

# **Ανώτατο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης**

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



**Πτυχιακή εργασία**

**Σύστημα παροχής υπηρεσιών δρομολόγησης στους χρήστες του για τις καθημερινές τους δραστηριότητες**

**Κώστας Χρήστος  
Καλαϊτζής Αναστάσιος**

Επιβλέπων καθηγητής : Παπαδάκης Νικόλαος

**Ηράκλειο, 2012**

## Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε καταρχήν τον καθηγητή – καθοδηγητή μας κ. Νίκο Παπαδάκη ο οποίος μας στήριξε όλο αυτόν τον καιρό προκειμένου να πραγματοποιηθεί αυτή η πτυχιακή στο καλύτερο δυνατό χρόνο παρόλες της δυσκολίας και προκειμένου να αποκτήσουμε το μέγιστο των γνώσεων ώστε να μπορούμε να συνεχίσουμε με τα καλύτερα εφόδια για την μετέπειτα πορεία μας. Επίσης θα θέλαμε να ευχαριστήσουμε και τις οικογένειες μας η οποίες μας στήριξαν ψυχολογικά και οικονομικά όλα αυτά τα χρόνια.

# Πίνακας περιεχομένων

<b>Κεφάλαιο 1: Εισαγωγή</b>	<b>4</b>
<b>Κεφάλαιο 2: Παγκόσμιος Ιστός</b>	<b>5</b>
2.1 Η ιστορία του WWW (World Wide Web)	5
2.1 Το Web και το HTTP	5
2.2 Επισκόπηση του HTTP	6
<b>Κεφάλαιο 3: Γλώσσες Προγραμματισμού Και Τεχνολογίες που Χρησιμοποιήθηκαν</b>	<b>9</b>
3.1 HTML	9
3.2 JavaScript	10
3.3 CSS	12
3.4 Php	15
3.5 MySQL	16
3.6 XML	17
3.7 Ajax	19
<b>Κεφάλαιο 4: Google Maps API</b>	<b>21</b>
4.1 Εισαγωγή	21
4.2 Events	22
4.3 Controls	22
4.4 Overlays	23
<b>Κεφάλαιο 5: Υλοποίηση της Εφαρμογής</b>	<b>26</b>
5.1 Περιγραφή της Βάσης Δεδομένων	26
5.2 Αρχιτεκτονική της Βάσης Δεδομένων	26
5.3 Δημιουργία Ιστοσελίδας	30
<b>Βιβλιογραφία</b>	<b>112</b>

# Κεφάλαιο 1

## Εισαγωγή

Η πτυχιακή αυτή έχει σαν θέμα την ανάπτυξη ενός συστήματος το οποίο θα παρέχει δρομολόγηση και καθοδήγηση στους χρήστες του, για διάφορες καθημερινές δραστηριότητες τους. Αναλυτικότερα δίνει την δυνατότητα σε κάποιο χρήστη να προγραμματίσει μια ή περισσότερες δραστηριότητες όπως είναι, το να πάει σε ένα κινηματογράφο για να δει μια ταινία, σε ένα σούπερ μάρκετ για να κάνει ψώνια, ή σε ένα βενζινάδικο για να βάλει βενζίνη. Μέσα από ένα πραγματικό χάρτη (Google Maps) ο χρήστης επιλέγει την ώρα, το σημείο στο οποίο βρίσκεται, τι θέλει να κάνει (να πάει σε κινηματογράφο, σουπερ μάρκετ ή βενζινάδικο) και το σύστημα του εμφανίζει την συντομότερη διαδρομή με αναλυτικές οδηγίες για να πάει στο σημείο ενδιαφέροντος. Υπάρχει η δυνατότητα να αποθηκεύσει στην βάση δεδομένων (MySQL) την δραστηριότητα αυτή και αργότερα να δει συγκεντρωτικά όλες τις δραστηριότητες που έχει σε μία σελίδα. Η εφαρμογή αυτή αναπτύχθηκε ως μια δυναμική ιστοσελίδα στο Web κάνοντας χρήση των τεχνολογιών HTML, CSS, JavaScript, AJAX, PHP, MySQL και Google Maps API v3.

## Κεφάλαιο 2

# Παγκόσμιος Ιστός

### 2.1 Η ιστορία του WWW (World Wide Web)

Ο παγκόσμιος Ιστός είναι μια αρχιτεκτονική για την προσπέλαση διασυνδεδεμένων εγγράφων τα οποία κατανέμονται σε εκατομμύρια μηχανές σε ολόκληρο το Internet. Μέσα σε 10 χρόνια έφτασε από το να είναι ένας τρόπος διανομής δεδομένων φυσικής υψηλής ενέργειας σε μια εφαρμογή η οποία θεωρείται από εκατομμύρια ανθρώπους ότι είναι “Το Internet”.



Ο Ιστός (γνωστός Επίσης ως WWW, **Εικόνα 2.1 To World Wide Web** από τα αρχικά των λέξεων World Wide Web) ξεκίνησε το 1989 στο CERN, το Ευρωπαϊκό κέντρο πυρηνικής έρευνας. Η αρχική πρόταση για έναν ιστό συνδεδεμένων εγγράφων προήλθε από τον φυσικό του CERN Tim Berners-Lee το Μάρτιο του 1989.

Το 1994 το CERN και το MIT υπέγραψαν τη συμφωνία ίδρυσης της **Κοινοπραξίας Παγκόσμιου Ιστού** (World Wide Web Consortium, που μερικές φορές συντομεύεται σε **W3C**), ενός οργανισμού που είναι αφιερωμένος στην περαιτέρω ανάπτυξη του Ιστού, την τυποποίηση των πρωτοκόλλων, και την ενθάρρυνση της διαλειτουργικότητας μεταξύ διαφόρων τοποθεσιών.

### 2.2 Το Web και το HTTP

Μέχρι τις αρχές της δεκαετίας του '90, το Διαδίκτυο χρησιμοποιούνταν κυρίως από ερευνητές, ακαδημαϊκούς και σπουδαστές πανεπιστημίων για σύνδεση σε απομακρυσμένους υπολογιστές, για μεταφορά αρχείων από τοπικούς υπολογιστές σε απομακρυσμένους υπολογιστές και το αντίστροφο, για λήψη και αποστολή ειδήσεων και για λήψη και αποστολή

ηλεκτρονικού ταχυδρομείου. Αν και αυτές οι εφαρμογές ήταν (και συνεχίζουν να είναι) εξαιρετικά χρήσιμες, το Διαδίκτυο ήταν ουσιαστικά άγνωστο έξω από την ακαδημαϊκή και ερευνητική κοινότητα. Κατόπιν, στις αρχές της δεκαετίας του '90, μια νέα κορυφαία εφαρμογή εμφανίστηκε – το World Wide Web [Berners-Lee 1994]. Το Web είναι η εφαρμογή του Διαδικτύου που τράβηξε την προσοχή του μεγάλου κοινού. Άλλαξε δραματικά τον τρόπο με τον οποίο ο κόσμος αλληλεπιδρά μέσα και έξω από το περιβάλλον εργασίας του. Μετέτρεψε ουσιαστικά το Διαδίκτυο από ένα από τα πολλά δίκτυα δεδομένων, στο μοναδικό δίκτυο δεδομένων.

Το στοιχείο εκείνο στο Web που ίσως άσκησε μεγαλύτερη έλξη στους χρήστες ήταν ότι λειτουργεί κατ' απαίτηση. Οι χρήστες δέχονται ότι θέλουν, όταν το θέλουν. Αυτό έρχεται σε αντίθεση με το ραδιόφωνο και την τηλεόραση, που υποχρεώνουν τους χρήστες να «φρυμίζουν τον δέκτη» τους όταν ο πάροχος περιεχομένου κάνει το περιεχόμενο διαθέσιμο. Εκτός του ότι είναι κατ' απαίτηση, το Web πολλά ακόμη θαυμαστά χαρακτηριστικά που οι άνθρωποι λατρεύουν και θαυμάζουν. Είναι πολύ εύκολο για κάθε άτομο να κάνει πληροφορίες διαθέσιμες στο Web. Ο καθένας μπορεί να γίνει εκδότης με ιδιαίτερα χαμηλό κόστος. Οι συνδέσεις και οι μηχανές αναζήτησης μας βοηθούν να κινηθούμε μέσα σε έναν ωκεανό ιστοτόπων. Τα γραφικά ερεθίζουν τις αισθήσεις μας. Οι φόρμες, οι μικροεφαρμογές Java, όπως και πολλές άλλες διατάξεις, μας επιτρέπουν να αλληλεπιδράσουμε με ιστοσελίδες και ιστοτόπους. Ακόμη περισσότερο, το Web παρέχει μια διεπαφή μενού με τεράστιες ποσότητες υλικού ήχου και βίντεο αποθηκευμένες στο Διαδίκτυο – πολυμέσα, τα οποία μπορείτε να προσπελάσετε κατ' απαίτηση.

## 2.2 Επισκόπηση του HTTP



Εικόνα2.2 HTTP

Το **Πρωτόκολλο Μεταφοράς Υπερκειμένου (HyperText Transfer Protocol, HTTP)** το πρωτόκολλο επιπέδου εφαρμογής του Web, είναι ο πυρήνας του Web. Ορίζεται στα [RFC 1945] και [RFC 2616]. Το HTTP υλοποιείται σε δύο προγράμματα: ένα πρόγραμμα πελάτη και ένα πρόγραμμα εξυπηρέτη. Το πρόγραμμα πελάτη και το πρόγραμμα εξυπηρέτη που εκτελούνται σε διαφορετικά τερματικά

συστήματα, συνομιλούν μεταξύ τους ανταλλάσσοντας μηνύματα HTTP. Το HTTP ορίζει την δομή αυτών των μηνυμάτων και το πώς ο πελάτης και ο εξυπηρέτης ανταλλάσσουν τα μηνύματα. Πριν να εξηγήσουμε το HTTP λεπτομερώς, είναι χρήσιμο να μελετήσουμε κάποια από την ορολογία του Web.

Μια **ιστοσελίδα (Web page)**, που επίσης καλείται έγγραφο) αποτελείται από αντικείμενα. Ένα **αντικείμενο (object)** είναι απλώς ένα αρχείο – όπως ένα αρχείο HTML, μια εικόνα JPEG, μια εικόνα GIF, μια μικροεφαρμογή Java, ένα βίντεο κλπ, το οποίο διευθυνσιοδοτείται από ένα URL. Οι περισσότερες ιστοσελίδες αποτελούνται από ένα **αρχείο**

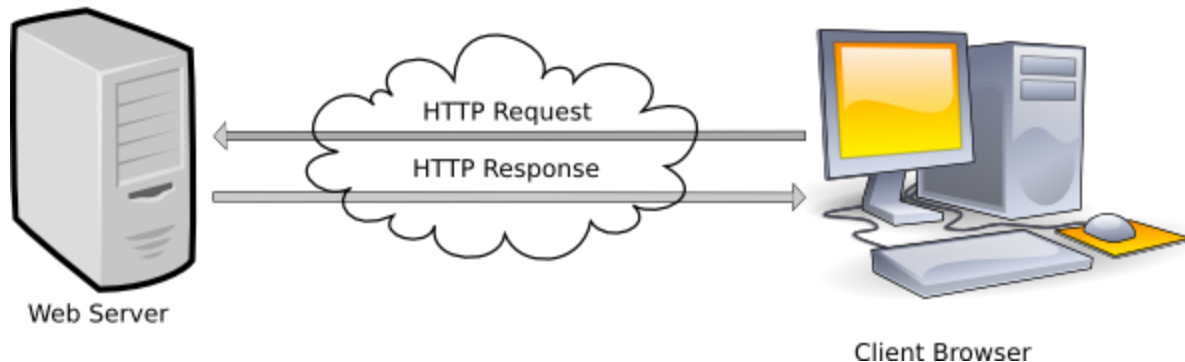
**HTML βάσης (base HTML file)** και αρκετά αναφερόμενα αντικείμενα. Για παράδειγμα, αν μια ιστοσελίδα περιέχει κείμενο HTML και πέντε εικόνες JPEG, τότε η ιστοσελίδα έχει έξι αντικείμενα: το αρχείο HTML βάσεις και τις πέντε εικόνες. Το αρχείο HTML βάσης αναφέρεται σ' αυτά τα αντικείμενα μέσα στην σελίδα με τα URL των αντικειμένων. Κάθε URL έχει δύο συστατικά: το όνομα υπολογιστή του εξυπηρέτη, που φιλοξενεί το αντικείμενο, και το όνομα διαδρομής του αντικειμένου. Για παράδειγμα, το URL

`http://www.someSchool.edu/someDepartment  
/picture.gif`

έχει ως όνομα υπολογιστή το `www.someSchool.edu` και ως όνομα διαδρομής το

`/someDepartment/picture.gif`. Επειδή τα **προγράμματα περιήγησης (web browsers)**, όπως ο Internet Explorer και ο Firefox, υλοποιούν την πλευρά πελάτη του HTTP, στο περιβάλλον του Web, θα χρησιμοποιούμε εναλλακτικά τους όρους πρόγραμμα περιήγησης και πελάτης. Οι εξυπηρέτες του Web (web servers), που υλοποιούν την πλευρά εξυπηρέτη του HTTP, φιλοξενούν αντικείμενα Web, όπου το καθένα διευθυνσιοδοτείται από ένα URL. Δημοφιλές εξυπηρέτες Web περιλαμβάνουν τους Apache και Microsoft Internet Information Server.

Το HTTP ορίζει πώς οι πελάτες Web ζητούν ιστοσελίδες από το Web και πώς οι εξυπηρέτες μεταφέρουν ιστοσελίδες σε πελάτες. Όταν ο χρήστης ζητά μια ιστοσελίδα (π.χ., κάνει κλικ σε μια σύνδεση), το πρόγραμμα περιήγησης στέλνει μηνύματα αίτησης HTTP για τα αντικείμενα της σελίδας, στον εξυπηρέτη. Ο εξυπηρέτης δέχεται τις αιτήσεις και ανταποκρίνεται με μηνύματα απόκρισης HTTP, τα οποία περιέχουν τα αντικείμενα.



**Εικόνα 2.3** Συμπεριφορά αίτησης-απόκρισης του HTTP

Το HTTP χρησιμοποιεί το TCP ως υποκείμενο πρωτόκολλο μεταφοράς (αντί εκτέλεσης επάνω στο UDP). Ο πελάτης HTTP εκκινεί πρώτα μια σύνδεση TCP με τον εξυπηρέτη. Όταν αποκατασταθεί η σύνδεση, οι διεργασίες του προγράμματος περιήγησης και του εξυπηρέτη προσπελαίνουν το TCP μέσω των διεπαφών socket τους. Στη πλευρά πελάτη της διεπαφής socket υπάρχει η πόρτα ανάμεσα στην διεργασία εξυπηρέτη και στην σύνδεση TCP. Ο πελάτης στέλνει μηνύματα αιτήσεων HTTP στην διεπαφή socket και δέχεται μηνύματα

απόκρισης HTTP από την διεπαφή socket. Παρόμοια, ο εξυπηρέτης HTTP δέχεται μηνύματα αιτήσεων από την διεπαφή socket του και στέλνει μηνύματα απόκρισης στην διεπαφή socket του. Όταν ο πελάτης στέλνει ένα μήνυμα στην διεπαφή socket του, το μήνυμα “φεύγει από τα χέρια του πελάτη” και βρίσκεται στα χέρια του TCP. Το TCP παρέχει μία υπηρεσία αξιόπιστης μεταφοράς δεδομένων στο HTTP. Αυτό υπονοεί ότι κάθε μήνυμα αίτησης HTTP που εκπέμπεται από μια διεργασία πελάτη φτάνει τελικά ανέπαφο στον εξυπηρέτη. Παρόμοια, κάθε μήνυμα απόκρισης HTTP που εκπέμπεται από την διεργασία εξυπηρέτη φτάνει τελικά ανέπαφο στον πελάτη. Εδώ βλέπουμε ένα από τα μεγαλύτερα πλεονεκτήματα μιας αρχιτεκτονικής οργανωμένης σε διαδοχικά επίπεδα – το HTTP δεν χρειάζεται να ασχολείται για χαμένα δεδομένα ή για τις λεπτομέρειες περί του πως το TCP επανορθώνει από την απώλεια ή από την αλλαγή σειράς των δεδομένων μέσα στο δίκτυο. Αυτή είναι δουλειά του TCP και των πρωτοκόλλων στα χαμηλότερα επίπεδα της στοίβας πρωτοκόλλων.

Είναι σημαντικό να σημειώσετε ότι ο εξυπηρέτης στέλνει τα αιτούμενα αρχεία σε πελάτες, χωρίς να αποθηκεύει πληροφορίες κατάστασης για τον πελάτη. Αν ένας συγκεκριμένος πελάτης ζητήσει το ίδιο αντικείμενο δυο φορές μέσα σε μία περίοδο λίγων δευτερολέπτων ο εξυπηρέτης δεν αποκρίνεται λέγοντας ότι μόλις έδωσε το αντικείμενο στον πελάτη. Αντί αυτού, ο εξυπηρέτης στέλνει εκ νέου το αντικείμενο, επειδή έχει ξεχάσει τελείως ότι το έστειλε και προηγουμένως. Επειδή ένας εξυπηρέτης HTTP δεν κρατά πληροφορίες για τους πελάτες, το HTTP λέγεται ότι είναι ένα ακαταστασικό πρωτόκολλο (stateless protocol). Ένας εξυπηρέτης Web είναι πάντα ενεργός, με μία σταθερή διεύθυνση IP, και εξυπηρετεί αιτήσεις από πιθανώς εκατομμύρια διαφορετικά προγράμματα περιήγησης.



## Κεφάλαιο 3

# Γλώσσες Προγραμματισμού Και Τεχνολογίες Που Χρησιμοποιήθηκαν

### 3.1 HTML (HyperText Markup Language)

Η HTML (HyperText Markup Language) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων.

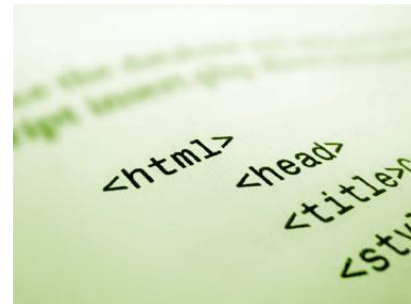
Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες, οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από»

(<html>), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (π.χ. <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 τα συνθέτει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας.

Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους του ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που

μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML.



Εικόνα 3.1

```

<tr>
  <th><label for="contact_name">Name</label></th>
  <td><input type="text" name="contact[name]" id="contact_name" /></td>
</tr>
<tr>
  <th><label for="contact_email">Email</label></th>
  <td>
    <ul class="error_list">
      <li>L'adresse email est invalide.</li>
    </ul>
    <input type="text" name="contact[email]" value="fabien" id="contact_email" />
  </td>
</tr>
<tr>
  <th><label for="contact_subject">Subject</label></th>
  <td>
    <select name="contact[subject]" id="contact_subject">
      <option value="0" selected="selected">Subject A</option>
      <option value="1">Subject B</option>
      <option value="2">Subject C</option>
    </select>
  </td>
</tr>
<tr>
  <th><label for="contact_message">Message</label></th>
  <td>
    <ul class="error_list">
      <li>Le message "foo" est trop court. Il faut au moins 4 caractères.</li>
    </ul>
    <textarea rows="4" cols="30" name="contact[message]" id="contact_message">foo</td>
</tr>

```

Εικόνα 3.2 Δείγμα κώδικα HTML

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

## 3.2 JavaScript

Η **JavaScript** είναι γλώσσα προγραμματισμού η οποία έχει σαν σκοπό την παραγωγή δυναμικού περιεχομένου και την εκτέλεση κώδικα στην πλευρά του πελάτη (client-side) σε ιστοσελίδες. Το πρότυπο της γλώσσας κατά τον οργανισμό τυποποίησης ECMA ονομάζεται **ECMAScript**.

```

view plain copy to clipboard print ?
01. string url = "<a href=\" + someObj.getUrl() + \"\" target=\\
02.
03. // single line comments
04. // second single line
05. override protected void OnLoad(EventArgs e)
06. {
07.     if(Attributes["class"] != null)
08.     {
09.         // _year.CssClass = _month.CssClass = _day.CssClass :
10.     }
11.     base.OnLoad(e);
12. }
13.
14. /*****
15.  ** Multiline block comments
16.  *****/
17.
18. /// <summary>
19. ///     Gets or sets currently selected date of birth or <si

```

Εικόνα 3.3 Κώδικας JavaScript

Η αρχική έκδοση της Javascript βασίστηκε στη σύνταξη στη γλώσσα προγραμματισμού C, αν και έχει εξελιχθεί, ενσωματώνοντας πια χαρακτηριστικά από νεότερες γλώσσες.

The image shows a snippet of HTML code on a dark blue background with faint binary code (0s and 1s) scattered throughout. The code includes various HTML tags and attributes, such as ``, `<td bgcolor="#b0b190" width="1"></td>`, `<table border="0" cellpadding="1" width="470">`, `<tbody><tr>`, `<td rowspan="3" align="right" bgcolor="#e9ead6" valign="top" width="130">`, `<target="_blank"></a>`, `</td>`, `<tr bgcolor="#e9ead6">`, `<td colspan="5" align="center">`, `<table width="100%">`, `<tbody><tr>`, and `<td rowspan="2" class="h"`.

Εικόνα 3.4

Αρχικά χρησιμοποιήθηκε για προγραμματισμό από την πλευρά του πελάτη (client), που ήταν ο φυλλομετρητής (browser) του χρήστη, και χαρακτηρίστηκε σαν *client-side γλώσσα προγραμματισμού*. Αυτό σημαίνει ότι η επεξεργασία του κώδικα Javascript και η παραγωγή του τελικού περιεχομένου HTML δεν πραγματοποιείται στο διακομιστή, αλλά στο πρόγραμμα περιήγησης των επισκεπτών, ενώ μπορεί να ενσωματωθεί σε στατικές

σελίδες HTML. Αντίθετα, άλλες γλώσσες όπως η PHP εκτελούνται στο διακομιστή (*server-side γλώσσες προγραμματισμού*).

Παρά την ευρεία χρήση της Javascript για συγγραφή προγραμμάτων σε περιβάλλον φυλλομετρητή, αξίζει να σημειωθεί ότι από την αρχή χρησιμοποιήθηκε και για τη συγγραφή κώδικα από την πλευρά του διακομιστή, από την ίδια τη Netscape στο προϊόν LiveWire, με μικρή επιτυχία. Η χρήση της Javascript στο διακομιστή εμφανίζεται πάλι σήμερα, με τη διάδοση του Node.js, ενός μοντέλου προγραμματισμού βασισμένο στα γεγονότα (events).

Ο κώδικας Javascript μιας σελίδας περικλείεται από τις ετικέτες της HTML `<script type="text/javascript">` και `</script>`.

Για παράδειγμα, ο ακόλουθος κώδικας Javascript εμφανίζει ένα πλαίσιο διαλόγου με το κείμενο "Γεια σου, κόσμε!":

```
<script type="text/javascript">
alert('Γεια σου, κόσμε!');
</script>
```

Αν ο κώδικας Javascript περιέχει περισσότερες από μία εντολές, αυτές θα πρέπει να διαχωριστούν μεταξύ τους με το χαρακτήρα του ελληνικού ερωτηματικού ';' (δηλαδή της λατινικής άνω τελείας). Η χρήση του χαρακτήρα αυτού για την τελευταία εντολή δεν είναι απαραίτητη. Η διαχώριση των εντολών στους νεότερους φυλλομετρητές (browsers) δεν είναι απαραίτητη.

Μια άλλη βασική εντολή, η `window.prompt("μήνυμα προς το χρήστη")`, ζητάει από το χρήστη να συμπληρώσει ένα κομμάτι μιας αίτησης απευθείας ώστε τα δεδομένα να χρησιμοποιηθούν σαν κείμενο:

```
<script>
var FIRSTvariable = window.prompt("PLEASE FILL IN YOUR NAME")
alert("Your name is " + FIRSTvariable + ".")
</script>
```

### 3.3 CSS (Cascading Style Sheets)

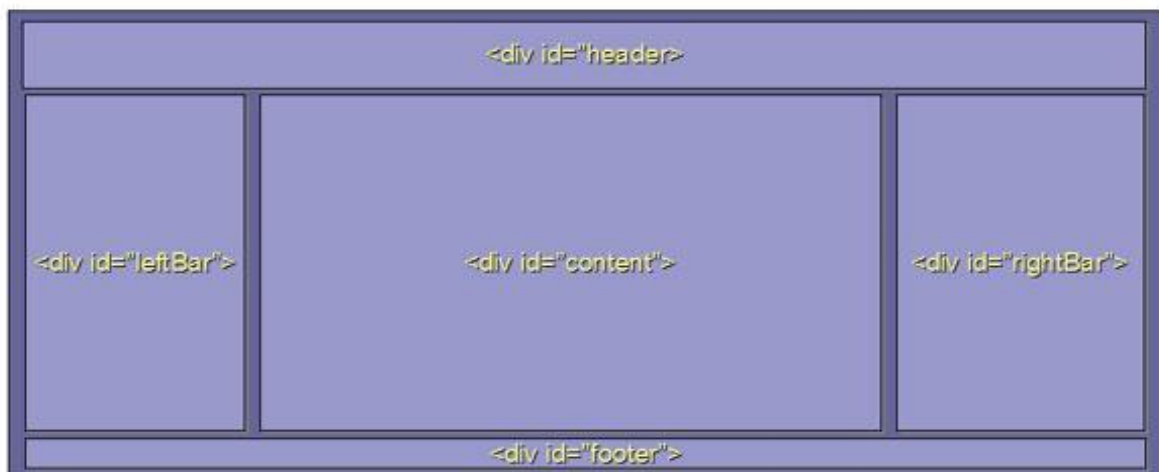
Το **CSS** (*Cascading Style Sheets-Διαδοχικά Φύλλα Στυλ*) ή ( αλληλουχία φύλλων στυλ ) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Το CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στιλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει

περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την html. Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα η χρήση της CSS κρίνεται ως απαραίτητη.

