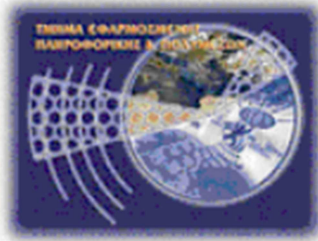




**Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης**

**Σχολή Τεχνολογικών Εφαρμογών  
Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων**



## **Πτυχιακή εργασία**

### **Σύστημα παροχής υπηρεσιών δρομολόγησης και καθοδήγησης**

**Ιωάννης Αλμπάνης (Α.Μ. 2269)**

**Σπύρος Νταλιάνης (Α.Μ. 2193)**

**Επιβλέπων Καθηγητής: Νίκος Παπαδάκης**



## **Abstract**

The development of electronic maps is one of the key developments in computer technology, as it is a function that has direct applications to the real world and is used by a very large number of users, with several of them having no other experience in the use of internet or computers. The possibilities are numerous and have sizable utility in everyday life.

In this thesis we will cover the process of creating a system that allows the planning of everyday tasks of users using such maps and advanced navigation features they offer. Our aim is to present the relevant technologies, their potential and how to use them.

## Σύνοψη

Η ανάπτυξη των ηλεκτρονικών χαρτών αποτελεί μία από τις κυριότερες εξελίξεις της τεχνολογίας των υπολογιστών, καθώς πρόκειται για μία λειτουργία που έχει άμεση εφαρμογή στον πραγματικό κόσμο και χρησιμοποιείται από έναν πολύ μεγάλο αριθμό χρηστών, με αρκετούς από αυτούς να μην έχουν καμία άλλη εμπειρία στην χρήση του διαδικτύου. Οι δυνατότητες που προσφέρονται είναι πάρα πολλές και έχουν αρκετά μεγάλη χρησιμότητα στην καθημερινότητα.

Σε αυτή την εργασία θα καλύψουμε την διαδικασία δημιουργίας ενός συστήματος που επιτρέπει τον προγραμματισμό των καθημερινών εργασιών των χρηστών με την χρήση τέτοιων χαρτών και των προηγμένων λειτουργιών πλοήγησης που προσφέρουν. Σκοπός μας είναι να παρουσιάσουμε τις σχετικές τεχνολογίες, τις δυνατότητές τους και τον τρόπο χρήση τους.

## Περιεχόμενα

Abstract .....	3
Σύνοψη .....	4
Περιεχόμενα.....	5
1 Εισαγωγή.....	8
1.1 Περίληψη.....	8
1.2 Κίνητρο για την διεξαγωγή της εργασίας.....	8
1.3 Σκοπός και στόχοι εργασίας.....	8
1.4 Δομή εργασίας .....	8
2 Ανάγκες και Τεχνολογίες.....	9
2.1 Έξυπνοι προσωπικοί βοηθοί .....	9
2.2 Συστήματα δρομολόγησης.....	10
2.2.1 Google Maps .....	10
2.2.2 Γενικά στοιχεία.....	10
2.2.3 Προβολές.....	11
2.2.4 Εύρεση τοποθεσίας.....	14
2.2.5 Οδηγίες διαδρομής .....	14
2.3 Τεχνολογίες web.....	15
2.3.1 Η γλώσσα περιγραφής HTML.....	16
2.3.2 Η γλώσσα μορφοποίησης CSS.....	17
2.3.3 Η γλώσσα προγραμματισμού JavaScript .....	20
2.3.4 Το πρότυπο HTML5 .....	24
2.3.5 Η γλώσσα προγραμματισμού Php .....	27
2.3.6 XML.....	33
2.3.7 Βάσεις Δεδομένων XML .....	35
3 Εργαλεία ανάπτυξης.....	39
3.1 Γενικά.....	39
3.2 WAMPServer .....	39
3.2.1 Σχετικά με το WAMPServer.....	39
3.2.2 Εγκατάσταση .....	40
3.3 BaseX .....	40
3.3.1 Σχετικά με την BaseX.....	40
3.3.2 Εγκατάσταση της BaseX .....	40
3.3.3 Λειτουργίες και εκτέλεση.....	41
3.4 Εργαλεία συγγραφής κώδικα.....	42
3.4.1 Notepad++.....	42

4	Ανάλυση εφαρμογής.....	44
4.1	Γενικά.....	44
4.2	Τμήματα εφαρμογής.....	44
4.2.1	Αρχεία XML.....	44
4.2.2	Αρχεία PHP .....	44
4.2.3	Κώδικας JavaScript .....	44
4.2.4	Κώδικας CSS .....	44
4.2.5	Κώδικας HTML.....	44
4.3	Δεδομένα XML.....	45
4.3.1	Αρχείο χρηστών.....	45
4.3.2	Αρχείο βενζινάδικων .....	45
4.3.3	Αρχείο super market .....	46
4.3.4	Αρχείο ταινιών.....	46
4.4	Παρουσίαση κώδικα .....	47
4.4.1	Αρχείο μορφοποίησης style.css .....	47
4.4.2	Βιβλιοθήκη JavaScript map.js.....	48
5	Οδηγίες χρήσης .....	59
5.1	Είσοδος στο σύστημα.....	59
5.2	Αναζήτηση βενζινάδικου .....	59
5.3	Εύρεση super market .....	61
5.4	Εύρεση σινεμά.....	62
6	Συμπεράσματα .....	65
6.1	Μελλοντική επέκταση.....	65
7	Βιβλιογραφία.....	66
8	Παράρτημα – Κώδικας εφαρμογής.....	68
8.1	Αρχεία CSS .....	68
8.1.1	style.css .....	68
8.2	Αρχεία PHP .....	73
8.2.1	_page_end.php.....	73
8.2.2	_page_start.php .....	73
8.2.3	cinemas.php .....	75
8.2.4	do_login.php .....	77
8.2.5	do_logout.php.....	77
8.2.6	gasstations.php .....	78
8.2.7	index.php.....	79
8.2.8	map.php .....	80
8.2.9	supermarkets.php .....	80
8.3	Αρχεία javascript .....	82

8.3.1	maps.js.....	82
-------	--------------	----

# 1 Εισαγωγή

## 1.1 Περίληψη

## 1.2 Κίνητρο για την διεξαγωγή της εργασίας

Η χρήση ηλεκτρονικών χαρτών έχει γνωρίσει πολύ μεγάλη εξάπλωση τα τελευταία χρόνια. Αν και αρχικά το μοναδικό πλεονέκτημα των ηλεκτρονικών χαρτών ήταν η ευχρηστία τους σε σχέση με τους τυπωμένους χάρτες, πλέον προσφέρονται αρκετές επιπλέον υπηρεσίες όπως η εύρεση της καλύτερης διαδρομής ή η προβολή συγκεκριμένων καταστημάτων ή αξιοθέατων που βρίσκονται γύρω από τον χρήστη.

Αυτή η εξάπλωση μας ώθησε στην δημιουργία μίας εφαρμογής που θα χρησιμοποιεί το Google Maps, μία από τις πιο δημοφιλείς υπηρεσίες ηλεκτρονικών χαρτών, σαν βασικό κομμάτι για τη δημιουργία μίας δυναμικής ιστοσελίδας.

## 1.3 Σκοπός και στόχοι εργασίας

Στόχος αυτής της εργασίας είναι να εξερευνήσουμε και να χρησιμοποιήσουμε τις δυνατότητες της υπηρεσίας Google Maps σε μία διαδικτυακή εφαρμογή που θα δίνει στον χρήστη οδηγίες για την καλύτερη διαδρομή σε μία πλειάδα διαφορετικών προορισμών.

Μέσα από τη συγγραφή της εφαρμογής σκοπεύουμε να αποκτήσουμε εξοικείωση τόσο με τον τρόπο λειτουργίας των Google Maps από τη σκοπιά του προγραμματιστή που το χρησιμοποιεί για την ανάπτυξη εφαρμογών, όσο και από την πλευρά του χρήστη που κάνει χρήση της εφαρμογής.

## 1.4 Δομή εργασίας

Πέρα από το πρώτο κεφάλαιο της εργασίας, που είναι το παρών, τα υπόλοιπα κεφάλαια καλύπτουν τα εξής θέματα:

- Στο δεύτερο κεφάλαιο εξετάζουμε σε θεωρητικό επίπεδο τις τεχνολογίες που χρησιμοποιήσαμε.
- Στο τρίτο κεφάλαιο έχουμε την παρουσίαση των εργαλείων που χρειάζονται για την ανάπτυξη του προγράμματος.
- Το τέταρτο κεφάλαιο είναι η παρουσίαση κάποιων βασικών στοιχείων της εφαρμογής.
- Το πέμπτο κεφάλαιο περιέχει τις οδηγίες χρήσης.
- Στο έκτο κεφάλαιο παρουσιάζουμε τα συμπεράσματά μας και τις προτάσεις για μελλοντική επέκταση.
- Τέλος στο παράρτημα υπάρχουν τα σημαντικά κομμάτια κώδικα της εφαρμογής.



## 2 Ανάγκες και Τεχνολογίες

Η αναζήτηση είναι η διέξοδος που έχουν οι περισσότεροι χρήστες προς τον παγκόσμιο ιστό. Για τους περισσότερους, έχει γίνει δεύτερη φύση η απόσταση μιας εντολής σε λέξεις κλειδιά τα οποία θα οδηγήσουν στα επιθυμητά εργαλεία ή πληροφορίες. Η συνεχώς αυξανόμενες απαιτήσεις στα προϊόντα λογισμικού έχουν δημιουργήσει την τάση της εξέλιξης των μηχανών αναζήτησης σε "do engines", μηχανές που όχι μόνο θα συλλέγουν πληροφορίες αλλά θα μπορούν να ολοκληρώνουν αυτόματα εντολές και εργασίες.

Σε αυτό το κεφάλαιο θα γίνει μια συνοπτική παρουσίαση των τεχνολογιών και εργαλείων που χρησιμοποιήθηκαν για την ολοκλήρωση του συστήματος που αναπτύχθηκε.

### 2.1 Έξυπνοι προσωπικοί βοηθοί

Οι έξυπνοι προσωπικοί βοηθοί (Intelligent Personal Assistant (IPA)) είναι εφαρμογές λογισμικού που ενσωματώνουν τα δεδομένα που εισάγει ένας χρήστης με δυνατότητες εντοπισμού θέσης (localization) και πρόσβαση σε πληροφορίες από διάφορες βάσεις δεδομένων με σκοπό να παρέχουν προσωποποιημένες υπηρεσίες.

Οι IPAs ενεργούν εκ μέρους του χρήστη προκειμένου να ολοκληρώσουν διάφορες εργασίες ή να παρέχουν πληροφορίες βάση των εντολών του χρήστη. Μπορούν επιπλέον να διαχειρίζονται αυτόματα εργασίες και να ενημερώνουν-αναβαθμίζουν πληροφορίες χωρίς την πρωτοβουλία του χρήστη

Οι «έξυπνοι προσωπικοί βοηθοί» φαίνεται να εκθρονίζουν τους «ψηφιακούς προσωπικούς βοηθούς» (personal digital assistants (PDA)) και στις περισσότερες περιπτώσεις έχουν ξεπεράσει την υπολογιστική ισχύ των PDA. Επιπλέον οι IPA έχουν εισαχθεί ομοιόμορφα στην αγορά με την ενσωμάτωσή τους στα smartphones χωρίς επιπλέον κόστος και για το μέλλον προβλέπεται η περαιτέρω εξάπλωσή τους.



Όταν διάφοροι χρήστες αλληλοεπιδρούν με μια IPA εφαρμογή πέρα από την ανάκτηση πληροφοριών, έχουν το πλεονέκτημα της αυτόματης ολοκλήρωσης εργασιών. Είτε πληκτρολογώντας είτε με φωνητική εντολή οι εφαρμογές IPA μπορούν να εκτελέσουν διάφορες εντολές, όπως να παρακολουθούν την εξέλιξη ραντεβού και προγράμματος να στείλουν e-mails, να πραγματοποιήσουν μια αναζήτηση στο παγκόσμιο ιστό κ.ο.κ. Χρησιμοποιώντας τεχνολογίες εντοπισμού θέσης, μπορούν να αναγνωρίσουν το περιβάλλον στο οποίο βρίσκεται ο χρήστης και να καλέσουν πχ ένα ταξί, να κάνουν κράτηση σε ένα εστιατόριο ή να παρέχουν πληροφορίες για τον καιρό ή την κίνηση. Όσο ο χρήστης χρησιμοποιεί αυτή τη τεχνολογία, η εφαρμογή αξιολογεί τα δεδομένα και οι επαναληπτικές εργασίες θα απαιτούν λιγότερο χρόνο.

Τα πλεονεκτήματα των εφαρμογών λογισμικού IPA είναι η εύκολη πρόσβαση στο λογισμικό, η αυτοματοποίηση επαναληπτικών εργασιών και η παροχή στο χρήστη τη δυνατότητα να γίνει περισσότερο αποτελεσματικός και επαρκής. Τα μειονεκτήματα έχουν να κάνουν με το ειδικό λεξιλόγιο που χρησιμοποιείται για την πραγματοποίηση των εντολών καθώς και προβλήματα που υπάρχουν στην φωνητική αναγνώριση και στο γεγονός ότι οι πηγές και οι βάσεις δεδομένων είναι περιορισμένες, μειώνοντας την ποιότητα των παρεχόμενων υπηρεσιών.

## 2.2 Συστήματα δρομολόγησης

Τα συστήματα δρομολόγησης αναφέρονται στο λογισμικό σχεδιασμού διαδρομής τα οποία είναι σχεδιασμένα για να υπολογίζουν την βέλτιστη διαδρομή μεταξύ δύο γεωγραφικών σημείων. Χρησιμοποιούν μια μηχανή σχεδιασμού ταξιδιού ειδικευμένη για οδικά δίκτυα που λειτουργεί σαν σχεδιαστής διαδρομής. Συνήθως παρέχει μία λίστα με μέρη που βρίσκονται στη διαδρομή που χαράσσεται, διασταυρώσεις και οδηγίες που πρέπει να ακολουθηθούν, αριθμούς δρόμων, αποστάσεις κλπ. Στις περισσότερες περιπτώσεις επίσης παρέχει έναν διαδραστικό χάρτη με μια προτεινόμενη διαδρομή πάνω σε αυτόν.

### 2.2.1 Google Maps

Για την υλοποίηση των χαρτών στην εφαρμογή μας χρησιμοποιήσαμε την υπηρεσία Google Maps που προσφέρεται από την εταιρεία Google δωρεάν για μικρή χρήση. Αν και τα Google Maps δεν είναι μία τεχνολογία από μόνη της, εξαιτίας την μεγάλης σημασίας που έχουν για την ανάπτυξη της σελίδας παρουσιάζονται ξεχωριστά.

Οι χάρτες που δημιουργούνται από το Google Map είναι αρκετά πολύπλοκα στοιχεία HTML που περιέχουν οδηγίες CSS για την εμφάνιση του χάρτη και JavaScript για την κίνηση και την εμφάνιση των δεδομένων. Πρόκειται για μία υπηρεσία που ήταν ριζοσπαστική όταν έγινε διαθέσιμη και ακόμη και σήμερα προσφέρει πάρα πολλές δυνατότητες και μεγάλη ακρίβεια.

### 2.2.2 Γενικά στοιχεία

Η υπηρεσία Google Maps, που αρχικά είχε παρουσιαστεί με το όνομα Google Local είναι μια web εφαρμογή που παρέχει πρόσβαση σε μία ομάδα υπηρεσιών και τεχνολογιών της εταιρείας Google, που τροφοδοτούν πολλές υπηρεσίες, όπως το δικτυακό τόπο της Google Maps, το Google Ride Finder, το Google Transit, και χάρτες ενσωματωμένους σε τρίτες ιστοσελίδες μέσω του Google Maps API. Προσφέρει οδικούς χάρτες, λειτουργίες για το σχεδιασμό διαδρομών για τα ταξίδια με τα πόδια, με αυτοκίνητο, με ποδήλατο, ή με τα μέσα μαζικής μεταφοράς και εντοπισμού για τις αστικές επιχειρήσεις σε πολλές χώρες σε όλο τον κόσμο.

Η βασική δομή του χάρτη προκύπτει μέσω δορυφορικών εικόνων. Αξίζει να σημειωθεί ότι αυτές οι εικόνες ανανεώνονται σποραδικά. Ο χρόνος που ανανεώνονται τα μέρη του χάρτη διαφέρει από μήνες σε χρόνια, ανάλογα με το πόσο δημοφιλής είναι η τοποθεσία και αν υπάρχει νέο υλικό για αυτή.

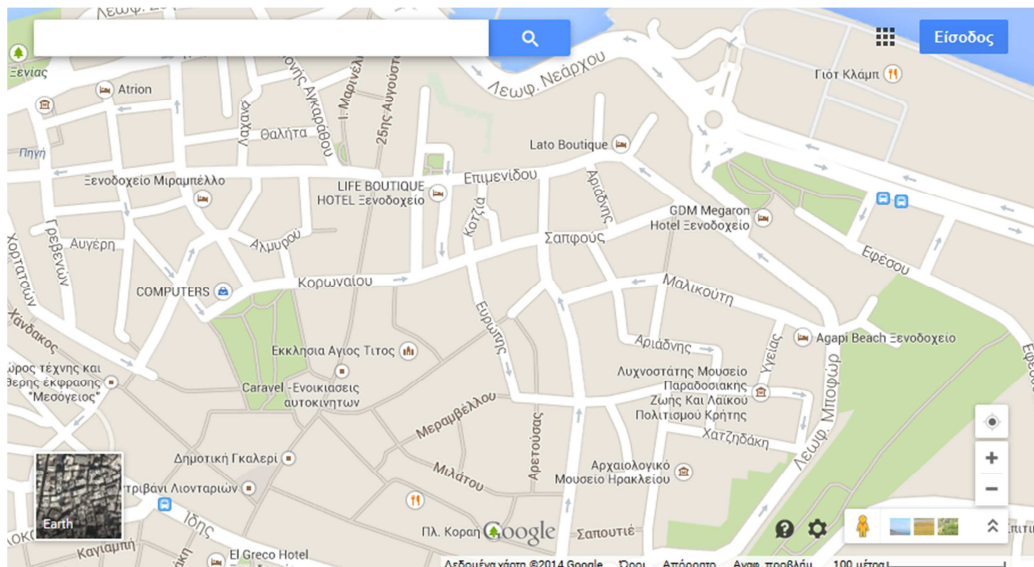
Κάθε χάρτης χρησιμοποιεί μία μέθοδο προβολής ώστε να δείξει τη σφαιρική επιφάνεια της γης σε μία επίπεδη εικόνα. Το Google Maps χρησιμοποιεί μια παραλλαγή της προβολής Mercator, η οποία επιλέχθηκε για να είναι πιο εύκολος ο υπολογισμός των συντεταγμένων. Όπως και όλοι οι χάρτες αυτό έχει σαν αποτέλεσμα την παραμόρφωση του

αποτελέσματος. Ειδικά για αυτή την προβολή το μέγεθος της στεριάς διογκώνεται όσο πλησιάζουμε στους πόλους και δεν είναι εφικτή η προβολή αυτών, αν και προσφέρεται η υπηρεσία Google Earth που μπορεί να καλύψει τέτοιες ανάγκες.

Παραπέρα θα δούμε μερικά αξιόλογα στοιχεία της υπηρεσίας, τα οποία αξίζουν εξέτασης είτε χρησιμοποιούμε στην εφαρμογή μας.

### 2.2.3 Προβολές

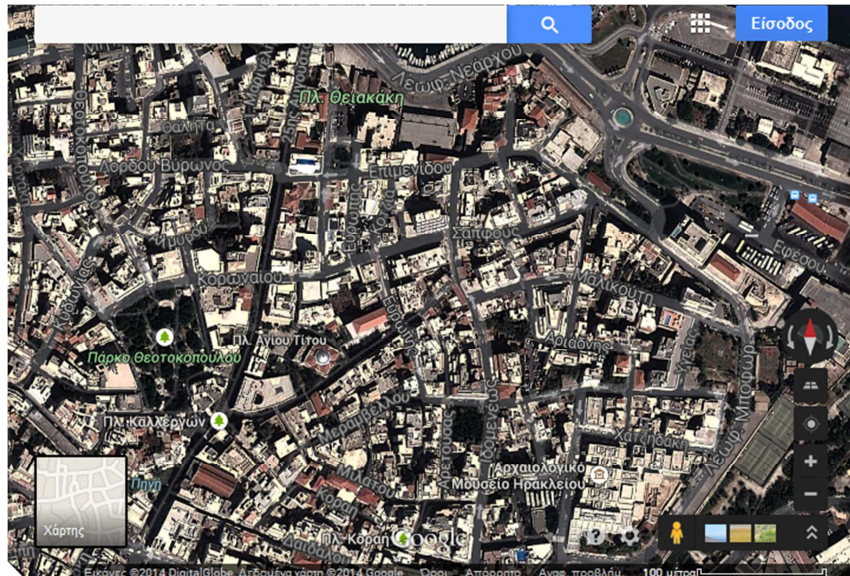
Το πρώτο στοιχείο που βλέπει κάποιος όταν χρησιμοποιήσει τα Google Maps είναι πως υπάρχουν διαθέσιμες διαφορετικών ειδών προβολές του ίδιου χάρτη. Η βασική είναι η προβολή δρόμων που δείχνει έναν απλό χάρτη στον οποίο φαίνονται μόνο οι δρόμοι που υπάρχουν σε αυτόν.



Εικόνα 2-1 Προβολή δρόμων

Η παραπάνω προβολή είναι αρκετά χρήσιμη για τον προσδιορισμό μίας διαδρομής και την εύρεση δρόμων. Τα στοιχεία ανανεώνονται τακτικά, και μάλιστα υπάρχουν στοιχεία για το αν ένας δρόμος είναι διπλής κατεύθυνσης ή μονόδρομος κ.α.

Όμως καθώς οι χάρτες δημιουργούνται από δορυφορικές φωτογραφίες είναι δυνατή και η πρόσβαση σε αυτές.

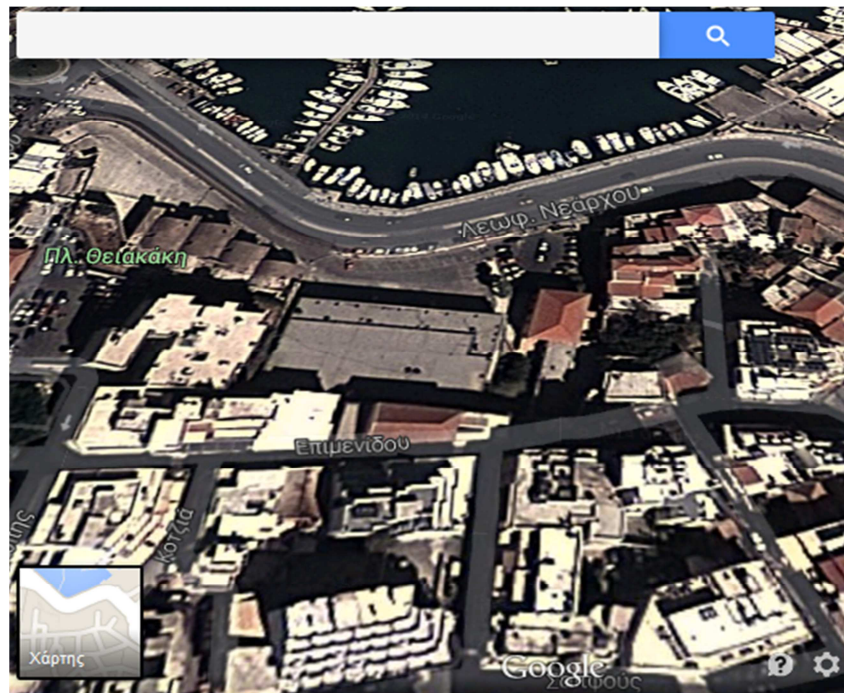


Εικόνα 2-2 Προβολή δορυφόρου

Το περιεχόμενο συνδυάζεται από υψηλής ανάλυσης αεροφωτογραφίες. Ανάλογα με το πόσο δημοφιλής κρίνεται μία περιοχή μπορεί να παρέχονται και μεγαλύτερης ανάλυσης αεροφωτογραφίες, δηλαδή να υπάρχει δυνατότητα για περισσότερα επίπεδα zoom.

