

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΑΝΑΠΤΥΞΗ ΠΛΑΤΦΟΡΜΑΣ ΚΟΙΝΩΝΙΚΗΣ ΔΙΚΤΥΩΣΗΣ ΦΟΙΤΗΤΩΝ
ΣΕ ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ**

Γραμματικοπούλου Μαριάννα ΑΜ: 2675

Παπαδούρου Αιμιλία ΑΜ: 2673

Επιβλέπων Καθηγητής:
Παναγιώτης Σπ. Αλεφραγκής
Διδάκτωρ Ηλεκτρολόγος Μηχανικός & Τεχνολογίας

1.Εισαγωγή	3
1.1 Περίληψη	3
1.2 Κίνητρο Δημιουργίας της Πλατφόρμας	3
1.3 Δομή Εργασίας	4
2. Μεθοδολογία Υλοποίησης	4
2.1 Δομή Εργασίας & Μέθοδος Ανάλυσης	5
2.1.1 Εξυπηρετητής (Server)	5
2.1.2 Βάση δεδομένων	6
2.1.3 Admin Panel	7
2.1.4 Διαδικτυακή Εφαρμογή	7
2.1.4.1 Php-Laravel Framework	7
2.1.4.2 HTML5	10
2.1.4.3 Css/Bootstrap	10
2.1.4.4 JavaScript/JQuery	10
2.1.5 Progressive Web App	11
2.2 Μέθοδος Ανάπτυξης	11
3. Σχέδιο Δράσης	12
3.1 Σχεδιασμός και Υλοποίηση συστήματος	12
3.1.1 Υλοποίηση και σχεδιασμός της Βάσης Δεδομένων	12
3.1.1.1 Database Schema-Entity Relationship (ER) Diagram	13
3.1.2 Υλοποίηση και σχεδιασμός Admin Panel	14
3.1.2.1 Admin Panel - Dashboard	14
3.1.2.2 Admin Panel - News	14
3.1.2.3 Admin Panel - Courses	15
3.1.2.4 Admin Panel - Users Enrolls	16
3.1.2.5 Admin Panel - Notes	16
3.1.2.6 Admin Panel - User Management	17
3.1.2.7 Admin Panel - Change Password	19
3.1.3 Υλοποίηση και σχεδιασμός Web Application	19
3.1.3.1 Το μοντέλο MVC(Model-View-Controller)	19
3.1.3.2 Στοιχεία του μοντέλου MVC	20
3.1.3.2.1 Routes	20
3.1.3.2.2 Controllers	23
3.1.3.2.3 Models	31
3.1.3.2.3.1 Migrations	37
3.1.3.2.4 Views	37
3.1.4 Υλοποίηση και σχεδιασμός Progressive Web App	39
4. Κύριο μέρος – Σενάρια Χρήσης	40
4.1 Flow Diagram	40
4.2 User perspective	43
5. Αποτελέσματα	50
5.1 Συμπεράσματα	51
5.2 Μελλοντική Επέκταση	51

1.Εισαγωγή

Η διπλωματική εργασία —ένα μικρό χρονικά κομμάτι των προπτυχιακών σπουδών ενός φοιτητή— χρήζει σπουδαίας σημαντικότητας, καθώς προσφέρει στον φοιτητή μία πτυχή του εργασιακού τομέα που έχει επιλέξει να ακολουθήσει, εντάσσοντας τον σε μία πρώτη γενική κατεύθυνση αυτού. Παράλληλα, δίνει την ευκαιρία στον σπουδαστή να πειραματιστεί με τεχνολογίες και πρακτικές, που ακολουθούνται στον κλάδο του, παρέχοντας ταυτόχρονα την ευελιξία της επιλογής και του συνδυασμού, ώστε να προσκομίσει όσο το δυνατόν μία πιο σφαιρική εικόνα.

1.1 Περίληψη

Η εν λόγω πτυχιακή εργασία έχει ως στόχο το σχεδιασμό και την ανάπτυξη μιας πλατφόρμας για τους Φοιτητές του Πανεπιστημίου, με σκοπό την καλύτερη οργάνωσή τους κατά τη περίοδο φοίτησης.

Αναλυτικότερα, η πλατφόρμα θα έχει δύο βασικές λειτουργίες. Η πρώτη λειτουργία αφορά την πρόοδο του φοιτητή. Κατά την δημιουργία λογαριασμού και είσοδο του στην πλατφόρμα, θα υπάρχουν όλα τα διαθέσιμα μαθήματα ανά εξάμηνο, στα οποία ο φοιτητής καλείται να κάνει εγγραφή. Μετά την εγγραφή και ολοκλήρωση των μαθημάτων, καλείται να εισάγει τον βαθμό που έλαβε για το μάθημα. Με αυτόν τον τρόπο, στο προφίλ του χρήστη εμφανίζονται αυτόματα το ποσοστό ολοκλήρωσης των μαθημάτων(progress bar), οι Διδακτικές Μονάδες που διαθέτει καθώς και ο τρέχων βαθμός πτυχίου.

Η δεύτερη λειτουργία αφορά το διαμοιρασμό σημειώσεων μεταξύ των φοιτητών. Συγκεκριμένα σε κάθε μάθημα θα υπάρχουν σημειώσεις που το αφορούν, αναρτημένες από οποιονδήποτε χρήστη έχει παρακολουθήσει το μάθημα. Επίσης θα υπάρχει η επιλογή της αξιολόγησης αυτών των σημειώσεων για την αποφυγή παραπληροφόρησης. Η αξιολόγηση αυτή θα γίνεται μέσω like και dislike. Οι σημειώσεις θα δημοσιεύονται μετά από έγκριση του διαχειριστή για την αποφυγή ακατάλληλου περιεχομένου.

Τέλος, η πλατφόρμα θα δίνει τη δυνατότητα εγκατάστασης σε όλες τις συσκευές (μέσω Progressive Web App) ώστε ο φοιτητής να έχει όσο το δυνατόν καλύτερη και ευκολότερη πρόσβαση.

1.2 Κίνητρο Δημιουργίας της Πλατφόρμας

Κατά την φοίτηση μας στο Πανεπιστήμιο Πελοποννήσου, είχαμε εντοπίσει κάποιες αδυναμίες στις ήδη υπάρχουσες πλατφόρμες καθώς και στην εύρεση υλικού για την καλύτερη προετοιμασία του φοιτητή στις επερχόμενες εξεταστικές . Εμπνευστήκαμε λοιπόν

από αυτό και θελήσαμε να φτιάξουμε μια cross-device πλατφόρμα όπου ο φοιτητής θα βρίσκει ότι πληροφορία θέλει, είτε αφορά κάποιο μάθημα, είτε την πρόοδο του.

1.3 Δομή Εργασίας

Η εν λόγω εργασία είναι δομημένη σε πέντε (5) κεφάλαια τα οποία έχουν ως στόχο τη ανάλυση των βασικών σκελών της, τόσο δηλαδή το σχεδιαστικό όσο και το κομμάτι της ανάπτυξης, αλλά και να μεταφέρει στον αναγνώστη, τα προβλήματα τα οποία επιλύει το συγκεκριμένο σύστημα.

Επί των κεφαλαίων, πιο συγκεκριμένα, το **Κεφάλαιο 1** αποτελεί μία πρώτη επαφή, καθώς στοχεύει στην εισαγωγή της ιδέας —παρουσιάζοντας βασικούς όρους και έννοιες συνοπτικά— και πως η συγκεκριμένη πλατφόρμα μπορεί να βοηθήσει στο ζήτημα της οργάνωσης των σημειώσεων του φοιτητή και στη γενική προετοιμασία του για την εξεταστική περίοδο και ακόμα στον υπολογισμό των διδακτικών μονάδων του με βάση τα μαθήματα που ο βαθμός είναι επαρκής.

Στο **Κεφάλαιο 2**, θα υπάρξει αναφορά στους τρόπους αντιμετώπισης των προβλημάτων που θέτει για επίλυση η εν λόγω εργασία, καθώς και την μεθοδολογία που ακολουθείται για να έρθουν εις πέρας. Επίσης, θα γίνει αναφορά σε σχετικές τεχνολογίες που εφαρμόστηκαν για να γίνει δυνατή η επίλυση τους.

Στο αμέσως επόμενο **Κεφάλαιο 3**, γίνεται μια σύντομη παρουσίαση της διαδικτυακής πλατφόρμας αλλά και της cross-platform εφαρμογής, με επίκεντρο τις βασικές λειτουργίες τους.

Στο **Κεφάλαιο 4** γίνεται λεπτομερής αναφορά και ανάλυση και της πλατφόρμας. Επίσης, αναλύονται τα βασικά μέρη κάθε υλοποίησης, με την βοήθεια ενός γενικού σεναρίου, που έχει ως στόχο την εξοικείωση του αναγνώστη με όλα τα κομμάτια και τις δυνατότητες του συστήματος. Στην προσπάθεια αυτή, θα γίνει χρήση εικόνων καθώς και αποσπασμάτων πηγαίου κώδικα, ώστε να υπάρξει διαφάνεια μεταξύ των εκάστοτε λειτουργιών και τον συστημάτων που τα απαρτίζουν.

Τέλος, η παρούσα αναφορά ολοκληρώνεται με το **Κεφάλαιο 5**, με μία σύνοψη στα οφέλη της συγκεκριμένης πτυχιακής εργασίας, καθώς και στα επιτεύγματα μέσα από την διαδικασία ολοκλήρωσης αυτής, τόσο σε προσωπικό επίπεδο όσο και στο συνολικό αντίκτυπο.

2. Μεθοδολογία Υλοποίησης

Στην ενότητα αυτή επιχειρείται μία συνοπτική περιγραφή της μεθοδολογίας που ακολουθείται και στην αντιμετώπιση ζητημάτων που προέκυψαν κατά τον σχεδιασμό και την ανάπτυξη του συγκεκριμένου συστήματος. Επίσης, θα γίνει αναφορά στις βασικές θεωρητικές και τεχνολογικές έννοιες που σχετίζονται με την εν λόγω εργασία, με αρχή το

γενικότερο αντικείμενο της οργάνωσης της μελέτης του φοιτητή και της συνολικής προόδου του.

2.1 Δομή Εργασίας & Μέθοδος Ανάλυσης

Η βασική δομή της εργασίας αποτελείται από τέσσερα (4) κύρια μέρη, το καθένα με τη δική του ανάγκη για ξεχωριστή μελέτη και έρευνα. Η δομή αυτή προέκυψε ύστερα από μελέτη των απαιτήσεων, στόχων αλλά και ζητούμενων βάσει των προβλημάτων που είχαν τεθεί στο στόχαστρο επίλυσης. Τα μέρη αυτά αναφέρονται επιγραμματικά παρακάτω, ενώ θα αναλυθούν σε ξεχωριστές υπό-ενότητες, αναπτύσσοντας τον τρόπο σκέψης και προσέγγισης που ακολουθήθηκε για κάθε ένα ξεχωριστά, ώστε να γίνει η ανάλογη επιλογή τεχνολογιών για κάθε ένα εξ'αυτών:

- Εξυπηρετητής(Server)
- Βάση δεδομένων
- Admin Panel
- Διαδικτυακή εφαρμογή
- Progressive Web App(Cross-Platform εφαρμογή)

2.1.1 Εξυπηρετητής (Server)

Ο Server ή αλλιώς εξυπηρετητής ή διακομιστής, στην πιο απλή του μορφή είναι ένας ηλεκτρονικός υπολογιστής που τρέχει κατάλληλο λογισμικό ώστε να εξυπηρετεί τους χρήστες που συνδέονται με αυτόν για κάποιο σκοπό. Ανάλογα με τον σκοπό, ο server τρέχει και τις κατάλληλες υπηρεσίες και έχει και κατάλληλη ονομασία.

Για παράδειγμα:

– αν ο server εξυπηρετεί ιστοσελίδες, ονομάζεται web server και τρέχει λογισμικό κατάλληλο για να εξυπηρετεί τα αρχεία με τα οποία είναι κατασκευασμένη μία ιστοσελίδα.

– αν ο server εξυπηρετεί e-mail λογαριασμούς ώστε οι χρήστες να έχουν το e-mail τους σε αυτόν, τότε θα είναι e-mail server και θα τρέχει κατάλληλο λογισμικό για αποστολή και λήψη e-mail μηνυμάτων.

Όπως είπαμε, στην πιο απλή του μορφή ένας server είναι ένας ηλεκτρονικός υπολογιστής ο οποίος τρέχει κατάλληλο λογισμικό. Στην πράξη βέβαια, δεν είναι απλοί υπολογιστές αλλά είναι ηλεκτρονικοί υπολογιστές αυξημένων δυνατοτήτων. Ακριβώς επειδή η δουλειά ενός server είναι να εξυπηρετεί τους χρήστες που τον επισκέπτονται, ανάλογα με το πόσο μεγάλη είναι αυτή η επισκεψιμότητα (πόσοι χρήστες τον επισκέπτονται) αλλά και το τι εργασίες πρέπει να εκτελέσει ο server για κάθε επισκέπτη που συνδέεται με αυτόν ώστε να του δώσει αυτό που θέλει και μάλιστα αρκετά γρήγορα, οι servers έχουν αυξημένες δυνατότητες σε:

1. χωρητικότητα και ταχύτητα δίσκων
2. μνήμη RAM
3. επεξεργαστική ισχύ (CPU)
4. χωρητικότητα δικτύου

5. αξιολογιστικά υλικών

Ο server που χρησιμοποιήθηκε στο συγκεκριμένο project παρέχεται από την εταιρία CretaForce, και καλύπτει πλήρως τις ανάγκες μας με τις δυνατότητές του.

2.1.2 Βάση δεδομένων

Για τις ανάγκες της εφαρμογής του Pa-Pel χρησιμοποιήθηκε η MySQL(Relational Database). Με τον όρο σχεσιακή βάση δεδομένων εννοείται μία συλλογή δεδομένων οργανωμένη σε συσχετισμένους πίνακες που παρέχει ταυτόχρονα ένα μηχανισμό για ανάγνωση, εγγραφή, τροποποίηση ή και πιο πολύπλοκες διαδικασίες πάνω στα δεδομένα. Ο σκοπός μιας βάσης δεδομένων είναι η οργανωμένη αποθήκευση πληροφορίας και η δυνατότητα εξαγωγής της πληροφορίας αυτής, ιδίως σε πιο οργανωμένη μορφή, σύμφωνα με ερωτήματα που τίθενται στη σχεσιακή βάση δεδομένων. Τα δεδομένα είναι δυνατόν να αναδιοργανώνονται με πολλούς διαφορετικούς τρόπους, σε νοητούς πίνακες, χωρίς να είναι απαραίτητη η αναδιοργάνωση των φυσικών πινάκων που τα αποθηκεύουν.

Πιο συγκεκριμένα, η MySQL μας παρέχει αρκετά πλεονεκτήματα με τη χρήση της όπως είναι:

1. **Μειωμένο κόστος χρήσης.** Η MySQL πρόκειται για το πιο διαδεδομένο open source σύστημα διαχείρισης βάσεων δεδομένων που επιτρέπει την διαχείριση σχεσιακών βάσεων (Relational Database).
2. **Φορητότητα.** Η MySQL είναι ένας cross-platform database server. Μπορεί να τρέξει σε διαφορετικές πλατφόρμες όπως Windows, Linux, Solaris, κλπ. Είναι μια ιδανική επιλογή για αυτούς που το project τους στοχεύει σε πολλαπλά συστήματα, πιο συγκεκριμένα μια διαδικτυακή εφαρμογή.
3. **Αδιάλειπτη συνδεσιμότητα.** Υπάρχουν πολλοί ασφαλείς και αδιάλειπτοι μηχανισμοί συνδεσιμότητας που είναι διαθέσιμοι για να συνδεθούμε με έναν MySQL server. Αυτές οι διασυνδέσεις περιλαμβάνουν ονομασμένους σωλήνες (named pipes), TCP/IP sockets και Unix Sockets.
4. **Ασφάλεια δεδομένων.** Η MySQL είναι παγκοσμίως αναγνωρισμένη ως το πιο ασφαλές και αξιόπιστο σύστημα διαχείρισης βάσεων δεδομένων που χρησιμοποιείται σε δημοφιλείς web εφαρμογές όπως το WordPress, Drupal, Joomla, Facebook και το Twitter. Τα δεδομένα προστατεύονται με κωδικούς και το θετικό με αυτούς τους κωδικούς είναι ότι αποθηκεύονται με κρυπτογραφημένη μορφή και είναι αρκετά δύσκολο να σπάσουν οι περίπλοκοι αυτοί αλγόριθμοι κρυπτογράφησης.

2.1.3 Admin Panel

Ένα Admin Panel(διαχειριστικός πίνακας ή πίνακας ελέγχου) είναι ένα σύστημα που επιτρέπει στους admins(διαχειριστές) μιας ιστοσελίδας ή μιας εφαρμογής να διεξάγουν διάφορες διεργασίες όπως η παρακολούθηση, η συντήρηση, η διατήρηση και ο έλεγχος διαφόρων διαδικασιών και δεδομένων της σελίδας ή εφαρμογής. Είναι ένα περιβάλλον όπου μπορούμε να ελέγχουμε όλες τις ρυθμίσεις και το περιεχόμενο της σελίδας μας και να βλέπουμε την κατάστασή της ώστε να προβούμε σε αντίστοιχες ενέργειες.

