



τμήμα
μηχανικών
πληροφορικής τ.ε.
Τ.Ε.Ι. Δυτικής Ελλάδας

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

“Travel the World”

ΣΠΟΥΔΑΣΤΡΙΑ

ΕΜΜΑΝΟΥΕΛΑ ΝΤΟΥΛΕ ΑΜ 1998

ΕΙΣΗΓΗΤΗΣ

ΑΣΗΜΑΚΟΛΠΟΥΛΟΣ ΓΕΩΡΓΙΟΣ

Περιεχόμενα

Εισαγωγή	
1ο Κεφάλαιο	7
Σκοπός Εργασίας	7
Στόχοι	8
Παραδοτέα	9
Δομή Εργασίας	10
2ο Κεφάλαιο	
Διαδίκτυο και Οφέλη	11
Εύκολη Επικοινωνία	12
Ηλεκτρονικό ταχυδρομείο	13
Επικοινωνία από Η/Υ σε Η/Υ (Skype)	13
Η Πληροφορία	15
Εκπαίδευση	16
Εύκολη πρόσβαση στη γνώση	16
Εκπαίδευση εξ' αποστάσεως	17
3ο Κεφάλαιο	
Web εφαρμογές	19
Client–server μοντέλο	19
HTML Hyper Text Markup Language	21

Προέλευση	22
Πρώτες προδιαγραφές	23
Dynamic web page (Τι είναι οι δυναμικές σελίδες)	24
Ajax	25
JSON	26
CSS - Cascading Style Sheets	27
JavaScript	28
Προγραμματισμός Εφαρμογής σε επίπεδο Εξυπηρετητής (server)-Side ...	29
PHP	30
Τι είναι το framework και γιατί χρειάζεται	31
Symfony	32
Τι είναι το Autocomplete.....	33
Τι είναι το twig	34
Βάση Δεδομένων MySQL	34
PHP – MySQL	35
Αρχιτεκτονική 3Tier	36
Συμπέρασμα	37
ΥΛΟΠΟΙΗΣΗ	37
Μεθοδολογία ανάπτυξης λογισμικού	37
Σχεδίαση Βάσης Δεδομένων	38
4ο Κεφάλαιο	
ΥΛΟΠΟΙΗΣΗ	44
Εισαγωγή	44
Βάσεις Δεδομένων Doctrine ORM (Symfony)	45

Installing Doctrine (Εγκατάσταση)	45
Creating an Entity Class (Δημιουργία κλάσης οντοτήτων)	45
Model-view-controller	47
Επεξήγηση κώδικα	49
Μελλοντική εργασία	59
Βιβλιογραφία	60

Travel the World

An aerial photograph of a city skyline, likely Hong Kong, with a large circular overlay in the center. The overlay contains the text 'BETTER TO SEE SOMETHING ONCE, THAN TO HEAR ABOUT IT A THOUSAND TIMES.' and 'Asian Proverb' below it. The background shows a dense urban landscape with many skyscrapers and a harbor with boats.

BETTER TO SEE
SOMETHING ONCE,
THAN TO HEAR ABOUT
IT A THOUSAND TIMES.

Asian Proverb

Εισαγωγή

Οι πιο ευτυχισμένοι άνθρωποι είναι αναμφισβήτητα εκείνοι που έχουν εμπειρίες ζωής και όχι αρκετά ή και ακριβά αντικείμενα στο σπίτι τους. Όσο πιο σύντομα το καταλάβεις, τόσο πιο σύντομα θα αυξήσεις τον ψυχικό σου πλούτο. Τα αντικείμενα είναι άψυχα, ενώ η κάθε εμπειρία που θα αποκομίσεις κουβαλάει μαζί της συναισθήματα, μαθήματα και αλλαγές.

Το να αποκτήσεις μια τεράστια τηλεόραση μπορεί να είναι τέλειο γιατί θα βλέπεις σε μεγάλη οθόνη τις αγαπημένες σου ταινίες και θα περνάς την ώρα σου με αγαπημένα πρόσωπα. Ή και μόνος! Όμως για πόσο πιστεύεις ότι αυτό μπορεί να σε γεμίσει; Τα υλικά τα βαριόμαστε! Είναι νόμος. Για αυτό και ο καταναλωτισμός έχει αυξηθεί απεριόριστα τα τελευταία χρόνια. Αλλάζουμε κάθε χρόνο.. κινητό και κάθε εποχή.. ρούχα!

Αν με τα ίδια χρήματα αγοράσεις ένα ταξίδι, θα αποκομίσεις πολλά περισσότερα. Εμπειρίες, μοναδικές στιγμές, μαθήματα ζωής, νέους φίλους, αναμνήσεις και τόσα ακόμη. Ένα ταξίδι δεν σου χαρίζει μόνο μια ακόμη χώρα στο ενεργητικό σου, αλλά μια βαθύτερη βουτιά στον βυθό του εαυτού σου. Επενδύοντας χρήματα στα ταξίδια καταφέρνεις να γίνεσαι καλύτερος και να εκτιμάς περισσότερο τα πράγματα που ήδη έχεις.

Ο Δρ. Thomas Gilovich είναι καθηγητής του Cornell University και λέει χαρακτηριστικά επί του θέματος: «Αγοράζουμε πράγματα για να μας κάνουν χαρούμενους, και το πετυχαίνουμε. Αλλά για λίγο, τα καινούργια αντικείμενα μας ενθουσιάζουν αλλά μετά τα συνηθίζουμε». Και συνεχίζει: «Δεν υποστηρίζω ότι δεν πρέπει να ανταμείψουμε τον εαυτό μας μετά από σκληρή δουλειά με ένα νέο ρούχο ή μια έξοδο, αλλά να επενδύουμε περισσότερα από εμπειρίες που δημιουργούν αναμνήσεις παρά σε αντικείμενα που δε θα έχουν ενδιαφέρον σε σύντομο διάστημα. Οι εμπειρίες είναι το μεγαλύτερο μέρος του εαυτού μας από τα αντικείμενα. Μπορεί να θεωρείτε ότι η σας συνδέεται με αυτά τα αντικείμενα, αλλά είναι έξω από εσάς. Αντίθετα, οι εμπειρίες σας είναι μέρος σας και αυτοί που είσαστε τελικά.»

Συμφωνούμε απόλυτα με τον Δρ. Thomas Gilovich. Δεν είναι τυχαίο που οι πιο ευτυχισμένοι άνθρωποι «επενδύουν» τα χρήματά τους σε ταξίδια. Αν σκεφτείς τον εαυτό σου μετά από αρκετά χρόνια και άσπρα μαλλιά, καθισμένο σε μια καρέκλα τι θα θέλεις να θυμάται, «κοιτώντας πίσω»; Μια ζωή γεμάτη από ένα υπέροχο σπίτι, τέλεια έπιπλα, πανάκριβα ρούχα και φανταστικό αμάξι; Ή μια ζωή γεμάτη ταξίδια, νέους ανθρώπους, πολλά χαμόγελα και εμπειρίες;

Σίγουρα διάλεξες το δεύτερο. Αν μάλιστα θέλεις να το κάνεις πραγματικότητα πρέπει να ξεκινήσεις από τώρα. Η ζωή δεν περιμένει, ούτε και το αεροπλάνο που θα σε πάει στον προορισμό της καρδιάς σου. Φτιάξε βαλίτσα και ξεκίνα για το ταξίδι της ζωής σου! ΤΩΡΑ!

1^ο Κεφάλαιο

Σκοπός Εργασίας

Σκοπός της εργασίας είναι η σχεδίαση και κατασκευή ενός δικτυακού τόπου που θα υλοποιεί ολοκληρωμένο σύστημα για την παρουσίαση ταξιδιωτικών προορισμών, άλλα και δραστηριοτήτων που θα μπορεί ο χρήστης να συμμετέχει ανάλογα με τα ενδιαφέροντα του. Επίσης στην ιστοσελίδα αυτή θα παρουσιάζονται πληροφορίες για αξιοθέατα που μπορεί ο χρήστης να επισκεφτεί στους προορισμούς που έχει επιλέξει.

Εν συνεχεία η ιστοσελίδα θα δίνει την δυνατότητα οι χώρες, οι πόλεις, τα νησιά άλλα και τα αξιοθέατα και οι προορισμοί που βρίσκονται σε αυτά να βαθμολογούνται έτσι ώστε ανα βδομάδα η σελίδα να προτείνει τους επισκέπτες τα αξιοθέατα, τους προορισμούς άλλα και τις δραστηριότητες που έχουν την μεγαλύτερη απήχηση

Στόχοι

Η εργασία περιλαμβάνει οι ερευνητικούς στόχους και στόχους που αφορούν στην ανάλυση και σχεδίαση της εφαρμογής και στόχους υλοποίησης της web εφαρμογής. Έτσι οι στόχοι της συγκεκριμένης εργασίας είναι οι ακόλουθοι:

- Έρευνα των λειτουργικών απαιτήσεων των χρηστών μιας ανάλογης εφαρμογής.
- Μελέτη της σχεδίασης και δημιουργίας Βάσεων Δεδομένων με τη χρήση της MySQL workbench.
- Μελέτη γλώσσας HTML και javascript
- Μελέτη γλώσσας PHP για τη δημιουργία ενός δικτυακού τόπου.
- Χρήση PHPstorm .
- Χρήση framework symfony.
- Υλοποίηση της βάσης
- Υλοποίηση της εφαρμογής με τη χρήση τεχνολογία PHP, HTML και Javascript.

Παραδοτέα

Η εργασία περιλαμβάνει το θεωρητικό μέρος που περιγράφει τον τρόπο εργασίας, την μεθοδολογία και τον τρόπο υλοποίησης καθώς επίσης και την τεκμηρίωση που αφορά στον Παγκόσμιο Ιστό και στην ανάπτυξη web εφαρμογών. Επίσης η εργασία περιλαμβάνει την εφαρμογή που αποτελείται από τις δυναμικές ιστοσελίδες καθώς επίσης και την υλοποίηση της βάσης δεδομένων που διαχειρίζεται τις εγγραφές του δικτυακού τύπου. Έτσι συνοπτικά τα παραδοτέα της συγκεκριμένης εργασίας είναι τα εξής:

- Θεωρητικό μέρος
- Βάση Δεδομένων
- Δυναμικές Ιστοσελίδες εμφάνισης αποτελεσμάτων στον τελικό χρήστη.
- Σύστημα διαχείρισης του περιεχομένου των ιστοσελίδων.

Δομή Εργασίας

Στο πρώτο κεφάλαιο γίνεται μια πλήρης περιγραφή της εργασίας όσον αφορά το θέμα της, τους στόχους που επιδιώκει να καλύψει καθώς επίσης και τη δομή της.

Στο δεύτερο κεφάλαιο γίνεται μια εκτενής αναφορά στις υπηρεσίες του Διαδικτύου, στις τεχνολογίες προγραμματισμού στον Παγκόσμιο Ιστό και τους τρόπους που διακινείται το περιεχόμενό του καθώς επίσης και στα εργαλεία που χρησιμοποιούνται στο χώρο της ανάπτυξης web εφαρμογών.

Στο τρίτο κεφάλαιο περιγράφεται η μεθοδολογία ανάπτυξης της συγκεκριμένης εφαρμογής καθώς επίσης και τα στάδια ανάλυσης και σχεδίασης της εφαρμογής.

Στο τέταρτο κεφάλαιο γίνεται πλήρης περιγραφή των δυναμικών ιστοσελίδων της εφαρμογής με σχόλια για τον κώδικα και προβολή των πιο σημαντικών τμημάτων του ενώ παράλληλα παρουσιάζονται φωτογραφίες που δείχνουν τον τρόπο προβολής και περιήγησης σε κάθε ιστοσελίδα.

Στο πέμπτο κεφάλαιο γίνεται μια αξιολόγηση της εφαρμογής καθώς και μια αξιολόγηση της εργασίας συνολικά. Γίνεται μια αποτίμηση για το βαθμό στον οποίο οι στόχοι επιτεύχθηκαν και γίνονται προτάσεις για βελτιώσεις και προσθήκες στο μέλλον.

2^ο Κεφάλαιο

Διαδίκτυο και Οφέλη

Το Διαδίκτυο (αγγλ. Internet) είναι παγκόσμιο σύστημα διασυνδεδεμένων δικτύων υπολογιστών, οι οποίοι χρησιμοποιούν καθιερωμένη ομάδα πρωτοκόλλων, η οποία συχνά αποκαλείται "TCP/IP" (αν και αυτή δεν χρησιμοποιείται από όλες τις υπηρεσίες του Διαδικτύου) για να εξυπηρετεί εκατομμύρια χρήστες καθημερινά σε ολόκληρο τον κόσμο. Οι διασυνδεδεμένοι ηλεκτρονικοί υπολογιστές ανά τον κόσμο, οι οποίοι βρίσκονται σε ένα κοινό δίκτυο επικοινωνίας, ανταλλάσσουν μηνύματα (πακέτα) με τη χρήση διαφόρων πρωτοκόλλων (τυποποιημένοι κανόνες επικοινωνίας), τα οποία υλοποιούνται σε επίπεδο υλικού και λογισμικού. Το κοινό αυτό δίκτυο καλείται Διαδίκτυο.

Εύκολη Επικοινωνία

- Ηλεκτρονικό ταχυδρομείο

Ίσως ένα από σημαντικότερα πλεονεκτήματα του internet που φάνηκε από την αρχή είναι το e-mail (ηλεκτρονικό ταχυδρομείο). Πλέον η επικοινωνία των ανθρώπων ανά σημείο του κόσμου ήταν εφικτή σχεδόν σε πραγματικό χρόνο. Μέσω του e-mail οι άνθρωποι μπορούσαν να ανταλλάσσουν μηνύματα, εικόνες και άλλα ψηφιακά αρχεία. Εκεί που ένα γράμμα έκανε κάποιες μέρες να φτάσει μεσώ του συμβατικού ταχυδρομείου τώρα μεσώ του ηλεκτρονικού ταχυδρομείου φτάνει σε δευτερόλεπτα.

Η χρήση του ηλεκτρονικού ταχυδρομείου είναι επίσης σημαντική και για την εύρεση εργασίας (αποστολή βιογραφικού), επικοινωνία με δημόσιες υπηρεσίες και άλλα. Η δημιουργία ενός προσωπικού λογαριασμού e-mail είναι δωρεάν και υπάρχουν αρκετοί εναλλακτικοί πάροχοι όπως το yahoo, gmail, hotmail και άλλα. Επίσης σχεδόν όλοι οι πάροχοι διαθέτουν webmail (ή web-based mail) και έτσι ο χρήστης μπορεί από οποιαδήποτε συσκευή που διαθέτει internet και έναν πλοηγητή διαδικτύου (web browser) να μπει στο λογαριασμό του.