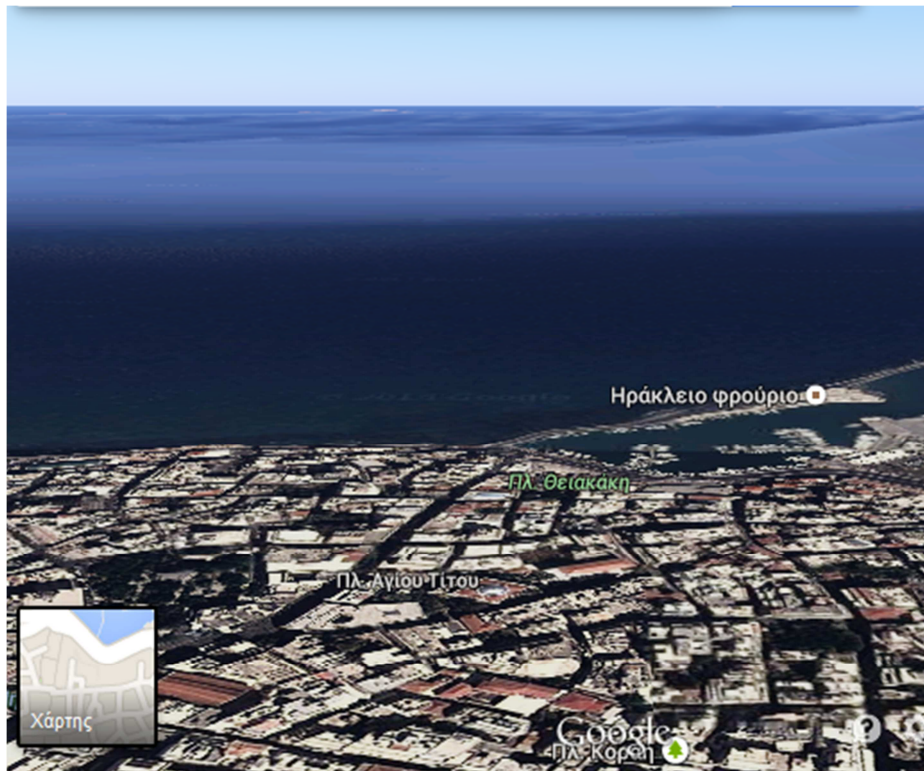
Αξίζει να σημειωθεί ότι περιοχές που θεωρούνται ευαίσθητες από τις εκάστοτε κυβερνήσεις, όπως ο Λευκός οίκος ή το αμερικάνικο Πεντάγωνο παρουσιάζονται επίτηδες με χαμηλή ανάλυση ή εντελώς μουτζουρωμένες.

Μία επιπλέον δυνατότητα είναι η εμφάνιση του χάρτη σε τρισδιάστατη μορφή.



Εικόνα 2-3 Τρισδιάστατη προβολή

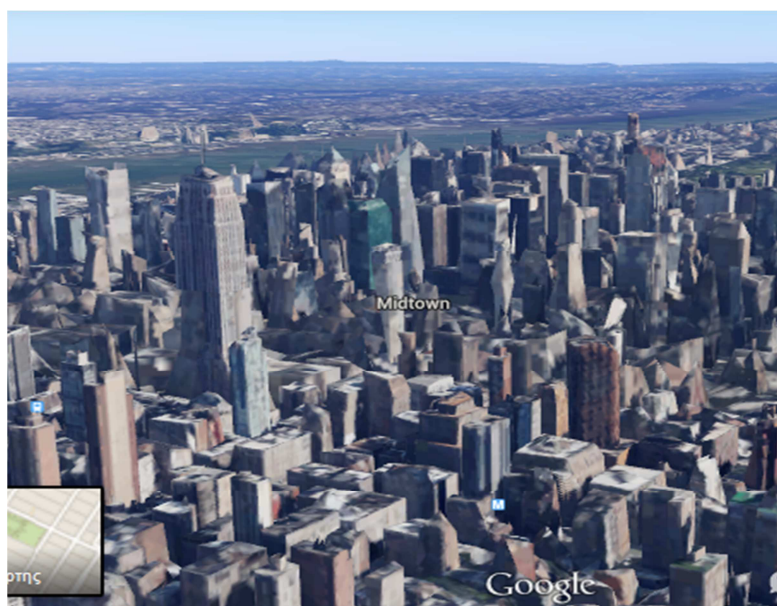
Αυτή μπορεί να έχει και διαφορετικά επίπεδα.



Εικόνα 2-4 Χάρτης με μεγάλη κλίση

Η πληροφορία για το ύψος των κτηρίων δεν περιέχεται στις φωτογραφίες αλλά δημιουργείται είτε αυτόματα με την ανάλυση τους από προγράμματα επεξεργασίας εικόνας είτε χειροκίνητα. Στην παραπάνω φωτογραφία που παρουσιάζεται η πόλη του Ηρακλείου δεν υπάρχει σχεδόν καθόλου λεπτομέρεια, αφού το Ηράκλειο δεν θεωρείται δημοφιλής τοποθεσία, τουλάχιστον με τα κριτήρια των χαρτών.

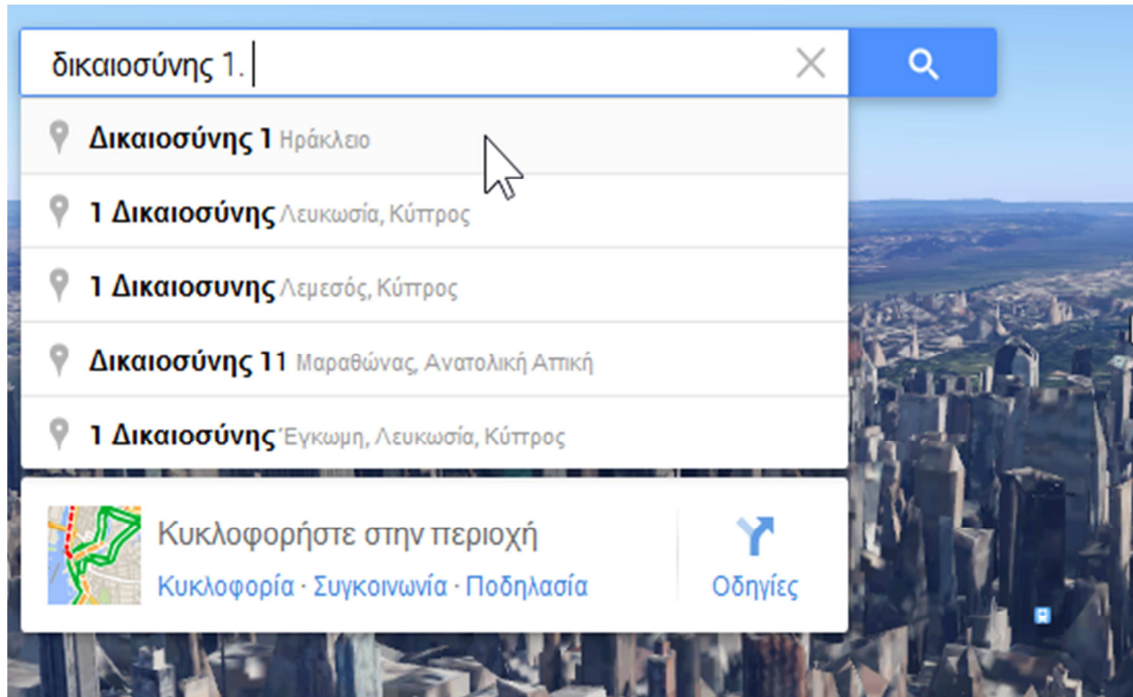
Αυτό όμως αλλάζει όταν πρόκειται για την Νέα Υόρκη.



Εικόνα 2-5 Πλήρης απεικόνιση των τρισδιάστατων χαρτών

## 2.2.4 Εύρεση τοποθεσίας

Η εταιρεία Google ειδικεύεται στην αναζήτηση και αυτή η λειτουργία υπάρχει και στους χάρτες της. Συγκεκριμένα με την εισαγωγή μίας διεύθυνσης μπορεί να γίνει αναζήτηση και να παρουσιαστεί στον χάρτη.



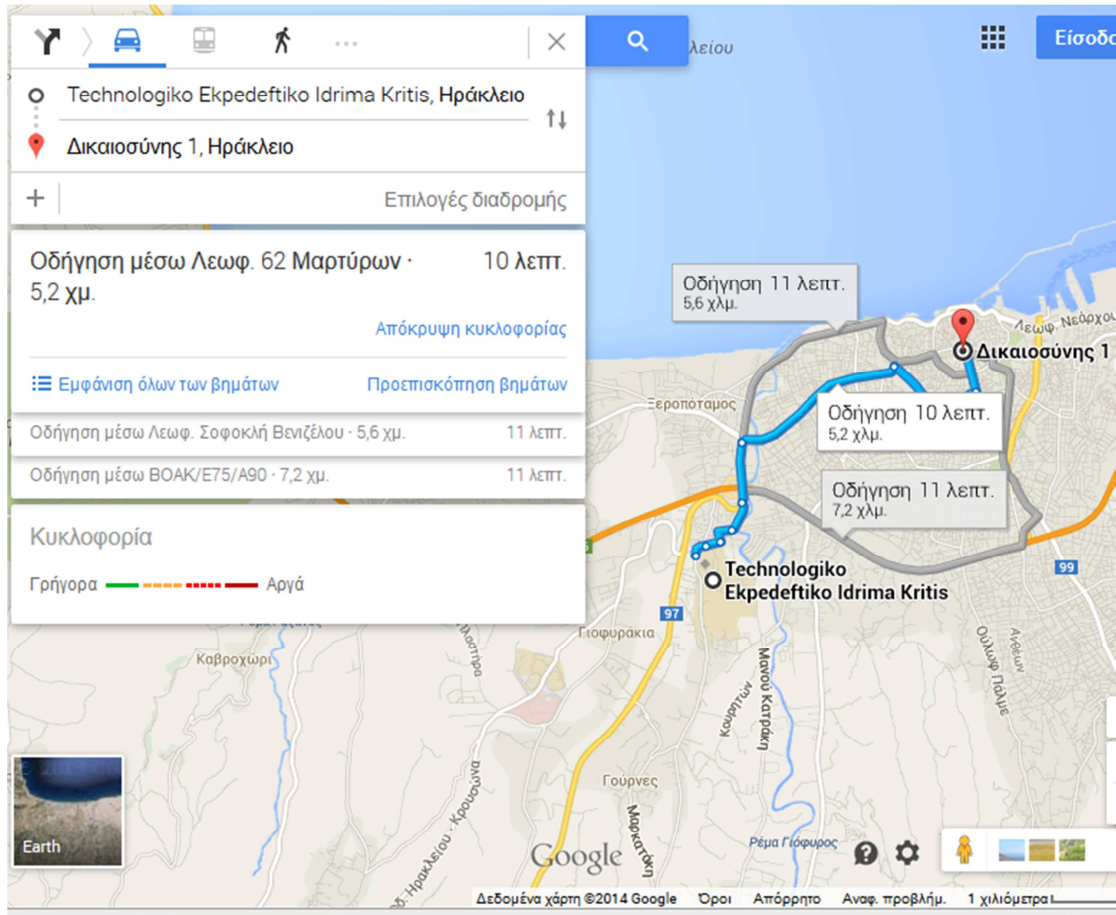
Εικόνα 2-6 Αναζήτηση τοποθεσίας

Αυτό σημαίνει πως για το μεγαλύτερο μέρος των χαρτών υπάρχει και πληροφορία σχετικά με τους δρόμους και τους αριθμούς τους, αν και κάποια ιδιαίτερα απομονωμένα μέρη του κόσμου εξαιρούνται.

Πέρα από τους δρόμους υπάρχουν και αρκετά ιδρύματα και αξιοθέατα.

## 2.2.5 Οδηγίες διαδρομής

Μία από τις βασικότερες λειτουργίες των χαρτών είναι οι οδηγίες για τη μετάβαση από ένα μέρος σε ένα άλλο. Αυτό γίνεται με την εισαγωγή δύο ή περισσότερων διευθύνσεων.



Εικόνα 2-7 Οδηγίες μετάβασης

Αν και αυτή η λειτουργία ήταν αρχικά περιορισμένη, πλέον παρέχονται πέρα από μία βασική διαδρομή και εναλλακτικές. Γίνεται αυτόματα υπολογισμός του χρόνου ανάλογα με την απόσταση και το είδος του κάθε δρόμου και τα όρια ταχύτητας που υπάρχουν σε αυτόν. Για αρκετά μέρη του κόσμου (όχι όμως για το Ηράκλειο) υπάρχουν επίσης πληροφορίες για την κίνηση του δρόμου ανάλογα με την ώρα.

Τέλος υπάρχουν διαφορετικά είδη οδηγιών ανάλογα με το αν η μετακίνηση θα γίνει με τα πόδια, με αυτοκίνητο (οπότε και λαμβάνονται υπ' όψιν οι μονόδρομοι) ή με δημόσιες συγκοινωνίες.

Ένα ιδιαίτερο χαρακτηριστικό είναι η ενσωμάτωση και των θαλάσσιων συγκοινωνιών, και μάλιστα η δυνατότητα σε πολλές τέτοιες διαδρομές ο χρήστης να μεταβεί επιτόπου στη σελίδα μίας ακτοπλοϊκής εταιρείας για έκδοση εισιτηρίου.

## 2.3 Τεχνολογίες web

Για την δημιουργία μίας απλής αλλά άρτιας ιστοσελίδας παλιότερα ήταν αρκετή η γνώση μόνο της γλώσσας HTML. Όμως σήμερα χρειάζονται τουλάχιστον άλλες δύο γλώσσες, η JavaScript και το CSS ώστε να πληρεί τις βασικές προϋποθέσεις ευχρηστίας και εικαστικής αρτιότητας.

Παρακάτω θα δείξουμε τα βασικά στοιχεία των γλωσσών αυτών. Επιπλέον θα κάνουμε μία παρουσίαση των σχετικών με το θέμα της εργασίας λειτουργιών των ηλεκτρονικών χαρτών της υπηρεσίας Google και της γλώσσας XML.

Οι τεχνολογίες που χρησιμοποιούνται για την συγγραφή μίας ιστοσελίδας ονομάζονται γλώσσες client. Με τον όρο τεχνολογίες client ή client-side, εννοούμε τα κομμάτια του κώδικα που εκτελούνται στον πελάτη. Στην περίπτωση μας ο client είναι ο χρήστης που επισκέπτεται την σελίδα της εφαρμογής και λαμβάνει από τον server τα κατάλληλα αρχεία.

Αυστηρά ο όρος πελάτης δεν αναφέρεται στον χρήστη αλλά στο πρόγραμμα που συνδέεται στον server. Αυτός είναι ο browser που χρησιμοποιείται καθώς ο ρόλος του είναι να ερμηνεύσει τις οδηγίες (δηλαδή τον κώδικα) που στέλνει ο server και να παρουσιάσει τη σελίδα ή να εκτελέσει τον κώδικα.

### 2.3.1 Η γλώσσα περιγραφής HTML

Η HTML (ακρωνύμιο του αγγλικού HyperText Markup Language) είναι η κύρια γλώσσα σήμανσης 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 διαβάσει τα έγγραφα HTML και να τα συνθέσει 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.

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD
2 "http://www.w3.org/TR/html4/stri
3 <html>
4 <head>
5   <title>Example</title>
6   <link rel="stylesheet" href="s
7 </head>
8 <body>
9   <div id="header">
10    <h1><a href="." title="Back
11  </div>
12  <div id="toolbar">
13    <span class="left">Today <sp
14    <span class="right">
15      <span id="time">&nbsp;</sp
16      <select id="timezone">
17        <option value="-12">(GMT
18        <option value="-11">(GMT
```



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.

Οποιαδήποτε σελίδα εμφανίζεται στον χρήστη είναι αποτέλεσμα των οδηγιών που δίνονται από τις τρεις διαφορετικές γλώσσες που αναφέραμε προηγουμένως, της HTML, της Javascript και του CSS. Η βασικότερη από αυτές είναι η HTML, όχι από άποψη δυσκολίας ή χρησιμότητας, αλλά γιατί πρόκειται για τον σκελετό της σελίδας, τον οποίο χρησιμοποιούν και τροποποιούν οι υπόλοιπες γλώσσες.

Ο στόχος της HTML είναι να περιγράψει κάθε ένα στοιχείο που περιέχει η σελίδα βάση του τύπου του, δηλαδή αν είναι μία επικεφαλίδα, μία παράγραφος ή ένα πεδίο κειμένου. Τα στοιχεία είναι ιεραρχικά τοποθετημένα, για παράδειγμα ένα στοιχείο μενού (nav) μπορεί να περιέχει μία επικεφαλίδα (που θα περιγράψει το είδος του μενού) και μία λίστα με επιλογές.

Πέρα από μία βασική δομή που πρέπει να έχουν όλες οι σελίδες και κάποιους γενικούς κανόνες δεν υπάρχει κάποιος περιορισμός στη σειρά των στοιχείων ή στο ποιο στοιχείο μπορεί να περιέχεται μέσα σε κάποιο άλλο. Αυτό εξαρτάται από το πώς θέλουμε να φαίνεται και τι να περιέχει η σελίδα. Για παράδειγμα το περιεχόμενο του στοιχείου header περιγράφει τα περιεχόμενα ενός άλλου στοιχείου. Το header μπορεί να βρίσκεται μέσα στο στοιχείο που περιγράφει, να προηγείται ή να έπεται.

Επιπλέον το ίδιο το περιεχόμενο του header δεν είναι δεδομένο. Μπορεί να περιέχει ένα στοιχείο με κείμενο επικεφαλίδας h1, το οποίο περιέχει το μεγαλύτερο κείμενο τίτλου. Όμως αν το header αναφέρεται σε κάποιο εσωτερικό στοιχείο της σελίδας μπορεί αντί για h1 να περιέχει h2. Επίσης μπορεί να υπάρχει ένα στοιχείο img που δείχνει μία εικόνα, πχ το λογότυπο της σελίδας.

Από τα παραπάνω φαίνεται πως δεν υπάρχει σωστός ή λάθος τρόπος για την συγγραφή του κώδικα HTML, αλλά είναι ένας συνδυασμός απλής λογικής και ιεραρχίας των στοιχείων της σελίδας. Αυτό αναφέρεται βέβαια στη δομή της σελίδας και τη διάρθρωση των στοιχείων, καθώς οι συντακτικοί κανόνες είναι απαραίτητο να ακολουθούνται.

### **2.3.2 Η γλώσσα μορφοποίησης CSS**

Η CSS (Cascading Style Sheets) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστότοπου. Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στιλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και να δίνει περισσότερες δυνατότητες σε σχέση με την html. Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα η χρήση της CSS κρίνεται ως απαραίτητη.

### **2.3.2.1 Η αλληλουχία εφαρμογής των φύλλων στυλ**

Για ένα έγγραφο `px.html` θα υπάρχουν παραπάνω από ένα φύλλα στυλ τα οποία περιέχουν δηλώσεις για την εμφάνιση ενός συγκεκριμένου στοιχείου. Το Φύλλο στυλ που εφαρμόζεται σε ένα έγγραφο μπορεί να προέρχεται από :

το συγγραφέα μιας ιστοσελίδας

το χρήστη του πλοηγού

τον ίδιο τον πλοηγό, αν έχει το δικό του προκαθορισμένο φύλλο στυλ .

Συνεπώς για ένα `html` στοιχείο θα υπάρχουν παραπάνω από μια δηλώσεις που πιθανόν να είναι συγκρουόμενες πρότυπου `css` για να επιλύσει παρόμοιες συγκρούσεις έχει καθορίσει μια αλληλουχία-σειρά στην οποία θα μπουν αυτές οι δηλώσεις και με βάση την οποία θα επιλεγεί `px` η δήλωση που είναι πρώτη στη σειρά.

Ο αλγόριθμος δημιουργίας αυτής της σειράς-αλληλουχίας είναι ο ακόλουθος:

Βρες όλες τις δηλώσεις που εφαρμόζονται στο στοιχείο που μας ενδιαφέρει. Οι δηλώσεις εφαρμόζονται στο στοιχείο αν ο επιλογέας του το επιλέξει (ταιριάζει με αυτό).

Ταξινόμησε με βάση τη σημασία (κανονική ή σημαντική) και προέλευση ( συγγραφέας , χρήστη ή πλοηγός χρήστη). Με αύξουσα σειρά προτεραιότητας:

Δηλώσεις πλοηγού χρήστη

Κανονικές δηλώσεις χρήστη

Κανονικές δηλώσεις συγγραφέα

Σημαντικές δηλώσεις συγγραφέα

Σημαντικές δηλώσεις χρήστη

Ταξινόμησε τις δηλώσεις ίδιας σημασίας και προέλευσης με κριτήριο την εξειδίκευση του επιλογέα: οι πιο εξειδικευμένοι επιλογείς υπερισχύουν των πιο γενικών. Τα ψευδό-στοιχεία και οι ψευδο-κλάσεις λογαριάζονται σαν κανονικά στοιχεία και κλάσεις αντίστοιχα.

Τέλος ταξινόμησε ανάλογα με τη σειρά καθορισμού: αν δύο δηλώσεις έχουν το ίδιο βάρος , προέλευση και εξειδίκευση , αυτή που προσδιορίστηκε τελευταία επικρατεί. Οι δηλώσεις σε εισαγόμενα φύλλα στυλ θεωρούνται ότι δηλώνονται πριν από τις δηλώσεις στο ίδιο το φύλλο στυλ .

### **2.3.2.2 Παράδειγμα**

Όπως αναφέραμε η γλώσσα HTML περιγράφει την δομή και το περιεχόμενο μίας σελίδας, όμως δεν περιέχει καμία πληροφορία για την εμφάνισή της. Κάθε στοιχείο εμφανίζεται σαν απλό κείμενο χωρίς σχεδόν καμία μορφοποίηση. Παρακάτω φαίνεται ένα απόσπασμα κώδικα που δημιουργεί μία παράγραφο και μία επικεφαλίδα.

```
<h1>This is a header</h1>
```

```
<p> A paragraph may contain text and images</p>
```

Εικόνα 2-8 Κώδικας HTML

Το αποτέλεσμα όταν το ανοίξουμε με ένα browser είναι το παρακάτω.

# This is a header

A paragraph may contain text and images

Εικόνα 2-9 Μία απλή σελίδα HTML

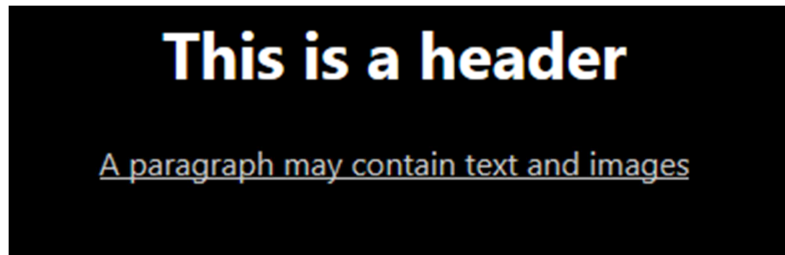
Όπως φαίνεται, αν και παρουσιάζεται όλο το περιεχόμενο που έχουμε εισάγει στον κώδικα, δεν γίνεται καμία οπτική διαφοροποίηση ανάμεσα στα στοιχεία, πέρα από τα πιο βασικά, δηλαδή οι τίτλοι χρησιμοποιούν πιο έντονα γράμματα, οι σύνδεσμοι έχουν άλλο χρώμα και το κουμπί είναι συμπαγές.

Ο ρόλος του CSS είναι να παρέχει τις κατάλληλες πληροφορίες για την εμφάνιση των στοιχείων. Παρακάτω προσθέσαμε στον κώδικα ένα στοιχείο style που δίνει πληροφορίες για την επιθυμητή εμφάνιση των στοιχείων.

```
<style>
  body {
    background-color: #000;
    font-family: Segoe UI;
  }
  h1 {
    font-size:22px;
    color: #FFF;
    text-align: center;
  }
  p {
    font-size:14px;
    color: #CCC;
    text-align: center;
    text-decoration: underline;
  }
</style>
<h1>This is a header</h1>
<p> A paragraph may contain text and images</p>
```

Εικόνα 2-10 Κώδικας HTML με στοιχεία για την εμφάνιση

Με αυτή την αλλαγή, χωρίς να πειράξουμε τίποτα άλλο στην HTML έχουμε το παρακάτω αποτέλεσμα:



Εικόνα 2-11 η ίδια σελίδα με την χρήση CSS

Βέβαια ο συνδυασμός HTML και CSS, αν και γίνεται είναι γενικά πολύ κακή πρακτική. Γενικός στόχος είναι να ξεχωρίζουμε το περιεχόμενο από την εμφάνιση, ώστε οι αλλαγές στο ένα να μην επηρεάζουν το άλλο. Για αυτό το λόγο τυπικά οι οδηγίες CSS δίνονται σε ένα ξεχωριστό αρχείο CSS, το οποίο συμπεριλαμβάνουμε στη σελίδα με την κατάλληλη εντολή. Για να εισάγουμε ένα αρχείο CSS προσθέτουμε μέσα στο στοιχείο head μία γραμμή που καθορίζει το αρχείο που θέλουμε να χρησιμοποιηθεί.