Στο Admin Panel της εφαρμογής PaPel, οι admins έχουν τη δυνατότητα να προσθέσουν και να αφαιρέσουν μαθήματα, να προβάλλουν την λίστα όλων των εγγεγραμμένων φοιτητών μαζί με τις αναλυτικές βαθμολογίες τους σε κάθε μάθημα, αλλά και τα στοιχεία εγγραφής τους. Επίσης, μέσω του Panel μπορούν να προβάλλουν τις σημειώσεις των φοιτητών μέσα στην εφαρμογή PaPel, και μόνο με τη δική τους έγκριση μπορούν οι σημειώσεις να “ανέβουν” και να είναι ορατές σε όλους τους χρήστες. Τέλος, οι admins έχουν την ικανότητα να προσθέσουν και να αφαιρέσουν users.

2.1.4 Διαδικτυακή Εφαρμογή

Με τον όρο διαδικτυακή εφαρμογή, αναφερόμαστε στην διεπαφή αλλά και λογική ενός προγράμματος, το οποίο διανέμεται εντός ενός περιηγητή/browser, χωρίς να απαιτείται κάποια εγκατάσταση από τον χρήστη της. Η έννοια αυτή διαφοροποιείται από αυτή μίας απλά δυναμικής σελίδας, καθώς ο στόχος τους διαφέρει. Η δυναμική σελίδα έχει ο στόχο απλά την καλύτερη εμπειρία σε μία σελίδα για ένα χρήστη, ενώ μία διαδικτυακή εφαρμογή, εξυπηρετεί και είναι χτισμένη με επίκεντρο μία λογική και προσφέρει στο χρήστη ένα σύνολο υπηρεσιών. Μία διαδικτυακή εφαρμογή, είναι ταυτόχρονα και δυναμική σελίδα, ενώ το ανάποδο δεν ισχύει πάντοτε.

Για την ανάπτυξη της συγκεκριμένης διαδικτυακής εφαρμογής χρησιμοποιήθηκαν οι παρακάτω τεχνολογίες:

1. **Php / Laravel Framework**
2. **HTML5**
3. **Css / Bootstrap**
4. **JavaScript / JQuery**

2.1.4.1 Php-Laravel Framework

Η **PHP** (PHP: Hypertext Preprocessor) είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που είτε θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML ή θα επεξεργαστεί τις εισόδους δίχως να προβάλλει την έξοδο στο χρήστη, αλλά θα τις μεταβιβάσει σε κάποιο άλλο PHP script.

Η PHP αποτελεί μια από τις πιο διαδεδομένες τεχνολογίες στο Παγκόσμιο Ιστό, καθώς χρησιμοποιείται από πληθώρα εφαρμογών και ιστοτόπων. Διάσημες εφαρμογές που κάνουν

εκτενή χρήση της PHP είναι το γνωστό Σύστημα Διαχείρισης Περιεχομένου (WordPress και το Drupal).

Το **Laravel** θεωρείται το καλύτερο PHP Framework αυτή τη στιγμή, από την κοινότητα των προγραμματιστών. Προσθέτει εγκυρότητα και υπεροχή στο σύνολο του έργου ενώ διευκολύνει την ανάπτυξη με τα ενσωματωμένα εργαλεία προγραμματιστών που διαθέτει καθώς και με την ευκολία εγκατάστασης σε σχέση με άλλα frameworks.

Παρακάτω θα αναλύσουμε τα βασικά πλεονεκτήματα της χρήσης του Laravel και τους λόγους που μας έκαναν να κινηθούμε με το συγκεκριμένο framework στην ανάπτυξη της εφαρμογής μας.

- 1. Αρχιτεκτονική MVC:** Το **Laravel** δουλεύει με την αρχιτεκτονική του **MVC**, δηλαδή Model — View — Controller, που εξασφαλίζει την καλύτερη διάκριση μεταξύ της λογικής της εφαρμογής και της παρουσίας της. Επιτρέπει τον διαχωρισμό της διαδικασίας ανάπτυξης και έτσι πετυχαίνει βελτιωμένη απόδοση σε κάθε τομέα. Έτσι, οι προγραμματιστές μπορούν να γράψουν «καθαρότερο» κώδικα και φυσικά να έχουν μία πιο εμπειριστατωμένη τεκμηρίωση του έργου.
- 2. Εργαλείο templating:** Το ενσωματωμένο templating εργαλείο ονομάζεται **Blade** και βοηθά στην δυναμική παρουσίαση του περιεχομένου της web εφαρμογής. Τα templates του Laravel φορτώνουν γρήγορα και μπορούν εύκολα να επεκταθούν μέσω μιας πληθώρας widgets που χρησιμοποιούν CSS και JS κώδικα.
- 3. Απλό και γρήγορο σύστημα ORM:** Το **ORM** ή **Object-Relational Mapping**, είναι μία βασική λειτουργία για την ανάπτυξη σύνθετων web εφαρμογών. Το Laravel έχει το γρηγορότερο και καλύτερο σύστημα ORM που ονομάζεται Eloquent. Περιλαμβάνει ένα Active Record pattern, που μας επιτρέπει να γράφουμε σύνθετα ερωτήματα SQL με απλή PHP σύνταξη. Ο κάθε πίνακας της βάσης δεδομένων, έχει τα ανάλογα μοντέλα για την διασύνδεση του με την εφαρμογή.
- 4. Γραμμή εντολών — Artisan Command Line:** Η γραμμή εντολών δημιουργεί και διαχειρίζεται το περιβάλλον του έργου. Απλοποιεί επαναλαμβανόμενες διαδικασίες και εργασίες που χρειάζεται να γίνουν manual. Το Laravel έχει ένα ενσωματωμένο εργαλείο που ονομάζεται Artisan για χρήση command line. Αυτό το εργαλείο βοηθά στη δημιουργία του λεγόμενου skeleton code καθώς και της αρχιτεκτονικής της βάσης δεδομένων μαζί με τα migrations. Αποτέλεσμα αυτού είναι η ευκολότερη διαχείριση της βάσης δεδομένων. Επίσης χρησιμοποιείται για την δημιουργία αρχείων MVC καθώς και τη δημιουργία προσαρμοσμένων εντολών.
- 5. Σύστημα Modular Packaging:** Η ανάπτυξη εφαρμογών σε Laravel βασίζεται στις 20 βιβλιοθήκες που χρησιμοποιούνται σε διαφορετικά modules. Αυτό το αρθρωτό σύστημα βοηθά στην ανάπτυξη responsive εφαρμογών. Βασίζεται πάνω στις πιο πρόσφατες αρχές της PHP για να ολοκληρώσει την διαδικασία ανάπτυξης σε έναν πιο λογικό χρόνο. Επίσης, μπορούμε να προσθέσουμε πάνω από 5,000 πακέτα για διάφορες χρήσεις.

6. **Σύστημα Migration:** Με το σύστημα **database migration**, το Laravel διευκολύνει την επέκταση της βάσης δεδομένων χωρίς να χρειάζεται η επαναδημιουργία της κάθε φορά. Προστατεύει τα σημαντικά δεδομένα ανάπτυξης και επιτρέπει αλλαγές στην σύνταξη PHP αντί για την χρήση της SQL. Έχει ενσωματωμένο το Laravel Schema Builder και μπορούμε να έχουμε σε version control την δημιουργία πινάκων με γρήγορο και αποτελεσματικό τρόπο.
7. **Ενσωματωμένες βιβλιοθήκες:** Το Laravel έχει ένα τεράστιο αριθμό προδιαμορφωμένων βιβλιοθηκών object-oriented για την εξομάλυνση και τον εξορθολογισμό της διαδικασίας ανάπτυξης. Η βιβλιοθήκη Authentication προσφέρει σημαντικές λειτουργίες όπως προστασία CSRF , Bcrypt hashing, κρυπτογράφηση κωδικών, επαναφορά κωδικών, κ.λ.π.
8. **Μοναδικό Micro-Framework:** Υπάρχει συνοδευτικά με το laravel ένα μοναδικό micro-framework που ονομάζεται Lumen. Το Lumen δουλεύει σύμφωνα με τις βασικές αρχές του Laravel αλλά εστιάζει σε πολύ γρήγορες εφαρμογές, ενώ συνήθως χρησιμοποιείται για δημιουργία υπηρεσιών που βασίζονται στο Laravel με πολύ πιο γρήγορα APIs.
9. **Εξαιρετικές λειτουργίες ασφάλειας:** Το Laravel είναι ένα framework που παρέχει μια αλάνθαστη ασφάλεια σε όλη τη διαδικασία ανάπτυξης αλλά και στο τελικό έργο. Προστατεύει την εφαρμογή από σοβαρές απειλές όπως CSR forgery, SQL injections και cross-site scripting. Το Laravel χρησιμοποιεί hashed και salted passwords που δεν μπορούν να αποθηκευτούν σε plain text στην βάση δεδομένων. Λειτουργεί με τον αλγόριθμο Bcrypt hashing που δημιουργεί κρυπτογραφημένα passwords. Επίσης, χρησιμοποιεί SQL statements για να αποτρέψει επιθέσεις.
10. **Απλοποιημένη διαδικασία Unit Testing:** Αυτό είναι το πιο θαυμαστό χαρακτηριστικό του Laravel. Προστατεύει από μη αναμενόμενες συμπεριφορές της εφαρμογής, τρέχοντας διάφορα test κάθε φορά. Έχει stable releases και μπορεί εύκολα να αναγνωρίσει γνωστά σφάλματα. Αυτή η λειτουργία του unit testing καταναλώνει αρκετό χρόνο ανάπτυξης, αλλά επιτρέπει την ορθή λειτουργία και την απαλοιφή των λαθών της εφαρμογής. Εκτός από αυτό, περιλαμβάνει και advanced helper modes για πιο περιγραφικό testing που δουλεύει σε πραγματικές καταστάσεις και συμπεριφορές πραγματικών χρηστών.

Τα συμπεράσματα που προκύπτουν είναι πως το Laravel προσφέρει μοναδικές και επεκτάσιμες λύσεις. Έχει μία τεράστια κοινότητα προγραμματιστών και είναι εξαιρετικά προσιτό οικονομικά. Για όλους τους παραπάνω λόγους επιλέξαμε το συγκεκριμένο framework για την ανάπτυξη της διαδικτυακής μας εφαρμογής λαμβάνοντας υπόψη τα πλεονεκτήματά του καθόλη τη διάρκεια του σχεδιασμού του project.

2.1.4.2 HTML5

Με τον όρο HTML[24] ορίζεται ένα σετ από συγκεκριμένα πρότυπα σήμανσης και συμπεριφορών που με τη σειρά τους ορίζουν το περιεχόμενο μιας διαδικτυακής σελίδας. HTML5 είναι η πέμπτη (5η) έκδοση της, όπου και θεωρείται σημαντική ενημέρωση όπου είχε ως σκοπό να υποστηρίξει της περαιτέρω ανάγκες των πολυμέσων στο διαδίκτυο αλλά και να εισάγει νέες δυνατότητες. Ταυτόχρονα, να κρατήσει τη μορφή της κατανοητή από τους υπολογιστές και τις συσκευές, χωρίς να περιπλέξει τα πράγματα για τους προγραμματιστές και να είναι συμβατή και με τις παλαιότερες εκδόσεις της.

2.1.4.3 Css/Bootstrap

CSS3 είναι η τρίτη (3η) έκδοση της γλώσσας μορφοποίησης CSS. Χρησιμοποιείται για την μορφοποίηση εγγράφου γραμμένου σε γλώσσες σήμανσης, όπως HTML. Είναι μία από τις γλώσσες «πυρήνας» του διαδικτύου και είναι πλέον το στάνταρ σε όλους τους browsers. Για την ακρίβεια, το CSS2.1 είναι αυτό που προτείνεται σαν χρήση και είναι ασφαλές για όλους τους browsers, ενώ το CSS3 τεμαχίζεται σταδιακά σε μικρότερα κομμάτια και περνάει την διαδικασία της ένταξης σαν το νέο στάνταρ. Κάθε browser ξεχωριστά, θα πρέπει να φροντίσει να μπορεί να τρέξει τα νέα δεδομένα που θέτει αυτή η τρίτη (3η) έκδοση της γλώσσας, πριν μπορέσει να θεωρηθεί το νέο standard.

Καθώς η ανάπτυξη τόσο διαδικτυακών εφαρμογών, όσο και δυναμικών ιστοσελίδων έχει γίνει αρκετά περίπλοκη και απαιτητική, βιβλιοθήκες μορφοποίησης όπως το Bootstrap είναι αυτές που έρχονται να δώσουν τη λύση σε τέτοιου είδους προβλήματα. Η βιβλιοθήκη αυτή είναι ανοιχτού κώδικα και είναι σχεδιασμένη να προσφέρει «responsive» σελίδες, γνωστές ως mobile-first. Με τον όρο αυτό νοείται η δυνατότητα—πλέον—του προγραμματιστή σχεδιάζοντας μία φορά την εφαρμογή να μπορεί να τρέξει ακόμα και σε κινητές συσκευές χωρίς να πρέπει να προβλέψει ξεχωριστά για κανόνες και σχεδιασμό. Η παραπάνω λύση προσφέρει χαρακτηριστικά όπως Typography, φόρμες, κουμπιά, μενού – πλοηγούς ενώ είναι βασισμένο σε δικές του αρχές σχεδίασης. Ένα από τα δυνατά χαρτιά και αυτής της βιβλιοθήκης, είναι τα Grid patterns που έχουν σχεδιάσει, που στην πραγματικότητα σου προσφέρουν μια γραμμή με δώδεκα (12) πάντα ίσες στήλες, ανεξάρτητα του διαθέσιμου χώρου. Το μαγικό του συγκεκριμένου συντακτικού, είναι ότι μπορείς για τον ίδιο χώρο να ορίσεις διαφορετικό αριθμό στηλών ανά συσκευή. Επομένως, ανάλογα τη συσκευή να διαμορφώνεται και ο χώρος στο επιθυμητό μέγεθος. Έτσι, για παράδειγμα, ένα χώρος που στην οθόνη του υπολογιστή θέλουμε να χωριστεί σε δώδεκα (12) ίσες στήλες, μπορούμε ταυτόχρονα να δηλώσουμε ότι θέλουμε σε tablet συσκευή αυτές να γίνουν δύο (2) ενώ σε κινητή συσκευή να υπάρχει μόνο μία (1). Αυτό χάρις της βιβλιοθήκης αυτής, επιτυγχάνεται χωρίς πολύ κόπο.

2.1.4.4 JavaScript/JQuery

Η JavaScript συχνά αναφερόμενη και ως JS είναι μία scripting γλώσσα προγραμματισμού που μαζί με την HTML και τη CSS αποτελούν τις τεχνολογίες «πυρήνες» του παγκόσμιου ιστού. Η συγκεκριμένη γλώσσα δίνει τη δυνατότητα προσθήκης διαδραστικότητας σε μία

σελίδα, ενώ βιβλιοθήκες για την πιο απλή και κατανοητή σύνταξη της, έκαναν πολύ γρήγορα την εμφάνιση τους. Μία εξ' αυτών, η jQuery η οποία είχε ως στόχο την απλοποίηση κανόνων ή ακόμα και τον συνδυασμό πολλαπλών κανόνων, δημιουργώντας ένα νέο σετ κανόνων πιο κατανοητό για τον τελικό χρήστη. Η επανάσταση που έφεραν βιβλιοθήκες σαν αυτή, ώθησαν όλο και περισσότερο κόσμο να ασχοληθεί με τη χρήση τους, όπου ήταν και ένας από τους κύριους παράγοντες εμφάνισης των Web Applications.

2.1.5 Progressive Web App

Η μετατροπή της εφαρμογής μας σε PWA(Progressive Web App) δηλαδή να μπορεί είναι λειτουργική σε desktop και mobile συσκευές έγινε με την προσθήκη δύο (2) αρχείων στον υπάρχων κώδικα. Τα αρχεία αυτά βρίσκονται στο root directory του project μας και είναι τα εξής: manifest.json και service-worker.js .

- **manifest.json:** Το manifest.json είναι ένα απλό JSON αρχείο όπου “ενημερώνει” τον browser σχετικά με την εφαρμογή και πώς αυτή πρέπει να συμπεριφέρεται όταν είναι εγκατεστημένη στο desktop, στο κινητό ή στο tablet μας. Ένα τυπικό manifest αρχείο περιλαμβάνει, μαζί με άλλα, το όνομα της εφαρμογής, τα εικονίδια που θα χρησιμοποιεί η εφαρμογή(π.χ. για χρήση thumbnail), και το URL που θα ανοίγει όταν θα γίνεται η χρήση του app. Έχοντας ένα manifest αρχείο είναι απαιτούμενο από τον browser ώστε να δείξει το χαρακτηριστικό εικονίδιο για το “κατέβασμα” της εφαρμογής μας.



- **service-worker.js:** Τα service-worker αρχεία είναι ειδικά JavaScript αρχεία τα οποία λειτουργούν ως proxies μεταξύ των web browsers και των web servers. Στην ουσία, είναι scripts κώδικα τα οποία εκτελούνται στο background, σε ξεχωριστά threads από το UI του browser. Τα service-worker αρχεία κάνουν εφικτό το να λειτουργεί ένα website όσο είναι offline. Είναι το εργοστάσιο παραγωγής που αναβαθμίζει μια ιστοσελίδα σε Progressive Web App. Χωρίς το service-worker αρχείο, μία ιστοσελίδα είναι απλά μια ιστοσελίδα, προσθέτοντάς το έχουμε πλέον μία εφαρμογή διαθέσιμη για download και εγκατάσταση.

2.2 Μέθοδος Ανάπτυξης

Με γνώμονα την παραπάνω δομή και ανάλυση κάθε ενότητας αλλά και της ανάγκες που έχει ως στόχο να καλύψει η εν λόγω πτυχιακή εργασία, έγιναν οι απαραίτητες επιλογές στις τεχνολογίες που θεωρούμε ότι θα προσφέρουν την κατάλληλη ευελιξία για την πιο άμεση και σωστή επίλυση των προβλημάτων.

