα στοιχεία του πελάτη, όπως παρουσιάζεται στην Εικόνα 3.2.39.



Εικόνα 3.2.39: Εισαγωγή στοιχείων πελάτη χειροκίνητου ραντεβού

Στην παραπάνω φόρμα έχει εφαρμοστεί επικύρωση στοιχείων, καθώς ελέγχει αν η φόρμα είναι πλήρως συμπληρωμένη ή κάποιο πεδίο της είναι μη έγκυρο. Μία μη έγκυρα συμπληρωμένη φόρμα απεικονίζεται στην Εικόνα 3.2.40.

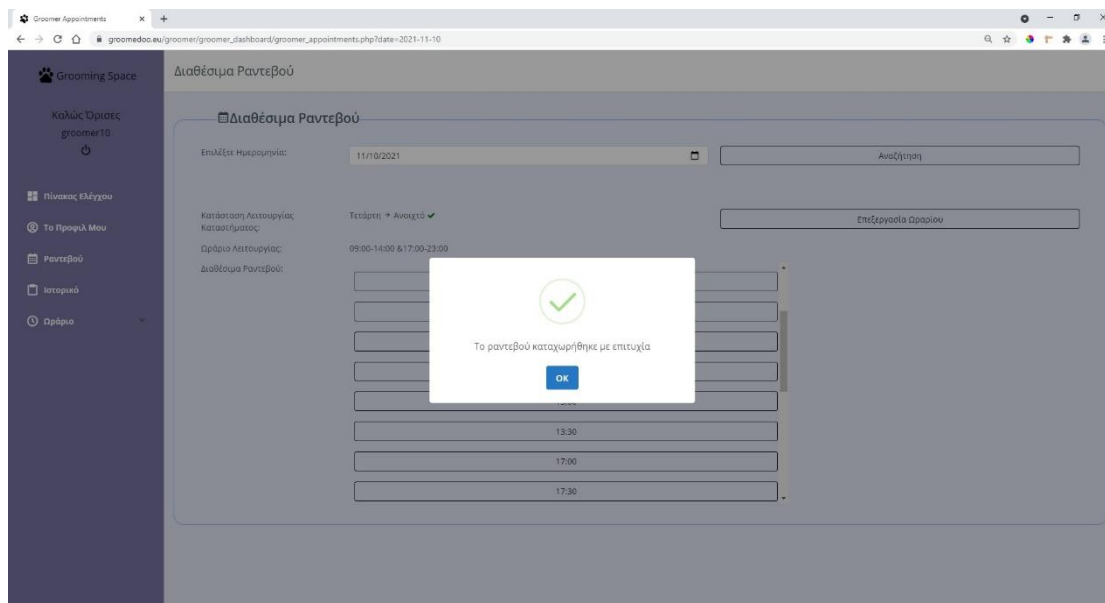


Εικόνα 3.2.40: Μη έγκυρη φόρμα στοιχείων πελάτη

Αφού η φόρμα υποβληθεί με επιτυχία, θα εμφανιστεί στον groomer ένα αναδυόμενο παράθυρο ότι το ραντεβού καταχωρήθηκε με επιτυχία, όπως

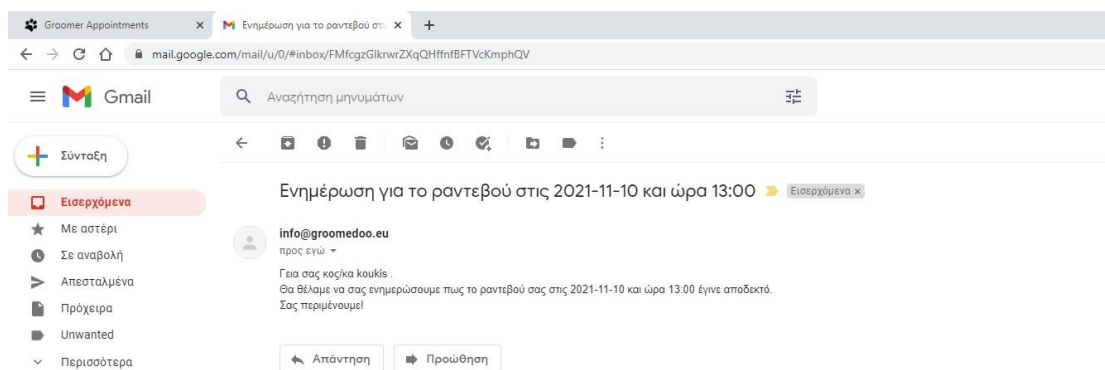


παρουσιάζεται στην Εικόνα 3.2.41. Επίσης η ώρα που πραγματοποιήθηκε η κράτηση για το ραντεβού θα γίνει μη διαθέσιμη για τους υπόλοιπους πελάτες.



Εικόνα 3.2.41: Επιτυχημένη καταχώρηση χειροκίνητου ραντεβού

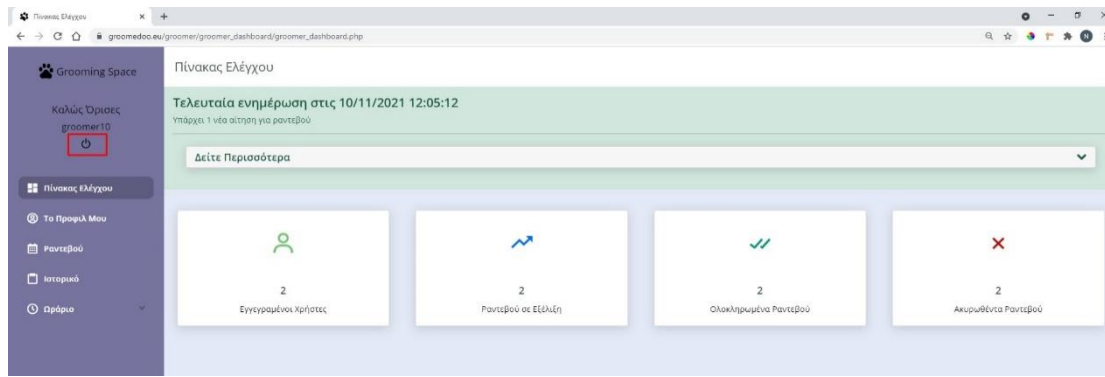
Τέλος, θα σταλθεί ένα ενημερωτικό email στο πελάτη για το ραντεβού του. Ένα τέτοιο email παρουσιάζεται στο παρακάτω στιγμιότυπο.



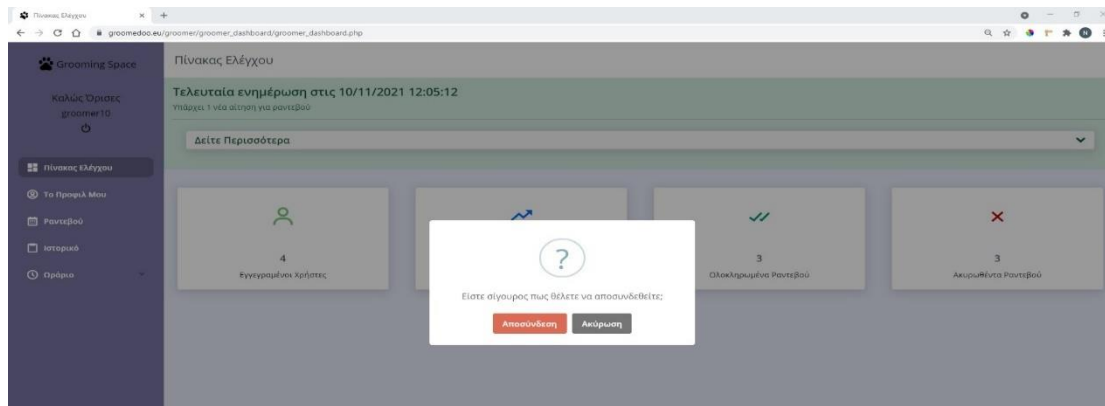
Εικόνα 3.2.42: Email επιβεβαίωσης ραντεβού

### Αποσύνδεση Groomer από την εφαρμογή

Για να αποσυνδεθεί ένας groomer από την εφαρμογή, αρκεί να πατήσει το εικονίδιο της αποσύνδεσης από την πλαϊνή μπάρα και στην συνέχεια να πατήσει «Αποσύνδεση» από το αναδυόμενο παράθυρο, όπως παρουσιάζεται στην Εικόνα 3.2.43 και 3.2.44.



Εικόνα 3.2.43: Αποσύνδεση groomer



Εικόνα 3.2.44: Αποσύνδεση groomer (συνέχεια)

## ΕΠΙΛΟΓΟΣ/ ΣΥΜΠΕΡΑΣΜΑ

Η παρούσα εργασία στόχευε στην ανάλυση των εργαλείων κατασκευής αλλά και στην παρουσίαση της εφαρμογής που κατασκεύασα, η οποία αφορά την ανάπτυξη Web-App με Server/Client Side για ολική διαχείριση, έξυπνη αναζήτηση και Online κράτηση σε Pet Grooming Services. Με την ολοκλήρωση της εργασίας αυτής συμπεραίνουμε ότι η συγκεκριμένη εφαρμογή καλύπτει τις βασικές ανάγκες μιας ολοκληρωμένης διαδικτυακής εφαρμογής, ως προς την ολοκληρωμένη εγγραφή/είσοδο χρηστών σε διαδικτυακή εφαρμογή, ως προς την online κράτηση ραντεβού και ως προς την διατήρηση λίστας ραντεβού και πελατών. Οι γλώσσες προγραμματισμού και οι τεχνολογίες που χρησιμοποιήθηκαν όπως η HTML(5), η CSS(3), η JavaScript, η PHP, η MySQL καθώς και οι βιβλιοθήκες Bootstrap και jQuery, κάλυψαν πλήρως τις ανάγκες που προέκυψαν κατά την ανάπτυξη των λειτουργιών της εφαρμογής, καθώς και την επίτευξη μίας φιλικής προς τον χρήστη παρουσίασης.

Στο πρώτο σκέλος της εργασίας αναφέρθηκαν και αναλύθηκαν τα εργαλεία που χρησιμοποιήθηκαν για την δημιουργία της εφαρμογής. Στο δεύτερο μέρος ασχοληθήκαμε με την δημιουργία του Web-App και την βάση δεδομένων του. Στο τρίτο σκέλος της πτυχιακής μου εργασίας πραγματοποιήθηκε η παρουσίαση της εφαρμογής στην ολοκληρωμένη της μορφή. Στην συνέχεια, στο πεδίο του παραρτήματος, θα δοθεί συμπληρωματικό υλικό ως προς τον τρόπο δημιουργίας της εφαρμογής, καθώς θα παραθέσω κάποια βασικά σημεία του κώδικα.

Η συγκεκριμένη εφαρμογή κατασκευάστηκε, αναλύθηκε και παρουσιάστηκε στα πλαίσια της εκπόνησης της ακαδημαϊκής διπλωματικής μου εργασίας, σε ορισμένο χρονικό διάστημα και με πρακτική εφαρμογή, που την καθιστά επεκτάσιμη και εξελίσιμη. Να σημειωθεί πως υπό κανονικές συνθήκες χρήσης της εφαρμογής είναι πιθανόν να εμφανιστούν σφάλματα, τα οποία δεν έχουν παρατηρηθεί μέχρι στιγμής. Τέλος, στην παρούσα εφαρμογή είναι δυνατή η προσθήκη επιπλέον δυνατοτήτων, με βασικό στόχο την ευκολότερη πλοήγηση και την καλύτερη εξυπηρέτηση των χρηστών της.

## ΠΑΡΑΡΤΗΜΑ 1

### Μέρη του κώδικα για τα κοινά σημεία μεταξύ των αρχείων του πελάτη και του διαχειριστή

#### Σύνδεση εφαρμογής με βάση δεδομένων

Στα περισσότερα αρχεία, όπου απαιτείται σύνδεση με τη βάση δεδομένων, συμπεριλαμβάνεται το παρακάτω τμήμα κώδικα, Sourcecode 1.1.

```
1 <?php
2
3 require(__DIR__ . '/../../vendor/autoload.php');
4
5 $dotenv = \Dotenv\Dotenv::createImmutable(__DIR__ . '/../../');
6 $dotenv->load();
7
8 $localhost = $_ENV["LOCALHOST"];
9 $username = $_ENV["NAME"];
10 $password = $_ENV["PASSWORD"];
11 $database = $_ENV["DATABASE"];
12
13 // Create connection
14 $db = new mysqli($localhost, $username, $password, $database);
15 mysqli_set_charset($db, "utf8");
16
17 if ($db->connect_error) {
18     die("Connection failed: " . $db->connect_error);
19 }
20
21 ?>
22
23
```

Sourcecode 1.1

Για να είναι εφικτή η πρόσβαση στη βάση δεδομένων πρέπει πρώτα να γίνει σύνδεση με το σύστημα διαχείρισης βάσεων δεδομένων, στην περίπτωσή μας με την MySQL. Αρχικά θα πρέπει να ορίσουμε την IP διεύθυνση του Server, που για τοπικές συνδέσεις είναι localhost. Στη συνέχεια, πρέπει να οριστεί το όνομα και ο κωδικός πρόσβασης του χρήστη που θα δοθούν στην MySQL για σύνδεση του συγκεκριμένου χρήστη. Στο database ορίζουμε το όνομα της βάσης δεδομένων όπου θα συνδεθεί ο συγκεκριμένος χρήστης. Για την σύνδεση με την βάση χρησιμοποιούμε την εντολή new mysqli και της περνάμε σαν παραμέτρους την IP διεύθυνση του εξυπηρετητή, το όνομα και τον κωδικό πρόσβασης του χρήστη και το όνομα της βάσης δεδομένων. Όλα αυτά τα αποθηκεύουμε σε μια μεταβλητή \$db. Εάν δεν πραγματοποιηθεί σύνδεση εμφανίζεται αντίστοιχο μήνυμα με τα αίτια της αποτυχίας.