```
<!doctype html>  
<html>  
<head>  
  <meta charset="utf-8">  
  <title>Υπηρεσίες δρομολόγησης και καθοδήγησης</title>  
  <link href="style.css" rel="stylesheet" type="text/css">
```

Εικόνα 2-12 Η αρχή μίας σελίδας HTML που χρησιμοποιεί ένα αρχείο CSS

Το παραπάνω παράδειγμα δείχνει ξεκάθαρα τον ρόλο του CSS σε αντιδιαστολή με την HTML. Το αρχείο style.css που εισήγαμε έχει μέγεθος περίπου διακόσιες γραμμές και αλλάζει ριζικά τον τρόπο εμφάνισης της σελίδας. Όμως όλες οι πληροφορίες για την εμφάνιση είναι περιορισμένες σε αυτό, και για την χρήση του δεν χρειάζεται καμία επιπλέον αλλαγή στην HTML. Ο κώδικας της σελίδας παραμένει ξεκάθαρος και διαβάζοντάς τον είναι εύκολο να καταλάβουμε το περιεχόμενό της και τον ρόλο του κάθε στοιχείου.

### 2.3.3 Η γλώσσα προγραμματισμού JavaScript

#### 2.3.3.1 Εισαγωγή

Η JavaScript (JS) είναι διερμηνευμένη γλώσσα προγραμματισμού για ηλεκτρονικούς υπολογιστές.] Αρχικά αποτέλεσε μέρος της υλοποίησης των browser, ώστε τα client-side scripts να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται.

Η JavaScript είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξή της είναι επηρεασμένη από τη C. Η JavaScript αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java, αλλά γενικά οι δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm),

υποστηρίζοντας αντικειμενοστραφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού.

Η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων — τέτοια παραδείγματα είναι τα έγγραφα PDF, οι εξειδικευμένοι browser (site-specific browsers) και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets). Οι νεότερες εικονικές μηχανές και πλαίσια ανάπτυξης για JavaScript (όπως το Node.js) έχουν επίσης κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη server-side εφαρμογών ..

### **2.3.3.2 Λειτουργίες**

Παραπάνω είδαμε πως η δομή μιας σελίδας περιγράφεται από τη γλώσσα HTML και η εμφάνισή της από τη γλώσσα CSS. Είναι φανερό πως και οι δύο γλώσσες είναι στατικές, δηλαδή απλά δίνουν οδηγίες για το τι και πώς θα φαίνεται, και όχι για το πώς θα λειτουργεί.

Καθώς οι σελίδες HTML παρουσιάζουν περιεχόμενο στον χρήστη, αρκετές φορές αυτό αρκεί. Όμως υπάρχουν περιπτώσεις που θέλουμε να κάνουμε μία σελίδα να αντιδράει στις επιλογές του χρήστη με διάφορους τρόπους. Κάτι τέτοιο δεν είναι απαραίτητο, όμως μπορεί να κάνει την σελίδα πολύ πιο εύκολη στη χρήση.

Ένα παράδειγμα έχει να κάνει με τον έλεγχο στοιχείων μίας φόρμας. Ένα τυπικό σενάριο είναι μία φόρμα εγγραφής νέου χρήστη. Σε αυτή ο χρήστης συμπληρώνει τα στοιχεία του και υποβάλλει την φόρμα, δηλαδή στέλνει αυτά τα στοιχεία στον server.

Πριν γίνει η αποθήκευση των στοιχείων του νέου χρήστη χρειάζεται να γίνει έλεγχος των στοιχείων που στάλθηκαν, για παράδειγμα αν έχουν συμπληρωθεί όλα τα υποχρεωτικά πεδία και αν έχουν δοθεί έγκυρα στοιχεία, πχ σε ένα πεδίο που ζητάει το email του χρήστη. Αν τα στοιχεία δεν είναι έγκυρα τότε δεν μπορεί να γίνει αποθήκευση, οπότε είναι απαραίτητο να εμφανιστεί πάλι η φόρμα και να εμφανιστεί το κατάλληλο μήνυμα που να ειδοποιεί τον χρήστη για τα λάθη.

Παρόλο που η παραπάνω προσέγγιση λειτουργεί, είναι κουραστική και δυσνόητη για τον χρήστη. Αφού αυτός συμπληρώνει την φόρμα εγγραφής περιμένει να δει το αποτέλεσμα της ενέργειάς του. Το να εμφανίζεται πάλι η ίδια φόρμα μπορεί να τον αποπροσανατολίσει, ειδικά αν δεν έχει εμπειρία, ενώ χρειάζεται να καταβάλει προσπάθεια για καταλάβει τι λάθος έκανε.

Στο παραπάνω παράδειγμα η φόρμα θα μπορούσε να ελέγχεται με JavaScript πριν την υποβολή της. Με αυτόν τον τρόπο τα λάθη θα εμφανίζονταν μόλις ο χρήστης πατούσε το κουμπί υποβολής, χωρίς να ξαναφορτωθεί η σελίδα. Με αυτόν τον τρόπο μπορεί να καταλάβει αμέσως τι έχει συμβεί, αφού το λάθος εμφανίστηκε σαν αντίδραση στο πάτημα του κουμπιού, κάνοντας την εφαρμογή πιο φιλική.

Αξίζει να σημειωθεί πως αν και με τη JavaScript είναι δυνατόν να τροποποιηθεί το περιεχόμενο μιας σελίδας, υπάρχουν κάποιοι βασικοί περιορισμοί. Ο πρώτος είναι πως για την εκτέλεση πολύπλοκων λειτουργιών ο κώδικας JavaScript που απαιτείται είναι αρκετά πιο μπερδεμένος σε σχέση με κάποιες παραδοσιακές γλώσσες προγραμματισμού. Επιπλέον υπάρχουν αρκετές μικρές διαφοροποιήσεις στον ακριβή τρόπο λειτουργίας της JavaScript ανάμεσα σε διαφορετικούς browsers, κάτι το οποίο κάνει απαραίτητο τον έλεγχο της σελίδας σε κάθε έναν από αυτούς.

Ο βασικότερος περιορισμός της JavaScript είναι πως, αν και είναι δυνατή η επικοινωνία με τον server μέσω AJAX, δεν γίνεται με κανέναν τρόπο να επικοινωνήσει με την βάση δεδομένων, εκτός και αν παρεμβάλλεται μία σελίδα γραμμένη σε κάποια γλώσσα που εκτελείται στον server. Έτσι ο ρόλος της είναι βοηθητικός, στο να προσφέρει κάποια επιπλέον λειτουργικότητα, αλλά η δημιουργία της βασικής σελίδας γίνεται στον server.

### **2.3.3.3 JavaScript Frameworks**

Όπως αναφέραμε προηγουμένως κάθε browser έχει διαφορετικές υλοποιήσεις της JavaScript, οι οποίες μπορούν να διαφέρουν μεταξύ τους. Για να εξασφαλιστεί η απρόσκοπτη λειτουργία μίας σελίδας που χρησιμοποιεί JavaScript χρειάζεται να ελεγχθεί σε όλους τους πιθανούς browsers που θα την τρέξουν, κάτι το οποίο είναι εξαιρετικά επίπονο και πολλές φορές πρακτικά αδύνατο, καθώς δεν γίνεται να καλυφθούν όλες οι εκδόσεις του κάθε browser, ιδιαίτερα αν λάβουμε υπόψη μας και τους διαφορετικούς browsers που υπάρχουν σε ένα πλήθος κινητών τηλεφώνων και tablets.

Αν και υπάρχουν μικρές ασυμβατότητες στην υλοποίηση και τον τρόπο εμφάνισης των στοιχείων HTML και των κανόνων CSS, το πρόβλημα είναι μεγαλύτερο στην JavaScript γιατί πχ, μία διαφορά σε έναν κανόνα CSS συνήθως οδηγεί σε διαφορετική απεικόνιση ενός στοιχείου, ενώ ένα κομμάτι κώδικα JavaScript μπορεί να πάψει να λειτουργεί. Γι' αυτόν τον λόγο σπάνια γίνεται άμεσα χρήση εντολών μόνο της JavaScript που περιέχεται σε έναν browser, αλλά συνήθως χρησιμοποιείται ένα επιπλέον Framework.

Ένα framework είναι μία βιβλιοθήκη που περιέχει κάποια ήδη ορισμένα σύμβολα (μεταβλητές, κλάσης και συναρτήσεις) τα οποία παρέχουν κάποιες λειτουργίες, οι οποίες μπορεί να αντικαθιστούν τις λειτουργίες που περιέχει η JavaScript ή και να προσφέρουν επιπλέον δυνατότητες. Είναι και αυτό γραμμένο σε JavaScript, όμως με τέτοιο τρόπο ώστε να εγγυάται την συμβατότητα με αρκετούς browsers και πολλές φορές να προσφέρει έτοιμες κάποιες ενέργειες, συνήθως με την μορφή έτοιμων συναρτήσεων, οι οποίες αν και μπορούν να υλοποιηθούν με JavaScript είναι είτε τετριμμένες είτε αρκετά δύσκολες στην υλοποίηση.

Ένα βασικό framework μπορεί να προσφέρει κάποιες έτοιμες δυνατότητες, όμως αρκετά από αυτά είναι αρκετά πιο εξελιγμένα και έχουν συγκεκριμένους κανόνες χρήσης. Η εκμάθηση ενός framework χρειάζεται κάποια προσπάθεια, όμως αποδίδει γιατί στην πορεία μειώνει τον κόπο για την υλοποίηση των επιμέρους λειτουργιών που χρησιμοποιούνται και καθιστά περιττό τον έλεγχο σε κάθε browser.

Το βασικό μειονέκτημα που έχει η χρήση ενός framework, πέρα από τον κόπο που χρειάζεται για την εκμάθησή του, είναι πως τυπικά για την εκτέλεση μίας λειτουργίας απαιτείται η χρήση πιο πολύπλοκου κώδικα, τον οποίον ο προγραμματιστής δεν βλέπει. Αυτό μπορεί να έχει επιπτώσεις στην ταχύτητα εκτέλεσης των λειτουργιών σε μία σελίδα.

Η πτώση των επιδόσεων ήταν παλιότερα μεγάλο πρόβλημα, όμως με το πέρασμα του χρόνου τα περισσότερα frameworks έχουν βελτιστοποιηθεί αρκετά σε αυτόν τον τομέα, και σε συνδυασμό με τη συνεχή αύξηση των επιδόσεων των υπολογιστών έχει πάψει να είναι αποτρεπτικός παράγοντας.

### **2.3.3.4 jQuery**

Για την ανάπτυξη της εφαρμογής μας επιλέξαμε να χρησιμοποιήσουμε το framework jQuery. Αν και υπάρχουν πάρα πολλές παρόμοιες λύσεις το jQuery συνδυάζει μικρό μέγεθος, πάρα πολλές δυνατότητες και αρκετά καλές επιδόσεις, με αποτέλεσμα να

είναι ένα από τα πιο δημοφιλή frameworks. Είναι χαρακτηριστικό πως χρησιμοποιείται στο 65% των 10000 σελίδων με τις περισσότερες επισκέψεις. Πρόκειται για ένα λογισμικό ανοικτού κώδικα, που συντηρείται από μία ομάδα εθελοντών.

Η βασική διαφορά που έχει από την απλή JavaScript είναι πως αλλάζει αρκετά το βασικό συντακτικό, από την άποψη ότι κάνει χρήση μίας συνάρτησης που ονομάζεται jQuery και συνήθως χρησιμοποιείται για αυτή το σύμβολο του δολαρίου (\$).

Παρακάτω θα δώσουμε ένα παράδειγμα στο οποίο φαίνεται η διαφορά του jQuery από την απλή JavaScript, όπως φαίνεται μέσα από μία κλήση AJAX. Για να κάνουμε κάτι τέτοιο με απλή JavaScript απαιτείται ο παρακάτω κώδικας.

```
function loadXMLDoc() {
    var xmlhttp;

    // code for IE7+, Firefox, Chrome, Opera, Safari
    if (window.XMLHttpRequest) {
        xmlhttp=new XMLHttpRequest();
    } else { // code for IE6, IE5
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }

    xmlhttp.onreadystatechange=function() {
        if (xmlhttp.readyState==4 && xmlhttp.status==200) {
            document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
        }
    }

    xmlhttp.open("POST","ajax_info.txt",true);
    xmlhttp.send();
}
```

Όπως φαίνεται στην αρχή χρειάζεται να πάρουμε περιπτώσεις για διαφορετικούς browsers, ενώ μετά ορίζεται μία μέθοδος που θα καλεστεί αφού έρθει η απάντηση ή αν αποτύχει το αίτημα στον server. Τέλος ορίζεται η διεύθυνση και το τρόπος αποστολής του αιτήματος και αποστέλλονται οι πληροφορίες.

Για να γίνει η ίδια διαδικασία με το jQuery χρειάζεται ο παρακάτω κώδικας:

```
$.ajax({  
    url: "ajax_info.txt",  
    success: function(data) {  
        $("#myDiv").html(data);  
    }  
});
```

Απευθείας φαίνεται η σαφώς μικρότερη έκταση που καταλαμβάνει ο κώδικας, καθώς επίσης και η παράλειψη των περιπτώσεων που έχουν να κάνουν με το είδος και την έκδοση του browser. Επίσης φαίνεται η απλοποίηση των περιπτώσεων, δηλαδή ο έλεγχος σχετικά με το αν η απάντηση είναι σωστή έχει αντικατασταθεί με τον ορισμό μίας μεθόδου στην ιδιότητα success.

Ο κώδικας του jQuery είναι αρκετά πυκνός, πράγμα το οποίο μειώνει αρκετά την έκταση, αλλά την ίδια στιγμή είναι πιο εύκολο να γίνει κάποιο λάθος. Παραπάνω έχουμε μία συνάρτηση που παίρνει για παράμετρο έναν πίνακα – αντικείμενο javascript, ένα από τα στοιχεία του είναι μία συνάρτηση. Τέτοιες κλήσεις είναι αρκετά συνηθισμένες και απαιτούν από τον χρήστη να είναι αρκετά προσεκτικός.

### 2.3.4 Το πρότυπο HTML5

Κατά την ανάπτυξη της εφαρμογής χρησιμοποιήσαμε αρκετές δυνατότητες που προσφέρονται από την έκδοση 5 της γλώσσας HTML. Ο λόγος που δεν αναφέρουμε παραπάνω ρητά την χρήση της τεχνολογίας HTML5 είναι η αρκετά πολύπλοκη διαδικασία που ακολουθείται για την ανάπτυξη της συγκεκριμένης τεχνολογίας.

Γενικά οι εκδόσεις των γλωσσών προγραμματισμού καθορίζονται από παγκόσμια πρότυπα. Ένα πρότυπο περιέχει λεπτομερείς οδηγίες σχετικά με τις δυνατότητες και τον τρόπο λειτουργίας της γλώσσας, το οποίο συνήθως γίνεται αποδεκτό και παγιώνεται έπειτα από κοινή συμφωνία των κυριότερων φορέων της βιομηχανίας που αφορά η γλώσσα. Έπειτα τα προγράμματα τα οποία χρησιμοποιούν τη γλώσσα (συνήθως compilers, interpreters ή στην περίπτωση της HTML οι browsers) τροποποιούνται για να υποστηρίξουν τις νέες δυνατότητες.



Ο όρος παγιώθηκε χρησιμοποιείται συνήθως για τα πρότυπα γιατί αρχικά παρουσιάζεται μία υποψήφια πρόταση (draft) η οποία στη συνέχεια τροποποιείται βάσει της κριτικής που δέχεται από τη βιομηχανία. Όταν όλα τα μέρη που συμμετέχουν στη διαδικασία συμφωνήσουν (ή καταλήξουν σε μία γενικά αποδεκτή έκδοση) τότε αυτό το draft παγιώνεται σαν μία επίσημη έκδοση.

#### **2.3.4.1 Περιορισμοί πριν την HTML5**

Η τελευταία έκδοση της HTML είναι η έκδοση 4.01 που παγιώθηκε το 1999. Με την έκδοση 4.01 ο ρόλος που προβλεπόταν να έχει η HTML ήταν η συγγραφή σελίδων κειμένου, με την προσθήκη αρκετών για την εποχή επιλογών μορφοποίησης (καθώς το πρότυπο περιέγραφε τη γλώσσα HTML, το CSS και κάποιες βασικές λειτουργίες JavaScript). Όμως η ραγδαία αύξηση των χρηστών internet οδήγησε στην μέγιστη εκμετάλλευση όλων των παρεχόμενων δυνατοτήτων, καθώς οι σελίδες είτε ήταν αρκετά περίτεχνες από αισθητική άποψη ή είχαν λειτουργικότητα που συναγωνίζονταν τις κλασσικές εφαρμογές.

Αυτή η χρήση οδήγησε στην εμφάνιση αρκετών αδυναμιών του προτύπου, η οποία εστιαζόταν σε δύο τομείς. Ο πρώτος ήταν η έλλειψη αρκετών δυνατοτήτων και ο δεύτερος ήταν η έλλειψη ακρίβειας στην περιγραφή αρκετών στοιχείων της γλώσσας. Αυτά οδήγησαν στον κατακερματισμό της αγοράς, καθώς κάθε browser είχε μικρές διαφορές στην λειτουργία του από τους υπόλοιπους, κάνοντας την ανάπτυξη μίας σελίδας που να λειτουργεί με τον ίδιο τρόπο σε όλους τους browsers εξαιρετικά δύσκολη. Επιπλέον παρουσιάστηκε μία πληθώρα τρίτων τεχνολογιών όπως το Adobe Flash, Java Applets, Microsoft Silverlight με σκοπό να προσφέρουν την επιπλέον λειτουργικότητα, αλλά έχοντας το πρόβλημα πως δεν ήταν πάντα συμβατά με όλους τους browsers.

#### **2.3.4.2 Δημιουργία της HTML5**

Σε αντίθεση με τις προηγούμενες εκδόσεις της HTML οι οποίες στόχευαν στο να δημιουργήσουν μία γλώσσα κατάλληλη για την περιγραφή της εμφάνισης των σελίδων και την βασική αλληλεπίδρασή τους με τον χρήστη, η HTML5 περιλαμβάνει πολλά τμήματα με τα οποία θα μπορούν να δημιουργηθούν ανεξάρτητες και αρκετά πολύπλοκες εφαρμογές. Βάση το εύρους των θεμάτων που καλύπτουν οι προτάσεις που αναπτύσσονται μπορεί να την χαρακτηριστεί σαν μία πλήρης γλώσσα προγραμματισμού.

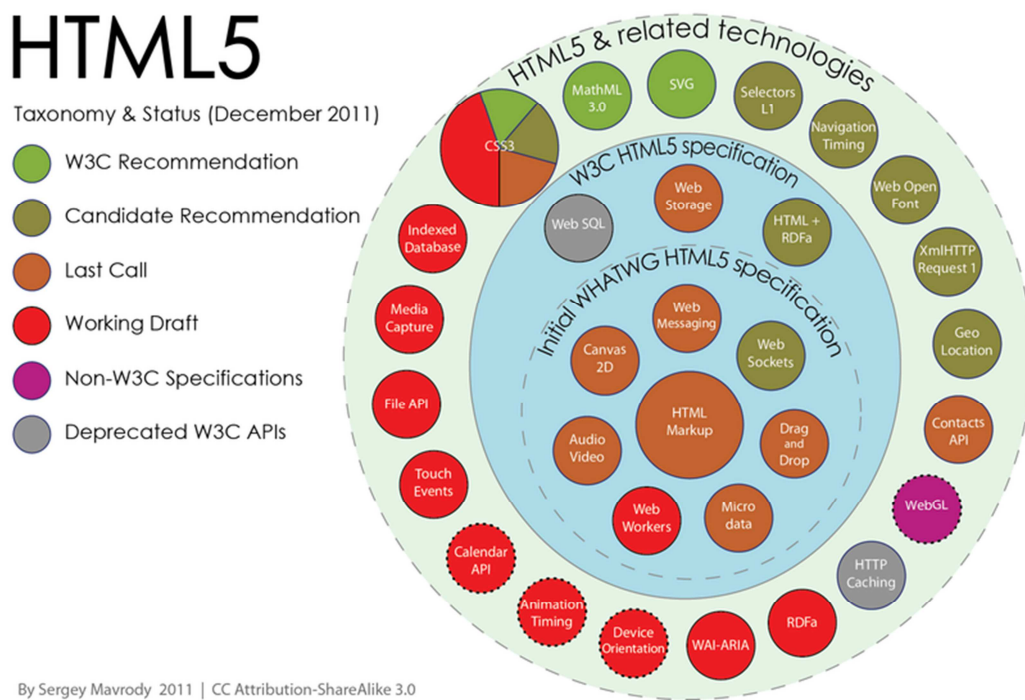
Αυτό οφείλεται στο ότι καλείται να αντικαταστήσει ένα πλήθος τρίτων τεχνολογιών που σήμερα χρησιμοποιούνται στις υπάρχουσες ιστοσελίδες χωρίς να υπολείπεται σε δυνατότητες από αυτές. Επιπλέον πολλές από τις αλλαγές που έχουν προταθεί επιδιώκουν να διορθώσουν άμεσες ελλείψεις και κενά της γλώσσας.

Σε αντίθεση με τις προηγούμενες εκδόσεις της HTML οι οποίες στόχευαν στο να δημιουργήσουν μία γλώσσα κατάλληλη για την περιγραφή της εμφάνισης των σελίδων και την βασική αλληλεπίδρασή τους με τον χρήστη, η HTML5 περιλαμβάνει πολλά τμήματα με τα οποία θα μπορούν να δημιουργηθούν ανεξάρτητες και αρκετά πολύπλοκες εφαρμογές. Βάση το εύρους των θεμάτων που καλύπτουν οι προτάσεις που αναπτύσσονται μπορεί να την χαρακτηριστεί σαν μία πλήρης γλώσσα προγραμματισμού.

Αυτό οφείλεται στο ότι καλείται να αντικαταστήσει ένα πλήθος τρίτων τεχνολογιών που σήμερα χρησιμοποιούνται στις υπάρχουσες ιστοσελίδες χωρίς να υπολείπεται σε δυνατότητες από αυτές. Επιπλέον πολλές από τις αλλαγές που έχουν προταθεί επιδιώκουν να διορθώσουν άμεσες ελλείψεις και κενά της γλώσσας.

Οι αλλαγές προφανώς δεν περιέχουν αλλαγές μόνο στην γλώσσα HTML αλλά προσθέτουν πολλά καινούργια στοιχεία στη JavaScript και το CSS. Επίσης, με σκοπό να μην επαναληφθούν προβλήματα που παρουσιάστηκαν στο παρελθόν, η διατύπωση των προτύπων γίνεται με αρκετά μεγάλη λεπτομέρεια ώστε να είναι ξεκάθαρος ο τρόπος υλοποίησής τους.

Ένα πρότυπο τέτοιου μεγέθους τυπικά χρειάζεται είκοσι χρόνια για να οριστικοποιηθεί, κάτι που τοποθετεί την τελική ημερομηνία στο 2022. Όμως η μεγαλύτερη διαφορά που έχει η γλώσσα HTML5 σε σχέση με παρόμοια πρότυπα είναι το γεγονός ότι οι ανάγκες που καλείται να ικανοποιήσει είναι ιδιαίτερα άμεσες και πιεστικές. Επιπλέον οι συμμετέχοντες στην διαδικασία και αυτοί που τελικά θα το υλοποιήσουν είναι οι εταιρείες και οι οργανισμοί που δημιουργούν τους σύγχρονους browsers.



By Sergey Mavrody 2011 | CC Attribution-ShareAlike 3.0

Εικόνα 2-13 Τα τμήματα του προτύπου HTML5 και τα στάδια ολοκλήρωσης του καθενός

Για αυτό τον λόγο το πρότυπο χωρίστηκε σε αρκετά επιμέρους τμήματα, κάθε ένα από τα οποία αναπτύσσεται και εγκρίνεται ξεχωριστά. Αυτό σημαίνει πως κάποια από αυτά έχουν ήδη ολοκληρωθεί και εγκριθεί, κάποια άλλα υπάρχουν απλά σαν προτάσεις και τα υπόλοιπα βρίσκονται σε διάφορες ενδιάμεσες καταστάσεις.

**2.3.4.3 Συμβατότητα μίας εφαρμογής με την HTML5**

Όπως φαίνεται από τα παραπάνω η HTML5 δεν υπάρχει σαν ένα ολοκληρωμένο πρότυπο. Όμως αρκετά τμήματά του προτύπου είναι ήδη ολοκληρωμένα και έχουν υλοποιηθεί στους σύγχρονους browsers. Μία εφαρμογή που είναι γραμμένη σε HTML5 πρακτικά είναι γραμμένη σε HTML καθώς η νέα έκδοση είναι σε μεγάλο βαθμό συμβατή με την προηγούμενη και χρησιμοποιεί κάποιες από τις επιπλέον δυνατότητες.

Με αυτή τη λογική η εφαρμογή μας είναι γραμμένη σε HTML5. Τα στοιχεία της HTML5 που χρησιμοποιούμε είναι τα παρακάτω:

Κάποια tag που ορίστηκαν στην Πέμπτη έκδοση της γλώσσας, όπως τα header, footer, nav.

Κάποιες ιδιότητες που χρησιμοποιούμε στο αρχείο CSS, όπως το min-height.

Κάποιες δυνατότητες της JavaScript που χρησιμοποιούνται κυρίως εσωτερικά από το JQuery.

## 2.3.5 Η γλώσσα προγραμματισμού Php

### 2.3.5.1 Γενικά

Η PHP, ακρωνύμιο του όρου PHP Hypertext Preprocessor, είναι μια γλώσσα προγραμματισμού ανοιχτού κώδικα, η οποία χρησιμοποιείται ευρέως για τη συγγραφή κομματιών κώδικα (scripts) που τρέχουν στο web server και όπως θα δείξουμε παρακάτω αποτελεί άριστο εργαλείο για την ανάπτυξη δικτυακών εφαρμογών που χρησιμοποιούν βάσεις δεδομένων.

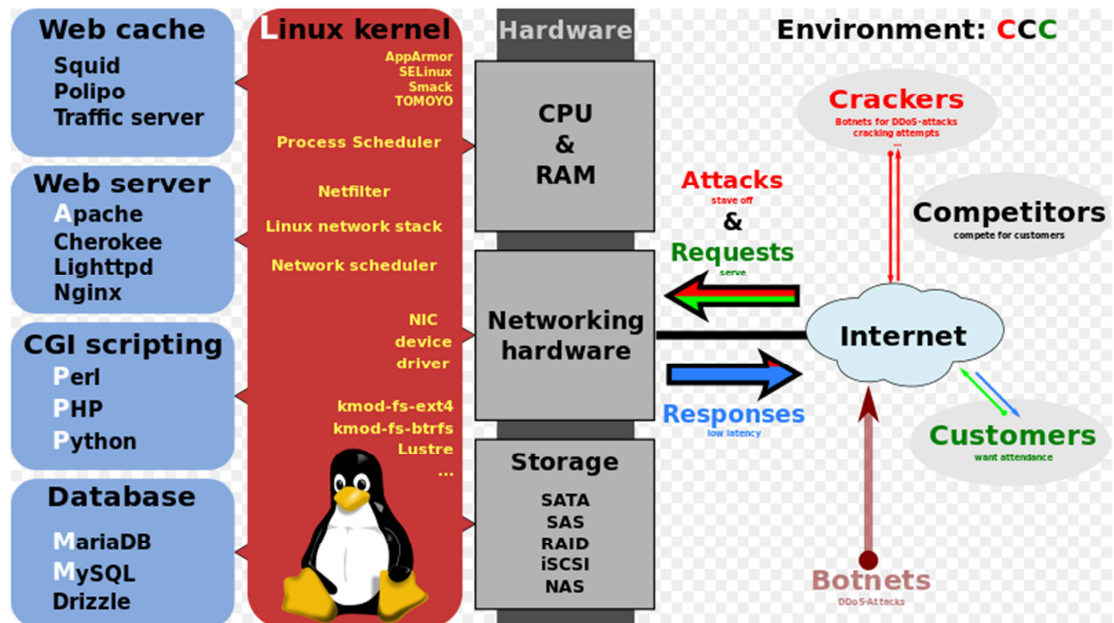
Η PHP πρωτοπαρουσιάστηκε το 1994 και από τότε έχει υποστεί πάρα πολλές αλλαγές μέχρι την πρόσφατη έκδοση η οποία είναι και η πιο δημοφιλής.

Τα χαρακτηριστικά που την καθιστούν την πιο δημοφιλή επιλογή είναι τα παρακάτω:

- Είναι γλώσσα ανοιχτού κώδικα (Open Source). Αυτό σημαίνει ότι διατίθεται δωρεάν, (με την άδεια χρήσης: [http://www.php.net/license/3\\_0.txt](http://www.php.net/license/3_0.txt)) αλλά και ότι υποστηρίζεται από μια διεθνής κοινότητα προγραμματιστών, η οποία συμβάλλει στη γρήγορη ανάπτυξη και βελτίωση σφαλμάτων του πυρήνα της γλώσσας.
- Έχει πολύ καλές επιδόσεις, κυρίως σε μικρής και μεσαίας κλίμακας εφαρμογές, χωρίς να απαιτεί ακριβό εξοπλισμό. Διάφορα συγκριτικά τεστ στο διαδίκτυο αποδεικνύουν την υπεροχή της PHP έναντι του ανταγωνισμού:
- Είναι πολύ εύκολη στην κατανόηση (η σύνταξη μοιάζει πολύ με της C και της Perl) και αρκετά γρήγορη στη ανάπτυξη του κώδικα, μειώνοντας κατά πολύ το χρόνο ανάπτυξης μιας εφαρμογής σε σχέση με πιο πολύπλοκες γλώσσες. Επίσης, υπάρχουν πολλές ενσωματωμένες συναρτήσεις οι οποίες παρέχουν χρήσιμες λειτουργίες, όπως διαχείριση αρχείων, αποστολή email, δημιουργία εγγράφων PDF, διαχείριση cookies κ.α.
- Μπορεί να λειτουργήσει χωρίς μετατροπές σε πολλές διαφορετικές πλατφόρμες (Linux, FreeBSD, Solaris, MS Windows). Η PHP υποστηρίζει τους Apache, MS IIS, Netscape και iPlanet servers και πολλούς άλλους web servers. Επίσης διαθέτει συνδέσεις για πολλά συστήματα βάσεων δεδομένων μεταξύ των οποίων: MySQL, PostgreSQL, Oracle, Informix, Sybase, κ.α. Ο συνδυασμός Apache- PHP - MySQL είναι αυτός που χρησιμοποιείται ευρέως σε δικτυακές εφαρμογές μικρής και μέσης κλίμακας και θεωρείται ως ο πλέον σταθερός και αποδοτικός.
- Ένα PHP script μπορεί να ενσωματωθεί πολύ απλά σε ένα έγγραφο HTML. Όταν αυτό το έγγραφο ζητηθεί, η μηχανή μετάφρασης PHP στον web server εκτελεί το script και το παραγόμενο αποτέλεσμα στέλνεται στον web browser, ο οποίος δεν μπορεί να δει το αρχικό script. Αυτό σημαίνει ότι η PHP είναι ανεξάρτητη από το web browser που χρησιμοποιείται στο στρώμα πελάτη (client tier).
- Επειδή η γλώσσα έχει γίνει τόσο δημοφιλής κυκλοφορούν ευρέως πολλά κομμάτια κώδικα ακόμα και προγραμματιστικά πακέτα, γραμμένα σε PHP, τα οποία

διατίθενται ελεύθερα και κάνουν ακόμη πιο γρήγορη την ανάπτυξη μιας δικτυακής εφαρμογής. Μια τέτοια κοινή αποθήκη είναι η PEAR (<http://pear.php.net>)

Οι παραπάνω λόγοι εξηγούν τους λόγους για τους οποίους η PHP θεωρείται ως η ενδεδειγμένη γλώσσα για την ανάπτυξη μιας δικτυακής εφαρμογής σαν τη δική μας.



Μια ένδειξη για την έκρηξη της δημοτικότητας αποτελεί η έρευνα της εταιρείας Netcraft (<http://www.netcraft.com/survey/>) η οποία δείχνει ότι η PHP χρησιμοποιείται από 18.500.000 περίπου domain στο Internet.

Η τελευταία έκδοση, PHP 5, υποστηρίζει περισσότερα αντικειμενοστραφή χαρακτηριστικά, XML, διαχείριση σφαλμάτων και άλλα χαρακτηριστικά που την φέρνουν πιο κοντά σε πιο ολοκληρωμένες γλώσσες προγραμματισμού, όπως η Java και η C++.

Παραθέτουμε ένα μικρό παράδειγμα για να δείξουμε πώς ενσωματώνεται η PHP σε ένα αρχείο HTML:

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <?php
      $output="Hello World";
      echo $output;
    ?>
  </body>
</html>
```