- Επικοινωνία από Η/Υ σε Η/Υ (Skype)

Για την εύκολη επικοινωνία εκτός από το ηλεκτρονικό ταχυδρομείο μετέπειτα εμφανίστηκε και το Skype (VoIP εφαρμογή). Αρχικά ήταν για επικοινωνία από Η/Υ σε Η/Υ. Πλέον προσφέρει κλήσεις σε οποιοδήποτε μέρος του κόσμου, σε οποιοδήποτε δίκτυο τηλεφωνίας, σταθερής και κινητής, με χαμηλές χρεώσεις, Instant Messenger, δυνατότητα αποστολής SMS, αποστολής αρχείων και δυνατότητα συνδιάσκεψης. Αυτό που απαιτείται είναι η εγκατάσταση του λογισμικού (δωρεάν από την επίσημη ιστοσελίδα <http://www.skype.com>) και στους υπολογιστές από όπου οι χρήστες θα έρθουν σε επικοινωνία.

Σαφώς το Skype δεν είναι η μοναδική VoIP εφαρμογή μίας και οι τηλεπικοινωνιακοί πάροχοι στη χώρα μας έχουν ήδη αρχίσει να προσφέρουν αντίστοιχες υπηρεσίες. Μερικά από τα πλεονεκτήματα της VoIP τηλεφωνίας είναι:

Μέσω Internet να μπορείς να επικοινωνείς με οποιονδήποτε με πάρα πού φθηνές χρεώσεις. Το πρωτόκολλο VoIP συνδυάζεται με άλλες υπηρεσίες όπως την αποστολή και λήψη fax, βιντεοκλήσεις και πολλές άλλες.

Δεν πληρώνεις το παραδοσιακό πάγιο για τους τηλεπικοινωνιακούς παρόχους. Δηλαδή απολαμβάνεις αρκετές υπηρεσίες χωρίς επιπλέον χρεώσεις.



Η Πληροφορία

Εκτός από το e-mail ένα από τα σημαντικότερα πλεονεκτήματα του internet είναι η εύκολη πρόσβαση στην πληροφορία. Πλέον στο internet μπορούμε να βρούμε οποιαδήποτε πληροφορία (δεν παύει όμως η ουσιαστική να βρίσκεται στις πραγματικές βιβλιοθήκες) μιας και τα πάντα τείνουν προς την ψηφιακή εποχή όπως εφημερίδες, άλλα μέσα ενημέρωσης και πολλά άλλα. Όμως επειδή υπάρχει ένας τεράστιος όγκος από δεδομένα στο internet και για την διευκόλυνση μας δημιουργήθηκαν οι μηχανές αναζήτησης όπως μια από τις πιο γνωστές που έχει επικρατήσει σήμερα ή google. Έκτος από την πρόσβαση στην πληροφορία είναι εφικτή και το να ανεβάσει κάποιος χρήστης στο internet πληροφορία όπως μέσω ιστοσελίδων, ιστολογιών (blogs), φόρουμς και πολλά άλλα. Και το κυριότερο μέσω οποιουδήποτε πλοηγητή μπορούμε να διαβάσουμε πληροφορία σε οποιαδήποτε γλώσσα.



Εκπαίδευση

- Εύκολη πρόσβαση στη γνώση

Το internet μπορεί να χρησιμοποιηθεί ως πηγή μεγάλου όγκου εκπαιδευτικού υλικού και δεδομένων επίσης μπορούμε να έχουμε πρόσβαση σε ηλεκτρονικές βιβλιοθήκες σε πολλά μέρη του κόσμου σε πολύ γρήγορο χρόνο.

- Εκπαίδευση εξ' αποστάσεως

Η εξ αποστάσεως εκπαίδευση απευθύνεται σε όλους, αλλά ιδιαίτερα σε άτομα που είναι εργαζόμενα ή μεγαλύτερης ηλικίας ακόμα και σε άτομα Α.Μ.Ε.Α. (άτομα με ειδικές ανάγκες), δηλαδή σε ανθρώπους που δεν μπόρεσαν να ακολουθήσουν την παραδοσιακή εκπαίδευση. Όμως, συνήθως η εκπαίδευση εξ αποστάσεως διαρκεί περισσότερο σε σχέση με την παραδοσιακή εκπαίδευση. Επίσης, μπορεί ο εκπαιδευόμενος να συμμετέχει σε επιμορφωτικά σεμινάρια όπου μπορεί να βελτιωθεί σε διάφορους τομείς. Μερικά από τα πλεονεκτήματα της εκπαίδευσης εξ αποστάσεως είναι:

Χρήση των νέων τεχνολογιών για την εξασφάλιση συνθηκών αμφίδρομης επικοινωνίας μεταξύ εκπαιδευτή και εκπαιδευόμενου.

Ο εκπαιδευόμενος εκπαιδύεται να μαθαίνει ενώ βρίσκεται στο σπίτι του ή στη δουλειά του χωρίς τη φυσική παρουσία του εκπαιδευτή του.

Χρησιμοποιούνται ευέλικτες διδακτικές τεχνικές προκειμένου να βοηθήσουν κάθε εκπαιδευόμενο να προσαρμόσει την εκπαιδευτική διαδικασία στις ανάγκες του, στο ρυθμό της ζωής του, στο επίπεδο γνώσεών του και στο προσωπικό του στυλ μάθησης.

Μια σειρά τεχνολογικών μέσων χρησιμοποιούνται για να φέρουν σε επαφή τον εκπαιδευόμενο με τον εκπαιδευτή του και να αποκτήσει πρόσβαση στο εκπαιδευτικό υλικό.

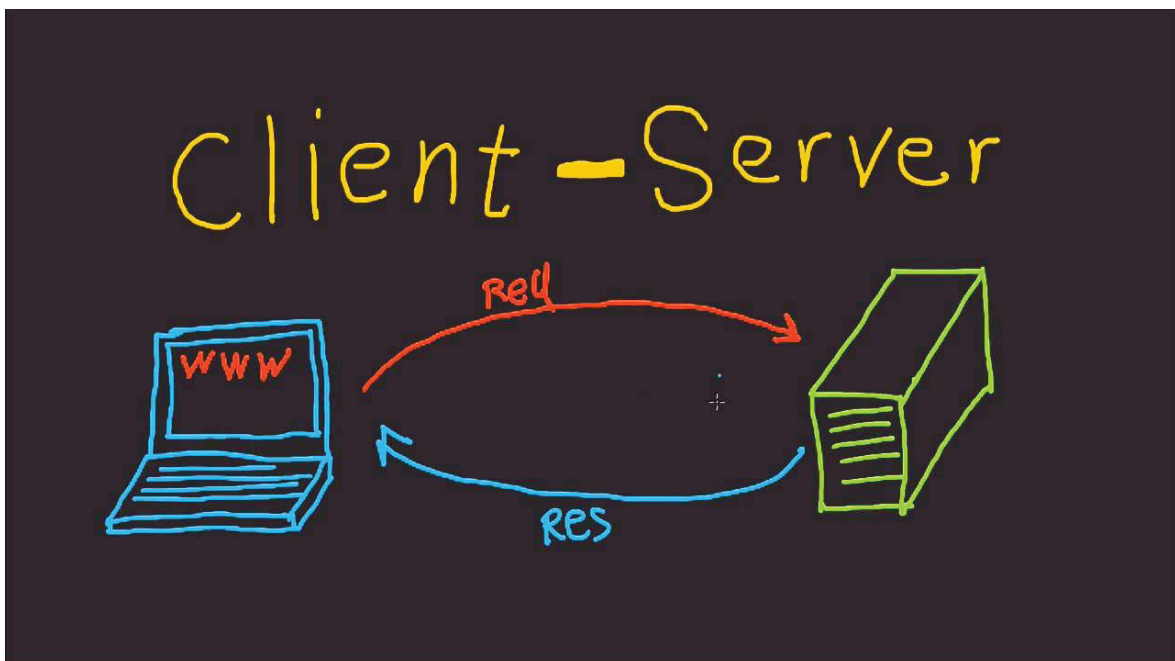


3^ο Κεφάλαιο

Web εφαρμογές

Client-server μοντέλο

Το μοντέλο πελάτη-διακομιστή είναι μια κατανεμημένη δομή εφαρμογής που χωρίζει εργασίες ή φόρμες εργασίας μεταξύ των παρόχων ενός πόρου ή μιας υπηρεσίας, που ονομάζεται διακομιστές και οι αιτούντες υπηρεσία, που καλούνται πελάτες. Συχνά, οι πελάτες και οι διακομιστές επικοινωνούν μέσω ενός δικτύου υπολογιστών σε ξεχωριστό υλικό, αλλά και ο πελάτης και ο διακομιστής μπορεί να βρίσκονται στο ίδιο σύστημα. Ένας κεντρικός υπολογιστής διακομιστή εκτελεί ένα ή περισσότερα προγράμματα διακομιστών που μοιράζονται τους πόρους τους με τους πελάτες. Ένας πελάτης δεν μοιράζεται κανέναν από τους πόρους του, αλλά ζητά τη λειτουργία ενός περιεχομένου ή μιας υπηρεσίας ενός διακομιστή. Έτσι, οι πελάτες ξεκινούν συνεδρίες επικοινωνίας με διακομιστές που περιμένουν εισερχόμενες αιτήσεις. Παραδείγματα εφαρμογών



ηλεκτρονικών υπολογιστών που χρησιμοποιούν το μοντέλο πελάτη-διακομιστή είναι το ηλεκτρονικό ταχυδρομείο, η εκτύπωση δικτύου και το World Wide Web.

Το χαρακτηριστικό πελάτη-διακομιστή περιγράφει τη σχέση των συνεργαζόμενων προγραμμάτων σε μια εφαρμογή. Το στοιχείο διακομιστή παρέχει μια λειτουργία ή μια υπηρεσία σε έναν ή πολλούς πελάτες, οι οποίοι εκκινούν αιτήσεις για τέτοιες υπηρεσίες. Οι διακομιστές ταξινομούνται από τις υπηρεσίες που παρέχουν. Για παράδειγμα, ένας διακομιστής ιστού εξυπηρετεί ιστοσελίδες και ένας διακομιστής αρχείων εξυπηρετεί αρχεία υπολογιστή. Ένας κοινός πόρος μπορεί να είναι οποιοδήποτε από τα λογισμικά και ηλεκτρονικά στοιχεία του υπολογιστή του διακομιστή, από προγράμματα και δεδομένα σε επεξεργαστές και συσκευές αποθήκευσης. Η κοινή χρήση πόρων ενός διακομιστή αποτελεί υπηρεσία.

Το αν ο υπολογιστής είναι πελάτης, διακομιστής ή και οι δύο καθορίζεται από τη φύση της εφαρμογής που απαιτεί τις λειτουργίες της υπηρεσίας. Για παράδειγμα, ένας μόνο υπολογιστής μπορεί να τρέξει συγχρόνως λογισμικό διακομιστή ιστού και διακομιστή αρχείων για να εξυπηρετήσει διαφορετικά δεδομένα σε πελάτες που κάνουν διαφορετικά είδη αιτημάτων. Το λογισμικό πελάτη μπορεί επίσης να επικοινωνεί με λογισμικό διακομιστή στον ίδιο υπολογιστή. Η επικοινωνία μεταξύ διακομιστών, όπως για το συγχρονισμό δεδομένων, καλείται μερικές φορές επικοινωνία μεταξύ διακομιστών ή διακομιστή-διακομιστή.

Γενικά, μια υπηρεσία είναι μια αφαίρεση των πόρων του υπολογιστή και ένας πελάτης δεν χρειάζεται να ασχολείται με τον τρόπο με τον οποίο ο διακομιστής εκτελεί την εκπλήρωση του αιτήματος και την παράδοση της απάντησης. Ο πελάτης πρέπει μόνο να κατανοήσει την απάντηση βάσει του γνωστού πρωτοκόλλου εφαρμογής, δηλαδή του περιεχομένου και της μορφοποίησης των δεδομένων για την ζητούμενη υπηρεσία.

Οι πελάτες και οι διακομιστές ανταλλάσσουν μηνύματα σε ένα πρότυπο μηνυμάτων αίτησης-απόκρισης. Ο πελάτης στέλνει ένα αίτημα και ο διακομιστής επιστρέφει μια απάντηση. Αυτή η ανταλλαγή μηνυμάτων είναι ένα παράδειγμα επικοινωνίας μεταξύ των διαδικασιών. Για να επικοινωνούν, οι υπολογιστές πρέπει να έχουν μια κοινή γλώσσα και πρέπει να ακολουθούν κανόνες ώστε ο πελάτης και ο εξυπηρετητής να γνωρίζουν τι πρέπει να περιμένουν. Η γλώσσα και οι κανόνες επικοινωνίας ορίζονται

σε ένα πρωτόκολλο επικοινωνιών. Όλα τα πρωτόκολλα πελάτη-διακομιστή λειτουργούν στο επίπεδο εφαρμογής. Το πρωτόκολλο εφαρμογής στρώματος καθορίζει τα βασικά μοτίβα του διαλόγου. Για να επισημοποιήσει ακόμη περισσότερο την ανταλλαγή δεδομένων, ο διακομιστής μπορεί να εφαρμόσει μια διεπαφή προγραμματισμού εφαρμογών (API). Το API είναι ένα επίπεδο αφαίρεσης για την πρόσβαση σε μια υπηρεσία. Περιορίζοντας την επικοινωνία σε μια συγκεκριμένη μορφή περιεχομένου, διευκολύνει την ανάλυση. Απελευθερώνοντας την πρόσβαση, διευκολύνει την ανταλλαγή δεδομένων μεταξύ των πλατφορμών .

Ένας διακομιστής μπορεί να λάβει αιτήματα από πολλούς διαφορετικούς πελάτες σε σύντομο χρονικό διάστημα. Ένας υπολογιστής μπορεί να εκτελέσει μόνο ένα περιορισμένο αριθμό εργασιών ανά πάσα στιγμή και βασίζεται σε ένα σύστημα προγραμματισμού για να δώσει προτεραιότητα σε εισερχόμενες αιτήσεις από πελάτες για να τους φιλοξενήσει. Για να αποτρέψετε την κατάχρηση και τη μεγιστοποίηση της διαθεσιμότητας, το λογισμικό διακομιστή ενδέχεται να περιορίσει τη διαθεσιμότητα σε πελάτες. Οι επιθέσεις άρνησης εξυπηρέτησης έχουν σχεδιαστεί για να εκμεταλλεύονται την υποχρέωση ενός διακομιστή να επεξεργάζεται αιτήματα, υπερφορτώντας τον με υπερβολικά ποσοστά αιτήσεων.

HTML Hyper Text Markup Language

Η HTML (αρχικοποίηση του αγγλικού HyperText Markup Language, Γλώσσα Σήμανσης Υπερκειμένου) είναι η κύρια γλώσσα σύμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων.

Η HTML γράφεται υπό μορφή στοιχείων `_HTML` τα οποία αποτελούνται από ετικέτες (tags), οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα `<html>`), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα `<h1>` και `</h1>`), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης

(ή σε άλλες περιπτώσεις ετικέτα ανοίγματος και ετικέτα κλεισίματος αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ.

Ο σκοπός ενός web browser είναι να διαβάζει τα έγγραφα HTML και τα συνθέτει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας.

Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους του ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML.

Οι Web browsers μπορούν επίσης να αναφέρονται σε στυλ μορφοποίησης CSS για να ορίζουν την εμφάνιση και τη διάταξη του κειμένου και του υπόλοιπου υλικού. Ο οργανισμός W3C, ο οποίος δημιουργεί και συντηρεί τα πρότυπα για την HTML και τα CSS, ενθαρρύνει τη χρήση των CSS αντί διαφόρων στοιχείων της HTML για σκοπούς παρουσίασης του περιεχομένου.