## Αποστολή Email

Παρακάτω παρουσιάζεται το Sourcecode 1.2 που χρησιμοποιείται για την αποστολή email.

```
<?php
require_once '../includes/database/config.php';

use PHPMailer\PHPMailer\PHPMailer;

$mail = new PHPMailer(true);
$mail->SMTPDebug = 1;
$mail->SMTPAuth = true;
$mail->Host = $_ENV["EMAIL_HOST"];
$mail->Username = $_ENV["EMAIL_NAME"];
$mail->Password = $_ENV["EMAIL_PASS"];
$mail->SMTPSecure = $_ENV["EMAIL_SECURITY"];
$mail->Port = $_ENV["EMAIL_PORT"];

$mail->addAddress($user_email);
$mail->setFrom($mail->Username);

$mail->isHTML(true);
$mail->CharSet = 'UTF-8';
$mail->Subject = 'Επιβεβαίωση Λογαριασμού';
$url = "<a href='https://groome.do.eu/user/verification/verified_account.php?token=$token'>α σύνδεσμο</a>";
$mail->Body = "Πατήστε τον $url για να επαληθεύσετε τον λογαριασμό σας με τον παρακάτω 6ψήφιο κωδικό.<br><br><b>Κωδικός Επαλήθευσης:</b> $six_digit_random_number";

if($mail->send()) {
    echo "Email sending successfully...";
} else {
    echo "Email sending failed...";
}
?>
```

### Sourcecode 1.2

Η παρούσα εφαρμογή υποστηρίζει αποστολή email με την χρήση της κλάσης PHPMailer. Η κλάση PHPMailer είναι ένα σύνολο από βιβλιοθήκες και χρησιμοποιείται σε πολλές εφαρμογές. Η χρήση της βιβλιοθήκης αυτής, θα σας επιτρέψει την αποστολή emails μέσω του SMTP διακομιστή με την σωστή κωδικοποίηση και πιστοποίηση των χρηστών ώστε να αποφευχθούν σφάλματα κατά την αποστολή των email. Το SMTP είναι ένα πρωτόκολλο αποστολής μηνυμάτων μεταξύ των sever. Χρησιμοποιεί το TCP πρωτόκολλο με αριθμό θύρας 587, η οποία είναι κατάλληλη για αποστολή email ώστε να αποφευχθούν ανεπιθύμητα email (spam). Επίσης, υπάρχει ακεραιότητα δεδομένων καθώς χρησιμοποιείται το πρωτόκολλο ασφαλείας TLS (Transport Layer Security). Η διαδικασία χρήσης της συγκεκριμένης βιβλιοθήκης είναι αρκετά απλή. Αρχικά δημιουργείται το αντικείμενο PHPMailer που περιλαμβάνει και τα μηνύματα για διαχείριση σφαλμάτων με το όρισμα true από την ακόλουθη γραμμή \$mail = new PHPMailer(true); Στη συνέχεια, ακολουθούν μερικοί παράμετροι με τα ορίσματα τους. Τέλος, είναι ιδανικό να χρησιμοποιηθεί μια συνθήκη ελέγχου (if-else) έτσι ώστε σε περίπτωση μη αποστολής του email να εμφανιστούν τα σφάλματα.



```

function check_user_register_email(){

    var user_register_email = $("#user_register_email").val();
    var user_email_register_pattern = new RegExp(/^[+a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/i);

    if(user_register_email == ""){
        $("#user_register_email_error").html("Το πεδίο είναι υποχρεωτικό");
        $("#user_register_email_error").focus();
        $("#user_register_email_error").show();
        document.getElementById("user_register_email").style.borderColor = "red";
        error_user_register_email = true;
    }else if(user_email_register_pattern.test($("#user_register_email").val())){
        $("#user_register_email_error").hide();
        document.getElementById("user_register_email").style.borderColor = "green";
    }else{
        $("#user_register_email_error").html("Μη έγκυρη διεύθυνση email");
        $("#user_register_email_error").show();
        $("#user_register_email").focus();
        document.getElementById("user_register_email").style.borderColor = "red";
        error_user_register_email = true;
    }
}

function check_user_register_name(){
    var user_register_name = $("#user_register_name").val();

    if(user_register_name == ""){
        $("#user_register_name_error").html("Το πεδίο είναι υποχρεωτικό");
        $("#user_register_name_error").show();
        $("#user_register_name").focus();
        document.getElementById("user_register_name").style.borderColor = "red";
        error_user_register_name = true;
    }else{
        $("#user_register_name_error").hide();
        document.getElementById("user_register_name").style.borderColor = "green";
    }
}

function check_user_register_surname(){
    var user_register_surname = $("#user_register_surname").val();

    if(user_register_surname == ""){
        $("#user_register_surname_error").html("Το πεδίο είναι υποχρεωτικό");
        $("#user_register_surname_error").show();
        $("#user_register_surname").focus();
        document.getElementById("user_register_surname").style.borderColor = "red";
        error_user_register_surname = true;
    }else{
        $("#user_register_surname_error").hide();
        document.getElementById("user_register_surname").style.borderColor = "green";
    }
}
}

```

#### Sourcecode 1.5

```

function check_user_register_phone(){

    var user_register_phone = $("#user_register_phone").val();
    var user_phone_pattern = new RegExp(/^[0-9]\d{9}$/);

    if(user_register_phone == ""){
        $("#user_register_phone_error").html("Το πεδίο είναι υποχρεωτικό");
        $("#user_register_phone_error").show();
        $("#user_register_phone").focus();
        document.getElementById("user_register_phone").style.borderColor = "red";
        error_user_register_phone = true;
    }else if(user_phone_pattern.test($("#user_register_phone").val())){
        $("#user_register_phone_error").hide();
        document.getElementById("user_register_phone").style.borderColor = "green";
    }else{
        $("#user_register_phone_error").html("Μη έγκυρος αριθμός κινητού");
        $("#user_register_phone_error").show();
        $("#user_register_phone").focus();
        document.getElementById("user_register_phone").style.borderColor = "red";
        error_user_register_phone = true;
    }
}

function check_user_register_city(){
    var user_register_city = $("#user_register_city").val();

    if(user_register_city == 0){
        $("#user_register_city_error").html("Το πεδίο είναι υποχρεωτικό");
        $("#user_register_city").focus();
        $("#user_register_city_error").show();
        document.getElementById("user_register_city").style.borderColor = "red";
        error_user_register_city = true;
    }else{
        $("#user_register_city_error").hide();
        document.getElementById("user_register_city").style.borderColor = "green";
    }
}
}

```

#### Sourcecode 1.6

```

function check_user_register_password(){
    var user_register_password = $("#user_register_password").val();

    if(user_register_password == ""){
        $("#user_register_password_error").html("Το πεδίο είναι υποχρεωτικό");
        $("#user_register_password_error").show();
        $("#user_register_password").focus();
        document.getElementById("user_register_password").style.borderColor = "red";
        error_user_register_password = true;
    }else{
        $("#user_register_password_error").hide();
        document.getElementById("user_register_password").style.borderColor = "green";
    }
}

function check_user_register_confirm_password(){
    var user_register_password = $("#user_register_password").val();
    var user_register_confirm_password = $("#user_register_confirm_password").val();

    if(user_register_confirm_password == ""){
        $("#user_register_confirm_password_error").html("Το πεδίο είναι υποχρεωτικό");
        $("#user_register_confirm_password_error").show();
        $("#user_register_confirm_password").focus();
        document.getElementById("user_register_confirm_password").style.borderColor = "red";
        error_user_register_confirm_password = true;
    }else if (user_register_password != user_register_confirm_password){
        $("#user_register_confirm_password_error").html("Οι κωδικοί δε τατριάζουν");
        $("#user_register_confirm_password_error").show();
        $("#user_register_confirm_password").focus();
        document.getElementById("user_register_confirm_password").style.borderColor = "red";
        error_user_register_confirm_password = true;
    }else{
        $("#user_register_confirm_password_error").hide();
        document.getElementById("user_register_confirm_password").style.borderColor = "green";
    }
}
}

```

Sourcecode 1.7

Παραπάνω γίνεται έλεγχος στα πεδία της φόρμας. Πιο συγκεκριμένα δημιουργούνται συναρτήσεις για επικύρωση στοιχείων για να χρησιμοποιηθούν όταν ο χρήστης πατήσει το κουμπί της εγγραφής. Στην συνάρτηση που αφορά το email, δημιουργούμε μία κοινή έκφραση (Regular Expression) που υποδηλώνει τι είδους email θεωρείται σωστό (someone@gmail.com). Οποιαδήποτε άλλη μορφή θεωρείται μη έγκυρη. Επίσης εξετάζεται το ενδεχόμενο κάποιο πεδίο της φόρμας να είναι κενό. Με βάση τις ιδιαιτερότητες κάθε πεδίου χρησιμοποιείται και η αντίστοιχη επικύρωση στοιχείων. Για παράδειγμα, ο χρήστης να πληκτρολογήσει κάποιο λεκτικό στο πεδίο εισαγωγής αριθμού τηλεφώνου.

Εφόσον γίνει η επικύρωση στοιχείων θα πρέπει όλα τα δεδομένα να σταλθούν στον εξυπηρετητή (server). Για να γίνει αυτό χρησιμοποιείται μία ασύγχρονη λειτουργία η οποία ονομάζεται Ajax Request. Με Ajax Request, οι Web εφαρμογές μπορούν να ανταλλάζουν δεδομένα από έναν διακομιστή(server) ασύγχρονα (τρέχοντας στο παρασκήνιο), χωρίς να παρεμβαίνουν στην εμφάνιση και τη συμπεριφορά της υπάρχουσας σελίδας. Η διαδικασία αυτή παρουσιάζεται στο Sourcecode 1.8 και Sourcecode 1.9.



```

$('#btn_register_user').click(function () {

    error_user_register_email = false;
    error_user_register_name = false;
    error_user_register_surname = false;
    error_user_register_phone = false;
    error_user_register_city = false;
    error_user_register_password = false;
    error_user_register_confirm_password = false;
    error_password_strength = false;

    check_user_register_email();
    check_user_register_name();
    check_user_register_surname();
    check_user_register_phone();
    check_user_register_city();
    check_user_register_password();
    check_user_register_confirm_password();
    checkStrength();

    var user_register_email = $('#user_register_email').val().trim();
    var user_register_name = $('#user_register_name').val().trim();
    var user_register_surname = $('#user_register_surname').val().trim();
    var user_register_phone = $('#user_register_phone').val().trim();
    var user_register_city = $('#user_register_city option:selected').text();
    var user_register_password = $('#user_register_password').val().trim();

    if(error_user_register_email == 0 && error_user_register_name == 0 && error_user_register_surname == 0 &&
    error_user_register_phone == 0 && error_user_register_city == 0 && error_user_register_password == 0
    && error_user_register_confirm_password == 0 && error_password_strength == 0){

        $.ajax({
            url: '../user/user_server.php',
            type: 'POST',
            data:
            {
                user_register_from_ajax: 1,
                user_register_email:user_register_email,user_register_name:user_register_name,
                user_register_surname:user_register_surname, user_register_phone:user_register_phone,
                user_register_city:user_register_city,user_register_password:user_register_password
            },
            success:function(response){

                if(response.indexOf('error_user_email_exist') >= 0){
                    $("#user_register_email_error").html("Το email χρησιμοποιείται ήδη");
                    $("#user_register_email").focus();
                    $("#user_register_email_error").show();
                    document.getElementById("user_register_email").style.borderColor = "red";
                    error_user_register_email = true;
                }else{

```

Sourcecode 1.8

```

                }else{

                    Swal.fire({
                        icon: 'success',
                        title: 'Συγχαρητήρια',
                        text: "Η εγγραφή σας ολοκληρώθηκε με επιτυχία. \n\n Παρακαλώ επαληθεύστε" +
                            " τον λογαριασμό σας πατώντας τον σύνδεσμο που σας στείλαμε στο email σας!"
                    })

                    .then(function() {
                        $.ajax({
                            url: '../user/user_server.php',
                            type: 'POST',
                            data:
                            {
                                user_register_send_email_from_ajax: 1,
                                user_register_email:user_register_email
                            },
                            success:function(response){

                                }
                            });
                        });
                    });
                }
            });
        event.preventDefault();
        return true;
    }else{
        return false;
    }
});

```

Sourcecode 1.9

Όταν ο χρήστης πατήσει το κουμπί της εγγραφής τότε καλούνται όλες οι συναρτήσεις που είχαμε παρουσιάσει παραπάνω για την επικύρωση των στοιχείων. Όταν στην φόρμα δεν έχει υποβληθεί κανένα λάθος, τότε με Ajax Request στέλνουμε τα δεδομένα στον server για να τα αποθηκεύσει στην βάση. Αν ο server επιστρέψει επιτυχές αποτέλεσμα, τότε θα ακολουθήσουν οι ενέργειες εμφάνισης αναδυόμενου παράθυρου και αποστολής email, που θα παρουσιαστούν παρακάτω.

Παρακάτω παρουσιάζεται το Sourcecode 1.10, όπου ο εξυπηρετητής (server) επεξεργάζεται το Ajax Request του χρήστη για την πραγματοποίηση εγγραφής.

```
if(isset($_POST['user_register_from_ajax'])){  
    $user_email = $db->real_escape_string($_POST['user_register_email']);  
    $user_name = $db->real_escape_string($_POST['user_register_name']);  
    $user_surname = $db->real_escape_string($_POST['user_register_surname']);  
    $user_phone = $db->real_escape_string($_POST['user_register_phone']);  
    $user_city = $db->real_escape_string($_POST['user_register_city']);  
    $password = $db->real_escape_string($_POST['user_register_password']);  
    $user_password = password_hash($password, PASSWORD_BCRYPT);  
  
    $check_user_email = "SELECT * FROM user WHERE email='$user_email'";  
    $check_email_result = mysqli_query($db,$check_user_email);  
  
    if($check_email_result->num_rows > 0){  
        exit("error_user_email_exist");  
    }else{  
        $query = "INSERT INTO user (email, name, surname, phone, city, password)  
        VALUES ('$user_email', '$user_name', '$user_surname', '$user_phone', '$user_city', '$user_password'  
        )";  
        $result = mysqli_query($db,$query);  
    }  
}
```

Sourcecode 1.10

Αρχικά, ελέγχει στην βάση αν υπάρχει κάποια εγγραφή με το ίδιο email. Αν υπάρχει επιστρέφει μήνυμα λάθους, αλλιώς γίνεται εισαγωγή μιας νέας εγγραφής στη βάση δεδομένων. Επίσης χρησιμοποιείται και η κρυπτογράφηση του κωδικού πρόσβασης για λόγους ασφαλείας.