Για ένα έγγραφο πχ xhtml 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 xhtml στοιχείο θα υπάρχουν παραπάνω από μια δηλώσεις που πιθανόν να είναι συγκρουόμενες πρότυπη css για να επιλύσει παρόμοιες συγκρούσεις έχει καθορίσει



**Εικόνα 3.5 Fixed Layout**

μια αλληλουχία-σειρά στην οποία θα μπουν αυτές οι δηλώσεις και με βάση την οποία θα επιλεγεί πχ η δήλωση που είναι πρώτη στη σειρά.

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

1. 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).
2. Ταξινόμησε με βάση τη σημασία (κανονική ή σημαντική) και προέλευση (συγγραφέας, χρήστη ή πλοηγός χρήστη). Με αύξουσα σειρά προτεραιότητας:
  1. Δηλώσεις πλοηγού χρήστη
  2. Κανονικές δηλώσεις χρήστη
  3. Κανονικές δηλώσεις συγγραφέα

4. Σημαντικές δηλώσεις συγγραφέα
5. Σημαντικές δηλώσεις χρήστη
3. 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.
4. 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 .

Αφού λοιπόν προκύψει μια σειρά-αλληλουχία κανόνων εμφάνισης που αφορούν το ίδιο στοιχείο θα επιλεγεί προς εφαρμογή (για την αποφυγή συγκρούσεων ) η δήλωση που θα είναι τελευταία στην σειρά που αναλύθηκε πιο πάνω.



Εικόνα 3.6 CSS Layout

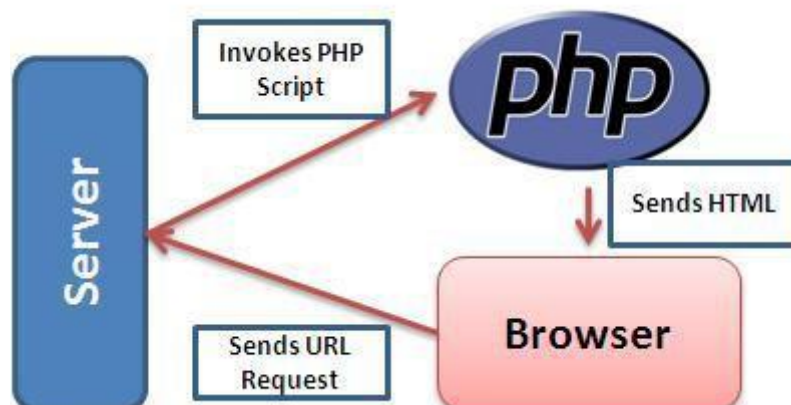
### 3.4 PHP

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.



Εικόνα 3.7 Το λογότυπο της PHP

Η ιστορία της PHP ξεκινά από το 1994, όταν ένας φοιτητής, ο Rasmus Lerdorf δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl ένα απλό script με όνομα `php.cgi`, για προσωπική χρήση. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Αργότερα αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. Το 1997 η PHP/FI έφθασε στην έκδοση 2.0, βασιζόμενη αυτή τη φορά στη γλώσσα C και αριθμώντας περισσότερους από 50.000 ιστότοπους που τη χρησιμοποιούσαν, ενώ αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρκετά στην PHP/FI 2.0. Έτσι η PHP έφθασε στην έκδοση 3.0 η οποία θύμιζε περισσότερο τη σημερινή μορφή της. Στη συνέχεια, οι Zeev και Andi δημιούργησαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP. Ακολούθησε το 1998 η έκδοση 4 της PHP, τον Ιούλιο του 2004 διατέθηκε η έκδοση 5, ενώ αυτή τη στιγμή έχουν ήδη διατεθεί και οι πρώτες δοκιμαστικές εκδόσεις της επερχόμενης PHP 6, για οποιονδήποτε προγραμματιστή θέλει να τη χρησιμοποιήσει. Οι περισσότεροι ιστότοποι επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της PHP.



Εικόνα 3.8 Η λειτουργία της PHP

### 3.5 MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Έλαβε το όνομά της από την κόρη του Μόντυ Βιντένιους, τη Μάι (αγγλ. My). Το πρόγραμμα τρέχει έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων.



Εικόνα 3.9 Το λογότυπο της MySQL

Ο κωδικός του εγχειρήματος είναι διαθέσιμος μέσω της GNU General Public License, καθώς και μέσω ορισμένων ιδιόκτητων συμφωνιών. Ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική MySQL AB, η οποία σήμερα ανήκει στην Oracle.

```
C:\windows\system32\cmd.exe - mysql -u root
Your MySQL connection id is 46
Server version: 5.0.67-community-nt MySQL Community Edition (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use wordpress;
Database changed
mysql> show tables;
+-----+
| Tables_in_wordpress |
+-----+
| wp_comments          |
| wp_links             |
| wp_options           |
| wp_postmeta          |
| wp_posts             |
| wp_term_relationships|
| wp_term_taxonomy    |
| wp_terms             |
| wp_usermeta          |
| wp_users             |
+-----+
10 rows in set (0.00 sec)

mysql>
```

Εικόνα 3.10 Παράθυρο γραμμής εντολών



## 3.6 XML

Η XML (Extensible Markup Language) 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, συμπεριλαμβανομένων του RSS, του SOAP και της XHTML.

Το περιεχόμενο αυτής της ενότητας, βασίζεται στην προδιαγραφή XML 1.0. Δεν αποτελεί μία πλήρη λίστα όλων των όρων που υπάρχουν στη γλώσσα XML. Είναι μία εισαγωγή στα βασικά στοιχεία, που συναντώνται στην καθημερινή της χρήση.

### Χαρακτήρας Unicode

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

### 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*.

### 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.

## 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/>.

## 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/>.

## 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".

## 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"?>
```

## Παράδειγμα

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`.

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.

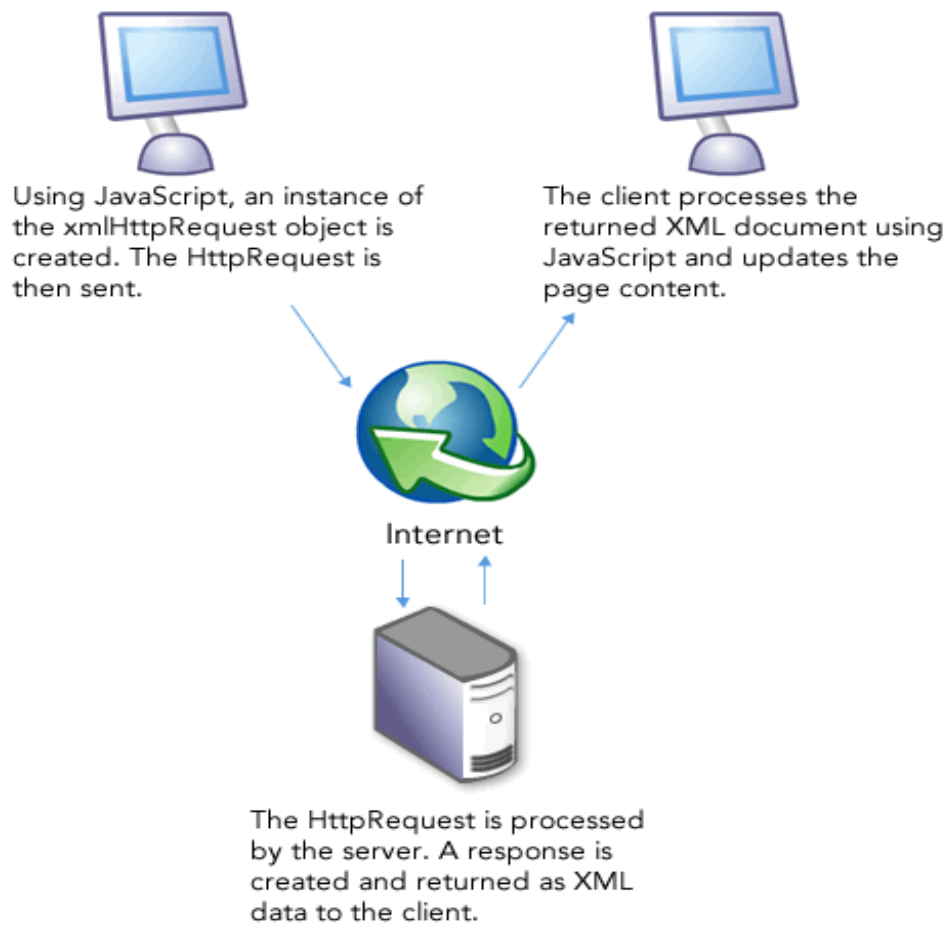
Η XML παρέχει κάποιες διευκολύνσεις για την ταυτοποίηση της κωδικοποίησης των χαρακτήρων Unicode που απαρτίζουν ένα κείμενο και για την απεικόνιση χαρακτήρων που, για τον έναν ή τον άλλο λόγο, δεν μπορούν να χρησιμοποιηθούν ευθέως.

## 3.7 AJAX (Asynchronous Javascript Xml)

Η τεχνολογία AJAX (Asynchronous JavaScript And XML) δεν είναι μια νέα γλώσσα προγραμματισμού, αλλά απλά μία νέα τεχνική για τη δημιουργία πιο γρήγορων και πιο φιλικών για το χρήστη διαδικτυακών εφαρμογών. Η τεχνολογία AJAX βασίζεται σε JavaScript και σε κλήσεις HTTP. Η τεχνολογία AJAX χρησιμοποιεί JavaScript για την αποστολή και λήψη δεδομένων μεταξύ ενός φυλλομετρητή (web browser) και του εξυπηρετητή (web server). Η τεχνική AJAX κάνει τις ιστοσελίδες πιο διαδραστικές επιτρέποντας την αποστολή δεδομένων στο παρασκήνιο χωρίς να χρειαστεί ο χρήστης να μεταφορτώνει μία ιστοσελίδα κάθε φορά που κάνει κάτι στην σελίδα.

Η τεχνολογία AJAX εκτελείται στους υπολογιστές που υπάρχει ο φυλλομετρητής. Χρησιμοποιεί ασύγχρονη μεταφορά δεδομένων (HTTP κλήσεις) μεταξύ του φυλλομετρητή και του κεντρικού εξυπηρετητή, επιτρέποντας στις σελίδες web να ζητούν μικρές πληροφορίες από τον εξυπηρετητή αντί για πλήρεις σελίδες. Η τεχνολογία AJAX είναι πολύ

χρήσιμη μιας και επιτρέπει της διαδικτυακές εφαρμογές να είναι μικρές, να φορτώνονται γρήγορα και να είναι πολύ φιλικές για το τελικό χρήστη. Θα πρέπει να σημειωθεί πως η τεχνολογία AJAX είναι κυρίως τεχνολογία που υποστηρίζεται στο φυλλομετρητή (web browser) και όχι τεχνολογία του εξυπηρετητή.



**Εικόνα 3.11** Τρόπος Λειτουργίας της Τεχνολογίας AJAX

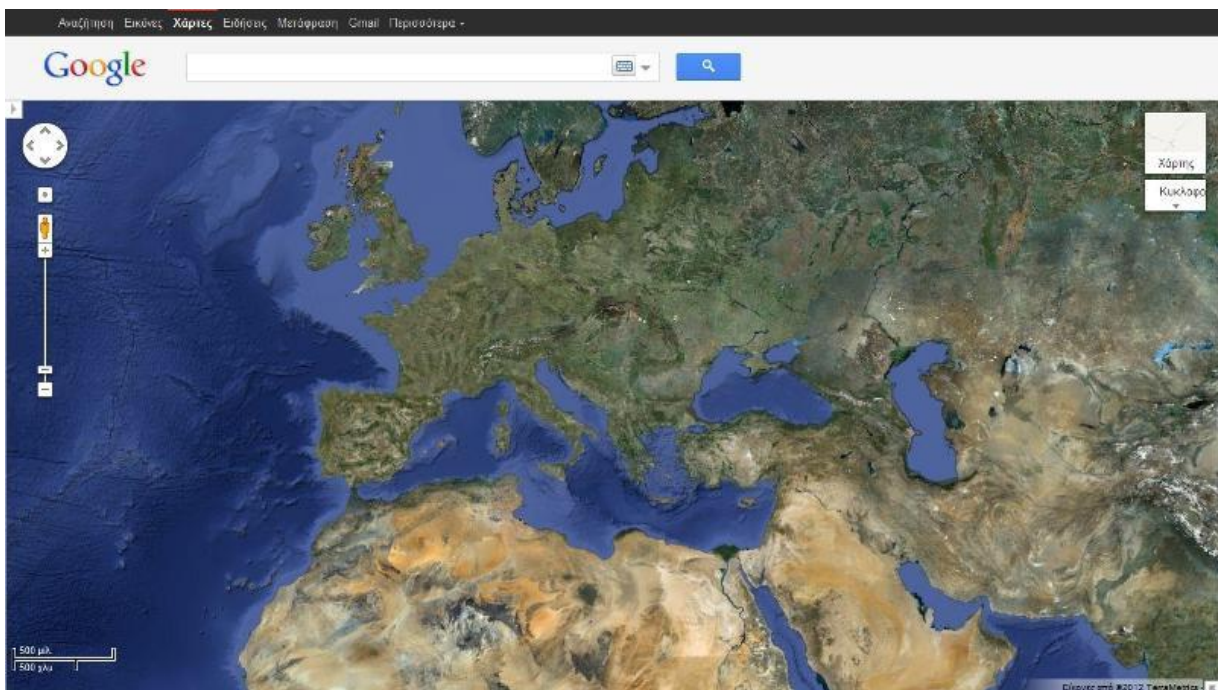
## Κεφάλαιο 4

# Google Maps API

### 4.1 Εισαγωγή

Το Google Maps API είναι μια διεπαφή προγραμματισμού εφαρμογών (Application Programming Interface – API) της Google το οποίο δίνει τη δυνατότητα ενσωμάτωσης ενός δυναμικού χάρτη Google σε ιστοσελίδες (web σελίδες) με τη χρήση της javascript. Το API παρέχει έναν αριθμό εργαλείων για το χειρισμό χαρτών και την προσθήκη περιεχομένου σε αυτούς μέσω διαφόρων υπηρεσιών, όπως και στην επίσημη ιστοσελίδα του Google Maps (<http://maps.google.com>), επιτρέποντας έτσι τη δημιουργία εύρωστων εφαρμογών που εκμεταλλεύονται γεωγραφικές πληροφορίες.

Το Google Maps API είναι μια δωρεάν υπηρεσία, διαθέσιμη για οποιοδήποτε ιστότοπο (web site) που είναι ελεύθερος για το κοινό. Το μόνο που χρειάζεται είναι ένα κλειδί API (API key) που μπορεί να ζητηθεί εφόσον υπάρχει λογαριασμός στη Google και αυτομάτως συνδέεται με το λογαριασμό. Κάθε κλειδί είναι έγκυρο μόνο για ένα συγκεκριμένο όνομα τομέα.



Εικόνα 4.1 Google Maps

Βασικό στοιχείο οποιασδήποτε Google Maps API εφαρμογής είναι ο ίδιος ο χάρτης, ο

οποίος φορτώνεται, αρχικοποιείται με συγκεκριμένες διαστάσεις (ύψος και πλάτος), κέντρο (γεωγραφικό πλάτος, γεωγραφικό μήκος), επίπεδο εστίασης και τύπο (κανονικό, δορυφορικό, υβριδικό). Στη συνέχεια είναι δυνατή η αλληλεπίδραση με το χάρτη μέσω διαφόρων μηχανισμών όπως είναι τα events, τα controls, τα overlays αλλά και συναρτήσεις επεξεργασίας των χαρακτηριστικών του, τα οποία αναλύονται παρακάτω:

## 4.2 Γεγονότα (Events)

Το Google Maps API εμπεριέχει δηλώσεις γεγονότων (events) τα οποία εμφανίζονται μέσω των φυλλομετρητών ιστοσελίδων (web browsers) πάνω σε αντικείμενα του Google Maps. Ας σημειωθεί ωστόσο ότι τα γεγονότα αυτά είναι διαφορετικά από τα standard γεγονότα πάνω στο Μοντέλο Αντικειμένου Εγγράφου (Document Object Model – DOM), το οποίο είναι μια προγραμματιστική διεπαφή που επιτρέπει σε προγράμματα και εκτελέσιμα σενάρια δράσης (scripts) την ενημέρωση του περιεχομένου και του στυλ όλου του εγγράφου με δυναμικό τρόπο. Έτσι το Google Maps API δημιουργεί και διαχειρίζεται δικά του γεγονότα ενώ το Μοντέλο Αντικειμένου Εγγράφου δημιουργεί άλλα γεγονότα και τα επεξεργάζεται σύμφωνα με το μοντέλο αντικειμένου που χρησιμοποιεί ο συγκεκριμένος φυλλομετρητής. Επιπλέον, το Google Maps API έχει το πλεονέκτημα ότι προσφέρει μηχανισμούς που ακούνε (listen) και απαντάνε (respond) στα DOM γεγονότα ανεξάρτητα από τις ιδιαιτερότητες του κάθε φυλλομετρητή. Παραδείγματα γεγονότων του Google Maps API είναι το κλικ του ποντικιού, το διπλό κλικ, η κίνηση του ποντικιού τα οποία σηματοδοτούν την έναρξη κάποιων διαδικασιών που περιγράφονται στον κώδικα javascript. Εξάλλου η javascript είναι μια scripting γλώσσα που «καθοδηγεί γεγονότα» (“event driven”) εξελισσόμενα στους φυλλομετρητές.

## 4.3 Controls

Όπως μπορεί να διαπιστώσει κανείς και στη σελίδα του Google Maps (<http://maps.google.com>) υπάρχουν διάφορα στοιχεία στη διεπιφάνεια με το χρήστη (user interface) που επιτρέπουν την αλληλεπίδρασή του χρήστη με το χάρτη. Τα στοιχεία αυτά είναι γνωστά ως ρυθμιστές (controls) και μπορούν να περιληφθούν στις Google Maps εφαρμογές σε οποιαδήποτε σημείο πάνω στο χάρτη. Τα κυριότερα είναι:

- `GLatLngControl3D`: Τοποθετεί ένα μεγάλο τρισδιάστατο κουμπί ελέγχου εστίασης και κίνησης του χάρτη, όπως χρησιμοποιείται τώρα στη σελίδα του Google Maps. Εμφανίζεται στην πάνω αριστερή γωνία του χάρτη αυτόματα.

- **GLargeMapControl**: Τοποθετεί ένα μεγάλο απλό κουμπί ελέγχου εστίασης και κίνησης του χάρτη. Εμφανίζεται στην πάνω αριστερή γωνία του χάρτη αυτόματα.
- **GSmallMapControl** : Τοποθετεί ένα μικρό κουμπί ελέγχου εστίασης και κίνησης του χάρτη. Εμφανίζεται στην πάνω αριστερή γωνία του χάρτη αυτόματα.
- **GSmallZoomControl3D**: Τοποθετεί ένα μικρό τρισδιάστατο κουμπί ελέγχου εστίασης (χωρίς έλεγχο κίνησης) του χάρτη. Εμφανίζεται στην πάνω αριστερή γωνία του χάρτη αυτόματα.
- **GSmallZoomControl**: Τοποθετεί ένα μικρό κουμπί ελέγχου εστίασης (χωρίς έλεγχο κίνησης) του χάρτη που χρησιμοποιείται στα μικρά παράθυρα που εμφανίζονται στο χάρτη Google για επεξήγηση των οδικών οδηγιών.
- **GScaleControl**: Τοποθετεί μια κλίμακα με στάδια μεγέθυνσης του χάρτη.
- **GMapTypeControl**: Τοποθετεί κουμπιά που επιτρέπουν στο χρήστη να διαλέγει μεταξύ τύπων του χάρτη (όπως δορυφορικός, εδαφικός κ.ά.)
- **GHierarchicalMapTypeControl**: Τοποθετεί κουμπιά επιλογών (checkboxes) για ιεραρχική προτίμηση των τύπων του χάρτη.
- **GOverviewMapControl**: Τοποθετεί μια πτυσσόμενη επισκόπηση του χάρτη στη γωνία της οθόνης.
- **GNavLabelControl**: Τοποθετεί μια δυναμική ετικέτα που επισημαίνει τη «διεύθυνση» της συγκεκριμένης περιοχής χάρτη ανάλογα με το επίπεδο εστίασης.

Επιπλέον υπάρχει η δυνατότητα δημιουργίας ειδικών ρυθμιστικών στοιχείων από τον προγραμματιστή.

#### **4.4 Overlays**

Τα overlays είναι αντικείμενα πάνω στο χάρτη που αντιστοιχούν σε συγκεκριμένες συντεταγμένες και γι' αυτό μετακινούνται ανάλογα όταν μετακινείται ή αλλάζει εστίαση ο χάρτης. Χρησιμοποιούνται για το σχεδιασμό και την επίδειξη σημείων, γραμμών ή

ολόκληρων περιοχών. Το Google Maps API διαθέτει πολλά είδη overlays, τα οποία αναλύονται παρακάτω:

- Σημεία

Αναπαριστώνται με τη χρήση δεικτών (markers) πάνω στο χάρτη και συχνά προβάλλουν μια σχετική εικόνα στο σημείο που επιδεικνύουν. Οι markers είναι σχεδιασμένοι να είναι διαδραστικοί, δηλαδή επάνω τους πραγματοποιούνται γεγονότα, όπως ένα κλικ ή δεξί κλικ, και συχνά προβάλεται παράλληλα με αυτούς ένα παράθυρο πληροφοριών για το σημείο. Μια επιπλέον ιδιότητα τους είναι ότι μπορεί να είναι συρόμενοι (draggable) από ένα σημείο σε άλλο στο χάρτη. Επίσης δυνατή είναι η επεξεργασία και αλλαγή της εικόνας (icon) που αντιστοιχεί σε κάθε δείκτη. Η εικόνα αποτελείται από μια εικόνα στο προσκήνιο και μια άλλη ως σκία τοποθετημένες σε συγκεκριμένες θέσεις μεταξύ τους. Παρότι οι δείκτες έχουν μεγάλη χρησιμότητα, η προσθήκη πολλών από αυτούς σε ένα χάρτη Google μπορεί να έχει αρνητική επίδραση στην ταχύτητα απεικόνισης του χάρτη (map rendering) ειδικά σε συγκεκριμένα επίπεδα εστίασης. Λύση σε αυτό δίνει ο Διαχειριστής Δεικτών (Marker Manager), ο οποίος έχει την ικανότητα να παρακολουθεί ποιοί δείκτες είναι ορατοί σε ποιά επίπεδα εστίασης και να στέλνει μόνο τους απαραίτητους στο χάρτη για σχεδιαστικούς σκοπούς. Δηλαδή προσθέτει ή αφαιρεί δυναμικά δείκτες από το χάρτη με αποτέλεσμα την αύξηση της ταχύτητας απεικόνισης του χάρτη και της μείωσης των άχρηστων οπτικών στοιχείων.

- Γραμμές

Αναπαριστώνται με τη χρήση των polylines (που ουσιαστικά είναι μια ακολουθία συνδεδεμένων σημείων) πάνω στο χάρτη, για τις οποίες μπορεί να καθοριστεί χρώμα, πάχος και επίπεδο διαφάνειας. Χωρίζονται σε τρεις κατηγορίες: τις γραμμικές (drawing), τις γεωδαιτικές (geodesic) και τις κωδικοποιημένες (encoded). Οι γραμμικές polylines εμφανίζονται ως ευθύγραμμα τμήματα στο χάρτη και για την απεικόνισή τους το Google Maps API εκμεταλλεύεται πιθανές σχεδιαστικές ικανότητες του φυλλομετρητή, αλλιώς γίνεται αίτηση για μια εικόνα polyline από τους εξυπηρετητές (servers) της Google. Οι γεωδαιτικές polylines εμφανίζονται ως καμπυλωτές γραμμές λαμβάνοντας υπόψιν τους την καμπυλότητα της γης. Οι κωδικοποιημένες polylines είναι συνήθως μεγαλύτερες και πιο περίπλοκες από τις υπόλοιπες και αποτελούνται από πολλά ευθύγραμμα τμήματα που συνδέονται με τυχαίο τρόπο μεταξύ τους. Γι' αυτό το λόγο απαιτούν περισσότερη μνήμη και χρόνο για να απεικονισθούν. Έχουν όμως το πλεονέκτημα ότι μπορούν να καθορίζουν πόσο λεπτομερής θα είναι μια polyline σε ένα δοσμένο επίπεδο εστίασης καθώς δίνουν τη δυνατότητα να αγνοούνται τμήματα της γραμμής σε συγκεκριμένες ομάδες επιπέδων εστίασης.

- Περιοχές

Αναπαριστώνται με τη χρήση είτε πολυγώνων (polygons) πάνω στο χάρτη, αν η περιοχή έχει τη μορφή αυθαίρετου σχήματος, είτε με τη χρήση ground overlays, αν η περιοχή έχει τη μορφή ορθογώνιου σχήματος. Τα πολύγωνα είναι παρόμοια με τις polylines αφού και



αυτά είναι μια ακολουθία σημείων, προσθέτοντας ότι σχηματίζουν κύκλο και μπορούν να πάρουν οποιαδήποτε μορφή. Τα ground overlays είναι συχνά χρήσιμα για περιοχές που τα άκρα τους στοχεύουν κάθετα σε σημεία στο χάρτη.

- Ο ίδιος ο χάρτης

Αναπαρίσταται με τη χρήση του tile overlay, κάτι το οποίο μπορεί να τροποποιηθεί μέσω του Google Maps API. Τα διαθέσιμα tiles δεν καλύπτουν όλες τις περιοχές σε όλα τα επίπεδα εστίασης. Για παράδειγμα πολλές περιοχές στον Ειρηνικό Ωκεανό δεν εμφανίζονται σε υψηλά επίπεδα εστίασης, ενώ το Μανχάταν προσφέρει πολύ λεπτομερείς εικόνες στον δορυφορικό τύπο χάρτη. Αξίζει να σημειωθεί ότι στο μικρότερο επίπεδο εστίασης (επίπεδο 0), ένα tile αναπαριστά ολόκληρη τη γη. Αν N είναι το επίπεδο εστίασης, τότε το Google Maps διαιρεί το tile ολόκληρης της γης σε  $4^N$  tiles και προβάλλει το κατάλληλο.

- Το παράθυρο πληροφοριών (info window)

Αποτελεί ένα ειδικό overlay. Προστίθεται στο χάρτη αυτόματα και σε κάθε χάρτη αντιστοιχεί ένα μόνο αντικείμενο παραθύρου πληροφοριών.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <title>Google Maps JavaScript API Example</title>

  <script
src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAAjxgax2HtvdoQYMOBWLHs9hRUX
IyXixIDfW7VAZYX_vCT2lirwBRM-L1XfLEQ-FDSu4xxHNP4jD4LFg"
type="text/javascript"></script>

  <script type="text/javascript">

    //<![CDATA[
    function initializeMyMap() {
      if (GBrowserIsCompatible()) {
        var myMap = new GMap2(document.getElementById("myMapContainer"));
        myMap.setCenter(new GLatLng(37.4419, -122.1419), 13);
        var myPlacemark = new GMarker(new GLatLng(37.4419, -122.1419));
        myMap.addOverlay(myPlacemark);
      }
      else {
        alert("Sorry, the Google Maps API is not compatible with this browser.");
      }
    }
    //]]>
  </script>

</head>
<body onload="initializeMyMap()" onunload="GUnload()">
  <div id="myMapContainer" style="width: 500px; height: 300px">
  </div>
  <noscript>
  <p>
    JavaScript must be enabled in order to view Google Maps<sup>®</sup>. However, it seems
    JavaScript is either disabled or not supported by your web browser. To view Google
    Maps<sup>®</sup>, enable JavaScript by changing your browser settings and refresh the web
    page.</p>
  </noscript>
</body>
</html>
```