Προέλευση

Το 1980, ο φυσικός Τιμ Μπέρνερς Λι, ο οποίος εργαζόταν στο CERN, επινόησε το ENQUIRE, ένα σύστημα χρήσης και διαμοιρασμού εγγράφων για τους ερευνητές του CERN, και κατασκεύασε ένα πρωτότυπό του. Αργότερα, το 1989, πρότεινε ένα σύστημα βασισμένο στο διαδίκτυο, το οποίο θα χρησιμοποιούσε υπερκείμενο. Έτσι, έφτιαξε την προδιαγραφή της HTML και έγραψε τον browser και το λογισμικό εξυπηρετητή στα τέλη του 1990. Τον ίδιο χρόνο, ο Μπέρνερς Λι και ο μηχανικός συστημάτων πληροφορικής του CERN Robert Cailliau συνεργάστηκαν σε μια κοινή

προσπάθεια εύρεσης χρηματοδότησης, αλλά το έργο δεν υιοθετήθηκε ποτέ επίσημα από το CERN. Στις προσωπικές του σημειώσεις από το 1990, ο Μπέρνερς Λι αριθμεί «μερικές από τις πολλές χρήσεις του υπερκειμένου», και αναφέρει πρώτα από όλες μια εγκυκλοπαίδεια

Πρώτες προδιαγραφές

Η πρώτη δημόσια διαθέσιμη περιγραφή της HTML ήταν ένα έγγραφο με το όνομα Ετικέτες HTML, το οποίο πρωτοαναφέρθηκε στο Διαδίκτυο από τον Μπέρνερς Λι στα τέλη του 1991. Περιέγραφε τα 20 στοιχεία τα οποία αποτελούσαν τον αρχικό και σχετικά απλό σχεδιασμό της HTML. Εκτός από την ετικέτα υπερσυνδέσμου, οι υπόλοιπες ήταν έντονα επηρεασμένες από την SGMLguid, μια μορφή δημιουργίας τεκμηρίωσης, φτιαγμένη στο CERN και βασισμένη στην SGML. Δεκατρία από εκείνα τα αρχικά στοιχεία υπάρχουν ακόμα σήμερα στην HTML .

Το ίδιο το πρότυπο SGML αναπαράγει μερικές από τις τεχνικές των τυπογράφων, αλλά εκτός από απλή μίμηση της τυπογραφίας προσθέτει γενικευμένη σήμανση βασισμένη σε στοιχεία, τα οποία μπορούν να εμφωλεύονται το ένα μέσα στο άλλο και να φέρουν ιδιότητες. Ακόμα, το SGML διαχωρίζει τη δομή από το περιεχόμενο, κατεύθυνση προς την οποία αργότερα κινήθηκε και η HTML, με τα CSS. Πολλά από τα στοιχεία κειμένου προέρχονται από την τεχνική αναφορά ISO TR 9537, Techniques for using SGML (τεχνικές χρήσης της SGML), η οποία με τη σειρά της καλύπτει τα χαρακτηριστικά των πρώιμων γλωσσών μορφοποίησης κειμένου που χρησιμοποιούνταν από τα TYPSET και RUNOFF, και είχαν αναπτυχθεί στις αρχές της δεκαετίας του 1960 για το λειτουργικό σύστημα CTSS.

Ο Μπέρνερς Λι θεώρησε την HTML ως μια υλοποίηση του SGML. Αυτό ορίστηκε και επίσημα από το Internet Engineering Task Force (IETF) με τη δημοσίευση της πρώτης πρότασης για μια προδιαγραφή της HTML, στα μέσα του 1993, η οποία περιλάμβανε και έναν Ορισμός τύπου εγγράφου (DTD, Document Type Definition) της SGML, ο οποίος όριζε την γραμματική. Αυτό το πρόχειρο έληξε μετά την πάροδο έξι μηνών, αλλά περιέχει κάτι αξιοσημείωτο: την αναγνώριση της ετικέτας του NCSA Mosaic για την ενσωμάτωση εικόνων μέσα στο κείμενο, η οποία αντικατοπτρίζει την φιλοσοφία του IETF για ενσωμάτωση επιτυχημένων πρωτότυπων μέσα στα πρότυπα. Κάτι παρόμοιο περιείχε και το ανταγωνιστικό

πρόχειρο του Dave Raggett, «HTML+ (Hypertext Markup Format)», από τα τέλη του, το οποίο πρότεινε την προτυποποίηση μερικών ήδη υλοποιημένων δυνατοτήτων, όπως οι πίνακες και οι φόρμες.

Μετά που τα πρόχειρα HTML και HTML+ έληξαν, στις αρχές του 1994, το IETF δημιούργησε την Ομάδα Εργασίας για την HTML, η οποία το 1995 ολοκλήρωσε την «HTML 2.0», με την πρόθεση να αποτελέσει την πρώτη προδιαγραφή πάνω στην οποία θα βασίζονταν οι μελλοντικές υλοποιήσεις. Η HTML 2.0 δημοσιεύτηκε ως RFC 1866, και περιείχε ιδέες από τα πρόχειρα HTML και HTML . Η αρίθμηση 2.0 σκόπευε απλά να ξεχωρίσει την νέα έκδοση από τα πρόχειρα που προηγήθηκαν. designation was intended to distinguish the new edition from previous drafts.

Η περαιτέρω ανάπτυξη κάτω από την επίβλεψη του IETF καθυστέρησε λόγω σύγκρουσης ενδιαφερόντων. Από το 1996 και μετά, οι προδιαγραφές της HTML τηρούνται, μαζί με ανάδραση από τους δημιουργούς λογισμικού, από το World Wide Consortium (W3C). Ωστόσο, το 2000 η HTML έγινε επίσης παγκόσμιο πρότυπο (ISO/IEC 15445:2000). Η τελευταία προδιαγραφή της HTML, η HTML 4.01 δημοσιεύτηκε από το W3C το 1999, και το 2001 δημοσιεύτηκαν επίσης και τα λάθη και οι παραλείψεις της (errata).

Dynamic web page (Τι είναι οι δυναμικές σελίδες)

Μια δυναμική διαδικτυακή σελίδα διακομιστή είναι μια ιστοσελίδα ή κατασκευή της οποίας ελέγχεται από ένα σενάριο επεξεργασίας διακομιστή εφαρμογών. Στο scripting από την πλευρά του διακομιστή, οι παράμετροι καθορίζουν τον τρόπο με τον οποίο προχωράει η συναρμολόγηση κάθε νέας ιστοσελίδας, συμπεριλαμβανομένης της δημιουργίας περισσότερης επεξεργασίας από την πλευρά του πελάτη. Μια δυναμική ιστοσελίδα από την πλευρά του πελάτη επεξεργάζεται την ιστοσελίδα χρησιμοποιώντας scripting HTML που εκτελείται στο πρόγραμμα περιήγησης καθώς φορτώνεται. Το JavaScript και άλλες γλώσσες δέσμης ενεργειών καθορίζουν τον τρόπο με τον οποίο αναλύεται η HTML στην λαμβανόμενη σελίδα

στο Μοντέλο αντικειμένου εγγράφου ή το DOM (Document Object Model) που αντιπροσωπεύει την φορτωμένη ιστοσελίδα. Οι ίδιες τεχνικές από την πλευρά του πελάτη μπορούν στη συνέχεια να ενημερώσουν δυναμικά ή να αλλάξουν το DOM με τον ίδιο τρόπο. Στη συνέχεια, μια δυναμική ιστοσελίδα επαναφορτώνεται από το χρήστη ή από ένα πρόγραμμα υπολογιστή για να αλλάξει κάποιο μεταβλητό περιεχόμενο. Οι πληροφορίες ενημέρωσης θα μπορούσαν να προέρχονται από το διακομιστή ή από τις αλλαγές που έγιναν στο DOM της συγκεκριμένης σελίδας. Αυτό μπορεί να περιορίσει το ιστορικό περιήγησης ή να δημιουργήσει μια αποθηκευμένη έκδοση για να επιστρέψει, αλλά μια δυναμική ενημέρωση διαδικτυακής σελίδας που χρησιμοποιεί τεχνολογίες Ajax δεν θα δημιουργήσει ούτε μια σελίδα για να επιστρέψει ούτε να περικόψει το ιστορικό περιήγησης στο Web προς τα εμπρός της εμφανιζόμενης σελίδας. Χρησιμοποιώντας τις τεχνολογίες Ajax, ο τελικός χρήστης αποκτά μία δυναμική σελίδα που διαχειρίζεται ως μία σελίδα στο πρόγραμμα περιήγησης ιστού, ενώ το πραγματικό περιεχόμενο ιστού που εμφανίζεται σε αυτήν τη σελίδα μπορεί να διαφέρει. Ο μηχανισμός Ajax βρίσκεται μόνο στο πρόγραμμα περιήγησης που ζητά τμήματα του DOM του, DOM, για τον πελάτη του, από διακομιστή εφαρμογών. Το DHTML είναι ο ομπρέλα για τεχνολογίες και μεθόδους που χρησιμοποιούνται για τη δημιουργία ιστοσελίδων που δεν είναι στατικές ιστοσελίδες. Η δημιουργία σεναρίων από πλευράς πελάτη, η δημιουργία σε δέσμες ενεργειών από πλευράς διακομιστή ή ένας συνδυασμός αυτών για τη δυναμική εμπειρία ιστού σε ένα πρόγραμμα περιήγησης.

Ajax

Το Ajax είναι ένα σύνολο τεχνικών ανάπτυξης ιστού που χρησιμοποιεί πολλές τεχνολογίες ιστού στην πλευρά του πελάτη για τη δημιουργία ασύγχρονων εφαρμογών Web. Με το Ajax, οι εφαρμογές Web μπορούν να αποστέλλουν και να ανακτούν δεδομένα από έναν διακομιστή ασύγχρονα (στο παρασκήνιο) χωρίς να παρεμβαίνουν στην εμφάνιση και τη συμπεριφορά της υπάρχουσας σελίδας. Με την αποσύνδεση του στρώματος ανταλλαγής δεδομένων από το στρώμα παρουσίασης, το Ajax επιτρέπει σε ιστοσελίδες και κατ'επέκταση σε εφαρμογές Web να αλλάζουν δυναμικά το περιεχόμενο χωρίς να χρειάζεται να επαναφορτώσουν ολόκληρη τη σελίδα. Στην πράξη, οι σύγχρονες εφαρμογές χρησιμοποιούν συνήθως JSON, λόγω

των πλεονεκτημάτων του JSON που είναι εγγενής στη JavaScript. Ajax δεν είναι μία τεχνολογία, αλλά μια ομάδα τεχνολογιών. Το HTML και το CSS μπορούν να χρησιμοποιηθούν σε συνδυασμό για να επισημάνουν πληροφορίες επάνω και στυλ. Η ιστοσελίδα μπορεί στη συνέχεια να τροποποιηθεί από τη JavaScript για να εμφανιστεί δυναμικά - και να επιτρέψει στο χρήστη να αλληλεπιδράσει με - τις νέες πληροφορίες. Το ενσωματωμένο αντικείμενο XMLHttpRequest μέσα στο JavaScript χρησιμοποιείται συνήθως για την εκτέλεση του Ajax σε ιστοσελίδες, επιτρέποντας στους ιστότοπους να φορτώνουν περιεχόμενο στην οθόνη χωρίς να ανανεώνουν τη σελίδα. Το Ajax δεν είναι μια νέα τεχνολογία, ή διαφορετική γλώσσα, απλώς υπάρχουσες τεχνολογίες που χρησιμοποιούνται με νέους τρόπους.

JSON

Το JSON (JavaScript Object Notation) είναι ένα ελαφρύ πρότυπο ανταλλαγής δεδομένων. Είναι εύκολο για τους ανθρώπους να το διαβάσουν και γράψουν. Είναι εύκολο για τις μηχανές να το αναλύσουν (parse) και να το παράγουν (generate). Είναι βασισμένο πάνω σε ένα υποσύνολο της γλώσσας προγραμματισμού JavaScript, Standard ECMA-262 Έκδοση 3η - Δεκέμβριος 1999. Το JSON είναι ένα πρότυπο κειμένου το οποίο είναι τελείως ανεξάρτητο από γλώσσες προγραμματισμού αλλά χρησιμοποιεί πρακτικές (conventions) οι οποίες είναι γνωστές στους προγραμματιστές της οικογένειας προγραμματισμού C, συμπεριλαμβανομένων των C, C++, C#, Java, JavaScript, Perl, Python, και πολλών άλλων. Αυτές οι ιδιότητες κάνουν το JSON μια ιδανική γλώσσα προγραμματισμού ανταλλαγής δεδομένων.

Το JSON είναι χτισμένο σε δύο δομές:

- Μια συλλογή από ζευγάρια ονομάτων/τιμών. Σε διάφορες γλώσσες προγραμματισμού, αυτό αντιλαμβάνεται ως ένα object, καταχώριση, δομή, λεξικό, πίνακα hash (hash table), λίστα κλειδιών, ή associative πίνακα.
- Μία ταξινομημένη λίστα τιμών. Στις περισσότερες γλώσσες προγραμματισμού, αυτό αντιλαμβάνεται ως ένας πίνακας (array), διάνυσμα, λίστα, ή ακολουθία.

Αυτά είναι τα universal data structures. Ουσιαστικά όλες οι μοντέρνες γλώσσες προγραμματισμού τα υποστηρίζουν με τον έναν ή τον άλλον τρόπο. Λογικό είναι πως ένα πρότυπο δεδομένων το οποίο είναι εύκολα μεταβαλλόμενο με γλώσσες προγραμματισμού οι οποίες επίσης είναι βασισμένες σε αυτές τις δομές.

CSS - Cascading Style Sheets

Τα Cascading Style Sheets (CSS) είναι ένας μηχανισμός για τη διαμόρφωση του περιεχομένου των εγγράφων που είναι γραμμένα σε HTML ή XML, εφαρμόζοντας στυλ διαμόρφωσης σε τύπους στοιχείων ή κλάσεων καθορισμένων από τον συντάκτη του εγγράφου ή συγκεκριμένες περιπτώσεις αυτών των στοιχείων. Τα Cascading style sheets είναι εκείνη η τεχνολογία που επιτρέπει στον browser να λάβει οδηγίες σχετικά με τη μορφή του περιεχομένου μιας ιστοσελίδας. Οι οδηγίες αυτές μπορεί να είναι ενσωματωμένες στην επικεφαλίδα του αρχείου html ή μπορεί να είναι ένα αρχείο ξεχωριστό από την ιστοσελίδα που περιγράφει τον τρόπο με τον οποίο θα παρουσιαστεί η πληροφορία μέσα στην ιστοσελίδα. Έτσι γίνεται διαχωρισμός του περιεχομένου της ιστοσελίδας από τον τρόπο παρουσίασής του και επίσης διασφαλίζεται ο ενιαίος τρόπος με τον οποίο θα εμφανίζεται το περιεχόμενο μέσα από οποιονδήποτε browser. Υποστηρίχτηκαν από τον φορέα W3C έτσι ώστε η διαμόρφωση των HTML σελίδων να στηρίζεται σε stylesheets, συμβάλλοντας έτσι σε ένα πιο απλό και σωστά δομημένο Παγκόσμιο Ιστό.