Ο κεντρικός server ο οποίος παρέχεται από την εταιρία CretaForce, και καλύπτει πλήρως τις ανάγκες μας με τις δυνατότητές του για τις διεργασίες που απαιτούνται από την διαδικτυακή εφαρμογή, μας δίνει τη δυνατότητα αποθήκευσης και ανάκτησης δεδομένων από μία βάση MySQL.

Η διαδικτυακή εφαρμογή αναπτύχθηκε με τη χρήση της γλώσσας PHP και πιο συγκεκριμένα το framework Laravel, την HTML5 αλλά και CSS3. Παράλληλα η βιβλιοθήκη jQuery, βοηθάει στην σωστή προσέγγιση προβλημάτων, όπως responsiveness, ασύγχρονες επικοινωνίες, ασφαλείς μεταφορές δεδομένων, user experience και ταχύτητας.

Τέλος, η μετατροπή της εφαρμογής σε Progressive Web App(PWA), έγινε με την απλή προσθήκη δύο(2) βασικών αρχείων στον υπάρχον κώδικά μας: του manifest.json και server-worker.js.

3. Σχέδιο Δράσης

Σύμφωνα με την παραπάνω έρευνα και παράλληλα με τις ανάγκες της πτυχιακής εργασίας αλλά και πάντα με γνώμονες την ταχύτητα και εμπειρία του χρήστη διαμορφώθηκε το τεχνολογικό stack που ακολουθήσαμε. Οι επιλογές αυτές δεν ήταν μονόδρομος, αντίθετα, οι συνδυασμοί για την συγκεκριμένη υλοποίηση θα μπορούσαν να είναι πολλοί, αλλά η τελική επιλογή και διαχωρισμός βασίστηκε επίσης στις τάσεις της εποχής αλλά και τη ζήτηση τεχνολογιών στην αγορά εργασίας.

3.1 Σχεδιασμός και Υλοποίηση συστήματος

Καθώς ένα μεγάλο κομμάτι της λύσης αυτής βασίζεται στην συνεχή επικοινωνία του χρήστη με τον server, η επιλογή του σωστού εξυπηρετητή είναι πολύ σημαντική καθώς αυτός θα είναι υπεύθυνος για όλο το “heavy lifting”. Πολύ σημαντική είναι επίσης και η δομή της βάσης (database) καθώς μία σωστά δομημένη και γεμάτη εξαρτήσεις βάση θα γλυτώσει αρκετούς μπελάδες στην ανάκτηση περίπλοκων και πολλαπλά συνδεδεμένων αποτελεσμάτων.

Από την πλευρά του χρήστη η δουλειά δεν είναι πιο απλή. Αντίθετα, περιπλέκει καθώς θα πρέπει η εφαρμογή και το Progressive Web App να επιστρέφουν στον χρήστη μία συνολική εμπειρία, διαισθητική και πολύ γρήγορη σε πλοήγηση.

3.1.1 Υλοποίηση και σχεδιασμός της Βάσης Δεδομένων

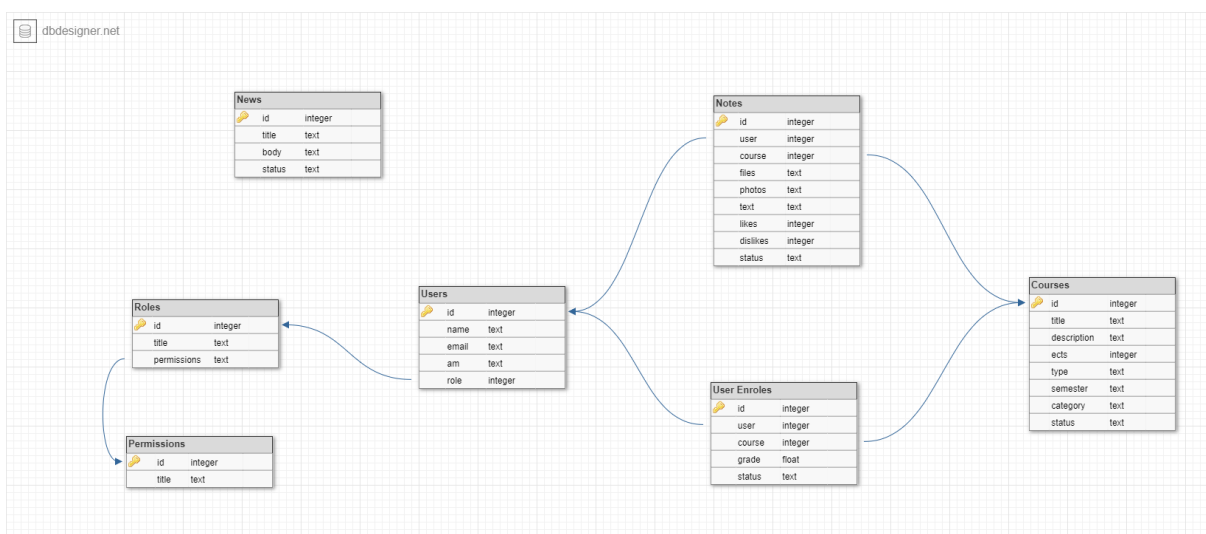
Επιλέγοντας την MySQL, μία σχεσιακή μορφή βάσης, μας δίνεται η δυνατότητα να απομονώσουμε τις οντότητες μας για ξεκάθαρη οργάνωση και παράλληλα να αναπτύξουμε δεσμούς μεταξύ των πινάκων για εύκολη ανάκτηση όλων των δεδομένων. Οι οντότητες της συγκεκριμένης υλοποίησης είναι οι εξής: users, roles, permissions, User Enrolls, Courses, Notes, News. Στον πίνακα users θα αποθηκευτούν όλες οι πληροφορίες που είναι σχετικές με τον χρήστη μας και μας είναι απαραίτητες για τη σωστή και αξιόλογη εμπειρία του στη εφαρμογή μας.

Η τελική μορφή των πινάκων καθώς και των data types που χρησιμοποιήθηκαν για την εκάστοτε τιμή φαίνονται παρακάτω στο σχεσιακό διάγραμμα που ακολουθεί, ενώ μπορείτε

επίσης να δείτε τις συσχετίσεις που έχουν αναπτυχθεί μεταξύ των πινάκων που είναι απαραίτητο, για την σωστή σύνδεση/ανάκτηση δεδομένων.

3.1.1.1 Database Schema-Entity Relationship (ER) Diagram

Ένα database schema ορίζει το πώς οργανώνονται τα δεδομένα μέσα σε μια σχεσιακή βάση δεδομένων. Περιλαμβάνει τα ονόματα των πινάκων, τα πεδία τους, τους τύπους των δεδομένων και τις σχέσεις μεταξύ αυτών των οντοτήτων. Παρακάτω παρουσιάζεται το μοντέλο οντοτήτων της εφαρμογής Pa-Pel.

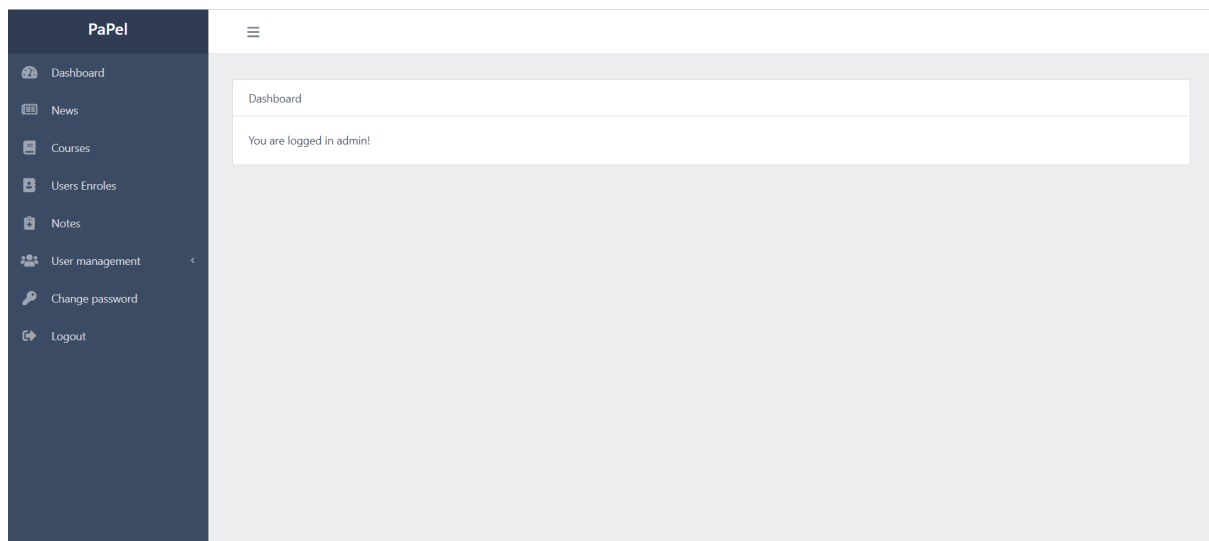


- **Users:** Η οντότητα Users αντιπροσωπεύει τον κάθε χρήστη που είναι εγγεγραμμένος στην εφαρμογή. Σχετίζεται με τις οντότητες Roles, User Enrolls και Notes.
- **Roles:** Η οντότητα Roles αντιπροσωπεύει τον κάθε ρόλο που μπορεί να έχει ο User. Οι ρόλοι που μπορεί να λάβει ένας User είναι admin και student. Σχετίζεται με τις οντότητες Users και Permissions.
- **Permissions:** Η οντότητα Permissions αντιπροσωπεύει τα δικαιώματα που έχει ο κάθε User ανάλογα τον ρόλο του. Σχετίζεται με την οντότητα Roles.
- **User Enrolls:** Η οντότητα User Enrolls αντιπροσωπεύει την κάθε εγγραφή ενός User(student) σε ένα μάθημα(course). Σχετίζεται με τις οντότητες Users και Courses.
- **Courses:** Η οντότητα Courses αντιπροσωπεύει τα μαθήματα που μπορεί ένας User(student) να κάνει εγγραφή. Σχετίζεται με τις οντότητες User Enrolls και Notes.

- **Notes:** Η οντότητα Notes αντιπροσωπεύει τις σημειώσεις που μπορεί να αφήσει ένας User σε ένα μάθημα. Σχετίζεται με τις οντότητες Users και Courses.
- **News:** Η οντότητα News αντιπροσωπεύει τα άρθρα σχετικά με τα νέα του τμήματος.

3.1.2 Υλοποίηση και σχεδιασμός Admin Panel

Για τη δημιουργία του admin dashboard του Pa-Pel χρησιμοποιήθηκε το QuickAdminPanel (<https://quickadminpanel.com>). Το QuickAdminPanel πρόκειται για έναν online generator που αποσκοπεί στη δημιουργία admin Panel της Laravel χρησιμοποιώντας CRUD builder. Κάνοντας εγγραφή στη σελίδα του QuickAdminPanel, είχαμε τη δυνατότητα να δημιουργήσουμε μενού και πεδία αλλά και τις συσχετίσεις online και έπειτα να κατεβάσουμε το όλο project και να το εγκαταστήσουμε στο σέρβερ μας.



3.1.2.1 Admin Panel - Dashboard

Το Tab Dashboard είναι ένα από τα βασικά συστατικά στοιχεία ενός διαχειριστικού πίνακα. Δίνει στους χρήστες άμεση πρόσβαση σε δεδομένα, reports, και στατιστικά που προκύπτουν από την εφαρμογή μας.

3.1.2.2 Admin Panel - News

Το Tab News περιέχει όλα τα άρθρα που συμπεριλαμβάνονται στο News Feed του Pa-Pel. Η καρτέλα αυτή περιέχει τις στήλες με: το ID που αντιστοιχεί στη βάση δεδομένων, τον τίτλο του άρθρου, το Body που περιέχει το κείμενο του άρθρου και το Status που μπορεί να είναι είτε Draft ή Publish. Μέχρι στιγμής, έχουμε πραγματοποιήσει τη συγκεκριμένη υλοποίηση όπου ο User μπορεί να συντάσσει, να τροποποιεί και να διαγράφει τα άρθρα μέσω του admin panel. Ιδανικά θα θέλαμε να εξελίξουμε περεταίρω αυτή την υλοποίηση, κάνοντας

απευθείας διασύνδεση με το API του τμήματος , ώστε να προσθέτει τα άρθρα με τις ενημερώσεις του Πανεπιστημίου με δυναμικό τρόπο.

ID	Title	Body	Status
3	Διαδραστικό Σεμινάριο για την ανάπτυξη εφαρμογών Python με WSL2, Hadoop, Yarn – Spark – Σάββατο 7 Μαΐου	Το Τμήμα ΗΜΜΥ, σε συνεργασία με το IEEE Student Branch του Πανεπιστημίου Πελοποννήσου, διοργανώνει διαδικτυακό σεμινάριο με θέμα «Ενιαίο περιβάλλον ανάπτυξης εφαρμογών Python σε Pseudo Cluster με την χρήση των τεχνολογιών WSL2, Hadoop, Yarn – Spark», το Σάββατο 7 Μαΐου (15:00 – 18:00). Με πρωτοβουλία του μέλους ΕΤΕΠ του Τμήματος, κ. Ευάγγελου Περάνη – ο οποίος και θα είναι ο εισηγητής του σεμιναρίου – και υπό την αιγίδα του φοιτητικού παραρτήματος ΙΕΕΕ, που στόχο έχει το να προσεγγίσουν οι φοιτητές θέματα που αφορούν στην ευρύτερη επιστήμη του Ηλεκτρολόγου Μηχανικού, προσκαλούνται όλοι οι φοιτητές να εκδηλώσουν ενδιαφέρον παρακολούθησης ενός εξαιρετικά χρήσιμου σεμιναρίου σχετικά με τις τεχνολογίες Big Data και την ανάπτυξη αντίστοιχων εφαρμογών, με χρήση της γλώσσας προγραμματισμού Python. Οι συμμετέχοντες φοιτητές θα έχουν τη δυνατότητα να παρακολουθήσουν και να εξασκηθούν πρακτικά στη δημιουργία ενός ενιαίου περιβάλλοντος ανάπτυξης, με τη χρήση τεχνολογιών και εργαλείων, που απαιτούν μικρούς υπολογιστικούς πόρους. Το σεμινάριο θα πραγματοποιηθεί μέσω Microsoft Teams και είναι ανοικτό σε κάθε ενδιαφερόμενο φοιτητή του Τμήματος, σε οποιοδήποτε στάδιο σπουδών. Αναλυτικές πληροφορίες για το περιεχόμενο του σεμιναρίου, θα βρείτε εδώ	Publish
2	13ο Συνέδριο ΣΦΗΜΜΥ: 15 – 17 Απριλίου στην Πάτρα	Το Συνέδριο – θεαμός του Συλλόγου Φοιτητών Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών (ΣΦΗΜΜΥ) διοργανώνεται για 13η χρονιά και θα πραγματοποιηθεί το τμήμερο 15 – 17 Απριλίου στην Πάτρα, στο Συνεδριακό Κέντρο του Πανεπιστημίου Πατρών. Με θέμα, φέτος, τη διδύμη μετάβαση (twin transition), δηλαδή την πρόσηνη μετάβαση και τον ψηφιακό μετασχηματισμό, το συνέδριο το οποίο στοχεύει στη διερεύνηση των διαφόρων τεχνολογιών Ηλεκτρολογίας και Υπολογιστών που διαμορφώνουν ένα ανταγωνιστικό και αξιόπιστο οικοσύστημα, έχει ένα πλούσιο πρόγραμμα από ομιλίες, παρουσιάσεις papers και workshops, που προσφέρουν στους φοιτητές Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών το υπόβαθρο για παγκόσμια πρωτοπορία. Μεταξύ των 32	Publish

3.1.2.3 Admin Panel - Courses

Το Tab Courses περιέχει όλα τα διαθέσιμα μαθήματα του τμήματος. Στην καρτέλα αυτή τα μαθήματα που έχουν προστεθεί από τον admin περιέχουν : το ID που αντιστοιχεί στη βάση δεδομένων, έναν Τίτλο, μια Περιγραφή του μαθήματος, τις διδακτικές μονάδες που του αντιστοιχεί, τον τύπο του μαθήματος δηλαδή εάν είναι μάθημα Θεωρίας ή Εργαστηρίου, το εξάμηνο στο οποίο διδάσκεται, την κατηγορία του μαθήματος (.....) και το status του μαθήματος το οποίο μπορεί να είναι Active ή Inactive. Όλα τα μαθήματα μπορούν να επεξεργαστούν και να διαγραφούν από τον admin.