Το παράδειγμα αυτό δείχνει τη διαφορά της PHP από άλλες γλώσσες προγραμματισμού όπως η Perl ή η C. Αντί να γράφεται ένα πρόγραμμα με πολλές εντολές για να παράγει HTML, γράφεται ένα HTML script με ενσωματωμένο κώδικα PHP για να κάνει κάτι- στο παραπάνω παράδειγμα να εμφανίσει κάποιο κείμενο. Ο κώδικας PHP

περικλείεται από τις ειδικές ετικέτες αρχής και τέλους (<?php και ?>), οι οποίες επιτρέπουν την εύκολη μεταφορά μέσα και έξω από τη «λειτουργία PHP». Αφού η μηχανή μετάφρασης της PHP στον web server εκτελέσει το script, στον web browser στέλνεται τελικά ο παρακάτω HTML κώδικας:

```
<html>

  <head>

    <title>Example </title>

  </head>

  <body>

    Hello World

  </body>

</html>
```

Βλέπουμε ότι το κομμάτι με το PHP script έχει αντικατασταθεί με το παραγόμενο αποτέλεσμα του (εδώ, την εμφάνιση ενός μηνύματος) με αποτέλεσμα να κρύβεται από τον τελικό χρήστη ο κώδικας που βρίσκεται πίσω από την εμφάνιση της ιστοσελίδας. Για αυτό και η PHP ονομάζεται server-side γλώσσα προγραμματισμού, επειδή τρέχει στο server σε αντίθεση από την Javascript, για παράδειγμα, η οποία τρέχει στην πλευρά του χρήστη (client-side).

Μια άλλη παρατήρηση από το παραπάνω παράδειγμα είναι ότι η PHP δεν απαιτεί τον ορισμό των μεταβλητών πριν αυτές χρησιμοποιηθούν. Εδώ η μεταβλητή \$output ορίζεται κατά τη μετάφραση ως μεταβλητή τύπου string, επειδή αρχικοποιήθηκε με το string «Hello World». Αυτό το χαρακτηριστικό οδηγεί στην πιο εύκολη και γρήγορη συγγραφή κώδικα, με το μειονέκτημα ότι πρέπει ο προγραμματιστής να είναι προσεκτικός στην τιμή και τον τύπο που παίρνουν οι μεταβλητές που χρησιμοποιεί.

### **2.3.5.2 Επικοινωνία της PHP με βάσεις δεδομένων**

Όπως ειπώθηκε ήδη, η PHP θεωρείται ιδανική γλώσσα για τη συγγραφή δυναμικών εφαρμογών που επικοινωνούν με βάσεις δεδομένων, αφού διαθέτει ενσωματωμένες συναρτήσεις που επικοινωνούν με πλήθος εμπορικών συστημάτων βάσεων δεδομένων. Επίσης υπάρχει μια αφαιρετική επέκταση DBX βάσεων δεδομένων (DBX database abstraction extension) που επιτρέπει να χρησιμοποιείται διάφανα οποιαδήποτε βάση δεδομένων υποστηρίζεται από αυτή την επέκταση. Επιπλέον η PHP υποστηρίζει το ODBC, το Open Database Connection standard (Ανοιχτό πρότυπο Σύνδεσης Βάσεων δεδομένων) κι έτσι μπορεί να συνδεθεί σε οποιαδήποτε βάση δεδομένων που υποστηρίζει αυτό το παγκόσμιο πρότυπο.

Εδώ παρουσιάζουμε συνοπτικά, με τη βοήθεια ενός παραδείγματος, τις συναρτήσεις της PHP που χρησιμοποιούνται για τη σύνδεση με μια βάση δεδομένων στη

MySQL, τη δημιουργία ερωτήσεων και την ανάκτηση των αποτελεσμάτων για περαιτέρω επεξεργασία τους από την εφαρμογή.

Έστω το ακόλουθο κομμάτι κώδικα:

```
<?php

    //Δημιουργία σύνδεσης προς τον MySQL Server

    $connection =
mysql_connect("localhost", "username", "passwd");

    //Εμφάνιση κωδικού σφάλματος σε περίπτωση αποτυχημένης σύνδεσης
    if (!$connection) {
        die('Could not connect: ' . mysql_error());
    }

    //Επιλογή μιας βάσης δεδομένων

    mysql_select_db("mydb", $connection);

    // Δημιουργία μιας ερωτήσης (query) προς τη βάση

    $result = mysql_query ("SELECT * FROM mytable",
    $connection);

    // Επιστροφή των αποτελεσμάτων του query

    while ($row = mysql_fetch_array($result, MYSQL_NUM))

    {

        // επεξεργασία των αποτελεσμάτων

    }

    //Κλείσιμο της σύνδεσης με τον Server

    mysql_close ($connection);

?>
```

- Η συνάρτηση `mysql_connect()` χρησιμοποιείται για τη σύνδεση με τον MySQL Server και δέχεται 3 παραμέτρους: το hostname ή την IP διεύθυνση του server, το όνομα του χρήστη που έχει πρόσβαση στον server και τον κωδικό ασφαλείας του. Η τιμή που επιστρέφει η συνάρτηση χρησιμοποιείται σαν παράμετρος στις μετέπειτα συναρτήσεις που καλούνται προς τον MySQL Server.
- Η συνάρτηση `mysql_error()` επιστρέφει το μήνυμα λάθους σε περίπτωση αποτυχίας της προηγούμενης πράξης που έγινε προς τον MySQL Server.
- Η συνάρτηση `mysql_select_db()` επιλέγει μια συγκεκριμένη βάση δεδομένων η οποία έχει ήδη δημιουργηθεί στον MySQL Server. Οι παράμετροι που δίνονται είναι το όνομα της βάσης και η τιμή που επιστράφηκε από την `mysql_connect()`.

- Η συνάρτηση `mysql_query()` χρησιμοποιείται για την αποστολή μιας επερώτησης (query) στη βάση δεδομένων που ορίζει η δεύτερη παράμετρος (`$connection`). Η πρώτη παράμετρος είναι το query που θέλουμε να εκτελεστεί. Η σύνταξη της επερώτησης SQL είναι ακριβώς όμοια με τη σύνταξη που χρησιμοποιείται κατά τη δημιουργία ενός query από το command line interface του MySQL Server.
- Η συνάρτηση `mysql_fetch_array()` επιτρέπει την ανάκτηση των εγγραφών (rows) που επέστρεψε προηγούμενο query προς τη βάση.
- Επίσης αν δεν μας ενδιαφέρουν τα δεδομένα που επιστράφηκαν, αλλά το πόσες εγγραφές επηρεάστηκαν από την επερώτηση μπορεί να χρησιμοποιηθεί η συνάρτηση `mysql_affected_rows()`.
- Τέλος η συνάρτηση `mysql_close()` μπορεί να χρησιμοποιηθεί για να κλείσει την σύνδεση που ορίζει η παράμετρος `$connection`, αν και αυτό δεν είναι απαραίτητο αφού όλες οι συνδέσεις προς την βάση δεδομένων κλείνουν αυτόματα με το τέλος του script.

### 2.3.5.3 Εγκατάσταση της PHP

Η τελευταία έκδοση της PHP, σε μορφή κώδικα για εγκατάσταση σε Linux και μεταγλωττισμένη για εγκατάσταση σε Windows, μπορεί να ανακτηθεί από τον δικτυακό τόπο της PHP:

<http://www.php.net/downloads.php>.

Επίσης, στη διεύθυνση: <http://www.php.net/download-docs.php> βρίσκεται η τεκμηρίωση της PHP, η οποία εκτός των άλλων περιλαμβάνει λεπτομερείς οδηγίες για την εγκατάσταση της PHP σε διάφορα λειτουργικά και εξυπηρετητές δικτύου.

Στον server μας εγκαταστάθηκε η έκδοση 4.3.9 της PHP, με υποστήριξη για τον Apache2 και τον MySQL server. Στο Παράρτημα Α παραθέτουμε τα αποτελέσματα της συνάρτησης `phpinfo()` η οποία εμφανίζει το σύνολο των πληροφοριών που αφορούν το περιβάλλον της εγκατάστασης την PHP στον server μας.

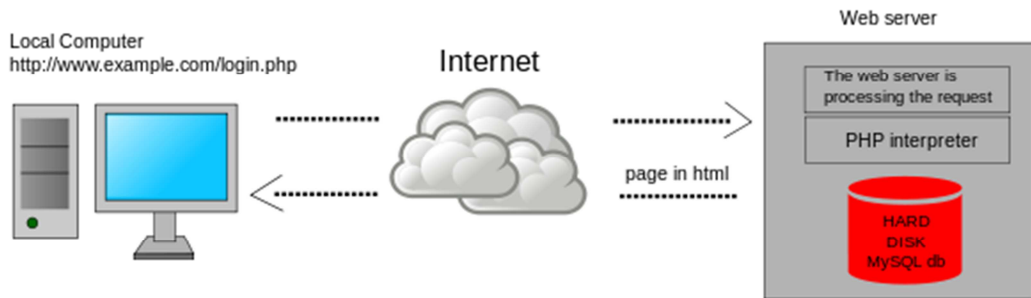
Μετά την επιτυχή εγκατάσταση της PHP, μπορούμε να ρυθμίσουμε διάφορες παραμέτρους της που αφορούν την ασφάλεια ή την επίδοση του συστήματος, στο αρχείο κειμένου 'php.ini'. Στο κεφάλαιο 8, όπου μελετάμε το θέμα της ασφάλειας αναφερόμαστε στην αλλαγή τριών μεταβλητών που πρέπει να γίνουν για να είναι η εφαρμογή μας πιο ασφαλής.

Τέλος, για να ορίσουμε στον Apache web server πως να εκτελεί PHP scripts αρκεί να προσθέσουμε την παρακάτω γραμμή στο αρχείο ρυθμίσεων του Apache 'httpd.conf', στο σημείο όπου δηλώνονται τα πρόσθετα modules που επιθυμούμε να φορτώνονται:

```
LoadModule php4_module libexec/libphp4.so
```

#### 2.3.5.4 Server side scripting

Η PHP χρησιμοποιεί Server side scripting , μια τεχνική κατά την οποία ενσωματώνονται scripts σε κώδικα HTML. Το αποτέλεσμα είναι σε ένα αίτημα του χρήστη-πελάτη προς το server του website να διαχειρίζεται από ένα script το οποίο εκτελείται στο server-side πριν ο server απαντήσει στο αίτημα και όχι στον browser όπως γίνεται στην περίπτωση της Javascript.



Το Server side scripting χρησιμοποιείται κυρίως για να παρέχει ένα interface για τον πελάτη και για να περιορίσει την πρόσβαση του πελάτη σε κλειστές βάσεις δεδομένων ή άλλες πηγές δεδομένων. Αυτά τα script μπορούν να συγκεντρώνουν τα χαρακτηριστικά του πελάτη προκειμένου να προσαρμόσουν την απάντηση σε αυτά τα χαρακτηριστικά, τις απαιτήσεις του χρήστη τα δικαιώματα χρήσης κ.ο.κ. Το Server side scripting επίσης δίνει τη δυνατότητα στον ιδιοκτήτη της ιστοσελίδας να μειώσει την πρόσβαση των χρηστών στον πηγαίο κώδικα των server-side scripts που μπορεί να είναι ιδιόκτητα ή πολύτιμα. Το μειονέκτημα αυτής της τεχνικής είναι ότι ο server της ιστοσελίδας πρέπει να παρέχει το μεγαλύτερο μέρος των υπολογιστικών πόρων πριν στείλει την ιστοσελίδα στον πελάτη

#### 2.3.5.5 Επεκτάσεις αρχείων και διακομιστές

Ένα αρχείο με κώδικα PHP θα πρέπει να έχει την κατάλληλη επέκταση (π.χ. \*.php, \*.php4, \*.phtml κ.ά.). Η ενσωμάτωση κώδικα σε ένα αρχείο επέκτασης .html δεν θα λειτουργήσει και θα εμφανίσει στον browser τον κώδικα χωρίς καμία επεξεργασία, εκτός αν έχει γίνει η κατάλληλη ρύθμιση στα MIME types του server. Επίσης ακόμη κι όταν ένα αρχείο έχει την επέκταση .php, θα πρέπει ο server να είναι ρυθμισμένος για να επεξεργάζεται και να μεταγλωττίζει τον κώδικα PHP σε HTML που καταλαβαίνει το πρόγραμμα πελάτη. Ο διακομιστής Apache, που χρησιμοποιείται σήμερα ευρέως σε συστήματα με τα λειτουργικά συστήματα GNU/Linux, Microsoft Windows, Mac OS



Χυποστηρίζει εξ ορισμού την εκτέλεση κώδικα PHP, είτε με την χρήση ενός πρόσθετου (mod\_php) ή με την αποστολή του κώδικα προς εκτέλεση σε εξωτερική διεργασία CGI ή FCGI ή με την έλευση της php5.4 υποστηρίζονται η εκτέλεση σε πολυάσχολους ιστοχώρους, FastCGI Process Manager (FPM).

### **2.3.5.6 Εναλλακτικός τρόπος εκτέλεσης ιστοσελίδων χωρίς χρονοβόρες διαδικασίες**

Ο συνδυασμός Linux/Apache/PHP/MySQL, που είναι η πιο δημοφιλής πλατφόρμα εκτέλεσης ιστοσελίδων είναι γνωστός και με το ακρωνύμιο LAMP. Παρόμοια, ο συνδυασμός \*/Apache/PHP/MySQL ονομάζεται \*AMP, όπου το πρώτο αρχικό αντιστοιχεί στην πλατφόρμα, στην οποία εγκαθίστανται ο Apache, η PHP και η MySQL (π.χ. Windows, Mac OS X).

Ο LAMP συνήθως εγκαθίσταται και ρυθμίζεται στο Linux με τη βοήθεια του διαχειριστή πακέτων της εκάστοτε διανομής. Στην περίπτωση άλλων λειτουργικών συστημάτων, επειδή το κατέβασμα και η ρύθμιση των ξεχωριστών προγραμμάτων μπορεί να είναι πολύπλοκη, υπάρχουν έτοιμα πακέτα προς εγκατάσταση, όπως το XAMPP και το WAMP για τα Windows και το MAMP για το Mac OS X.

### **2.3.6 XML**

Η XML είναι μία γλώσσα σήμανσης, που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων. Ορίζεται, κυρίως, στην προδιαγραφή XML 1.0 (XML 1.0 Specification), που δημιούργησε ο διεθνής οργανισμός προτύπων W3C (World Wide Web Consortium), αλλά και σε διάφορες άλλες σχετικές προδιαγραφές ανοιχτών προτύπων. Η XML σχεδιάστηκε δίνοντας έμφαση στην απλότητα, τη γενικότητα και τη χρησιμότητα στο Διαδίκτυο. Είναι μία μορφοποίηση δεδομένων κειμένου, με ισχυρή υποστήριξη Unicode για όλες τις γλώσσες του κόσμου. Αν και η σχεδίαση της XML εστιάζει στα κείμενα, χρησιμοποιείται ευρέως για την αναπαράσταση αυθαίρετων δομών δεδομένων, που προκύπτουν για παράδειγμα στις υπηρεσίες ιστού.



Υπάρχει μία ποικιλία διεπαφών προγραμματισμού εφαρμογών, που μπορούν να χρησιμοποιούν οι προγραμματιστές, για να προσπελαίνουν δεδομένα XML, αλλά και διάφορα συστήματα σχημάτων XML, τα οποία είναι σχεδιασμένα για να βοηθούν στον ορισμό γλωσσών, που προκύπτουν από την XML. Έως το 2009, έχουν αναπτυχθεί εκατοντάδες γλώσσες που βασίζονται στην XML,[3] συμπεριλαμβανομένων του RSS, του SOAP και της XHTML. Προεπιλεγμένες κωδικοποιήσεις βασισμένες στην XML, υπάρχουν για τις περισσότερες σουίτες εφαρμογών γραφείου, συμπεριλαμβανομένων του Microsoft Office (Office Open XML), του OpenOffice.org (OpenDocument) και του iWork[4] της εταιρίας Apple.

#### **2.3.6.1 Βασική ορολογία**

Το περιεχόμενο αυτής της ενότητας είναι μία εισαγωγή στα βασικά στοιχεία, που συναντώνται στην καθημερινή χρήση της XML.

### **2.3.6.2 Χαρακτήρας Unicode**

Εξ ορισμού, ένα κείμενο XML είναι μία ακολουθία χαρακτήρων. Σχεδόν κάθε χαρακτήρας Unicode μπορεί να εμφανίζεται σε ένα κείμενο XML.

### **2.3.6.3 Επεξεργαστής και Εφαρμογή**

Είναι το λογισμικό που επεξεργάζεται ένα κείμενο XML. Είναι αναμενόμενο, ότι ένας επεξεργαστής δουλεύει για μία εφαρμογή. Υπάρχουν μερικές πολύ συγκεκριμένες απαιτήσεις, σχετικά με το τι μπορεί και τι δεν μπορεί να κάνει ένας επεξεργαστής XML, αλλά καμία, όσον αφορά στη συμπεριφορά της εφαρμογής. Ο επεξεργαστής (όπως ονοματίζεται από την προδιαγραφή), αναφέρεται συχνά, με τον αγγλικό όρο XML parser.

### **2.3.6.4 Σήμανση και Περιεχόμενο**

Οι χαρακτήρες που απαρτίζουν ένα κείμενο XML, αποτελούν είτε τη σήμανση είτε το περιεχόμενό του. Η σήμανση και το περιεχόμενο, μπορούν να επισημανθούν και να διακριθούν, ύστερα από την εφαρμογή κάποιων απλών συντακτικών κανόνων. Όλα τα αλφαριθμητικά που συνιστούν τη σήμανση, είτε ξεκινούν με το χαρακτήρα "<" και καταλήγουν στο χαρακτήρα ">", είτε ξεκινούν με το χαρακτήρα "&" και καταλήγουν στο χαρακτήρα ";". Ακολουθίες χαρακτήρων που δε συνιστούν τη σήμανση, αποτελούν το περιεχόμενο ενός κειμένου XML.

### **2.3.6.5 Ετικέτα**

Ένα στοιχείο σήμανσης που ξεκινά με το χαρακτήρα "<" και καταλήγει στο χαρακτήρα ">". Υπάρχουν τρία είδη ετικέτας: ετικέτες-αρχής, για παράδειγμα <section>, ετικέτες-τέλους, για παράδειγμα </section>, και ετικέτες-χωρίς-περιεχόμενο, για παράδειγμα <line-break/>.

### **2.3.6.6 Στοιχείο**

Ένα λογικό απόσπασμα ενός κειμένου, που είτε ξεκινά με μία ετικέτα-αρχής και καταλήγει σε μία ετικέτα-τέλους, είτε αποτελείται μόνο από μία ετικέτα-χωρίς-περιεχόμενο. Οι χαρακτήρες που υπάρχουν, αν υπάρχουν, μεταξύ μιας ετικέτας-αρχής και μιας ετικέτας-τέλους, συνιστούν το περιεχόμενο του στοιχείου, το οποίο μπορεί να περιέχει σήμανση, συμπεριλαμβανομένων και άλλων στοιχείων, που ονομάζονται στοιχεία-παιδιά. Ένα παράδειγμα ενός στοιχείου είναι το<Greeting>Hello, world.</Greeting>. Ένα άλλο είναι το <line-break/>.

### **2.3.6.7 Χαρακτηριστικό**

Ένα στοιχείο σήμανσης που αποτελείται από ένα ζευγάρι όνομα/τιμή, το οποίο υπάρχει μέσα σε μία ετικέτα-αρχής ή σε μία ετικέτα-χωρίς-περιεχόμενο. Στο παράδειγμα παρακάτω, το στοιχείο img έχει δύο χαρακτηριστικά, τα src και alt:. Ένα άλλο παράδειγμα θα ήταν το <step number="3">Connect A to B.</step>, όπου το όνομα του χαρακτηριστικού είναι "number" και η τιμή του είναι "3".