### *Αποστολή Email Επαλήθευσης Λογαριασμού*

Όταν πραγματοποιείται κάποια εγγραφή τότε στέλνεται ένα email στον χρήστη για να επαληθεύσει τον λογαριασμό του, για να μπορεί στη συνέχεια να πραγματοποιήσει είσοδο. Παρακάτω παρουσιάζεται το Sourcecode 1.11 που παρουσιάζει τη διαδικασία επαλήθευσης λογαριασμού.

```

if(isset($_POST['user_register_send_email_from_ajax'])){
    $user_email = $db->real_escape_string($_POST['user_register_email']);

    $token = uniqid(md5(time()));
    $six_digit_random_number = mt_rand(100000, 999999);
    $query = "INSERT INTO email_verification(email,token,code) VALUES ('$user_email','$token','$six_digit_random_number')";
    $result = mysqli_query($db,$query);

    if($result){
        $mail = new PHPMailer(true);
        $mail->SMTPDebug = 1;
        $mail->SMTPAuth = true;
        $mail->Host = $_ENV["EMAIL_HOST"];
        $mail->Username = $_ENV["EMAIL_NAME"];
        $mail->Password = $_ENV["EMAIL_PASS"];
        $mail->SMTPSecure = $_ENV["EMAIL_SECURITY"];
        $mail->Port = $_ENV["EMAIL_PORT"];

        $mail->addAddress($user_email);
        $mail->setFrom($mail->Username);

        $mail->isHTML(true);
        $mail->CharSet = 'UTF-8';
        $mail->Subject = 'Επαλήθευση Λογαριασμού';
        $url = "<a href='\"https://groomedoo.eu/user/verification/verified_account.php?token=$token\">σύνδεσμο</a>";
        $mail->Body = "Πατήστε τον $url για να επαληθεύσετε τον λογαριασμό σας με τον παρακάτω 6ψήφιο κωδικό.<br><br>Κωδικός Επαλήθευσης:<br> $six_digit_random_number";

        if($mail->send() {
            echo "Email sending successfully...";
        } else {
            echo "Email sending failed...";
        }
    }
}
}

```

### Sourcecode 1.11

Όταν ολοκληρωθεί η εγγραφή γίνεται εισαγωγή στον πίνακα email\_verification, το email που χρησιμοποίησε ο χρήστης κατά την εγγραφή, ένα τυχαίο string που δημιουργείται από το \$token = uniqid(md5(time())); και ένας τυχαίος 6ψηφίος κωδικός που δημιουργείται από το \$six\_digit\_random\_number = mt\_rand(100000, 999999);

### Επαλήθευση Λογαριασμού

Παρακάτω παρουσιάζεται το Sourcecode 1.12 για την επαλήθευση του λογαριασμού.

```

if(isset($_POST['verification'])) {
    $verified_email = $db->real_escape_string($_POST['verified_email']);
    $verified_code = $db->real_escape_string($_POST['verified_code']);

    $query = "SELECT * FROM email_verification WHERE email='$verified_email' AND code='$verified_code'";
    $result = mysqli_query($db,$query);

    if($result->num_rows > 0){
        $query2 = "UPDATE user SET verified=1 WHERE email='$verified_email'";
        $result2 = mysqli_query($db,$query2);
        if($result2){
            $del_query = "DELETE FROM email_verification WHERE email='$verified_email'";
            $del_result = mysqli_query($db,$del_query);
            exit("success");
        }else{
            exit("error");
        }
    }
    exit("match");
}
else{
    exit("dont match");
}
}
}

```

### Sourcecode 1.12

Ο χρήστης καθώς πατήσει τον σύνδεσμο που θα του σταλθεί στο email για επαλήθευση λογαριασμού θα μεταφερθεί σε μια φόρμα όπου πρέπει να εισάγει το email που πραγματοποίησε εγγραφή και τον διψήφιο κωδικό. Αν τα στοιχεία είναι ορθά τότε ο χρήστης επαληθεύεται με επιτυχία.

*Σύνδεση (Login) χρηστών και διαχειριστών (user - groomer) στην εφαρμογή*

Τα στοιχεία του κάθε χρήστη (user) ή διαχειριστή (groomer) συμπληρώνονται στα αντίστοιχα πεδία μιας φόρμας, η οποία παρουσιάζεται στο Sourcecode 1.13.

```
<form>
  <div class="row">
    <div class="form-group col-md-12">
      <label for="user_email">Email</label><span class="required"> *</span>
      <input type="email" class="form-control" id="user_email" name="user_email" autocomplete="off" >
      <p class="error_form" id="user_email_error"></p>
    </div>
  </div>
  <div class="row">
    <div class="form-group col-md-12">
      <label for="user_password">Κωδικός Πρόσβασης</label><span class="required"> *</span>
      <div class="input-group">
        <input type="password" class="form-control" id="user_password" name="user_password" autocomplete="off">
        <div class="input-group-prepend">
          <span class="input-group-text"><i onclick="show_hide_password_modal_login()" id="eye" class="bi bi-eye-fill"></i></span>
        </div>
      </div>
      <p class="error_form" id="user_password_error"></p>
    </div>
  </div>
  <div class="form-check">
    <input class="form-check-input" type="checkbox" id="user_remember_me" name="user_remember_me" >
    <label class="form-check-label" for="user_remember_me">
      <b>Να με θυμάσαι</b>
    </label>
  </div>
  <div id="links">
    <p class="forget_password">Ξεχάσατε τον κωδικό σας;</p>
    <a href="user/user_forget_password.php" class="forget_password">Εναλλαφικά</a>
  </div>
  <div>
    <p class="link_not_member">Δεν έχετε λογαριασμό;</p>
    <a href="#" class="link_not_member">Εγγραφείτε</a>
  </div>
  <div class="modal-footer">
    <div class="form-group col-md-12">
      <button type="button" id="btn_user_login" class="btn btn-primary btn-block">Συνδέσου</button>
      <p class="error_form" id="user_invalid_credentials_error"></p>
    </div>
  </div>
</form>
```

Sourcecode 1.13

Στην παραπάνω φόρμα εφαρμόζεται επικύρωση στοιχείων με την χρήση JavaScript, όπως παρουσιάσαμε στο προηγούμενο Sourcecode (1.13).

Παρακάτω παρουσιάζεται το Sourcecode 1.14, όπου ο εξυπηρετητής (server) επεξεργάζεται το Ajax Request του χρήστη για την πραγματοποίηση σύνδεσης.

```

if(isset($_POST['user_login_from_ajax'])){

    $user_email = $db->real_escape_string($_POST['user_email']);
    $user_password = $db->real_escape_string($_POST['user_password']);
    $user_remember_me = isset($_POST['user_remember_me']);

    $query = "SELECT * FROM user WHERE email='$user_email' ";
    $result = mysqli_query($db,$query);

    if($result->num_rows > 0){

        $row = mysqli_fetch_array($result);
        $verified = $row['verified'];

        if(password_verify($user_password, $row['password'])){

            if($verified == false){
                exit("not_verified");
            }else{

                if (!empty($_POST['user_remember_me'])) {
                    setcookie("email",$user_email, time() + (10*365*24*60*60));
                }else{
                    if (isset($_COOKIE["email"])) {
                        setcookie("email","");
                    }
                }

                $_SESSION['userLoggedIn'] = '1';
                $_SESSION['email'] = $user_email;
                $_SESSION['user_remember_me'] = $user_remember_me;
                exit("success");

            }

        }else{
            exit("error");
        }
    }else{
        exit("error_login");
    }
}
}

```

Sourcecode 1.14

Συγκεκριμένα γίνεται μια ερώτηση στην βάση δεδομένων αν τα στοιχεία σύνδεσης αντιστοιχούν σε κάποια εγγραφή του πίνακα της βάσης δεδομένων. Αν δεν υπάρχει, τότε ο εξυπηρετητής επιστρέφει μήνυμα λάθους, αλλιώς πραγματοποιείται η σύνδεση. Κατά τη σύνδεση, χρησιμοποιείται η \$\_SESSION. Η \$\_SESSION είναι μια μεταβλητή που περιέχει κάποιες πληροφορίες και αποθηκεύεται στον server μέχρι να κλείσει ο browser ή η εφαρμογή να κάνει unset τη μεταβλητή αυτή. Επιπλέον, δίνει τη δυνατότητα να χρησιμοποιηθούν σε πολλά αρχεία οι τιμές των μεταβλητών που είναι ορισμένες σε ένα αρχείο.

## Δημιουργία νέου κωδικού πρόσβασης

Επιλέγοντας «Ξεχάσατε τον κωδικό σας; Επαναφορά» από την σελίδα εισόδου, μεταφέρεται σε μία φόρμα συμπλήρωσης των στοιχείων του, η οποία αποστέλλεται με τη μέθοδο Ajax Request στο παρακάτω Sourcecode 1.15

```
if(isset($_POST['user_change_pass_from_ajax'])) {  
  
    $user_email = $db->real_escape_string($_POST['user_email']);  
    $password = $db->real_escape_string($_POST['user_password']);  
    $user_password = password_hash($password, PASSWORD_BCRYPT);  
  
    $query = "UPDATE user SET password='$user_password' WHERE email='$user_email'";  
    $result = mysqli_query($db, $query);  
  
    if ($result) {  
        $del_query = "DELETE FROM reset_password WHERE email='$user_email'";  
        $del_result = mysqli_query($db, $del_query);  
        exit("success");  
    } else {  
        exit("error");  
    }  
  
}
```

Sourcecode 1.15

## Εμφάνιση των πληροφοριών χρήστη.

Παρακάτω παρουσιάζεται το Sourcecode 1.16 που χρησιμοποιείται σε κάθε αρχείο για την εμφάνιση των πληροφοριών του χρήστη.

```
κ?php  
  
require_once 'includes/database/config.php';  
  
session_start();  
  
$email = $_SESSION['email'];  
$query = "SELECT * FROM user WHERE email='$email'";  
$result = mysqli_query($db,$query);  
while($row = mysqli_fetch_array($result)){  
    $user_name = $row['name'];  
    $user_id = $row['id'];  
}  
?>
```

Sourcecode 1.16

## Αποσύνδεση από την εφαρμογή

Για την διαδικασία της αποσύνδεσης από την εφαρμογή χρησιμοποιείται το Sourcecode 1.17 από το αρχείο user\_logout.php ή groomer\_logout.php. Με αυτό το



```

function checkStrength() {
    var strength = 0;
    var user_password = $("#user_register_password").val();
    var user_password_pattern_1 = new RegExp(/[a-z].*[A-Z]([A-Z].*[a-z])?/);
    var user_password_pattern_2 = new RegExp(/[a-zA-Z]{2}/);
    var user_password_pattern_3 = new RegExp(/[0-9]{2}/);
    var user_password_pattern_4 = new RegExp(/([!%,&,@,#,$,^,*,?,,~])/);

    if(user_password_pattern_1.test($("#user_register_password").val())){
        strength += 1;
        $('.low-upper-case').addClass('text-success');
        $('.low-upper-case i').removeClass('fa-times').addClass('fa-check');
        error_password_strength = true;
        document.getElementById("user_register_password").style.borderColor = "red";
    }else {
        $('.low-upper-case').removeClass('text-success');
        $('.low-upper-case i').addClass('fa-times').removeClass('fa-check');
        error_password_strength = true;
        document.getElementById("user_register_password").style.borderColor = "red";
    }

    if(user_password_pattern_2.test($("#user_register_password").val())){
        strength += 1;
        $('.one-number').addClass('text-success');
        $('.one-number i').removeClass('fa-times').addClass('fa-check');
        error_password_strength = true;
        document.getElementById("user_register_password").style.borderColor = "red";
    } else {
        $('.one-number').removeClass('text-success');
        $('.one-number i').addClass('fa-times').removeClass('fa-check');
        error_password_strength = true;
        document.getElementById("user_register_password").style.borderColor = "red";
    }

    if(user_password_pattern_3.test($("#user_register_password").val())){
        strength += 1;
        $('.one-number').addClass('text-success');
        $('.one-number i').removeClass('fa-times').addClass('fa-check');
        error_password_strength = true;
        document.getElementById("user_register_password").style.borderColor = "red";
    } else {
        $('.one-number').removeClass('text-success');
        $('.one-number i').addClass('fa-times').removeClass('fa-check');
        error_password_strength = true;
        document.getElementById("user_register_password").style.borderColor = "red";
    }
}