ID	Title	Description	Credits	Type	Category	Level	Status
18	Διαδικασιακός Προγραμματισμός (Θ)	Βασικός στόχος του μαθήματος είναι η απόκτηση γνώσεων και ικανότητας επίλυσης διαφόρων υπολογιστικών προβλημάτων με διαδικασιακό προγραμματισμό και ικανότητα χρήσης της γλώσσας C, η οποία αποτελεί τη βάση για αρκετές από τις γλώσσες προγραμματισμού που θα διδαχθούν στα επόμενα εξάμηνα (όπως Java, Javascript, PHP, κιά).	2.5	Θεωρία	B	Γενικής	Active
17	Δομές Δεδομένων και Αλγόριθμοι (Ε)	Στόχος του μαθήματος είναι η εισαγωγή των φοιτητών στις βασικές έννοιες των Δομών Δεδομένων και Αλγορίθμων με σκοπό αφενός να αποκτήσουν θεωρητικές και πρακτικές γνώσεις που θα επιτρέψουν την αποδοτικότερη αναπαράσταση των δεδομένων οδηγώντας σε προγράμματα υψηλού επιπέδου ποιότητας και τεκμηρίωσης και αφετέρου να αναπτύξουν ελεύθερη, δημιουργική και επαγγελματική σκέψη.	2.5	Εργαστήριο	Γ	Γενικής	Active
16	Δομές Δεδομένων και Αλγόριθμοι (Θ)	Στόχος του μαθήματος είναι η εισαγωγή των φοιτητών στις βασικές έννοιες των Δομών Δεδομένων και Αλγορίθμων με σκοπό αφενός να αποκτήσουν θεωρητικές και πρακτικές γνώσεις που θα επιτρέψουν την αποδοτικότερη αναπαράσταση των δεδομένων οδηγώντας σε προγράμματα υψηλού επιπέδου ποιότητας και τεκμηρίωσης και αφετέρου να αναπτύξουν ελεύθερη, δημιουργική και επαγγελματική σκέψη.	2.5	Θεωρία	Γ	Γενικής	Active
15	Πιθανοθεωρία και Στατιστική	Το μάθημα απευθύνεται σε δευτεροετείς φοιτητές του Τμήματος Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών και έχει ως κύριο στόχο του να τους εισάγει στις βασικές έννοιες και εφαρμογές της	5	Θεωρία	Γ	Γενικής	Active

3.1.2.4 Admin Panel - Users Enrolls

Το Tab User Enrolls περιέχει όλες τις εγγραφές των χρηστών που αντιστοιχούν με τα μαθήματα που παρακολουθεί. Η καρτέλα αυτή περιέχει τις στήλες με : το ID που αντιστοιχεί στη βάση δεδομένων, το όνομα του User(φοιτητή), το όνομα του μαθήματος, τη βαθμολογία του μαθήματος και το Status του μαθήματος που μπορεί να είναι Passed ή Not Passed.

The screenshot displays the 'Users Enrolls' section of the PaPel Admin Panel. It features a table with the following data:

ID	User	Course	Grade	Status	
64	SomeName	Μαθηματικός Λογισμός Ι	6	Passed	View Edit Delete
63	MARIANNA GRAMMATIKOPOULOU	Μαθηματικός Λογισμός Ι	10	Passed	View Edit Delete
62	emily	Ηλεκτροτεχνικά Υλικά	5	Passed	View Edit Delete
61	ΑΙΜΙΛΙΑ ΠΑΠΑΔΟΥΡΟΥ	Φυσική	10	Passed	View Edit Delete

Below the table, it indicates 'Showing 1 to 4 of 4 entries' and includes navigation buttons for 'Previous', '1', and 'Next'. The top of the table area has a search bar and several action buttons: 'Select all', 'Deselect all', 'Copy', 'CSV', 'Excel', 'PDF', 'Print', 'Columns', and 'Delete selected'.

3.1.2.5 Admin Panel - Notes

Το Tab Notes περιέχει τις σημειώσεις όλων των Users. Η καρτέλα αυτή περιέχει τις στήλες με : το ID που αντιστοιχεί στην βάση δεδομένων, το όνομα του User που έχει προσθέσει τη σημείωση, το μάθημα στο οποίο έχει προστεθεί η σημείωση, τα Files όπου προστίθεται η σημείωση εάν είναι σε μορφή pdf, τις Photo όπου προστίθεται η σημείωση εάν είναι σε μορφή εικόνας, τα Likes όπου περιέχει τον αριθμό των likes που έχει λάβει η συγκεκριμένη σημείωση από άλλους χρήστες, τα Dislikes που περιέχει τον αριθμό των dislikes που έχει λάβει η σημείωση από άλλους χρήστες και το Status όπου μπορεί να πάρει τις τιμές Active και Disabled. Να σημειωθεί πως η κάθε σημείωση λαμβάνει πάντα αρχικά το status Disabled έως ότου ο admin που θα είναι ο εκάστοτε καθηγητής να αξιολογήσει την καταλληλότητα του περιεχομένου της σημείωσης και να αλλάξει το status σε Active ώστε να μπορέσει η σημείωση να γίνει ορατή σε όλους τους χρήστες.

ID	User	Course	Files	Photo	Likes	Dislikes	Status
103	SomeName	Μαθηματικός Λογισμός Ι			7		Disabled
102	Takis	Τεχνικές Προγραμματισμού (Θ+Ε)					Disabled
100	emily	Μαθηματικός Λογισμός Ι					Active
98	Test Student	Μαθηματικός Λογισμός Ι	View file				Active
96	Admin	Μαθηματικός Λογισμός Ι					Disabled
95	Admin	Τεχνικές Προγραμματισμού (Θ+Ε)	View file				Disabled
93	Admin	Τεχνικές Προγραμματισμού (Θ+Ε)	View file				Disabled

3.1.2.6 Admin Panel - User Management

Το Tab User Management αποτελείται από τρία (3) tabs: Την καρτέλα **Permissions**, την καρτέλα **Roles** και την καρτέλα **Users**.

Permissions: Το tab Permissions περιέχει όλα τα δικαιώματα που μπορεί να έχει ένας User είτε είναι student είτε είναι admin. Ο admin έχει το δικαίωμα να προσθέσει, να επεξεργαστεί και να διαγράψει permissions.

ID	Title
37	profile_password_edit
36	news_access
35	news_delete
34	news_show
33	news_edit
32	news_create
31	note_access
30	note_delete
29	note_show
28	note_edit
27	note_create
26	user_create_admin

Roles: Το tab Roles περιέχει τους ρόλους που μπορεί να λάβει ένας User. Η καρτέλα αυτή περιέχει τις στήλες με: το ID που αντιστοιχεί στην βάση δεδομένων, τον τίτλο του ρόλου και τα δικαιώματα (Permissions) που αντιστοιχούν στον κάθε ρόλο. Αυτή τη στιγμή έχουμε δημιουργήσει δύο ρόλους: τον ρόλο Student και τον ρόλο Admin με τον Admin να έχει φυσικά όλα τα διαθέσιμα permissions στον ρόλο του.

The screenshot shows the 'Roles' management page in PaPel. The 'Role List' table is as follows:

ID	Title	Permissions	Actions
2	Student	course_show, course_access, users_enrole_create, users_enrole_edit, users_enrole_show, users_enrole_delete, users_enrole_access, note_create, note_edit, note_show, note_delete, note_access, news_create, news_show, news_access, profile_password_edit	View, Edit, Delete
1	Admin	user_management_access, permission_create, permission_edit, permission_show, permission_delete, permission_access, role_create, role_edit, role_show, role_delete, role_access, user_create, user_edit, user_show, user_delete, user_access, course_create, course_edit, course_show, course_delete, course_access, users_enrole_create, users_enrole_edit, users_enrole_show, users_enrole_delete, users_enrole_access, note_create, note_edit, note_show, note_delete, note_access, news_create, news_edit, news_show, news_delete, news_access, profile_password_edit	View, Edit, Delete

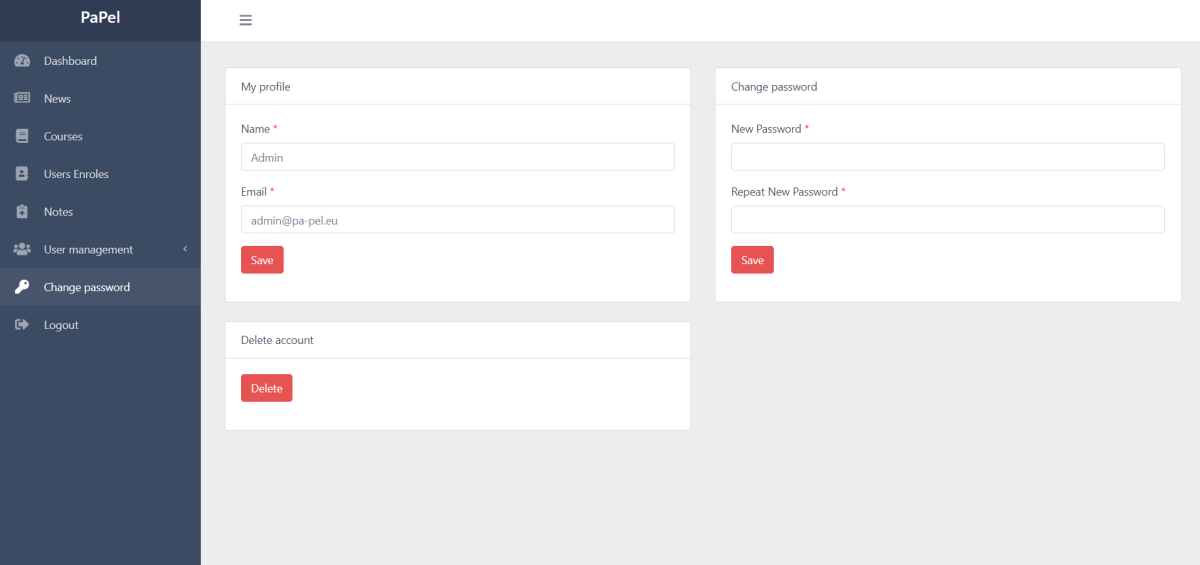
Users: Το tab Users περιέχει όλους τους Users που είναι εγγεγραμμένοι στο σύστημά μας. Η καρτέλα αυτή περιέχει τις στήλες με: το ID που αντιστοιχεί στην βάση δεδομένων, το όνομα του εγγεγραμμένου χρήστη, τον αριθμό μητρώου του, το email με το οποίο έχει κάνει εγγραφή, την ημερομηνία με το πότε έγινε Verified το email του λογαριασμού του και τον ρόλο του χρήστη. Ένας Admin User μπορεί να προσθέσει, να προβάλλει και να επεξεργαστεί τα στοιχεία ενός User αλλά και να διαγράψει όποιον User επιθυμεί.

The screenshot shows the 'Users' management page in PaPel. The 'User List' table is as follows:

ID	Name	Am	Email	Email verified at	Roles	Actions
18	Takis		alefrag@gmail.com		Admin	View, Edit, Delete
17	MARIANNA GRAMMATIKOPOULOU	2675	student@tel.eu		Student	View, Edit, Delete
16	AIMINIA ΠΑΠΑΔΑΟΥΡΟΥ	2673	emily13s1997@hotmail.com		Student	View, Edit, Delete
2	Test Student		student@pa-pel.eu		Student	View, Edit, Delete
1	Admin		admin@pa-pel.eu		Admin	View, Edit, Delete

3.1.2.7 Admin Panel - Change Password

Το tab Change Password επιτρέπει στον Admin User να επεξεργαστεί τα προσωπικά του στοιχεία όπως το Όνομα και το Email του αλλά και να πραγματοποιήσει αλλαγή του προσωπικού του κωδικού(Password). Ο Admin User έχει επίσης την δυνατότητα να διαγράψει τον λογαριασμό του με το πάτημα ενός κουμπιού.



The screenshot displays the PaPel Admin Panel interface. On the left is a dark blue sidebar with a menu containing: Dashboard, News, Courses, Users Enroles, Notes, User management, Change password (highlighted), and Logout. The main content area is light gray and contains three white panels. The top-left panel, titled 'My profile', has input fields for 'Name' (containing 'Admin') and 'Email' (containing 'admin@pa-pel.eu'), with a red 'Save' button below. The top-right panel, titled 'Change password', has input fields for 'New Password' and 'Repeat New Password', with a red 'Save' button below. The bottom panel, titled 'Delete account', has a red 'Delete' button.

3.1.3 Υλοποίηση και σχεδιασμός Web Application

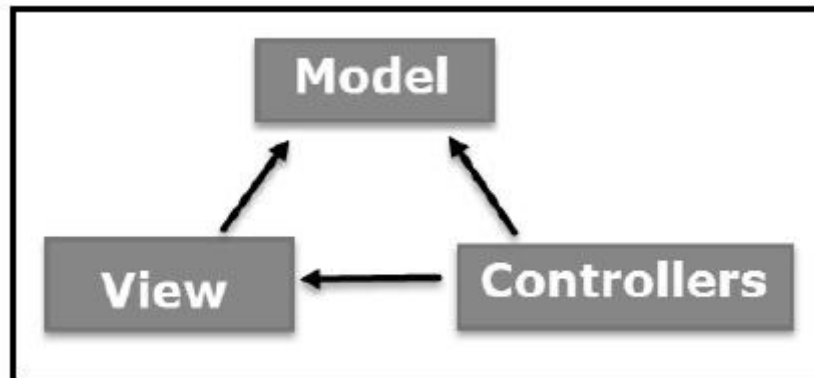
Η διαδικτυακή εφαρμογή αναπτύχθηκε κυρίως με τη χρήση του Laravel Framework. Οι λόγοι που προτιμήθηκε η συγκεκριμένη τεχνολογία όπως εξηγήθηκε και παραπάνω είναι κυρίως η ευκολία, η απλότητα στη χρήση του αλλά και η μεγάλη ταχύτητά του, πράγμα πολύ ουσιαστικό για τον τελικό χρήστη. Εκτός από όλα αυτά, το Laravel είναι ένα κοινώς διαδεδομένο framework στον χώρο του προγραμματισμού, καθιστώντας το το πλήρως κατάλληλο για την εφαρμογή μας. Τα βασικά κομμάτια που θα αναλύσουμε στη συγκεκριμένη ενότητα αναφέρονται στην αρχιτεκτονική του Laravel framework, δηλαδή θα αναλύσουμε το μοντέλο στο οποίο βασίζεται και θα περιγράψουμε αναλυτικά το κάθε κομμάτι της εφαρμογής μας.

3.1.3.1 Το μοντέλο MVC(Model-View-Controller)

Το Laravel Framework ακολουθεί την αρχιτεκτονική του MVC(Model,View,Controller). Το Model-View-Controller (MVC) είναι ένα αρχιτεκτονικό μοντέλο που διαχωρίζει την εφαρμογή σε τρία (3) λογικά μέρη: το model, το view και τον controller. Κάθε ένα από τα στοιχεία αυτά, έχουν δημιουργηθεί για να χειρίζονται διαφορετικά στοιχεία του development μιας εφαρμογής. Το μοντέλο MVC είναι ένα από τα πιο χρησιμοποιούμενα frameworks για την δημιουργία κλιμακούμενων και επεκτάσιμων πρότζεκτ.

3.1.3.2 Στοιχεία του μοντέλου MVC

Στη συνέχεια ακολουθούν τα συστατικά μέρη του μοντέλου MVC. Το κάθε ένα θα αναπτυχθεί σε δική του υποενότητα.



3.1.3.2.1 Routes

Στο μοντέλο MVC, το routing(δρομολόγηση) είναι η διαδικασία αντιστοίχισης ενός browser request σε μία ενέργεια ενός controller και η επιστροφή ενός response από αυτόν. Το routing με λίγα λόγια, είναι το να λαμβάνεις ένα HTTP request και να το μετατρέπεις σε ένα route(μονοπάτι) το οποίο χρειάζεται να ακολουθήσουμε για να φτάσουμε σε ένα endpoint όπου εννοούμε το κομμάτι του κώδικα που χειρίζεται το συγκεκριμένο HTTP request. Με αυτόν τον τρόπο ορίζεται ποιός controller και ποιες ενέργειες εκτελούνται με βάση το request.

```
Route::get('user/profile', 'UserProfileController@show')->name('profile');
```

Η δομή ενός route είναι η εξής:

1. το get απευθύνεται στην HTTP μέθοδο. Μπορεί να πάρει τις τιμές GET,POST,PUT,κλπ.
2. το κάθε route δέχεται δύο (2) παραμέτρους:
 - a. το 'user/profile' απευθύνεται στο unique endpoint που βλέπει ο τελικός χρήστης.
 - b. το 'UserProfileController' αναφέρεται στον controller και το 'show' στη συνάρτηση που περιέχει ο controller και θα εκτελέσει το request μας.
3. το name είναι ένα unique όνομα που αντιπροσωπεύει το route. Αυτό μας βοηθάει στο να καλέσουμε το route μόνο με το name του.

Στην εφαρμογή μας όλα τα routes βρίσκονται στον ομώνυμο φάκελο. Μέσα στον φάκελο αυτό βρίσκονται τέσσερα(4) αρχεία: το api.php, το channels.php, το console.php και το web.php .