### **2.3.6.8 Δήλωση XML**

Τα κείμενα XML μπορούν να αρχίζουν, με τη δήλωση κάποιων πληροφοριών σχετικών με αυτά, όπως στο ακόλουθο παράδειγμα:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Το παρακάτω είναι ένα μικρό, αλλά πλήρες κείμενο XML, που κάνει χρήση όλων των παραπάνω εννοιών και στοιχείων.

```
<?xml version="1.0" encoding='UTF-8'?>

<painting>

  <caption>This is Raphael's "Foligno" Madonna, painted in

  <date>1511</date>-<date>1512</date>.</caption>

</painting>
```

Υπάρχουν πέντε στοιχεία σε αυτό το κείμενο του παραδείγματος: τα painting, img, caption, και δύο date. Τα στοιχεία date, είναι παιδιά του στοιχείου caption, το οποίο είναι παιδί του στοιχείου-ρίζας painting. Το στοιχείο img έχει δύο χαρακτηριστικά, τα src και alt.

### 2.3.6.9 Χαρακτήρες και διαφυγή

Τα κείμενα XML αποτελούνται εξ ολοκλήρου από χαρακτήρες Unicode. Εκτός από ένα μικρό αριθμό, ειδικά εξαιρούμενων χαρακτήρων ελέγχου, κάθε χαρακτήρας που ορίζεται στο Unicode, μπορεί να εμφανίζεται στο περιεχόμενο ενός κειμένου XML. Το σύνολο των χαρακτήρων που μπορούν να εμφανίζονται στη σήμανση, αν και κάπως περιορισμένο, παραμένει μεγάλο. Η XML παρέχει κάποιες διευκολύνσεις για την ταυτοποίηση της κωδικοποίησης των χαρακτήρων Unicode που απαρτίζουν ένα κείμενο και για την απεικόνιση χαρακτήρων που, για τον έναν ή τον άλλο λόγο, δεν μπορούν να χρησιμοποιηθούν ευθέως.

### 2.3.7 Βάσεις Δεδομένων XML

Μια βάση δεδομένων XML είναι λογισμικό βάσεων δεδομένων που διατηρεί την πληροφορία (data persistence software) και που επιτρέπει στα δεδομένα να αποθηκεύονται σε μορφή XML. Αυτά τα δεδομένα μπορούν να εξερευνηθούν, να εξαχθούν και να καταναμηθούν με την επιθυμητή μορφοποίηση. Οι βάσεις δεδομένων XML συνήθως σχετίζονται με βάσεις δεδομένων εγγράφων.

Υπάρχουν δύο κύριες κατηγορίες βάσεων δεδομένων XML

1. XML-enabled: αυτού του τύπου οι βάσεις είτε χαρτογραφούν σε πιο παραδοσιακές δομές βάσεων (όπως πχ μια σχεσιακή βάση δεδομένων) που δέχονται την XML ως είσοδο και αποδίδουν δεδομένα σε XML, ή όπως πιο πρόσφατα, υποστηρίζουν τοπικά XML τύπους εντός της παραδοσιακής βάσης δεδομένων. Αυτό σημαίνει ότι η βάση επεξεργάζεται την ίδια την XML αντί να βασίζεται σε ενδιάμεσο λογισμικό.
2. Native XML (NXD): το εσωτερικό μοντέλο τέτοιων βάσεων δεδομένων εξαρτάται από την XML και χρησιμοποιεί έγγραφα XML ως την βασική μονάδα αποθήκευσης, τα οποία δεν αποθηκεύονται απαραίτητα με την μορφή αρχείων text.

### 2.3.7.1 Rationale for XML in databases

Ένας λόγος για να χρησιμοποιήσει κάποιος XML σε βάσεις δεδομένων είναι η αυξανόμενη χρήση της XML για τη μεταφορά αρχείων, το οποίο σημαίνει ότι δεδομένα εξάγονται από βάσεις δεδομένων και εισάγονται σε αρχεία XML και το ανάποδο. Μπορεί να αποδειχτεί πιο αποδοτικό και πιο εύκολη η αποθήκευση δεδομένων σε XML μορφή από άποψη κόστους μετατροπής. Οι δυνατότητες των native XML βάσεων δεδομένων επίσης ελαχιστοποιούν την ανάγκη της εξαγωγής ή εισαγωγής metadata για την υποστήριξη της αναζήτησης και της πλοήγησης για εφαρμογές που βασίζονται στο περιεχόμενο (content-based applications). Σε native XML περιβάλλον, όλο το αποθηκευμένο περιεχόμενο μετατρέπεται σε metadata μέσω γλωσσών επερωτήσεων (query languages) όπως XPath and XQuery, συμπεριλαμβανόμενων του περιεχομένου, των χαρακτηριστικών και των συσχετίσεων εντός της XML. Καθώς αυτό το επίπεδο δυνατότητας αναζήτησης είναι δυνατό σε εξωτερικά metadata, απαιτεί πιο περίπλοκη και δύσκολη επεξεργασία προκειμένου να παράγει το δέντρο περιεχομένων σε metadata.

### 2.3.7.2 XML Enabled databases

Οι βάσεις δεδομένων, XML enabled, προσφέρουν μία ή περισσότερες από τις παρακάτω προσεγγίσεις στην αποθήκευση XML εντός μια παραδοσιακής σχεσιακή δομής:

1. Η XML αποθηκεύεται σε ένα CLOB (Character large object)
2. Η XML τεμαχίζεται σε μια σειρά πινάκων που βασίζονται σε ένα σχήμα βάσης
3. Η XML αποθηκεύεται σε μια μορφή native XML όπως αυτή είναι ορισμένη από το ISO

Τα συστήματα διαχείρισης βάσεων δεδομένων που υποστηρίζουν τον ISO τύπο XML είναι :

- IBM DB2 (pureXML)
- Microsoft SQL Server
- Oracle Database
- PostgreSQL

Κατά βάση, μια βάση δεδομένων που είναι XML enabled ταιριάζει καλύτερα σε περιπτώσεις όπου η πλειοψηφία των δεδομένων δεν είναι σε μορφή XML. Στις περιπτώσεις όπου η πλειοψηφία των δεδομένων είναι σε μορφή XML, προτείνεται μια Native XML βάση δεδομένων.

### 2.3.7.3 Παράδειγμα ερωτήματος XML σε IBM DB2 SQL

```
SELECT
  id, vol, xmlquery('$j/name', passing journal AS "j") AS name
FROM
  journals
WHERE
  xmlexists('$j[license="CreativeCommons"]', passing journal AS "j")
```

#### 2.3.7.4 Βάσεις δεδομένων τύπου Native XML (NXD)

Ο όρος native XML βάση δεδομένων (NXD) μπορεί να οδηγήσει σε σύγχυση καθώς πολλές NXD δεν λειτουργούν σαν αυτόνομες βάσεις δεδομένων και δεν αποθηκεύουν την αρχική μορφή της XML

Η επίσημη περιγραφή των βάσεων δεδομένων XML όπως αυτός δίνεται από την πρωτοβουλία XML:DB :

- Ορίζει ένα λογικό μοντέλο για ένα αρχείο XML και όχι για τα δεδομένα που υπάρχουν στο αρχείο, και αποθηκεύει και ανακτά τα αρχεία βάση αυτού του μοντέλου. Κατ' ελάχιστον το μοντέλο πρέπει να περιλαμβάνει στοιχεία, χαρακτηριστικά, PCDATA, και την θέση του αρχείου. Παραδείγματα τέτοιων μοντέλων περιλαμβάνουν το XPath του μοντέλου, το σετ πληροφοριών της XML και τα μοντέλα που συνεπάγονται από το DOM και τα γεγονότα στο SAX 1.0.
- Έχει ένα αρχείο XML ως την θεμελιώδη μονάδα αποθήκευσης, όπως μια σχεσιακή βάση δεδομένων έχει μια σειρά σε ένα πίνακα ως την θεμελιώδη μονάδα αποθήκευσης
- Δεν έχει την ανάγκη κάποιου συγκεκριμένου μοντέλου φυσικής αποθήκευσης. Για παράδειγμα οι NXD μπορούν να χρησιμοποιούν συσχετιστικές, ιεραρχικές ή αντικειμενοστραφής δομές βάσεων δεδομένων ή να χρησιμοποιούν διαφορετικούς τύπους αποθήκευσης (όπως ευρετήρια ή συμπιεσμένα αρχεία)

Επιπλέον, πολλές βάσεις δεδομένων XML παρέχουν ένα λογικό μοντέλο για την ομαδοποίηση των εγγράφων το οποίο ονομάζεται «συλλογές». Οι βάσεις δεδομένων μπορούν να εγκαταστήσουν και να διαχειριστούν πολλές συλλογές τη φορά. Σε κάποιες εφαρμογές μπορεί να υπάρχει ιεραρχία στις συλλογές με τον ίδιο τρόπο που λειτουργεί μια δομή ευρετηρίου σε ένα λειτουργικό σύστημα.

Όλες οι βάσεις δεδομένων XML υποστηρίζουν τουλάχιστον έναν τύπο σύνταξης ερωτημάτων. Σχεδόν όλες υποστηρίζουν το XPath για την πραγματοποίηση ερωτημάτων για έγγραφα ή συλλογές ή συλλογές εγγράφων. Το XPath παρέχει ένα απλό σύστημα διαδρομών (pathing system) που επιτρέπει στο χρήστη να αναγνωρίζει κόμβους που συσχετίζονται με συγκεκριμένα σετ κριτηρίων.

Εκτός από το XPath, πολλές βάσεις δεδομένων XML υποστηρίζουν τα XSLT ως μέθοδο μετατροπής εγγράφων ή των αποτελεσμάτων μιας επερώτησης που ανακτήθηκε από τη βάση. Τα XSLT παρέχουν μια γλώσσα δηλώσεων που χρησιμοποιεί τη σύνταξη της XML. Σκοπός είναι να οριστεί ένα σετ φίλτρων XPath που μπορούν να μετατρέψουν τα έγγραφα εξολοκλήρου ή μερικώς σε άλλες μορφές, συμπεριλαμβανομένου του απλού κειμένου (text), XML, η HTML

Πολλές βάσεις δεδομένων επίσης υποστηρίζουν το XQuery για να πραγματοποιήσουν επερωτήσεις. Το XQuery συμπεριλαμβάνει το XPath σαν μέθοδο επιλογής κόμβων αλλά ή επέκταση του XPath μπορεί να παρέχει και δυνατότητες μετατροπής. Αυτή η σύνταξη πολλές φορές αναφέρεται ως FLWOR, γιατί η επερώτηση μπορεί να συμπεριλαμβάνει τις παρακάτω εντολές 'for', 'let', 'where', 'order by' και 'return'. Οι κατασκευαστές συστημάτων διαχείρισης βάσεων δεδομένων (που παραδοσιακά είχαν αποκλειστικά μηχανές SQL) πλέον κατασκευάζουν υβριδικές μηχανές SQL και XQuery. Αυτές οι μηχανές επιτρέπουν την αναζήτηση XML δεδομένων μαζί με συσχετιστικά δεδομένα στην ίδια έκφραση.

Οι περισσότερες βάσεις δεδομένων XML υποστηρίζουν ένα API μη περιορισμένο από συγκεκριμένο κατασκευαστή που ονομάζεται XQuery API για Java (XQJ). Το XQJ API δημιουργήθηκε στο JCP ως μια κοινή διεπαφή για μια πηγή δεδομένων XML/XQuery επιτρέποντας στους προγραμματιστές Java να υποβάλλουν ερωτήσεις που συμμορφώνονται με την προδιαγραφή XQuery 1.0 του World Wide Web Consortium (W3Cs) και να επεξεργάζονται τα αποτελέσματα αυτών των ερωτήσεων. Γενικά το XQJ API επιτελεί για τις βάσεις δεδομένων XML και XQuery την ίδια λειτουργία που επιτελεί το JDBC API για τις σχεσιακές βάσεις δεδομένων και SQL.

## 3 Εργαλεία ανάπτυξης

### 3.1 Γενικά

Τα εργαλεία ανάπτυξης (development tools) μίας εφαρμογής είναι όλα τα προγράμματα που πρέπει να εγκατασταθούν σε ένα σύστημα ώστε να είναι δυνατή η επεξεργασία του κώδικα της εφαρμογής και η εκτέλεσή της.

Ιδιαίτερη σημασία έχουν και οι ρυθμίσεις που πρέπει να γίνουν ώστε να μπορεί να τρέξει μία εφαρμογή, καθώς πολλές φορές τα προγράμματα που εγκαθίστανται δεν μπορούν να εκτελεστούν ή να συνεργαστούν μεταξύ τους χωρίς αυτές.

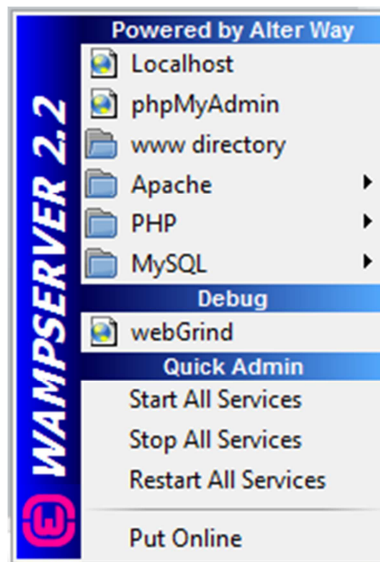
Σε αυτή την ενότητα θα καλύψουμε τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής μας.

### 3.2 WAMPServer

#### 3.2.1 Σχετικά με το WAMPServer

Για τη λειτουργία της εφαρμογής μας είναι απαραίτητο λόγω του τρόπου που λειτουργεί το Google maps η σελίδα να βρίσκεται σε έναν web server.

Η εγκατάσταση ενός web server και οι απαραίτητες ρυθμίσεις είναι αρκετά χρονοβόρες, πολύπλοκες και η ανάλυσή τους έξω από τους στόχους αυτής της εργασίας. Για αυτό τον λόγο χρησιμοποιήσαμε το πρόγραμμα WAMPServer. Αυτό είναι μια ομάδα προγραμμάτων που περιέχει μία εγκατάσταση του Apache web server, μαζί με μία βάση δεδομένων MySQL και τη γλώσσα PHP ρυθμισμένη ώστε να συνεργάζεται με τον Apache.



Το όνομα WAMP προκύπτει σε αντιστοιχία με το ακρωνύμιο LAMP, από τα αρχικά των Linux, Apache (HTTP Server), MySQL και PHP. Καθώς το πρόγραμμα τρέχει στα Windows το L του Linux έχει αντικατασταθεί από το W των Windows.

Αν και δεν χρησιμοποιούμε τη βάση δεδομένων χρησιμοποιούμε το WAMP ώστε να έχουμε έτοιμη μία λειτουργική εγκατάσταση του Apache και αγνοούμε τα υπόλοιπα στοιχεία.

### 3.2.2 Εγκατάσταση

Το WAMPServer είναι διαθέσιμο δωρεάν και μπορούμε να το κατεβάσουμε από τη διεύθυνση:

<http://sourceforge.net/projects/wampserver/files/WampServer%202/WampServer%202.2/wampserver2.2e/wampserver2.2e-php5.3.13-httpd2.2.22-mysql5.5.24-32b.exe/download>

Η διαδικασία της εγκατάστασης ζητάει μόνο τον φάκελο εγκατάστασης, όπου μπορούμε να αφήσουμε τον προεπιλεγμένο C:\wamp.

Κατόπιν, για τα δούμε τη σελίδα μας χρειάζεται να τρέχουμε κάθε φορά το WAMP ώστε να είναι διαθέσιμος ο HTTP server.

## 3.3 BaseX

### 3.3.1 Σχετικά με την BaseX

Η BaseX είναι η βάση δεδομένων XML που χρησιμοποιήσαμε για την ανάπτυξη της εφαρμογής. Αν και τα δεδομένα μας μπορούν να αποθηκευτούν σε οποιαδήποτε βάση XML προτιμήσαμε την BaseX γιατί προσφέρει όλες τις δυνατότητες που χρειαζόμαστε και είναι αρκετά ελαφριά και γρήγορη. Κατά συνέπεια προσφέρει άμεση εκτέλεση των ερωτημάτων που υποβάλλουμε χωρίς να έχει μεγάλες απαιτήσεις και να επιβαρύνει το μηχάνημα.

Η BaseX είναι ένα πρόγραμμα ανοικτού κώδικα που συμμορφώνεται με τους όρους του BSD License. Αναπτύσσεται από την εταιρεία BaseX GmbH που δίνει το βασικό πρόγραμμα δωρεάν και προσφέρει υποστήριξη επί πληρωμή σε πελάτες που το επιθυμούν. Περιέχει μία πλήρη βάση δεδομένων XML και έναν επεξεργαστή ερωτημάτων XPath / XQuery 3.0. Έχει δοθεί μεγάλη βάση στις επιδόσεις, στην οικονομία πόρων και στην ευχρηστία της, καθιστώντας της ανταγωνιστική σε σχέση με οποιαδήποτε άλλη βάση XML.

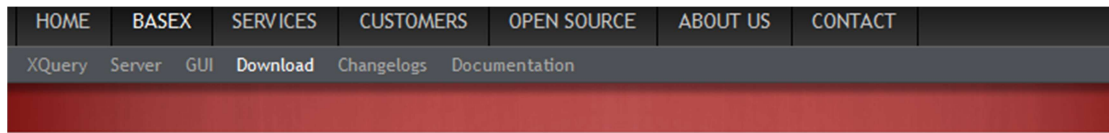
Κατά την ανάπτυξη της εφαρμογής χρησιμοποιήσαμε την έκδοση 7.8.1.

### 3.3.2 Εγκατάσταση της BaseX

Για την εγκατάσταση της BaseX αρχικά πρέπει να την κατεβάσουμε από τη διεύθυνση:

<http://basex.org/products/download/all-downloads/>





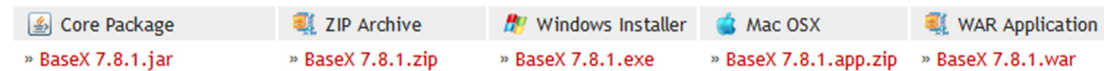
## BaseX Download

BaseX is very light-weight, easy to use, and runs out of the box!

It is based on Java 6 and has been tested and optimized for various platforms. The GUI and the Server are bundled in a single file, including all APIs, start scripts and sample documents.



### Official Release: 7.8.1 (2014/02/26)



### Linux Distributions

- [Debian GNU \(unstable/testing\)](#)
- [Ubuntu](#)
- [Archlinux \(AUR\)](#)
- [openSUSE: 11.4, 12.1, 12.2, 12.3, 13.1, Tumbleweed, Factory](#)
- [Fedora: 18 \(Spherical Cow\), 19 \(Schrödinger's Cat\), 20 \(Heisenbug\)](#)
- [Red Hat Enterprise Linux 6, CentOS 6](#)

### More Links

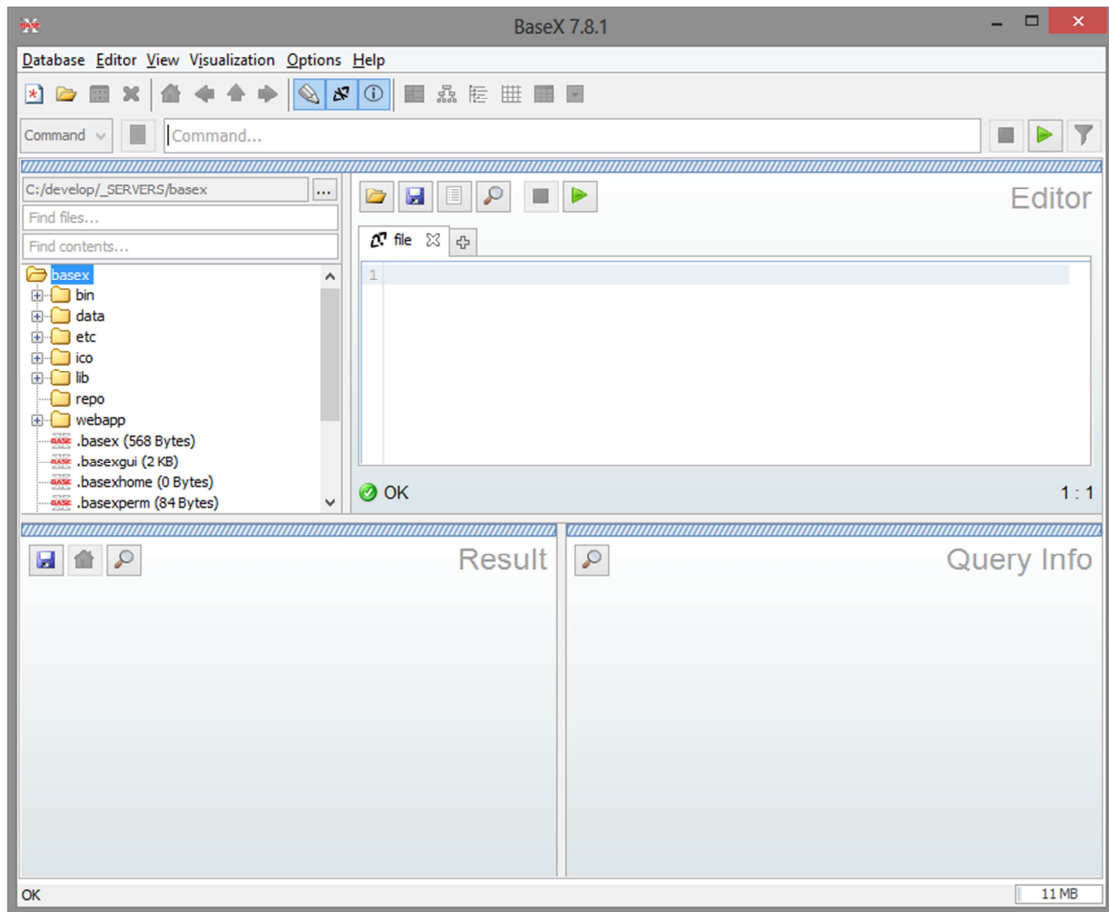
- [BaseX XQJ API](#), written by Charles Foster
- [Latest stable snapshot](#) (recommended for developers)
- [All former releases](#)

Για εγκατάσταση σε σύστημα Windows χρειάζεται η επιλογή Windows Installer και υπάρχουν επιπλέον και δυνατότητες για εγκατάσταση σε άλλες πλατφόρμες.

### 3.3.3 Λειτουργίες και εκτέλεση

Για τα ξεκινήσει να τρέχει η βάση χρειάζεται να εκτελέσουμε τη συντόμευση BaseX Server (Start) από τον αντίστοιχο φάκελο στο μενού Έναρξης των Windows.

Για την διαχείριση της βάσης υπάρχει το πρόγραμμα BaseX GUI το οποίο είναι ένα εργαλείο για την διαχείριση της βάσης.



Μέσω αυτού μπορούμε να δημιουργήσουμε εύκολα νέες βάσεις και να δούμε και να τροποποιήσουμε τα περιεχόμενα των υπάρχοντων.

### 3.4 Εργαλεία συγγραφής κώδικα

Σε αντίθεση με τις παραδοσιακές γλώσσες προγραμματισμού, οι εφαρμογές για web, τουλάχιστον με τις τεχνολογίες που χρησιμοποιούμε, δεν χρειάζονται compile, αλλά χρησιμοποιούν κατευθείαν τα αρχεία του πηγαίου κώδικα για να τρέξουν. Κατά συνέπεια δεν χρειάζεται ένα πολύπλοκο περιβάλλον ανάπτυξης αλλά αρκεί ένας απλός διορθωτής κειμένου.