```

Sourcecode 1.19

```

    if(user_password_pattern_4.test($("#user_register_password").val())){
        strength += 1;
        $('.one-special-char').addClass('text-success');
        $('.one-special-char i').removeClass('fa-times').addClass('fa-check');
        error_password_strength = true;
        document.getElementById("user_register_password").style.borderColor = "red";
    } else {
        $('.one-special-char').removeClass('text-success');
        $('.one-special-char i').addClass('fa-times').removeClass('fa-check');
        error_password_strength = true;
        document.getElementById("user_register_password").style.borderColor = "red";
    }

    if (user_password.length > 7) {
        strength += 1;
        $('.eight-character').addClass('text-success');
        $('.eight-character i').removeClass('fa-times').addClass('fa-check');
        error_password_strength = true;
        document.getElementById("user_register_password").style.borderColor = "red";
    } else {
        $('.eight-character').removeClass('text-success');
        $('.eight-character i').addClass('fa-times').removeClass('fa-check');
        error_password_strength = true;
        document.getElementById("user_register_password").style.borderColor = "red";
    }

    if (strength < 2) {
        $('#result').removeClass()
        $('#password-strength').addClass('progress-bar bg-danger');
        $('#result').addClass('text-danger').text('Πολύ αδύναμος');
        $('#password-strength').css('width', '20%');
    } else if (strength == 4) {
        $('#result').addClass('good');
        $('#password-strength').removeClass('progress-bar bg-danger');
        $('#password-strength').addClass('progress-bar bg-warning');
        $('#result').addClass('text-warning').text('Αδύναμος');
        $('#password-strength').css('width', '50%');
        return 'Weak'
    } else if (strength == 5) {
        $('#result').removeClass()
        $('#result').addClass('strong');
        $('#password-strength').removeClass('progress-bar bg-warning');
        $('#password-strength').addClass('progress-bar bg-success');
        $('#result').addClass('text-success').text('Δυνατός');
        $('#password-strength').css('width', '100%');
        document.getElementById("user_register_password").style.borderColor = "green";
        error_password_strength = false;
        return 'Strong'
    }
}

```

Sourcecode 1.20



## Απενεργοποίηση ωρών σε μορφή button με βάση την τωρινή ώρα

Ακολουθεί το Sourcecode 1.21 , με το οποίο απενεργοποιούμε τα buttons που αναπαριστούν τις ώρες, οι οποίες έχουν περάσει την τωρινή ώρα.

```
/* Σε αυτό το σημείο τσεκάρουμε αν το διαθέσιμο ραντεβού έχει λήξει με βάση την τωρινή ώρα και ημερομηνία
 * Αν για παράδειγμα η ώρα είναι 12:00 τότε όλα τα buttons έως τις 12 θα είναι απενεργοποιημένα ώστε να μη μπορεί
 * ο χρήστης να κλείσει ραντεβού
 * */
var CurrentHours = new Date().getHours();
var CurrentMinutes = new Date().getMinutes();
var CurrentTime = CurrentHours + ":" + CurrentMinutes;

var NewDate = new Date();
var SelectedDate = new Date(year+"-"+month+"-"+day);

if(SelectedDate.getFullYear() == NewDate.getFullYear() && SelectedDate.getMonth() == NewDate.getMonth() && SelectedDate.getDate() == NewDate.getDate()) {

    $(".appointment_button").each(function () {

        var time = $(this).val();

        if (CurrentTime >= time) {
            $(".appointment_button[value='"+ time + "']").prop("disabled", true);
        }

    });
}
/*-----*/
```

Sourcecode 1.21

## Λήξη ραντεβού με βάση την τωρινή ώρα και ημερομηνία

Παρουσιάζεται το Sourcecode 1.22 με το οποίο θέτουμε κάποιο ραντεβού ως «is\_expired» εφόσον περάσει η ημερομηνία και ώρα διεξαγωγής του ραντεβού με βάση την τωρινή ώρα και ημερομηνία.

```
<?php
require_once 'includes/database/config.php';

/*Με αυτό το αρχείο και με την βοήθεια cron jobs απο cpanel θα θέτουμε κάποιο ραντεβού ως is_expired εφόσον
περάσει η ημερομηνία και ώρα διεξαγωγής του ραντεβού.
Άρα με το 1ο query θέτουμε is_expired=1
*/

$query = "SELECT * FROM appointments WHERE MINUTE(appointment_time)<=MINUTE(CURTIME())
AND HOUR(appointment_time)<=HOUR(CURTIME())
AND appointment_date <= CURDATE()";

$result = mysqli_query($db, $query);

foreach ($result as $row) {
    $id = $row['id'];
    $query2 = "UPDATE appointments SET is_expired=1 WHERE id=$id";
    $result2 = mysqli_query($db, $query2);
}

/* Με αυτό το query θέτουμε in_completed=1 όσα appointments γίνουν is_expired=1 και είναι μόνο in_process=1
*/

$query2 = "SELECT * FROM appointments WHERE MINUTE(appointment_time)<=MINUTE(CURTIME())
AND HOUR(appointment_time)<=HOUR(CURTIME())
AND appointment_date <= CURDATE() AND in_process=1
AND in_process=1";

$result2 = mysqli_query($db, $query2);

while($row = mysqli_fetch_array($result2)){

    $id = $row['id'];

    $query = "UPDATE appointments SET completed=1,in_process=0 WHERE id=$id";
    $result = mysqli_query($db, $query);
}

}>
```

Sourcecode 1.22

## Ενημέρωση ωραρίου λειτουργίας καταστήματος

Στα παρακάτω Sourcecode 1.23, 1.24, 1.25 και 1.26 θέτουμε το κατάστημα είτε ως ανοιχτό είτε ως κλειστό, με βάση την τωρινή ώρα.

```
<?php
require_once 'includes/database/config.php';

$query = "SELECT * FROM groomer_working_days,groomer_working_hours WHERE groomer_working_days.id=groomer_working_hours.day_id
AND groomer_working_days.name_engl_complete=DAYNAME(CURDATE());";
$result = mysqli_query($db, $query);
foreach ($result as $row) {
    $id = $row['id'];

    $query = "SELECT * FROM groomer_working_hours WHERE
TIME(CURTIME()) >= TIME(from_hour_extra) AND TIME(CURTIME()) <= TIME(until_hour_extra) AND id=$id";
$result = mysqli_query($db, $query);
while($row = mysqli_fetch_array($result)){
    $id = $row['id'];
    $update_query = "UPDATE groomer_working_hours SET status='open' WHERE id=$id";
    $update_result = mysqli_query($db, $update_query);
}
}
?>
```

### Sourcecode 1.23

```
<?php
require_once 'includes/database/config.php';

$query = "SELECT * FROM groomer_working_days,groomer_working_hours WHERE groomer_working_days.id=groomer_working_hours.day_id
AND groomer_working_days.name_engl_complete=DAYNAME(CURDATE());";
$result = mysqli_query($db, $query);
foreach ($result as $row) {
    $id = $row['id'];

    $query = "SELECT * FROM groomer_working_hours WHERE
TIME(CURTIME()) >= TIME(from_hour) AND TIME(CURTIME()) <= TIME(until_hour) AND id=$id";
$result = mysqli_query($db, $query);
while($row = mysqli_fetch_array($result)){
    $id = $row['id'];
    $update_query = "UPDATE groomer_working_hours SET status='open' WHERE id=$id";
    $update_result = mysqli_query($db, $update_query);
}
}
?>
```

### Sourcecode 1.24

```

<?php
require_once 'includes/database/config.php';

$query = "SELECT * FROM groomer_working_days,groomer_working_hours WHERE groomer_working_days.id=groomer_working_hours.day_id
AND groomer_working_days.name_eng1_complete=DAYNAME(CURDATE())";

$result = mysqli_query($db, $query);
foreach ($result as $row) {
    $id = $row['id'];

    $query = "SELECT * FROM groomer_working_hours WHERE
TIME(CURTIME()) >= TIME(until_hour) AND id=$id";

    $result = mysqli_query($db, $query);
    while($row = mysqli_fetch_array($result)){
        $id = $row['id'];

        $update_query = "UPDATE groomer_working_hours SET status='closed' WHERE id=$id";
        $update_result = mysqli_query($db, $update_query);
    }
}

?>

```

### Sourcecode 1.25

```

<?php
require_once 'includes/database/config.php';

$query = "SELECT * FROM groomer_working_days,groomer_working_hours WHERE groomer_working_days.id=groomer_working_hours.day_id
AND groomer_working_days.name_eng1_complete=DAYNAME(CURDATE())";

$result = mysqli_query($db, $query);
foreach ($result as $row) {
    $id = $row['id'];

    $query = "SELECT * FROM groomer_working_hours WHERE
TIME(CURTIME()) >= TIME(until_hour_extra) AND id=$id";

    $result = mysqli_query($db, $query);
    while($row = mysqli_fetch_array($result)){
        $id = $row['id'];

        $update_query = "UPDATE groomer_working_hours SET status='closed' WHERE id=$id";
        $update_result = mysqli_query($db, $update_query);
    }
}

?>

```

### Sourcecode 1.26

## ΠΑΡΑΡΤΗΜΑ 2

### Μέρη του κώδικα από την πλευρά του user (client).

Συνάρτηση που μας επιτρέπει να τραβήξουμε δεδομένα από το URL

Παρακάτω, παρουσιάζεται το Sourcecode 2.1, με το οποίο δημιουργείται μια συνάρτηση που τραβάει δεδομένα από το URL.

```
/* Function που μας επιτρέπει να τραβήξουμε δεδομένα από το URL*/
var getUrlParameter = function getUrlParameter(sParam) {
    var sPageURL = decodeURIComponent(window.location.search.substring(1)),
        sURLVariables = sPageURL.split('&'),
        sParameterName,
        i;

    for (i = 0; i < sURLVariables.length; i++) {
        sParameterName = sURLVariables[i].split('=');

        if (sParameterName[0] === sParam) {
            return sParameterName[1] === undefined ? true : sParameterName[1];
        }
    }
};

/* Χρησιμοποιώντας τη συνάρτηση getUrlParameter που ορίσαμε στην αρχή τραβάμε τα δεδομένα από το URL*/
var current_date = getUrlParameter('current_date');
var city = getUrlParameter('city');
var year = getUrlParameter('year');
var month = getUrlParameter('month');
var date = getUrlParameter('date');
```

Sourcecode 2.1

### Φόρμα Αναζήτησης Διαθέσιμων Groomer

Στο παρακάτω Sourcecode 2.2, παρουσιάζεται η φόρμα με πεδία την ημερομηνία και την πόλη που συμπληρώνει ο χρήστης ώστε να δει τους διαθέσιμους groomers.

```
<div class="container">
  <div class="container_header">
    <div class="container_title">
      <h1>Pet Grooming Services</h1>
    </div>
    <div class="container_search">
      <div class="input-group input-group-lg">
        <input type="date" class="form-control" id="date">
      </div>
      <p class="error_form" id="date_error"></p>
      <select class="form-select form-select-lg mb-3" id="city">
        <option selected value="0">-- Επιλέξτε Πόλη--</option>
        <option value="1">Πάτρα</option>
        <option value="2">Αθήνα</option>
        <option value="3">Θεσσαλονίκη</option>
      </select>
      <p class="error_form" id="city_error"></p>
    </div>
    <div class="container_search_btn">
      <button type="button" id="btn_search" class="btn btn-primary btn-lg btn-block">Αναζήτηση</button>
    </div>
  </div>
</div>
```

Sourcecode 2.2

Με την χρήση JavaScript, μεταφερόμαστε στη σελίδα των αποτελεσμάτων της αναζήτησης με παραμέτρους τα στοιχεία που συμπλήρωσε ο χρήστης στη φόρμα. Η τεχνική αυτή παρουσιάζεται στο παρακάτω Sourcecode 2.3.

```
$('#btn_search').click(function () {  
  
    error_city = false;  
    error_date = false;  
  
    check_city();  
    check_date();  
  
    if(error_city == 0 && error_date == 0){  
  
        var d = new Date($('#date').val());  
        var current_date = d.toString().substring(0,3);  
        var year = d.getFullYear();  
        var month = ("0" + (d.getMonth() + 1)).slice(-2);  
        var date = ("0" + (d.getDate() )).slice(-2);  
        var city = $('#city option:selected').text();  
  
        window.location = 'search_result.php?current_date=' + current_date + '&city=' + city + '&year=' + year  
            + '&month=' + month + '&date=' + date;  
  
    }  
  
});
```

Sourcecode 2.3

### Σελίδα αποτελεσμάτων αναζήτησης διαθέσιμων Groomer

Στο παρακάτω Sourcecode 2.4 χρησιμοποιείται η μέθοδος GET με την οποία τραβάμε τα δεδομένα από το URL.

```
require_once 'includes/database/config.php';  
  
/* Redirect στην αρχική σελίδα αν στο url δεν υπάρχουν παράμετροι  
π.χ https://groomedoo.eu/search_result.php/ */  
  
if(!isset($_GET['current_date']) || !isset($_GET['city']) || !isset($_GET['year']) || !isset($_GET['month']) || !isset($_GET['date'])) {  
    header("Location: ./");  
    exit;  
}  
  
$date = $db->real_escape_string($_GET['current_date']);  
$city = $db->real_escape_string($_GET['city']);  
$year = $db->real_escape_string($_GET['year']);  
$month = $db->real_escape_string($_GET['month']);  
$day = $db->real_escape_string($_GET['date']);  
  
$query = "SELECT groomer.id,groomer.email,groomer.company,groomer.city,groomer.address,groomer.phone,status  
FROM groomer INNER JOIN groomer_working_hours ON groomer.id=groomer_working_hours.groomer_id  
INNER JOIN groomer_working_days ON groomer_working_hours.day_id=groomer_working_days.id  
WHERE groomer_working_days.name_engl='$date' AND groomer.city='$city' AND groomer_working_hours.is_closed=0";  
  
$result = mysqli_query($db,$query);
```

Sourcecode 2.4

Στην συνέχεια υλοποιούμε μια ερώτηση στην βάση δεδομένων ώστε να μας φέρει όλες τις εγγραφές εκείνες με τους διαθέσιμους groomer ανά πόλη και ημερομηνία.