- 1. api.php:** Το αρχείο api.php περιέχει όλα τα routes τα οποία είναι stateless, δηλαδή όταν ο σέρβερ δεν αποθηκεύει καμία κατάσταση κατά τη διάρκεια ενός client session. Τα routes που βρίσκονται στο αρχείο api.php επικοινωνούν με το api middleware. (Middleware ονομάζεται το λογισμικό που παρέχει υπηρεσίες και ιδιότητες σε εφαρμογές, πέρα από από αυτές που παρέχονται από το λειτουργικό σύστημα.)
- 2. channels.php:** Στο αρχείο channels.php μπορούμε να δηλώνουμε την εξουσιοδότηση των χρηστών στα broadcasting channels. Το κύριο κόνσεπτ πίσω από το broadcasting είναι απλό: Οι clients (οι clients διευκολύνουν τα εξερχόμενα HTTP requests χωρίς να χρειάζονται επιπλέον περιττό κώδικα), συνδέονται σε συγκεκριμένα κανάλια στο frontend, ενώ η εφαρμογή μας εκπέμπει events σε αυτά τα κανάλια στο backend.
- 3. console.php:** Στο αρχείο console.php μπορούμε να ορίσουμε απλές Closure Based εντολές για την γραμμή εντολών. Οι Closure Based εντολές ορίζονται χρησιμοποιώντας την Artisan::command(). Artisan ονομάζεται το interface της γραμμής εντολών που συμπεριλαμβάνει η Laravel. Η Artisan υπάρχει στο root folder της εφαρμογής μας και παρέχει έναν αριθμό από χρήσιμες εντολές που μπορούν να μας βοηθήσουν να χτίσουμε την εφαρμογή μας.
- 4. web.php:** Το αρχείο web.php περιέχει όλα τα routes που σχετίζονται με το web περιβάλλον της εφαρμογής μας. Αυτά τα routes επικοινωνούν με το web middleware και είναι τα routes που θα επισκέπτονται οι τελικοί χρήστες της εφαρμογής. Αυτή τη στιγμή στο αρχείο web.php έχουμε τα εξής routes:

```
/*
 * Enable Register page and functionalities
 */
Auth::routes(['register' => true]);
```

To `Auth::routes(['register => true'])` πρόκειται για το register route που ενεργοποιεί το user registration στην εφαρμογή μας.

```
/* App Routes */
/*
 * Public routes without being logged in
 */
Route::get('/', 'HomeController@index')->name('home');
Route::get('/blog', 'NewsController@index')->name('blog');
Route::get('/blog/post/{id}', 'NewsController@show')->name('blog-show');
```

Τα public routes τα οποία είναι προσβάσιμα από τους χρήστες χωρίς να έχουν κάνει register. Τα routes αυτά είναι: Η αρχική μας σελίδα, η σελίδα των blogs και τα blog posts.

```
/*
 * Routes that can be accessed while being logged in
 *
 * middleware checks if user is logged in or not
 */
Route::group(['middleware' => ['auth']], function () {
    Route::get('/my-account', 'AccountController@index')->name('my-account');

    Route::get('/courses', 'CoursesController@index')->name('courses-index');
    Route::get('/courses/course/{id}', 'CoursesController@show')->name('courses-show');
    Route::post('/courses/update/grade/{id}', 'CoursesController@updateCourseGrade')->name('courses-grade-update');
    // Route::post('/search/', 'CoursesController@search')->name('search');

    Route::get('/course/enroll/{id}', 'EnrolesController@create')->name('user-course-enroll');
    Route::get('/course/enroll-remove/{id}', 'EnrolesController@destroy')->name('user-course-enroll-remove');

    Route::post('/add-new-note', 'NotesController@store')->name('add-new-note');
    Route::get('/note/remove/{id}', 'NotesController@destroy')->name('note-remove');

    Route::get('/note/add/like/{id}', 'NotesController@addLike')->name('note-add-like');
    Route::get('/note/add/dislike/{id}', 'NotesController@addDislike')->name('note-add-dislike');
});
```

Τα παραπάνω routes βρίσκονται κάτω από το auth middleware, αυτό σημαίνει ότι μόνο οι logged in users έχουν πρόσβαση σε αυτά. Εδώ συμπεριλαμβάνονται οι σελίδες των courses ,των σημειώσεων, του προφίλ του user αλλά και όλες οι ενέργειες που μπορούν να εκτελέσουν οι users π.χ show/update course grade, store/destroy notes και add like/dislike.

```

/* App Admin Routes */
Route::group(['prefix' => 'admin', 'as' => 'admin.', 'namespace' => 'Admin', 'middleware' => ['auth']], function () {

    // Permissions
    Route::delete('permissions/destroy', 'PermissionsController@massDestroy')->name('permissions.massDestroy');
    Route::resource('permissions', 'PermissionsController');

    // Roles
    Route::delete('roles/destroy', 'RolesController@massDestroy')->name('roles.massDestroy');
    Route::resource('roles', 'RolesController');

    // Users
    Route::delete('users/destroy', 'UsersController@massDestroy')->name('users.massDestroy');
    Route::resource('users', 'UsersController');

    // Courses
    Route::delete('courses/destroy', 'CoursesController@massDestroy')->name('courses.massDestroy');
    Route::resource('courses', 'CoursesController');

    // Users Enroles
    Route::delete('users-enroles/destroy', 'UsersEnrolesController@massDestroy')->name('users-enroles.massDestroy');
    Route::resource('users-enroles', 'UsersEnrolesController');

    // Notes
    Route::delete('notes/destroy', 'NotesController@massDestroy')->name('notes.massDestroy');
    Route::post('notes/media', 'NotesController@storeMedia')->name('notes.storeMedia');
    Route::post('notes/ckmedia', 'NotesController@storeCKEditorImages')->name('notes.storeCKEditorImages');
    Route::resource('notes', 'NotesController');

    // News
    Route::delete('news/destroy', 'NewsController@massDestroy')->name('news.massDestroy');
    Route::resource('news', 'NewsController');

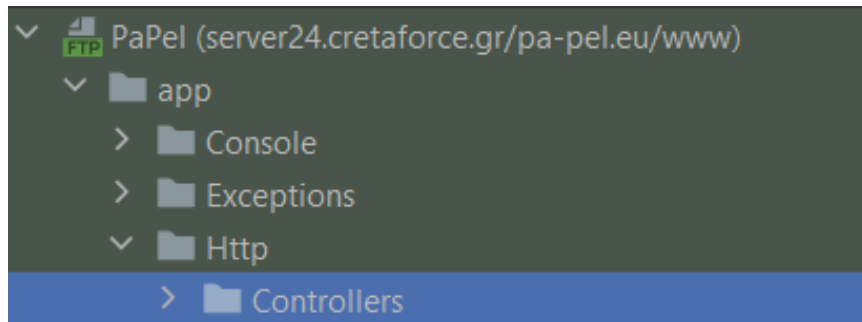
});

```

Τα παραπάνω routes βρίσκονται και αυτά κάτω από το auth middleware, όμως από την πλευρά του admin. Εδώ συμπεριλαμβάνονται οι ενέργειες που μπορεί να εφαρμόζει ο admin πάνω στην εφαρμογή. Μέσω του resource το οποίο μας παρέχει CRUD (Create, Read, Update, Delete) routes σε μια γραμμή κώδικα, ο admin μπορεί να δημιουργήσει, να διαβάσει, να ενημερώσει και να διαγράψει permissions, roles, users, μαθήματα, εγγραφές χρηστών σε μαθήματα, σημειώσεις, και blog posts. Τέλος, μέσω της συνάρτησης massDestroy ο admin έχει την ικανότητα να διαγράψει πολλαπλές εγγραφές με μία μόνο εντολή.

3.1.3.2.2 Controllers

Οι Controllers μπορούν να ομαδοποιήσουν συσχετιζόμενα requests μέσα σε μία κλάση. Για παράδειγμα, μία κλάση με τίτλο UserController μπορεί να διαχειρίζεται όλα τα εισερχόμενα requests που σχετίζονται με τους users, συμπεριλαμβάνοντας την προβολή, την δημιουργία, την ενημέρωση και την διαγραφή users. Από προεπιλογή, οι controllers αποθηκεύονται στον φάκελο που βρίσκεται στο directory **app/Http/Controllers**.

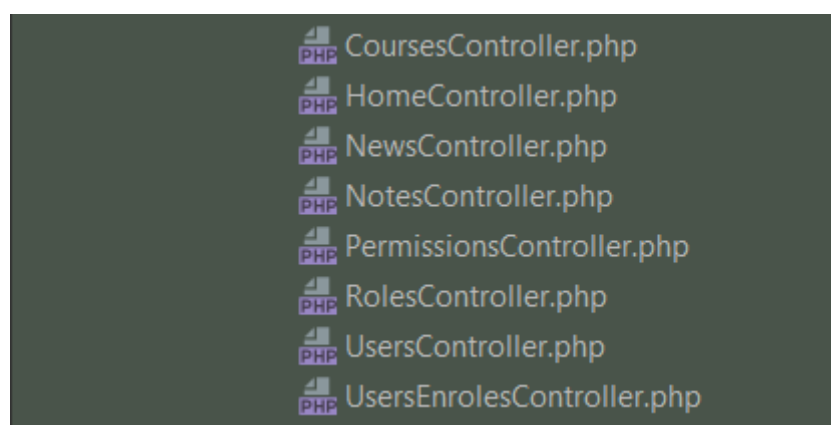


Μέσα στον φάκελο Controllers βρίσκονται οι controllers που σχετίζονται με τον Admin και οι controllers που σχετίζονται με τους υπόλοιπους χρήστες.

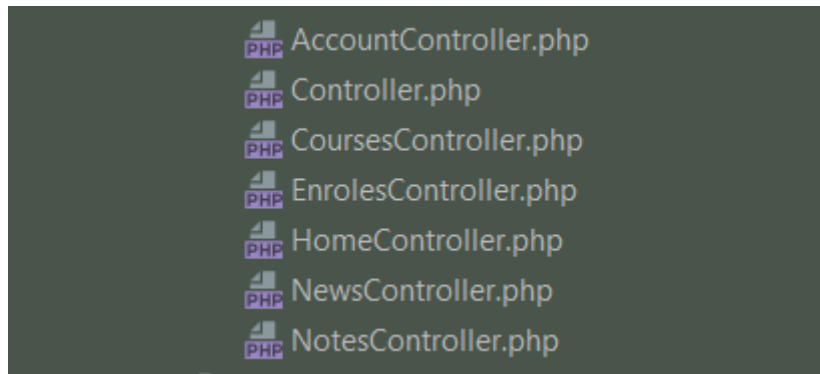
Admin controllers: Οι controllers που σχετίζονται με τον Admin είναι οι ακόλουθοι και εφαρμόζουν παρόμοιες λειτουργικότητες(CRUD) έπειτα από έλεγχο απαραίτητων δικαιωμάτων(permissions):

- **Index:** Συλλέγει και επιστρέφει αντικείμενα(που σχετίζονται με τον αντίστοιχο controller) σε μία λίστα.
- **Show:** Εμφανίζει ένα συγκεκριμένο αντικείμενο.
- **Create:** Εμφανίζει την κατάλληλη φόρμα για την εισαγωγή ενός νέου αντικειμένου.
- **Store:** Δημιουργεί ένα καινούργιο αντικείμενο(π.χ. course, note, κλπ).
- **Edit:** Εμφανίζει την κατάλληλη φόρμα για να κάνουμε τροποποίηση ενός αντικειμένου.
- **Update:** Ενημερώνει το αντικείμενο με τα νέα δεδομένα.
- **Destroy:** Διαγράφει ένα συγκεκριμένο αντικείμενο.
- **MassDestroy:** Διαγράφει πολλαπλά αντικείμενα.

Ορισμένοι controllers μπορεί να έχουν και άλλες συναρτήσεις κυρίως για την διαχείριση αρχείων(π.χ. NotesController).



User's controllers: Οι controllers που σχετίζονται με τους users είναι οι ακόλουθοι:



1. **AccountController.php:** Σχετίζεται με τον λογαριασμό του χρήστη. Περιέχει τη συνάρτηση `index()`, η οποία συγκεντρώνει δεδομένα που αφορούν τα περασμένα μαθήματα, τα μη περασμένα μαθήματα, τις σημειώσεις και τέλος, υπολογίζει το σύνολο των διδακτικών μονάδων. Τα δεδομένα αυτά αναθέτονται στις αντίστοιχες μεταβλητές οι οποίες μεταφέρουν την πληροφορία στο ανάλογο view.

```
class AccountController extends Controller
{
    public function index(){
        /*
         * Here we collect student's account data
         * We filter passed and not passed courses
         * Also, we retrieve the student's notes that are sorted by the likes number
         */
        $passedCourses = auth()->user()->enroles->where('status', 'passed') ?: null;
        $notPassedCourses = auth()->user()->enroles->where('status', 'not_passed') ?: null;
        $notes = auth()->user()->notes->sortByDesc('likes') ?: null;

        /*
         * Here we calculate the total ects value for passed courses.
         */
        $ectsTotal = 0;
        foreach ($passedCourses as $passedCourse) {
            $ectsTotal = $ectsTotal + $passedCourse->course->ects;
        }
    }
}
```

2. **CoursesController.php:** Σχετίζεται με τα διαθέσιμα μαθήματα. Περιέχει τις ακόλουθες συναρτήσεις:
 - a. **index():** Επιστρέφει το σύνολο των μαθημάτων έπειτα από εφαρμογή φίλτρων του χρήστη με βάση τον τύπο(type) μαθήματος ή/και το εξάμηνο(semester) που ανήκει. Τέλος, τα ομαδοποιεί με βάση το εξάμηνο σε αύξουσα σειρά. Κατά την επιστροφή των δεδομένων στο view, επιστρέφουμε

και το όνομα των φίλτρων που έχουν εφαρμοστεί από τον χρήστη.

```
public function index(){
    /*
     * We filter the courses with the given filters(type,semester)
     */
    $courses = Course::query()
        ->when(request('type'), function ($course){
            return $course->where('type', request('type'));
        })
        ->when(request('semester'), function ($course){
            return $course->where('semester', request('semester'));
        })
        ->get()
        ->groupBy('semester');

    /*
     * We return the view with the filtered courses and also the filters we applied
     */
    return view('courses.index', [
        'typeFilters' => request('type'),
        'semesterFilters' => request('semester'),
        'courses' => $courses
    ]);
}
```

- b. **show(\$courseId)**: Ελέγχει εάν ο τρέχων χρήστης είναι εγγεγραμμένος ή όχι σε ένα μάθημα. Αφού αναζητήσουμε όλες τις εγγραφές των χρηστών στα μαθήματα και ελέγξουμε εάν υπάρχει το id ενός συγκεκριμένου μαθήματος ή όχι σε αυτές, τότε θα εμφανίζουμε το ανάλογο κουμπί στον χρήστη : 'Εγγραφή' ή 'Απεγγραφή'.

```
public function show($courseId){

    $userEnroll = null;
    foreach (auth()->user()->enrols as $enroll) {
        if($enroll->course_id == $courseId) {
            $userEnroll = $enroll;
            break;
        }
    }

    $course = Course::find($courseId);

    return view('courses.show', [
        'userEnroll' => $userEnroll,
        'course' => $course,
        'notes' => $course->notes->where('status', 'active')->sortByDesc('likes')
    ]);
}
```

- c. **updateCourseGrade(\$enrollID)**: Αναζητά την εγγραφή ενός χρήστη στο μάθημα βάσει του δοθέντος id και εάν αυτό υπάρχει ενημερώνει την βαθμολογία που του δίνει ο user. Εάν η βαθμολογία που δίνεται είναι ίση ή μεγαλύτερη του πέντε (5), τότε το status του μαθήματος ορίζεται σε passed. Εάν είναι μικρότερη του πέντε, το status του μαθήματος ορίζεται σε

not_passed. Τέλος, η εγγραφή του χρήστη αποθηκεύεται με την βαθμολογία που όρισε, και ο η οθόνη του χρήστη επιστρέφει στην σελίδα του μαθήματος. .

```
public function updateCourseGrade($enrolId){  
  
    $enroll = UsersEnrole::find($enrolId);  
  
    if($enroll) {  
        $enroll->grade = request('grade');  
  
        if(request('grade') >= 5) {  
            $enroll->status = 'passed';  
        }  
  
        if(request('grade') < 5) {  
            $enroll->status = 'not_passed';  
        }  
  
        $enroll->save();  
    }  
  
    return redirect()->back();  
}
```

3. EnrolesController.php: Σχετίζεται με την εγγραφή του χρήστη σε ένα μάθημα.

Περιέχει τις ακόλουθες συναρτήσεις:

- a. **create(\$courseId):** Αναζητά στις εγγραφές μία εγγραφή χρήστη παίρνοντας το user id και course id που δίνεται από τον χρήστη. Εάν από την αναζήτηση αυτή το id του χρήστη δεν αντιστοιχεί με κάποιο από αυτό στις εγγραφές των μαθημάτων αλλά το id του μαθήματος υπάρχει, τότε δημιουργείται μια νέα εγγραφή με νέα στοιχεία όπως: user id, course id, κενό βαθμό μαθήματος, και status μαθήματος not passed. Στη συνέχεια, η εγγραφή αποθηκεύεται και ο χρήστης ανακατευθύνεται στη σελίδα του λογαριασμού του. Στην περίπτωση που ο χρήστης είναι ήδη εγγεγραμμένος στο μάθημα, τότε εμφανίζεται το ακόλουθο μήνυμα στον χρήστη: “You have already enrolled!”.

```

public function create($courseId)
{
    $alreadyEnroll = UsersEnrole::where('user_id', '=', auth()->user()->id)
        ->where('course_id', '=', $courseId);

    if($alreadyEnroll->count() == 0){
        if(Auth::user() && Course::find($courseId)){
            $newEnrole = new UsersEnrole();
            $newEnrole->user_id = Auth::user()->id;
            $newEnrole->course_id = $courseId;
            $newEnrole->grade = null;
            $newEnrole->status = 'not_passed';
            $newEnrole->save();

            return redirect(route('my-account'));
        }
    }
    return redirect(route('my-account'))->with('You have already enrolled!');
}

```

- b. **destroy(\$usersEnroleId)**: Διαγράφει την εγγραφή που αντιστοιχεί στο id της εγγραφής που δίνεται από τον χρήστη. Αφού γίνει η διαγραφή της εγγραφής από ένα μάθημα, ο χρήστης ανακατευθύνεται στη σελίδα του λογαριασμού του.

```

public function destroy($usersEnroleId)
{
    /*
     * Find the enroll with the given id and delete it, then redirect to account
     */
    $usersEnrole = UsersEnrole::find($usersEnroleId);
    $usersEnrole->delete();

    return redirect(route('my-account'));
}

```

4. **HomeController.php**: Σχετίζεται με την αρχική σελίδα της ιστοσελίδας PaPel. Περιέχει τις ακόλουθες συναρτήσεις:

- a. **index()**: Εμφανίζει τη λίστα με τα posts για τα Νέα της σχολής στην αρχική μας σελίδα.

```

public function index()
{
    return view('home', [
        'news' => News::all()
    ]);
}

```

5. **NewsController.php**: Σχετίζεται με τα posts για τα Νέα της σχολής. Περιέχει τις ακόλουθες συναρτήσεις:

- a. **index()**: Εμφανίζει τη λίστα με τα posts που έχουμε εισάγει.

```

public function index()
{
    return view('news.index', [
        'news' => News::all()
    ]);
}

```

b. **show(\$newsId)**: Εμφανίζει ένα συγκεκριμένο post με ένα συγκεκριμένο id.

```

public function show($newsId)
{
    return view('news.show', [
        'post' => News::find($newsId)
    ]);
}

```

6. **NotesController.php**: Σχετίζεται με τις σημειώσεις του χρήστη σε ένα μάθημα.