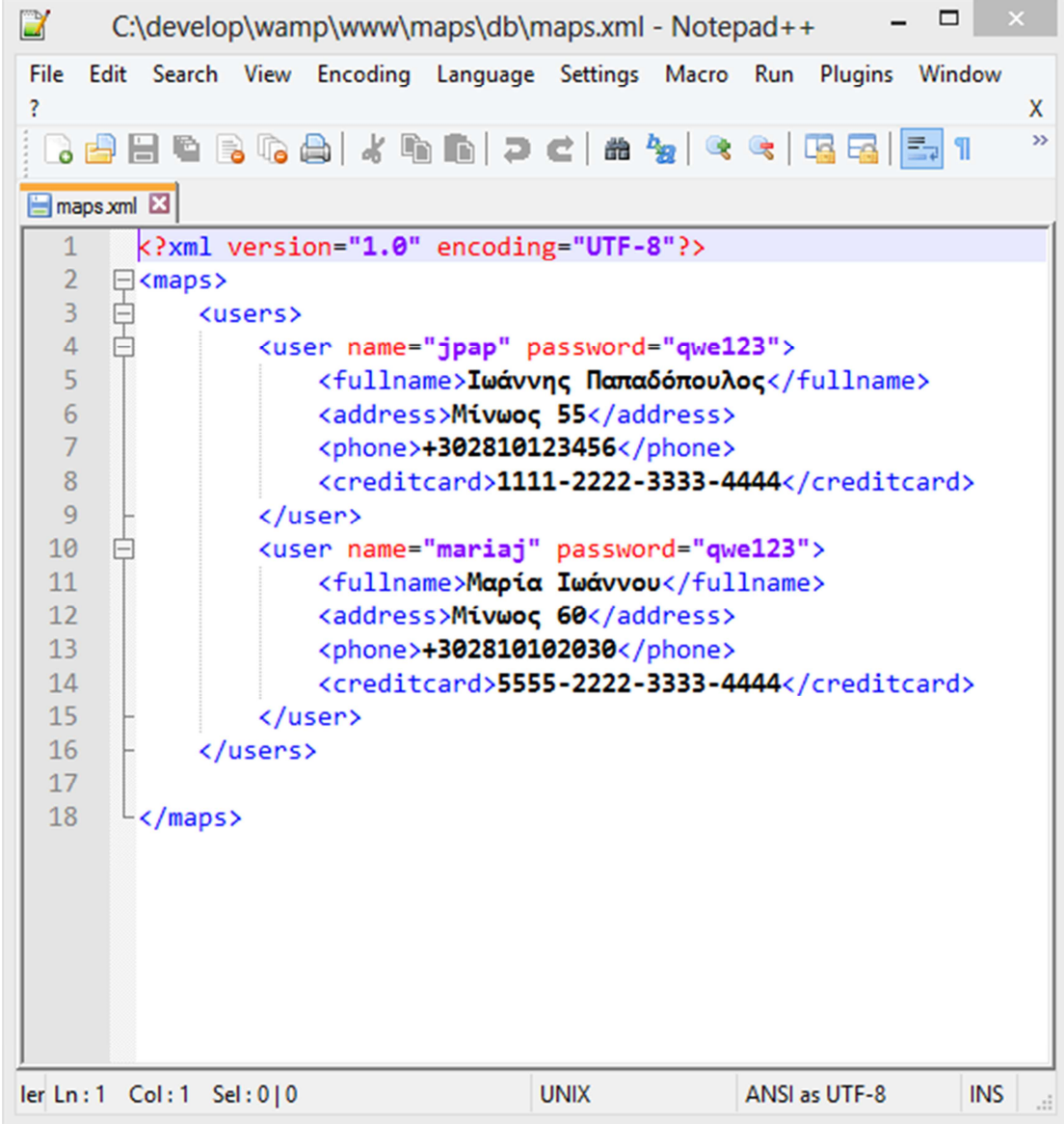
Αυτό δε σημαίνει ότι ενδείκνυται η χρήση ενός απλοϊκού διορθωτή, όπως το Notepad των Windows. Ακόμη και σε αυτήν την κατηγορία υπάρχουν αρκετά προγράμματα που παρέχουν βοήθεια στον προγραμματιστή και μπορούν να διευκολύνουν την εργασία του.

Στην περίπτωση της εφαρμογής μας χρησιμοποιήθηκε το πρόγραμμα Notepad++.

#### 3.4.1 Notepad++

Το Notepad++ είναι ένας διορθωτής κειμένου και πηγαίου κώδικα για το λειτουργικό σύστημα Windows. Διαφέρει από το βασικό Notepad στο ότι παρέχει αρκετές επιπλέον δυνατότητες, όπως η ταυτόχρονη επεξεργασία πολλών αρχείων σε διαφορετικές καρτέλες και στο ότι μπορεί να ανοίξει αρκετά μεγαλύτερα αρχεία. Το όνομα του προέρχεται σε αντιστοιχία με το διπλό συν που υπάρχει στη γλώσσα C++.

Το πρόγραμμα διανέμεται σαν ελεύθερο λογισμικό. Βρίσκεται στη σελίδα sourceforge.net, από την οποία έχει ήδη κατεβαστεί πάνω από 30 εκατομμύρια φορές και έχει κερδίσει πολλά βραβεία. Βασίζεται στην βιβλιοθήκη συγγραφής κώδικα Scintilla.



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <maps>
3   <users>
4     <user name="jrap" password="qwe123">
5       <fullname>Ιωάννης Παπαδόπουλος</fullname>
6       <address>Μίνως 55</address>
7       <phone>+302810123456</phone>
8       <creditcard>1111-2222-3333-4444</creditcard>
9     </user>
10    <user name="mariaj" password="qwe123">
11      <fullname>Μαρία Ιωάννου</fullname>
12      <address>Μίνως 60</address>
13      <phone>+302810102030</phone>
14      <creditcard>5555-2222-3333-4444</creditcard>
15    </user>
16  </users>
17 </maps>
```

Αυτή τη στιγμή το πρόγραμμα βρίσκεται στη διεύθυνση:

<http://notepad-plus-plus.org/download/>

## 4 Ανάλυση εφαρμογής

### 4.1 Γενικά

Σε αυτή την ενότητα θα αναλύσουμε τον κώδικα της εφαρμογής και τυχόν ρυθμίσεις που κάναμε για την λειτουργία της.

### 4.2 Τμήματα εφαρμογής

Η εφαρμογή μπορεί να χωριστεί σε διαφορετικά τμήματα, ανάλογα με τις τεχνολογίες που χρησιμοποιούνται. Εδώ θα επιδιώξουμε να εξηγήσουμε επιγραμματικά τι κάνει το καθένα πριν ασχοληθούμε με τα σημαντικά κομμάτια τους.

#### 4.2.1 Αρχεία XML

Αυτά είναι τα αρχεία δεδομένων που αποτελούν τα περιεχόμενα της βάσης. Για την χρήση τους πρέπει να δημιουργήσουμε μία νέα βάση δεδομένων που να χρησιμοποιεί αυτά τα αρχεία και από εκεί και πέρα μπορούμε να κάνουμε ερωτήσεις για τις λειτουργίες που μας ενδιαφέρουν.

#### 4.2.2 Αρχεία PHP

Οι περισσότερες σελίδες της εφαρμογής είναι αρχεία με κατάληξη PHP. Η γλώσσα δημιουργείται για να ελέγχει το τρέχον session και για να δημιουργήσει κάποια τμήματα των σελίδων που δεν ήταν πρακτικό να γίνουν μέσω JavaScript.

Αρκετές από αυτές τις σελίδες επικοινωνούν με την BaseX για να ζητήσουν δεδομένα, βάση των οποίων δημιουργούν τις επόμενες σελίδες.

#### 4.2.3 Κώδικας JavaScript

Η εφαρμογή περιέχει μεγάλα τμήματα κώδικα JavaScript. Αυτός χρειάζεται ώστε η σελίδα να ανταποκρίνεται στις ενέργειες του χρήστη, όποτε αυτό είναι επιθυμητό, χωρίς να χρειάζεται να φορτωθεί από την αρχή.

Ο βασικός κώδικας JavaScript βρίσκεται στο αρχείο βιβλιοθήκης map.js το οποίο περιέχει αρκετές συναρτήσεις που χρησιμοποιούνται σε όλες τις σελίδες. Επιπλέον υπάρχουν τα αρχεία του jQuery που είναι το βασικό framework που χρησιμοποιούμε για τις βιβλιοθήκες και κομμάτια κώδικα μέσα σε κάθε σελίδα που έχουμε, για της επιπλέον παραμετροποίηση.

Επιπλέον χρησιμοποιείται και το Google Maps, το οποίο είναι μία κατ' εξοχήν εφαρμογή JavaScript.

#### 4.2.4 Κώδικας CSS

Για τη μορφοποίηση της εφαρμογής χρησιμοποιούμε ένα κοινό αρχείο CSS με τους κανόνες που είναι απαραίτητοι για όλες τις επιπλέον σελίδες. Αυτό είναι το style.css.

#### 4.2.5 Κώδικας HTML

Αν και δεν υπάρχει κανένα αρχείο HTML οι σελίδες που στέλνονται στον χρήστη περιέχουν προφανώς κώδικα HTML, όπως αυτός παράγεται από τα αρχεία PHP.

## 4.3 Δεδομένα XML

Τα δεδομένα της βάσης αποθηκεύονται σαν αρχεία XML στον φάκελο db. Αν και δεν υπάρχει λόγος να βρίσκονται μαζί με τα υπόλοιπα αρχεία της εφαρμογής, τα έχουμε τοποθετήσει σε αυτόν τον φάκελο για ευκολία στη διαχείρισή τους.

Παρακάτω θα παρουσιάσουμε τη δομή κάθε αρχείου.

### 4.3.1 Αρχείο χρηστών

Τα δεδομένα των χρηστών αποθηκεύονται στο αρχείο users.xml. Αυτό περιέχει ένα βασικό στοιχείο users, και μέσα σε αυτό κάθε χρήστης είναι ένα στοιχείο user.

Αρχικά υπάρχει η κατάλληλη οδηγία για την κωδικοποίηση του αρχείου σε UTF-8.

```
<?xml version="1.0" encoding="UTF-8"?>
<users>
```

Έπειτα υπάρχουν τα δεδομένα των χρηστών.

```
<user name="jrap" password="qwe123">
  <fullname>Ιωάννης Παπαδόπουλος</fullname>
  <address>Μίνως 55, Ηράκλειο, Ελλάδα</address>
  <phone>+302810123456</phone>
  <creditcard>1111-2222-3333-4444</creditcard>
</user>
```

Όπως φαίνεται το όνομα του χρήστη και ο κωδικός του είναι ιδιότητες στο στοιχείο user. Έπειτα αυτό περιέχει το πλήρες όνομα, τη διεύθυνση, το τηλέφωνο και τον αριθμό της πιστωτικής του κάρτας.

Ειδικά για τη διεύθυνση πρέπει να δοθεί προσοχή στο να έχει τέτοια μορφή ώστε να την αναγνωρίζει το Google Maps, δηλαδή να την εμφανίζει σωστά στον χάρτη.

### 4.3.2 Αρχείο βενζινάδικων

Το αρχείο με τα αποθηκευμένα βενζινάδικα είναι το gasstations.xml. Αυτό έχει σαν βασικό στοιχείο το gasstations και μέσα σε αυτό κάθε βενζινάδικο είναι ένα στοιχείο τύπου gasstation.

Η μορφή του καθενός φαίνεται παρακάτω.

```
<gasstation>
  <name>ΕΚΟ - Αφοί Σαμολή</name>
  <address>Μουστακλή Σπύρου 26, Ηράκλειο</address>
  <phone>+302810252525</phone>
  <hours>
```

```
<opening>07:00</opening>
<closing>20:00</closing>
</hours>
</gasstation>
```

Όπως και με τους χρήστες τα βασικά στοιχεία είναι αντικείμενα το όνομα, τη διεύθυνση και το τηλέφωνο. Επιπλέον για το ωράριο υπάρχει ένα στοιχείο hours.

Αυτό περιέχει σαν παιδιά του δύο στοιχεία opening και closing, τα οποία είναι η ώρα που ανοίγει και η ώρα που κλείνει κάθε βενζινάδικο. Καθώς μας ενδιαφέρει να κρατήσουμε μόνο ώρες και όχι ημερομηνίες, η μορφή είναι 09:45, με τις ώρες και τα λεπτά να είναι πάντα διψήφιος με την προσθήκη ενός μηδέν στην αρχή όταν χρειάζεται.

### 4.3.3 Αρχείο super market

Το αρχείο που αποθηκεύει τα super market είναι το supermarkets.xml. Η μορφή του είναι παρόμοια με τα βενζινάδικα, με το βασικό στοιχείο να είναι το supermarkets και κάθε εγγραφή να είναι ένα αντικείμενο τύπου supermarket.

```
<supermarket>
  <name>ΕΛΙΤ Ε.Π.Ε.</name>
  <address>Λεωφόρος Κνωσού 250, Ηράκλειο, Ελλάδα</address>
  <phone>+302810262021</phone>
  <hours>
    <opening>08:30</opening>
    <closing>21:00</closing>
  </hours>
</supermarket>
```

Και εδώ υπάρχουν τα στοιχεία του καταστήματος στα πεδία name, address, phone, ενώ το ωράριο αποθηκεύεται στο πεδίο hours και συγκεκριμένα στα πεδία του opening και closing.

### 4.3.4 Αρχείο ταινιών

Οι ταινίες βρίσκονται αποθηκευμένες στο αρχείο movies.xml. Η μορφή τους είναι πιο πολύπλοκη από τα προηγούμενα δεδομένα.

Συγκεκριμένα το βασικό στοιχείο είναι το movies, με κάθε ταινία να είναι ένα στοιχείο movie.

```
<movies>
  <movie>
    <title>Captain America 2</title>
```

Αυτό περιέχει τον τίτλο της ταινίας, όπως επιπλέον περιέχει ένα στοιχείο `cinemas`. Αυτό περιέχει όλους τους κινηματογράφους στους οποίους παίζεται η ταινία, με κάθε έναν από αυτούς να είναι ένα αντικείμενο τύπου `cinema`.

```
<cinema>
  <name>Τεχνόπολις</name>
  <address>Τεχνόπολις, Λεωφόρος Ανδρέα Παπανδρέου, Ηράκλειο,
Ελλάδα</address>
  <hours>
    <opening>20:30</opening>
    <closing>22:00</closing>
  </hours>
</cinema>
```

Κάθε ένα από αυτά έχει τη μορφή που έχουμε γενικά για τα καταστήματα, δηλαδή έχει όνομα και διεύθυνση στα πεδία `name` και `address`.

Οι ώρες προβολής αποθηκεύονται σαν τα πεδία `opening` και `closing`, τα οποία είναι παιδιά του πεδίου `hours`.

#### 4.4 Παρουσίαση κώδικα

Σε αυτή την ενότητα θα παρουσιάσουμε τον κώδικα της εφαρμογής. Ο πλήρης κώδικας συμπεριλαμβάνεται στο παράρτημα στο τέλος της εργασίας, οπότε η εξαντλητική παρουσίασή του δεν εξυπηρετεί κάποιο σκοπό.

Αντίθετα θα δείξουμε τα ποιο σημαντικά τμήματα του κώδικα, με ποιο τρόπο λειτουργούν και όπου αυτό είναι απαραίτητο για πιο λόγο υλοποιήθηκαν με αυτόν τον τρόπο και τι προβλήματα λύνουν.

Επιπλέον θα αναφέρουμε διάφορα ζωτικά αρχεία απαραίτητα για την λειτουργία της εφαρμογής.

##### 4.4.1 Αρχείο μορφοποίησης `style.css`

Το βασικό αρχείο που χρησιμοποιείται για να καθορίσουμε την εμφάνιση της εφαρμογής είναι το `style.css`. Όλες οι σελίδες το συμπεριλαμβάνουν με την εντολή:

```
<link href="style.css" rel="stylesheet" type="text/css">
```

Αυτό περιέχει έναν μεγάλο αριθμό από κανόνες, όπως ο παρακάτω:

```
header {
  padding:20px 20px 10px 20px;
  margin-bottom:20px;
  color: #000; text-shadow: 0px 0px 3px #fff;
  background: url('images/logo.png') no-repeat 20px 20px #e74c3c;
  box-shadow: 10px 10px 5px #888888;
  border-radius: 5px;
```

```
}  
  
header h1 {  
  float: left;  
  width: 500px;  
  margin-left: 60px;  
  line-height: 32px;  
}
```

Αυτός ο κανόνας καθορίζει την εμφάνιση της επικεφαλίδας της σελίδας, δηλαδή της μπάρας που βρίσκεται στο επάνω μέρος.

Η αναλυτική παρουσίαση των κανόνων CSS δεν έχει κάποιο ιδιαίτερο νόημα καθώς ξεφεύγει πολύ από τον στόχο της εργασίας και έχει πολύ μεγάλη έκταση.

#### 4.4.2 Βιβλιοθήκη JavaScript map.js

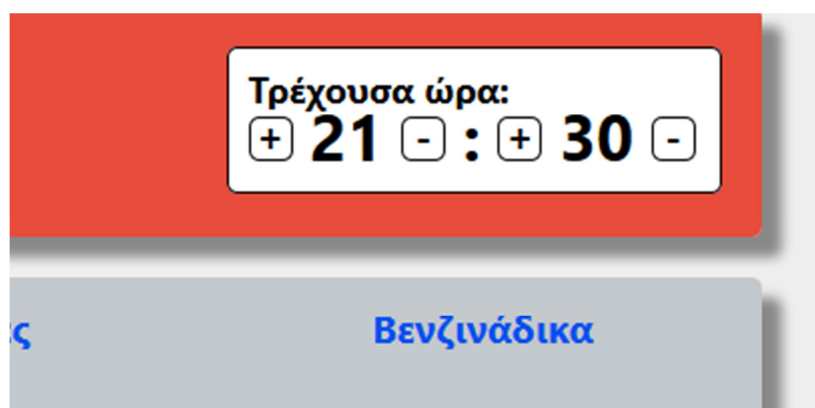
Το αρχείο map.js περιέχει έναν αριθμό από μεταβλητές και μεθόδους JavaScript που είναι απαραίτητες για την λειτουργία διαφόρων τμημάτων της εφαρμογής.

Για την κατανόηση της λειτουργίας τους είναι καλύτερο να χωρίσουμε το αρχείο σε τμήματα ανάλογα με την λειτουργία που εξυπηρετούν τα περιεχόμενα του καθενός.

##### 4.4.2.1 Λειτουργία ρολογιού

Η εφαρμογή μας βασίζεται στον προγραμματισμό πολλών λειτουργιών ανάλογα με την τρέχουσα ώρα, πχ. το αν θα προτείνει κάποιο βενζινάδικο εξαρτάται από το τι ώρα είναι όταν γίνεται η ερώτηση και το τι ώρα αυτό κλείνει.

Καθώς δεν είναι δυνατόν ούτε για την ανάπτυξη ούτε για την επίδειξη της εφαρμογής να υπάρχουν βενζινάδικα που οι ώρες λειτουργίας τους να σχετίζονται με την τρέχουσα ώρα, προσθέσαμε ένα εικονικό ρολόι στο οποίο μπορούμε να ρυθμίζουμε την τρέχουσα ώρα. Αυτό βρίσκεται στην επάνω μπάρα κάθε σελίδας.



Ο κώδικας HTML που δημιουργεί αυτό το ρολόι είναι ο παρακάτω:



```
<div id="timer">
  <h6>Τρέχουσα ώρα:</h6>
  <a href="#" class="button hour plus">+</a>
  <span class="hour">00</span>
  <a href="#" class="button hour minus">-</a>
  :
  <a href="#" class="button minute plus">+</a>
  <span class="minute">00</span>
  <a href="#" class="button minute minus">-</a>
</div>
```

Σε αυτόν χρησιμοποιούμε τα links σαν κουμπιά και τα span σαν οθόνη για την εμφάνιση των τιμών.

Αρχικά υπάρχουν οι global μεταβλητές για την ώρα και τα λεπτά:

```
var hour = 0;
var minute = 0;

var minutesIncrement = 10;
```

Το minutesIncrement αναπαριστά το πόσα λεπτά θα προστίθενται ή θα αφαιρούνται όταν πατάμε το αντίστοιχο κουμπί.

Η μέθοδος initClock χρειάζεται να καλείται κάθε φορά που φορτώνεται μία σελίδα.

```
function initClock() {
  $('#timer a.hour.plus').click(hourInc);
  $('#timer a.hour.minus').click(hourDec);

  $('#timer a.minute.plus').click(minuteInc);
  $('#timer a.minute.minus').click(minuteDec);
}
```

Αντιστοιχεί κάποιες συναρτήσεις event listeners στα στοιχεία της HTML που χρησιμοποιούμε σαν κουμπιά όταν τα πατάει ο χρήστης.

```
function initClock() {  
...  
    $('#timer a.hour.plus').click(hourInc);  
    $('#timer a.hour.minus').click(hourDec);  
  
    $('#timer a.minute.plus').click(minuteInc);  
    $('#timer a.minute.minus').click(minuteDec);  
}
```

Κάθε φορά που πατιέται κάποιο από αυτά καλείται η αντίστοιχη μέθοδος.

Η μέθοδος ξεκινάει βρίσκοντας την τρέχουσα ώρα και έπειτα κάνοντας την στρογγυλοποίηση στο πλησιέστερο δεκάλεπτο.

```
var currentDate = new Date();  
hour = currentDate.getHours();  
minute = Math.round(currentDate.getMinutes() / 10) * 10;  
  
if (minute == 60) {  
    minute = 50;  
}  
timerUpdate();
```

Αν θέλαμε να πάρουμε την τρέχουσα ώρα θα αφήναμε το μέρος της στρογγυλοποίησης εκτός. Ο μοναδικός λόγος που το έχουμε βάλει είναι για να μπορεί να φαίνεται με εύκολο τρόπο το πώς παίζει η εφαρμογή.

Η μέθοδος hourInc καλείται όταν αυξάνεται η ώρα.

```
function hourInc(){  
    hour++;  
    if(hour>23)  
        hour = 0;  
  
    timerUpdate();  
}
```

Αρχικά προσθέτει ένα στην μεταβλητή και μετά αν έχει τιμή πάνω από 23, δηλαδή 24, την κάνει 0 ώστε να μη βγει εκτός ορίων.

Η μέθοδος hourDec καλείται όταν πατιέται το κουμπί για να μικρύνει η τιμή της ώρας.

```
function hourDec(){
    hour--;
    if(hour<0)
        hour = 23;

    timerUpdate();
}
```

Αντίστοιχα μειώνει την τιμή της ώρας κατά ένα και αν γίνει αρνητικό την κάνει 23.

Η minuteInc καλείται όταν αυξάνονται τα λεπτά.

```
function minuteInc(){
    minute += minutesIncrement;

    if(minute >= 60) {
        minute = 0;
        hourInc();
    }
    timerUpdate();
}
```

Αν ξεπεράσουν το 59 τότε τα κάνει 0 και καλεί την hoursInc για να αυξήσει την ώρα κατάλληλα, ώστε και αυτή να μην ξεπεράσει τα κατάλληλα όρια.

Η συνάρτηση minuteDec καλείται για να μειώσει τα λεπτά.

```
function minuteDec(){
    minute -= minutesIncrement;

    if(minute < 0) {
        minute = 60 - minutesIncrement;
        hourDec();
    }
    timerUpdate();
}
```

Ομοίως αφαιρεί από την τιμή τους το καθορισμένο ποσό και αν είναι αρνητικά βάζει τη σωστή τιμή.

Όλες οι παραπάνω συναρτήσεις καλούν στο τέλος την timerUpdate. Ο λόγος είναι γιατί αυτές απλά πειράζουν τιμές μεταβλητών, οι οποίες δεν εμφανίζονται στην οθόνη. Για να γίνει αυτό χρειάζεται να αλλάξουν οι τιμές των στοιχείων HTML που απεικονίζουν αυτές τις τιμές.

Θέλουμε να εμφανίζουμε τις μονοψήφιες τιμές με ένα μηδενικό μπροστά, για να είναι πιο κατανοητές. Για παράδειγμα στις 9 και 6 θέλουμε να εμφανίζεται 09:06 και όχι 9:6.

```
var displayHour = hour;  
if(hour < 10 )  
    displayHour = "0" + hour;
```

#### 4.4.2.2 Λειτουργίες χάρτη

Το μεγαλύτερο μέρος του κώδικα JavaScript που έχουμε γράψει αναλαμβάνει την διαχείριση των χαρτών και την εύρεση της βέλτιστης διαδρομής. Αρχικά έχουμε κάποιες μεταβλητές για αυτές τις λειτουργίες.

Η πρώτη ομάδα σχετίζεται με τον ίδιο το χάρτη και τις υπηρεσίες του.

```
var map;  
var geocoder;  
var directionsDisplay = null;  
var directionsService;
```

Η μεταβλητή map είναι ο ίδιος ο χάρτης, ενώ το geocoder αναλαμβάνει να διαβάσει διευθύνσεις και να τις μετατρέψει σε συντεταγμένες. Εξαιτίας αυτού μπορούμε να αποθηκεύουμε κανονικές διευθύνσεις και δε χρειάζεται να καταφύγουμε σε αριθμητικές συνταγμένες.

Η directionsDisplay είναι μία μεταβλητή ενός αντικειμένου Display Rendered που αναλαμβάνει να ζωγραφίσει πάνω στο χάρτη τις διαδρομές που χρησιμοποιούμε.

Η μεταβλητή directionsService είναι μία μεταβλητή που αναπαριστά την υπηρεσία που προφέρει το Google για την εύρεση διευθύνσεων.

```
var mapOptions = {  
  zoom: 17,  
  center: new google.maps.LatLng(35.33909, 25.13366),  
  mapTypeId: google.maps.MapTypeId.ROADMAP  
};
```

Το αντικείμενο `mapOptions` περιλαμβάνει διάφορες παραμέτρους για τη δημιουργία του χάρτη. Ορίζεται το αρχικό επίπεδο `zoom`, το κέντρο σαν ένα ζεύγος συντεταγμένων του κέντρου του ηρακλείου και το είδος του χάρτη, το οποίο θέλουμε να προβάλλει μόνο τους δρόμους και όχι δορυφορικές φωτογραφίες, οι οποίες αποσπούν την προσοχή από τη διαδρομή.

Αξίζει να σημειωθεί πως πρακτικά οι συντεταγμένες του κέντρου δε χρησιμεύουν πουθενά, γιατί με το που ξεκινάει ο χάρτης κεντράρει στην τοποθεσία του χρήστη. Υπάρχουν απλά επειδή είναι απαραίτητες για την αρχικοποίηση.