Ακολουθεί το Sourcecode 2.5, με το οποίο μεταφερόμαστε στην σελίδα για να πραγματοποιηθεί το ραντεβού πατώντας το button «ΚΛΕΙΣΤΕ PANTEBOY».

```

$( ".btn_schedule_appoin" ).each(function() {
    $(this).on("click", function () {
        var current_date = getUrlParameter('current_date');
        var groomer_id = $(this).val();
        var year = $('#year').text();
        var month = $('#month').text();
        var day = $('#day').text();

        window.location = 'appointment.php?groomer_id=' + groomer_id + '&city=' + city + '&year=' + year
            + '&month=' + month + '&date=' + day + '&current_date=' + current_date;
    });
});

```

Sourcecode 2.5

## Σελίδα αίτησης ραντεβού

Στο παρακάτω Sourcecode 2.6 παρουσιάζονται τα διαθέσιμα ραντεβού με την μορφή buttons. Στην περίπτωση που κάποια ώρα για ραντεβού είναι μη διαθέσιμη τότε το button είναι disabled.

```

<div class="second-content">
<div class="appointment_header">
<div id="buttons_available_appointments">
<!-- Populate Buttons For Available Appointments with PHP -->
<?php
if( $groomer_from_hour_extra != 0 && $groomer_until_hour_extra != 0){ ?>
<?php
    $start_time = strtotime($groomer_from_hour);
    $end_time = strtotime($groomer_until_hour);
    $start_extra_time = strtotime($groomer_from_hour_extra);
    $end_extra_time = strtotime($groomer_until_hour_extra);
    for ($i=$start_time; $i<=$end_time-30; $i = $i + 30*60){
        echo '<button type="button" id="btn_appointment" class="btn btn-primary appointment_button" value="'. date('Hi:i',$i) . "' data-toggle="modal" data-target="#modal'. $i . "'>' . date('Hi:i',$i) . '</button>';
    }
    for ($i=$start_extra_time; $i<=$end_extra_time-30; $i = $i + 30*60){
        echo '<button type="button" id="btn_appointment" class="btn btn-primary appointment_button" value="'. date('Hi:i',$i) . "' data-toggle="modal" data-target="#modal'. $i . "'>' . date('Hi:i',$i) . '</button>';
    }
} else{ ?>
<?php
    $start_time = strtotime($groomer_from_hour);
    $end_time = strtotime($groomer_until_hour);
    for ($i=$start_time; $i<=$end_time-30; $i = $i + 30*60){
        echo '<button type="button" id="btn_appointment" class="btn btn-primary appointment_button" value="'. date('Hi:i',$i) . "' data-toggle="modal" data-target="#modal'. $i . "'>' . date('Hi:i',$i) . '</button>';
    }
}
<?php } ?>
</div>
</div>

```

Sourcecode 2.6

## Αναδυόμενο παράθυρο με στοιχεία πελάτη

Στα παρακάτω Sourcecode, 2.7 και 2.8 παρουσιάζεται το αναδυόμενο παράθυρο που εμφανίζεται όταν ο χρήστης πατήσει μια διαθέσιμη ώρα για ραντεβού, με τα στοιχεία του πελάτη

```

<!-- Modal -->
<div class="appointment_modal modal fade" id="modal" tabindex="-1" role="dialog" aria-labelledby="exampleModalCenterTitle" aria-hidden="true">
  <div class="modal-dialog modal-dialog-centered" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLongTitle">Κλείστε Ραντεβού</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <i class="bx bx-message-square-x"></i>
        </button>
      </div>
      <div class="modal-body">
        <div class="container-fluid">
          <p hidden id="user_id"><?php echo $user_id; ?></p>
          <div class="row">
            <div class="col-8 col-sm-6">
              <p class="edit_appointment_text">Όρα:</p>
            </div>
            <div class="col-4 col-sm-6">
              <p class="edit_appointment_time"></p>
            </div>
          </div>
          <div class="row">
            <div class="col-8 col-sm-6">
              <p class="edit_appointment_text">Ημερομηνία:</p>
            </div>
            <div class="col-4 col-sm-6">
              <p class="edit_appointment_date"></p>
            </div>
          </div>
          <div class="row">
            <div class="col-12 col-sm-12">
              <p class="client_info_text">Στοιχεία Πελάτη:</p>
            </div>
          </div>
          <div class="row">
            <div class="col-12 col-sm-12">
              <input type="email" class="form-control" id="client_email" value="<?php echo $user_email; ?>" disabled />
            </div>
          </div>
          <div class="row">
            <div class="col-12 col-sm-12">
              <input type="text" class="form-control" id="client_name" value="<?php echo $user_name; ?>" disabled />
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

Sourcecode 2.7

```

      <div class="row">
        <div class="col-12 col-sm-12">
          <input type="text" class="form-control" id="client_surname" value="<?php echo $user_surname; ?>" disabled />
        </div>
      </div>
      <div class="row">
        <div class="col-12 col-sm-12">
          <input type="text" class="form-control" id="client_phone" value="<?php echo $user_phone; ?>" disabled />
        </div>
      </div>
    </div>
  <div class="modal-footer">
    <div class="form-group col-md-12">
      <button type="button" class="btn btn-primary btn-block" id="btn_book_appointment">Κλείσε Ραντεβού</button>
    </div>
  </div>
</div>
</div>
</div>

```

Sourcecode 2.8

Με τη χρήση JavaScript και με τη μέθοδο Ajax Request, στέλνουμε όλα αυτά τα δεδομένα στον Server. Ακολουθεί το JavaScript Sourcecode 2.9.

```

$('#btn_book_appointment').click(function () {
  var appointment_date = date;
  var appointment_time = $('#edit_appointment_time').text();
  var groomer_id = getUrlParameter('groomer_id');

  var user_id = $('#user_id').text();
  var modal_email = $('#client_email').val();
  var modal_name = $('#client_name').val();
  var modal_surname = $('#client_surname').val();
  var modal_phone = $('#client_phone').val();

  $.ajax({
    url: 'server.php',
    type: 'POST',
    data: {
      entry_appointment: 1,
      groomer_id: groomer_id, user_id: user_id,
      appointment_date: appointment_date, appointment_time: appointment_time,
      modal_email: modal_email, modal_name: modal_name,
      modal_surname: modal_surname, modal_phone: modal_phone
    },
    success: function(response){
      if(response.indexOf('new_customer_added') >= 0){
        $('#appointment_modal').modal('hide');
        Swal.fire({
          icon: 'success',
          html: "Το ραντεβού καταγράφηκε με επιτυχία. \n" +
            "Θα ενημερωθείτε με email όταν γίνει έγκριση από τον διαχειριστή",
          confirmButtonColor: "#5c5cff",
        }).then(function() {
          location.reload();
        });
      }
      if(response.indexOf('to_radevou_uparxei') >= 0){
        $('#appointment_modal').modal('hide');
        Swal.fire("Κάτι πήγε στραβά!", "Το ραντεβού είναι μη διαθέσιμο. Επιλέξτε κάποια άλλη ώρα", "error")
          .then(function() {
            location.reload();
          });
      }
    }
  });
});

```

Sourcecode 2.9

Στο παρακάτω Sourcecode 2.10 και Sourcecode 2.11 απεικονίζεται ο κώδικας που χρησιμοποιείται για την δημιουργία αιτήματος ραντεβού.

```

if(isset($_POST['entry_appointment'])) {
    $groomer_id = $db->real_escape_string($_POST['groomer_id']);
    $user_id = $db->real_escape_string($_POST['user_id']);
    $appointment_date = $db->real_escape_string($_POST['appointment_date']);
    $appointment_time = $db->real_escape_string($_POST['appointment_time']);
    $modal_email = $db->real_escape_string($_POST['modal_email']);
    $modal_name = $db->real_escape_string($_POST['modal_name']);
    $modal_surname = $db->real_escape_string($_POST['modal_surname']);
    $modal_phone = $db->real_escape_string($_POST['modal_phone']);

    $query = "SELECT * FROM appointments WHERE appointment_date = '$appointment_date'
    AND appointment_time = '$appointment_time' AND booked = 1 AND is_expired=0";

    $result = mysqli_query($db, $query);
    if ($result->num_rows > 0) {
        exit('to_radevou_uparxei!');
    }else{
        $re_entry_for_decline_query = "SELECT customer.id,customer.name,customer.user_id,appointments.customer_id,appointments.groomer_id,appointments.appointment_date,appointments.appointment_time
        FROM customer,appointments
        WHERE customer.id=appointments.customer_id AND customer.user_id=$user_id AND appointments.groomer_id = $groomer_id
        AND appointments.appointment_date = '$appointment_date' AND appointments.appointment_time = '$appointment_time'
        AND appointments.canceled = 1 AND is_expired=0";

        $result_re_entry_for_decline = mysqli_query($db, $re_entry_for_decline_query);
        while($row = mysqli_fetch_array($result_re_entry_for_decline)){
            $customer_id = $row['id'];
        }

        if ($result_re_entry_for_decline->num_rows > 0) {
            $update_query = "UPDATE appointments SET appointment_date='$appointment_date',appointment_time = '$appointment_time',booked=1,canceled=0
            WHERE groomer_id=$groomer_id AND appointment_date = '$appointment_date' AND appointment_time = '$appointment_time'
            AND customer_id=$customer_id AND canceled = 1 AND is_expired=0";
            $result_update_query = mysqli_query($db, $update_query);

            if($result_update_query){
                $update_query = "UPDATE customer SET entry_appointment=CURRENT_TIMESTAMP() WHERE user_id=$user_id AND id=$customer_id";
                $result_update_query = mysqli_query($db, $update_query);
            }
            exit("new_customer_added");
        }
    }
}

```

### Sourcecode 2.10

```

}else{
    $query = "INSERT INTO customer (email, name, surname, phone ,appointment_date, appointment_time, is_user, user_id)
    VALUES ('$modal_email', '$modal_name', '$modal_surname', '$modal_phone',
    '$appointment_date', '$appointment_time', 1,$user_id)";

    $result = mysqli_query($db, $query);

    $return_arr = array();

    $query2 = "SELECT * FROM customer WHERE email='$modal_email' AND appointment_date='$appointment_date' AND appointment_time='$appointment_time'";
    $result2 = mysqli_query($db, $query2);

    while ($row = mysqli_fetch_array($result2)) {
        $customer_id = $row['id'];
        $insert_appointment = "INSERT INTO appointments (customer_id, groomer_id,appointment_date, appointment_time, booked)
        VALUES ('$customer_id',$groomer_id,'$appointment_date','$appointment_time',1)";
        $result_ins_appointment = mysqli_query($db, $insert_appointment);

        $update_user_is_customer_query = "UPDATE user set is_customer=1 WHERE id=$user_id";
        $result_update_user_is_customer = mysqli_query($db, $update_user_is_customer_query);

        if ($result_ins_appointment) {
            exit("new_customer_added");
        }
    }
}
}
}

```

### Sourcecode 2.11



## ΠΑΡΑΡΤΗΜΑ 3

### Μέρη του κώδικα από την πλευρά του groomer (server)

#### Groomer Sidebar

Στο παρακάτω Sourcecode 3.1 και Sourcecode 3.2 παρουσιάζεται το πλαίσιο μενού (sidebar) που χρησιμοποιείται σε όλες τις σελίδες του διαχειριστικού.

```
<div class="sidebar">
  <div class="sidebar-logo">
    <a href="https://groomedoo.eu/groomer/groomer_dashboard/groomer_dashboard.php">
      <i class='bx bxl-baidu'></i>
      <p>Grooming Space</p>
    </a>
    <div class="sidebar-close" id="sidebar-close">
      <i class='bx bx-left-arrow-alt'></i>
    </div>
  </div>
  <div class="profile_content">
    <div class="profile">
      <div class="profile_details">
        <div class="name_groomer">
          <div class="name">Καλώς Όρισες</div>
          <div class="groomer" id="groomer"><?echo $groomer_company ?></div>
        </div>
      </div>
    </div>
  </div>
  <button class="btn btn-outline" id="btn_logout">
    <i class='bx bx-power-off'></i>
  </button>
</div>
```

#### Sourcecode 3.1

```
<!-- SIDEBAR MENU -->
<ul class="sidebar-menu" id="sidebar-menu">
  <li>
    <a href="groomer_dashboard.php">
      <i class='bx bxs-dashboard'></i>
      <span>Πίνακας Ελέγχου</span>
    </a>
  </li>
  <li>
    <a href="groomer_profile.php">
      <i class='bx bx-user-circle'></i>
      <span>Το προφίλ μου</span>
    </a>
  </li>
  <li>
    <a href="groomer_appointment_page.php">
      <i class='bx bx-calendar'></i>
      <span>Ραντεβού</span>
    </a>
  </li>
  <li>
    <a href="groomer_appointments_history.php">
      <i class='bx bx-clipboard'></i>
      <span>Ιστορικό</span>
    </a>
  </li>
  <li class="sidebar-submenu">
    <a href="#" class="sidebar-menu-dropdown">
      <i class='bx bx-time-five'></i>
      <span>Οράριο</span>
      <div class="dropdown-icon"></div>
    </a>
    <ul class="sidebar-menu sidebar-menu-dropdown-content" id="dropdown_menu">
      <li>
        <a href="groomer_working_hours.php">
          Οράριο Λειτουργίας
        </a>
      </li>
      <li>
        <a href="groomer_appointments.php">
          Κλείσε Ραντεβού
        </a>
      </li>
    </ul>
  </li>
</ul>
<!-- END SIDEBAR MENU -->
```

#### Sourcecode 3.2

## Ενημερωτικό κομμάτι στο διαχειριστικό του groomer για αιτήσεις ραντεβού

Ακολουθούν τα Sourcecode 3.3, Sourcecode 3.4, Sourcecode 3.5 και Sourcecode 3.6 για την απεικόνιση των αιτήσεων για ραντεβού.