JavaScript

Η JavaScript, συχνά συντομευμένο ως JS, είναι μια υψηλού επιπέδου, ερμηνευμένη γλώσσα προγραμματισμού. Είναι μια γλώσσα που χαρακτηρίζεται επίσης ως δυναμική, ασθενώς δακτυλογραφημένη, βασισμένη σε πρωτότυπο και πολυ-παραδειγματική. Παράλληλα με την HTML και την CSS, η JavaScript είναι μία από τις τρεις βασικές τεχνολογίες της τεχνολογίας περιεχομένου στον Παγκόσμιο Ιστό. Χρησιμοποιείται για να κάνει ιστοσελίδες διαδραστικό και να παρέχει online προγράμματα, συμπεριλαμβανομένων των βιντεοπαιχνιδιών. Η πλειοψηφία των ιστότοπων την χρησιμοποιούν και όλα τα σύγχρονα προγράμματα περιήγησης στο Web την υποστηρίζουν χωρίς την ανάγκη για plug-ins μέσω ενσωματωμένης μηχανής JavaScript. Κάθε ένας από τους πολλούς μηχανισμούς JavaScript αντιπροσωπεύει μια διαφορετική υλοποίηση του JavaScript, με βάση την προδιαγραφή ECMAScript, με κάποιους κινητήρες που δεν υποστηρίζουν πλήρως το spec και με πολλούς κινητήρες που υποστηρίζουν πρόσθετα χαρακτηριστικά πέρα

από την ECMA. Ως γλώσσα πολλαπλών παραδειγμάτων, η JavaScript υποστηρίζει στυλ προγραμματισμού που βασίζονται σε γεγονότα, λειτουργικά και επιτακτικά (συμπεριλαμβανομένων αντικειμενοστραφών και πρωτότυπων). Διαθέτει API για εργασία με κείμενο, συστοιχίες, ημερομηνίες, κανονικές εκφράσεις και βασικούς χειρισμούς του DOM, αλλά η ίδια η γλώσσα δεν περιλαμβάνει I / O, όπως εγκαταστάσεις δικτύωσης, αποθήκευσης ή γραφικών, βασιζόμενη σε αυτές περιβάλλον στο οποίο είναι ενσωματωμένο. Αρχικά εφαρμοζόταν μόνο από πλευράς πελάτη σε προγράμματα περιήγησης ιστού, οι μηχανές JavaScript ενσωματώνονται πλέον σε πολλούς άλλους τύπους λογισμικού κεντρικού υπολογιστή, συμπεριλαμβανομένου του διακομιστή σε διακομιστές και βάσεις δεδομένων ιστού και σε προγράμματα εκτός του Διαδικτύου, όπως επεξεργαστές κειμένου και λογισμικό PDF, και σε χρόνο εκτέλεσης περιβάλλοντα που καθιστούν το JavaScript διαθέσιμο για τη σύνταξη εφαρμογών για κινητά και επιτραπέζιους υπολογιστές, συμπεριλαμβανομένων γραφικών γραφικών.

Προγραμματισμός Εφαρμογής σε επίπεδο Εξυπηρετητής (server)-Side

Πρόκειται για τεχνολογίες προγραμματισμού και ανάπτυξης εφαρμογών οι οποίες εκτελούνται στην μεριά του web server πριν το περιεχόμενο αποσταλεί στον browser του τελικού χρήστη.

- PHP. Η PHP είναι η πιο διαδεδομένη γλώσσα script που εκτελείται σε επίπεδο εξυπηρετητής (server) και χρησιμοποιείται στη δημιουργία ιστοσελίδων δυναμικού περιεχομένου. Είναι γλώσσα ανοικτού κώδικα.
- ASP. Η ASP είναι μια επίσης διαδεδομένη γλώσσα σε επίπεδο εξυπηρετητής (server) που αναπτύχθηκε από τη Microsoft και στηρίζεται στη χρήση της γλώσσας VBscript για τη δημιουργία ιστοσελίδων δυναμικού περιεχομένου.
- MySQL. Η MySQL είναι ένα ισχυρό Σύστημα Διαχείρισης Βάσεων Δεδομένων ανοικτού κώδικα που συνδυάζεται συνήθως με PHP και υποστηρίζεται από όλα τα λειτουργικά συστήματα.

Η πλατφόρμα ανάπτυξης PHP και MySQL αποτελούν πλέον τις πιο διαδεδομένες τεχνολογίες ανάπτυξης δυναμικών εφαρμογών σε επίπεδο Εξυπηρετητή.

PHP

Η PHP είναι μια γλώσσα προγραμματισμού για διακομιστές που έχει σχεδιαστεί για ανάπτυξη ιστού, αλλά χρησιμοποιείται επίσης ως γλώσσα προγραμματισμού γενικής χρήσης . Αρχικά δημιουργήθηκε από τον Rasmus Lerdorf το 1994, η υλοποίηση της αναφοράς PHP δημιουργείται τώρα από το The PHP Group . Η PHP αρχικά ήταν για

την προσωπική αρχική σελίδα , αλλά τώρα σημαίνει το αναδρομικό ακρωνύμιο PHP: Hypertext Preprocessor. Ο κώδικας PHP μπορεί να ενσωματωθεί σε κώδικα HTML ή μπορεί να χρησιμοποιηθεί σε συνδυασμό με διάφορα συστήματα πρότυπων ιστού, συστήματα διαχείρισης περιεχομένου ιστού και πλαίσια ιστού. Ο κώδικας PHP επεξεργάζεται συνήθως από έναν ερμηνευτή PHP που υλοποιείται ως ενότητα στον εξυπηρετητή ιστού ή ως εκτελέσιμος διεπαφής Common Gateway Interface (CGI). Ο διακομιστής ιστού συνδυάζει τα αποτελέσματα του ερμηνευμένου και εκτελεσθέντος κώδικα PHP, ο οποίος μπορεί να είναι οποιοσδήποτε τύπος δεδομένων, συμπεριλαμβανομένων των εικόνων, με την παραγόμενη ιστοσελίδα. Ο κώδικας PHP μπορεί επίσης να εκτελεστεί με μια διεπαφή γραμμής εντολών (CLI) και μπορεί να χρησιμοποιηθεί για την υλοποίηση αυτόνομων γραφικών εφαρμογών. Ο τυποποιημένος διερμηνέας PHP, που τροφοδοτείται από το Zend Engine, είναι ελεύθερο λογισμικό που εκδίδεται υπό την άδεια PHP. Η PHP έχει μεταφερθεί ευρέως και μπορεί να αναπτυχθεί στους περισσότερους διακομιστές ιστού σε σχεδόν κάθε λειτουργικό σύστημα και πλατφόρμα, δωρεάν. Η γλώσσα PHP εξελίχθηκε χωρίς γραπτή επίσημη προδιαγραφή ή πρότυπο μέχρι το 2014, αφήνοντας τον κανονικό διερμηνέα PHP ως de facto πρότυπο. Από το 2014 οι εργασίες συνεχίστηκαν για τη δημιουργία μιας επίσημης προδιαγραφής PHP. Κατά τη διάρκεια της δεκαετίας του 2010 σημειώθηκαν αυξημένες προσπάθειες για την τυποποίηση και την κοινή χρήση κωδικών σε εφαρμογές PHP από έργα όπως η PHP-FIG υπό τη μορφή πρωτοβουλιών PSR, καθώς και ο διαχειριστής εξάρτησης Composer και ο χώρος αποθεματοποίησης των συσκευασιών.

Βασικό της χαρακτηριστικό είναι ότι οι σελίδες αυτές σχεδιάζονται δυναμικά ανάλογα με την εκτέλεση του κώδικα. Τα βασικά πλεονεκτήματα των PHP σελίδων είναι τα εξής:

- Είναι πολύ εύκολη στην εκμάθηση της. Ο προγραμματισμός σε PHP είναι οικείος σε προγραμματιστές C, Perl και Java
- Υποστηρίζει πολλές πλατφόρμες (Windows, Linux, Unix, κτ)
- Υπάρχει συμβατότητα με σχεδόν όλους τους εξυπηρετητές (server)s (Apache, IIS, κτ)

- Παρέχει εύκολη συνδεσιμότητα με Βάσεις Δεδομένων όπως MySQL, Oracle, Sybase, PostgreSQL, Generic ODBC κα.
- Παρέχει ενσωματωμένες βιβλιοθήκες για πολλές Web διεργασίες όπως να δημιουργήσει δυναμικά αρχεία εικόνων, να αναλύσει XML, να στείλει ηλεκτρονικό μήνυμα, να δημιουργήσει cookies και PDF έγγραφα
- Ανήκει στην κατηγορία του Λογισμικού Ανοικτού Κώδικα (Open Source software – OSS).
- Συνεργάζεται με την επίσης Ανοικτού Κώδικα βάση Δεδομένων MySQL.
- Είναι διαθέσιμη δωρεάν <http://www.php.net>. Η χρήση είναι δωρεάν.

Τι είναι το framework και γιατί χρειάζεται ;

Ένα framework είναι μια συλλογή καθολικών, επαναχρησιμοποιήσιμων προγραμμάτων που επιτελούν συγκεκριμένα καθήκοντα για να διευκολύνουν την ανάπτυξη εφαρμογών λογισμικού.

Μεταφορικά, είναι ένας σκελετός, σκελετός ή υπόγειο για μια εφαρμογή στο διαδίκτυο ή την ανάπτυξη ιστότοπου. Με τον ίδιο τρόπο που αγοράζετε ένα άδειο σπίτι και το γεμίζετε με έπιπλα, οι προγραμματιστές παίρνουν ένα πλαίσιο και το γεμίζουν με ενότητες χαρακτηριστικών.

Η χρήση ενός framework δεν είναι απαραίτητο, αλλά είναι μια καλή πρακτική απλούστερη τη ζωή του κωδικοποιητή. Οι λόγοι για τη χρήση ενός πλαισίου είναι:

- Ταχύτερη ανάπτυξη. Μπορείτε να αποθηκεύσετε εργασίες δύο ή τριών ημερών κατά την επαναχρησιμοποίηση γενικών εξαρτημάτων που εξέρχονται . Ο αποθηκευμένος χρόνος μπορεί να αφιερωθεί στην ανάπτυξη συγκεκριμένων χαρακτηριστικών.

- Ευκολότερη συντήρηση. Η χρήση ενός framework επιβάλλει μια συγκεκριμένη δομή και κοινούς κανόνες που πρέπει να ακολουθηθούν. Έτσι, δεν θα είναι ένα πρόβλημα για έναν νέο προγραμματιστή ή ομάδα να συμμετάσχει στο τρέχον έργο.
- Λιγότερα θέματα. Ο κώδικας ενός framework είναι σαφής και αποδεδειγμένος, πράγμα που μειώνει τον αριθμό των λαθών.
- Καλύτερη ασφάλεια. Ένα framework υποστηρίζεται με έγκαιρες αναβαθμίσεις και ενίσχυση της ασφάλειας

Symfony

Το Symfony κυκλοφόρησε το 2005 και έχει κάνει πολύ δρόμο για να κερδίσει τη φήμη του πιο ώριμου και αξιόπιστου framework PHP. Είναι η πρώτη επιλογή για τις εφαρμογές web για επιχειρήσεις. Και εδώ είναι ο λόγος:

Υψηλή ελαστικότητα .Το Symfony είναι ένα από τα πιο πλούσια σε χαρακτηριστικά πλέγματα PHP. Τα δύο πιο σημαντικά τεχνολογικά οφέλη της Symfony που το κάνουν τόσο φοβερό είναι τα και Components. Το πακέτο είναι σχεδόν το ίδιο με ένα plugin. Θεωρείτε σαν ένα πακέτο αρχείων (αρχεία PHP, φύλλα στυλ, JavaScript, εικόνες) για την εφαρμογή οποιουδήποτε χαρακτηριστικού (για παράδειγμα, ένα blog, ένα καλάθι αγορών κ.λπ.). Το κύριο όφελος των δεσμών είναι ότι αποσυνδέονται. Μπορείτε να τα επαναπροσδιορίσετε και να τα επαναχρησιμοποιήσετε για πολλές εφαρμογές για να μειώσετε το συνολικό κόστος ανάπτυξης.

Τα συστατικά στοιχεία είναι γενικές λειτουργίες που μειώνουν τις καθημερινές εργασίες και επιτρέπουν στους προγραμματιστές να εστιάζουν σε συγκεκριμένες επιχειρηματικές λειτουργίες. Υπάρχουν 30 χρήσιμα στοιχεία Symfony που διευκολύνουν τη διαδικασία ανάπτυξης. Μπορείτε να χρησιμοποιήσετε εξαρτήματα

ανεξάρτητα και να προσθέσετε τις δικές σας προσαρμοσμένες ενότητες χωρίς προβλήματα στην αρχιτεκτονική. Τα στοιχεία Symfony μπορούν επίσης να χρησιμοποιηθούν αυτόνομα σε άλλα πλαίσια (για παράδειγμα, Laravel) ή σε απλές λύσεις PHP. Οι δέσμες και τα εξαρτήματα βοηθούν στην εξάλειψη των αυστηρών εξαρτήσεων της αρχιτεκτονικής.

Τι είναι το Autocomplete

Η αυτόματη ολοκλήρωση ή η ολοκλήρωση λέξεων είναι μια λειτουργία στην οποία μια εφαρμογή προβλέπει την υπόλοιπη λέξη που πληκτρολογεί ένας χρήστης. Στις γραφικές διεπαφές χρηστών, οι χρήστες μπορούν συνήθως να πιέσουν το πλήκτρο καρτελών για να αποδεχθούν μια πρόταση ή το κάτω βέλος για να αποδεχτούν ένα από τα πολλά. Η αυτόματη συμπλήρωση επιταχύνει τις αλληλεπιδράσεις ανθρώπου-υπολογιστή όταν προβλέπει σωστά τη λέξη που σκοπεύει να εισάγει ένας χρήστης αφού έχουν εισαχθεί μόνο μερικοί χαρακτήρες σε ένα πεδίο εισαγωγής κειμένου. Λειτουργεί καλύτερα σε τομείς με περιορισμένο αριθμό πιθανών λέξεων (όπως σε διερμηνείς γραμμής εντολών), όταν κάποιες λέξεις είναι πολύ συχνότερες (όπως όταν απευθύνονται σε ένα ηλεκτρονικό ταχυδρομείο) ή γράφοντας δομημένο και προβλέψιμο κείμενο (όπως στον πηγαίο κώδικα εκδότες). Πολλοί αλγόριθμοι αυτόματης συμπλήρωσης μαθαίνουν νέες λέξεις αφού ο χρήστης τις έχει γράψει μερικές φορές και μπορεί να προτείνει εναλλακτικές λύσεις που βασίζονται στις μαθησιακές συνήθειες του μεμονωμένου χρήστη.