```
var addresses = [];  
var opening = [];  
var closing = [];  
var calculated = [];  
  
var currentRoute = -1;  
var bestRoute = -1;  
  
var routes = [];
```

Οι επόμενες μεταβλητές χρησιμοποιούνται για την εύρεση των κατευθύνσεων. Όλες αυτές στην αρχή έχουν κενές ή άκυρες τιμές, ενώ γεμίζουν κατά την διαδικασία.

Ο πίνακας `addresses` περιέχει τις διευθύνσεις όλων των προορισμών ανάμεσα στους οποίους πρέπει κάθε φορά να διαλέξει η εφαρμογή.

Οι πίνακες `opening` και `closing` περιέχουν την ώρα που ανοίγει ή κλείνει το κατάστημα – προορισμός ή στην περίπτωση των ταινιών η ώρα που ξεκινάει και τελειώνει κάθε προβολή. Γεμίζουν με τις αντίστοιχες ώρες που τις μετατρέπουμε σε δευτερόλεπτα για να είναι συγκρίσιμες με τους χρόνους που επιστρέφει το Google για τις διαδρομές του.

Όλοι αυτοί οι πίνακες περιέχουν τα δεδομένα σε αντίστοιχες μεταξύ τους θέσεις για να μπορούμε να αναφερθούμε εύκολα σε αυτά. Για παράδειγμα το βενζινάδικο που θα περιέχει τη διεύθυνση του στην τρίτη θέση του πίνακα `addresses` έχει την ώρα που ανοίγει στην τρίτη θέση του πίνακα `opening` και την ώρα που κλείνει στην τρίτη θέση του πίνακα `closing`.

Η μεταβλητή `currentRoute` χρησιμοποιείται παρακάτω για να δείχνει το τρέχον κατάστημα του οποίου η διαδρομή αναζητείται, πχ αν ψάχνουμε βενζινάδικο και έχει τιμή 2 σημαίνει ότι εξετάζεται το δεύτερο από το σύνολό τους.

Η τιμή `bestRoute` περιέχει η θέση των πινάκων στην οποία βρίσκεται το καλύτερο κατάστημα.

Τέλος η μεταβλητή `routes` είναι ένας πίνακας που περιέχει όλες τις πληροφορίες διαδρομής που επέστρεψε το Google Directions για κάθε έναν προορισμό.

```
function reset() {
  addresses = [];
  opening = [];
  closing = [];
  calculated = [];

  currentRoute = -1;
  bestRoute = -1;

  routes = [];
}
```

Η μέθοδος `reset` καλείται για να επαναφέρει τις παραπάνω μεταβλητές στις αρχικές τους τιμές, ώστε να μπορούν να γίνουν επιπλέον αναζητήσεις.

Έχουμε μία βοηθητική `timeToSecond` η οποία μετατρέπει ένα `string` της μορφής 10:39 που αναπαριστά την τρέχουσα ώρα σε δευτερόλεπτα.

```
function timeToSeconds(time) {
  var res = time.split(":");
  var seconds = res[0] * 60 * 60 + res[1] * 60;

  return seconds;
}
```

Αρχικά κόβει το `string` σε δύο τμήματα που χωρίζονται στην άνω κάτω τελεία και πολλαπλασιάζει το πρώτο με  $60 \cdot 60$ , που είναι τα δευτερόλεπτα που έχει μία ώρα, και το δεύτερο με 60 που είναι τα δευτερόλεπτα που έχει κάθε λεπτό. Αυτές οι δύο τιμές έπειτα προστίθενται και επιστρέφονται.

Πέρα από αυτό υπάρχει η συνάρτηση `findNearestListener` που είναι ο `listener` που μπαίνει στο κουμπί (`link`) που πατάει ο χρήστης για να βρει τον πλησιέστερο προορισμό.

Αρχικά αρχικοποιεί τα αντικείμενα των υπηρεσιών του Google `DirectionService` και `DirectionsRenderer`. Το πρώτο χρειάζεται ώστε να μπορούμε να κάνουμε ερωτήσεις στο Google για να πάρουμε οδηγίες για διάφορες διαδρομές συμπεριλαμβανομένων των στοιχείων για την απόσταση και τη διάρκειά τους.

Το αντικείμενο `DirectionsRenderer` είναι υπεύθυνο για τον σχεδιασμό των διαδρομών στον χάρτη.

```
directionsService = new google.maps.DirectionsService();  
if (directionsDisplay != null) {  
    directionsDisplay.setMap(null);  
    directionsDisplay = null;  
}  
  
directionsDisplay = new google.maps.DirectionsRenderer();  
directionsDisplay.setMap(map);
```

Για να σβήσουμε τις υπάρχουσες διαδρομές από τον χάρτη στην περίπτωση που έχει ήδη σχεδιαστεί κάποια προηγουμένως κάνουμε null τις τιμές του directionsDisplay αν έχει ήδη σχεδιαστεί πριν δώσουμε νέες τιμές.

Έπειτα η μέθοδος καλεί την getDestinations για να αρχικοποιηθούν οι πίνακες με τις διευθύνσεις των έγκυρων προορισμών και τέλος την findNearest για να ξεκινήσει η διαδικασία εύρεσης των διαδρομών.

```
getDestinations(".list");  
  
findNearest();  
  
return false;
```

Η συνάρτηση τέλος επιστρέφει false επειδή μπαίνει σαν listener σε ένα link το οποίο δεν θέλουμε να πηγαίνει τον χρήστη σε κάποια άλλη σελίδα.

Η μέθοδος getDestinations καλείται για να δώσει αρχικές τιμές στους πίνακες με τις διευθύνσεις και τους προορισμούς.

```
function getDestinations(element) {  
    $(element + ' .address').each(function()) {  
        addresses.push(this.innerHTML);  
    });  
    $(element + ' .opening').each(function()) {  
        opening.push(timeToSeconds(this.innerHTML));  
    });  
    $(element + ' .closing').each(function()) {  
        closing.push(timeToSeconds(this.innerHTML));  
    });  
}
```

Παίρνει σαν παράμετρο τον CSS descriptor του στοιχείου που περιέχει τα αντικείμενα με τα στοιχεία της λίστας και μετά διαλέγει με χρήση jQuery τα αντικείμενα που περιέχουν τις συγκεκριμένες πληροφορίες. Κατά τη δημιουργία του κώδικα της HTML

φροντίζουμε ώστε η δομή του εγγράφου να περιέχει τις απαραίτητες πληροφορίες με αυτόν τον τρόπο.

Η μέθοδος `findNearest` αναλαμβάνει να συγκεντρώσει όλες τις πληροφορίες για όλες τις δυνατές διαδρομές και έπειτα να διαλέξει την καλύτερη. Αρχικά αυξάνει τον δείκτη της τρέχουσας διαδρομής και έπειτα, αν υπάρχουν ακόμη διαδρομές υπολογίζει τις οδηγίες για την επόμενη. Σε αντίθετη περίπτωση βρίσκει την καλύτερη από αυτές που υπολόγισε.

```
currentRoute++;  
  
if (currentRoute < addresses.length) {  
    ...  
} else {  
    ...  
}
```

Αν πρέπει να υπολογίσει μία νέα διαδρομή τότε καλεί την `calcRoute` με παραμέτρους την τρέχουσα διεύθυνση του χρήστη και την διεύθυνση προορισμού της τρέχουσας διαδρομής.

```
calcRoute(currentAddress, addresses[currentRoute]);
```

Σε αντίθετη περίπτωση υπολογίζει την τρέχουσα ώρα σε δευτερόλεπτα από τις τιμές των μεταβλητών `hour` και `minute` του ρολογιού. Έπειτα διατρέχει όλες τις υπολογισμένες διαδρομές.

```
var currentTime = hour * 60 * 60 + minute * 60;  
var openingTime, closingTime, calcTime;  
  
var i;  
  
for (i = 0; i < routes.length; i++) {  
    calculated.push(currentTime +  
routes[i].routes[0].legs[0].duration.value + overhead);  
  
    if (calculated[i] > opening[i] && calculated[i] <  
closing[i]) {  
        if (bestRoute == -1 || calculated[bestRoute] >  
calculated[i]) {  
            bestRoute = i;  
        }  
    }  
}
```



Για κάθε μία από αυτές προσθέτει στον πίνακα `calculated` την τρέχουσα ώρα συν το χρόνο της διαδρομής συν τυχόν επιπλέον χρόνο που χρειάζεται με τη μεταβλητή `overhead`, όπως για παράδειγμα στην αναζήτηση `super market`, που χρειάζεται επιπλέον 40 λεπτά.

Έπειτα αν ο χρόνος που υπολογίστηκε ότι ο χρήστης θα φτάσει στον προορισμό είναι μικρότερος από τον χρόνο που κλείνει το κατάστημα αλλά μεγαλύτερος από την ώρα που ανοίγει και καλύτερο από το προηγούμενο αποτέλεσμα τότε καταχωρεί αυτή την τιμή σαν καλύτερο αποτέλεσμα.

Όταν τελειώσει, αν έχει βρεθεί κάποια καλύτερη διαδρομή τότε τη ζωγραφίζει.

```
if (bestRoute != -1) {
    directionsDisplay.setDirections(routes[bestRoute]);
    $(".response p").html("Στοιχεία διαδρομής: Απόσταση " +
        routes[bestRoute].routes[0].legs[0].distance.text + ", διάρκεια " +
        routes[bestRoute].routes[0].legs[0].duration.text);
} else {
    $(".response p").html("Αυτή τη στιγμή δεν υπάρχει κάποιος
    έγκυρος προορισμός.");
}
```

Αν δεν υπάρχει κανένα αποτέλεσμα, για παράδειγμα επειδή δεν υπάρχουν ανοιχτά μαγαζιά εκείνη την ώρα τότε εμφανίζει το κατάλληλο μήνυμα.

Στο τέλος καλεί την μέθοδο `reset`, ώστε ο χρήστης να μπορέσει να εκτελέσει και άλλες αναζητήσεις.

Η μέθοδος `calcRoute` παίρνει για παραμέτρους δύο διευθύνσεις, σαν `string`, και βρίσκει την μεταξύ τους διαδρομή.

```
function calcRoute(start, end) {

    var request = {
        origin: start,
        destination: end,
        travelMode: google.maps.TravelMode.DRIVING
    };
    directionsService.route(request, function(response, status) {
        if (status == google.maps.DirectionsStatus.OK) {
            routes.push(response);

            findNearest();
        }
    });
}
```

Επειδή η κλήση είναι ασύγχρονη δίνεται σαν callback η μέθοδος findNearest ώστε είτε να την ξανακαλέσει με τον νέο προορισμό είτε να βρει την καλύτερη διαδρομή.

## 5 Οδηγίες χρήσης

Εδώ θα δείξουμε τις βασικές οδηγίες χρήσης της εφαρμογής μας. Για την λειτουργία της εφαρμογής θα πρέπει να τρέχουν ο Wampserver και ο BaseX server.

### 5.1 Είσοδος στο σύστημα

Για την είσοδο στο σύστημα ο χρήστης χρειάζεται να κατευθυνθεί στη διεύθυνση <http://localhost/maps/>. Αν δεν έχει ήδη μπει στο σύστημα τότε θα του δείξει τη φόρμα εισόδου για την είσοδο στο σύστημα.

Υπηρεσίες δρομολόγησης και καθοδήγησης

Τρέχουσα ώρα: + 00 - : + 00 -

**Είσοδος**

Εισάγετε τα στοιχεία σας στη φόρμα για να συνεχίσετε:

Όνομα χρήστη:

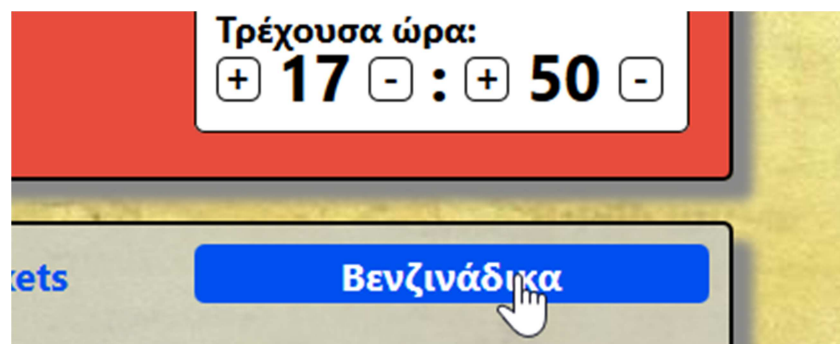
Κωδικός:

Τμήμα Μηχανικών Πληροφορικής, ΑΤΕΙ Ηρακλείου  
Σύστημα παροχής υπηρεσιών δρομολόγησης και καθοδήγησης  
Πτυχιακή εργασία Ιωάννη Αλμπάνη και Σπύρου Νταλιάνη

Μετά την επιτυχή εισαγωγή των στοιχείων μπορεί να προχωρήσει στην επόμενη σελίδα.

### 5.2 Αναζήτηση βενζινάδικου

Για να μπει ο χρήστης στην λειτουργία αναζήτησης βενζινάδικου πρέπει να πατήσει το αντίστοιχο πλήκτρο από το μενού.



Εκεί παρουσιάζεται ο χάρτης με τη διεύθυνση του χρήστη και μία λίστα με τα καταχωρημένα βενζινάδικα.

## Καταχωρημένα βενζινάδικα

### **ΕΚΟ - Αφοί Σαμολή**

Μουστακλή Σπύρου 26, Ηράκλειο

+302810252525

Ωράριο: 07:00 - 20:00

### **ΕΚΟ - Χριστοδουλάκης Π. - Φουντάκη Μ. Ο.Ε.**

Μεσσαράς 2, Ηράκλειο, Ελλάδα

+302810223344

Ωράριο: 07:00 - 20:00

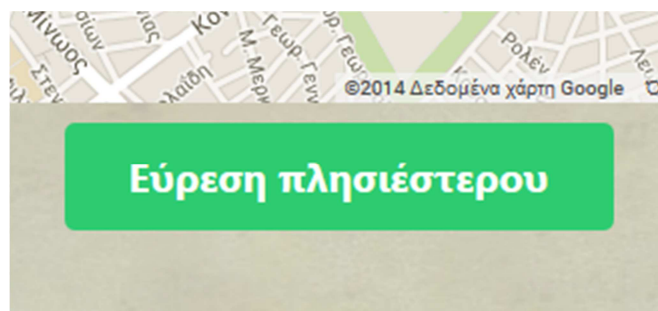
### **ΕΚΟ - Χαράλαμπος Μαυράκης και ΣΙΑ Ε.Ε.**

Κρασαδάκη Νίκου, 71409 Ηράκλειο

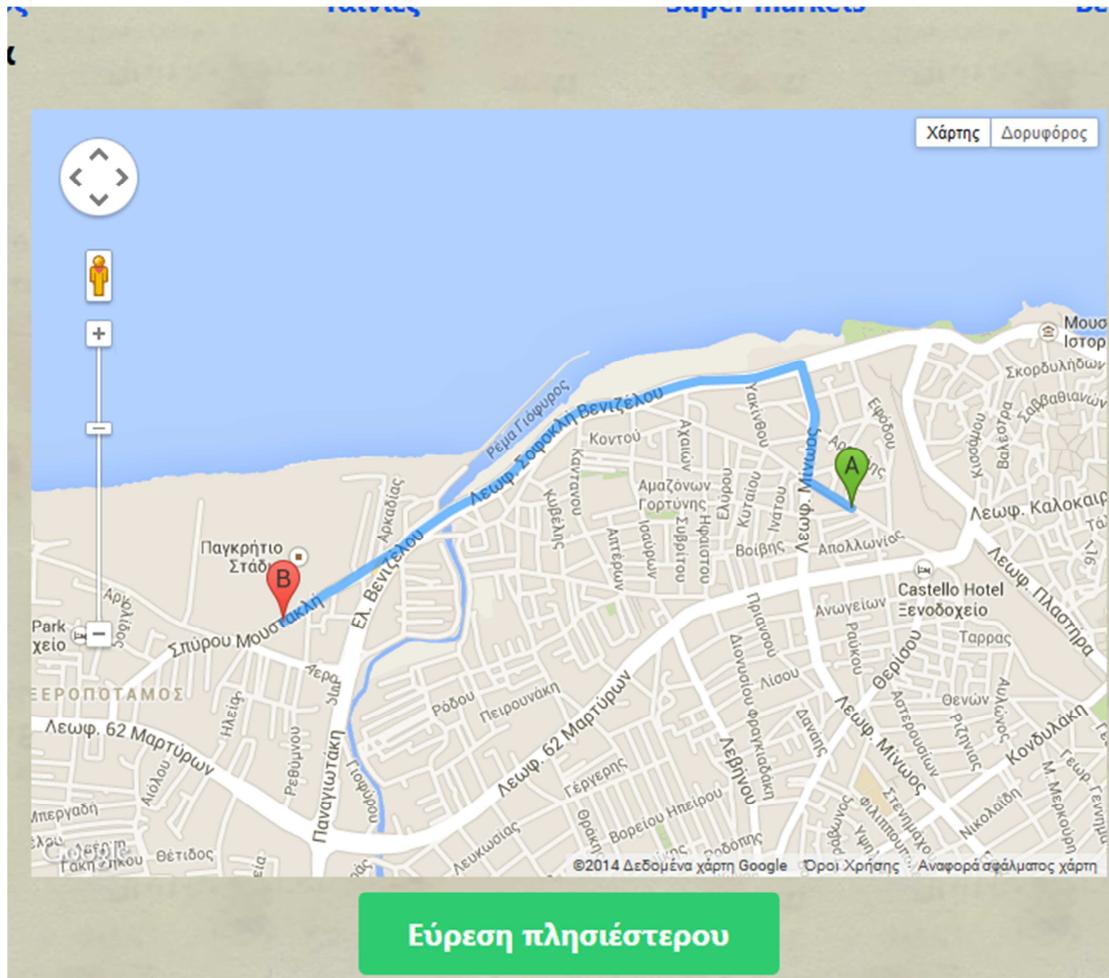
+302810555888

Ωράριο: 07:00 - 20:00

Για αναζήτηση του προτιμότερου βενζινάδικου μπορεί να πατήσει το πλήκτρο εύρεση πλησιέστερου.



Η εφαρμογή θα δοκιμάσει να βρει την προτιμότερη διαδρομή και θα την εμφανίσει.



### 5.3 Εύρεση super market

Για τη λειτουργία εύρεσης super market ο χρήστης διαλέγει την επιλογή Super markets από το μενού.



Εμφανίζεται μία παρόμοια σελίδα με την εύρεση του βενζινάδικου. Πατώντας το κουμπί εύρεσης πάλι θα εμφανιστεί η διαδρομή στο χάρτη αλλά αυτή τη φορά λαμβάνεται και ένα επιπλέον χρονικό διάστημα 40 λεπτών για τα ψώνια που θα κάνει.



## Ταινίες

### Captain America 2

Εύρεση πλησιέστερου

#### Τεχνόπολις

Τεχνόπολις, Λεωφόρος Ανδρέα Παπανδρέου, Ηράκλειο, Ελλάδα, Προβολή: 20:30 - 22:00

#### Τεχνόπολις

Τεχνόπολις, Λεωφόρος Ανδρέα Παπανδρέου, Ηράκλειο, Ελλάδα, Προβολή: 22:00 - 23:30

#### Βιτσέντζος Κορνάρος

Μαλικούτη 18 - 20, Ηράκλειο, Ελλάδα, Προβολή: 21:30 - 23:00

### Dallas Buyers Club

Εύρεση πλησιέστερου

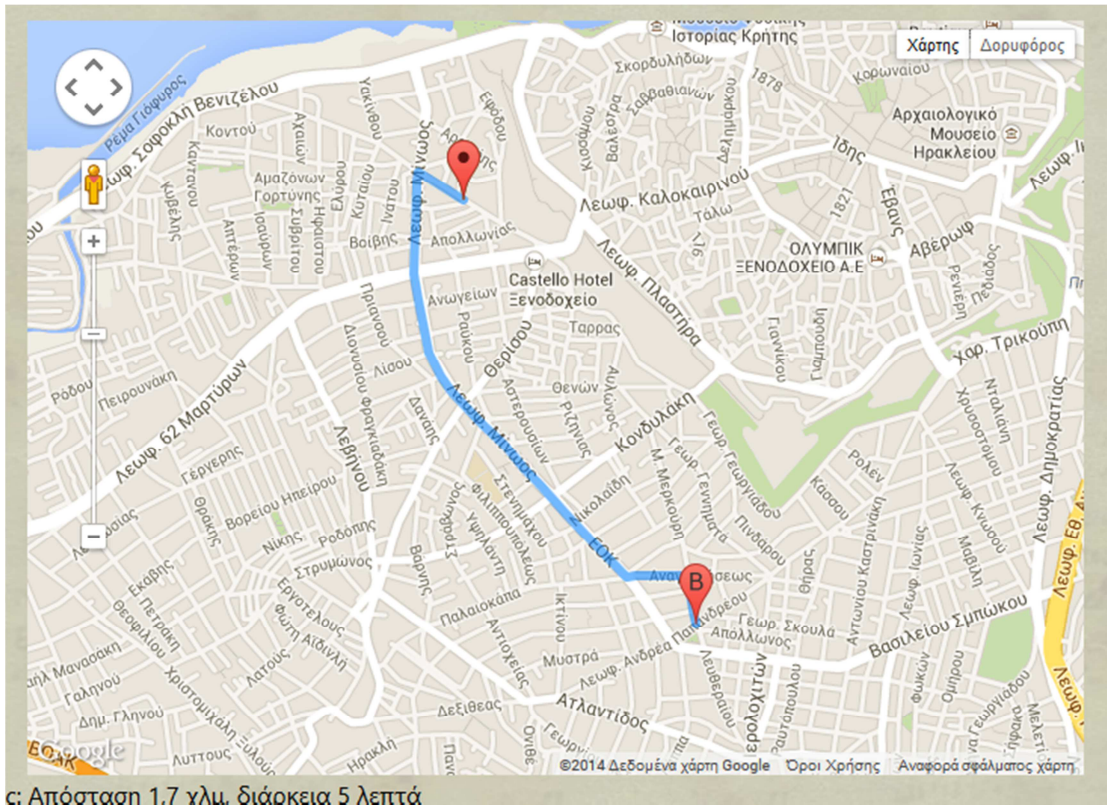
#### Τεχνόπολις

Τεχνόπολις, Λεωφόρος Ανδρέα Παπανδρέου, Ηράκλειο, Ελλάδα, Προβολή: 21:15 - 23:00

#### Βιτσέντζος Κορνάρος

Μαλικούτη 18 - 20, Ηράκλειο, Ελλάδα, Προβολή: 21:30 - 23:00

Ο χρήστης πρέπει να πατήσει το κουμπί δίπλα από τον τίτλο της ταινίας που τον ενδιαφέρει για να βρει την καλύτερη.





## 6 Συμπεράσματα

Κατά την ενασχόλησή μας με τις υπηρεσίες των ηλεκτρονικών χαρτών δείξαμε πολλές από τις δυνατότητές τους και τον βασικό τρόπο ανάπτυξης ιστοσελίδων που χρησιμοποιούν μία βάση δεδομένων XML για την αποθήκευση των δεδομένων.

Οι τεχνολογίες έχουν εντυπωσιακές δυνατότητες, γεγονός που μέχρι κάποιο βαθμό το περιμέναμε καθώς αποτελούν τη βάση του παγκόσμιου ιστού. Τόσο σε εικαστικό επίπεδο όσο και σε επίπεδο χρησιμότητας είναι αρκετά αναπτυγμένες και αποτελούν πολύτιμα εργαλεία για τον προγραμματιστή που είναι αρκετά ενημερωμένων ώστε να αξιοποιήσει το μέγιστο των δυνατοτήτων τους.

Ιδιαίτερη μνεία πρέπει να γίνει στην υπηρεσία Google Maps. Και μόνο το γεγονός πως οποιοσδήποτε έχει πρόσβαση σε ένα εργαλείο ηλεκτρονικών χαρτών το οποίο μπορεί να το ενσωματώσει στη σελίδα του είναι από μόνο του εντυπωσιακό, λαμβάνοντας υπόψιν την πολυπλοκότητα του και τις υποδομές που χρειάζονται για την λειτουργία του. Όμως η υπηρεσία πηγαίνει ένα βήμα παραπέρα και δίνει και έξτρα δυνατότητες όπως η παροχή οδηγιών που χρησιμοποιήσαμε. Βέβαια για εμπορική χρήση αυτό έχει κόστος, καθώς υπάρχει ένα όριο στο πλήθος των ερωτήσεων που παρέχονται δωρεάν.

### 6.1 Μελλοντική επέκταση

Μια βασική αλλαγή που θα μπορούσε να γίνει στην εφαρμογή σε κάποια διαφορετική υλοποίηση είναι η χρήση μίας σχεσιακής βάσης δεδομένων ώστε να γίνει σύγκριση με την βάση XML. Επίσης η γλώσσα PHP που έχει χρησιμοποιηθεί για την δημιουργία των σελίδων μπορεί να αντικατασταθεί από κάποια άλλη γλώσσα όπως η Java ή το .Net.

Οι παραπάνω αλλαγές βέβαια, αν και απαιτούν πολύ δουλειά δεν επηρεάζουν ιδιαίτερα τη λειτουργικότητα της εφαρμογής. Το σημείο στο οποίο θα φανεί ιδιαίτερα η χρησιμότητα των χαρτών είναι σε μία εφαρμογή κινητών τηλεφώνων με τις ίδιες δυνατότητες. Σε αυτή την περίπτωση θα μπορούσε να χρησιμοποιηθεί η λειτουργία εύρεσης θέσης, ώστε να προσφέρονται ζωτικές πληροφορίες στον χρήστη σε πραγματικό χρόνο.

## 7 Βιβλιογραφία

BaseX Download. [Ηλεκτρονικό] <http://basex.org/products/download/all-downloads/>.

Database. *Wikipedia*. [Ηλεκτρονικό] <http://en.wikipedia.org/wiki/Database>.

Documentation of BaseX. [Ηλεκτρονικό] [http://docs.basex.org/wiki/Main\\_Page](http://docs.basex.org/wiki/Main_Page).

Dynamic Website. *Marvel Infomatics*. [Ηλεκτρονικό]  
[http://www.marvelinfomatics.com/web-development/dynamic\\_website.html](http://www.marvelinfomatics.com/web-development/dynamic_website.html).

Dynamic Websites . *smooth-step.com*. [Ηλεκτρονικό] <http://www.smooth-step.com/web-design/dynamic-websites>.

HTML. *Βικιπαίδεια*. [Ηλεκτρονικό] <http://el.wikipedia.org/wiki/HTML>.

HTML 4.0 Specification. *World Wide Web Consortium*. [Ηλεκτρονικό]  
<http://www.w3.org/TR/1998/REC-html40-19980424/>.

HTML5. *Βικιπαίδεια*. [Ηλεκτρονικό] <http://el.wikipedia.org/wiki/HTML5>.

HTML5. *Wikipedia*. [Ηλεκτρονικό] <http://en.wikipedia.org/wiki/HTML5>.

HTML5 Differences from HTML4. *World Wide Web Consortium*. [Ηλεκτρονικό]  
<http://www.w3.org/TR/2011/WD-html5-diff-20110405/>.

MySQL. *Βικιπαίδεια*. [Ηλεκτρονικό] <http://el.wikipedia.org/wiki/MySQL>.

onload Event. *w3schools*. [Ηλεκτρονικό]  
[http://www.w3schools.com/jsref/event\\_onload.asp](http://www.w3schools.com/jsref/event_onload.asp).

PHP: Hypertext Preprocessor. *PHP.net*. [Ηλεκτρονικό] <http://php.net/>.

**Pilgrim, Mark**. Dive Into HTML5. *Dive Into HTML5*. [Ηλεκτρονικό]  
<http://diveintohtml5.info/>.

Processing static web pages. *etutorials.org*. [Ηλεκτρονικό]  
<http://etutorials.org/Macromedia/Dream+Weaver+Online+Help/Getting+Started+with+Dreamweaver/Understanding+Web+Applications/How+a+web+application+works/Processing+static+web+pages/>.

Usage share of web browsers. *Wikipedia*. [Ηλεκτρονικό]  
[https://en.wikipedia.org/wiki/Usage\\_share\\_of\\_web\\_browsers](https://en.wikipedia.org/wiki/Usage_share_of_web_browsers).

Γλώσσες Προγραμματισμού. [Ηλεκτρονικό]  
[https://foss.ntua.gr/wiki/index.php/%CE%93%CE%BB%CF%8E%CF%83%CF%83%CE%B5%CF%82\\_%CE%A0%CF%81%CE%BF%CE%B3%CF%81%CE%B1%CE%BC%CE%BC%CE%B1%CF%84%CE%B9%CF%83%CE%BC%CE%BF%CF%8D#PHP](https://foss.ntua.gr/wiki/index.php/%CE%93%CE%BB%CF%8E%CF%83%CF%83%CE%B5%CF%82_%CE%A0%CF%81%CE%BF%CE%B3%CF%81%CE%B1%CE%BC%CE%BC%CE%B1%CF%84%CE%B9%CF%83%CE%BC%CE%BF%CF%8D#PHP).

Εισαγωγή στις Βάσεις Δεδομένων (DataBases). *KENTPO ΠΛΗ.ΝΕ.Τ. Ν. ΦΛΩΡΙΝΑΣ*. [Ηλεκτρονικό] <http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-DataBases.html>.

Η Γλώσσα Προγραμματισμού PHP . ΚΕΝΤΡΟ ΠΛΗ.ΝΕ.Τ. Ν. ΦΛΩΡΙΝΑΣ. [Ηλεκτρονικό]  
<http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-Php-Analytical.html>.

Τι είναι html? [Ηλεκτρονικό] <http://pacific.jour.auth.gr/html/>.

## 8 Παράρτημα – Κώδικας εφαρμογής

Εδώ παραθέτουμε τον κώδικα της εφαρμογής.

### 8.1 Αρχεία CSS

#### 8.1.1 style.css