```
<div class="main">
  <div class="main-header" id="main-header">
    <div class="mobile-toggle" id="mobile-toggle">
      <i class="bx bx-menu-alt-right"></i>
    </div>
    <div class="main-title">
      Πίνακας Ελέγχου
    </div>
  </div>

  <?php
  $query_check_appointment = "SELECT * FROM customer,appointments WHERE appointments.groomer_id=$groomer_id AND booked=true AND is_expired=0
                              AND appointments.customer_id = customer.id ORDER BY appointments.appointment_date ASC, appointments.appointment_time ASC";
  $result_check_appointment = mysqli_query($db,$query_check_appointment);

  $query_count_appointment = "SELECT count(*) c FROM appointments WHERE groomer_id=$groomer_id AND booked=true AND is_expired=0";
  $result_count_appointment = mysqli_query($db,$query_count_appointment);
  while($row = mysqli_fetch_array($result_count_appointment)){
    $count = $row['c'];
  }
}
```

### Sourcecode 3.3

```
if ($result_check_appointment->num_rows == 1) {
  >>
  <div class="alert alert-success" role="alert">
    <h4 class="alert-heading"></h4>
    <p>Υπάρχει <?php echo $count; > νέα αιτήση για ραντεβού</p>
    <hr>
    <div id="accordion">
      <div class="card">
        <div class="card-header">
          <a id="check_appointment_dropdown" data-toggle="collapse" href="#check_appointment_collapse" aria-expanded="true" aria-controls="collapse-example" class="d-block">
            <i class="fa fa-chevron-down pull-right"></i>
            Δείτε Ημερολόγιο
          </a>
          <div id="check_appointment_collapse" class="collapse" aria-labelledby="heading-example">
            <?php
            while($row = mysqli_fetch_array($result_check_appointment)){
              $appointment_id = $row['id'];
              $name = $row['name'];
              $surname = $row['surname'];
              $email = $row['email'];
              $phone = $row['phone'];
              $appointment_date = $row['appointment_date'];
              $appointment_time = $row['appointment_time'];
              $entry_appointment = $row['entry_appointment'];

              >>
              <div class="card text-center">
                <div class="card-body">
                  <div class="card-header">
                    <p>Νέα Αίτηση</p>
                  </div>
                  <p class="card-text name">Όνομα: <?php echo $name; ></p>
                  <p class="card-text surname" value="<?php echo $surname; >">Επώνυμο: <?php echo $surname; ></p>
                  <p class="card-text email" value="<?php echo $email; >">Email: <?php echo $email; ></p>
                  <p class="card-text phone" value="<?php echo $phone; >">Τηλέφωνο: <?php echo $phone; ></p>
                  <p class="card-text appointment_date" value="<?php echo $appointment_date; >">Ημερομηνία: <?php echo $appointment_date; ></p>
                  <p class="card-text appointment_time" value="<?php echo $appointment_time; >">Ώρα: <?php echo $appointment_time; ></p>
                  <div class="form-group col-md-12" id="div_btn">
                    <button type="button" value="<?php echo $appointment_id >" id="btn_accept" class="btn btn-primary btn-block accept">ΑΦΟΡΩΧΗ ΑΙΤΗΣΗ</button>
                  </div>
                  <div class="form-group col-md-12" id="div_btn">
                    <button type="button" value="<?php echo $appointment_id >" id="btn_decline" class="btn btn-primary btn-block decline">ΑΠΟΡΡΙΨΗ ΑΙΤΗΣΗΣ</button>
                  </div>
                </div>
                <div class="card-footer">
                  Ημερομηνία Αίτησης: <?php echo $entry_appointment; >
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
}
```

### Sourcecode 3.4



```

if(isset($_POST['accept_appointment'])) {
    $appointment_id = $db->real_escape_string($_POST['appointment_id']);
    $appointment_date = $db->real_escape_string($_POST['appointment_date']);
    $appointment_time = $db->real_escape_string($_POST['appointment_time']);
    $email = $db->real_escape_string($_POST['email']);
    $surname = $db->real_escape_string($_POST['surname']);
    $phone = $db->real_escape_string($_POST['phone']);

    $query = "UPDATE appointments SET booked = false, in_process = true WHERE id=$appointment_id";
    $result = mysqli_query($db, $query);
    if($result){
        $mail = new PHPMailer(true);
        $mail->SMTPDebug = 1;
        $mail->SMTPAuth = true;
        $mail->Host = $_ENV["EMAIL_HOST"];
        $mail->Username = $_ENV["EMAIL_NAME"];
        $mail->Password = $_ENV["EMAIL_PASS"];
        $mail->SMTPSecure = $_ENV["EMAIL_SECURITY"];
        $mail->Port = $_ENV["EMAIL_PORT"];

        $mail->addAddress($email);
        $mail->setFrom($mail->Username);

        $mail->isHTML(true);
        $mail->CharSet = 'UTF-8';
        $mail->Subject = "Επιβεβαίωση για το ραντεβού στις " . $appointment_date . " και ώρα " . $appointment_time;
        $mail->Body = "Γεια σας κο/κα $surname . <br> Θα θέλαμε να σας ενημερώσουμε πως το ραντεβού σας στις $appointment_date και ώρα $appointment_time  
έγινε αποδεκτό.<br>Σας περιμένουμε!";

        if($mail->send() {
            echo "Email sending successfully...";
        } else {
            echo "Email sending failed...";
        }
        exit('appointment_accepted');
    }else{
        exit('error_in_appointment_accepted');
    }
}
}

```

Sourcecode 3.8

## Απόρριψη Αίτησης Ραντεβού

Ακολουθούν τα Sourcecode 3.9 και Sourcecode 3.10 για την απόρριψη της αίτησης για ραντεβού από τον Groomer.

```

$( ".decline" ).each(function() {
    $(this).on("click", function () {
        var appointment_id = $(this).val();
        var appointment_date = $(' .appointment_date').attr('value');
        var appointment_time = $(' .appointment_time').attr('value');
        var email = $(' .email').attr('value');
        var surname = $(' .surname').attr('value');
        var phone = $(' .phone').attr('value');

        Swal.fire({
            icon: 'question',
            text: 'Είστε σίγουρος πως θέλετε να ακυρώσετε το συγκεκριμένο ραντεβού;',
            showCloseButton: true,
            showCancelButton: true,
            confirmButtonText: 'Απόρριψη Αίτησης',
            cancelButtonText: 'Ακύρωση',
        }).then((result) => {
            if (result.isConfirmed) {
                $.ajax({
                    url: '../groomer/groomer_server.php',
                    type: 'POST',
                    data: {
                        decline_appointment: 1,
                        appointment_id:appointment_id,appointment_date:appointment_date,
                        appointment_time:appointment_time,email:email,surname:surname,phone:phone
                    },
                    success:function(response){
                        if(response.indexOf('appointment_declined') >= 0){
                            Swal.fire({
                                icon: 'success',
                                text: 'Το ραντεβού ακυρώθηκε με επιτυχία',
                            }).then(function () {
                                location.reload();
                            });
                        }
                    }
                });
            }
        });
    });
});

```

Sourcecode 3.9

```

if(isset($_POST['decline_appointment'])) {
    $appointment_id = $db->real_escape_string($_POST['appointment_id']);
    $appointment_date = $db->real_escape_string($_POST['appointment_date']);
    $appointment_time = $db->real_escape_string($_POST['appointment_time']);
    $email = $db->real_escape_string($_POST['email']);
    $surname = $db->real_escape_string($_POST['surname']);
    $phone = $db->real_escape_string($_POST['phone']);
    $query = "UPDATE appointments SET booked = false, canceled = true WHERE id=$appointment_id;";
    $result = mysqli_query($db, $query);

    if($result){
        $mail = new PHPMailer(true);
        $mail->SMTPDebug = 1;
        $mail->SMTPAuth = true;
        $mail->Host = $_ENV["EMAIL_HOST"];
        $mail->Username = $_ENV["EMAIL_NAME"];
        $mail->Password = $_ENV["EMAIL_PASS"];
        $mail->SMTPSecure = $_ENV["EMAIL_SECURITY"];
        $mail->Port = $_ENV["EMAIL_PORT"];

        $mail->addAddress($email);
        $mail->setFrom($mail->Username);

        $mail->isHTML(true);
        $mail->CharSet = 'UTF-8';
        $mail->Subject = "Ενημέρωση για το ραντεβού στις " . $appointment_date . " και ώρα " . $appointment_time;
        $mail->Body = "Γεια σας κος/κα $surname . <br> Θα θέλαμε να σας ενημερώσουμε πως το ραντεβού σας στις $appointment_date και ώρα $appointment_time απορρίφθηκε. Παρακαλώ προσπαθήστε ξανά";

        if($mail->send() {
            echo "Email sending successfully...";
        } else {
            echo "Email sending failed...";
        }
        exit('appointment_declined');
    }else{
        exit('error_in_appointment_declined');
    }
}
}

```

Sourcecode 3.10

### Απεικόνιση στατιστικών εγγεγραμμένων χρηστών

Το παρακάτω Sourcecode 3.11 απεικονίζει τον κώδικα για το συνολικό αριθμό εγγεγραμμένων χρηστών στον συγκεκριμένο groomer.

```

$query_count_user = "SELECT COUNT(DISTINCT(user_id)) count_user FROM appointments,customer
WHERE customer.id=appointments.customer_id AND groomer_id=$groomer_id";
$result_count_user = mysqli_query($db,$query_count_user);
while($row = mysqli_fetch_array($result_count_user)){
    $count_user = $row['count_user'];
}

```

Sourcecode 3.11

### Απεικόνιση στατιστικών ραντεβού σε εξέλιξη

Το παρακάτω Sourcecode 3.12 απεικονίζει τον κώδικα για το συνολικό αριθμό των ραντεβού που είναι σε εξέλιξη στον συγκεκριμένο groomer.

```

$query_count_in_process_appoint = "SELECT COUNT(id) count_in_process_appoint FROM appointments
WHERE groomer_id=$groomer_id AND in_process=1";
$result_count_in_process_appoint = mysqli_query($db,$query_count_in_process_appoint);
while($row = mysqli_fetch_array($result_count_in_process_appoint)){
    $count_in_process_appoint = $row['count_in_process_appoint'];
}

```

Sourcecode 3.12

### Απεικόνιση στατιστικών ολοκληρωμένων ραντεβού

Το παρακάτω Sourcecode 3.13 απεικονίζει τον κώδικα για το συνολικό αριθμό των ολοκληρωμένων ραντεβού για τον συγκεκριμένο groomer.

```

$query_count_completed_appoint = "SELECT COUNT(id) count_completed_appoint FROM appointments
                                WHERE groomer_id=$groomer_id AND completed=1";
$result_count_completed_appoint = mysqli_query($db,$query_count_completed_appoint);
while($row = mysqli_fetch_array($result_count_completed_appoint)){
    $count_completed_appoint = $row['count_completed_appoint'];
}
$query_count_canceled_appoint = "SELECT COUNT(id) count_canceled_appoint FROM appointments

```

Sourcecode 3.13

### Απεικόνιση στατιστικών ακυρωθέντα ραντεβού

Το παρακάτω Sourcecode 3.14 απεικονίζει τον κώδικα για το συνολικό αριθμό των ακυρωμένων ραντεβού για τον συγκεκριμένο groomer.

```

$query_count_canceled_appoint = "SELECT COUNT(id) count_canceled_appoint FROM appointments
                                WHERE groomer_id=$groomer_id AND canceled=1";
$result_count_canceled_appoint = mysqli_query($db,$query_count_canceled_appoint);
while($row = mysqli_fetch_array($result_count_canceled_appoint)){
    $count_canceled_appoint = $row['count_canceled_appoint'];
}

```

Sourcecode 3.14

### Πίνακας Προφίλ Groomer

Παρακάτω παρουσιάζεται το Sourcecode 3.15 για την συλλογή δεδομένων του groomer για το προφίλ του.

```

if(isset($_POST['groomer_profile_from_ajax'])){
    $email = $db->real_escape_string($_POST['email']);
    $query = "SELECT * FROM groomer WHERE email='$email'";
    $result = mysqli_query($db,$query);

    $return_arr = array();

    while($row = mysqli_fetch_array($result)){
        $company = $row['company'];
        $city = $row['city'];
        $address = $row['address'];
        $postcode = $row['postcode'];
        $phone = $row['phone'];
        $afm = $row['afm'];

        $return_arr[] = array("company" => $company, "city" => $city, "address" => $address, "postcode" => $postcode, "phone" => $phone, "afm" => $afm);
    }

    exit(json_encode($return_arr));
}

```

Sourcecode 3.15

Με τη χρήση JavaScript ενημερώνουμε την φόρμα στοιχείων του groomer (Sourcecode 3.16).

```

//----- Load Groomer Profile Data -----//
var email = $('#email').val();

$.ajax({
  url: '../../groomer/groomer_server.php',
  type: 'post',
  dataType: 'JSON',
  data: {
    groomer_profile_from_ajax:1,
    email:email
  },
  success: function(response){
    var len = response.length;
    for(var i=0; i<len; i++){
      var company = response[i].company;
      var city = response[i].city;
      var address = response[i].address;
      var postcode = response[i].postcode;
      var phone = response[i].phone;
      var afm = response[i].afm;
    }

    $('#company').val(company);

    $('#[id=city] option').filter(function() {
      return ($(this).text() == city);
    }).prop('selected', true);

    $('#address').val(address);

    $('#postcode').val(postcode);

    $('#phone').val(phone);

    $('#afm').val(afm);
  }
});

```

Sourcecode 3.16

### Ενημέρωση ενός στοιχείου από το προφίλ του Groomer

Ακολουθεί το Sourcecode 3.17 για την ενημέρωση ενός στοιχείου από του προφίλ του Groomer, με τη χρήση Ajax Request.

```

$('#profile_edit_company_btn').click(function () {
  error_profile_company = false;
  check_profile_company();

  var email = $('#email').val().trim();
  var company = $('#company').val().trim();

  if(error_profile_company == 0){
    $.ajax({
      url: '../../groomer/groomer_server.php',
      type: 'post',
      data: {
        profile_edit_company_from_ajax:1,
        company:company,email:email
      },
      success: function(response){
        if(response.indexOf('success') >= 0){
          Swal.fire({
            icon: 'success',
            text: "Το όνομα της εταιρείας ενημερώθηκε με επιτυχία",
            confirmButtonColor: "#b5c5ff",
          });
          document.getElementById("company").style.borderColor = "#ddd";
        }
      }
    });
    return true;
  }else{
    return false;
  }
});

```

Sourcecode 3.17

Στη συνέχεια με τη χρήση PHP ενημερώνουμε τη βάση δεδομένων, με βάση το παρακάτω Sourcecode 3.18.