Τι είναι το twig

Το Twig χρησιμοποιεί ένα κεντρικό αντικείμενο που ονομάζεται περιβάλλον (της κλάσης `Twig_Environment`). Οι παρουσίες αυτής της κλάσης χρησιμοποιούνται για την αποθήκευση των παραμέτρων και των επεκτάσεων και χρησιμοποιούνται για τη φόρτωση προτύπων από το σύστημα αρχείων ή από άλλες τοποθεσίες.

Οι περισσότερες εφαρμογές θα δημιουργήσουν ένα αντικείμενο `Twig_Environment` στην αρχικοποίηση εφαρμογών και θα το χρησιμοποιήσουν για να φορτώσουν πρότυπα. Σε ορισμένες περιπτώσεις, είναι χρήσιμο να υπάρχουν πολλαπλά περιβάλλοντα δίπλα-δίπλα, εάν χρησιμοποιούνται διαφορετικές διαμορφώσεις.

Βάση Δεδομένων MySQL

Η web εφαρμογή χρησιμοποιεί συνήθως μια βάση δεδομένων για την καταχώρηση και ανάκτηση δεδομένων. Στην αρχιτεκτονική πελάτης - εξυπηρετητής υπάρχει ένα σύστημα Διαχείρισης Βάσης Δεδομένων συνήθως Σχεσιακής (Relational Database System - RDBMS) όπου καταχωρούνται τα δεδομένα. Ανάλογα με τις ενέργειες και τις αιτήσεις του χρήστη, ο εξυπηρετητής (server) επικοινωνεί με το σύστημα διαχείρισης της βάσης δεδομένων και εκτελεί ερωτήματα (queries) σε γλώσσα SQL.

Το σύστημα διαχείρισης της Βάσης Δεδομένων με τη σειρά του απαντάει σε αυτά τα queries του εξυπηρετητή είτε αποστέλλοντας τα δεδομένα που προέκυψαν σαν αποτελέσματα των ερωτημάτων ή εκτελώντας κάποια εισαγωγή ή διαγραφή δεδομένων στην περίπτωση εντολών `insert` ή `update` ή `delete`. Η επικοινωνία μεταξύ εφαρμογής και Βάσης Δεδομένων γίνεται με τη χρήση οδηγών (Database Connectivity drivers).

Στη συγκεκριμένη εφαρμογή χρησιμοποιείται η MySQL. Η MySQL είναι ένα Σύστημα Διαχείρισης Σχεσιακής Βάσης Δεδομένων και περιέχει και έναν μικρό server της βάσης. Είναι ένα πολύ γρήγορο δυνατό, σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων. Ο διακομιστής MySQL ελέγχει την πρόσβαση στα δεδομένα για να διασφαλίσει ότι πολλοί χρήστες μπορούν να δουλεύουν ταυτόχρονα, για να παρέχει γρήγορη πρόσβαση και για να διασφαλίσει ότι μόνο εξουσιοδοτημένοι χρήστες μπορούν να έχουν πρόσβαση. Χρησιμοποιεί την SQL (Structured Query Language) γλώσσα ερωτημάτων για βάσεις δεδομένων. Η MySQL είναι διαθέσιμη από το 1996, αλλά η ιστορία της ξεκινάει από το 1979. Αναπτύχθηκε σαν μια εφαρμογή της γλώσσας SQL από την TcX. Είναι αρκετά σταθερό σύστημα και πολύ ευέλικτο. Υποστηρίζει όλες τις λειτουργίες και τους τύπους δεδομένων της standard. Είναι πλέον παγκοσμίως η πιο δημοφιλής βάση δεδομένων ανοιχτού κώδικα και έχει κερδίσει αρκετές φορές το βραβείο Choice Award του Linux Journal Readers. Εκτός των παραπάνω τα πιο σημαντικά χαρακτηριστικά της MySQL είναι τα ακόλουθα:

- Η MySQL ανήκει στο λογισμικό Ανοικτού Κώδικα (Open Source). Η χρήση της είναι δωρεάν
- Είναι γρήγορη και υποστηρίζει multi-thread και πολυχρηστικό περιβάλλον.
- Είναι συμβατή με πολλές και ποικίλες πλατφόρμες.

PHP – MySQL

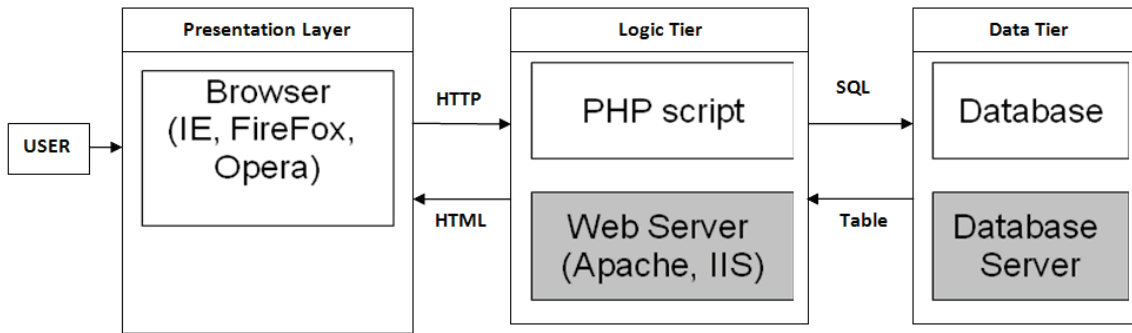
Η χρήση PHP με το σύστημα διαχείρισης της MySQL έχει σαν βασικό πλεονέκτημα τη συμβατότητα με πολλές πλατφόρμες και επίσης ότι ανήκουν και οι δύο στις εφαρμογές Ανοικτού Κώδικα και τα δικαιώματα χρήσης τους είναι δωρεάν. Εξαιτίας αυτών των χαρακτηριστικών, οι διακομιστές που φιλοξενούν δυναμικές σελίδες PHP συνδυάζουν υπηρεσίες για αυτές τις δύο πλατφόρμες και την άμεση διασύνδεσή τους. Η λειτουργία τους βασίζεται στον εκτελέσιμο κώδικα που είναι ενσωματωμένος στις PHP σελίδες. Ο κώδικας αυτός εκτελείται στον διακομιστή. Ο κώδικας εκτελεί ερωτήματα σε SQL τα οποία μεταβιβάζονται μέσω ενός middleware της MySQL στη βάση MySQL. Ανάλογα με την αίτηση του συστήματος προς τη βάση εκτελείται είτε μια εντολή ανάκτησης, ή εγγραφής ή μια τροποποίησης ή διαγραφής δεδομένων στη Βάση. Η αίτηση μεταβιβάζεται στη Βάση Δεδομένων και το σύστημα διαχείρισης δεδομένων επιστρέφει μια απάντηση στο Web server με τα αποτελέσματα. Στη συνέχεια τα δεδομένα χρησιμοποιούνται από τον εκτελέσιμο κώδικα στη δημιουργία

του HTML περιεχομένου της δυναμικής ιστοσελίδας που τελικά αποστέλλεται στην πελάτης (client) εφαρμογή δηλ. τον web browser του χρήστη όπου και προβάλλεται το περιεχόμενο που ζήτησε.

Αρχιτεκτονική 3-Tier

Το μοντέλο Πελάτης - Εξυπηρετητής (client - server) αναφέρεται και ως αρχιτεκτονική 3-tier διότι αποτελείται από τρία μέρη. Στις περιπτώσεις που τα δεδομένα αποθηκεύονται σε κάποια βάση δεδομένων σε έναν ξεχωριστό εξυπηρετητή, το μοντέλο αποτελείται από τρία διαφορετικά συστατικά. Οι εφαρμογές Πελάτης στέλνουν δεδομένα στην εφαρμογή Εξυπηρετητή η οποία αποθηκεύει τα δεδομένα σε ένα ξεχωριστό Σύστημα Διαχείρισης Βάσεων. Το ξεχωριστό αυτό σύστημα είναι υπεύθυνο για τη διαχείριση και τη διανομή των δεδομένων. Συνεπώς το συγκεκριμένο μοντέλο αποτελείται από τρία συστατικά και ονομάζεται αρχιτεκτονική 3-tier. Η προσέγγιση αυτή είναι καλύτερη από το απλό 2-tier μοντέλο επειδή απομονώνει τη λειτουργία της αποθήκευσης και διαχείρισης των δεδομένων από τη λειτουργία της επεξεργασίας τους με συνέπεια να απελευθερώνονται σημαντικοί πόροι για τον εξυπηρετητή επεξεργασίας των δεδομένων. Επιπλέον το μοντέλο 3-tier αυξάνει την συνδεσιμότητα και επεκτασιμότητα όλου του πληροφοριακού συστήματος αφού η διαχείριση των δεδομένων αποδεδμεύεται από τη συγκεκριμένη εφαρμογή και εκτελείται από ένα σύστημα που μπορεί να είναι συμβατό και διασυνδεδεμένο και με άλλες εφαρμογές [Edelstein 1994]. Η διαδικασία παρουσιάζεται στο

Σχήμα 1.



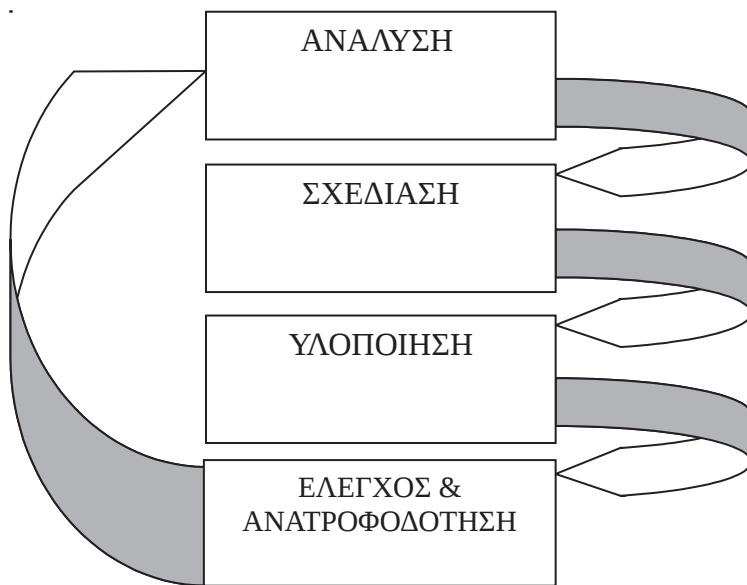
Συμπέρασμα

Η σχεδίαση της εφαρμογής θα βασιστεί στην κατάλληλη ανάλυση απαιτήσεων με βάση τους στόχους που έχουν οριστεί στην εισαγωγή. Επίσης σαν τεχνολογία ανάπτυξης της web εφαρμογής επιλέχτηκε η χρήση MySQL για τη δημιουργία της βάσης δεδομένων και της γλώσσας PHP για τη δημιουργία των δυναμικών σελίδων. Η επιλογή βασίστηκε στα πλεονεκτήματα που προσφέρει ο συνδυασμός των δύο αυτών τεχνολογιών όπως αναφέρθηκαν.

ΥΛΟΠΟΙΗΣΗ

Μεθοδολογία ανάπτυξης λογισμικού

Η μεθοδολογία που θα χρησιμοποιήσουμε για την υλοποίηση της εφαρμογής είναι με βάση το μοντέλο Καταρράκτη (waterfall) με ανατροφοδότηση όπως φαίνεται και στο Σχήμα 3.



Στην αρχική φάση γίνεται ανάλυση των απαιτήσεων των χρηστών όσον αφορά τις λειτουργίες.

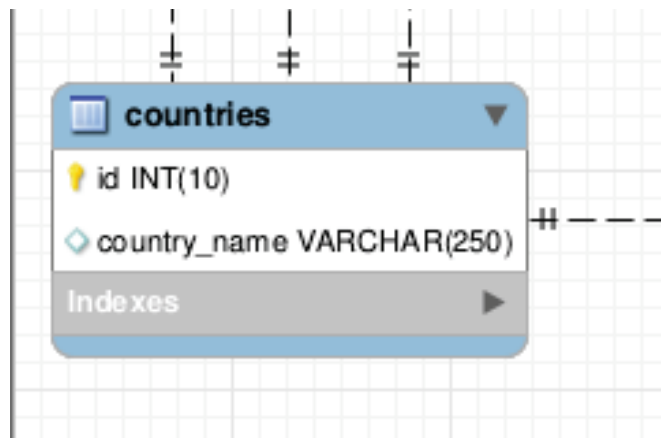
Στη φάση του σχεδιασμού γίνεται μια πρώτη προσέγγιση του τρόπου που θα υλοποιηθούν οι λειτουργίες της εφαρμογής κι επιλέγονται εκείνες οι τεχνολογίες που θεωρούνται πιο κατάλληλες. Τέλος στο στάδιο της υλοποίησης μέσα από τα κατάλληλα περιβάλλοντα ανάπτυξης εφαρμογών, γράφεται ο κώδικας, δημιουργείται η βάση δεδομένων, ενώ ταυτόχρονα γίνεται η ενοποίηση των διαφορετικών συστατικών της εφαρμογής δηλαδή της βάσης δεδομένων με τις δυναμικές ιστοσελίδες. Στην τελική φάση γίνεται έλεγχος των λειτουργιών της εφαρμογής και τυχόν συμπεράσματα χρησιμοποιούνται για ανατροφοδότηση της διαδικασίας.

Σχεδίαση Βάσης Δεδομένων

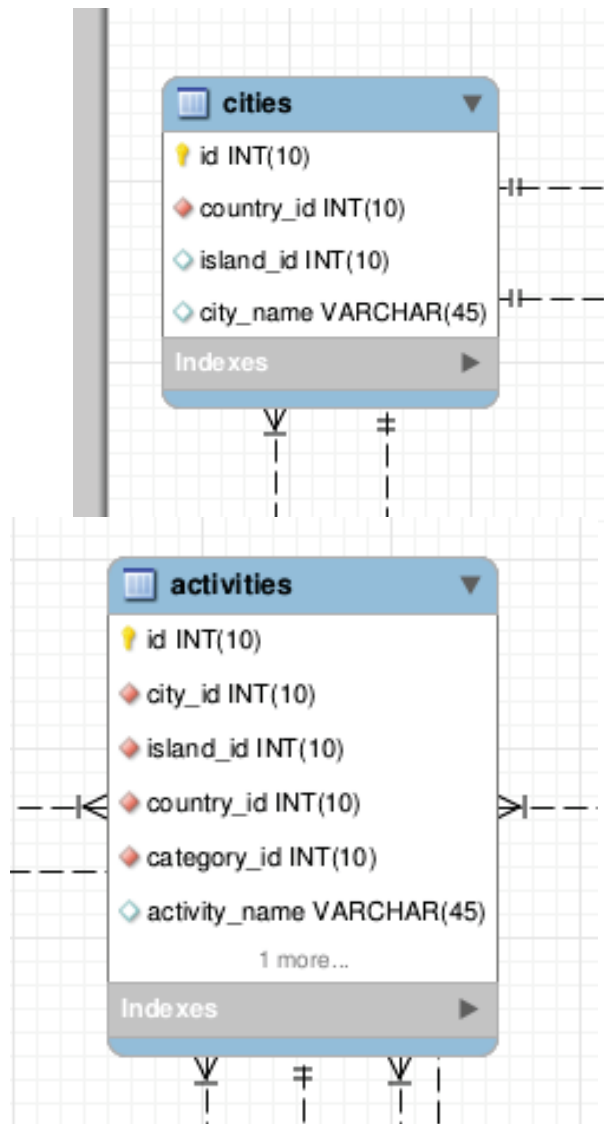
Κατόπιν ανάλυσης των απαιτήσεων των χρηστών από την εφαρμογή προκύπτει ότι τα στοιχεία που είναι απαραίτητα για την υλοποίηση του συστήματος περιλαμβάνουν τους προορισμούς τις δραστηριότητες ,τα αξιοθέατα ,τις εικόνες και την επικοινωνία για την κάθε πόλη της επιλογής του χρήστη. Όλα αυτά αποτελούν οντότητες του συστήματος και θα διατηρούνται σε μια σχεσιακή βάση δεδομένων.