Εικόνα 4.2 Κώδικας για την τοποθέτηση Marker στον χάρτη.

## Κεφάλαιο 5

# Υλοποίηση της Εφαρμογής

### 5.1 Περιγραφή της Βάσης Δεδομένων

Για την δημιουργία της εφαρμογής είναι απαραίτητη η δημιουργία μιας βάσης δεδομένων στην οποία θα αποθηκεύονται όλα τα δεδομένα που χρειάζονται για την σωστή λειτουργία της. Η βάση δεδομένων της εφαρμογής ονομάζεται “myDatabase” και αποτελείται από τρεις πίνακες στους οποίους αποθηκεύονται όλα τα δεδομένα.

### 5.2 Αρχιτεκτονική της Βάση Δεδομένων

#### Σύνδεση με τη βάση

Η σύνδεση με τη βάση δεδομένων γίνεται με χρήση εντολών της PHP και συγκεκριμένα με την κλήση της συνάρτησης `mysql_connect()`, δίνοντας τα σχετικά ορίσματα.

<?

```
$mysql_hostname = "localhost"; // server to connect to.
```

```
$mysql_user = "root"; // mysql username to access the database with.
```

```
$mysql_password = ""; // mysql password to access the database with.
```

```
$mysql_database = "mydatabase"; // the name of the database.
```

```
$mysql_tablename = "users";
```

```
$bd = mysql_connect($mysql_hostname, $mysql_user, $mysql_password) or die("cannot connect");
```

```
mysql_select_db($mysql_database, $bd) or die("cannot select DB");
```

?>

#### Σχέσεις πινάκων της βάσης

Το storage engine της βάσης δεδομένων μας είναι το MyISAM το οποίο δεν υποστηρίζει σχέσεις πινάκων όπως το INNODB. Το INNODB ενδείκνυται για εφαρμογές τέτοιου είδους, καθώς είναι πιο ασφαλές επειδή επιτρέπει την δημιουργία σχέσεων μεταξύ των πινάκων, από την άλλη όμως

υστερεί σε ταχύτητα.

Έτσι, επιλέξαμε MySQL γιατί θέλαμε το σύστημά μας να είναι πιο γρήγορο παρόλο που είναι πιο επικίνδυνο για τη δημιουργία λαθών. Στην περίπτωσή μας όμως, αυτό ήταν εφικτό λόγω του μικρού αριθμού πινάκων στη βάση του συστήματος.

Η βάση δεδομένων αποτελείται από τους παρακάτω πίνακες:

- **Πίνακας Users**

Αυτός είναι ο Πίνακας **Users** έχει πρωτεύον κλειδί το πεδίο **id** και χρησιμοποιείτε για την εγγραφή των χρηστών (Users) στο σύστημα. Σε αυτόν τον πίνακα αποθηκεύονται προσωπικά στοιχεία για τους χρήστες όπως είναι το όνομα (**name**), το επίθετο (**lastname**), η διεύθυνση (**address**), το τηλέφωνο (**phone**), ο αριθμός της πιστωτικής κάρτας (**creditcard**), το **username** και ο κωδικός πρόσβασης (**password**) τους.

#	Στήλη	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Πρόσθετα
<input type="checkbox"/>	1 <b>id</b>	int(11)			Όχι	Καμία	AUTO_INCREMENT
<input type="checkbox"/>	2 <b>name</b>	varchar(60)	latin1_swedish_ci		Όχι	Καμία	
<input type="checkbox"/>	3 <b>address</b>	varchar(80)	latin1_swedish_ci		Όχι	Καμία	
<input type="checkbox"/>	4 <b>lat</b>	float(10,6)			Όχι	Καμία	
<input type="checkbox"/>	5 <b>lng</b>	float(10,6)			Όχι	Καμία	
<input type="checkbox"/>	6 <b>type</b>	varchar(30)	latin1_swedish_ci		Όχι	Καμία	
<input type="checkbox"/>	7 <b>open_time</b>	time			Όχι	06:00:00	
<input type="checkbox"/>	8 <b>close_time</b>	time			Όχι	20:00:00	

id	name	lastname	address	phone	creditcard	username	password
1	Christos	Kostas	Greece	2381011111	5588320123456789	Chris	d8578edf8458ce06fbc5bb76a58c5ca4
2	Tasos	Kalaitzis	Greece	6900000000	4555878975895242	Tasos	006d2143154327a64d86a264aea225f3
3	User	User	Greece	2101111111	4433320123456790	User	9eecb7db59d16c80417c72d1e1f4fbf1

- **Πίνακας Markers**

Αυτός είναι ο Πίνακας **Markers** έχει σαν πρωτεύον κλειδί το πεδίο **id** και χρησιμοποιείτε για την αποθήκευση σημείων (Markers) στα οποία βρίσκονται Βενζινάδικα, Σουπερ Μάρκετ και Κινηματογράφοι. Στον πίνακα αυτό αποθηκεύονται: το όνομα (**name**) του σημείου ενδιαφέροντος, η διεύθυνση (**address**) του, οι συντεταγμένες latitude (**lat**), longitude (**lng**) επάνω στο χάρτη στις οποίες βρίσκεται, το είδος (**type**) του σημείου ενδιαφέροντος (βενζινάδικο, σουπερ μάρκετ, κινηματογράφος), η ώρα που ανοίγει (**open\_time**) και η ώρα που κλείνει (**close\_time**).

#	Στήλη	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Πρόσθετα
1	<b>id</b>	int(11)			Όχι	Καμία	AUTO_INCREMENT
2	<b>name</b>	varchar(60)	latin1_swedish_ci		Όχι	Καμία	
3	<b>address</b>	varchar(80)	latin1_swedish_ci		Όχι	Καμία	
4	<b>lat</b>	float(10,6)			Όχι	Καμία	
5	<b>lng</b>	float(10,6)			Όχι	Καμία	
6	<b>type</b>	varchar(30)	latin1_swedish_ci		Όχι	Καμία	
7	<b>open_time</b>	time			Όχι	06:00:00	
8	<b>close_time</b>	time			Όχι	20:00:00	

id	name	address	lat	lng	type	open_time	close_time
1	Gas Station 1	1521 1st Ave, Seattle, WA	47.608940	-122.340141	Gas_station	06:00:00	13:05:00
2	Gas Station 2	2222 2nd Ave, Seattle, WA	47.613590	-122.344391	Gas_station	12:00:00	20:00:00
3	Gas Station 3	14 Mercer St, Seattle, WA	47.624561	-122.356445	Gas_station	06:00:00	20:00:00
4	Gas Station 4	1225 1st Ave, Seattle, WA	47.606365	-122.337654	Gas_station	06:00:00	22:10:00
5	Gas Station 5	2230 1st Ave, Seattle, WA	47.612823	-122.345673	Gas_station	14:00:00	21:00:00
6	Gas Station 6	1301 Alaskan Way, Seattle, WA	47.605961	-122.340363	Gas_station	12:00:00	23:11:00
7	Gas Station 7	2234 2nd Ave, Seattle, WA	47.613976	-122.345467	Gas_station	08:00:00	20:10:00
8	Gas Station 8	1416 E Olive Way, Seattle, WA	47.617214	-122.326584	Gas_station	06:00:00	13:00:00
9	Gas Station 9	1908 Pike pl, Seattle, WA	47.610126	-122.342834	Gas_station	10:00:00	23:55:00
14	Supermarket1	22 2nd Ave, Seattle, WA	47.617847	-122.303452	Supermarket	08:00:00	21:00:00
15	Supermarket2	2234 2nd Ave, Seattle, WA	47.619926	-122.335297	Supermarket	07:00:00	15:00:00
16	Supermarket3	2230 2st Ave, Seattle, WA	47.595741	-122.307404	Supermarket	08:00:00	20:00:00
17	Supermarket4	116 E Olive Way, Seattle, WA	47.602802	-122.314270	Supermarket	06:00:00	20:00:00
18	Supermarket5	2230 2st Ave, Seattle, WA	47.619926	-122.310661	Supermarket	12:00:00	23:00:00
19	Cinema1	1301 Alaskan Way, Seattle, WA	47.605961	-122.307404	Cinema	10:00:00	15:00:00
20	Cinema2	1416 E Olive Way, Seattle, WA	47.613976	-122.303452	Cinema	11:00:00	17:00:00
21	Cinema3	2234 2nd Ave, Seattle, WA	47.610126	-122.307571	Cinema	15:00:00	23:00:00
22	Cinema4	1908 Pike pl, Seattle, WA	47.619926	-122.314270	Cinema	12:00:00	16:00:00
23	Cinema5	1416 E Olive Way, Seattle, WA	47.624561	-122.310661	Cinema	16:00:00	21:00:00

- **Πίνακας, ΌνομαΧρήστη\_Tasks**

Αυτός είναι ο Πίνακας **ΌνομαΧρήστη\_Tasks**. Δημιουργείτε κάθε φορά που κάνει εγγραφή κάποιος χρήστης, έχει σαν πρώτη λέξη του ονόματος του το Username του χρήστη και χρησιμοποιείτε για να αποθηκεύουμε τις καθημερινές δραστηριότητες του χρήστη. Έχει πρωτεύον κλειδί το πεδίο **id**. Επίσης έχει τα παρακάτω πεδία: το όνομα (**name**) και τον τύπο (**type**) του σημείου ενδιαφέροντος, την ώρα έναρξης (**startTime**) και την ώρα λήξης (**endTime**) της δραστηριότητας και τις συντεταγμένες

(**userLat**), (**userLng**) της τοποθεσίας από την οποία θα ξεκινήσει ο χρήστης για να πάει στη συγκεκριμένη δραστηριότητα.

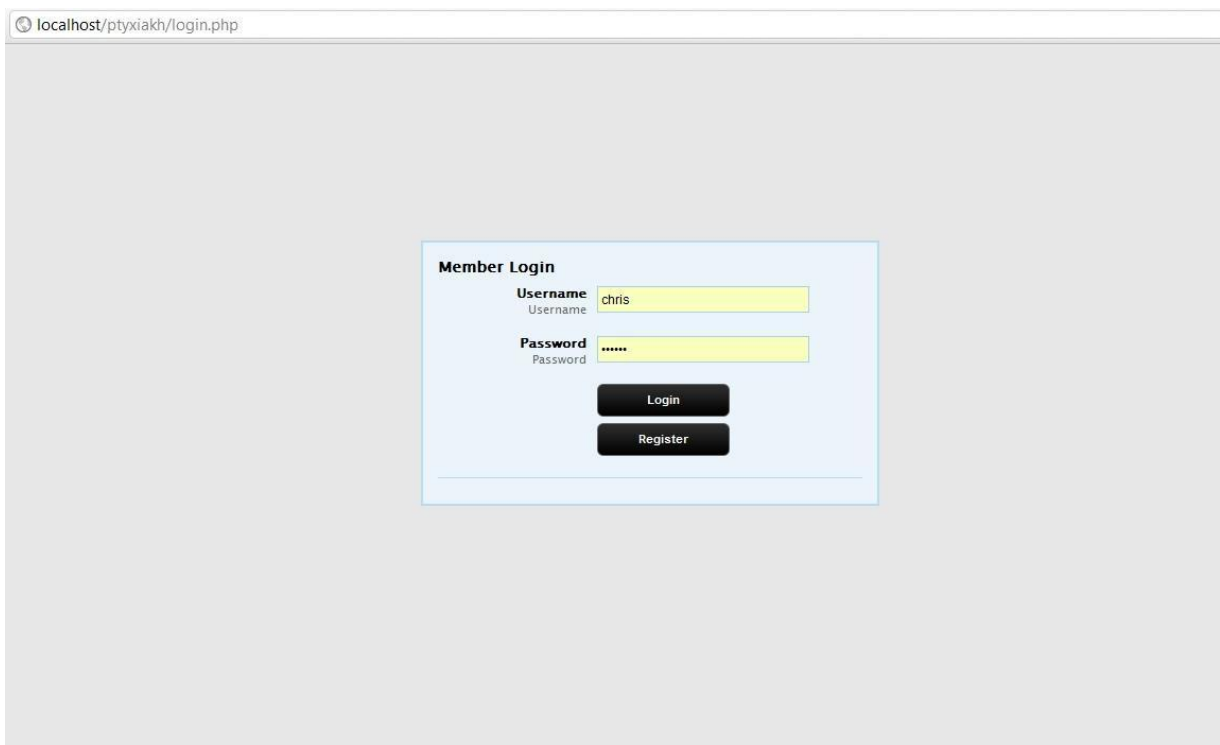
#	Στήλη	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Πρόσθετα
<input type="checkbox"/>	1 <b>id</b>	int(11)			Όχι	Καμία	AUTO_INCREMENT
<input type="checkbox"/>	2 <b>Name</b>	text	latin1_swedish_ci		Όχι	Καμία	
<input type="checkbox"/>	3 <b>Type</b>	text	latin1_swedish_ci		Όχι	Καμία	
<input type="checkbox"/>	4 <b>StartTime</b>	time			Όχι	Καμία	
<input type="checkbox"/>	5 <b>EndTime</b>	time			Όχι	Καμία	
<input type="checkbox"/>	6 <b>UserLat</b>	float(10,6)			Όχι	Καμία	
<input type="checkbox"/>	7 <b>UserLng</b>	float(10,6)			Όχι	Καμία	

id	Name	Type	StartTime	EndTime	UserLat	UserLng
1	Gas Station 4	Gas_station	19:00:00	19:03:12	47.607662	-122.294014
2	Cinema3	Cinema	18:30:00	18:31:46	47.596897	-122.293671
3	Cinema3	Cinema	22:10:00	22:12:40	47.586941	-122.319244
4	Gas Station 4	Gas_station	17:00:00	18:20:18	47.798622	-121.278893
5	Gas Station 9	Gas_station	23:10:00	23:12:52	47.603725	-122.305000
6	Cinema2	Cinema	16:50:00	16:50:29	47.613216	-122.304138
7	Gas Station 8	Gas_station	07:00:00	07:02:32	47.613682	-122.292465
8	Gas Station 8	Gas_station	07:20:00	07:27:26	47.622242	-122.225517
10	Supermarket4	Supermarket	07:30:00	08:11:59	47.591919	-122.292641
11	Supermarket2	Supermarket	08:20:00	09:09:34	47.605347	-122.324913
12	Gas Station 4	Gas_station	09:20:00	09:40:05	47.497948	-122.558113

## 5.3 Δημιουργία ιστοσελίδας

### Σελίδα Member Login

Αυτή είναι η αρχική σελίδα που πραγματοποιείτε η είσοδος του χρήστη στο σύστημα. Στη περίπτωση που εισέρχεται για πρώτη φορά θα πρέπει να κάνει εγγραφή πατώντας στο κουμπί Register.



Εικόνα 5.1 Σελίδα εισόδου χρήστη.

### Κώδικας εισόδου χρήστη

```
<?php
include ('config.php');
session_start();
$error = "";

if($_SERVER["REQUEST_METHOD"] == "POST") {

    $myusername=addslashes($_POST['myusername']);
    $mypassword=addslashes($_POST['mypassword']);
    $mypassword = md5($mypassword);

    $sql="SELECT id FROM $mysql_tablename WHERE username='$myusername' and
    password='$mypassword'";
    $result=mysql_query($sql);
    $row=mysql_fetch_array($result);
    $active=$row['active'];
```

```

// Mysql_num_row is counting table row
$count=mysql_num_rows($result);

// If result matched $myusername and $mypassword, table row must be 1 row
if($count==1){
// Register $myusername, $mypassword and redirect to file "home.php"
$_SESSION['login_user']=$myusername;
header("Location: home.php");

}
else {
$error = "Wrong Username or Password";
}

}

?>

<html>
<head>
<title>Member Login</title>

<style type="text/css">
body{
    background-color: #e7e7e7;
    font-family:"Lucida Grande", "Lucida Sans Unicode", Verdana, Arial, Helvetica, sans-
serif;
    font-size:12px;
}
p, h1, form, button{border:0; margin:0; padding:0;}
.spacer{clear:both; height:1px;}
/* ----- My Form ----- */
.myform{
    margin:50 auto;
    width:400px;
    padding:14px;
}

/* ----- stylized ----- */
#stylized {
    border:solid 2px #b7ddf2;
    background:#ebf4fb;
}
#stylized h1 {
    font-size:14px;
    font-weight:bold;
    margin-bottom:8px;
}
#stylized p{
    font-size:11px;
    font-weight: bold;
    color:#FF0000;
    margin-top:20px;
    border-top:solid 1px #b7ddf2;

```

```

        padding-top: 10px;
    }
    #stylized label{
        display: block;
        font-weight: bold;
        text-align: right;
        width: 140px;
        float: left;
    }
    #stylized .small{
        color: #666666;
        display: block;
        font-size: 11px;
        font-weight: normal;
        text-align: right;
        width: 140px;
    }
    #stylized input{
        float: left;
        font-size: 12px;
        padding: 4px 2px;
        border: solid 1px #aacfe4;
        width: 200px;
        margin: 2px 0 20px 10px;
    }
    #stylized button{
        clear: both;
        margin-left: 150px;
        width: 125px;
        height: 31px;
        background: #666666 url(img/button.png) no-repeat;
        text-align: center;
        line-height: 31px;
        color: #FFFFFF;
        font-size: 11px;
        font-weight: bold;
    }
}

</style>

</head>
<body>
<div id="stylized" class="myform" >
  <form name="form1" method="post" action="" >
    <h1>Member Login</h1>
    <label>Username
    <span class="small">Username</span>
  </label>
  <input name="myusername" type="text" id="myusername">

    <label>Password
    <span class="small">Password</span>
  </label>
  <input name="mypassword" type="password" id="mypassword">

```



```
<button type="submit">Login</button>
<div class="spacer"></div>

<a href="register.php">Register</a>

</form>
<?php
    echo "<center><p>$error</p><center>";
?>
</div>

</body>
</html>
```

Όταν πατηθεί το κουμπί login εκτελείτε ένας κώδικας PHP και δημιουργείτε ένα PHP Session, στη συνέχεια εκτελείτε ένα SQL Query για να διαπιστωθεί εάν υπάρχει στον πίνακα Users, της βάσης δεδομένων, εγγραφή με το ίδιο Username και Password που δόθηκε. Σε περίπτωση που δεν βρεθεί ο χρήστης στην βάση δεδομένων θα εμφανιστή μήνυμα με κόκκινα γράμματα προκειμένου να προειδοποιήσει τον χρήστη ότι δεν έχει βρεθεί το όνομα του ή ότι έχει κάνει κάποιο τυπογραφικό λάθος



**Member Login**

**Username**   
Username

**Password**   
Password

**Login**

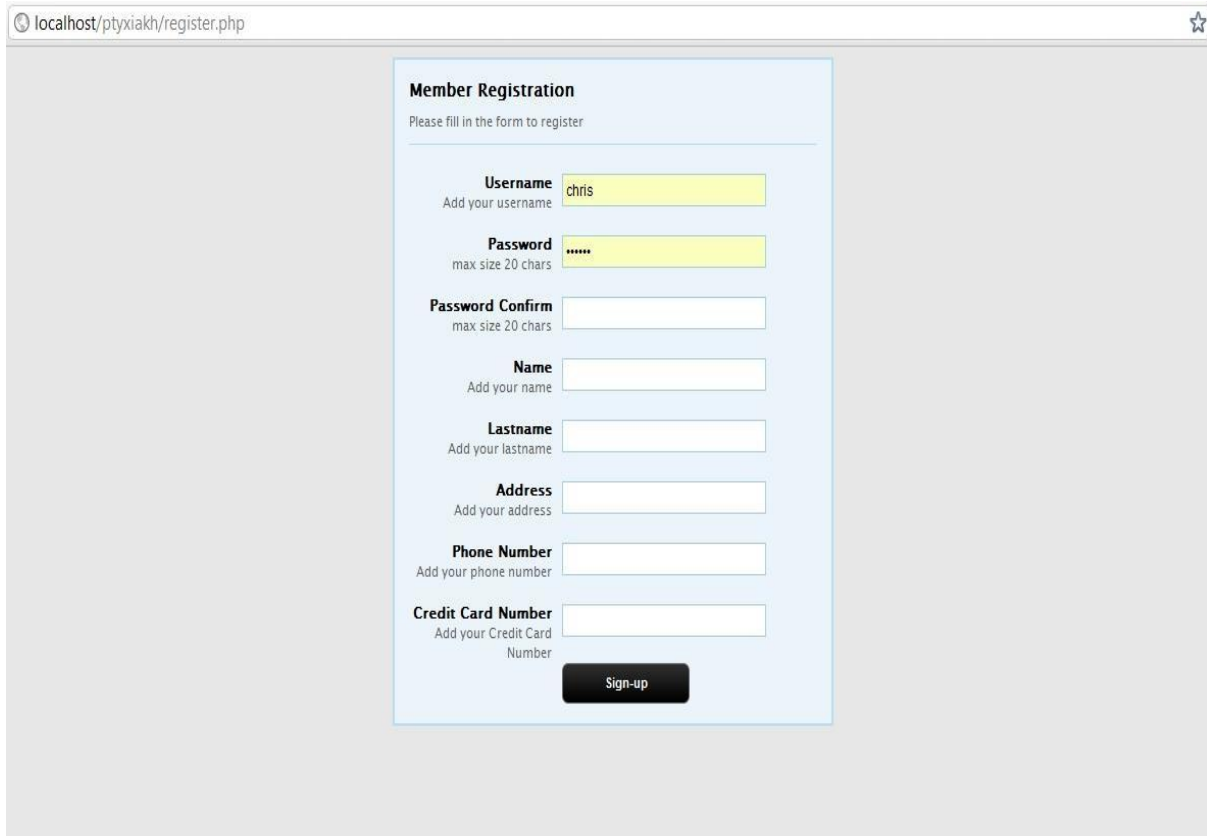
**Register**

**Wrong Username or Password**

Εικόνα 5.2 Αποτυχία εισόδου χρήστη.