```
if(isset($_POST['profile_edit_company_from_ajax'])) {  
    $company = $db->real_escape_string($_POST['company']);  
    $email = $db->real_escape_string($_POST['email']);  
  
    $query = "UPDATE groomer SET company='$company' WHERE email='$email'";  
  
    $result = mysqli_query($db,$query);  
  
    if($result){  
        exit("success");  
    }  
}
```

Sourcecode 3.18

### Πίνακας Ραντεβού Groomer

Στο παρακάτω Sourcecode 3.19 και Sourcecode 3.20 παρουσιάζεται ο πίνακας με τα ραντεβού του groomer.

```
<?php  
session_start();  
  
require_once '../includes/database/config.php';  
  
if(!isset($_SESSION['groomerLoggedIn'])){  
    header('Location: ../index.php');  
    exit();  
}  
  
$email = $_SESSION['email'];  
$query = "SELECT * FROM groomer WHERE email='$email'";  
$result = mysqli_query($db,$query);  
while($row = mysqli_fetch_array($result)){  
    $groomer_id = $row['id'];  
}  
  
$select_appointments = "SELECT * FROM appointments WHERE groomer_id=$groomer_id AND is_expired=0  
ORDER BY appointment_date DESC, appointment_time,appointment_date";  
$select_appointments_result = mysqli_query($db,$select_appointments);  
?>
```

Sourcecode 3.19

```
<?php if($select_appointments_result->num_rows > 0){ ?>  
<table class="table table-striped" >  
    <thead>  
        <tr>  
            <th scope="col">#</th>  
            <th scope="col">Ημερομηνία</th>  
            <th scope="col">Ώρα</th>  
            <th scope="col">Κατάσταση</th>  
            <th scope="col">Προβολή</th>  
        </tr>  
    </thead>  
    <tbody>  
        <tr>  
            <?php  
            $counter = 0;  
            while ($row = $select_appointments_result->fetch_assoc()) {  
                $counter++;  
                echo"<tr class='table_row'>";  
                echo"<td id='td'>". $counter . "</td>";  
                echo"<td id='td'>". $row['appointment_date'] . "</td>";  
                echo"<td id='td'>". $row['appointment_time'] . "</td>";  
  
                if($row['booked'] == 1){  
                    echo"<td id='td'><button id='btn_booked' type='button' class='btn btn-outline-primary'>Εκπεριε</button></td>";  
                }else if($row['in_process'] == 1){  
                    echo"<td id='td'><button type='button' class='btn btn-outline-primary'>Εε εξέλιξη</button></td>";  
                }else if($row['completed'] == 1){  
                    echo"<td id='td'><button id='btn_completed' type='button' class='btn btn-outline-primary'>Ολοκληρωμένο</button></td>";  
                }else if($row['canceled'] == 1){  
                    echo"<td id='td'><button id='btn_canceled' type='button' class='btn btn-outline-primary'>Ακυρωμένο</button></td>";  
                }  
                }  
                <td><button type="button" class="btn btn-info btn-icon btn_info" id="btn_info" value="<?php echo $row['id'] ?>" data-toggle="modal" data-target="#modal<?php echo $row['id'] ?>">  
                    <i class="fa fa-eye btn-icon"></i>  
                </button>  
            </td>  
            <?php  
            echo"</tr>";  
        }  
    }  
    </tbody>  
</table>  
<?php }else{ ?>  
    <div class="groomer_appoint_table_empty">Μέχρι στιγμής δεν υπάρχει κάποιο ραντεβού σε εξέλιξη.</div>  
</div>
```

Sourcecode 3.20



## Ιστορικό Ραντεβού Groomer

Στο παρακάτω Sourcecode 3.21 και Sourcecode 3.22 παρουσιάζεται το ιστορικό με τα ραντεβού του groomer

```
<?php
session_start();
require_once '../includes/database/config.php';
if(!isset($_SESSION['groomerLoggedIn'])){
    header('Location: ../index.php');
    exit();
}
$email = $_SESSION['email'];
$query = "SELECT * FROM groomer WHERE email='$email'";
$result = mysqli_query($db,$query);
while($row = mysqli_fetch_array($result)){
    $groomer_id = $row['id'];
}
$select_appointments = "SELECT * FROM appointments WHERE groomer_id=$groomer_id AND is_expired=1 ORDER BY appointment_date DESC, appointment_time,appointment_date,appointment_date";
$select_appointments_result = mysqli_query($db,$select_appointments);
?>
```

Sourcecode 3.21

```
<?php if($select_appointments_result->num_rows > 0){ ?>
<table class="table table-striped" >
<thead>
<tr>
<th scope="col">#</th>
<th scope="col">Ημερομηνία</th>
<th scope="col">Ωρα</th>
<th scope="col">Κατάσταση</th>
<th scope="col">Προβόλες</th>
</tr>
</thead>
<tbody>
<tr>
<?php
$count = 0;
while ($row = $select_appointments_result->fetch_assoc()) {
    $count++;
    echo<tr class="table-row">;
    echo<td id="td"> . $count . "</td>;
    echo<td id="td"> . $row['appointment_date'] . "</td>;
    echo<td id="td"> . $row['appointment_time'] . "</td>;

    if($row['booked'] == 1){
        echo<td id="td"><button id="btn_booked" type="button" class="btn btn-outline-primary">Εκπεφαι</button></td>;
    }else if($row['in_process'] == 1){
        echo<td id="td"><button type="button" class="btn btn-outline-primary">Εκτελείται</button></td>;
    }else if($row['completed'] == 1){
        echo<td id="td"><button id="btn_completed" type="button" class="btn btn-outline-primary">Ολοκληρωμένο</button></td>;
    }else if($row['canceled'] == 1){
        echo<td id="td"><button id="btn_canceled" type="button" class="btn btn-outline-primary">Ακυρωμένο</button></td>;
    }
    <td><button type="button" class="btn btn-info btn-icon btn-info" id="btn_info" value="<?php echo $row['id'] ?>" data-toggle="modal" data-target="#modal<?php echo $row['id'] ?>">
        <i class="fa fa-eye btn-icon"></i>
    </button>
</td>
</tr>
</tbody>
</table>
<?php elseif ?>
<div class="groomer_appoint_history_table_empty">Το ιστορικό σας είναι κενό.</div>
<?php ?>
```

Sourcecode 3.22

## Ωράριο Λειτουργίας Groomer

Στο παρακάτω Sourcecode 3.23 και Sourcecode 3.24 παρουσιάζεται η δημιουργία του dropdown για το ωράριο λειτουργίας.



```

$('#btn_save_monday_day').click(function () {

    error_from_monday_day = false;
    error_until_monday_day = false;
    error_from_monday_day_extra = false;
    error_until_monday_day_extra = false;

    var groomer_email = $('#session_email').val();

    var day_id = $("#date option:selected").attr('id');

    var from_monday_day = $("#from_monday_day option:selected").val();
    var until_monday_day = $("#until_monday_day option:selected").val();

    var from_monday_day_extra = $("#from_monday_day_extra option:selected").val();
    var until_monday_day_extra = $("#until_monday_day_extra option:selected").val();

    var isOpenChecked = $('#monday_day_open').is(':checked');
    var isOpenCheckedExtra = $('#monday_day_extra_open').is(':checked');

    var isClosedChecked = $('#monday_day_closed').is(':checked');

    var isOpenedCollapse = $('#collapseExtraVardiaMonday').attr("aria-expanded", true);

    if(isOpenChecked){
        check_monday_day_from();
        check_monday_day_until();
    }
    if(isOpenCheckedExtra){
        check_monday_day_extra_from();
        check_monday_day_extra_until();
    }
}

```

Sourcecode 3.25

```

if(isClosedChecked){

    $.ajax({
        url: '../groomer/groomer_server.php',
        type: 'post',
        data: {
            closed_monday_hours:1,
            groomer_email:groomer_email,day_id:day_id,
            from_monday_day:from_monday_day,until_monday_day:until_monday_day,
            from_monday_day_extra:from_monday_day_extra,until_monday_day_extra:until_monday_day_extra
        },
        success: function(response){
            if(response.indexOf('update_working_hours') >= 0){
                Swal.fire({
                    icon: 'success',
                    html: "Το ωράριο της Δευτέρας ενημερώθηκε με επιτυχία" + "</br>" + "Το κατάστημά σας θα παραμείνει κλειστό",
                    confirmButtonColor: "#b5c5ff",
                }).then(function() {
                    location.reload();
                });
                document.getElementById("from_monday_day").style.borderColor = "#ddd";
                document.getElementById("until_monday_day").style.borderColor = "#ddd";
                document.getElementById("from_monday_day_extra").style.borderColor = "#ddd";
                document.getElementById("until_monday_day_extra").style.borderColor = "#ddd";
            }

            if(response.indexOf('insert_working_hours') >= 0){
                Swal.fire({
                    icon: 'success',
                    html: "Το ωράριο της Δευτέρας καταχωρήθηκε με επιτυχία" + "</br>" + "Το κατάστημά σας θα παραμείνει κλειστό",
                    confirmButtonColor: "#b5c5ff",
                }).then(function() {
                    location.reload();
                });
                document.getElementById("from_monday_day").style.borderColor = "#ddd";
                document.getElementById("until_monday_day").style.borderColor = "#ddd";
                document.getElementById("from_monday_day_extra").style.borderColor = "#ddd";
                document.getElementById("until_monday_day_extra").style.borderColor = "#ddd";
            }
        }
    });
}
}

```

Sourcecode 3.26



```

##### Save Closed Monday Working Hours #####
if(isset($_POST['closed_monday_hours'])) {
    $groomer_email = $db->real_escape_string($_POST['groomer_email']);
    $day_id = $db->real_escape_string($_POST['day_id']);
    $from_monday_day = $db->real_escape_string($_POST['from_monday_day']);
    $until_monday_day = $db->real_escape_string($_POST['until_monday_day']);
    $from_monday_day_extra = $db->real_escape_string($_POST['from_monday_day_extra']);
    $until_monday_day_extra = $db->real_escape_string($_POST['until_monday_day_extra']);

    $query_groomer = "SELECT id FROM groomer WHERE email='$groomer_email'";
    $result_groomer = mysqli_query($db,$query_groomer);

    if($result_groomer->num_rows > 0) {
        $groomer_row = mysqli_fetch_array($result_groomer);
        $groomer_id = $groomer_row['id'];

        // Check If has hour with this groomer
        $query_hour = "SELECT * FROM groomer_working_hours WHERE groomer_id='$groomer_id' AND day_id=$day_id";
        $result_hour = mysqli_query($db,$query_hour);

        // If this groomer has set open hour, then update this hour
        // We dont want duplicated hours
        if($result_hour->num_rows > 0) {
            $query = "UPDATE groomer_working_hours SET day_id=$day_id,is_closed=1,status='closed',
                from_hour='$from_monday_day', until_hour='$until_monday_day', from_hour_extra='$from_monday_day_extra',
                until_hour_extra='$until_monday_day_extra' WHERE groomer_id=$groomer_id AND day_id=$day_id";

            $result = mysqli_query($db,$query);
            exit("update_working_hours");
        }else{
            // If this groomer has not set open hour, then insert this hour
            // We dont want duplicated hours
            $query = "INSERT INTO groomer_working_hours (groomer_id, day_id, from_hour, until_hour, from_hour_extra, until_hour_extra,is_closed,status)
                VALUES ($groomer_id,$day_id,'$from_monday_day','$until_monday_day',
                '$from_monday_day_extra','$until_monday_day_extra',1,'closed')";

            $result = mysqli_query($db,$query);
            exit("insert_working_hours");
        }
    }
}

```

### Sourcecode 3.29

```

##### Save Opened Monday Working Hours #####
if(isset($_POST['opened_monday_hours'])) {
    $groomer_email = $db->real_escape_string($_POST['groomer_email']);
    $day_id = $db->real_escape_string($_POST['day_id']);
    $from_monday_day = $db->real_escape_string($_POST['from_monday_day']);
    $until_monday_day = $db->real_escape_string($_POST['until_monday_day']);
    $from_monday_day_extra = $db->real_escape_string($_POST['from_monday_day_extra']);
    $until_monday_day_extra = $db->real_escape_string($_POST['until_monday_day_extra']);

    $query_groomer = "SELECT id FROM groomer WHERE email='$groomer_email'";
    $result_groomer = mysqli_query($db, $query_groomer);

    if ($result_groomer->num_rows > 0) {
        $groomer_row = mysqli_fetch_array($result_groomer);
        $groomer_id = $groomer_row['id'];

        // Check If has hour with this groomer
        $query_hour = "SELECT * FROM groomer_working_hours WHERE groomer_id='$groomer_id' AND day_id=$day_id";
        $result_hour = mysqli_query($db, $query_hour);

        // If this groomer has set open hour, then update this hour
        // We dont want duplicated hours
        if ($result_hour->num_rows > 0) {
            $query = "UPDATE groomer_working_hours SET day_id=$day_id,is_closed=0,
                from_hour='$from_monday_day', until_hour='$until_monday_day', from_hour_extra='$from_monday_day_extra',
                until_hour_extra='$until_monday_day_extra' WHERE groomer_id=$groomer_id AND day_id=$day_id";

            $result = mysqli_query($db, $query);
            if($result){
                date_default_timezone_set("Europe/Athens");
                $date = date('H:i');
                if(($date>$from_monday_day && $date<$until_monday_day) || ($date>$from_monday_day_extra && $date<$until_monday_day_extra)){
                    $update_query = "UPDATE groomer_working_hours SET status='open' WHERE groomer_id=$groomer_id AND day_id=$day_id";
                    $update_result = mysqli_query($db, $update_query);
                    exit("update_working_hours");
                }else{
                    $update_query = "UPDATE groomer_working_hours SET status='closed' WHERE groomer_id=$groomer_id AND day_id=$day_id";
                    $update_result = mysqli_query($db, $update_query);
                    exit("update_working_hours");
                }
            }
        }
    }
}

```

### Sourcecode 3.30

```

    } else {
        // If this groomer has not set open hour, then insert this hour
        // We dont want duplicated hours
        $query = "INSERT INTO groomer_working_hours (groomer_id, day_id, from_hour, until_hour, from_hour_extra, until_hour_extra, is_closed)
VALUES ($groomer_id,$day_id,'$from_monday_day','$until_monday_day',
'$from_monday_day_extra','$until_monday_day_extra',0)";