```
@charset "utf-8";
body {
    font: 100%/1.5 Segoe UI, Verdana, Arial, Helvetica, sans-
    serif;
    background: url('images/bg.jpg') repeat #e74c3c;
    margin: 0;
    padding: 0;
    color: #000000;
}
ul, ol, dl {
    padding: 0;
    margin: 0;
}
h1, h2, h3, h4, h5, h6, p {
    margin-top: 0;
    padding-right: 15px;
    padding-left: 15px;
}
a img {
    border: none;
}

.left {
    float: left;
}

.right {
    float: right;
}

.clear {
    clear: both;
}

.container {
    width: 1050px;
    /* background-color: #95a5a6; */
```

```
        margin: 10px auto;

    }

    header {
        padding:20px 20px 10px 20px;
        margin-bottom:20px;
        color: #000; text-shadow: 0px 0px 3px #fff;
        background: url('images/logo.png') no-repeat 20px 20px #e74c3c;
        box-shadow: 10px 10px 5px #888888;
        border: 2px solid #000;
        border-radius: 5px;
    }

    header h1 {
        float: left;
        width: 500px;
        margin-left: 60px;
        line-height: 32px;
    }

    header #timer {
        float: right;
        width: 225px;
        font-size: 32px;
        font-weight: bold;

        line-height: 29px;
        background-color: #fff;
        border: 1px solid #000;
        border-radius: 5px;
        padding: 10px;
    }

    header #timer h6 {
        margin: 0;
        padding: 0;
        font-size: 18px;
        line-height: 18px;
        font-weight: bold;
    }

    header #timer a{
```

```
display: inline-block;
width: 20px; height: 20px;
margin-bottom: 5px;

background-color: #fff;
border: 1px solid #000;
border-radius: 5px;

vertical-align: middle;
color: #000;
text-align: center;
line-height: 16px;
text-decoration: none;
font-size: 20px;
font-weight: bolder;
}

header #timer a:hover {
    background-color: #ccc;
}

article {
    background-color: rgba(195, 200, 204, 0.7);
    margin-bottom: 20px;
    border-radius: 5px;
    box-shadow: 10px 10px 5px #888888;
    border: 2px solid #000;
    padding: 10px;
    min-height: 300px;
}

nav li {
    display: block;
    float: left;
    width: 25%;
}

nav li a{
    border-radius: 5px;
    display: block;
    text-decoration: none;
    text-align: center;
    font-weight: bold;
```

```
    font-size: 1.2em;
    color: #0050ef;
}

nav li a:hover {
    background-color: #0050ef;
    color: #ffffff;
}

footer {
    background-color: #34495e;
    margin-bottom: 20px;
    border-radius: 5px;
    border: 2px solid #000;
    box-shadow: 10px 10px 5px #888888;
    padding: 10px;
}

footer p {
    font-size: 0.8em;
    color: #fff;
    margin-bottom: 5px;
}

article label, article input {
    display: inline-block;
    width: 180px;
    margin-left: 20px;
}

article input[type="submit"] {
    display: inline-block;
    width: 180px;
    margin-top: 10px;
    margin-left: 110px;
}

div.list-item {
    border-radius: 5px;
    display: block;
    float: left;
    width: 495px;
    background-color: #fff;
    margin: 5px;
```

```
padding: 5px;

}

div.list-item.wide {
  float: none;
  width: 700px;
  margin: 5px auto;
}

div.list-item:hover {
  background-color: #f1c40f;
}

div.list-item p {
  margin-bottom: 5px;
}

div.movie-list {
  margin-top: 50px;
}

div.movie-list h3 {
  float: left;
  width: 500px;
  padding-left: 100px;
}

.button-large, .button-small{
  display: block;
  width: 250px;
  margin: 10px auto;
  border-radius: 5px;
  padding: 10px;

  border: 2px solid #2ecc71;
  background-color: #2ecc71;
  color: #fff;

  font-size: 20px;
  font-weight: bold;
  text-decoration: none;
  text-align: center;
}
```



```
.button-small{
    display: block;
    float: right;
    width: 150px;
    margin: 5px 100px 5px 5px;
    border-radius: 5px;
    padding: 5px;

    font-size: 14px;
}

.button-large:hover, .button-small:hover {
    color: #2ecc71;
    background-color: #fff;
}
```

## 8.2 Αρχεία PHP

### 8.2.1 \_page\_end.php

```
</article><!-- END article -->

<footer>
    <p>Τμήμα Μηχανικών Πληροφορικής, ΑΤΕΙ Ηρακλείου</p>
    <p>Σύστημα παροχής υπηρεσιών δρομολόγησης και
καθοδήγησης</p>
    <p>Πτυχιακή εργασία Ιωάννη Αλπάνη και Σπύρου
Νταλιάνη</p>

</footer>
</div><!-- END .container -->
</body>
</html>
```

### 8.2.2 \_page\_start.php

```
<?php session_start() ?>
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>Υπηρεσίες δρομολόγησης και καθοδήγησης</title>
    <link href="style.css" rel="stylesheet" type="text/css">

    <script src="js/jquery-1.8.2.js"></script>
```

```
<script src="js/jquery.validate.js"></script>
</head>

<body>
  <div class="container">

    <header>
      <h1>Υπηρεσίες δρομολόγησης και καθοδήγησης</h1>

      <div id="timer">
        <h6>Τρέχουσα ώρα:</h6>
        <a href="#" class="button hour plus">+</a>
        <span class="hour">00</span>
        <a href="#" class="button hour minus">-</a>
        :
        <a href="#" class="button minute plus">+</a>
        <span class="minute">00</span>
        <a href="#" class="button minute minus">-</a>
      </div>

      <div class="clear"></div>
    </header>

    <article>

    <nav>
      <ul>
        <?php
          if(isset($_SESSION['name'])){
            ?>
            <li><a href="do_logout.php">Έξοδος</a></li>

            <li><a href="cinemas.php">Ταινίες</a></li>
            <li><a href="supermarkets.php">Super markets</a></li>
            <li><a href="gasstations.php">Βενζινάδικα</a></li>
            <?php
              }else{
                echo '<li><a href="index.php">Είσοδος</a></li>';
              }
            ?>
          </ul>
        </nav>
      <div class="clear"></div>
```

### 8.2.3 cinemas.php

```
<?php
require_once('_page_start.php');
?>
<script
src="http://maps.googleapis.com/maps/api/js?key=AIzaSyAigN5xKZuQcnVe
eNF0tsGMYo2YmHzGJos&sensor=false&language=el"></script>
<script src="map.js"></script>

<script>
    var overhead = 0;

    function initialize() {
        geocoder = new google.maps.Geocoder();
        map = new
google.maps.Map(document.getElementById("googleMap"), mapOptions);

        setMarkerAtAddress('<?php echo $_SESSION['address']; ?>',
map, geocoder, true);
    }

    $(document).ready(function() {
        initClock();
        $(".calcButton").click(findNearestCinemaListener);
        google.maps.event.addDomListener(window, 'load',
initialize);
        currentAddress = "<?php echo $_SESSION['address'] ?>";
    });

</script>

<h1>Ταινίες</h1>

<div id="googleMap"
style="width:700px;height:500px;margin:auto"></div>

<div class="response">
    <p></p>
</div>
```

<h2>Ταινίες</h2>

```
<?php
$path = "http://localhost:8984/rest/maps?wrap=yes&query=";
$query = "//movies/movie";

$response = file_get_contents($path . $query);
$response = str_replace('rest:', '', $response);

$xml = new SimpleXMLElement($response);

$i = 0;
foreach ($xml as $item) {
    $i++;
    $movie = $item->movie;
    echo '<div class="list list' . $i . '">';

    echo '<div class="movie-list">';
    echo '<h3>' . $movie->title . '</h3>';
    echo '<a class="calcButton centered button-small" href="#"'
    id="list' . $i . '">Εύρεση πλησιέστερου</a>';
    echo '<div class="clear"></div>';

    foreach ($movie->cinemas->cinema as $cinema) {
        echo '<div class="list-item wide">';
        echo '<h4>' . $cinema->name . "</h4>";
        echo '<p><span class="address">' . $cinema->address .
        "</span>, ";
        echo '<em>Προβολή: <span class="opening">' . $cinema->hours-
        >opening . '</span> - <span class="closing">' . $cinema->hours-
        >closing . '</span></em></p>';
        echo '</div>';
    }
    echo '<div class="clear"></div>';
    echo '</div>';
    echo '</div>';
}
?>

<div class="clear"></div>

<?php
require_once('_page_end.php');
```

?>

#### 8.2.4 do\_login.php

```
<?php
ob_start();

session_start();

$path = "http://localhost:8984/rest/maps?wrap=yes&query=";
$query = "//users/user[@name='#1' and @password='#2']";

$name = $_REQUEST['name'];
$password = $_REQUEST['password'];

$query = str_replace('#1', $name, $query);
$query = str_replace('#2', $password, $query);

$query = urlencode($query);

$response = file_get_contents($path.$query);
$response= str_replace('rest:', '', $response);

$xml = new SimpleXMLElement($response);

if($xml->result->user == '') {
    header("Location: index.php");
}else{

    $_SESSION['name'] = strval($xml->result->user['name']);
    $_SESSION['fullname'] = strval($xml->result->user->fullname);
    $_SESSION['address'] = strval($xml->result->user->address);
    session_write_close();
    header("Location: gasstations.php");
}
ob_flush();
?>
```

#### 8.2.5 do\_logout.php

```
<?php

session_start();

unset($_SESSION['name']);
unset($_SESSION['fullname']);
unset($_SESSION['address']);
```

```
header("Location: index.php");  
?>
```

### 8.2.6 gasstations.php

```
<?php  
require_once('_page_start.php');  
?>  
<script  
src="http://maps.googleapis.com/maps/api/js?key=AIzaSyAigN5xKZuQcnVe  
eNF0tsGMYo2YmHzGJos&sensor=false&language=el"></script>  
<script src="map.js"></script>  
  
<script>  
    var overhead = 0;  
  
    function initialize() {  
        geocoder = new google.maps.Geocoder();  
        map = new  
google.maps.Map(document.getElementById("googleMap") ,mapOptions);  
  
        setMarkerAtAddress('<?php echo $_SESSION['address']; ?>',  
map, geocoder, true);  
    }  
  
    $(document).ready(function() {  
        initClock();  
        $("#calcButton").click(findNearestListener);  
        google.maps.event.addDomListener(window, 'load',  
initialize);  
        currentAddress = "<?php echo $_SESSION['address'] ?>";  
    });  
  
</script>  
  
<h1>Βενζινάδικα</h1>  
  
<div id="googleMap"  
style="width:700px;height:500px;margin:auto"></div>  
  
<a id="calcButton" class="centered button-large" href="#">Εύρεση  
πλησιέστερου</a>  
  
<div class="response">  
    <p></p>  
</div>  
  
<h2>Καταχωρημένα βενζινάδικα</h2>
```

```
<div id="list" class="list">
<?php
$path = "http://localhost:8984/rest/maps?wrap=yes&query=";
$query = "//gasstations/gasstation";

$response = file_get_contents($path.$query);
$response= str_replace('rest:', '', $response);

$xml = new SimpleXMLElement($response);
//var_dump($xml);
foreach($xml as $item) {
    $gasstation = $item->gasstation;
    echo '<div class="list-item">';
    echo '<h4>'.$gasstation->name."</h4>";
    echo '<p class="address">'.$gasstation->address."</p>";
    echo '<p>'.$gasstation->phone."</p>";
    echo '<p>Ωράριο: <span class="opening">'.$gasstation->hours-
>opening.'</span> - <span class="closing">'.$gasstation->hours-
>closing.'</span></p>';
    echo '</div>';
}

?>
</div>
<div class="clear"></div>

<?php
require_once('_page_end.php');
?>
```

### 8.2.7 index.php

```
<?php
require_once('_page_start.php');
?>
<script>
    $(document).ready(function() {
        $("form").validate({
            errorClass: "invalid",
            errorElement: ""
        });
    });
</script>

<h1>Είσοδος</h1>
    <p>Εισάγετε τα στοιχεία σας στη φόρμα για να συνεχίσετε:</p>

    <form method="post" action="do_login.php">
        <label>Όνομα χρήστη:</label><input type="text" name="name"
required value="jrap"/><br/>
        <label>Κωδικός:</label><input type="password"
name="password" required value="qwe123"/><br/>

        <input type="submit" value="Υποβολή" />
```

```
</form>
<?php
require_once('_page_end.php');
?>
```

### 8.2.8 map.php

```
<?php
require_once('_page_start.php');
?>
<script
src="http://maps.googleapis.com/maps/api/js?key=AIzaSyAigN5xKZuQcnVe
eNF0tsGMYo2YmHzGJos&sensor=false"></script>

<script>
function initialize()
{
var mapProp = {
center:new google.maps.LatLng(35.33909,25.13366),
zoom:17,
mapTypeId:google.maps.MapTypeId.ROADMAP
};
var map=new google.maps.Map(document.getElementById("googleMap")
,mapProp);
}

//google.maps.event.addDomListener(window, 'load', initialize);
</script>
```

```
<h1>Διαδρομή</h1>
<section>
<div id="googleMap"
style="width:700px;height:500px;margin:auto"></div>
</section>
```

```
<?php
require_once('_page_end.php');
?>
```

### 8.2.9 supermarkets.php

```
<?php
require_once('_page_start.php');
?>
<script
src="http://maps.googleapis.com/maps/api/js?key=AIzaSyAigN5xKZuQcnVe
eNF0tsGMYo2YmHzGJos&sensor=false&language=el"></script>
<script src="map.js"></script>

<script>
```



```
var overhead = 40*60;

function initialize() {
    geocoder = new google.maps.Geocoder();
    map = new
google.maps.Map(document.getElementById("googleMap") ,mapOptions);

    setMarkerAtAddress('<?php echo $_SESSION['address']; ?>',
map, geocoder, true);
}

$(document).ready(function() {
    initClock();
    $("#calcButton").click(findNearestListener);
    google.maps.event.addDomListener(window, 'load',
initialize);
    currentAddress = "<?php echo $_SESSION['address'] ?>";
});

</script>

<h1>Βενζινάδικα</h1>

<div id="googleMap"
style="width:700px;height:500px;margin:auto"></div>

<a id="calcButton" class="centered button-large" href="#">Εύρεση
πλησιέστερου</a>

<div class="response">
    <p></p>
</div>

<h2>Καταχωρημένα super market</h2>

<div id="list" class="list">
<?php
$path = "http://localhost:8984/rest/maps?wrap=yes&query=";
$query = "//supermarkets/supermarket";

$response = file_get_contents($path.$query);
$response= str_replace('rest:', '', $response);

$xml = new SimpleXMLElement($response);
foreach($xml as $item) {
    $supermarket = $item->supermarket;
    echo '<div class="list-item">';
    echo '<h4>'.$supermarket->name."</h4>";
    echo '<p class="address">'.$supermarket->address."</p>";
    echo '<p>'.$supermarket->phone."</p>";
```

```
        echo '<p>Ωράριο: <span class="opening">'.$supermarket->hours-  
>opening.'</span> - <span class="closing">'.$supermarket->hours-  
>closing.'</span></p>';  
        echo '</div>';  
    }  
  
>>  
</div>  
<div class="clear"></div>  
  
<?php  
require_once('_page_end.php');  
>>
```

## 8.3 Αρχεία javascript

### 8.3.1 maps.js

```
var map;  
var geocoder;  
var directionsDisplay = null;  
var directionsService;  
  
var currentAddress;  
  
var addresses = [];  
var opening = [];  
var closing = [];  
var calculated = [];  
  
var currentRoute = -1;  
var bestRoute = -1;  
  
var routes = [];  
  
var mapOptions = {  
    zoom: 17,  
    center: new google.maps.LatLng(35.33909, 25.13366),  
    mapTypeId: google.maps.MapTypeId.ROADMAP  
};  
  
function reset() {  
    addresses = [];  
    opening = [];  
    closing = [];  
    calculated = [];  
  
    currentRoute = -1;  
    bestRoute = -1;  
  
    routes = [];
```

```
}

function setMarkerAtAddress(address, map, geocoder, center) {
    geocoder.geocode({'address': address}, function(results, status)
    {
        if (status === google.maps.GeocoderStatus.OK) {
            if (center === true) {
                map.setCenter(results[0].geometry.location);
            }

            var marker = new google.maps.Marker({
                map: map,
                position: results[0].geometry.location
            });
        }
    });
}

function calcRoute(start, end) {
    var request = {
        origin: start,
        destination: end,
        travelMode: google.maps.TravelMode.DRIVING
    };
    directionsService.route(request, function(response, status) {
        if (status == google.maps.DirectionsStatus.OK) {
            routes.push(response);

            findNearest();
        }
    });
}

function calcRouteToCinema(start, end) {
    var request = {
        origin: start,
        destination: end,
        travelMode: google.maps.TravelMode.DRIVING
    };
    directionsService.route(request, function(response, status) {
        if (status == google.maps.DirectionsStatus.OK) {
            routes.push(response);

            findNearestCinema();
        }
    });
}

function findNearest() {
    currentRoute++;

    if (currentRoute < addresses.length) {
```

```
        calcRoute(currentAddress, addresses[currentRoute]);
    } else {
        var currentTime = hour * 60 * 60 + minute * 60;
        var openingTime, closingTime, calcTime;

        var i;

        for (i = 0; i < routes.length; i++) {
            calculated.push(currentTime +
routes[i].routes[0].legs[0].duration.value + overhead);

            if (calculated[i] > opening[i] && calculated[i] <
closing[i]) {
                if (bestRoute == -1 || calculated[bestRoute] >
calculated[i]) {
                    bestRoute = i;
                }
            }

            if (bestRoute != -1) {
                directionsDisplay.setDirections(routes[bestRoute]);
                $(".response p").html("Στοιχεία διαδρομής: Απόσταση " +
routes[bestRoute].routes[0].legs[0].distance.text + ", διάρκεια " +
routes[bestRoute].routes[0].legs[0].duration.text);
            } else {
                $(".response p").html("Αυτή τη στιγμή δεν υπάρχει
κάποιος έγκυρος προορισμός.");
            }

            reset();
        }
    }
}

function findNearestCinema() {
    currentRoute++;

    if (currentRoute < addresses.length) {
        calcRouteToCinema(currentAddress, addresses[currentRoute]);
    } else {
        var currentTime = hour * 60 * 60 + minute * 60;
        var openingTime, closingTime, calcTime;

        var i;

        for (i = 0; i < routes.length; i++) {
            calculated.push(currentTime +
routes[i].routes[0].legs[0].duration.value + overhead);

            if (calculated[i] < opening[i]) {
                if (bestRoute == -1 || calculated[bestRoute] >
calculated[i]) {
                    bestRoute = i;
                }
            }
        }
    }
}
```

```
        }
    }
    console.log(bestRoute);
}

if (bestRoute != -1) {
    directionsDisplay.setDirections(routes[bestRoute]);
    $(".response p").html("Στοιχεία διαδρομής: Απόσταση " +
routes[bestRoute].routes[0].legs[0].distance.text + ", διάρκεια " +
routes[bestRoute].routes[0].legs[0].duration.text);
} else {
    $(".response p").html("Αυτή τη στιγμή δεν υπάρχει
κάποιος έγκυρος προορισμός.");
}
console.log(bestRoute);
reset();
}

}

function getDestinations(element) {
    $(element + ' .address').each(function() {
        addresses.push(this.innerHTML);
    });
    $(element + ' .opening').each(function() {
        opening.push(timeToSeconds(this.innerHTML));
    });
    $(element + ' .closing').each(function() {
        closing.push(timeToSeconds(this.innerHTML));
    });
}

function findNearestListener() {
    directionsService = new google.maps.DirectionsService();
    if (directionsDisplay != null) {
        directionsDisplay.setMap(null);
        directionsDisplay = null;
    }

    directionsDisplay = new google.maps.DirectionsRenderer();
    directionsDisplay.setMap(map);

    getDestinations(".list");

    findNearest();

    return false;
}

function findNearestCinemaListener() {
    directionsService = new google.maps.DirectionsService();
    if (directionsDisplay != null) {
        directionsDisplay.setMap(null);
        directionsDisplay = null;
    }
}
```

```
    }

    directionsDisplay = new google.maps.DirectionsRenderer();
    directionsDisplay.setMap(map);

    console.log(this.id);

    getDestinations("." + this.id);

    findNearestCinema();

    return false;
}

function timeToSeconds(time) {
    var res = time.split(":");
    var seconds = res[0] * 60 * 60 + res[1] * 60;

    return seconds;
}
/*****          Timer          *****/
var hour = 0;
var minute = 0;

var minutesIncrement = 10;

function initClock() {
    var currentDate = new Date();
    hour = currentDate.getHours();
    minute = Math.round(currentDate.getMinutes() / 10) * 10;

    if (minute == 60) {
        minute = 50;
    }
    timerUpdate();

    $('#timer a.hour.plus').click(hourInc);
    $('#timer a.hour.minus').click(hourDec);

    $('#timer a.minute.plus').click(minuteInc);
    $('#timer a.minute.minus').click(minuteDec);
}

function hourInc() {
    hour++;
    if (hour > 23)
        hour = 0;

    timerUpdate();
}

function hourDec() {
    hour--;
```

```
        if (hour < 0)
            hour = 23;

        timerUpdate();
    }

function minuteInc() {
    minute += minutesIncrement;

    if (minute >= 60) {
        minute = 0;
        hourInc();
    }
    timerUpdate();
}

function minuteDec() {
    minute -= minutesIncrement;

    if (minute < 0) {
        minute = 60 - minutesIncrement;
        hourDec();
    }
    timerUpdate();
}

function timerUpdate() {
    var displayHour = hour;
    if (hour < 10)
        displayHour = "0" + hour;

    var displayMinute = minute;
    if (minute < 10)
        displayMinute = "0" + minute;

    $("#timer span.hour").html(displayHour);
    $("#timer span.minute").html(displayMinute);
}
```