## Σελίδα Member Registration

Αυτή είναι η σελίδα εγγραφής ενός νέου χρήστη. Για την εγγραφή του ο χρήστης πρέπει να συμπληρώσει όλα τα απαραίτητα στοιχεία όπως είναι το όνομα, το επίθετο, η διεύθυνση, το τηλέφωνο, και ο αριθμό της πιστωτικής του κάρτας.



The screenshot shows a web browser window with the address bar displaying 'localhost/ptyxiakh/register.php'. The main content area features a light blue box titled 'Member Registration' with the instruction 'Please fill in the form to register'. The form contains the following fields:

- Username:** Input field containing 'chris', with the label 'Add your username'.
- Password:** Input field with masked characters '\*\*\*\*\*', with the label 'max size 20 chars'.
- Password Confirm:** Input field with the label 'max size 20 chars'.
- Name:** Input field with the label 'Add your name'.
- Lastname:** Input field with the label 'Add your lastname'.
- Address:** Input field with the label 'Add your address'.
- Phone Number:** Input field with the label 'Add your phone number'.
- Credit Card Number:** Input field with the label 'Add your Credit Card Number'.

A black 'Sign-up' button is located at the bottom of the form.

Εικόνα 5.3 Φόρμα εγγραφής χρήστη

### Κώδικας εγγραφής χρήστη

```
<html>
<head>
<title>Registration Form</title>
<style type="text/css">
body{
font-family:"Lucida Grande", "Lucida Sans Unicode", Verdana, Arial, Helvetica, sans-serif;
font-size:12px;
background-color: #e7e7e7;
}
p, h1, form, button{border:0; margin:0; padding:0;}
.spacer{clear:both; height:1px;}
/* ----- My Form ----- */
.myform{
margin:5 auto;
width:400px;
padding:14px;
```

```

}

/* ----- stylized ----- */
#stylized{
border:solid 2px #b7ddf2;
background:#ebf4fb;
}
#stylized h1 {
font-size:14px;
font-weight:bold;
margin-bottom:8px;
}
#stylized p{
font-size:11px;
color:#666666;
margin-bottom:20px;
border-bottom:solid 1px #b7ddf2;
padding-bottom:10px;
}
#stylized li{
font-size:11px;
font-weight: bold;
color:#FF0000;
margin-top:20px;
padding-top:10px;
}
#stylized ul{
border-top:solid 1px #b7ddf2;
}
#stylized label{
display:block;
font-weight:bold;
text-align:right;
width:140px;
float:left;
}
#stylized .small{
color:#666666;
display:block;
font-size:11px;
font-weight:normal;
text-align:right;
width:140px;
}
#stylized input{
float:left;
font-size:12px;
padding:4px 2px;
border:solid 1px #aacfe4;
width:200px;
margin:2px 0 20px 10px;
}

```

```
#stylized button{
clear:both;
margin-left:150px;
width:125px;
height:31px;
background:#666666 url(img/button.png) no-repeat;
text-align:center;
line-height:31px;
color:#FFFFFF;
font-size:11px;
font-weight:bold;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div id="stylized" class="myform">
```

```
<form action="register.php" method="POST" >
```

```
<h1>Member Registration</h1>
```

```
<p>Please fill in the form to register</p>
```

```
<input type="hidden" name="action" value="register">
```

```
<label>Username
```

```
<span class="small">Add your username</span>
```

```
</label>
```

```
<input type='text' name='username' maxlength=20 value='<?php echo
(isset($_REQUEST['username']) ? $_REQUEST['username'] : ""); ?>'>
```

```
<label>Password
```

```
<span class="small">max size 20 chars </span>
```

```
</label>
```

```
<input type="password" maxlength=20 name="password1">
```

```
<label>Password Confirm
```

```
<span class="small">max size 20 chars</span>
```

```
</label>
```

```
<input type="password" maxlength=20 name="password2">
```

```
<label>Name
```

```
<span class="small">Add your name</span>
```

```
</label>
```

```
<input type='text' name='name' maxlength=15 value='<?php echo
(isset($_REQUEST['name']) ? $_REQUEST['name'] : ""); ?>'>
```

```
<label>Lastname
```

```
<span class="small">Add your lastname</span>
```

```

</label>
<input type='text' name='lastname' maxlength=50 value='<?php echo
(isset($_REQUEST['lastname']) ? $_REQUEST['lastname'] : ""); ?>'>

<label>Address
<span class="small">Add your address</span>
</label>
<input type='text' name='address' maxlength=20 value='<?php echo
(isset($_REQUEST['address']) ? $_REQUEST['address'] : ""); ?>'>

<label>Phone Number
<span class="small">Add your phone number</span>
</label>
<input type='text' name='phone' maxlength=10 value='<?php echo
(isset($_REQUEST['phone']) ? $_REQUEST['phone'] : ""); ?>'>

<label>Credit Card Number
<span class="small">Add your Credit Card Number</span>
</label>
<input type='text' name='creditcard' maxlength=16 value='<?php echo
(isset($_REQUEST['creditcard']) ? $_REQUEST['creditcard'] : ""); ?>'>

<button type="submit" name="Submit">Sign-up</button>
<div class="spacer"></div>
</form>

```

```

<?php
include "config.php";

$error = "";

if($_SERVER["REQUEST_METHOD"] == "POST") {

//Check if all fields have value
if(empty($_POST['username']) || empty($_POST['password1']) ||
empty($_POST['password2']) || empty($_POST['name']) || empty($_POST['lastname'])
|| empty($_POST['address']) || empty($_POST['phone']) || empty($_POST['creditcard'])) {
echo "<font color='red'>*Please fill out all form fields</font>";
exit;
}
else {

// Get data from registration form
$username=$_POST['username'];
$password1=$_POST['password1'];
$password2=$_POST['password2'];
$name=$_POST['name'];

```

```

$lastname=$_POST['lastname'];
$address=$_POST['address'];
$phone=$_POST['phone'];
$creditcard=$_POST['creditcard'];

if(! formHasErrors($username, $password1, $password2, $phone, $creditcard, $error) ) {
//Make Query
$insert_query = 'insert into      users
(
name,
lastname,
address,
phone,
creditcard,
username,
password
)
values
(
" . $name . "',
" . $lastname . "',
" . $address . "',
" . $phone . "',
" . $creditcard . "',
" . $username . "',
" . md5($password1) . "'
)';

// Perform Query
$result = mysql_query($insert_query);
$query_create_table_tasks = 'CREATE TABLE '.$username.'_tasks (
`id` int(11) NOT NULL AUTO_INCREMENT,
`Name` text NOT NULL,
`Type` text NOT NULL,
`StartTime` time NOT NULL,
`EndTime` time NOT NULL,
`UserLat` float(10,6) NOT NULL,
`UserLng` float(10,6) NOT NULL,
PRIMARY KEY (`id`)
)';

$result2 = mysql_query($query_create_table_tasks);

// Check result
// This shows the actual query sent to MySQL, and the error. Useful for debugging.
if (!$result || !$result2) {
$message = 'Invalid query: ' . mysql_error() . "\n";
$message .= 'Whole query: ' . $insert_query;
die($message);
}
}

```

```

else {
/*echo "<br>";
echo "<b>". $username .", you have registered successfully!!</b>";
echo "<br>";
echo "<h1><a href='login.php'>Return to login page</a> </h1>";*/
header( 'Location: /ptyxiakh/login.php' );
}

}

}

}

function formHasErrors($username, $password1, $password2, $phone, $creditcard,
$error) {
//Check to see if username exists
$formHasErrors = false;
$sql = mysql_query("SELECT username FROM users WHERE username = '$username'");
if (mysql_num_rows($sql))
{
$error .= "*Username exists. Please select another username<br>";
$formHasErrors = true;
}

//Check if passwords are the same
if(!($password1==$password2)) {
$error .= "*The confirmation password did not match the initial password <br>";
$formHasErrors = true;
}

//Check if phone is a valid phone
if(!preg_match('/^[1-9]\d{9}$/', $phone)) {
$error .= "*Please enter a valid phone number<br>";
$formHasErrors = true;
}

//Check if credic card number is a valid number
if(!preg_match('/^[1-9]\d{15}$/', $creditcard)) {
$error .= "*Please enter a valid credit card number<br>";
$formHasErrors = true;
}

echo "<font color='red'>". $error . "</font>";
return $formHasErrors;

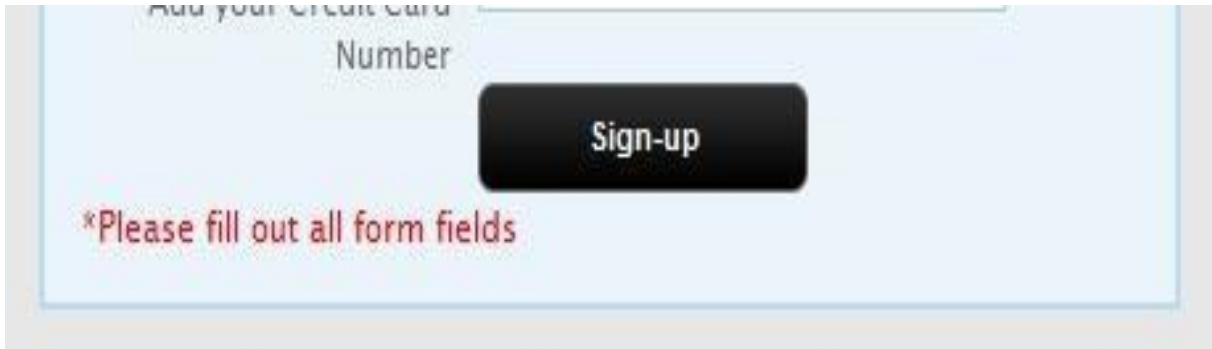
}

?>

</body>
</html>

```

Επίσης πρέπει να επιλέξει ένα δικό του username και password. Στη συνέχεια γίνεται έλεγχος εάν το username είναι μοναδικό και τα υπόλοιπα στοιχεία είναι σωστά. Αν τα στοιχεία είναι σωστά καταχωρείτε στον πίνακα users το username του χρήστη και δημιουργείτε ένας πίνακας με το όνομα “όνομαΧρήστη\_tasks” για την αποθήκευση των δραστηριοτήτων του.



Εικόνα 5.4 Έλεγχος της φόρμας στοιχείων

#### Κώδικας ελέγχου φόρμας

```
<?PHP

/**
 * Carries information about each of the form validations
 */
class ValidatorObj
{
    var $variable_name;
    var $validator_string;
    var $error_string;
}

/**
 * Base class for custom validation objects
 */
class CustomValidator
{
    function DoValidate(&$formars,&$error_hash)
    {
        return true;
    }
}

/** Default error messages*/
define("E_VAL_REQUIRED_VALUE","Please enter the value for %s");
define("E_VAL_MAXLEN_EXCEEDED","Maximum length exceeded for %s.");
define("E_VAL_MINLEN_CHECK_FAILED","Please enter input with length more than %d for %s");
define("E_VAL_ALNUM_CHECK_FAILED","Please provide an alpha-numeric input for %s");
```



```

define("E_VAL_ALNUM_S_CHECK_FAILED","Please provide an alpha-numeric input for %s");
define("E_VAL_NUM_CHECK_FAILED","Please provide numeric input for %s");
define("E_VAL_ALPHA_CHECK_FAILED","Please provide alphabetic input for %s");
define("E_VAL_ALPHA_S_CHECK_FAILED","Please provide alphabetic input for %s");
define("E_VAL_EMAIL_CHECK_FAILED","Please provide a valida email address");
define("E_VAL_LESSTHAN_CHECK_FAILED","Enter a value less than %f for %s");
define("E_VAL_GREATERTHAN_CHECK_FAILED","Enter a value greater than %f for %s");
define("E_VAL_REGEXP_CHECK_FAILED","Please provide a valid input for %s");
define("E_VAL_DONTSEL_CHECK_FAILED","Wrong option selected for %s");
define("E_VAL_SELMIN_CHECK_FAILED","Please select minimum %d options for %s");
define("E_VAL_SELONE_CHECK_FAILED","Please select an option for %s");
define("E_VAL_EQELMNT_CHECK_FAILED","Value of %s should be same as that of %s");
define("E_VAL_NEELMNT_CHECK_FAILED","Value of %s should not be same as that of %s");

```

```
/**
```

```
* FormValidator: The main class that does all the form validations
```

```
**/
```

```
class FormValidator
```

```
{
```

```
    var $validator_array;
```

```
    var $error_hash;
```

```
    var $custom_validators;
```

```
    function FormValidator()
```

```
    {
```

```
        $this->validator_array = array();
```

```
    $this->error_hash = array();
```

```
        $this->custom_validators=array();
```

```
    }
```

```
    function AddCustomValidator(&$customv)
```

```
    {
```

```
        array_push($this->custom_validators,$customv);
```

```
    }
```

```
    function addValidation($variable,$validator,$error)
```

```
    {
```

```
        $validator_obj = new ValidatorObj();
```

```
        $validator_obj->variable_name = $variable;
```

```
        $validator_obj->validator_string = $validator;
```

```
        $validator_obj->error_string = $error;
```

```
        array_push($this->validator_array,$validator_obj);
```

```
    }
```

```
    function GetErrors()
```

```
    {
```

```
        return $this->error_hash;
```

```
    }
```

```
    function ValidateForm()
```

```
    {
```

```
        $bret = true;
```

```

$error_string="";
$error_to_display = "";

if(strcmp($_SERVER['REQUEST_METHOD'],'POST')==0)
{
    $form_variables = $_POST;
}
else
{
    $form_variables = $_GET;
}

$vcount = count($this->validator_array);

foreach($this->validator_array as $val_obj)
{
    if(!$this->ValidateObject($val_obj,$form_variables,$error_string))
    {
        $bret = false;
        $this->error_hash[$val_obj->variable_name] = $error_string;
    }
}

if(true == $bret && count($this->custom_validators) > 0)
{
    foreach( $this->custom_validators as $custom_val)
    {
        if(false == $custom_val->DoValidate($form_variables,$this-
>error_hash))
        {
            $bret = false;
        }
    }
}
return $bret;
}

function ValidateObject($validatorobj,$formvariables,&$error_string)
{
    $bret = true;

    $splitted = explode("=", $validatorobj->validator_string);
    $command = $splitted[0];
    $command_value = "";

    if(isset($splitted[1]) && strlen($splitted[1])>0)
    {
        $command_value = $splitted[1];
    }

    $default_error_message="";

    $input_value = "";

```

```

        if(isset($formvariables[$validatorobj->variable_name]))
        {
            $input_value = $formvariables[$validatorobj->variable_name];
        }

        $bret = $this->ValidateCommand($command,$command_value,$input_value,
        $default_error_message,
        $validatorobj->variable_name,
        $formvariables);

        if(false == $bret)
        {
            if(isset($validatorobj->error_string) &&
                strlen($validatorobj->error_string)>0)
            {
                $error_string = $validatorobj->error_string;
            }
            else
            {
                $error_string = $default_error_message;
            }
        }
        //if
        return $bret;
    }

    function validate_req($input_value, &$default_error_message,$variable_name)
    {
        $bret = true;
        if(!isset($input_value) ||
            strlen($input_value) <=0)
        {
            $bret=false;
            $default_error_message =
            sprintf(E_VAL_REQUIRED_VALUE,$variable_name);
        }
        return $bret;
    }

    function
    validate_maxlen($input_value,$max_len,$variable_name,&$default_error_message)
    {
        $bret = true;
        if(isset($input_value) )
        {
            $input_length = strlen($input_value);
            if($input_length > $max_len)
            {
                $bret=false;
                $default_error_message =
                sprintf(E_VAL_MAXLEN_EXCEEDED,$variable_name);
            }
        }
    }

```

```

    }
    return $bret;
}

function
validate_minlen($input_value,$min_len,$variable_name,&$default_error_message)
{
    $bret = true;
    if(isset($input_value) )
    {
        $input_length = strlen($input_value);
        if($input_length < $min_len)
        {
            $bret=false;
            $default_error_message =
sprintf(E_VAL_MINLEN_CHECK_FAILED,$min_len,$variable_name);
        }
    }
    return $bret;
}

function test_datatype($input_value,$reg_exp)
{
    if(ereg($reg_exp,$input_value))
    {
        return false;
    }
    return true;
}

function validate_email($email)
{
    return eregi("^[_\.\0-9a-zA-Z-]+\@[0-9a-zA-Z][0-9a-zA-Z-]+\.[a-zA-Z]{2,6}$", $email);
}

function validate_for_numeric_input($input_value,&$validation_success)
{
    $more_validations=true;
    $validation_success = true;
    if(strlen($input_value)>0)
    {
        if(false == is_numeric($input_value))
        {
            $validation_success = false;
            $more_validations=false;
        }
    }
    else
    {
        $more_validations=false;
    }
    return $more_validations;
}

```

```

function validate_lessthan($command_value,$input_value,
    $variable_name,&$default_error_message)
{
    $bret = true;
    if(false == $this->validate_for_numeric_input($input_value,
        $bret))
    {
        return $bret;
    }
    if($bret)
    {
        $lessthan = doubleval($command_value);
        $float_inputval = doubleval($input_value);
        if($float_inputval >= $lessthan)
        {
            $default_error_message =
sprintf(E_VAL_LESSTHAN_CHECK_FAILED,
                $lessthan,
                $variable_name);
            $bret = false;
        }//if
    }
    return $bret ;
}

function
validate_greaterthan($command_value,$input_value,$variable_name,&$default_error_message)
{
    $bret = true;
    if(false == $this->validate_for_numeric_input($input_value,$bret))
    {
        return $bret;
    }
    if($bret)
    {
        $greaterthan = doubleval($command_value);
        $float_inputval = doubleval($input_value);
        if($float_inputval <= $greaterthan)
        {
            $default_error_message =
sprintf(E_VAL_GREATERTHAN_CHECK_FAILED,
                $greaterthan,
                $variable_name);
            $bret = false;
        }//if
    }
    return $bret ;
}

function
validate_select($input_value,$command_value,&$default_error_message,$variable_name)
{
    $bret=false;

```

```

        if(is_array($input_value))
        {
            foreach($input_value as $value)
            {
                if($value == $command_value)
                {
                    $bret=true;
                    break;
                }
            }
        }
        else
        {
            if($command_value == $input_value)
            {
                $bret=true;
            }
        }
    }
    if(false == $bret)
    {
        $default_error_message = sprintf(E_VAL_SHOULD_SEL_CHECK_FAILED,
            $command_value,$variable_name);
    }
    return $bret;
}

function
validate_dontselect($input_value,$command_value,&$default_error_message,$variable_name)
{
    $bret=true;
    if(is_array($input_value))
    {
        foreach($input_value as $value)
        {
            if($value == $command_value)
            {
                $bret=false;
                $default_error_message =
sprintf(E_VAL_DONTSEL_CHECK_FAILED,$variable_name);
                break;
            }
        }
    }
    else
    {
        if($command_value == $input_value)
        {
            $bret=false;
            $default_error_message =
sprintf(E_VAL_DONTSEL_CHECK_FAILED,$variable_name);
        }
    }
    return $bret;
}

```

```

function
ValidateCommand($command,$command_value,$input_value,&$default_error_message,$variable_name,$formvariables)
{
    $bret=true;
    switch($command)
    {
        case 'req':
            {
                $bret = $this->validate_req($input_value, $default_error_message,$variable_name);
                break;
            }

        case 'maxlen':
            {
                $max_len = intval($command_value);
                $bret = $this->validate_maxlen($input_value,$max_len,$variable_name,
                    $default_error_message);
                break;
            }

        case 'minlen':
            {
                $min_len = intval($command_value);
                $bret = $this->validate_minlen($input_value,$min_len,$variable_name,
                    $default_error_message);
                break;
            }

        case 'alnum':
            {
                $bret= $this->test_datatype($input_value,"[^A-Za-z0-9]");
                if(false == $bret)
                {
                    $default_error_message =
                    sprintf(E_VAL_ALNUM_CHECK_FAILED,$variable_name);
                }
                break;
            }

        case 'alnum_s':
            {
                $bret= $this->test_datatype($input_value,"[^A-Za-z0-9 ]");
                if(false == $bret)
                {
                    $default_error_message =
                    sprintf(E_VAL_ALNUM_S_CHECK_FAILED,$variable_name);
                }
            }
    }
}

```

```

        break;
    }

    case 'num':
    case 'numeric':
    {
        $bret= $this->test_datatype($input_value,"[^\0-9]");
        if(false == $bret)
        {
            $default_error_message =
            sprintf(E_VAL_NUM_CHECK_FAILED,$variable_name);
        }
        break;
    }

    case 'alpha':
    {
        $bret= $this->test_datatype($input_value,"[^A-Za-z]");
        if(false == $bret)
        {
            $default_error_message =
            sprintf(E_VAL_ALPHA_CHECK_FAILED,$variable_name);
        }
        break;
    }

    case 'alpha_s':
    {
        $bret= $this->test_datatype($input_value,"[^A-Za-z ]");
        if(false == $bret)
        {
            $default_error_message =
            sprintf(E_VAL_ALPHA_S_CHECK_FAILED,$variable_name);
        }
        break;
    }

    case 'email':
    {
        if(isset($input_value) &&
        strlen($input_value)>0)
        {
            $bret= $this->validate_email($input_value);
            if(false == $bret)
            {
                $default_error_message
            }
        }
        break;
    }

    case "lt":
    case "lessthan":

```



```

        {
            $bret = $this->validate_lessthan($command_value,
                $input_value,
                $variable_name,
                $default_error_message);
            break;
        }
        case "gt":
        case "greaterthan":
        {
            $bret = $this->validate_greaterthan($command_value,
                $input_value,
                $variable_name,
                $default_error_message);
            break;
        }
        case "regexp":
        {
            if(isset($input_value) &&
                strlen($input_value)>0)
            {
                if(!preg_match("$command_value",$input_value))
                {
                    $bret=false;
                    $default_error_message
                    = sprintf(E_VAL_REGEXP_CHECK_FAILED,$variable_name);
                }
            }
            break;
        }
        case "dontselect":
        case "dontselectchk":
        case "dontselectradio":
        {
            $bret = $this->validate_dontselect($input_value,
                $command_value,
                $default_error_message,
                $variable_name);
            break;
        }
        //case
        case "shouldselchk":

```

```

case "selectradio":
    {
        $bret = $this->validate_select($input_value,
                                    $command_value,
                                    $default_error_message,
                                    $variable_name);

        break;
    }//case
case "selmin":
    {
        $min_count = intval($command_value);

        if(isset($input_value))
        {
            if($min_count > 1)
            {
                $bret = (count($input_value) >=
$min_count )?true:false;
            }
            else
            {
                $bret = true;
            }
        }
        else
        {
            $bret= false;
            $default_error_message =
sprintf(E_VAL_SELMIN_CHECK_FAILED,$min_count,$variable_name);
        }
        break;
    }//case
case "selone":
    {
        if(false == isset($input_value)||
strlen($input_value)<=0)
        {
            $bret= false;
            $default_error_message =
sprintf(E_VAL_SELONE_CHECK_FAILED,$variable_name);
        }
        break;
    }
case "eqelmnt":
    {

        if(isset($formvariables[$command_value]) &&
strcmp($input_value,$formvariables[$command_value])==0 )
        {
            $bret=true;
        }
        else
        {

```

```

                                                                    $bret= false;
                                                                    $default_error_message =
sprintf(E_VAL_EQELMNT_CHECK_FAILED,$variable_name,$command_value);
                                                                    }
                                                                    break;
                                                                    }
        case "neelmnt":
                                                                    {

                if(isset($formvariables[$command_value]) &&
                strcmp($input_value,$formvariables[$command_value]) !=0 )
                                                                    {
                                                                    $bret=true;
                                                                    }
                else
                                                                    {
                                                                    $bret= false;
                                                                    $default_error_message =
sprintf(E_VAL_NEELMNT_CHECK_FAILED,$variable_name,$command_value);
                                                                    }
                                                                    break;
                                                                    }

                }//switch
                return $bret;
        }//validdate command

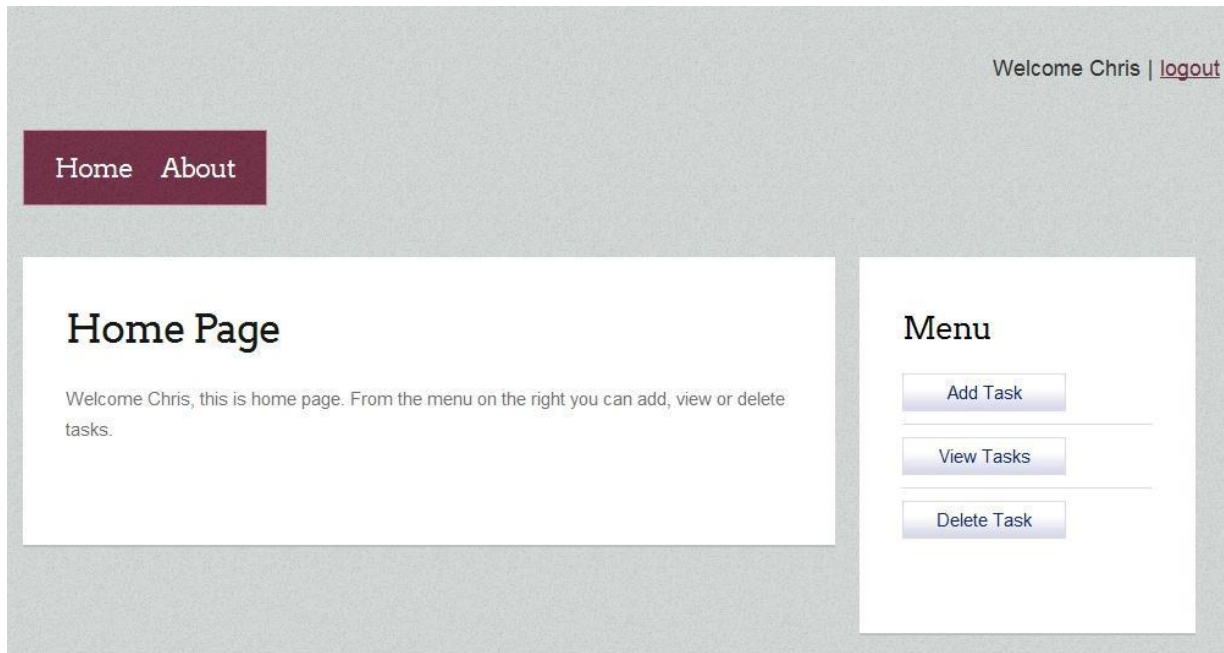
}

?>

```

## Αρχική Σελίδα

Η αρχική σελίδα της εφαρμογής είναι αυτή που φαίνεται στην **εικόνα 5.4**. Αυτή η σελίδα εμφανίζεται αφού κάνει login ο χρήστης. Από το μενού που υπάρχει στα δεξιά ο χρήστης μπορεί να προσθέσει / διαγράψει μια δραστηριότητα ή να δει τις δραστηριότητες που έχει προγραμματίσει.