        $result = mysqli_query($db, $query);
        if($result){
            date_default_timezone_set("Europe/Athens");
            $date = date('H:i');
            if(($date>$from_monday_day && $date<$until_monday_day) || ($date>$from_monday_day_extra && $date<$until_monday_day_extra )){
                $update_query = "UPDATE groomer_working_hours SET status='open' WHERE groomer_id=$groomer_id AND day_id=$day_id";
                $update_result = mysqli_query($db, $update_query);
                exit("insert_working_hours");
            }else{
                $update_query = "UPDATE groomer_working_hours SET status='closed' WHERE groomer_id=$groomer_id AND day_id=$day_id";
                $update_result = mysqli_query($db, $update_query);
                exit("insert_working_hours");
            }
        }
    }
}
}
}

```

Sourcecode 3.31

```

##### Save Opened Monday Working Hours #####
if(isset($_POST['opened_monday_hours'])) {
    $groomer_email = $db->real_escape_string($_POST['groomer_email']);
    $day_id = $db->real_escape_string($_POST['day_id']);
    $from_monday_day = $db->real_escape_string($_POST['from_monday_day']);
    $until_monday_day = $db->real_escape_string($_POST['until_monday_day']);
    $from_monday_day_extra = $db->real_escape_string($_POST['from_monday_day_extra']);
    $until_monday_day_extra = $db->real_escape_string($_POST['until_monday_day_extra']);

    $query_groomer = "SELECT id FROM groomer WHERE email='$groomer_email'";
    $result_groomer = mysqli_query($db, $query_groomer);

    if ($result_groomer->num_rows > 0) {
        $groomer_row = mysqli_fetch_array($result_groomer);
        $groomer_id = $groomer_row['id'];

        // Check if has hour with this groomer
        $query_hour = "SELECT * FROM groomer_working_hours WHERE groomer_id=$groomer_id AND day_id=$day_id";
        $result_hour = mysqli_query($db, $query_hour);

        // If this groomer has set open hour, then update this hour
        // We dont want duplicated hours
        if ($result_hour->num_rows > 0) {
            $query = "UPDATE groomer_working_hours SET day_id=$day_id,is_closed=0,
from_hour='$from_monday_day', until_hours='$until_monday_day', from_hour_extra='$from_monday_day_extra',
until_hour_extra='$until_monday_day_extra' WHERE groomer_id=$groomer_id AND day_id=$day_id";

            $result = mysqli_query($db, $query);
            if($result){
                date_default_timezone_set("Europe/Athens");
                $date = date('H:i');
                if(($date>$from_monday_day && $date<$until_monday_day) || ($date>$from_monday_day_extra && $date<$until_monday_day_extra )){
                    $update_query = "UPDATE groomer_working_hours SET status='open' WHERE groomer_id=$groomer_id AND day_id=$day_id";
                    $update_result = mysqli_query($db, $update_query);
                    exit("update_working_hours");
                }else{
                    $update_query = "UPDATE groomer_working_hours SET status='closed' WHERE groomer_id=$groomer_id AND day_id=$day_id";
                    $update_result = mysqli_query($db, $update_query);
                    exit("update_working_hours");
                }
            }
        }
    }
}
}

```

Sourcecode 3.32

```

    } else {
        // If this groomer has not set open hour, then insert this hour
        // We dont want duplicated hours
        $query = "INSERT INTO groomer_working_hours (groomer_id, day_id, from_hour, until_hour, from_hour_extra, until_hour_extra, is_closed)
VALUES ($groomer_id,$day_id,'$from_monday_day','$until_monday_day',
'$from_monday_day_extra','$until_monday_day_extra',0)";

        $result = mysqli_query($db, $query);
        if($result){
            date_default_timezone_set("Europe/Athens");
            $date = date('H:i');
            if(($date>$from_monday_day && $date<$until_monday_day) || ($date>$from_monday_day_extra && $date<$until_monday_day_extra )){
                $update_query = "UPDATE groomer_working_hours SET status='open' WHERE groomer_id=$groomer_id AND day_id=$day_id";
                $update_result = mysqli_query($db, $update_query);
                exit("insert_working_hours");
            }else{
                $update_query = "UPDATE groomer_working_hours SET status='closed' WHERE groomer_id=$groomer_id AND day_id=$day_id";
                $update_result = mysqli_query($db, $update_query);
                exit("insert_working_hours");
            }
        }
    }
}
}
}

```

Sourcecode 3.33

## Διαθέσιμα Ραντεβού

Στο παρακάτω Sourcecode 3.34 και Sourcecode 3.35 παρουσιάζονται τα διαθέσιμα ραντεβού για τον groomer με μορφή buttons.



Με τη χρήση JavaScript και Ajax Request στέλνουμε τα δεδομένα στον server  
( Sourcecode 3.37- Sourcecode 3.41).

```
$('#btn_modal_save').click(function () {  
  
    error_email = false;  
    error_name = false;  
    error_surname = false;  
    error_phone = false;  
  
    check_modal_email();  
    check_modal_name();  
    check_modal_surname();  
    check_modal_phone();  
  
    var groomer_id = $('#session_groomer_id').val();  
  
    var appointment_date = getUrlParameter('date');  
    var appointment_time = $('#edit_appointment_time').text();  
  
    var modal_email = $('#client_email').val().trim();  
    var modal_name = $('#client_name').val().trim();  
    var modal_surname = $('#client_surname').val().trim();  
    var modal_phone = $('#client_phone').val().trim();  
  
    if(error_email == 0 && error_name == 0 && error_surname == 0 && error_phone == 0 ) {  
  
        $.ajax({  
            url: '../groomer/groomer_server.php',  
            type: 'POST',  
            data: {  
                manual_entry_appointment: 1,  
                groomer_id: groomer_id,  
                appointment_date: appointment_date,  
                appointment_time: appointment_time,  
                modal_email: modal_email,  
                modal_name: modal_name,  
                modal_surname: modal_surname,  
                modal_phone: modal_phone  
            },  
            success: function (response) {  
  
                if (response.indexOf('mi_diathesimo_rantevou') >= 0) {  
  
                    $('#appointment_modal').modal('hide');  
  
                    Swal.fire('Κάτι πήγε στραβά!', "Το ραντεβού είναι μη διαθέσιμο. Επιλέξτε κάποια άλλη ώρα", "error")  
                        .then(function () {  
                            location.reload();  
                        });  
  
                }  
  
            }  
  
        });  
  
    }  
  
});  
});
```

Sourcecode 3.37

```
if (response.indexOf('success_appoint_customer') >= 0) {  
    $('#appointment_modal').modal('hide');  
  
    Swal.fire({  
        icon: 'success',  
        html: 'Το ραντεβού καταχωρήθηκε με επιτυχία',  
    }).then(function (result) {  
        if (result.isConfirmed) {  
            $.ajax({  
                url: '../groomer/groomer_server.php',  
                type: 'POST',  
                data: {  
                    success_appoint_customer_send_email: 1,  
                    modal_email: modal_email,  
                    modal_surname: modal_surname,  
                    appointment_date: appointment_date,  
                    appointment_time: appointment_time  
                },  
  
                success: function () {  
                    var modal_email = $('#client_email').val('');  
                    var modal_name = $('#client_name').val('');  
                    var modal_surname = $('#client_surname').val('');  
                    var modal_phone = $('#client_phone').val('');  
                    document.getElementById("client_email").style.borderColor = "#ddd";  
                    document.getElementById("client_name").style.borderColor = "#ddd";  
                    document.getElementById("client_surname").style.borderColor = "#ddd";  
                    document.getElementById("client_phone").style.borderColor = "#ddd";  
                    location.reload();  
                }  
            });  
        }  
    });  
};  
});  
});
```

Sourcecode 3.38



```

        if (response.indexOf('success_appoint_new_customer') >= 0) {
            $('#appointment_modal').modal('hide');

            Swal.fire({
                icon: 'success',
                html: 'Το πακέτο κατόχησής σας με επιτυχία',
            }).then(function (result) {
                if (result.isConfirmed) {
                    $.ajax({
                        url: '../groomer/groomer_server.php',
                        type: 'POST',
                        data: {
                            success_appoint_new_customer_send_email: 1,
                            modal_email: modal_email,
                            modal_surname: modal_surname,
                            appointment_date: appointment_date,
                            appointment_time: appointment_time
                        },
                    },
                    success: function () {
                        var modal_email = $('#client_email').val('');
                        var modal_name = $('#client_name').val('');
                        var modal_surname = $('#client_surname').val('');
                        var modal_phone = $('#client_phone').val('');
                        document.getElementById("client_email").style.borderColor = "#ddd";
                        document.getElementById("client_name").style.borderColor = "#ddd";
                        document.getElementById("client_surname").style.borderColor = "#ddd";
                        document.getElementById("client_phone").style.borderColor = "#ddd";
                        location.reload();
                    }
                )
            });
        }
    });
    event.preventDefault();
    return true;
} else {
    return false;
}
});

```

Sourcecode 3.39

```

#####
##### Manual Entry Appointment #####
if(isset($_POST['manual_entry_appointment'])) {

    $groomer_id = $db->real_escape_string($_POST['groomer_id']);
    $appointment_date = $db->real_escape_string($_POST['appointment_date']);
    $appointment_time = $db->real_escape_string($_POST['appointment_time']);
    $modal_email = $db->real_escape_string($_POST['modal_email']);
    $modal_name = $db->real_escape_string($_POST['modal_name']);
    $modal_surname = $db->real_escape_string($_POST['modal_surname']);
    $modal_phone = $db->real_escape_string($_POST['modal_phone']);

    $query_check_appointments = "SELECT * FROM appointments WHERE appointment_date = '$appointment_date' AND appointment_time = '$appointment_time'
    AND ( booked=1 OR in_process=1) AND is_expired=0 AND groomer_id=$groomer_id";
    $result_check_appointments = mysql_query($db,$query_check_appointments);

    if ($result_check_appointments->num_rows > 0) {
        exit("mi_diathesimo_rantevou");
    } else {

        $query_check_customer = "SELECT * FROM customer WHERE email='$modal_email'";
        $result_check_customer = mysql_query($db,$query_check_customer);

        if ($result_check_customer->num_rows > 0) {
            while ($row = mysql_fetch_array($result_check_customer)) {
                $is_user = $row['is_user'];
                $user_id = $row['user_id'];
            }
            $query_insert_customer = "INSERT INTO customer (email, name, surname, phone, appointment_date, appointment_time, is_user, user_id)
            VALUES ('$modal_email', '$modal_name', '$modal_surname', '$modal_phone', '$appointment_date', '$appointment_time', $is_user, $user_id)";
            $result_insert_customer = mysql_query($db,$query_insert_customer);

            if ($result_insert_customer) {
                $query_check_customer = "SELECT * FROM customer WHERE email='$modal_email' AND appointment_date='$appointment_date'
                AND appointment_time='$appointment_time'";
                $result_check_customer = mysql_query($db, $query_check_customer);

                while ($row = mysql_fetch_array($result_check_customer)) {
                    $customer_id = $row['id'];
                }

                $query_insert_appoint = "INSERT INTO appointments (customer_id, groomer_id, appointment_date, appointment_time, booked, in_process)
                VALUES ('$customer_id', $groomer_id, '$appointment_date', '$appointment_time', 0, 1)";
                $result_insert_appoint = mysql_query($db, $query_insert_appoint);

                if ($result_insert_appoint) {
                    exit("success_appoint_customer");
                }
            }
        }
    }
}

```

Sourcecode 3.40



## ΒΙΒΛΙΟΓΡΑΦΙΑ/ΔΙΚΤΥΟΓΡΑΦΙΑ

1. Δουλεύοντας με το Bootstrap. (14/09/15) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://www.toastedweb.gr>
2. ΤΑΜΠΑΚΑΣ, Β. (2017). ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ. ΠΑΤΡΑ: ΕΚΔΟΣΕΙΣ GOTSIS
3. Η πρώτη ιστοσελίδα javascript. (13/06/17) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://www.learncode.gr>
4. Ξεκινώντας με την jQuery. (21/11/11) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://gr.pcmag.com>
5. Τι είναι η HTML?. (χ.η.) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://www.ip.gr>
6. Τι Είναι η Web Εφαρμογή. (χ.η.) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://www.intel-soft.gr/>
7. Τι είναι μια βάση MySQL?. (χ.η.) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://www.ip.gr>
8. Τι είναι το CSS?. (χ.η.) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://www.ip.gr>
9. Τι είναι Cron job;. (χ.η.) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://ti-einai.gr>
10. Τι είναι Javascript;. (χ.η.) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://ti-einai.gr>
11. Bootstrap 4 Tutorial. (χ.η.) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://www.w3schools.com>
12. Bootstrap Framework. (χ.η.) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://elearn.ellak.gr>
13. Client-Side vs Server-Side Web Development. (20/08/20) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://careerkarma.com>
14. CSS Tutorial. (χ.η.) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://www.w3schools.com>
15. Database issues. (χ.η.) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://www.php.net>
16. Difference Between CSS and Bootstrap. (05/06/20) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://www.geeksforgeeks.org>
17. HTML Tutorial. (χ.η.) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://www.w3schools.com>
18. Javascript. (26/06/17) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://www.learncode.gr>

19. JavaScript. (29/07/21) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://developer.mozilla.org>
20. JavaScript Tutorial. (χ.η.) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://www.w3schools.com>
21. jQuery: Εισαγωγή. (χ.η.) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://webfocus.gr>
22. jQuery Tutorial. (χ.η.) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://www.w3schools.com>
23. PHP. (15/06/15) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://www.learncode.gr>
24. phpMyAdmin. (χ.η.) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://docs.phpmyadmin.net>
25. PHP Tutorial. (χ.η.) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://www.w3schools.com>
26. SQL Tutorial. (χ.η.) Ανακτήθηκε Νοέμβριο 5, 2021 από <https://www.w3schools.com>