Με βάση την ανάλυση των δεδομένων, οι πληροφορίες θα καταχωρούνται σε αντίστοιχους πίνακες στη βάση δεδομένων της MySQL. Οι βασικοί πίνακες είναι οι εξής:

- countries (χώρες) .Στον ακόλουθο πίνακα καταχωρούντε οι πόλεις της Ελλάδος .

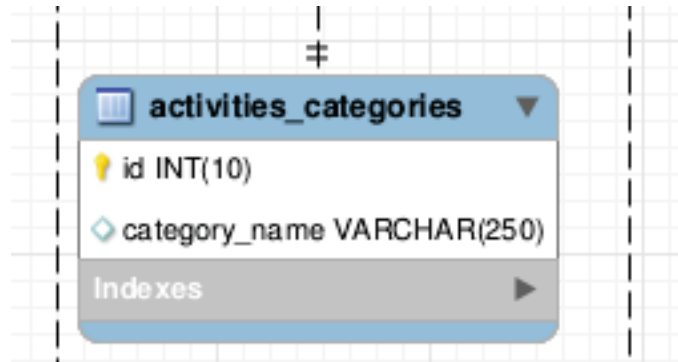


- cities (πολεις). Στον πίνακα αυτό καταχωρούνται οι πληροφορίες μιας χώρας στην περίπτωση μας της Ελλάδος . Επίσης οι πόλεις αλλά και τα νησιά διαχωρίζονται με το ανάλογο id το οποίο ανήκει η κάθε μια .

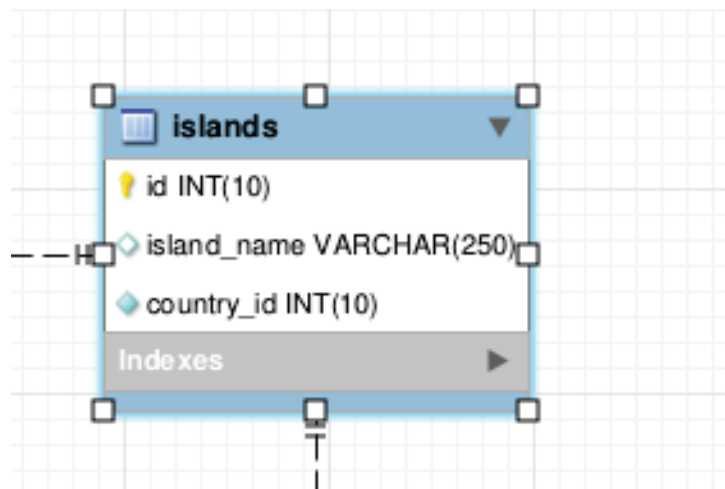


- activities (δραστηριότητες) .
 Στον παρακάτω πίνακα καταχωρούμε τις δραστηριότητες ανάλογα την πόλη το νησί αλλά και την χώρα επιλογής του χρήστη .Ο κάθε προορισμός έχει και την δική του δραστηριότητα .

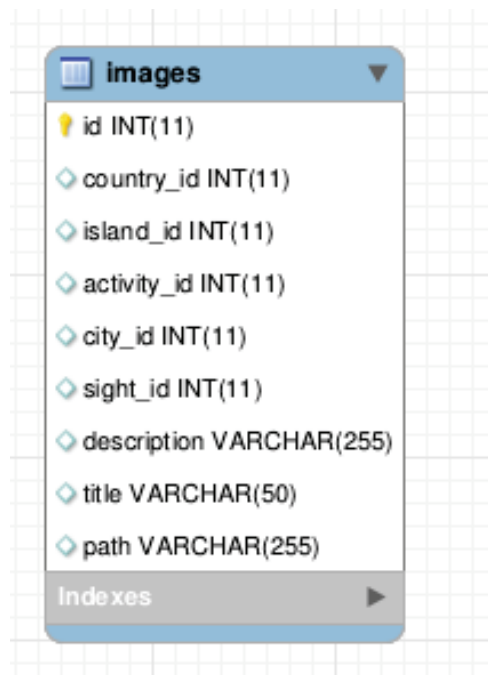
- activities_categories (κατηγορίες δραστηριοτήτων). Σε αυτόν τον πίνακα καταχωρούμε ξεχωριστά την κάθε κατηγορία δραστηριοτήτων για κάθε πόλη ή νησί.



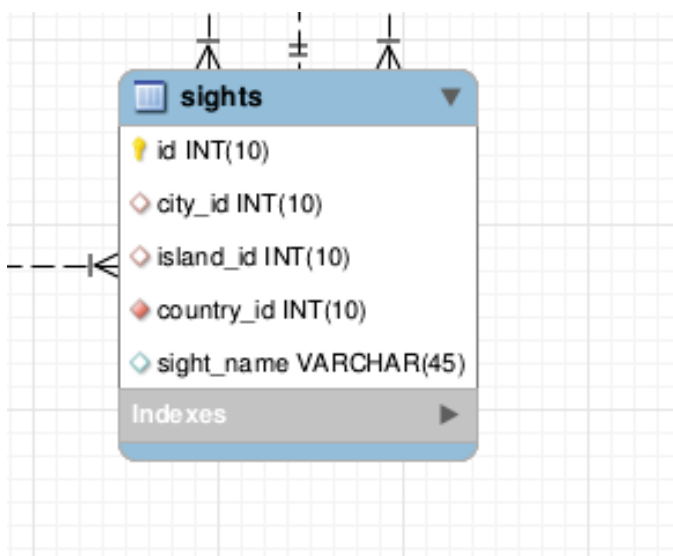
- islands (νησιά) Στο παρακάτω πίνακα καταχωρούμε τα νησιά . Επίσης γίνεται και η καταχώρηση της Κύπρου διότι στην περίπτωση μας το εκλαμβάνουμε ως νησί . Τα νησιά διαχωρίζονται ανάλογα το id της χώρας στην οποία ανήκουν .



- images (εικόνες) Στον πίνακα αυτόν γίνεται η καταχώρηση όλων των εικόνων τις κάθε χώρας , πόλης αλλά και νησιού της επιλογής του χρήστη .



- sights (αξιοθέατα) . Γίνεται η καταχώρηση για τα αξιοθέατα κάθε χώρας , νησιού αλλά και πόλης ξεχωριστά .



4^ο Κεφάλαιο

ΥΛΟΠΟΙΗΣΗ

Εισαγωγή

Στο στάδιο της ανάλυσης καταγράφηκαν οι βασικές λειτουργίες της εφαρμογής και τα βασικά στοιχεία τα οποία θα πρέπει να απασθηνκεύονται και να ανακαλούνται από τη βάση δεδομένων. Στο επόμενο στάδιο της σχεδίασης, σχεδιάστηκε η βάση δεδομένων και η δομή πλοήγησης στην εφαρμογή με βάση τις απαιτήσεις και τις λειτουργίες που θα πρέπει να υλοποιηθούν. Στη συνέχεια με βάση τις τεχνολογίες ανάπτυξης λογισμικού για web εφαρμογές που εξετάσαμε στο θεωρητικό μέρος, κατέληξε το στάδιο του σχεδιασμού στην επιλογή των δημοφιλέστερων και κατάλληλότερων εργαλείων για την υλοποίηση.

Έτσι η υλοποίηση βασίζεται στη χρήση ιστοσελίδων HTML μέσα στις οποίες και θα ενσωματωθεί κώδικας JavaScript για λειτουργίες που μπορούν να εκτελεστούν στον client και PHP κώδικας σε Symfony(framwork) για λειτουργίες που πρόκειται να εκτελούνται στον server. Η παρουσίαση του περιεχομένου των ιστοσελίδων θα γίνει με την τεχνική των Cascade Style Sheet (CSS). Ο τρόπος παρουσίασης βασίστηκε σε ένα παραμετροποιημένο πρότυπο παρουσίασης του περιεχόμενου στο οποίο έγιναν οι κατάλληλες μετατροπές ώστε να πληρούνται καλύτερα οι ανάγκες της εφαρμογής.

Βάσεις Δεδομένων Doctrine ORM (Symfony)

Το Symfony δεν παρέχει ένα στοιχείο που να συνεργάζεται με τη βάση δεδομένων, αλλά παρέχει μια βιβλιοθήκη που ονομάζεται Doctrine. Το Doctrine Project είναι το σπίτι πολλών βιβλιοθηκών PHP που επικεντρώνονται κυρίως στην αποθήκευση βάσεων δεδομένων και την χαρτογράφηση αντικειμένων.

Installing Doctrine (Εγκατάσταση)

Αρχικά καναμε εγκατάσταση του Doctrine με τις παρακάτω εντολές :

```
composer require doctrine maker
```

Το Doctrine θα δημιουργήσει τη βάση δεδομένων :

```
$ php bin/console doctrine:database:create
```

Creating an Entity Class (Δημιουργία κλάσης οντοτήτων)

```
$ php bin/console make:entity Product
```

Με την δημιουργία των Entities σύντομα, μπορούμε να αποθηκεύουμε και να ζητάμε αντικείμενα του Προϊόντος σε έναν πίνακα προϊόντων στη βάση δεδομένων.

Με την παραπάνω εντολή έχουμε δημιουργήσει τα δικά μας Entities τα οποία βρίσκονται

src/AppBundle/Entity.php φάκελο (είναι η ακριβής τοποθεσία των Entities ή αλλιώς το path) :

```
<?php
namespace AppBundle\Entity;

/**
 * @ORM\Entity(repositoryClass="AppBundle\Repository\ActivityRepository")
 * @ORM\Table(name="`activities`")
 */
use Doctrine\ORM\Mapping as ORM;

class Activity
{
    /**
     * @ORM\Column(type="integer")
     * @ORM\Id
     * @ORM\GeneratedValue(strategy="AUTO")
     */
    private $id;

    /**
     * @ORM\Column(type="integer")
     */
    private $city_id;

    /**
     * @ORM\Column(type="integer")
     */
    private $island_id;

    /**
     * @ORM\Column(type="integer")
     */
    private $country_id;

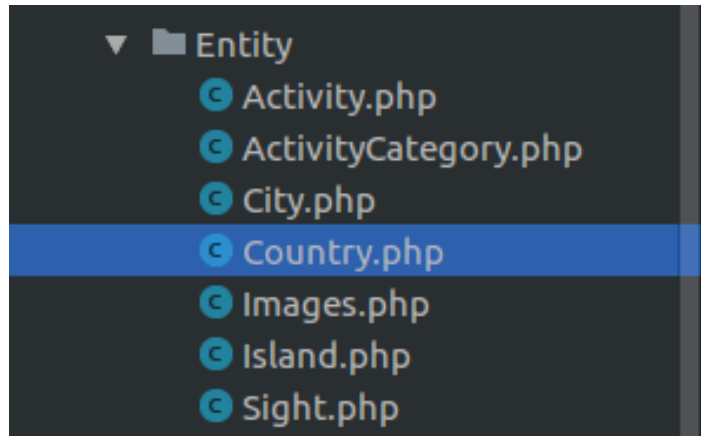
    /**
     * @ORM\Column(type="integer")
     */
    private $category_id;

    /**
     * @ORM\Column(type="string")
     */
    private $activity_name;

    /**
     * @ORM\Column(type="string")
     */

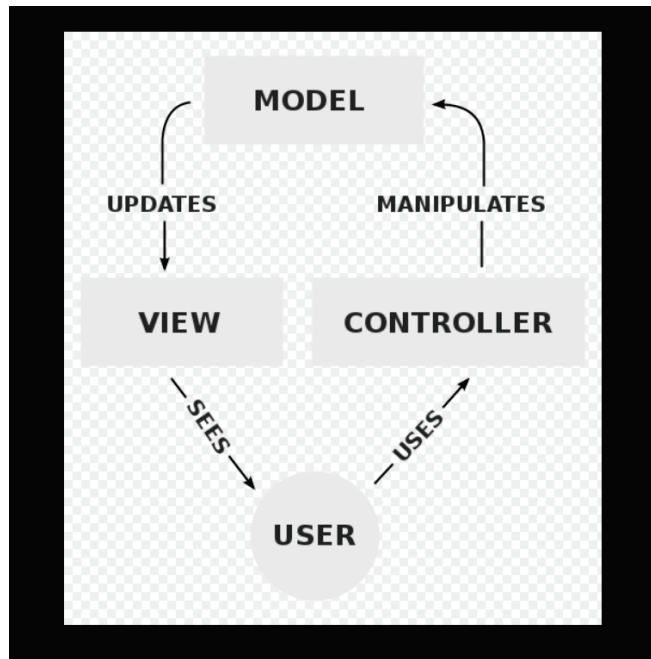
```

Επίσης να σημειωθεί πως ο κάθε πίνακας ξεχωριστά όπως αναφέραμε στις βάσεις έχει και τα δικά του Entities για να γίνει η ένωση όλων των δεδομένων από την βάση. Αυτό θα το δούμε και στο παρακάτω screenshot :



Model-view-controller

Το Model-view-controller (σε συντομογραφία αναφέρεται ως MVC) είναι ένα μοντέλο αρχιτεκτονικής λογισμικού το οποίο χρησιμοποιείται για την δημιουργία περιβαλλόντων αλληλεπίδρασης χρήστη. Στο μοντέλο αυτό η εφαρμογή διαιρείται σε τρία διασυνδεδεμένα μέρη ώστε να διαχωριστεί η παρουσίαση της πληροφορίας στον χρήστη από την μορφή που έχει αποθηκευτεί στο σύστημα. Το κύριο μέρος του μοντέλου είναι το αντικείμενο Model το οποίο διαχειρίζεται την ανάκτηση/αποθήκευση των δεδομένων στο σύστημα. Το αντικείμενο View χρησιμοποιείται μόνο για να παρουσιάζεται η πληροφορία στον χρήστη (π.χ. με γραφικό τρόπο). Το τρίτο μέρος είναι ο Controller ο οποίος δέχεται την είσοδο και στέλνει εντολές στο αντικείμενο Model και στο View.