Περιέχει τις ακόλουθες συναρτήσεις:

a. **store(StoreNoteRequest \$request)**: Για τη δημιουργία μιας νέας σημείωσης απαιτείται το id του μαθήματος που ανήκει η σημείωση, το id του χρήστη και το status της σημείωσης που αρχικά είναι πάντα disabled μέχρι να εγκριθεί από τον admin. Όσο η σημείωση έχει status 'disabled', δεν είναι ορατή από κανένα χρήστη. Για την προσθήκη αρχείου pdf ή εικόνας στη σημείωσή μας, χρησιμοποιούμε το package MediaUploadingTrait.

```

public function store(StoreNoteRequest $request)
{
    $request['course_id'] = $request->course_id;
    $request['status'] = 'disabled';
    $request['user_id'] = auth()->user()->id;

    $note = Note::create($request->all());

    foreach ($request->input('files', []) as $file) {
        $note->addMedia(storage_path('tmp/uploads/' . basename($file)))->toMediaCollection('files');
    }

    foreach ($request->input('photo', []) as $file) {
        $note->addMedia(storage_path('tmp/uploads/' . basename($file)))->toMediaCollection('photo');
    }

    return redirect()->back();
}

```

- b. **addLike(\$noteId)**: Για να πατήσουμε like σε μία σημείωση, αναζητούμε τη σημείωση με το id της. Από προεπιλογή το πεδίο like είναι null, οπότε χρειάζεται να το μετατρέψουμε σε numeric δίνοντάς του την τιμή ένα (1). Εάν στη σημείωση υπάρχουν ήδη likes, τότε απλά αυξάνουμε την τιμή του πεδίου likes κατά ένα.

```
public function addLike($noteId)
{
    $note = Note::find($noteId);
    if(is_null($note->likes)){
        $note->likes = 1;
    } else {
        $note->likes = $note->likes + 1;
    }
    $note->save();

    return redirect()->back();
}
```

- c. **addDislike(\$noteId)**: Για το dislike ακολουθούμε την ίδια μεθοδολογία με το like που περιγράψαμε από πάνω.

```
public function addDislike($noteId)
{
    $note = Note::find($noteId);
    if(is_null($note->dislikes)){
        $note->dislikes = 1;
    } else {
        $note->dislikes = $note->dislikes + 1;
    }
    $note->save();

    return redirect()->back();
}
```

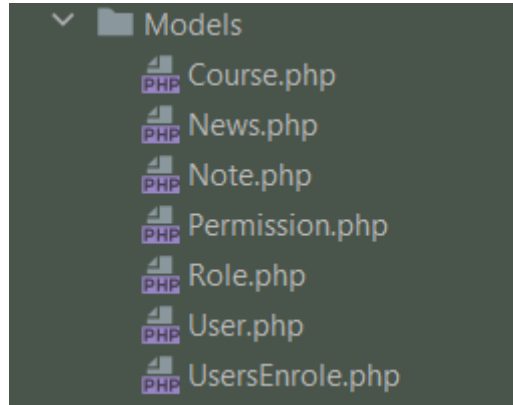
- d. **destroy(\$noteId)**: Διαγράφει την σημείωση που αντιστοιχεί στο id που δίνεται από τον χρήστη.

```
public function destroy($noteId)
{
    $note = Note::find($noteId);
    $note->delete();

    return redirect(route('my-account'));
}
```

3.1.3.2.3 Models

Τα Models αλληλεπιδρούν με την βάση δεδομένων και ανακτούν πληροφορίες των αντικειμένων. Περιέχουν όλη τη λογική που σχετίζεται με τα δεδομένα που ο χρήστης χρησιμοποιεί, όπως για παράδειγμα τις διεπαφές του πρότζεκτ, τις βάσεις δεδομένων και τα πεδία τους. Τα models βρίσκονται στο φάκελο κάτω από το directory **app/Http/Models**.



1. **Course.php:** Αντιπροσωπεύει τον πίνακα Courses της βάσης δεδομένων. Περιέχει πεδία τα οποία μπορούμε να επεξεργαστούμε και είναι τα :
 - title: Αφορά τον τίτλο ενός μαθήματος.
 - description: Αφορά την περιγραφή του μαθήματος
 - ects : Αφορά τις διδακτικές μονάδες ενός μαθήματος.
 - type: Αφορά τον τύπο του μαθήματος. Προέρχεται από τη μεταβλητή TYPE_SELECT και μπορεί να πάρει τις τιμές lab και theory.
 - semester: Αφορά τον αριθμό του εξαμήνου που ανήκει ένα μάθημα και μπορεί να πάρει τιμές απο το 'Α' έως το 'J'.
 - category: Αφορά τις κατηγορίες που μπορεί να ανήκει ένα μάθημα και μπορεί να πάρει τις τιμές: 'Γενικής', 'Ενεργειακών Συστημάτων', 'Σημάτων, Τηλεπικοινωνιών και Δικτύων', 'Ηλεκτρονικής. Υπολογιστών και Δικτύων', 'Πληροφορικής' και 'Επιλογής'.
 - status: Αφορά το status ενός μαθήματος και μπορεί να πάρει τις τιμές Active και Disabled.

Τέλος, περιγράφονται και οι συσχετίσεις του αντικειμένου course μεταξύ των άλλων οντοτήτων. Όπως :

```
public function enroles()
{
    return $this->hasMany(UsersEnrole::class, 'course_id', 'id');
}
```

Ένα μάθημα(course) μπορεί να έχει πολλές(hasMany) εγγραφές φοιτητών(UsersEnrole).

```
public function notes()
{
    return $this->hasMany(Note::class, 'course_id', 'id');
}
```

Ένα μάθημα(course) μπορεί να έχει πολλές(hasMany) σημειώσεις(notes).

2. **News.php:** Αντιπροσωπεύει τον πίνακα News της βάσης δεδομένων. Περιέχει πεδία τα οποία μπορούμε να επεξεργαστούμε και είναι τα :

- title: Αφορά τον τίτλο ενός άρθρου.
- body: Αφορά το κύριο μέρος, δηλαδή το "σώμα" του άρθρου.
- status: Αφορά το status ενός άρθρου. Προέρχεται από τη μεταβλητή STATUS_SELECT και μπορεί να πάρει τις τιμές 'Publish' και 'Draft'. Το status 'Publish' περιγράφει ένα άρθρο που έχει ανέβει στη σελίδα 'News' στο site του PaPel ενώ το 'Draft' πρόκειται για το status του άρθρου πριν ανέβει στη σελίδα. Για την αλλαγή του status είναι υπεύθυνος ο admin, καθώς μόνο αυτός έχει το permission για να ανεβάσει ένα άρθρο στη σελίδα.

3. **Note.php:** Αντιπροσωπεύει τον πίνακα Notes της βάσης δεδομένων. Περιέχει πεδία τα οποία μπορούμε να επεξεργαστούμε και είναι τα :

- user_id: Αφορά το user id του φοιτητή που δημιουργεί μια σημείωση.
- course_id: Αφορά το id του μαθήματος που ανήκει η σημείωση.
- text: Αφορά το κείμενο που μπορεί να περιέχει μια σημείωση.
- likes: Αφορά τα likes μιας σημείωσης.
- dislikes: Αφορά τα dislikes μιας σημείωσης.
- status: Αφορά το status μιας σημείωσης. Προέρχεται από τη μεταβλητή STATUS_SELECT και μπορεί να πάρει τις τιμές 'active' και 'disabled'. Το status 'active' περιγράφει μία ενεργή σημείωση, δηλαδή μια σημείωση σε ένα μάθημα που μπορεί να προβληθεί από όλους τους φοιτητές, ενώ το status 'disabled' περιγράφει μία σημείωση πριν γίνει active. Για την αλλαγή του status από disabled σε active είναι υπεύθυνος ο admin, καθώς μόνο μετά από δική του έγκριση μπορεί να γίνει μια σημείωση ορατή σε όλους.

Μέσα στο Model Note υπάρχουν και οι εξής συναρτήσεις:

- registerMediaConversions: Αποθηκεύει τα αρχεία των εικόνων σε συγκεκριμένες διαστάσεις(50x50, 120x120). Η συγκεκριμένη συνάρτηση μας δίνει ένα thumbnail της εικόνας και μια μικρότερη εκδοχή της για πιο γρήγορη φόρτωση του αρχείου.

```
public function registerMediaConversions(Media $media = null): void
{
    $this->addMediaConversion('thumb')->fit('crop', 50, 50);
    $this->addMediaConversion('preview')->fit('crop', 120, 120);
}
```


- `getFilesAttribute`: Μας φέρνει όλα τα αρχεία που βρίσκονται σε μία σημείωση.

```
public function getFilesAttribute()
{
    return $this->getMedia('files');
}
```

- `getPhotoAttribute`: Μας φέρνει όλα τα αρχεία των εικόνων σε όλες τις διαστάσεις που έχουν δωθεί προηγουμένως, συμπεριλαμβανομένου του url της πηγής του αρχείου εικόνας.

```
public function getPhotoAttribute()
{
    $files = $this->getMedia('photo');
    $files->each(function ($item) {
        $item->url = $item->getUrl();
        $item->thumbnail = $item->getUrl('thumb');
        $item->preview = $item->getUrl('preview');
    });

    return $files;
}
```

Τέλος, περιγράφονται και οι συσχετίσεις του αντικειμένου note μεταξύ των άλλων οντοτήτων. Όπως :

```
public function user()
{
    return $this->belongsTo(User::class, 'user_id');
}
```

Μία σημείωση (note) ανήκει σε μόνο σε έναν χρήστη (user).

```
public function course()
{
    return $this->belongsTo(Course::class, 'course_id');
}
```

Μία σημείωση(note) ανήκει μόνο σε ένα μάθημα(course).

4. **Permission.php:** Αντιπροσωπεύει τον πίνακα Permissions της βάσης δεδομένων. Περιέχει πεδία τα οποία μπορούμε να επεξεργαστούμε και είναι τα :
 - title: Αφορά τον τίτλο που μπορεί να πάρει ένα permission.
5. **Role.php:** Αντιπροσωπεύει τον πίνακα Roles της βάσης δεδομένων. Περιέχει πεδία τα οποία μπορούμε να επεξεργαστούμε και είναι τα :
 - title: Αφορά τον τίτλο που μπορεί να πάρει ένας ρόλος.

Τέλος, περιγράφονται και οι συσχετίσεις του αντικειμένου role μεταξύ των άλλων οντοτήτων. Όπως :

```
public function permissions()
{
    return $this->belongsToMany(Permission::class);
}
```

Ένας ρόλος (role) ανήκει σε πολλά permissions.

6. **User.php:** Αντιπροσωπεύει τον πίνακα Users της βάσης δεδομένων. Περιέχει πεδία τα οποία μπορούμε να επεξεργαστούμε και είναι τα :
 - name: Αφορά το όνομα του φοιτητή.
 - am: Αφορά τον αριθμό μητρώου του φοιτητή.
 - email: Αφορά το email του φοιτητή.
 - email_verified_at: Αφορά την ημερομηνία που ο φοιτητής πραγματοποίησε την εγγραφή του στο σύστημα του PaPel.
 - password: Αφορά τον κωδικό του φοιτητή.
 - remember_token: Αφορά το token που στέλνεται στον χρήστη κατά την επανεκκίνηση κωδικού(reset password).

Μέσα στο Model User υπάρχουν και οι εξής συναρτήσεις:

- **getIsAdminAttribute:** Ελέγχει αν ο χρήστης είναι admin ή όχι. Αυτή η διαδικασία γίνεται ελέγχοντας εάν ο ρόλος(role) του χρήστη έχει το id με αριθμό ένα (1) που αντιπροσωπεύει τον ρόλο του admin.

```
public function getIsAdminAttribute()
{
    return $this->roles()->where('id', 1)->exists();
}
```

- **getEmailVerifiedAtAttribute ,setEmailVerifiedAtAttribute:** Δημιουργεί και προσθέτει στη βάση ένα timestamp που περιγράφει το πότε έγινε η επαλήθευση της εγγραφή ενός χρήστη στο σύστημα μέσω email.

- setPasswordAttribute: Αποθηκεύει το password του χρήστη σε κρυπτογραφημένη μορφή(hash) εάν δεν έχει γίνει ήδη σε προηγούμενο στάδιο.

```

public function setPasswordAttribute($input)
{
    if ($input) {
        $this->attributes['password'] = app('hash')->needsRehash($input) ? Hash::make($input) : $input;
    }
}

```

Τέλος, περιγράφονται και οι συσχετίσεις του αντικειμένου user μεταξύ των άλλων οντοτήτων. Όπως :

```

public function enroles()
{
    return $this->hasMany(UsersEnrole::class, 'user_id', 'id');
}

```

Ένας χρήστης(user) ανήκει σε πολλές εγγραφές σε μάθημα(UsersEnrole).

```

public function notes()
{
    return $this->hasMany(Note::class, 'user_id', 'id');
}

```

Ένας χρήστης(user) ανήκει σε πολλές σημειώσεις(Note).

```

public function roles()
{
    return $this->belongsToMany(Role::class);
}

```

Ένας χρήστης(user) ανήκει σε πολλούς ρόλους(Role).

7. UsersEnrole.php: Αντιπροσωπεύει τον πίνακα UsersEnroles της βάσης δεδομένων.

Περιέχει πεδία τα οποία μπορούμε να επεξεργαστούμε και είναι τα :

- user_id: Αφορά το id του χρήστη που ανήκει σε μία εγγραφή.
- course_id: Αφορά το id ενός μαθήματος που ανήκει σε μία εγγραφή χρήστη.
- grade: Αφορά τον βαθμό που του μαθήματος σε μια εγγραφή χρήστη.

- **status:** Αφορά το status ενός μιας εγγραφής χρήστη σε ένα μάθημα. Προέρχεται από τη μεταβλητή STATUS_SELECT και μπορεί να πάρει τις τιμές 'passed' 'not_passed'. Το status 'passed' περιγράφει μια εγγραφή ενός χρήστη σε ένα μάθημα που έχει βαθμό ίσο ή μεγαλύτερο του πέντε (5), ενώ το status 'not_passed' περιγράφει μία εγγραφή ενός χρήστη σε μάθημα με μη προσβάσιμο βαθμό.

Τέλος, περιγράφονται και οι συσχετίσεις του αντικειμένου userEnrole μεταξύ των άλλων οντοτήτων. Όπως :

```
public function user()
{
    return $this->belongsTo(User::class, 'user_id');
}
```

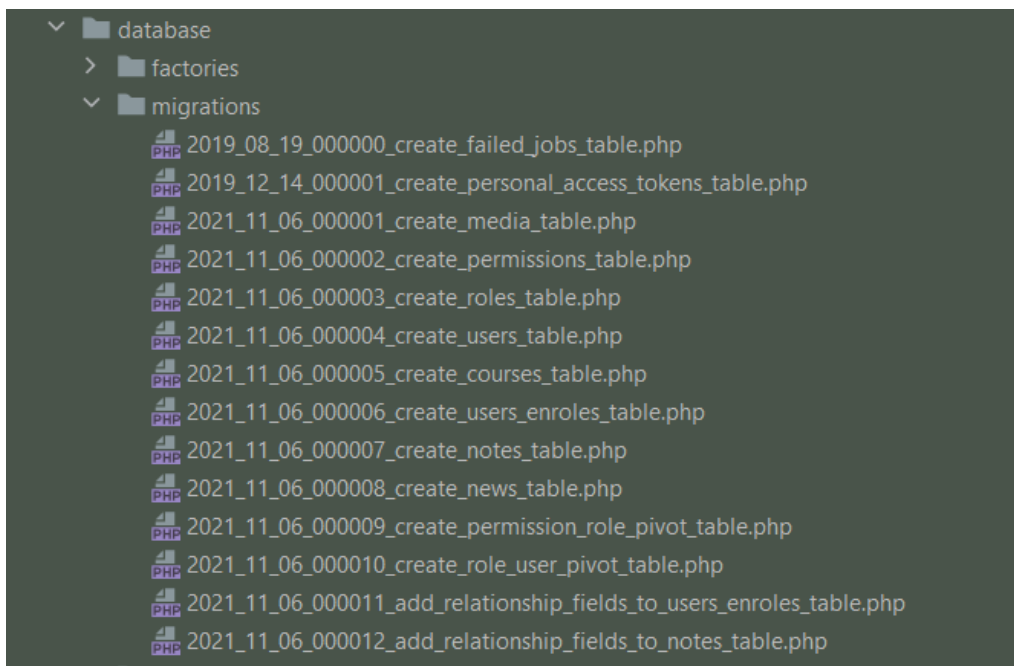
Μία εγγραφή χρήστη μάθημα(userEnrole) ανήκει μόνο σε έναν χρήστη(User).

```
public function course()
{
    return $this->belongsTo(Course::class, 'course_id');
}
```

Μία εγγραφή χρήστη μάθημα(userEnrole) ανήκει μόνο σε ένα μάθημα(Course).