**Εικόνα 5.4** Η Αρχική σελίδα

### Κώδικας αρχικής σελίδας

```
<?php
include('lock.php');
?>
<!DOCTYPE html ">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Home </title>
<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
<style type="text/css">

a.logout:link { color:#5B98FE; text-decoration: none;}
a.logout:visited {text-decoration: none}
a.logout:active {text-decoration: none}
a.logout:hover {color: #F14C4D; text-decoration: none;}

a.menu:link { color:#DE6E45; text-decoration: none;}
a.menu:visited {text-decoration: none}
a.menu:active {text-decoration: none}
a.menu:hover {color: #C9C9C9; text-decoration: none;}

body{
background-color: #e7e7e7;
}

html { height: 100% ;}
```

```

body {
  height: 100%;
  margin: 0px;
  padding: 0px;
}

p.pos_absolute {
  font-size: 120%;
  position: absolute;
  top: 3px;
  right: 170px;
}

p.pos_absolute2 {
  font-size: 100%;
  position: absolute;
  top: 65px;
  right: 170px;
}

#sidebar {
  position: relative;
  margin-left: auto;
  margin-right: auto;
  top: 100px;
  width: 250px;
  font-size: .8em;
}

#navlist {
  margin: 0; /*removes indent IE and Opera*/
  padding: 0; /*removes indent Mozilla and NN7*/
  list-style-type: none; /*turns off display of bullet*/
}

#navlist a, #navlist a:visited {
  display: block;
  padding: 2px 2px 2px 24px;
  border: 1px solid;
  border-color: #ddd #000 #000 #ddd;
  background-color: #999999;
  background-image: url(img/11_down.jpg);
  background-repeat: no-repeat;
  background-position: 0% 50%;
  color: #EEE;
  text-decoration: none;
}

#navlist a:hover {
  background-color: #9F6F9F;
  background-image: url(img/11_over.jpg);
  color: #2D1E2D;
}

#sidebar #navlist .uberlink {

```

```
background-color: #9F6F9F;
background-image: url(assets/11_over.jpg);
background-repeat: no-repeat;
background-position: 0% 50%;
color: #333 !important;
font-weight: bold;
cursor: default;
}
```

```
#container {
border: 1px solid #666666;
text-align: center;
margin-left: 100px;
margin-right: 100px;
margin-top: 0px;
padding-bottom: 200px;
background-color: white;
}
```

```
#div_clock {
font-size:100%;
position:absolute;
top:90px;
right:170px;
}
```

```
#head {
height: 30px;
margin: 20px 20px 5px 20px;
padding: 0px;
}
```

```
#content {
width: 80%;
height: 40%;
border: 1px solid #666666;
text-align: center;
margin-left: auto;
margin-right: auto;
margin-top: 110px;
padding-bottom: 100px;
background-color: #E5F4F6;
}
```

</style>

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

```
weekday=new Array("Sunday","Monday","Tuesday","Wednesday","Thursday",
"Friday","Saturday");
var mydate=new Date();
var year=mydate.getYear();
if (year < 1000)
year+=1900;
var day=mydate.getDay();
```

```

var month=mydate.getMonth()+1;
if (month<10)
month="0"+month;
var daym=mydate.getDate();
if (daym<10)
daym="0"+daym;
document.write("<p class='pos_absolute2'>" +weekday[mydate.getDay()]+
"+daym+"/"+month+"/"+year+"</p>");
//var currentDateString = weekday[mydate.getDay()+" "+month+"/"+daym+"/"+year;
//document.getElementById("date").firstChild.nodeValue = currentDateString;

function updateClock ( )
{
var currentTime = new Date ( );

var currentHours = currentTime.getHours ( );
var currentMinutes = currentTime.getMinutes ( );
var currentSeconds = currentTime.getSeconds ( );

// Pad the minutes and seconds with leading zeros, if required
currentMinutes = ( currentMinutes < 10 ? "0" : "" ) + currentMinutes;
currentSeconds = ( currentSeconds < 10 ? "0" : "" ) + currentSeconds;

// Choose either "AM" or "PM" as appropriate
var timeOfDay = ( currentHours < 12 ) ? "AM" : "PM";

// Convert the hours component to 12-hour format if needed
currentHours = ( currentHours > 12 ) ? currentHours - 12 : currentHours;

// Convert an hours component of "0" to "12"
currentHours = ( currentHours == 0 ) ? 12 : currentHours;

// Compose the string for display
var currentTimeString = currentHours + ":" + currentMinutes + ":" + currentSeconds + " " +
timeOfDay;

// Update the time display
document.getElementById("clock").firstChild.nodeValue = currentTimeString;
}
//-->
</script>
<script type="text/javascript" src="assets/p7uberlink.js"></script>
</head>

<body onload="updateClock(); setInterval('updateClock()', 1000 )"
onLoad="P7_Uberlink('uberlink','navlist')">
<div id="container">
<div id="head"><p class="pos_absolute">Welcome <?php echo $login_session; ?> | <a
class="logout" href="logout.php">logout</a></p></div>
<div id="div_clock"><span id="clock">&nbsp;</span></div>
<div id="content"><h2>Daily Tasks</h2>
<div id="sidebar">
<ul id="navlist">

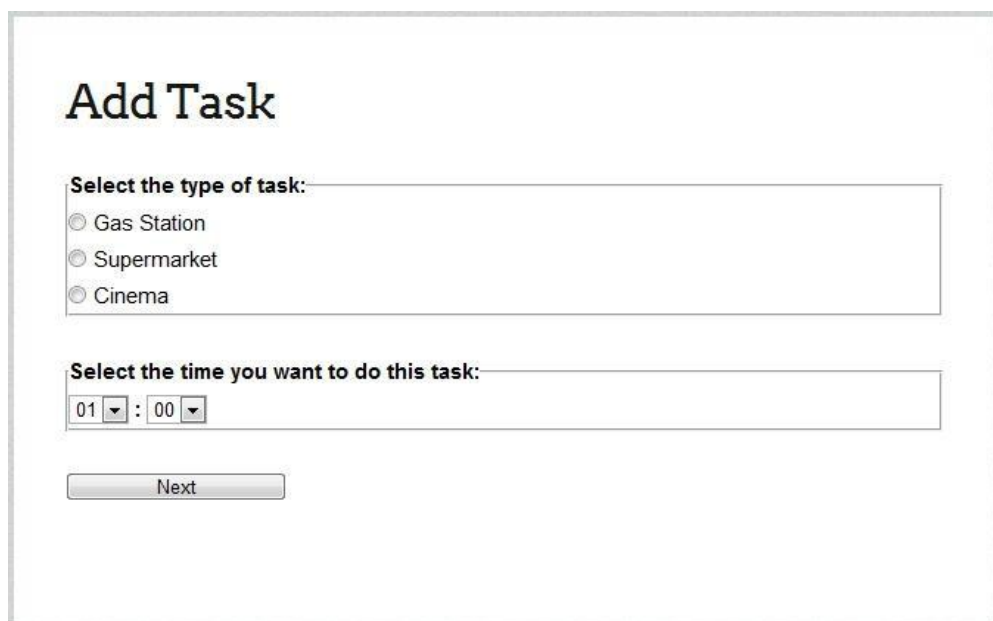
```

```
<li><a href="add.php">Add Task</a></li>
<li><a href="view.php">View Tasks</a></li>
<li><a href="view.php">Delete Task</a></li>
</ul>
</div>
</div>
</div>
</body>

</html>
```

## Σελίδα Προσθήκης Δραστηριότητας

Σε αυτή την σελίδα ο χρήστης μπορεί να προσθέσει μια καινούρια δραστηριότητα, επιλέγοντας τον τύπο της και την ώρα που θέλει να πραγματοποιηθεί. Στη συνέχεια πατώντας το κουμπί Next πηγαίνει στην επόμενη σελίδα όπου εκεί μέσα από το χάρτη του Google Maps βλέπει τα σημεία ενδιαφέροντος. Κάνοντας κλικ σε κάποιο



**Εικόνα 5.6 Προσθήκη Δραστηριότητας**

σημείο στο χάρτη δηλώνετε η τοποθεσία που βρίσκετε ο χρήστης. Στη συνέχεια μέσω του τύπου Haversine υπολογίζετε ποιο είναι το κοντινότερο σημείο ενδιαφέροντος και μέσω του Directions Service του Google Maps JavaScript API V3 υπολογίζετε η συντομότερη διαδρομή.



## Κώδικας προσθήκης δραστηριότητας

```
<<?php
include('lock.php');
?>
<!DOCTYPE html ">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Add Task </title>
<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
<style type="text/css">

a.logout:link { color:#5B98FE; text-decoration: none;}
a.logout:visited {text-decoration: none}
a.logout:active {text-decoration: none}
a.logout:hover {color: #F14C4D; text-decoration: none;}

a.menu:link { color:#DE6E45; text-decoration: none;}
a.menu:visited {text-decoration: none}
a.menu:active {text-decoration: none}
a.menu:hover {color: #C9C9C9; text-decoration: none;}

body{
background-color: #e7e7e7;
height: 100%;
margin: 0px;
padding: 0px;
}

html { height: 100% }

p.pos_absolute {
font-size:120%;
position:absolute;
top:3px;
right:170px;
}
p.pos_absolute2 {
font-size:100%;
position:absolute;
top:65px;
right:170px;
}

#sidebar {
position: relative;
left: 25px;
top: 100px;
width: 150px;
font-size: .8em;
}

#navlist {
```

```
margin: 0; /*removes indent IE and Opera*/
padding: 0; /*removes indent Mozilla and NN7*/
list-style-type: none; /*turns off display of bullet*/
}
```

```
#navlist a, #navlist a:visited {
display: block;
padding: 2px 2px 2px 24px;
border: 1px solid;
border-color: #ddd #000 #000 #ddd;
background-color: #999999;
background-image: url(img/11_down.jpg);
background-repeat: no-repeat;
background-position: 0% 50%;
color: #EEE;
text-decoration: none;
}
```

```
#navlist a:hover {
background-color: #9F6F9F;
background-image: url(img/11_over.jpg);
color: #2D1E2D;
}
```

```
#sidebar #navlist .uberlink {
background-color: #9F6F9F;
background-image: url(assets/11_over.jpg);
background-repeat: no-repeat;
background-position: 0% 50%;
color: #333 !important;
font-weight: bold;
cursor: default;
}
```

```
#container {
border: 1px solid #666666;
text-align: center;
margin-left: 100px;
margin-right: 100px;
margin-top: 0px;
padding-bottom: 200px;
background-color: white;
}
```

```
#div_clock {
font-size:100%;
position:absolute;
top:90px;
right:170px;
}
```

```
#head {
height: 30px;
margin: 20px 20px 5px 20px;
padding: 0px;
```

```

}

#content {
  width: 80%;
  height: 40%;
  border: 1px solid #666666;
  text-align: center;
  margin-left: auto;
  margin-right: auto;
  margin-top: 110px;
  padding-bottom: 100px;
  background-color: #E5F4F6;
}

#myFormDiv {

  margin-left: auto;
  margin-right: auto;
  background-color: #E5F4F6;
}

.input_radio {
  width: 90%;
  padding-top: 50px;
  margin-left: auto;
  margin-right: auto;
}

</style>

<script type="text/javascript">
<!--
weekday=new Array("Sunday","Monday","Tuesday","Wednesday","Thursday",
  "Friday","Saturday");
var mydate=new Date();
var year=mydate.getYear();
if (year < 1000)
year+=1900;
var day=mydate.getDay();
var month=mydate.getMonth()+1;
if (month<10)
month="0"+month;
var daym=mydate.getDate();
if (daym<10)
daym="0"+daym;
document.write("<p class='pos_absolute2'>" +weekday[mydate.getDay()]+
"+daym+"/"+month+"/"+year+"</p>");
//var currentDateString = weekday[mydate.getDay()]+ " "+month+"/"+daym+"/"+year;
//document.getElementById("date").firstChild.nodeValue = currentDateString;

var url;
var page;

```

```

var hours = "01";
var minutes = "00";

function updateClock ( )
{
    var currentTime = new Date ( );

    var currentHours = currentTime.getHours ( );
    var currentMinutes = currentTime.getMinutes ( );
    var currentSeconds = currentTime.getSeconds ( );

    // Pad the minutes and seconds with leading zeros, if required
    currentMinutes = ( currentMinutes < 10 ? "0" : "" ) + currentMinutes;
    currentSeconds = ( currentSeconds < 10 ? "0" : "" ) + currentSeconds;

    // Choose either "AM" or "PM" as appropriate
    var timeOfDay = ( currentHours < 12 ) ? "AM" : "PM";

    // Convert the hours component to 12-hour format if needed
    currentHours = ( currentHours > 12 ) ? currentHours - 12 : currentHours;

    // Convert an hours component of "0" to "12"
    currentHours = ( currentHours == 0 ) ? 12 : currentHours;

    // Compose the string for display
    var currentTimeString = currentHours + ":" + currentMinutes + ":" + currentSeconds + " " +
timeOfDay;

    // Update the time display
    document.getElementById("clock").firstChild.nodeValue = currentTimeString;
}

function GetSelectedItem() {
    chosen = "";
    len = document.f1.group1.length;

    for (i = 0; i <len; i++) {
        if (document.f1.group1[i].checked) {
            chosen = document.f1.group1[i].value;
            if(chosen=='1')
                page = "add_gastation.php";
            if(chosen=='2')
                page = "add_supermarket.php";
            if(chosen=='3')
                page = "add_cinema.php";
        }
    }
}

function nextPage() {
    boxes=document.getElementsByName("group1")
    checkedBox=0
    for(var i=0;i<boxes.length;i++){
        if(boxes[i].checked){
            checkedBox=i
            break // No need to check the rest since only one can be checked.
        }
    }
}

```

```

    }
  }
  if(checkedBox==0) {
    alert("You must select the type of task you want to do ");
    return
  }
  url = page+"?&hours=" + hours +"&minutes="+minutes;
  window.location.href = url;
}

function setHours(value) {
  hours = value;
}

function setMinutes(value) {
  minutes = value;
}

//-->
</script>
<script type="text/javascript" src="assets/p7uberlink.js"></script>
</head>

<body onload="updateClock(); setInterval('updateClock()', 1000 )"
onLoad="P7_Uberlink('uberlink','navlist')">
<div id="container">
<div id="head"><p class="pos_absolute">Welcome <?php echo $login_session; ?> | <a
class="logout" href="logout.php">logout</a></p></div>
<div id="div_clock"><span id="clock">&nbsp;</span></div>
<div id="content"><h2><a class="menu" href="home.php">Daily Tasks</a> -> Add Task</h2>
<div id="myFormDiv">
<form name="f1">

<style type='text/css'>
  fieldset
  {
    padding: 1em;
    font:80%/1 sans-serif;

    margin-left: auto;
    margin-right: auto;
  }
  label {
    float:left;
    width:25%;
    margin-right:0.5em;
    padding-top:0.2em;
    text-align:right;
    font-weight:bold;
  }
</style>
<div class="input_radio" >
  <fieldset>
    <legend><b>Select the type of task: </b></legend>
    <input type="hidden" name="group1" value="" id="radiobutton" />

```

```

    <input type="radio" name="group1" id="radiobutton1" value="1" onClick =GetSelectedItem()
/>
    <label for="radiobutton1">Gas Station</label>
    <br />
    <input type="radio" name="group1" id="radiobutton2" value="2" onClick =GetSelectedItem()
/>
    <label for="radiobutton2">Supermarket</label>
    <br />
    <input type="radio" name="group1" id="radiobutton3" value="3" onClick =GetSelectedItem()
/>
    <label for="radiobutton3">Cinema</label>
</fieldset>
    <br>
    <fieldset>
        <legend style="height: px"><b>Select the time you want to do this task:
</b></legend>
        <select name="hours" onChange='setHours(this.value);'>
            <option>01</option>
            <option>02</option>
            <option>03</option>
            <option>04</option>
            <option>05</option>
            <option>06</option>
            <option>07</option>
            <option>08</option>
            <option>09</option>
            <option>10</option>
            <option>11</option>
            <option>12</option>
            <option>13</option>
            <option>14</option>
            <option>15</option>
            <option>16</option>
            <option>17</option>
            <option>18</option>
            <option>19</option>
            <option>20</option>
            <option>21</option>
            <option>22</option>
            <option>23</option>
        </select>
        <b>:</b>
        <select name="minutes" onChange='setMinutes(this.value);'>
            <option>00</option>
            <option>10</option>
            <option>20</option>
            <option>30</option>
            <option>40</option>
            <option>50</option>
        </select>
    </fieldset>
    <br>
    <button type="button" name="next" style="width: 150px;"
onclick="nextPage()">Next</button>
</form>

```

```
</div>

</div>
</div>
</div>
</body>

</html>
```

### Κώδικας προσθήκης σινεμά

```
<?php
include('lock.php');
?>
<!DOCTYPE html ">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
  <title>Add Cinema </title>
  <script type="text/javascript"
src="http://maps.google.com/maps/api/js?sensor=false"></script>
  <script type="text/javascript" src="util.js"></script>
  <script type="text/javascript">
  //<![CDATA[

    var marker;
    var marker2;
    var directionDisplay;
    var directionsService = new google.maps.DirectionsService();
    var map;
    var allCinema = new Array();
    var openCinema = new Array();
    var startTime;
    var endTime;
    var userLat;
    var userLng;
    var cinemaName;
    var hours = getParam("hours");
    var minutes = getParam("minutes");

    var iconCinema = 'cinema.png';
    var iconCinemaClosed = 'cinema_red.png';
    var iconRed = 'http://labs.google.com/ridefinder/images/mm_20_red.png';

    function SetJsValues() {
      if(startTime == undefined || endTime == undefined) {
        alert("You must first click on the map");
        return false;
      }
    }
```

```

        else {
            window.location.href = "saveTaskCinema.php?&startTime=" + startTime
+ "&endTime=" + endTime + "&userLat=" + userLat + "&userLng=" + userLng
+ "&cinemaName=" + cinemaName + "&hours=" + hours + "&minutes=" + minutes;
            return true;
        }
    }

function initialize() {
    var myLatLng = new google.maps.LatLng(47.6063889, -122.3308333);
    var myOptions = {
        center: myLatLng,
        zoom: 13,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    }

    map = new google.maps.Map(document.getElementById("map"), myOptions);

    directionsDisplay = new google.maps.DirectionsRenderer();
    directionsDisplay.setMap(map);
    directionsDisplay.setOptions({suppressMarkers: true, preserveViewport: true});

    // Load markers from database and pace them to map
    downloadUrl("phpsqlajax_genxml.php", function(data) {
        var markers = data.documentElement.getElementsByTagName("marker");
        var j=0;
        for (var i = 0; i < markers.length; i++) {
            var type = markers[i].getAttribute("type");
            if(type=="Cinema") {
                var name = markers[i].getAttribute("name");
                var address = markers[i].getAttribute("address");
                var point = new
google.maps.LatLng(parseFloat(markers[i].getAttribute("lat")),
                parseFloat(markers[i].getAttribute("lng")));
                var openTimeStr = markers[i].getAttribute("open_time");
                var closeTimeStr = markers[i].getAttribute("close_time");

                var openTime = new Date();
                var closeTime = new Date();
                openTime.setHours(openTimeStr.substring(0,2));
                openTime.setMinutes(openTimeStr.substring(3,5));
                openTime.setSeconds(openTimeStr.substring(6,8));

                closeTime.setHours(closeTimeStr.substring(0,2));
                closeTime.setMinutes(closeTimeStr.substring(3,5));
                closeTime.setSeconds(closeTimeStr.substring(6,8));

                var myMarker = new CreateMarker(name, address, point,
openTime, closeTime);

                allCinema[j] = myMarker;
                j++;
            }
        }
        updateMarkers();
    });
}

```



```

});

google.maps.event.addListener(map, 'click', function(event) {
    if (marker2 == null) {
        placeMarker(event.latLng);
        userLat = marker2.getPosition().lat();
        userLng = marker2.getPosition().lng();
        google.maps.event.addListener(marker2, 'dragend', function(event)
{
            updateMarkersIn(event.latLng);
            closest_marker = find_closest_marker(event);
            if(closest_marker == undefined) {
                alert("All cinemas are closed!");
                return;
            }
            cinemaName = closest_marker.name;
            calcRoute(event.latLng , closest_marker.point);
        });
    }
    else {
        marker2.setPosition(event.latLng);
        userLat = marker2.getPosition().lat();
        userLng = marker2.getPosition().lng();
    }
    updateMarkersIn(event.latLng);
    closest_marker = find_closest_marker(event);
    if(closest_marker == undefined) {
        alert("All cinemas are closed!");
        return;
    }
    cinemaName = closest_marker.name;
    calcRoute(event.latLng , closest_marker.point );
});

}

function placeMarker(location) {
    marker2 = new google.maps.Marker({
        position: location,
        map: map,
        draggable: true
    });

    map.setCenter(location);
}

function updateMarkersIn(start) {
    var arriveDate;
    var currentDate = new Date();

    openCinema.length = 0;
    //Mark with red the cinema which will be closed when arrive
    for (var i = 0; i < allCinema.length; i++) {
        arriveDate = calcArriveDate(start, allCinema[i].point);
    }
}

```

```

        if(!allCinema[i].isOpenIn(arriveDate)) {
            allCinema[i].getMarker().setIcon('cinema_red.png');
        }
        else {
            allCinema[i].getMarker().setIcon('cinema.png');
            openCinema.push(allCinema[i]);
        }
    }
}

function updateMarkers() {
    openCinema.length = 0;
    for (var i = 0; i < allCinema.length; i++) {
        if(!allCinema[i].isOpen()) {
            allCinema[i].getMarker().setIcon('cinema_red.png');
        }
        else {
            openCinema.push(allCinema[i]);
        }
    }
}

function calcRoute(start, end) {

    var request = {
        origin:start,
        destination:end,
        travelMode: google.maps.DirectionsTravelMode.DRIVING
    };
    directionsService.route(request, function(result, status) {
        if (status == google.maps.DirectionsStatus.OK) {
            // Display the route
            directionsDisplay.setDirections(result);

            var arriveDate = calcArriveDate(start, end);
            var strArriveTime =
arriveDate.getHours()+":"+arriveDate.getMinutes()+":"+arriveDate.getSeconds() ;
            var timeinSeconds = calcTime(start, end);
            distance = result.routes[0].legs[0].distance.value;
            document.getElementById('duration').innerHTML = "Duration: "
+ timeinSeconds + " Seconds";
            document.getElementById('distance').innerHTML = "Distance: " +
distance + " meters";
            document.getElementById('arrive_time').innerHTML = "Arrive
Time: " + strArriveTime;
            endTime = strArriveTime;

        }
    });
}

function calcTime(start, end) {
    d = calcDistance(start, end);
    timeInSecs = d / 17;
    var startDate = new Date();
    startDate.setHours(hours);
}