Εκτός από το να διαιρείται η εφαρμογή σε τρία μοντέλα, η σχεδίαση model–view–controller ορίζει και τις αλληλεπιδράσεις των μοντέλων

- Ο controller μπορεί να στέλνει εντολές στο μοντέλο και να ενημερώνει την κατάσταση του μοντέλου. Μπορεί επίσης να στέλνει εντολές ώστε να γίνει η αντίστοιχη αναπαράσταση των δεδομένων του μοντέλου μέσω του View.
- Το model ενημερώνει τις αντίστοιχες αναπαραστάσεις views και τους controllers όταν υπάρχει αλλαγή στα δεδομένα. Αυτή η ενημέρωση επιτρέπει στα views να ενημερώνουν την γραφική απεικόνιση.
- Το view αναπαριστά με γραφικό τρόπο την πληροφορία που περιέχει το model δημιουργώντας γραφική παρουσίαση στο χρήστη.

Επεξήγηση κώδικα

Στην παραπάνω εικόνα βλέπουμε την συνάρτηση να καλεί την αρχική σελίδα .

```
class DefaultController extends Controller
{
    /**
     * @Route("/", name="homepage")
     */
    public function indexAction(Request $request)
    {
        // replace this example code with whatever you need
        return $this->render( view: 'default/homepage.html.twig' );
    }
}
```

Στην επόμενη συνάρτηση ανάλογα την αναζήτηση του χρήστη εμφανίζεται η ανάλογη σελίδα της επιλογής του.

```
/**
 * @Route("/search-homepage", name="HomepageSearch")
 */
public function searchHomepage(Request $request)
{
    $output = array();
    $repoCountry = $this->getDoctrine()->getRepository( persistentObject: Country::class);
    $repoCity = $this->getDoctrine()->getRepository( persistentObject: City::class);
    $repoIsland = $this->getDoctrine()->getRepository( persistentObject: Island::class);
    $repoSight = $this->getDoctrine()->getRepository( persistentObject: Sight::class);
    $repoActivity = $this->getDoctrine()->getRepository( persistentObject: Activity::class);
    $search = $request->query->get( key: "search");
    $country = $repoCountry->findOneBy(array('country_name' => $search));
    if ($country) {
        $id = $country->getId();
        $output['redirect'] = '/country?id='.$id;
    }
    $city = $repoCity->findOneBy(array("city_name" => $search));
    if ($city) {
        $id = $city->getId();
        $output['redirect'] = '/city?id='.$id;
    }
    $island = $repoIsland->findOneBy(array("island_name" => $search));
    if ($island) {
        $id = $island->getId();
        $output['redirect'] = '/island?id='.$id;
    }
    $sight = $repoSight->findOneBy(array("sight_name" => $search));
    if ($sight) {
        $id = $sight->getId();
        $output['redirect'] = '/sight?id='.$id;
    }
    $activity = $repoActivity->findOneBy(array("activity_name" => $search));
    if ($activity) {
        $id = $activity->getId();
        $output['redirect'] = '/activity?id='.$id;
    }
    return new JsonResponse($output);
}
```

Η πιο κατω συναρτηση ειναι υπευθινη να παρουσιασει στον χρηστη ολες τις πιθανες επιλογες κατα την διαρκεια της πληκτολογησης του .

```
/**
 * @Route("/autocomplete-homepage",name="autocomplete-homepage")
 */
public function autocompleteHomepage(Request $request)
{
    $search = $request->query->get( key: 'search' );
    $query = "select country_name
from countries
where country_name like '$search%'
union
select city_name
from cities
where city_name like '$search%'
union
select island_name
from islands
where island_name like '$search%'
union
select sight_name
from sights
where sight_name like '$search%'
union
select activity_name
from activities
where activity_name like '$search%'";
    $em = $this->getDoctrine()->getManager();
    $statement = $em->getConnection()->prepare($query);
    $statement->execute();

    $results = $statement->fetchAll();
    foreach ($results as $key=>$result)
    {
        $output[$key] = $result['country_name'];
    }
    return new JsonResponse($output);
}
```

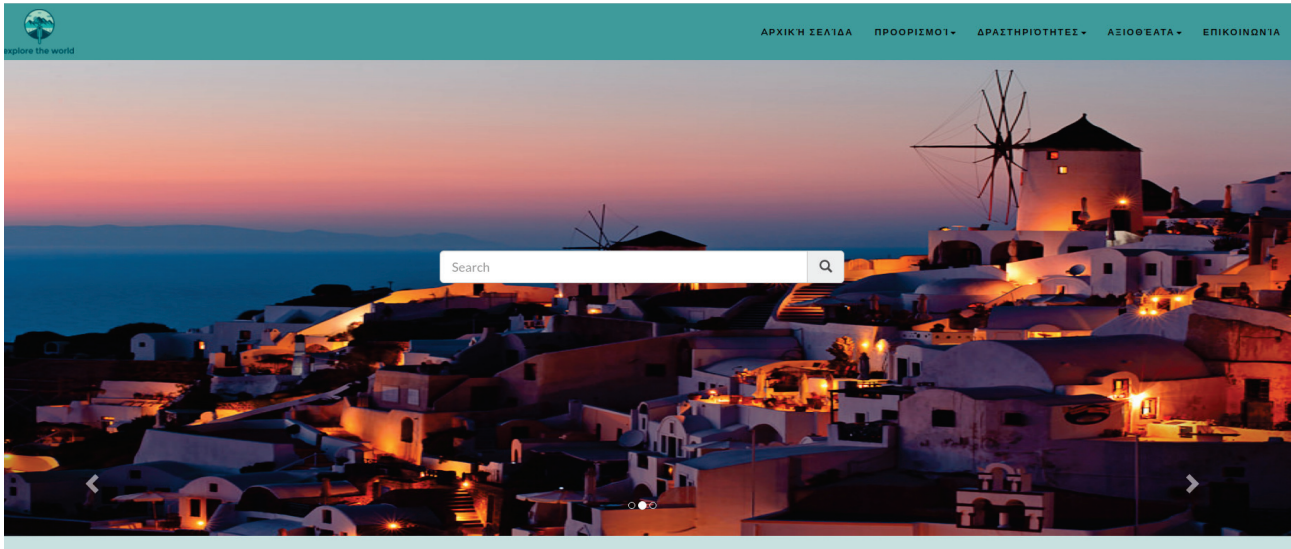
Η πιο κάτω συνάρτηση εμφανίζει όλους τους προορισμούς σε ένα drop down στο menu .

```
1  /**
2  * @Route("/getAllDestination",name="destination")
3  */
4  public function allDestination()
5  {
6      $output = array();
7
8      $repoCountry = $this->getDoctrine()->getRepository( persistentObject: Country::class);
9      $repoCity = $this->getDoctrine()->getRepository( persistentObject: City::class);
10     $repoIsland = $this->getDoctrine()->getRepository( persistentObject: Island::class);
11
12     $countries = $repoCountry->findAll();
13     $cities = $repoCity->findAll();
14     $islands = $repoIsland->findAll();
15
16     foreach ($countries as $country)
17     {
18         $output['countries'][]=array('country_name'=>$country->getCountryName(), 'id'=>$country->getId());
19     }
20     foreach ($cities as $city)
21     {
22         $output['cities'][]=array('city_name'=>$city->getCityName(), 'id'=>$city->getId());
23     }
24     foreach ($islands as $island)
25     {
26         $output['islands'][]=array('island_name'=>$island->getIslandName(), 'id'=>$island->getId());
27     }
28
29     return new JsonResponse($output);
30 }
31
```

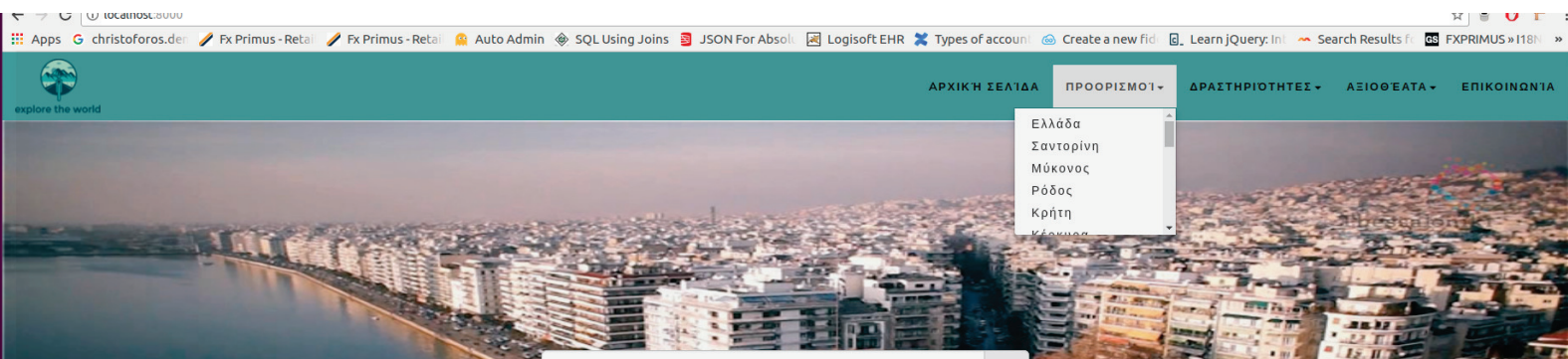
Όπως περιγράψαμε και στην παραπάνω εικόνα έτσι και εδώ εμφανίζονται όλες οι δραστηριότητες .

```
1  /**
2  * @Route("/getAllAdventure",name="getAllAdventure")
3  */
4  public function getAllAdventure()
5  {
6      $output = array();
7
8      $repoActivity = $this->getDoctrine()->getRepository( persistentObject: Activity::class);
9
10     $activities = $repoActivity->findAll();
11
12     foreach ($activities as $activity)
13     {
14         $output['adventure'][]=array('name'=>$activity->getActivityName(), 'id'=>$activity->getId());
15     }
16
17     return new JsonResponse($output);
18 }
19
```

Στο παρακάτω στιγμιότυπο βλέπουμε την αρχική σελίδα της ιστοσελίδας μας.



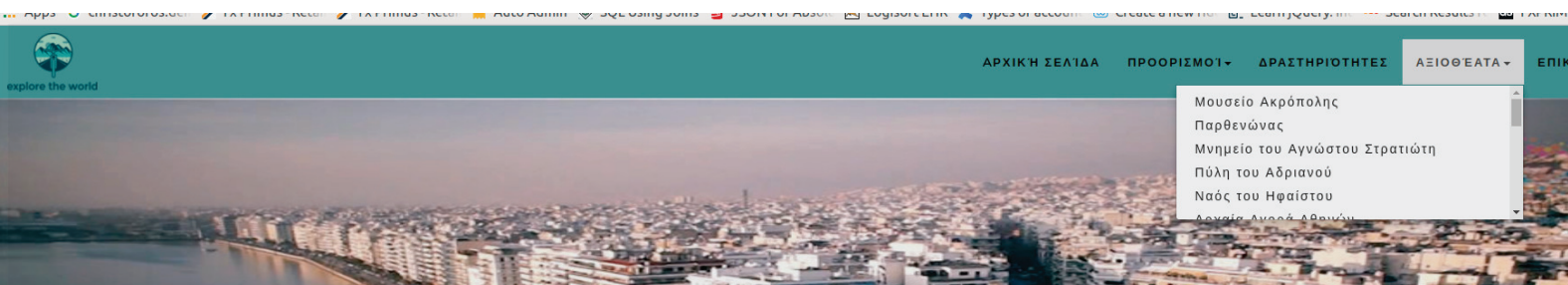
Η μπάρα επιλογών δίνει την δυνατότητα στον χρήστη να να διαλλάξει τον προορισμό της επιλογής σε ολόκληρη την Ελλάδα.



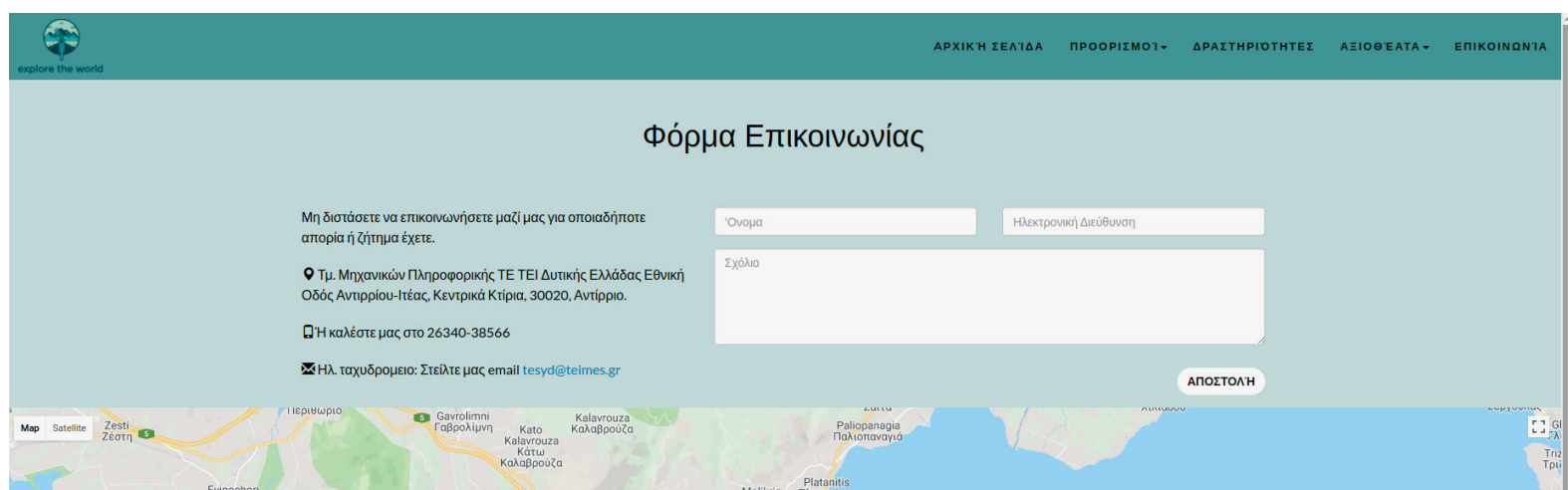
Η μπάρα επιλογών επίσης υλοποιήθηκε με τον παρακάτω κώδικα

```
1 <nav class="navbar navbar-default">
2   <div class="navbar-header">
3     <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#myPage" aria-expanded="true">
4       <span class="icon-bar"></span>
5       <span class="icon-bar"></span>
6       <span class="icon-bar"></span>
7     </button>
8     <a class="navbar-brand" href="#myPage"></a>
9   </div>
10  <div class="navbar-collapse collapse in" id="myPage" aria-expanded="true" style="background-color: #f8f9fa; padding: 5px 0;">
11    <ul class="nav navbar-nav navbar-right">
12      <li><a href="#">Αρχική Σελίδα </a></li>
13      <li class="dropdown">
14        <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">Προορισμοί<span class="fa fa-angle-down"></span></a>
15        <ul id="destinations" class="dropdown-menu">
16          <li><a href="#">Παρί</a></li>
17        </ul>
18      </li>
19      <li class="dropdown">
20        <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">Δραστηριότητες<span class="fa fa-angle-down"></span></a>
21        <ul id="adventure" class="dropdown-menu">
22          <li><a href="#">Καταδύσεις</a></li>
23        </ul>
24      </li>
25      <li class="dropdown">
26        <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">Αξιοθέατα<span class="fa fa-angle-down"></span></a>
27        <ul id="sight" class="dropdown-menu">
28          <li><a href="#">Μουσείο Ακρόπολης</a></li>
29          <li><a href="#">Παρθενώνας</a></li>
30          <li><a href="#">Μνημείο του Αγνώστου Στρατιώτη</a></li>
31          <li><a href="#">Πύλη του Αδριανού</a></li>
32          <li><a href="#">Ναός του Ηφαίστου</a></li>
33          <li><a href="#">Αρχαία Αγορά Αθηνών</a></li>
34        </ul>
35      </li>
36      <li><a href="/contact-us">Επικοινωνία</a></li>
37      <li><a href="/top-ten">Προτεινόμενα</a></li>
38    </ul>
39  </div>
40 </nav>
```

Η επόμενη επιλογή είναι τα αξιοθέατα . Μέσα από αυτήν την επιλογή ο χρήστης ανάλογα την πόλη που έχει επιλέξει σε παραπέμπει και στο ανάλογο αξιοθέατο .