3.1.3.2.3.1 Migrations

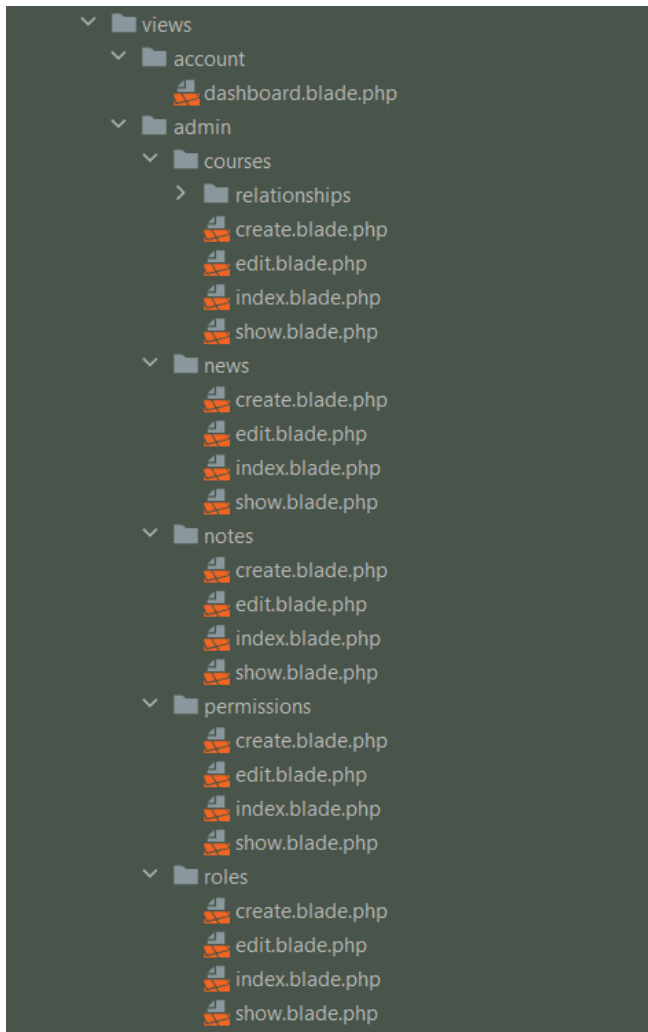
Ένα Database Migration είναι ο τρόπος που ορίζουμε τις αλλαγές μας σε μία Βάση Δεδομένων. Για κάθε πίνακα που προσθέτουμε και για κάθε στήλη που τροποποιούμε, δημιουργούμε ένα καινούριο database migration. Αυτό μας επιτρέπει να μεταβούμε από μία κενή Βάση Δεδομένων σε μία σωστά διαμορφωμένη Βάση με μόνο μία εντολή. Τα Database Migrations επιτρέπει στους developers να διαγράψουν και να επαναδημιουργήσουν μία Βάση Δεδομένων, χωρίς να χρειαστεί να συνδεθούν στο περιβάλλον της Βάσης ή να τρέξουν εντολές SQL. Τα migrations βρίσκονται στο φάκελο κάτω από το directory **database/migrations**.



3.1.3.2.4 Views

Τα Views περιέχουν τον html κώδικα που απαιτείται από την εφαρμογή μας και είναι μία μέθοδος του Laravel που διαχωρίζει την λογική των Controllers από την λογική της παρουσίασης. Τα Views βρίσκονται στο φάκελο resources κάτω από το directory

resources/views. Ο κάθε υπο-φάκελος περιέχει ένα σύνολο από `.blade.php` αρχεία.



Όλα τα Views στο Laravel είναι συνήθως φτιαγμένα με το Blade template(πρότυπο). Το Blade είναι μια απλή αλλά πολύ ισχυρή “μηχανή” templating που συμπεριλαμβάνεται στο Laravel framework. Σε αντίθεση με τα υπόλοιπα templating engines σε γλώσσα PHP, το Blade δεν μας περιορίζει στο να χρησιμοποιήσουμε απλό PHP κώδικα στα templates μας. Το Blade είναι πολύ γρήγορο επειδή πρόκειται για ένα σύνολο κανονικών εκφράσεων (regular expressions) το οποίο τρέχει στα templates μας και τα συντάσσει σε απλή PHP γλώσσα(pure PHP).

Μπορούμε να δημιουργήσουμε ένα View με το να τοποθετήσουμε ένα αρχείο με την επέκταση `.blade.php` κάτω από το directory **resources/views** στην εφαρμογή μας. Η επέκταση `.blade.php` ενημερώνει το framework ότι το αρχείο μας περιέχει το Blade Template. Όταν έχουμε δημιουργήσει το View μπορούμε να το επιστρέψουμε κάθε φορά μέσω των routes ή των controllers χρησιμοποιώντας το view helper:

3.1.4 Υλοποίηση και σχεδιασμός Progressive Web App

Όπως εξηγήθηκε και στο κεφάλαιο 2.1.5, με την προσθήκη δύο (2) αρχείων στο root directory η εφαρμογή μας γίνεται λειτουργική σε mobile συσκευές. Τα δύο αυτά αρχεία είναι το manifest.json και service-worker.js.

manifest.json: Πρόκειται για το JSON αρχείο όπου “ενημερώνει” τον browser σχετικά με την εφαρμογή και πώς αυτή πρέπει να συμπεριφέρεται όταν είναι εγκατεστημένη στο desktop, στο κινητό ή στο tablet μας. Περιλαμβάνει, μαζί με άλλα, το όνομα της εφαρμογής, τα εικονίδια που θα χρησιμοποιεί η εφαρμογή (π.χ. για χρήση thumbnail), και το URL που θα ανοίγει όταν θα γίνεται η χρήση του app.

```
{
  "name": "PaPeL",
  "short_name": "PaPeL",
  "start_url": "/",
  "icons": [
    {
      "src": "/storage/app-icon.png",
      "sizes": "72x72",
      "type": "image/png",
      "purpose": "maskable any"
    },
    {
      "src": "/storage/app-icon.png",
      "sizes": "96x96",
      "type": "image/png",
      "purpose": "maskable any"
    },
    {
      "src": "/storage/app-icon.png",
      "sizes": "144x144",
      "type": "image/png",
      "purpose": "maskable any"
    },
    {
      "src": "/storage/app-icon.png",
      "sizes": "180x180",
      "type": "image/png",
      "purpose": "maskable any"
    }
  ],
}
```

service-worker.js: Η βιβλιοθήκη Workbox μας βοηθάει στη διαχείριση του caching και τις ανάγκες του serving. Είναι η πιο χρησιμοποιημένη βιβλιοθήκη για service-workers και χρησιμοποιείται από το 32% των mobiles. Το workbox-routing είναι ένα module το οποίο μας διευκολύνει στο να δρομολογήσουμε requests σε διαφορετικές functions όπου θα μας παρέχουν τα κατάλληλα responses.

```
importScripts('https://storage.googleapis.com/workbox-cdn/releases/5.0.0/workbox-sw.js');

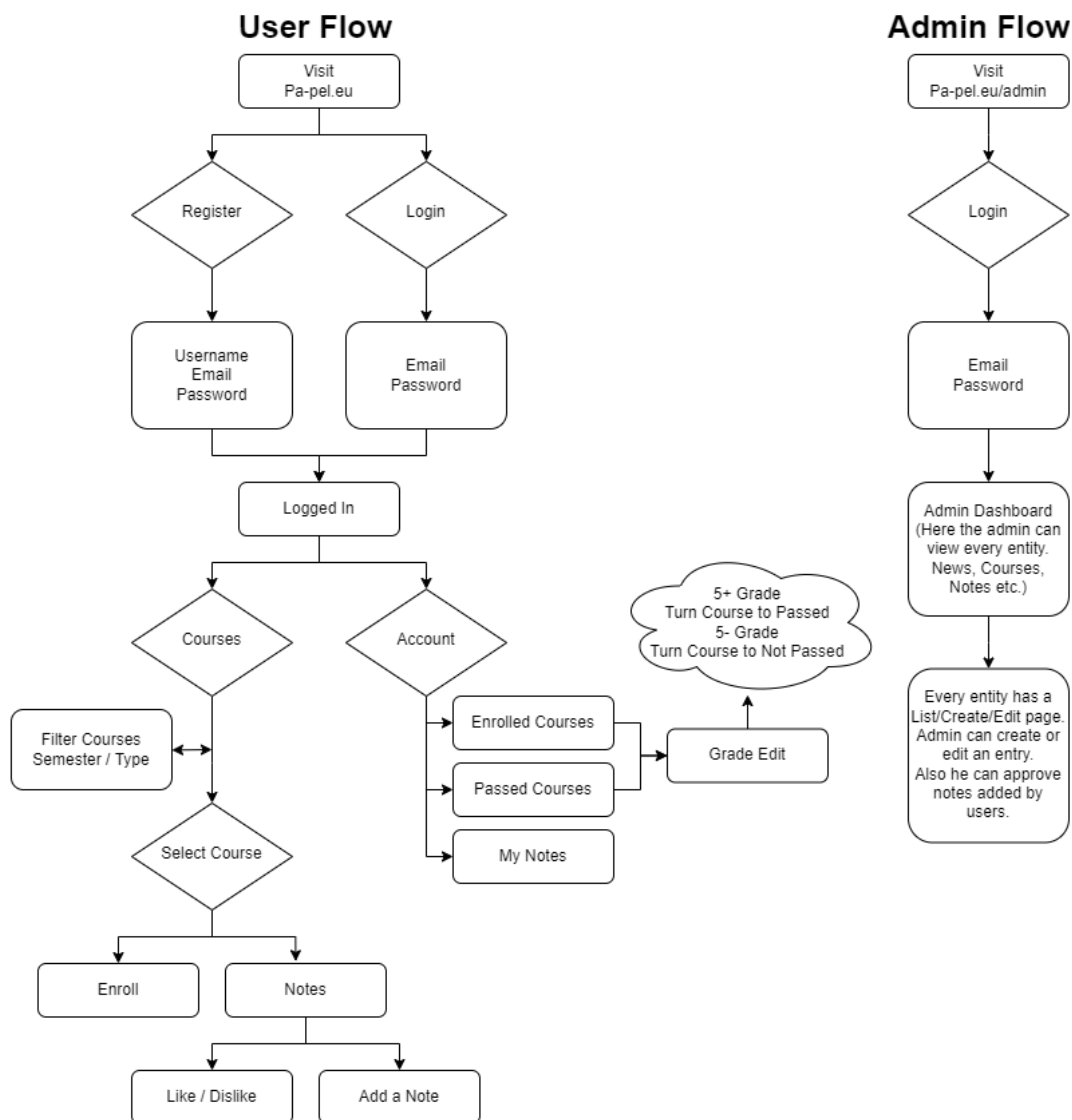
workbox.routing.registerRoute(
  ({request}) => request.destination === 'image',
  new workbox.strategies.NetworkFirst()
)
```

4. Κύριο μέρος – Σενάρια Χρήσης

Για να μπορέσουμε να καταλάβουμε καλύτερα τις δυνατότητες της συγκεκριμένης πλατφόρμας, θα προχωρήσουμε σε ένα σενάριο χρήσης περνώντας από όλες τις διαθέσιμες επιλογές και δυνατότητες που προσφέρει, καλύπτοντας πλήρως την λειτουργικότητα του.

4.1 Flow Diagram

Διάγραμμα ροής ή ροογράφημα (flowchart) είναι ένα κοινού τύπου διάγραμμα που αναπαριστά έναν αλγόριθμο ή μια διαδικασία, δείχνοντας τα βήματα ως κουτιά διαφόρων ειδών που συνδέονται μεταξύ τους με βέλη. Αυτή η διαγραμματική παρουσίαση μπορεί να δώσει λύση βήμα προς βήμα σε ένα γνωστό πρόβλημα. Τα δεδομένα αναπαριστώνται σε κουτιά και τα βέλη δείχνουν τη ροή των δεδομένων. Τα διαγράμματα ροής χρησιμοποιούνται στην ανάλυση, το σχεδιασμό, την τεκμηρίωση ή τον έλεγχο μιας διαδικασίας ή ενός προγράμματος σε διάφορα πεδία. Παρακάτω παρουσιάζεται το διάγραμμα ροής της εφαρμογής Pa-Pel.



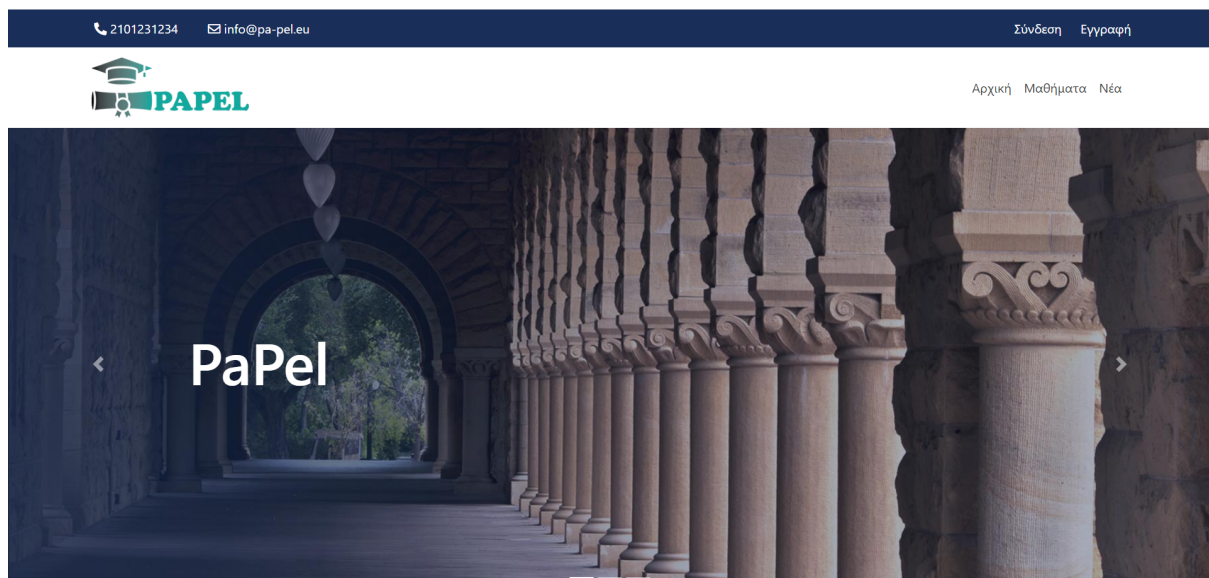
User Flow: Ένας φοιτητής επισκέπτεται την εφαρμογή του Pa-Pel. Εάν είναι η πρώτη φορά που επισκέπτεται το Pa-Pel, πραγματοποιεί Registration με τα απαιτούμενα στοιχεία του (Username, Email, Password). Εάν είναι ήδη εγγεγραμμένος χρήστης τότε πραγματοποιεί Login με τα ζητούμενα στοιχεία του (Email, Password). Όσο είναι συνδεδεμένος στην εφαρμογή, ο φοιτητής έχει τη δυνατότητα να πραγματοποιήσει 2 ενέργειες: Να πλοηγηθεί στην λίστα με τα μαθήματα ή να εισέλθει στη σελίδα του προσωπικού του λογαριασμού. Εισέρχοντας στην λίστα με τα μαθήματα, ο φοιτητής μπορεί να φιλτράρει τα μαθήματα που θα του εμφανίζονται ανά εξάμηνο ή/και τύπο μαθήματος (Θεωρία ή Εργαστήριο). Όταν επιλέξει το επιθυμητό μάθημα ο φοιτητής μπορεί είτε να κάνει εγγραφή σε αυτό (εάν δεν έχει κάνει ήδη) ,ώστε να μπορεί να το παρακολουθεί κατά τη διάρκεια του εξαμήνου, είτε να εισέλθει στο πεδίο με τις σημειώσεις του μαθήματος. Εκεί μπορεί να κάνει Like/Dislike στις υπάρχουσες σημειώσεις άλλων φοιτητών ή να αφήσει τις δικές του σημειώσεις σε μορφή κειμένου, φωτογραφίας αλλά και pdf. Εισέρχοντας στη

σελίδα του προσωπικού του λογαριασμού, ο φοιτητής έχει πρόσβαση στη λίστα των μαθημάτων που είναι εγγεγραμμένος, στα μαθήματα που ο βαθμός του είναι προσβάσιμος αλλά και στις σημειώσεις που έχει προσθέσει στο κάθε μάθημα. Τέλος, ο φοιτητής μπορεί να προσθέσει το βαθμό του στα μαθήματα που παρακολουθεί αλλά και να τον επεξεργαστεί σε περίπτωση αναβαθμολόγησης γραπτού.

Admin Flow: Ένας διαχειριστής επισκέπτεται τη σελίδα του admin dashboard του Pa-Pel. Κάνοντας LogIn με τα ζητούμενα στοιχεία του (Email,Password) και έχοντας τα κατάλληλα δικαιώματα, εισέρχεται στον πίνακα του διαχειριστικού όπου έχει πρόσβαση στην κάθε οντότητα της εφαρμογής όπως είναι τα Μαθήματα, οι Σημειώσεις, το News Feed, κλπ. Ο διαχειριστής έχει το δικαίωμα να προβάλλει, να δημιουργήσει, να επεξεργαστεί αλλά και να διαγράψει οποιαδήποτε εγγραφή στη βάση δεδομένων. Ακόμα, ο διαχειριστής εγκρίνει την καταλληλότητα των σημειώσεων που ανεβάζει ο κάθε φοιτητής, και μόνο έπειτα από δικό του approval οι σημειώσεις μπορούν να ανέβουν στο Pa-Pel app.