```

```

        startDate.setMinutes(minutes);
        startDate.setSeconds(00);
        //if time is 8:00-9:00 or 13:00-15:00 (rush hour) arrive time will be slower
        if(isRushHour(startDate)) {
            timeInSecs = timeInSecs * 2;
        }
        return Math.round(timeInSecs*100)/100;
    }

function calcArriveDate(start , end) {
    d = calcDistance(start, end);
    // Distance / 17 m/s
    timeInSecs = d / 17;
    var startDate = new Date();
    startDate.setHours(hours);
    startDate.setMinutes(minutes);
    startDate.setSeconds(00);
    startTime =
startDate.getHours()+":"+startDate.getMinutes()+":"+startDate.getSeconds() ;
    //if time is 8:00-9:00 or 13:00-15:00 (rush hour) arrive time will be slower
    if(isRushHour(startDate)) {
        timeInSecs = timeInSecs * 2;
    }
    timeInSecs = Math.round(timeInSecs*100)/100;
    time = new secondsToHms(timeInSecs);

    startDate.setHours(startDate.getHours() + time.hours);
    startDate.setMinutes(startDate.getMinutes() + time.minutes);
    startDate.setSeconds(startDate.getSeconds() + time.seconds);

    var arriveDate = startDate;
    return arriveDate;
}

function calcDistance(start, end) {
    var lat = start.lat();
    var lng = start.lng();
    var R = 6371;
    var mlat = end.lat();
    var mlng = end.lng();
    var dLat = rad(mlat - lat);
    var dLong = rad(mlng - lng);
    var a = Math.sin(dLat/2) * Math.sin(dLat/2) +
Math.cos(rad(lat)) * Math.cos(rad(lat)) * Math.sin(dLong/2) * Math.sin(dLong/2);
    var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
    var d = R * c *1000;
    return Math.round(d*100)/100;
}

function CreateMarker(name, address, point, openTime, closeTime) {
    this.name = name;
    this.address = address;
    this.point = point;
    this.openTime = openTime;
    this.closeTime = closeTime;
}

```

```

var marker = new google.maps.Marker({
    position: point,
    map: map,
    icon: iconCinema
});

google.maps.event.addListener(marker, 'click', function() {
    infowindow.open(map, marker);
});

var html = "<font size='5'>" + this.name + "</font><br/>"
    + "<b>Address:</b> " + this.address + "<br />"
    + "<b>Open Time:</b> " + getClockTime(this.openTime) + "<br />"
    + "<b>Close Time:</b> " + getClockTime(this.closeTime);
var infowindow = new google.maps.InfoWindow({
    content: html
});

this.getMarker = function() {
    return marker;
}

this.isOpen = function() {
    var now = new Date();
    if(now >= openTime && now < closeTime) {
        //alert(now+" > " +openTime + " && "+now + " < "+ closeTime+" .
True");
        return true;
    }
    else {
        //alert(now+" > " +openTime + " && "+now + " < "+ closeTime+" .
False");
        return false;
    }
}

this.isOpenIn = function(date) {
    if(date >= openTime && date < closeTime) {
        //alert(date+" > " +openTime + " && "+date + " < "+ closeTime+" .
True");
        return true;
    }
    else {
        //alert(date+" > " +openTime + " && "+date + " < "+ closeTime+" .
False");
        return false;
    }
}

}

//Haversine formula
function rad(x) {return x*Math.PI/180;}

```

```

function find_closest_marker( event ) {
    var lat = event.latLng.lat();
    var lng = event.latLng.lng();
    var R = 6371;
    var distances = [];
    var closest = -1;
    for( i=0;i<openCinema.length; i++ ) {
        var mlat = openCinema[i].point.lat();
        var mlng = openCinema[i].point.lng();
        var dLat = rad(mlat - lat);
        var dLong = rad(mlng - lng);
        var a = Math.sin(dLat/2) * Math.sin(dLat/2) +
Math.cos(rad(lat)) * Math.cos(rad(lat)) * Math.sin(dLong/2) * Math.sin(dLong/2);
        var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
        var d = R * c;
        distances[i] = d;

        if ( closest == -1 || d < distances[closest] ) {
            closest = i;
        }
    }

    return openCinema[closest];
}

function isRushHour(thisDate) {
    //08:00-09:00 & 13:00-15:00 is rush hour
    var rushHour1 = new Date();
    var rushHour2 = new Date();
    var rushHour3 = new Date();
    var rushHour4 = new Date();

    rushHour1.setHours(8);
    rushHour1.setMinutes(00);
    rushHour1.setSeconds(00);

    rushHour2.setHours(9);
    rushHour2.setMinutes(00);
    rushHour2.setSeconds(00);

    rushHour3.setHours(13);
    rushHour3.setMinutes(00);
    rushHour3.setSeconds(00);

    rushHour4.setHours(15);
    rushHour4.setMinutes(00);
    rushHour4.setSeconds(00);

    if(((thisDate >= rushHour1 && thisDate <= rushHour2)
        || (thisDate >= rushHour3 && thisDate <= rushHour4)) {
        return true;
    }
    return false;
}

function secondsToHms(secs) {

```

```

        d = Number(secs);
        this.hours = Math.floor(d / 3600);
        this.minutes = Math.floor(d % 3600 / 60);
        this.seconds = Math.floor(d % 3600 % 60);
    }

    function getClockTime(date) {
        var now = date;
        var hour = now.getHours();
        var minute = now.getMinutes();
        var second = now.getSeconds();
        var ap = "AM";
        if (hour > 11) { ap = "PM"; }
        if (hour > 12) { hour = hour - 12; }
        if (hour == 0) { hour = 12; }
        if (hour < 10) { hour = "0" + hour; }
        if (minute < 10) { minute = "0" + minute; }
        if (second < 10) { second = "0" + second; }
        var timeString = hour +
        ':' +
        minute +
        ':' +
        second +
        " " +
        ap;

        return timeString;
    }

    function getParam(name) {
        name = name.replace(/[\[]/, "\\["].replace(/[\]]/, "\\]");
        var regexS = "[\\?&]" + name + "=(^&#)*";
        var regex = new RegExp( regexS );
        var results = regex.exec( window.location.href );
        if( results == null )
            return "";
        else
            return results[1];
    }

//]]>
</script>
<style type="text/css">

a.logout:link { color:#5B98FE; text-decoration: none;}
a.logout:visited {text-decoration: none}
a.logout:active {text-decoration: none}
a.logout:hover {color: #F14C4D; text-decoration: none;}

a.menu:link { color:#DE6E45; text-decoration: none;}
a.menu:visited {text-decoration: none}
a.menu:active {text-decoration: none}
a.menu:hover {color: #C9C9C9; text-decoration: none;}

body{
background-color: #e7e7e7;

```

```

height: 100%;
margin: 0px;
padding: 0px;
}

html { height: 100% }

p.pos_absolute {
font-size:120%;
position:absolute;
top:3px;
right:170px;
}
p.pos_absolute2 {
font-size:100%;
position:absolute;
top:65px;
right:170px;
}

#sidebar {
position: relative;
left: 25px;
top: 100px;
width: 150px;
font-size: .8em;
}

#navlist {
margin: 0; /*removes indent IE and Opera*/
padding: 0; /*removes indent Mozilla and NN7*/
list-style-type: none; /*turns off display of bullet*/
}

#navlist a, #navlist a:visited {
display: block;
padding: 2px 2px 2px 24px;
border: 1px solid;
border-color: #ddd #000 #000 #ddd;
background-color: #999999;
background-image: url(img/l1_down.jpg);
background-repeat: no-repeat;
background-position: 0% 50%;
color: #EEE;
text-decoration: none;
}

#navlist a:hover {
background-color: #9F6F9F;
background-image: url(img/l1_over.jpg);
color: #2D1E2D;
}

#sidebar #navlist .uberlink {
background-color: #9F6F9F;
background-image: url(assets/l1_over.jpg);

```

```
background-repeat: no-repeat;
background-position: 0% 50%;
color: #333 !important;
font-weight: bold;
cursor: default;
}

#container {
border: 1px solid #666666;
text-align: center;
margin-left: 100px;
margin-right: 100px;
margin-top: 0px;
padding-bottom: 200px;
background-color: white;
}

#div_clock {
font-size: 100%;
position: absolute;
top: 90px;
right: 170px;
}

#head {
height: 30px;
margin: 20px 20px 5px 20px;
padding: 0px;
}

#content {
width: 80%;
height: 350px;
border: 1px solid #666666;
text-align: center;
margin-left: auto;
margin-right: auto;
margin-top: 110px;
padding-bottom: 600px;
background-color: #E5F4F6;
}

#myFormDiv {
margin-left: auto;
margin-right: auto;
background-color: #E5F4F6;
}

.input_radio {
width: 90%;
padding-top: 50px;
margin-left: auto;
margin-right: auto;
}
```



```

#map {
    margin-left: auto;
    margin-right: auto;
    position: relative;
}

</style>

<script type="text/javascript">
<!--
weekday=new Array("Sunday","Monday","Tuesday","Wednesday","Thursday",
    "Friday","Saturday");
var mydate=new Date();
var year=mydate.getYear();
if (year < 1000)
year+=1900;
var day=mydate.getDay();
var month=mydate.getMonth()+1;
if (month<10)
month="0"+month;
var daym=mydate.getDate();
if (daym<10)
daym="0"+daym;
document.write("<p class='pos_absolute2'>" +weekday[mydate.getDay()]+
"+daym+ "/" +month+ "/" +year+"</p>");
//var currentDateString = weekday[mydate.getDay()]+ " "+month+ "/" +daym+ "/" +year;
//document.getElementById("date").firstChild.nodeValue = currentDateString;

function updateClock ( )
{
    var currentTime = new Date ( );

    var currentHours = currentTime.getHours ( );
    var currentMinutes = currentTime.getMinutes ( );
    var currentSeconds = currentTime.getSeconds ( );

    // Pad the minutes and seconds with leading zeros, if required
    currentMinutes = ( currentMinutes < 10 ? "0" : "" ) + currentMinutes;
    currentSeconds = ( currentSeconds < 10 ? "0" : "" ) + currentSeconds;

    // Choose either "AM" or "PM" as appropriate
    var timeOfDay = ( currentHours < 12 ) ? "AM" : "PM";

    // Convert the hours component to 12-hour format if needed
    currentHours = ( currentHours > 12 ) ? currentHours - 12 : currentHours;

    // Convert an hours component of "0" to "12"
    currentHours = ( currentHours == 0 ) ? 12 : currentHours;

    // Compose the string for display
    var currentTimeString = currentHours + ":" + currentMinutes + ":" + currentSeconds + " " +
timeOfDay;

```

```

// Update the time display
document.getElementById("clock").firstChild.nodeValue = currentTimeString;
}

//-->
</script>

</head>

<body onload="updateClock(); setInterval('updateClock()', 1000 ); initialize()">
<div id="container">
<div id="head"><p class="pos_absolute">Welcome <?php echo $login_session; ?> | <a
class="logout" href="logout.php">logout</a></p></div>
<div id="div_clock"><span id="clock">&nbsp;</span></div>
<div id="content"><h2><a class="menu" href="home.php">Daily Tasks</a> -> <a class="menu"
href="add.php">Add Task</a> -> Cinema </h2>
<br>
<h3>Click on map to select your position</h3>
<div id="map" style="width:80%; height: 600px"></div>
<div id="duration">Duration: </div>
<div id="distance">Distance: </div>
<div id="arrive_time">Arrive Time: </div>
<br><br>
<form name="myForm" method="get">
<input type="button" name="saveTask" id="saveTask" value="Save Task"
onclick="SetJsValues();" style="width: 170px" />
</form>
</div>
</div>
</body>

</html>

```

### Κώδικας προσθήκης βενζινάδικου

```

<?php
include("lock.php");
?>
<!DOCTYPE html ">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
<title>Add Gas Station </title>
<script type="text/javascript"
src="http://maps.google.com/maps/api/js?sensor=false"></script>
<script type="text/javascript" src="util.js"></script>
<script type="text/javascript">
//<![CDATA[

var marker;
var marker2;
var directionDisplay;

```

```

var directionsService = new google.maps.DirectionsService();
var map;
var allGasStation = new Array();
var openGasStation = new Array();
var startTime;
var endTime;
var userLat;
var userLng;
var gasStationName;
var hours = getParam("hours");
var minutes = getParam("minutes");

var iconGasStation = 'gazstation.png';
var iconGasStationClosed = 'gazstation_red.png';
var iconRed = 'http://labs.google.com/ridefinder/images/mm_20_red.png';

function SetJsValues() {
    if(startTime == undefined || endTime == undefined) {
        alert("You must first click on the map");
        return false;
    }
    else {
        window.location.href = "saveTaskGasStation.php?&startTime=" +
startime + "&endTime=" + endTime + "&userLat=" + userLat + "&userLng=" + userLng
+"&gasStationName="+ gasStationName + "&hours="+ hours + "&minutes="+minutes;
        return true;
    }
}

function initialize() {
    var myLatLng = new google.maps.LatLng(47.6063889, -122.3308333);
    var myOptions = {
        center: myLatLng,
        zoom: 13,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    }

    map = new google.maps.Map(document.getElementById("map"), myOptions);

    directionsDisplay = new google.maps.DirectionsRenderer();
    directionsDisplay.setMap(map);
    directionsDisplay.setOptions({suppressMarkers: true, preserveViewport: true});

    // Load markers from database and pace them to map
    downloadUrl("phpsqlajax_genxml.php", function(data) {
    var markers = data.documentElement.getElementsByTagName("marker");

    for (var i = 0; i < markers.length; i++) {
        var type = markers[i].getAttribute("type");
        if(type=="Gas_station") {

```

```

        var name = markers[i].getAttribute("name");
        var address = markers[i].getAttribute("address");
        var point = new
google.maps.LatLng(parseFloat(markers[i].getAttribute("lat")),

        parseFloat(markers[i].getAttribute("lng")));
        var openTimeStr = markers[i].getAttribute("open_time");
        var closeTimeStr = markers[i].getAttribute("close_time");

        var openTime = new Date();
        var closeTime = new Date();
        openTime.setHours(openTimeStr.substring(0,2));
        openTime.setMinutes(openTimeStr.substring(3,5));
        openTime.setSeconds(openTimeStr.substring(6,8));

        closeTime.setHours(closeTimeStr.substring(0,2));
        closeTime.setMinutes(closeTimeStr.substring(3,5));
        closeTime.setSeconds(closeTimeStr.substring(6,8));
        var myMarker = new CreateMarker(name, address, point,
openTime, closeTime);

        allGasStation[i] = myMarker;
    }
}
updateMarkers();
});

google.maps.event.addListener(map, 'click', function(event) {
    if (marker2 == null) {
        placeMarker(event.latLng);
        userLat = marker2.getPosition().lat();
        userLng = marker2.getPosition().lng();
        google.maps.event.addListener(marker2, 'dragend',
function(event) {
            updateMarkersIn(event.latLng);
            closest_marker = find_closest_marker(event);
            if(closest_marker == undefined) {
                alert("All gas stations are closed!");
                return;
            }
            gasStationName = closest_marker.name;
            calcRoute(event.latLng , closest_marker.point);
        });
    }
    else {
        marker2.setPosition(event.latLng);
        userLat = marker2.getPosition().lat();
        userLng = marker2.getPosition().lng();
    }
    updateMarkersIn(event.latLng);
    closest_marker = find_closest_marker(event);
    if(closest_marker == undefined) {
        alert("All gas stations are closed!");
    }
}
});

```

```

        return;
    }
    gasStationName = closest_marker.name;
    calcRoute(event.latLng , closest_marker.point );
});

}

function placeMarker(location) {
    marker2 = new google.maps.Marker({
        position: location,
        map: map,
        draggable: true
    });

    map.setCenter(location);
}

function updateMarkersIn(start) {
    var arriveDate;
    var currentDate = new Date();

    openGasStation.length = 0;

    //Mark with red the gasStations which will be closed when arrive
    for (var i = 0; i < allGasStation.length; i++) {
        arriveDate = calcArriveDate(start, allGasStation[i].point);
        if(!allGasStation[i].isOpenIn(arriveDate)) {
            allGasStation[i].getMarker().setIcon('gazstation_red.png');
        }
        else {
            allGasStation[i].getMarker().setIcon('gazstation.png');
            openGasStation.push(allGasStation[i]);
        }
    }
}

function updateMarkers() {
    openGasStation.length = 0;

    for (var i = 0; i < allGasStation.length; i++) {
        if(!allGasStation[i].isOpen()) {
            allGasStation[i].getMarker().setIcon('gazstation_red.png');
        }
        else {
            openGasStation.push(allGasStation[i]);
        }
    }
}

```

```

function calcRoute(start, end) {

    var request = {
        origin:start,
        destination:end,
        travelMode: google.maps.DirectionsTravelMode.DRIVING
    };
    directionsService.route(request, function(result, status) {
        if (status == google.maps.DirectionsStatus.OK) {
            // Display the route
            directionsDisplay.setDirections(result);

            var arriveDate = calcArriveDate(start, end);
            var strArriveTime =
arriveDate.getHours()+":"+arriveDate.getMinutes()+":"+arriveDate.getSeconds() ;
            var timeInSeconds = calcTime(start, end);
            distance = result.routes[0].legs[0].distance.value;
            document.getElementById('duration').innerHTML = "Duration: " +
timeInSeconds + " Seconds";
            document.getElementById('distance').innerHTML = "Distance: " +
distance + " meters";
            document.getElementById('arrive_time').innerHTML = "Arrive
Time: " + strArriveTime;
            endTime = strArriveTime;

        }
    });
}

function calcTime(start, end) {
    d = calcDistance(start, end);
    timeInSecs = d / 17;
    var startDate = new Date();
    startDate.setHours(hours);
    startDate.setMinutes(minutes);
    startDate.setSeconds(00);
    //if time is 8:00-9:00 or 13:00-15:00 (rush hour) arrive time will be slower
    if(isRushHour(startDate)) {
        timeInSecs = timeInSecs * 2;
    }
    return Math.round(timeInSecs*100)/100;
}

function calcArriveDate(start , end) {
    d = calcDistance(start, end);
    // Distance / 17 m/s
    timeInSecs = d / 17;
    var startDate = new Date();
    startDate.setHours(hours);
    startDate.setMinutes(minutes);
    startDate.setSeconds(00);
}

```

```

        startTime =
startDate.getHours()+":"+startDate.getMinutes()+":"+startDate.getSeconds() ;
//if time is 8:00-9:00 or 13:00-15:00 (rush hour) arrive time will be slower
if(isRushHour(startDate)) {
    timeInSecs = timeInSecs * 2;
}
timeInSecs = Math.round(timeInSecs*100)/100;
time = new secondsToHms(timeInSecs);

startDate.setHours(startDate.getHours() + time.hours);
startDate.setMinutes(startDate.getMinutes() + time.minutes);
startDate.setSeconds(startDate.getSeconds() + time.seconds);

var arriveDate = startDate;
return arriveDate;
}

function calcDistance(start, end) {
    var lat = start.lat();
    var lng = start.lng();
    var R = 6371;
    var mlat = end.lat();
    var mlng = end.lng();
    var dLat = rad(mlat - lat);
    var dLong = rad(mlng - lng);
    var a = Math.sin(dLat/2) * Math.sin(dLat/2) +
Math.cos(rad(lat)) * Math.cos(rad(lat)) * Math.sin(dLong/2) * Math.sin(dLong/2);
    var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
    var d = R * c * 1000;
    return Math.round(d*100)/100;
}

function CreateMarker(name, address, point, openTime, closeTime) {
    this.name = name;
    this.address = address;
    this.point = point;
    this.openTime = openTime;
    this.closeTime = closeTime;

    var marker = new google.maps.Marker({
        position: point,
        map: map,
        icon: iconGasStation
    });

    google.maps.event.addListener(marker, 'click', function() {
        infowindow.open(map, marker);
    });

    var html = "<font size='5'>" + this.name + "</font><br/>"

```

```

        + "<b>Address:</b> " + this.address + "<br />"
        + "<b>Open Time:</b> " + getClockTime(this.openTime)+ "<br />"
        + "<b>Close Time:</b> " + getClockTime(this.closeTime);
var infowindow = new google.maps.InfoWindow({
    content: html
});

this.getMarker = function() {
    return marker;
}

this.isOpen = function() {
    var now = new Date();
    if(now >= openTime && now < closeTime) {
        //alert(now+" > " +openTime +" && "+now +" < "+ closeTime+" .
True");
        return true;
    }
    else {
        //alert(now+" > " +openTime +" && "+now +" < "+ closeTime+" .
False");
        return false;
    }
}

this.isOpenIn = function(date) {
    if(date >= openTime && date < closeTime) {
        //alert(date+" > " +openTime +" && "+date +" < "+ closeTime+" .
True");
        return true;
    }
    else {
        //alert(date+" > " +openTime +" && "+date +" < "+ closeTime+" .
False");
        return false;
    }
}

}

//Haversine formula
function rad(x) {return x*Math.PI/180;}
function find_closest_marker( event ) {
    var lat = event.latLng.lat();
    var lng = event.latLng.lng();
    var R = 6371;
    var distances = [];
    var closest = -1;
    for( i=0;i<openGasStation.length; i++ ) {
        var mlat = openGasStation[i].point.lat();
        var mlng = openGasStation[i].point.lng();

```



```

        var dLat = rad(mlat - lat);
        var dLong = rad(mlng - lng);
        var a = Math.sin(dLat/2) * Math.sin(dLat/2) +
Math.cos(rad(lat)) * Math.cos(rad(lat)) * Math.sin(dLong/2) * Math.sin(dLong/2);
        var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
        var d = R * c;
        distances[i] = d;

        if ( closest == -1 || d < distances[closest] ) {
            closest = i;
        }
    }

    return openGasStation[closest];
}

function isRushHour(thisDate) {
    //08:00-09:00 & 13:00-15:00 is rush hour
    var rushHour1 = new Date();
    var rushHour2 = new Date();
    var rushHour3 = new Date();
    var rushHour4 = new Date();

    rushHour1.setHours(8);
    rushHour1.setMinutes(00);
    rushHour1.setSeconds(00);

    rushHour2.setHours(9);
    rushHour2.setMinutes(00);
    rushHour2.setSeconds(00);

    rushHour3.setHours(13);
    rushHour3.setMinutes(00);
    rushHour3.setSeconds(00);

    rushHour4.setHours(15);
    rushHour4.setMinutes(00);
    rushHour4.setSeconds(00);

    if((thisDate >= rushHour1 && thisDate <= rushHour2)
        || (thisDate >= rushHour3 && thisDate <= rushHour4)) {
        return true;
    }
    return false;
}

function secondsToHms(secs) {
    d = Number(secs);
    this.hours = Math.floor(d / 3600);
    this.minutes = Math.floor(d % 3600 / 60);
    this.seconds = Math.floor(d % 3600 % 60);
}

```

```

function getClockTime(date)  {
    var now  = date;
    var hour  = now.getHours();
    var minute = now.getMinutes();
    var second = now.getSeconds();
    var ap = "AM";
    if (hour > 11) { ap = "PM";      }
    if (hour > 12) { hour = hour - 12;  }
    if (hour == 0) { hour = 12;      }
    if (hour < 10) { hour = "0" + hour; }
    if (minute < 10) { minute = "0" + minute; }
    if (second < 10) { second = "0" + second; }
    var timeString = hour +
    ':' +
    minute +
    ':' +
    second +
    " " +
                                ap;
    return timeString;
}

function getParam(name) {
    name = name.replace(/[\]/,"\\[").replace(/[\]/,"\\]");
    var regexS = "[\\?&]" + name + "=(^&#)*";
    var regex = new RegExp( regexS );
    var results = regex.exec( window.location.href );
    if( results == null )
        return "";
    else
        return results[1];
}

//]]>
</script>
<style type="text/css">

a.logout:link {color:#5B98FE; text-decoration: none;}
a.logout:visited {text-decoration: none}
a.logout:active {text-decoration: none}
a.logout:hover {color: #F14C4D; text-decoration: none;}

a.menu:link {color:#DE6E45; text-decoration: none;}
a.menu:visited {text-decoration: none}
a.menu:active {text-decoration: none}
a.menu:hover {color: #C9C9C9; text-decoration: none;}

body{
background-color: #e7e7e7;
height: 100%;

```

```

margin: 0px;
padding: 0px;
}

html { height: 100% }

p.pos_absolute {
font-size:120%;
position:absolute;
top:3px;
right:170px;
}
p.pos_absolute2 {
font-size:100%;
position:absolute;
top:65px;
right:170px;
}

#sidebar {
position: relative;
left: 25px;
top: 100px;
width: 150px;
font-size: .8em;
}

#navlist {
margin: 0; /*removes indent IE and Opera*/
padding: 0; /*removes indent Mozilla and NN7*/
list-style-type: none; /*turns off display of bullet*/
}

#navlist a, #navlist a:visited {
display: block;
padding: 2px 2px 2px 24px;
border: 1px solid;
border-color: #ddd #000 #000 #ddd;
background-color: #999999;
background-image: url(img/l1_down.jpg);
background-repeat: no-repeat;
background-position: 0% 50%;
color: #EEE;
text-decoration: none;
}

#navlist a:hover {
background-color: #9F6F9F;
background-image: url(img/l1_over.jpg);
color: #2D1E2D;
}