Επίσης μια απο τις επιλογές του χρήστη είναι και η επικοινωνία με τον αρμόδιο. Μέσα απο την φόρμα θα μπορεί να στείλει email . Μέσα στην φόρμα αναγράφεται η διευθυνσή αλλα και τηλέφωνο επικοινωνίας . Με τον παρακάτω χάρτη βλέπουμε και την ακριβής τοποθεσία .



explore the world

ΑΡΧΙΚΗ ΣΕΛΙΔΑ ΠΡΟΟΡΙΣΜΟΙ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ΑΞΙΟΘΕΑΤΑ ΕΠΙΚΟΙΝΩΝΙΑ

Φόρμα Επικοινωνίας

Μη διστάσετε να επικοινωνήσετε μαζί μας για οποιαδήποτε απορία ή ζήτηση έχετε.

📍 Τμ. Μηχανικών Πληροφορικής ΤΕ ΤΕΙ Δυτικής Ελλάδας Εθνική Οδός Αντιρρίου-Ιτέας, Κεντρικά Κτίρια, 30020, Αντίρριο.

📞 Η καλέστε μας στο 26340-38566


✉ Ηλ. ταχυδρομείο: Στείλτε μας email tesyd@teimes.gr

Όνομα

Ηλεκτρονική Διεύθυνση

Σχόλιο

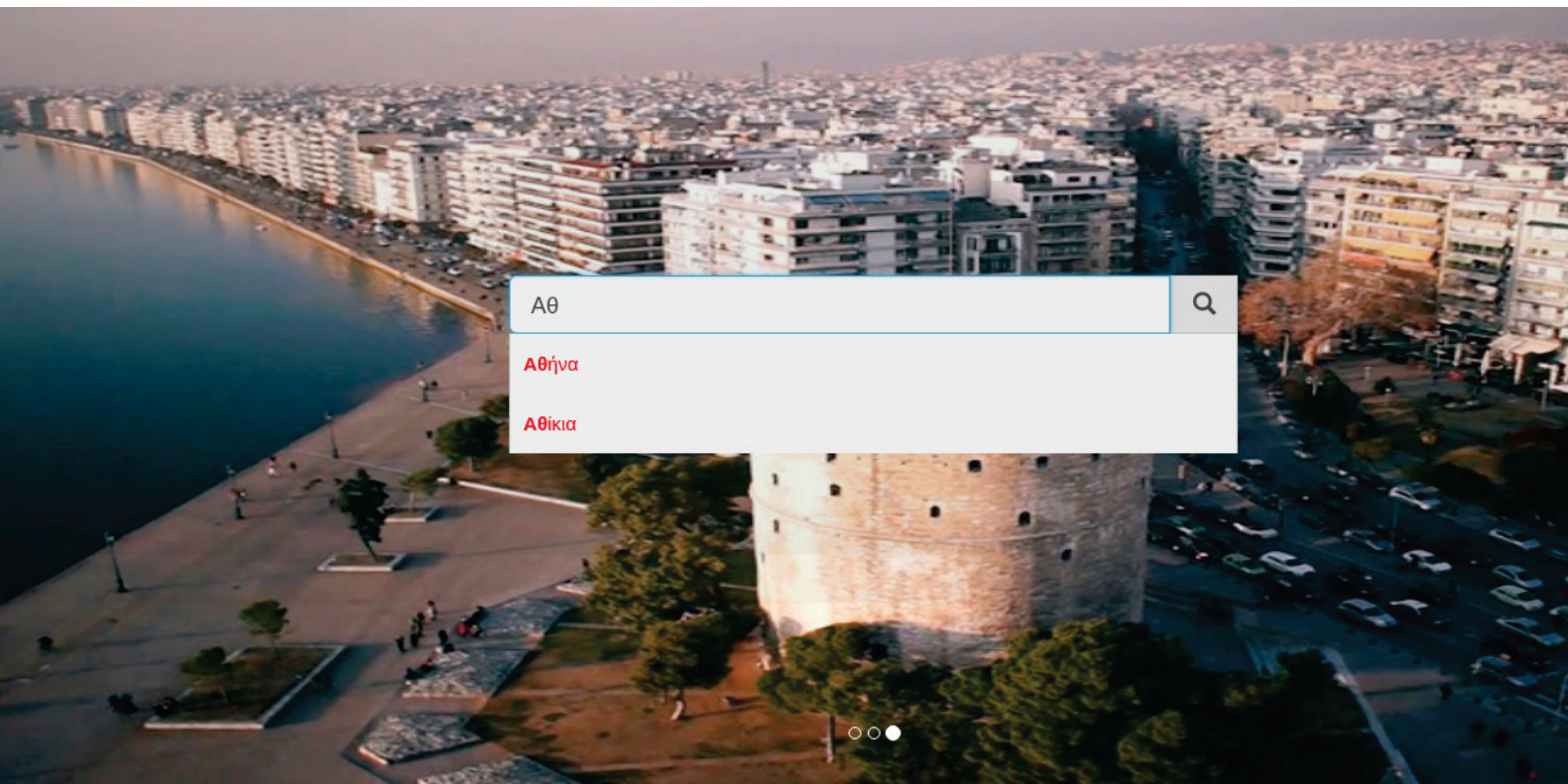
ΑΠΟΣΤΟΛΗ



Η παραπάνω φόρμα επικοινωνίας υλοποιήθηκε με τον παρακάτω κώδικα

```
1 {% extends 'base.html.twig' %}
2 {% block title %}Explore the World{% endblock %}
3 {% block body %}
4
5 <div id="myPage" data-spy="scroll" data-target=".navbar" data-offset="60">
6   {% include 'default/navbar.html.twig' %}
7
8   <!-- Include the above in your HEAD tag -->|
9 </div>
10 <div id="contact" class="container">
11   <br>
12   <h1 class="text-center" style="...">Φόρμα Επικοινωνίας</h1>
13   <br>
14   <br>
15   <div class="row">
16     <div class="col-sm-5" style="...">
17       <p>Μη διστάσετε να επικοινωνήσετε μαζί μας για οποιαδήποτε απορία ή ζήτημα έχετε.</p>
18       <p><span class="glyphicon glyphicon-map-marker"></span> Τμ. Μηχανικών Πληροφορικής ΤΕ
19         ΤΕΙ Δυτικής Ελλάδας
20         Εθνική Οδός Αντιρρίου-Ιτέας,
21         Κεντρικά Κτίρια,
22         30020, Αντίρριο. </p>
23       <p><span class="glyphicon glyphicon-phone"></span>Η καλέστε μας στο
24         26340-38566 </p>
25       <p><span class="glyphicon glyphicon-envelope"></span>
26         Ηλ. ταχυδρομείο: Στείλτε μας email <a href="mailto:tesyd@teimes.gr" target="_blank">tesyd@teimes.gr</a></p>
27     </div>
28     <div class="col-sm-7">
29       <div class="row">
30         <div class="col-sm-6 form-group">
31           <input class="form-control" id="sub" name="name" placeholder="Όνομα" type="text" required>
32         </div>
33         <div class="col-sm-6 form-group">
34           <input class="form-control" id="email" name="email" placeholder="Ηλεκτρονική Διεύθυνση" type="email" required>
35         </div>
36       </div>
37       <textarea class="form-control" id="comments" name="comments" placeholder="Σχόλιο" rows="5"></textarea><br>
38       <div class="row">
39         <div class="col-sm-12 form-group">
40           <button id="send" class="btn btn-default pull-right" type="submit">Αποστολή</button>
41         </div>
42       </div>
43     </div>
44   </div>
45 </div>
46
47 <!-- Add Google Maps -->
48 <div id="googleMap" style="..."></div>
```

Τέλος υπάρχει μια μπάρα αναζήτησης ποθ μέσα απο αυτην μπορεί ο χρηστης να πληκτολογήσει την πόλη ή το νησί της επιλογής του και δυναμικά να τον μεταφέρει στην ανάλογη σελίδα.



Επιπλέον η μπάρα αναζήτησης εκτελέστηκε με το παρακάτω στιγμιότυπο


```

</div>
</div>
<form class="col-sm-12" id="searchForm">
  <div class="form-group col-sm-4 col-sm-offset-4">
    <div class="input-group input-group-lg">
      <input id="search" type="text" class="autocomplete form-control" placeholder="Search">
      <span class="input-group-addon"><i class="glyphicon glyphicon-search"></i></span>
    </div>
  </div>
</form>

<!-- Controls -->

```

Επίσης για την εμφάνιση των πόλεων αλλά και νησιών στην μπάρα επιλογών το οποίο έγινε με autocomplete

```

1 function autocomplete(inp) {
2   /*the autocomplete function takes two arguments,
3   the text field element and an array of possible autocompleted values:*/
4   var currentFocus;
5   /*execute a function when someone writes in the text field:*/
6   inp.addEventListener("input", function(e) {
7     var a, b, i, val = this.value;
8     /*close any already open lists of autocompleted values*/
9     closeAllLists();
10    if (!val) { return false;}
11    currentFocus = -1;
12    /*create a DIV element that will contain the items (values):*/
13    a = document.createElement("DIV");
14    a.setAttribute("id", this.id + "autocomplete-list");
15    a.setAttribute("class", "autocomplete-items");
16    /*append the DIV element as a child of the autocomplete container:*/
17    this.parentNode.appendChild(a);
18    /*for each item in the array...*/
19    $.ajax({
20      url: "/autocomplete-homepage",
21      data: {search: val},
22      // dataType: 'json',
23      success: function (response) {
24        arr = response;
25      },
26      error: function (response) {
27        toastr.error(response.statusText + ' ' + response.status, {toastOptions});
28      }
29    });
30    for (i = 0; i < arr.length; i++) {
31      /*check if the item starts with the same letters as the text field value:*/
32      if (arr[i].substr(0, val.length).toUpperCase() == val.toUpperCase()) {
33        /*create a DIV element for each matching element:*/
34        b = document.createElement("DIV");
35        /*make the matching letters bold:*/
36        b.innerHTML = "<strong>" + arr[i].substr(0, val.length) + "</strong>";
37        b.innerHTML += arr[i].substr(val.length);
38        /*insert a input field that will hold the current array item's value:*/
39        b.innerHTML += "<input id='finallyValue' type='hidden' value='" + arr[i] + "'>";
40        /*execute a function when someone clicks on the item value (DIV element):*/
41        b.addEventListener("click", function(e) {
42          /*insert the value for the autocomplete text field:*/
43          inp.value = this.getElementsByTagName("input")[0].value;
44          /*close the list of autocompleted values,
45          (or any other open lists of autocompleted values:*/
46          closeAllLists();
47        });
48        a.appendChild(b);
49      }
50    }
51  });
52 }

```

```

    });
    for (i = 0; i < arr.length; i++) {
        /*check if the item starts with the same letters as the text field value:*/
        if (arr[i].substr(0, val.length).toUpperCase() == val.toUpperCase()) {
            /*create a DIV element for each matching element:*/
            b = document.createElement("DIV");
            /*make the matching letters bold:*/
            b.innerHTML = "<strong>" + arr[i].substr(0, val.length) + "</strong>";
            b.innerHTML += arr[i].substr(val.length);
            /*insert a input field that will hold the current array item's value:*/
            b.innerHTML += "<input id='finallyValue' type='hidden' value='" + arr[i] + "'>";
            /*execute a function when someone clicks on the item value (DIV element):*/
            b.addEventListener("click", function(e) {
                /*insert the value for the autocomplete text field:*/
                inp.value = this.getElementsByTagName("input")[0].value;
                /*close the list of autocompleted values,
                (or any other open lists of autocompleted values:*/
                closeAllLists();
            });
            a.appendChild(b);
        }
    }
});
/*execute a function presses a key on the keyboard:*/
inp.addEventListener("keydown", function(e) {
    var x = document.getElementById(this.id + "autocomplete-list");
    if (x) x = x.getElementsByTagName("div");
    if (e.keyCode == 40) {
        /*If the arrow DOWN key is pressed,
        increase the currentFocus variable:*/
        currentFocus++;
        /*and make the current item more visible:*/
        addActive(x);
    } else if (e.keyCode == 38) { //up
        /*If the arrow UP key is pressed,
        decrease the currentFocus variable:*/
        currentFocus--;
        /*and and make the current item more visible:*/
        addActive(x);
    } else if (e.keyCode == 13) {
        /*If the ENTER key is pressed, prevent the form from being submitted,*/
        e.preventDefault();
        if (currentFocus > -1) {
            /*and simulate a click on the "active" item:*/
            if (x) x[currentFocus].click();
        }
        // console.log(x[currentFocus].textContent)
        $ ajax({

```

Μελλοντική εργασία

Η εφαρμογή με βάση τον τρόπο σχεδίασης έχει πολλές δυνατότητες βελτίωσης. Η διαχείριση των παραμέτρων δημιουργήθηκε με δυναμικές σελίδες σε PHP και ως εκ τούτου έχει μεγάλες δυνατότητες επέκτασης των λειτουργιών που μπορούν να προστεθούν μελλοντικά. Επίσης ο τρόπος παρουσίασης των αποτελεσμάτων αναζήτησης γίνεται με πολύ φιλικό τρόπο και η σχεδίαση του γραφικού περιβάλλοντος δίνει τη δυνατότητα παρουσίασης της εφαρμογής σε μικρότερες συσκευές (tablet – έξυπνα κινητά).

Συμπερασματικά θα μπορούσαμε να πούμε ότι η συγκεκριμένη εργασία αποτελεί μια καλή αφετηρία για ένα πλήρες σύστημα το οποίο μπορεί να χρησιμοποιηθεί άμεσα στην προβολή προορισμών πολλών περιοχών αλλά και πρισσότερων κρατών και στην αναζήτησή τους μέσω ενός web browser με φιλικό τρόπο.

Βιβλιογραφία

1. Symfony documentation : “<https://symfony.com/doc/current/index.html>”.
2. About HTML and CSS : HTML and CSS: Visual QuickStart Guide, 8th Edition.
3. About PHP , JavaScript JQuery, MySQL and : Learning PHP, MySQL & JavaScript: With JQuery, CSS & HTML5.
4. About Ajax : Ajax: The Definitive Guide .
5. W3schools : “<https://www.w3schools.com/>”.
6. <http://php.net/>
7. <https://dev.mysql.com/doc/>