4.2 User perspective

Ο φοιτητής επισκέπτεται την ιστοσελίδα του PaPel.



Εάν είναι εγγεγραμμένος χρήστης, επιλέγει 'Σύνδεση',

2101231234 info@pa-pel.eu Σύνδεση Εγγραφή

PAPEL Αρχική Μαθήματα Νέα

Σύνδεση

Email

Κωδικός Πρόσβασης

Σύνδεση Ναι με θυμάσαι

PAPEL
2022 © PaPel

εάν δεν έχει πραγματοποιήσει εγγραφή στο παρελθόν, τότε επιλέγει 'Εγγραφή'.

2101231234 info@pa-pel.eu Σύνδεση Εγγραφή

PAPEL Αρχική Μαθήματα Νέα

Εγγραφή

Όνοματεπώνυμο

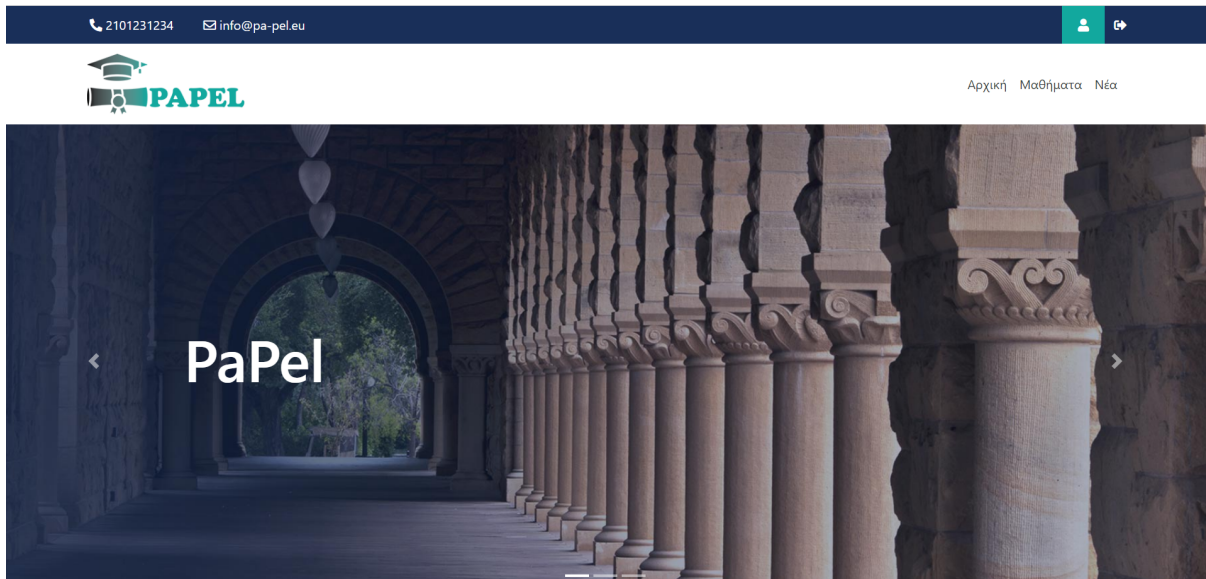
Email

Κωδικός Πρόσβασης

Επαλήθευση κωδικού

Εγγραφή

Όταν ο φοιτητής συνδεθεί με τα στοιχεία του τότε θα βρεθεί στην αρχική σελίδα.



Από την αρχική σελίδα μπορεί να πλοηγηθεί στον λογαριασμό του, στα Μαθήματά του, στα Νέα του τμήματος ή να αποσυνδεθεί από τον λογαριασμό του.

Μαθήματα

Πατώντας τη σελίδα των μαθημάτων από την αρχική σελίδα, ο φοιτητής καταλήγει στην παρακάτω οθόνη

Εξάμηνο	Μαθηματικό	Είδος
Εξάμηνο Α	A3 Μαθηματικός Λογισμός I	Θεωρία
	A4 Γραμμική Άλγεβρα	Θεωρία
	A5 Φυσική	Θεωρία
	A6 Ψηφιακή Λογική Σχεδίαση	Εργαστήριο
	A7 Τεχνικές Προγραμματισμού (Θ+Ε)	Θεωρία
	A8 Ηλεκτρικά Κυκλώματα I (Θ+ Ε)	Θεωρία
Εξάμηνο Β	B9 Μαθηματικός Λογισμός II	Θεωρία

Εξάμηνο

A B Γ Δ Ε ΣΤ Ζ

H Θ I

Εργαστήριο / Θεωρία

Εργαστήριο

Θεωρία

Επικοινωνία


Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Μεγάλου Αλεξάνδρου 1
ΤΚ 263-34, Κορινθίου, Πάτρας

Η σελίδα των μαθημάτων περιέχει όλα τα μαθήματα του τμήματος, χωρισμένα ανά εξάμηνο. Στη δεξιά στήλη βρίσκονται τα φίλτρα αναζήτησης που μπορούν να γίνουν με βάση το εξάμηνο του μαθήματος ή το είδος του μαθήματος που αναζητά ο χρήστης.

Πατώντας πάνω σε οποιοδήποτε μάθημα, ο φοιτητής έχει τη δυνατότητα να δει παραπάνω λεπτομέρειες του μαθήματος όπως η περιγραφή του, το εξάμηνο που διδάσκεται, τον τύπο του(εργαστήριο ή θεωρία) και τις διδακτικές του μονάδες. Εάν είναι ήδη εγγεγραμμένος στο μάθημα τότε μπορεί να πατήσει 'Απεγγραφή' εάν δεν είναι τότε μπορεί να πατήσει 'Εγγραφή' και να ανακατευθυνθεί στη λίστα των μαθημάτων που παρακολουθεί στο λογαριασμό του. Κάτω από τις λεπτομέρειες του μαθήματος, εμφανίζονται όλες οι σημειώσεις που έχουν αφήσει φοιτητές μαζί με την αξιολόγησή τους από likes και dislikes. Όλες οι σημειώσεις του πρώτου "ανεβούν" στο μάθημα έχουν εγκριθεί από τον admin.

Μαθηματικός Λογισμός I



Εγγραφή στο μάθημα

Κωδικός: A3	Διδακτικές Μονάδες: 5
Τύπος: Θεωρία	Εξάμηνο: A

Στα πλαίσια του μαθήματος επιδιώκεται η εισαγωγή του φοιτητή στις βασικές έννοιες του απειροστικού λογισμού συναρτήσεων μιας μεταβλητής ώστε να είναι σε θέση να αντιμετωπίσει απλά μαθηματικά μοντέλα στην ειδικότητά του. Μετά την ολοκλήρωση του μαθήματος ο φοιτητής/τρια θα είναι σε θέση να εκτελεί αλγεβρικές πράξεις στο σύνολο των πραγματικών και μιγαδικών αριθμών, να υπολογίζει παραγώγους και ολοκληρώματα συναρτήσεων μιας μεταβλητής, εμβαδόν μεταξύ καμπύλων, όγκους και επιφάνειες σχημάτων με αξονική συμμετρία, μήκη τόξων, ρυθμούς μεταβολής, να προσδιορίζει ακρότατα συναρτήσεων μιας μεταβλητής, να αθροίζει βασικές αριθμητικές σειρές και να υπολογίζει αναπτύγματα Taylor.

Σημειώσεις +

hi!

👍 0
👎 0

note

👍 0
👎 0

Πατώντας το κουμπί με το σύμβολο '+', ο φοιτητής μπορεί να αφήσει την δική του σημείωση.



Απεικόνιση από το μάθημα

Κωδικός A3

Διδακτικές Μονάδες 5

Τύπος Θεωρία

Εξάμηνο A

Στα πλαίσια του μαθήματος επιδιώκεται η εισαγωγή του φοιτητή στις βασικές έννοιες του απειροστικού λογισμού συναρτήσεων μιας μεταβλητής ώστε να είναι σε θέση να αντιμετωπίσει απλά μαθηματικά μοντέλα στην ειδικότητά του. Μετά την ολοκλήρωση του μαθήματος ο φοιτητής/τρια θα είναι σε θέση να εκτελεί αλγεβρικές πράξεις στο σύνολο των πραγματικών και μιγαδικών αριθμών, να υπολογίζει παραγώγους και ολοκληρώματα συναρτήσεων μιας μεταβλητής, εμβαδόν μεταξύ καμπύλων, όγκους και επιφάνειες σχημάτων με αξονική συμμετρία, μήκη τόξων, ρυθμούς μεταβολής, να προσδιορίζει ακρότατα συναρτήσεων μιας μεταβλητής, να αθροίζει βασικές αριθμητικές σειρές και να υπολογίζει αναπτύγματα Taylor.

Σημειώσεις

Κείμενο*

Αρχεία

Πατήστε εδώ για επιλογή αρχείων

Φωτογραφίες

Πατήστε εδώ για επιλογή φωτογραφιών

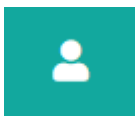


Αποθήκευση

Βλέπουμε πως ο φοιτητής έχει τη δυνατότητα να προσθέσει μια σημείωση με τρεις (3) επιλογές: 1) ως απλό κείμενο, 2) ως αρχείο εγγράφου και 3) ως φωτογραφία . Αφού πατήσει το κουμπί “Αποθήκευση” η σημείωση που ανέβηκε από τον φοιτητή δεν θα είναι ορατή στο μάθημα. Πρέπει πρώτα να περάσει από το στάδιο έγκρισης από τον admin.

Λογαριασμός

Πατώντας στο εικονίδιο του λογαριασμού του,



ο φοιτητής εισέρχεται στην παρακάτω οθόνη:



Αρχική Μαθήματα Νέα

Γειά σου,

ΑΙΜΙΛΙΑ ΠΑΠΑΔΟΥΡΟΥ

Μαθήματα

Περασμένα Μαθήματα

Σημειώσεις



2022 © PaPel

Στη συγκεκριμένη σελίδα, μπορεί να πραγματοποιήσει τρεις (3) ενέργειες: να προβάλλει τα μαθήματα που παρακολουθεί, να προβάλλει τα περασμένα μαθήματά του και μαζί με αυτά να δει τον αυτομάτως υπολογισμένο Μέσο Όρο του και τις Διδακτικές Μονάδες του και τέλος να προβάλλει τις σημειώσεις που έχει αφήσει στα μαθήματα, βλέποντας επίσης τα likes και τα dislikes που έχει πάρει η σημειώσή του.

Πατώντας το κουμπί **Μαθήματα** στο Λογαριασμό του, ο χρήστης προβάλλει τη λίστα των μαθημάτων που παρακολουθεί.

ΑΙΜΙΛΙΑ ΠΑΠΑΔΟΥΡΟΥ

Μαθήματα Περασμένα Μαθήματα Σημειώσεις

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ

Μαθηματικός Λογισμός Ι

Προβολή Βαθμός

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ

Διαδικασιακός Προγραμματισμός (Ε)

Προβολή Βαθμός

Στο κάθε μάθημα μπορεί να προσθέσει στο πεδίο 'Βαθμός' τη βαθμολογία του και να την αποθηκεύσει.

Εάν ο βαθμός που προσθέσει είναι ίσος ή μεγαλύτερος του πέντε(5) δηλαδή προσβάσιμος, τότε το μάθημα αυτό θα μεταφερθεί στην σελίδα 'Περασμένα Μαθήματα'.

Πατώντας το κουμπί **Περασμένα Μαθήματα** στον λογαριασμό του, ο φοιτητής έχει πρόσβαση στη λίστα με τα περασμένα μαθήματά του. Πάνω από τα μαθήματα, υπάρχει ένα progress bar το οποίο υπολογίζει αυτομάτως τον μέσο όρο της βαθμολογίας του αλλά και το σύνολο των διδακτικών μονάδων του.

Μαθήματα Περασμένα Μαθήματα Σημειώσεις
 Μέσος Όρος: 7.5 Διδ. Μονάδες: 10
 67%

Μάθημα	Προβολή	Αξιολόγηση
Φυσική	10	5
Μαθηματικός Λογισμός Ι	5	5

Τέλος, πατώντας το κουμπί Σημειώσεις στον Λογαριασμό του, ο φοιτητής μπορεί να προβάλλει το σύνολο των σημειώσεων που έχει αφήσει σε κάθε μάθημα μαζί με την αξιολόγηση της κάθε σημείωσης, δηλαδή το σύνολο των likes και των dislikes.

Γειά σου.

ΑΙΜΙΛΙΑ ΠΑΠΑΔΟΥΡΟΥ

Μαθήματα Περασμένα Μαθήματα Σημειώσεις

Η σημείωσή μου

👍 2 👎 1

Νέα

Πατώντας τη σελίδα Νέα από την αρχική σελίδα του PaPeI, ο φοιτητής έχει πρόσβαση σε ένα σύνολο από posts σχετικά με τα νέα του τμήματος.

Νέα

Άρθρα



Νέα έκδοση του συγγράμματος «Σήματα και Συστήματα Συνεχούς και Διακριτού Χρόνου με...

[Περισσότερα](#)

7 months ago



13ο Συνέδριο ΣΦΗΜΜΥ: 15 – 17 Απριλίου στην Πάτρα

[Περισσότερα](#)

7 months ago



Διαδικτυακό Σεμινάριο για την ανάπτυξη εφαρμογών Python με WSL2, Hadoop, Yarn ...

[Περισσότερα](#)

7 months ago

Στην παρούσα φάση τα άρθρα του τμήματος έχουν περαστεί με manual τρόπο , όμως στο μέλλον θα θέλαμε για μεγαλύτερη βελτιστοποίηση της διαδικασίας να μεταφέρονται από το κύριο site του τμήματος με τη χρήση API.

Για την αποσύνδεσή του από το σύστημα, ο φοιτητής μπορεί να πατήσει το εικονίδιο της εξόδου που βρίσκεται δίπλα από το λογαριασμό του από οποιοδήποτε σημείο του site.



5. Αποτελέσματα

Η συγκεκριμένη πλατφόρμα, θα μπορούσε να βοηθήσει τόσο τους φοιτητές που ψάχνουν τρόπο να οργανώσουν βοηθητικό ή εκπαιδευτικό περιεχόμενο που τους ενδιαφέρει για την προετοιμασία τους στην εξεταστική περίοδο, αλλά ταυτόχρονα και στο διαμοιρασμό αυτού, πάντα υπό την διαχείριση κάποιου υπεύθυνου καθηγητή. Πέρα από αυτά τα οφέλη, ο φοιτητής μπορεί να έχει στη διάθεσή του τις διδακτικές του μονάδες αλλά και τον πλήρη μέσο όρο του για τον υπολογισμό της τελικής βαθμολογίας του πτυχίου του. Η μοντέρνα σχεδιασμένη και προσαρμοσμένη για όλες τις συσκευές διαδικτυακή εφαρμογή, μπορεί να δώσει λύση σε μία μεγάλη γκάμα ζητουμένων, ενώ παράλληλα, κρατάει καθαρή την εικόνα και το συλλεγμένο περιεχόμενο του χρήστη, χωρίς να γεμίζει με θόρυβο την εμπειρία που του προσφέρει. Στοχεύει στην αύξηση της παραγωγικότητας του φοιτητή, ενώ τηρεί πάντα τα πλαίσια της εποπτείας από τον αρμόδιο καθηγητή-διαχειριστή.

5.1 Συμπεράσματα

Συμπεράσματα Έχοντας ολοκληρώσει την συγκεκριμένη εργασία με επιτυχία, προσθέτουμε στο προσωπικό μου portfolio, μία ολοκληρωμένη σουίτα οργάνωσης, δομημένη από την αρχή έως το τέλος με προσωπική έρευνα. Η έρευνα αυτή μας έδωσε την δυνατότητα να δουλέψουμε και να μάθουμε νέες τεχνολογίες, ενώ ταυτόχρονα να δοκιμάσουμε τεχνικές και υλοποιήσεις που χρησιμοποιούνται κατά κόρον από εταιρίες στο κλάδο. Επίσης η ευκαιρία να δουλέψουμε σε μία τέτοια λύση, μας βοήθησε να καταλάβουμε πως λειτουργεί μία ομάδα, πως δρουν τα άτομα μέσα σε αυτή και πως μπορούμε εμείς σαν προγραμματιστές να δώσουμε λύσεις σε πολλά από αυτά.

5.2 Μελλοντική Επέκταση

Βλέπουμε πως η παρούσα υλοποίηση προσφέρει μερικές χρήσιμες λειτουργίες που θα μπορούσαν να αποτελέσουν εργαλεία για το φοιτητή όσον αφορά την εύκολη διαχείριση των σημειώσεων του. Ακόμα, μπορεί να έχει άμεση πληροφόρηση σχετικά με την βαθμολογία του και τα μαθήματα που παρακολουθεί και όλα αυτά σε μία μόνο πλατφόρμα. Ευελπιστούμε πως η εφαρμογή μας θα μπορέσει να χρησιμοποιηθεί από τους φοιτητές αλλά και τους καθηγητές του τμήματος μας (και όχι μόνο) και να τους διευκολύνει στη σωστή οργάνωση της διδακτικής ύλης αλλά και στον τρόπο που έχουν πρόσβαση σε αυτή.

Βιβλιογραφία-Πηγές

<https://laravel.com/docs/9.x/>
<https://laravel.com/docs/9.x/controllers>
<https://laravel.com/docs/9.x/routing>
<https://laravel.com/docs/9.x/views>
<https://laravel.com/docs/9.x/blade>
<https://web.dev/learn/pwa/workbox/>
<https://www.mysql.com/>
<https://www.phpmyadmin.net/>
<https://www.cretaforce.gr/shared-hosting/>