```

```
#sidebar #navlist .uberlink {
  background-color: #9F6F9F;
  background-image: url(assets/l1_over.jpg);
  background-repeat: no-repeat;
  background-position: 0% 50%;
  color: #333 !important;
  font-weight: bold;
  cursor: default;
}
```

```
#container {
  border: 1px solid #666666;
  text-align: center;
  margin-left: 100px;
  margin-right: 100px;
  margin-top: 0px;
  padding-bottom: 200px;
  background-color: white;
}
```

```
#div_clock {
  font-size:100%;
  position:absolute;
  top:90px;
  right:170px;
}
```

```
#head {
  height: 30px;
  margin: 20px 20px 5px 20px;
  padding: 0px;
}
```

```
#content {
  width: 80%;
  height: 350px;
  border: 1px solid #666666;
  text-align: center;
  margin-left: auto;
  margin-right: auto;
  margin-top: 110px;
  padding-bottom: 600px;
  background-color: #E5F4F6;
}
```

```
#myFormDiv {
  margin-left: auto;
  margin-right: auto;
  background-color: #E5F4F6;
}
```

```

.input_radio {
    width: 90%;
    padding-top: 50px;
    margin-left: auto;
    margin-right: auto;
}

#map {
    margin-left: auto;
    margin-right: auto;
    position: relative;
}

</style>

<script type="text/javascript">
<!--
weekday=new Array("Sunday","Monday","Tuesday","Wednesday","Thursday",
    "Friday","Saturday");
var mydate=new Date();
var year=mydate.getYear();
if (year < 1000)
year+=1900;
var day=mydate.getDay();
var month=mydate.getMonth()+1;
if (month<10)
month="0"+month;
var daym=mydate.getDate();
if (daym<10)
daym="0"+daym;
document.write("<p class='pos_absolute2'>" +weekday[mydate.getDay()]+
"+daym+"/"+"month+"/"+"year+"</p>");
//var currentDateString = weekday[mydate.getDay()]+ " "+month+"/"+daym+"/"+year;
//document.getElementById("date").firstChild.nodeValue = currentDateString;

function updateClock ( )
{
    var currentTime = new Date ( );

    var currentHours = currentTime.getHours ( );
    var currentMinutes = currentTime.getMinutes ( );
    var currentSeconds = currentTime.getSeconds ( );

    // Pad the minutes and seconds with leading zeros, if required
    currentMinutes = ( currentMinutes < 10 ? "0" : "" ) + currentMinutes;
    currentSeconds = ( currentSeconds < 10 ? "0" : "" ) + currentSeconds;

    // Choose either "AM" or "PM" as appropriate
    var timeOfDay = ( currentHours < 12 ) ? "AM" : "PM";

```

```

// Convert the hours component to 12-hour format if needed
currentHours = ( currentHours > 12 ) ? currentHours - 12 : currentHours;

// Convert an hours component of "0" to "12"
currentHours = ( currentHours == 0 ) ? 12 : currentHours;

// Compose the string for display
var currentTimeString = currentHours + ":" + currentMinutes + ":" + currentSeconds + " " +
timeOfDay;

// Update the time display
document.getElementById("clock").firstChild.nodeValue = currentTimeString;
}

//-->
</script>

</head>

<body onload="updateClock(); setInterval('updateClock()', 1000 ); initialize()">
<div id="container">
<div id="head"><p class="pos_absolute">Welcome <?php echo $login_session; ?> | <a
class="logout" href="logout.php">logout</a></p></div>
<div id="div_clock"><span id="clock">&nbsp;</span></div>
<div id="content"><h2><a class="menu" href="home.php">Daily Tasks</a> -> <a class="menu"
href="add.php">Add Task</a> -> Gas Station </h2>
<br>
<h3>Click on map to select your position</h3>
<div id="map" style="width:80%; height: 600px"></div>
<div id="duration">Duration: </div>
<div id="distance">Distance: </div>
<div id="arrive_time">Arrive Time: </div>
<br><br>
<form name="myForm" method="get">
<input type="button" name="saveTask" id="saveTask" value="Save Task" onclick="SetJsValues();"
style="width: 170px" />
</form>
</div>
</div>
</body>

</html>

```

## Κώδικας προσθήκης supermarket

```
<?php
include('lock.php');
?>
<!DOCTYPE html ">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
  <title>Add Supermarket </title>
  <script type="text/javascript"
src="http://maps.google.com/maps/api/js?sensor=false"></script>
  <script type="text/javascript" src="util.js"></script>
  <script type="text/javascript">
  //

    var marker;
    var marker2;
    var directionDisplay;
    var directionsService = new google.maps.DirectionsService();
    var map;
    var allSupermarket = new Array();
    var openSupermarket = new Array();
    var startTime;
    var endTime;
    var userLat;
    var userLng;
    var supermarketName;
    var hours = getParam("hours");
    var minutes = getParam("minutes");

var iconSupermarket = 'supermarket.png';
var iconSupermarketClosed = 'supermarket_red.png';
var iconRed = 'http://labs.google.com/ridefinder/images/mm_20_red.png';

    function SetJsValues() {
        if(startTime == undefined || endTime == undefined) {
            alert("You must first click on the map");
            return false;
        }
        else {
            window.location.href = "saveTaskSupermarket.php?&amp;startTime=" +
startTime + "&amp;endTime=" + endTime + "&amp;userLat=" + userLat + "&amp;userLng=" + userLng
+"&amp;supermarketName="+ supermarketName+"&amp;hours="+ hours +"&amp;minutes="+minutes;
            return true;
        }
    }

function initialize() {
    var myLatLng = new google.maps.LatLng(47.6063889, -122.3308333);
    var myOptions = {</pre></div><div data-bbox="832 932 865 948" data-label="Page-Footer"><p>87</p></div>
```

```

        center: myLatLng,
        zoom: 13,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    }

    map = new google.maps.Map(document.getElementById("map"), myOptions);

    directionsDisplay = new google.maps.DirectionsRenderer();
    directionsDisplay.setMap(map);
    directionsDisplay.setOptions({suppressMarkers: true, preserveViewport: true});

// Load markers from database and pace them to map
downloadUrl("phpsqlajax_genxml.php", function(data) {
    var markers = data.documentElement.getElementsByTagName("marker");
        var j=0;
    for (var i = 0; i < markers.length; i++) {
        var type = markers[i].getAttribute("type");
        if(type=="Supermarket") {
            var name = markers[i].getAttribute("name");
            var address = markers[i].getAttribute("address");
            var point = new
google.maps.LatLng(parseFloat(markers[i].getAttribute("lat")),
                parseFloat(markers[i].getAttribute("lng")));
            var openTimeStr = markers[i].getAttribute("open_time");
            var closeTimeStr = markers[i].getAttribute("close_time");

            var openTime = new Date();
            var closeTime = new Date();
            openTime.setHours(openTimeStr.substring(0,2));
            openTime.setMinutes(openTimeStr.substring(3,5));
            openTime.setSeconds(openTimeStr.substring(6,8));

            closeTime.setHours(closeTimeStr.substring(0,2));
            closeTime.setMinutes(closeTimeStr.substring(3,5));
            closeTime.setSeconds(closeTimeStr.substring(6,8));

            var myMarker = new CreateMarker(name, address, point,
openTime, closeTime);

            allSupermarket[j] = myMarker;
            j++;
        }
    }
    updateMarkers();
});

    google.maps.event.addListener(map, 'click', function(event) {
        if (marker2 == null) {
            placeMarker(event.latLng);
            userLat = marker2.getPosition().lat();
            userLng = marker2.getPosition().lng();
            google.maps.event.addListener(marker2, 'dragend',

```



```

function(event) {
    updateMarkersIn(event.latLng);
    closest_marker = find_closest_marker(event);
    if(closest_marker == undefined) {
        alert("All supermarkets are closed!");
        return;
    }
    supermarketName = closest_marker.name;
    calcRoute(event.latLng , closest_marker.point);
});
}
else {
    marker2.setPosition(event.latLng);
    userLat = marker2.getPosition().lat();
    userLng = marker2.getPosition().lng();
}
updateMarkersIn(event.latLng);
closest_marker = find_closest_marker(event);
if(closest_marker == undefined) {
    alert("All supermarkets are closed!");
    return;
}
supermarketName = closest_marker.name;
calcRoute(event.latLng , closest_marker.point );
});
}

function placeMarker(location) {
    marker2 = new google.maps.Marker({
        position: location,
        map: map,
        draggable: true
    });

    map.setCenter(location);
}

function updateMarkersIn(start) {
    var arriveDate;
    var currentDate = new Date();

    openSupermarket.length = 0;
    //Mark with red the supermarket which will be closed when arrive
    for (var i = 0; i < allSupermarket.length; i++) {
        arriveDate = calcArriveDate(start, allSupermarket[i].point);
        if(!allSupermarket[i].isOpenIn(arriveDate)) {
            allSupermarket[i].getMarker().setIcon('supermarket_red.png');
        }
        else {

```

```

        allSupermarket[i].getMarker().setIcon('Supermarket.png');
        openSupermarket.push(allSupermarket[i]);
    }
}

function updateMarkers() {
    openSupermarket.length = 0;
    for (var i = 0; i < allSupermarket.length; i++) {
        if (!allSupermarket[i].isOpen()) {
            allSupermarket[i].getMarker().setIcon('Supermarket_red.png');
        }
        else {
            openSupermarket.push(allSupermarket[i]);
        }
    }
}

function calcRoute(start, end) {

    var request = {
        origin:start,
        destination:end,
        travelMode: google.maps.DirectionsTravelMode.DRIVING
    };
    directionsService.route(request, function(result, status) {
        if (status == google.maps.DirectionsStatus.OK) {
            // Display the route
            directionsDisplay.setDirections(result);

            var arriveDate = calcArriveDate(start, end);
            var strArriveTime =
arriveDate.getHours()+":"+arriveDate.getMinutes()+":"+arriveDate.getSeconds() ;
            var timeinSeconds = calcTime(start, end);
            distance = result.routes[0].legs[0].distance.value;
            document.getElementById('duration').innerHTML = "Duration: " +
timeinSeconds + " Seconds";
            document.getElementById('distance').innerHTML = "Distance: " +
distance + " meters";
            document.getElementById('arrive_time').innerHTML = "Arrive
Time: " + strArriveTime;

            //add 40 minutes for shopping
            var endDate = arriveDate;
            endDate.setMinutes(arriveDate.getMinutes()+40);
            endTime =
endTime.getHours()+":"+endTime.getMinutes()+":"+endTime.getSeconds() ;;

        }
    });
}

function calcTime(start, end) {

```

```

        d = calcDistance(start, end);
        timeInSecs = d / 17;
        var startDate = new Date();
        startDate.setHours(hours);
        startDate.setMinutes(minutes);
        startDate.setSeconds(00);
        //if time is 8:00-9:00 or 13:00-15:00 (rush hour) arrive time will be slower
        if(isRushHour(startDate)) {
            timeInSecs = timeInSecs * 2;
        }
        return Math.round(timeInSecs*100)/100;
    }

function calcArriveDate(start , end) {
    d = calcDistance(start, end);
    // Distance / 17 m/s
    timeInSecs = d / 17;
    var startDate = new Date();
    startDate.setHours(hours);
    startDate.setMinutes(minutes);
    startDate.setSeconds(00);
    startTime =
startDate.getHours()+":"+startDate.getMinutes()+":"+startDate.getSeconds() ;
    //if time is 8:00-9:00 or 13:00-15:00 (rush hour) arrive time will be slower
    if(isRushHour(startDate)) {
        timeInSecs = timeInSecs * 2;
    }

    timeInSecs = Math.round(timeInSecs*100)/100;
    time = new secondsToHms(timeInSecs);

    startDate.setHours(startDate.getHours() + time.hours);
    startDate.setMinutes(startDate.getMinutes() + time.minutes);
    startDate.setSeconds(startDate.getSeconds() + time.seconds);

    var arriveDate = startDate;
    return arriveDate;
}

function calcDistance(start, end) {
    var lat = start.lat();
    var lng = start.lng();
    var R = 6371;
    var mlat = end.lat();
    var mlng = end.lng();
    var dLat = rad(mlat - lat);
    var dLong = rad(mlng - lng);
    var a = Math.sin(dLat/2) * Math.sin(dLat/2) +
Math.cos(rad(lat)) * Math.cos(rad(lat)) * Math.sin(dLong/2) * Math.sin(dLong/2);
    var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
    var d = R * c * 1000;
    return Math.round(d*100)/100;
}

```

```

    }

function CreateMarker(name, address, point, openTime, closeTime) {
    this.name = name;
    this.address = address;
    this.point = point;
    this.openTime = openTime;
    this.closeTime = closeTime;

    var marker = new google.maps.Marker({
        position: point,
        map: map,
        icon: iconSupermarket
    });

    google.maps.event.addListener(marker, 'click', function() {
        infowindow.open(map, marker);
    });

    var html = "<font size='5'>" + this.name + "</font><br/>"
        + "<b>Address:</b> " + this.address + "<br />"
        + "<b>Open Time:</b> " + getClockTime(this.openTime) + "<br />"
        + "<b>Close Time:</b> " + getClockTime(this.closeTime);
    var infowindow = new google.maps.InfoWindow({
        content: html
    });

    this.getMarker = function() {
        return marker;
    }

    this.isOpen = function() {
        var now = new Date();
        if(now >= openTime && now < closeTime) {
            //alert(now+" > " +openTime +" && "+now +" < "+ closeTime+" .
True");

            return true;
        }
        else {
            //alert(now+" > " +openTime +" && "+now +" < "+ closeTime+" .
False");

            return false;
        }
    }

    this.isOpenIn = function(date) {
        if(date >= openTime && date < closeTime) {
            //alert(date+" > " +openTime +" && "+date +" < "+ closeTime+" .
True");

            return true;
        }
    }
}

```

```

        else {
            //alert(date+" > "+openTime +" && "+date +" < "+ closeTime+" .
False");
            return false;
        }
    }

}

//Haversine formula
function rad(x) {return x*Math.PI/180;}
function find_closest_marker( event ) {
    var lat = event.latLng.lat();
    var lng = event.latLng.lng();
    var R = 6371;
    var distances = [];
    var closest = -1;
    for( i=0;i<openSupermarket.length; i++ ) {
        var mlat = openSupermarket[i].point.lat();
        var mlng = openSupermarket[i].point.lng();
        var dLat = rad(mlat - lat);
        var dLong = rad(mlng - lng);
        var a = Math.sin(dLat/2) * Math.sin(dLat/2) +
Math.cos(rad(lat)) * Math.cos(rad(lat)) * Math.sin(dLong/2) * Math.sin(dLong/2);
        var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
        var d = R * c;
        distances[i] = d;

        if ( closest == -1 || d < distances[closest] ) {
            closest = i;
        }
    }

    return openSupermarket[closest];
}

function isRushHour(thisDate) {
    //08:00-09:00 & 13:00-15:00 is rush hour
    var rushHour1 = new Date();
    var rushHour2 = new Date();
    var rushHour3 = new Date();
    var rushHour4 = new Date();

    rushHour1.setHours(8);
    rushHour1.setMinutes(00);
    rushHour1.setSeconds(00);

    rushHour2.setHours(9);
    rushHour2.setMinutes(00);
    rushHour2.setSeconds(00);

```

```

rushHour3.setHours(13);
rushHour3.setMinutes(00);
rushHour3.setSeconds(00);

rushHour4.setHours(15);
rushHour4.setMinutes(00);
rushHour4.setSeconds(00);

if((thisDate >= rushHour1 && thisDate <= rushHour2)
    || (thisDate >= rushHour3 && thisDate <= rushHour4)) {
    return true;
}
return false;
}

function secondsToHms(secs) {
    d = Number(secs);
    this.hours = Math.floor(d / 3600);
    this.minutes = Math.floor(d % 3600 / 60);
    this.seconds = Math.floor(d % 3600 % 60);
}

function getClockTime(date) {
    var now = date;
    var hour = now.getHours();
    var minute = now.getMinutes();
    var second = now.getSeconds();
    var ap = "AM";
    if (hour > 11) { ap = "PM"; }
    if (hour > 12) { hour = hour - 12; }
    if (hour == 0) { hour = 12; }
    if (hour < 10) { hour = "0" + hour; }
    if (minute < 10) { minute = "0" + minute; }
    if (second < 10) { second = "0" + second; }
    var timeString = hour +
    ':' +
    minute +
    ':' +
    second +
    " " +
    ap;
    return timeString;
}

function getParam(name) {
    name = name.replace(/[\[\]\/, "\\\\"/], "\\");
    var regexS = "[\\?&]" + name + "=[^&#]*";
    var regex = new RegExp( regexS );
    var results = regex.exec( window.location.href );
    if( results == null )
        return "";
}

```

```

        else
            return results[1];
    }

    //]]>
</script>
<style type="text/css">

a.logout:link {color:#5B98FE; text-decoration: none;}
a.logout:visited {text-decoration: none}
a.logout:active {text-decoration: none}
a.logout:hover {color: #F14C4D; text-decoration: none;}

a.menu:link {color:#DE6E45; text-decoration: none;}
a.menu:visited {text-decoration: none}
a.menu:active {text-decoration: none}
a.menu:hover {color: #C9C9C9; text-decoration: none;}

body{
    background-color: #e7e7e7;
    height: 100%;
    margin: 0px;
    padding: 0px;
}

html { height: 100% }

p.pos_absolute {
    font-size:120%;
    position:absolute;
    top:3px;
    right:170px;
}
p.pos_absolute2 {
    font-size:100%;
    position:absolute;
    top:65px;
    right:170px;
}

#sidebar {
    position: relative;
    left: 25px;
    top: 100px;
    width: 150px;
    font-size: .8em;
}

#navlist {
    margin: 0; /*removes indent IE and Opera*/
    padding: 0; /*removes indent Mozilla and NN7*/

```

```
list-style-type: none; /*turns off display of bullet*/
}

#navlist a, #navlist a:visited {
display: block;
padding: 2px 2px 2px 24px;
border: 1px solid;
border-color: #ddd #000 #000 #ddd;
background-color: #999999;
background-image: url(img/l1_down.jpg);
background-repeat: no-repeat;
background-position: 0% 50%;
color: #EEE;
text-decoration: none;
}

#navlist a:hover {
background-color: #9F6F9F;
background-image: url(img/l1_over.jpg);
color: #2D1E2D;
}

#sidebar #navlist .uberlink {
background-color: #9F6F9F;
background-image: url(assets/l1_over.jpg);
background-repeat: no-repeat;
background-position: 0% 50%;
color: #333 !important;
font-weight: bold;
cursor: default;
}

#container {
border: 1px solid #666666;
text-align: center;
margin-left: 100px;
margin-right: 100px;
margin-top: 0px;
padding-bottom: 200px;
background-color: white;
}

#div_clock {
font-size:100%;
position:absolute;
top:90px;
right:170px;
}

#head {
height: 30px;
```



```

        margin: 20px 20px 5px 20px;
        padding: 0px;
    }

    #content {
        width: 80%;
        height: 350px;
        border: 1px solid #666666;
        text-align: center;
        margin-left: auto;
        margin-right: auto;
        margin-top: 110px;
        padding-bottom: 600px;
        background-color: #E5F4F6;
    }

    #myFormDiv {

        margin-left: auto;
        margin-right: auto;
        background-color: #E5F4F6;
    }

    .input_radio {
        width: 90%;
        padding-top: 50px;
        margin-left: auto;
        margin-right: auto;
    }

    #map {
        margin-left: auto;
        margin-right: auto;
        position: relative;
    }

</style>

<script type="text/javascript">
<!--
weekday=new Array("Sunday","Monday","Tuesday","Wednesday","Thursday",
    "Friday","Saturday");
var mydate=new Date();
var year=mydate.getYear();
if (year < 1000)
year+=1900;
var day=mydate.getDay();
var month=mydate.getMonth()+1;
if (month<10)
month="0"+month;
-->

```

```

var daym=mydate.getDate();
if (daym<10)
daym="0"+daym;
document.write("<p class='pos_absolute2'>" +weekday[mydate.getDay()]+
"+daym+"/"+"month+"/"+"year+"</p>");
//var currentDateString = weekday[mydate.getDay()]+ " "+month+"/"+"daym+"/"+"year;
//document.getElementById("date").firstChild.nodeValue = currentDateString;

function updateClock ( )
{
var currentTime = new Date ( );

var currentHours = currentTime.getHours ( );
var currentMinutes = currentTime.getMinutes ( );
var currentSeconds = currentTime.getSeconds ( );

// Pad the minutes and seconds with leading zeros, if required
currentMinutes = ( currentMinutes < 10 ? "0" : "" ) + currentMinutes;
currentSeconds = ( currentSeconds < 10 ? "0" : "" ) + currentSeconds;

// Choose either "AM" or "PM" as appropriate
var timeOfDay = ( currentHours < 12 ) ? "AM" : "PM";

// Convert the hours component to 12-hour format if needed
currentHours = ( currentHours > 12 ) ? currentHours - 12 : currentHours;

// Convert an hours component of "0" to "12"
currentHours = ( currentHours == 0 ) ? 12 : currentHours;

// Compose the string for display
var currentTimeString = currentHours + ":" + currentMinutes + ":" + currentSeconds + " " +
timeOfDay;

// Update the time display
document.getElementById("clock").firstChild.nodeValue = currentTimeString;
}

//-->
</script>

</head>

<body onload="updateClock(); setInterval('updateClock()', 1000 ); initialize()">
<div id="container">
<div id="head"><p class="pos_absolute">Welcome <?php echo $login_session; ?> | <a
class="logout" href="logout.php">logout</a></p></div>
<div id="div_clock"><span id="clock">&nbsp;</span></div>
<div id="content"><h2><a class="menu" href="home.php">Daily Tasks</a> -> <a class="menu"
href="add.php">Add Task</a> -> Supermarket </h2>
<br>
<h3>Click on map to select your position</h3>

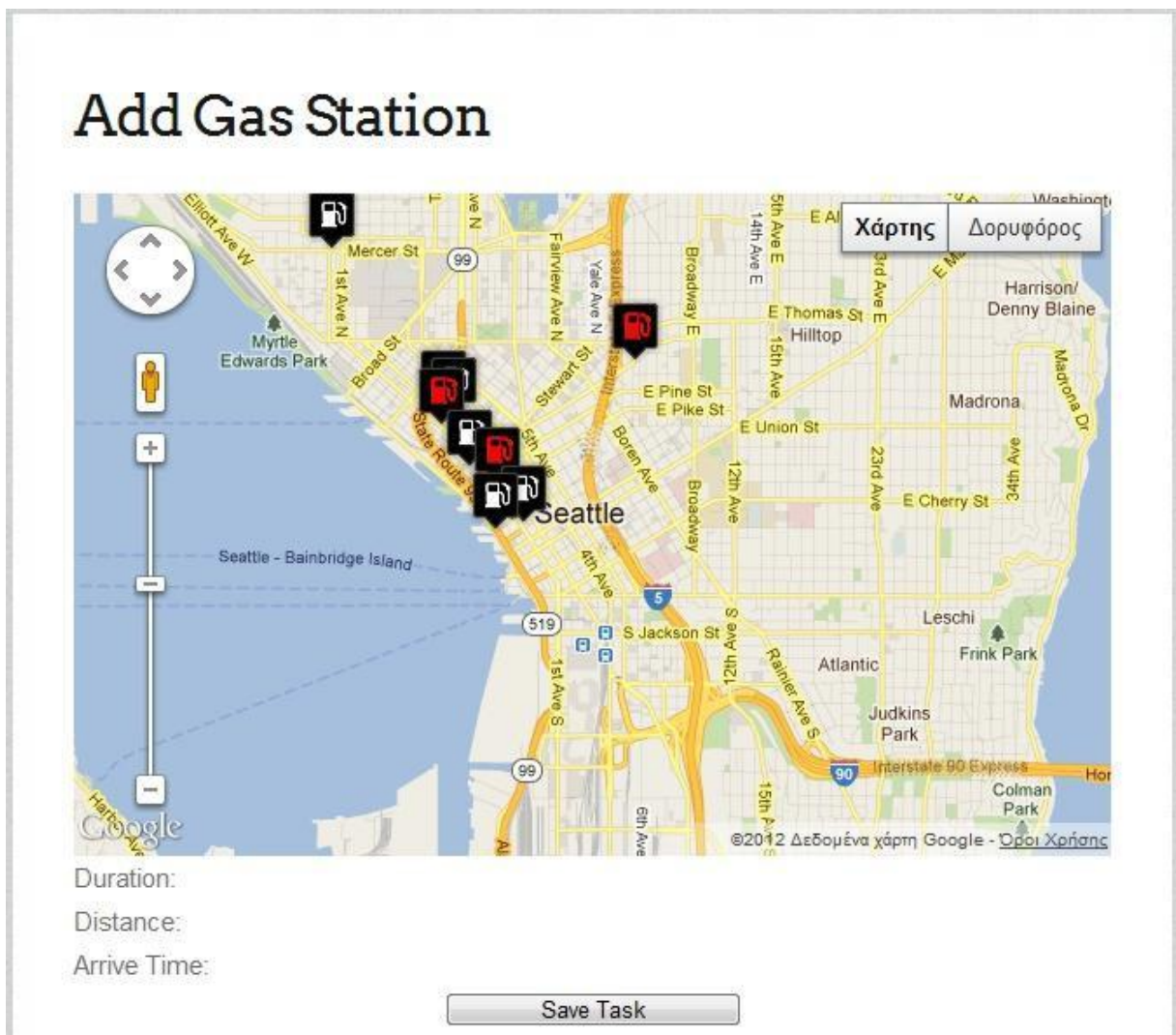
```

```

<div id="map" style="width:80%; height: 600px"></div>
<div id="duration">Duration: </div>
<div id="distance">Distance: </div>
<div id="arrive_time">Arrive Time: </div>
<br><br>
<form name="myForm" method="get">
<input type="button" name="saveTask" id="saveTask" value="Save Task" onclick="SetJsValues();"
style="width: 170px" />
</form>
</div>
</div>
</body>

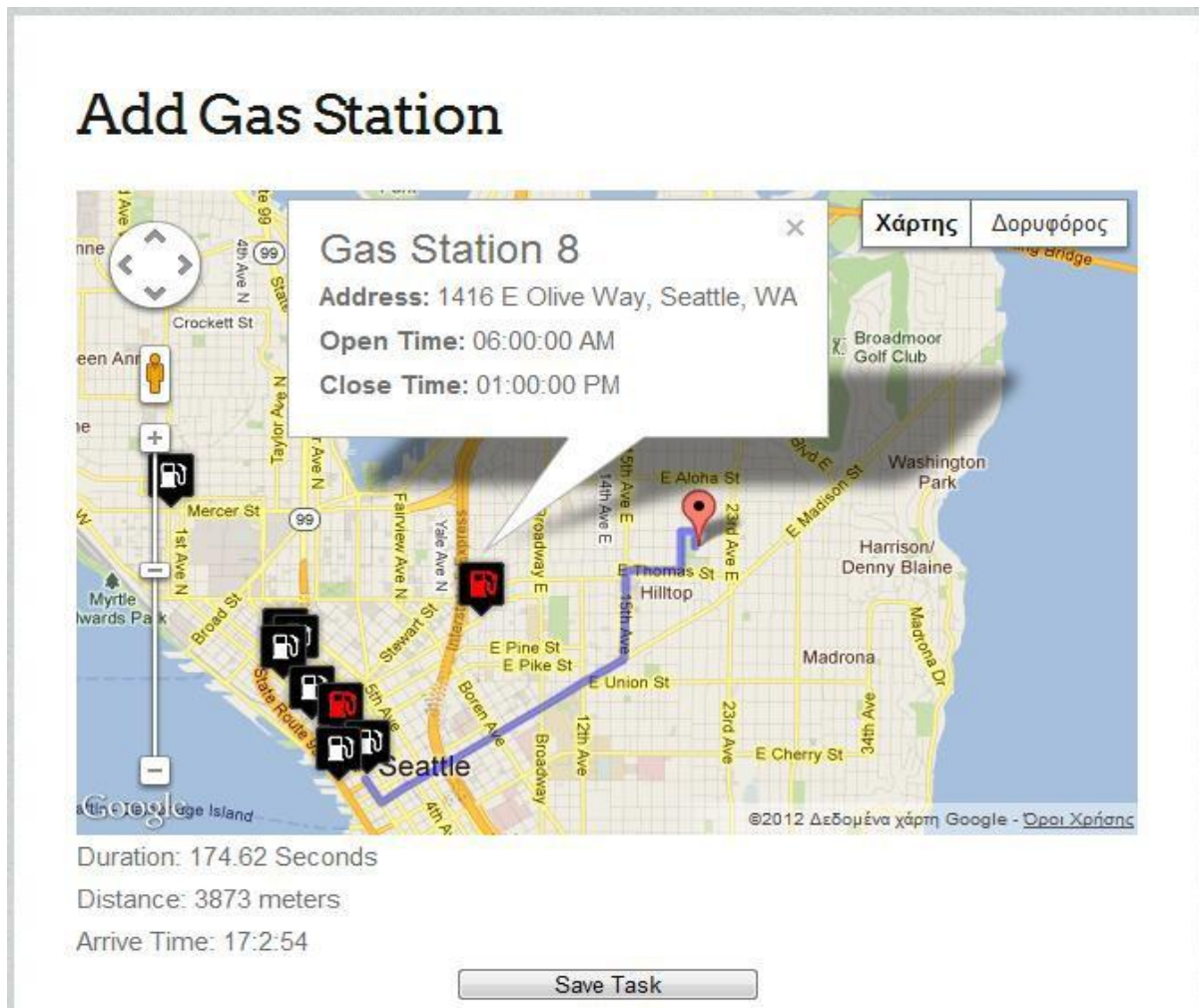
</html>

```



Εικόνα 5.7 Αποθήκευση Δραστηριότητας

Τα σημεία ενδιαφέροντος που είναι κλειστά εμφανίζονται με κόκκινο. Όταν ο χρήστης κάνει κλικ επάνω σε ένα σημείο ενδιαφέροντος εμφανίζεται ένα popup balloon που λέει το όνομα του, το ωράριο λειτουργίας του και τη διεύθυνσή του.



Εικόνα 5.7 Εμφάνιση Διαδρομής

Duration: 174.62 Seconds

Distance: 3873 meters

Arrive Time: 17:2:54

Εικόνα 5.8 Εμφάνιση Πληροφοριών Διαδρομής

## Σελίδα Διαγραφής Δραστηριότητας

Από αυτή τη σελίδα ο χρήστης μπορεί να διαγράψει όποια δραστηριότητα θέλει τσεκάροντας την και πατώντας στο κουμπί Delete.

### Κώδικας διαγραφής δραστηριότητας

```
<?php
include('lock.php');

// $query = " DELETE FROM ".$login_session."_tasks WHERE id IN (value1, value2, value3, value4,
value5)"
$query = " DELETE FROM ".$login_session."_tasks WHERE id IN (";

$firstTime=true;
foreach($_GET as $name => $value) {
    if($firstTime) {
        $query=$query.$value;
        $firstTime = false;
    }
    else {
        $query=$query.", ".$value;
    }
}

$query=$query."";

$result = mysql_query($query);

$myPage = '
<!DOCTYPE html ">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<script type="text/javascript">
alert("Tasks deleted successfully");
location = "view.php";
</script>

</head>
</html>';
echo $myPage;

?>
```

# Delete Tasks

ID	NAME	TYPE	START TIME	END TIME	CHECK
1	Gas Station 4	Gas_station	19:00:00	19:03:12	<input type="checkbox"/>
2	Cinema3	Cinema	18:30:00	18:31:46	<input type="checkbox"/>
3	Cinema3	Cinema	22:10:00	22:12:40	<input type="checkbox"/>
4	Gas Station 4	Gas_station	17:00:00	18:20:18	<input type="checkbox"/>
5	Gas Station 9	Gas_station	23:10:00	23:12:52	<input type="checkbox"/>
6	Cinema2	Cinema	16:50:00	16:50:29	<input type="checkbox"/>
7	Gas Station 8	Gas_station	07:00:00	07:02:32	<input type="checkbox"/>
8	Gas Station 8	Gas_station	07:20:00	07:27:26	<input type="checkbox"/>
10	Supermarket4	Supermarket	07:30:00	08:11:59	<input type="checkbox"/>
11	Supermarket2	Supermarket	08:20:00	09:09:34	<input type="checkbox"/>
12	Gas Station 4	Gas_station	09:20:00	09:40:05	<input type="checkbox"/>

Delete

Εικόνα 5.9 Διαγραφή Δραστηριότητας

## Σελίδα Δραστηριοτήτων

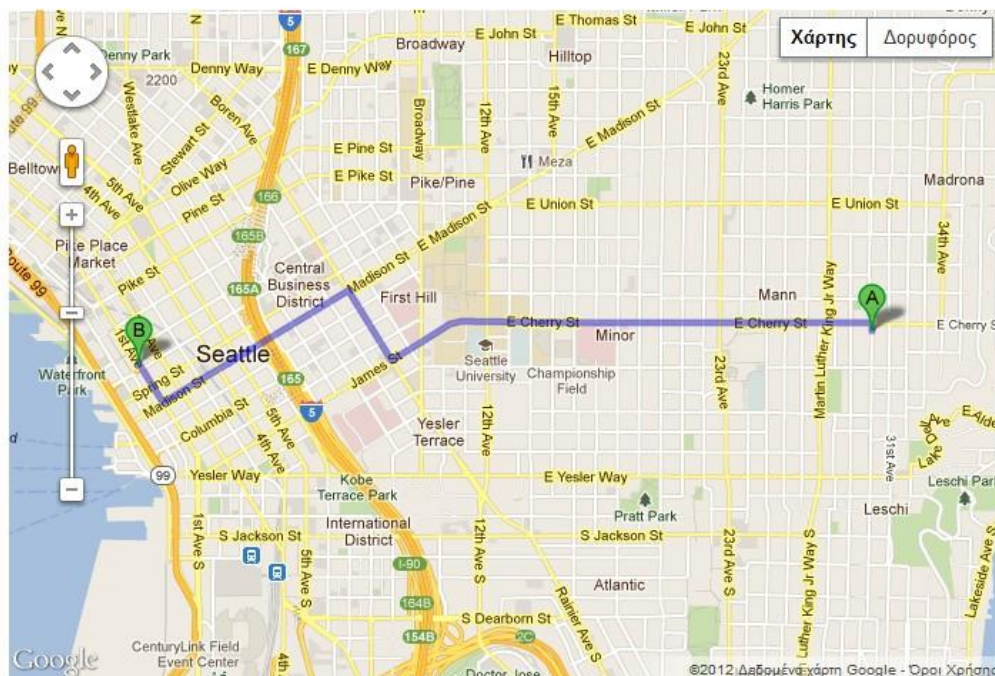
Σε αυτή τη σελίδα φαίνονται όλες οι δραστηριότητες που έχει προγραμματίσει ο χρήστης. Οι δραστηριότητες εμφανίζονται σε ένα πίνακα που δείχνει το όνομα της, τον τύπο της, την ώρα που ξεκινάει, και την ώρα που τελειώνει. Επίσης υπάρχει και μία επιλογή που δείχνει την διαδρομή που πρέπει να επακολουθήσει ο χρήστης και οδηγίες για να φτάσει εκεί.

# View Tasks


ID	NAME	TYPE	START TIME	END TIME	MAP
1	Gas Station 4	Gas_station	19:00:00	19:03:12	<a href="#">View in map</a>
2	Cinema3	Cinema	18:30:00	18:31:46	<a href="#">View in map</a>
3	Cinema3	Cinema	22:10:00	22:12:40	<a href="#">View in map</a>
4	Gas Station 4	Gas_station	17:00:00	18:20:18	<a href="#">View in map</a>
5	Gas Station 9	Gas_station	23:10:00	23:12:52	<a href="#">View in map</a>
6	Cinema2	Cinema	16:50:00	16:50:29	<a href="#">View in map</a>
7	Gas Station 8	Gas_station	07:00:00	07:02:32	<a href="#">View in map</a>
8	Gas Station 8	Gas_station	07:20:00	07:27:26	<a href="#">View in map</a>
10	Supermarket4	Supermarket	07:30:00	08:11:59	<a href="#">View in map</a>
11	Supermarket2	Supermarket	08:20:00	09:09:34	<a href="#">View in map</a>
12	Gas Station 4	Gas_station	09:20:00	09:40:05	<a href="#">View in map</a>

Εικόνα 5.10 View Tasks

## View Task



Εικόνα 5.10 Εμφάνιση διαδρομής στο χάρτη

 547-699 30th Ave, Σιάτλ, Ουάσινγκτον 98122,  
Ηνωμένες Πολιτείες της Αμερικής

2,4 μίλ. - σχετικά με 10 λεπτά

1. Head **north** on **30th Ave** toward **E Cherry** 121 πόδια  
**St**

2. Take the 1st **left** onto **E Cherry St** 1,1 μίλ.


3. Continue onto **E James Way** 0,3 μίλ.

4. Turn **right** onto **Boren Ave** 0,2 μίλ.

5. Turn **left** onto **Madison St** 0,6 μίλ.

6. Turn **right** onto **1st Ave** 0,1 μίλ.

Destination will be on the right

 1206 1st Ave, Σιάτλ, Ουάσινγκτον 98101, Ηνωμένες  
Πολιτείες της Αμερικής

©2012 Δεδομένα χάρτη Google

Εικόνα 5.11 Οδηγίες Δρομολόγησης

### Κώδικας υπολογισμού διαδρομής

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
    <title>Google Maps AJAX + MySQL/PHP </title>
    <script type="text/javascript"
src="http://maps.google.com/maps/api/js?sensor=false"></script>
    <script type="text/javascript" src="util.js"></script>
    <script type="text/javascript">
  //<![CDATA[

    var marker;
    var marker2;
    var directionDisplay;
    var directionsService = new google.maps.DirectionsService();
    var map;
    var allGasStation = new Array();
    var openGasStation = new Array();
```



```

var iconGasStation = 'gazstation.png';
    var iconGasStationClosed = 'gazstation_red.png';
    var iconRed = 'http://labs.google.com/ridefinder/images/mm_20_red.png';

function initialize() {
    var myLatLng = new google.maps.LatLng(47.6063889, -122.3308333);
    var myOptions = {
        center: myLatLng,
        zoom: 13,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    }

    map = new google.maps.Map(document.getElementById("map"), myOptions);

    directionsDisplay = new google.maps.DirectionsRenderer();
    directionsDisplay.setMap(map);
    directionsDisplay.setOptions({ suppressMarkers: true, preserveViewport: true });

    // Load markers from database and pace them to map
    downloadUrl("phpsqlajax_genxml.php", function(data) {
        var markers = data.documentElement.getElementsByTagName("marker");

        for (var i = 0; i < markers.length; i++) {
            var name = markers[i].getAttribute("name");
            var address = markers[i].getAttribute("address");
            var point = new google.maps.LatLng(parseFloat(markers[i].getAttribute("lat")),
                parseFloat(markers[i].getAttribute("lng")));
            var openTimeStr = markers[i].getAttribute("open_time");
            var closeTimeStr = markers[i].getAttribute("close_time");

            var openTime = new Date();
            var closeTime = new Date();
            openTime.setHours(openTimeStr.substring(0,2));
            openTime.setMinutes(openTimeStr.substring(3,5));
            openTime.setSeconds(openTimeStr.substring(6,8));

            closeTime.setHours(closeTimeStr.substring(0,2));
            closeTime.setMinutes(closeTimeStr.substring(3,5));
            closeTime.setSeconds(closeTimeStr.substring(6,8));
            var myMarker = new CreateMarker(name, address, point, openTime,
closeTime);

            allGasStation[i] = myMarker;
        }
        updateMarkers();
    });

    google.maps.event.addListener(map, 'click', function(event) {
        if (marker2 == null) {
            placeMarker(event.latLng);
            google.maps.event.addListener(marker2, 'dragend', function(event)
{
                updateMarkersIn(event.latLng);
                closest_marker = find_closest_marker(event);
                if(closest_marker == undefined) {

```

```

        alert("All gas stations are closed!");
        return;
    }

    calcRoute(event.latLng , closest_marker(event).point);
});
}
else {
    marker2.setPosition(event.latLng);
}
updateMarkersIn(event.latLng);
closest_marker = find_closest_marker(event);
if(closest_marker == undefined) {
    alert("All gas stations are closed!");
    return;
}
calcRoute(event.latLng , closest_marker.point );
});
}

function placeMarker(location) {
    marker2 = new google.maps.Marker({
        position: location,
        map: map,
        draggable: true
    });

    map.setCenter(location);
}

function updateMarkersIn(start) {
    var arriveDate;
    var currentDate = new Date();

    openGasStation.length = 0;

    //Mark with red the gasStations which will be closed when arrive
    for (var i = 0; i < allGasStation.length; i++) {
        arriveDate = calcArriveDate(start, allGasStation[i].point);
        if(!allGasStation[i].isOpenIn(arriveDate)) {
            allGasStation[i].getMarker().setIcon('gazstation_red.png');
        }
        else {
            allGasStation[i].getMarker().setIcon('gazstation.png');
            openGasStation.push(allGasStation[i]);
        }
    }
}

function updateMarkers() {
    openGasStation.length = 0;

    for (var i = 0; i < allGasStation.length; i++) {

```

```

        if(!allGasStation[i].isOpen()) {
            allGasStation[i].getMarker().setIcon('gazstation_red.png');
        }
        else {
            openGasStation.push(allGasStation[i]);
        }
    }
}

function calcRoute(start, end) {

    var request = {
        origin:start,
        destination:end,
        travelMode: google.maps.DirectionsTravelMode.DRIVING
    };
    directionsService.route(request, function(result, status) {
        if (status == google.maps.DirectionsStatus.OK) {
            // Display the route
            directionsDisplay.setDirections(result);

            var arriveDate = calcArriveDate(start, end);
            var timeinSeconds = calcTime(start, end);
            distance = result.routes[0].legs[0].distance.value;
            document.getElementById('duration').innerHTML = "Duration: "
+ timeinSeconds + " Seconds";
            document.getElementById('distance').innerHTML = "Distance: " +
distance + " meters";
            document.getElementById('arrive_time').innerHTML = "Arrive
Time: " + arriveDate.getHours()+":"+
arriveDate.getMinutes()+":"+arriveDate.getSeconds() ;

        }
    });
}

function calcTime(start, end) {
    d = calcDistance(start, end);
    timeInSecs = d / 17;
    var currentDate = new Date();
    //if time is 8:00-9:00 or 13:00-15:00 (rush hour) arrive time will be slower
    if(isRushHour(currentDate)) {
        timeInSecs = timeInSecs * 2;
    }
    return Math.round(timeInSecs*100)/100;
}

function calcArriveDate(start , end) {
    d = calcDistance(start, end);
    // Distance / 17 m/s
    timeInSecs = d / 17;
    var currentDate = new Date();
    //if time is 8:00-9:00 or 13:00-15:00 (rush hour) arrive time will be slower
    if(isRushHour(currentDate)) {
        timeInSecs = timeInSecs * 2;
    }
}

```

```

timeInSecs = Math.round(timeInSecs*100)/100;
time = new secondsToHms(timeInSecs);

currentDate.setHours(currentDate.getHours() + time.hours);
currentDate.setMinutes(currentDate.getMinutes() + time.minutes);
currentDate.setSeconds(currentDate.getSeconds() + time.seconds);

var arriveDate = currentDate;
return arriveDate;
}

function calcDistance(start, end) {
    var lat = start.lat();
    var lng = start.lng();
    var R = 6371;
    var mlat = end.lat();
    var mlng = end.lng();
    var dLat = rad(mlat - lat);
    var dLong = rad(mlng - lng);
    var a = Math.sin(dLat/2) * Math.sin(dLat/2) +
    Math.cos(rad(lat)) * Math.cos(rad(lat)) * Math.sin(dLong/2) * Math.sin(dLong/2);
    var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
    var d = R * c *1000;
    return Math.round(d*100)/100;
}

function CreateMarker(name, address, point, openTime, closeTime) {
    this.name = name;
    this.address = address;
    this.point = point;
    this.openTime = openTime;
    this.closeTime = closeTime;

    var marker = new google.maps.Marker({
        position: point,
        map: map,
        icon: iconGasStation
    });

    google.maps.event.addListener(marker, 'click', function() {
        infowindow.open(map, marker);
    });

    var html = "<font size='5'>" + this.name + "</font><br/>"
        + "<b>Address:</b> " + this.address + "<br />"
        + "<b>Open Time:</b> " +getClockTime(this.openTime)+ "<br />"
        + "<b>Close Time:</b> " +getClockTime(this.closeTime);
    var infowindow = new google.maps.InfoWindow({
        content: html
    });

    this.getMarker = function() {
        return marker;
    }
}

```

```

        this.isOpen = function() {
            var now = new Date();
            if(now >= openTime && now < closeTime) {
                //alert(now+" > " +openTime +" && "+now +" < "+ closeTime+" .
True");
                return true;
            }
            else {
                //alert(now+" > " +openTime +" && "+now +" < "+ closeTime+" .
False");
                return false;
            }
        }
        this.isOpenIn = function(date) {
            if(date >= openTime && date < closeTime) {
                //alert(date+" > " +openTime +" && "+date +" < "+ closeTime+" .
True");
                return true;
            }
            else {
                //alert(date+" > " +openTime +" && "+date +" < "+ closeTime+" .
False");
                return false;
            }
        }
    }

    //Haversine formula
    function rad(x) {return x*Math.PI/180;}
    function find_closest_marker( event ) {
        var lat = event.latLng.lat();
        var lng = event.latLng.lng();
        var R = 6371;
        var distances = [];
        var closest = -1;
        for( i=0;i<openGasStation.length; i++ ) {
            var mlat = openGasStation[i].point.lat();
            var mlng = openGasStation[i].point.lng();
            var dLat = rad(mlat - lat);
            var dLong = rad(mlng - lng);
            var a = Math.sin(dLat/2) * Math.sin(dLat/2) +
Math.cos(rad(lat)) * Math.cos(rad(lat)) * Math.sin(dLong/2) * Math.sin(dLong/2);
            var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
            var d = R * c;
            distances[i] = d;

            if ( closest == -1 || d < distances[closest] ) {
                closest = i;
            }
        }

        return openGasStation[closest];
    }

```

```

}

function isRushHour(currentDate) {
    //08:00-09:00 & 13:00-15:00 is rush hour
    var rushHour1 = new Date();
    var rushHour2 = new Date();
    var rushHour3 = new Date();
    var rushHour4 = new Date();

    rushHour1.setHours(8);
    rushHour1.setMinutes(00);
    rushHour1.setSeconds(00);

    rushHour2.setHours(9);
    rushHour2.setMinutes(00);
    rushHour2.setSeconds(00);

    rushHour3.setHours(13);
    rushHour3.setMinutes(00);
    rushHour3.setSeconds(00);

    rushHour4.setHours(15);
    rushHour4.setMinutes(00);
    rushHour4.setSeconds(00);

    if((currentDate >= rushHour1 && currentDate <= rushHour2)
        || (currentDate >= rushHour3 && currentDate <= rushHour4)) {
        return true;
    }
    return false;
}

function secondsToHms(secs) {
    d = Number(secs);
    this.hours = Math.floor(d / 3600);
    this.minutes = Math.floor(d % 3600 / 60);
    this.seconds = Math.floor(d % 3600 % 60);
}

function getClockTime(date) {
    var now = date;
    var hour = now.getHours();
    var minute = now.getMinutes();
    var second = now.getSeconds();
    var ap = "AM";
    if (hour > 11) { ap = "PM"; }
    if (hour > 12) { hour = hour - 12; }
    if (hour == 0) { hour = 12; }
    if (hour < 10) { hour = "0" + hour; }
    if (minute < 10) { minute = "0" + minute; }
    if (second < 10) { second = "0" + second; }
    var timeString = hour +
    ':' +
    minute +
    ':' +
    second +

```

```
        " " +
        ap;
    return timeString;
}

//]]>
</script>

</head>

<body onload="initialize()" >
  <div id="map" style="width:1350px; height:550px"></div>
    <div id="duration">Duration: </div>
  <div id="distance">Distance: </div>
    <div id="arrive_time">Arrive Time: </div>
</body>
</html>
```

# Βιβλιογραφία

- **Andrew S. Tanenbaum**, “Δικτύωση Υπολογιστών Προσέγγιση από Πάνω προς τα Κάτω”, James F. Kurose, Keith W. Ross , Εκδόσεις Μ. Γκιούρδας 2009.
- **Πώς λειτουργεί το World Wide Web**,  
[http://members.tripod.com/k\\_liak/2.html](http://members.tripod.com/k_liak/2.html)
- **Wikipedia, HTML**,  
<http://el.wikipedia.org/wiki/HTML>
- **Wikipedia, JavaScript**,  
<http://el.wikipedia.org/wiki/JavaScript>
- **Wikipedia, CSS**,  
<http://el.wikipedia.org/wiki/CSS>
- **Wikipedia, PHP**,  
<http://el.wikipedia.org/wiki/PHP>
- **Wikipedia, MySQL**,  
<http://el.wikipedia.org/wiki/MySQL>
- **Wikipedia, XML**,  
<http://el.wikipedia.org/wiki/XML>
- **Google, Google Maps API**,  
<http://code.google.com/intl/el-GR/apis/maps/index.html>
- **Google, Google Maps Javascript API V3 Reference**,  
<http://code.google.com/intl/el-GR/apis/maps/documentation/javascript/reference.html>
- **Google, Google Maps Javascript API V3 Controls**,  
<http://code.google.com/intl/el-GR/apis/maps/documentation/javascript/controls.html>
- **Google, Google Maps Javascript API V3 Events**,  
<http://code.google.com/intl/el-GR/apis/maps/documentation/javascript/events.html>