

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΤΙΤΛΟΣ:

"Σχεδίαση και Υλοποίηση Προηγμένης Εφαρμογής
Διαδικτυακών Προτάσεων Αγορών Βάσει
Πολυκριτηριακής Αναζήτησης"

Αγγελόπουλος Χρήστος
Α.Μ. 2010101

Υπεύθυνος Καθηγητής: Ν. Τσελίκας
Τρίπολη, Σεπτέμβριος 2014

Σχεδίαση και Υλοποίηση
Προηγμένης Εφαρμογής Διαδικτυακών
Προτάσεων Αγορών
Βάσει Πολυκριτηριακής Αναζήτησης

Περίληψη

Σκοπός της πτυχιακής αυτής εργασίας είναι η δημιουργία μιας ιστοσελίδας της οποίας να παρακολουθείται και να ενημερώνεται δυναμικά το περιεχόμενο και πιο συγκεκριμένα μιας ιστοσελίδας ηλεκτρονικού καταστήματος.

Η ιστοσελίδα αυτή θα πρέπει να ενημερώνεται δυναμικά και να αλλάζει το περιεχόμενό της, σε περιπτώσεις όπως π.χ. όταν κάποιος έμπορος προσθέσει κάποιο καινούριο προϊόν στο κατάστημα του ή όταν κάποιο προϊόν εξαντληθεί και πλέον είναι μη διαθέσιμο.

Αρχικά θα κατασκευάσουμε την ιστοσελίδα-ηλεκτρονικό κατάστημα η οποία θα πρέπει να παρουσιάζει όλα τα καταστήματα-εμπόρους και τα προϊόντα τους. Έπειτα θα κατασκευάσουμε τους μηχανισμούς πολυκριτηριακής αναζήτησης με τους οποίους θα μπορεί ο επισκέπτης να κάνει αναζήτηση σε όλα τα καταστήματα και να βρίσκει όλα τα προϊόντα που ταιριάζουν στα κριτήρια του και αργότερα θα εστιάσουμε στη δυναμική ενημέρωση του περιεχομένου τους. Συγκεκριμένα, θέλουμε η ιστοσελίδα να ενημερώνεται για κάθε νέο προϊόν που εισάγετε καθώς και για κάθε νέο κατάστημα-έμπορο.

Επίσης, είναι ζητούμενό μας κάνοντας χρήση του Google MAPs API, να πλοηγούμαστε στα καταστήματα και να βλέπουμε πληροφορίες για αυτά. Πρώτα θα κατασκευάσουμε χάρτη στον οποίο θα φαίνονται τα καταστήματα με πινέζες, μικρογραφίες και κάποιες πληροφορίες. Ενώ αργότερα θα υπάρχει η δυνατότητα να εμφανίζονται κάποιες πινέζες (καταστήματα) ή όχι ανάλογα με την αναζήτηση (π.χ. δείξε όλα τα καταστήματα, μόνο τα υποδήματα, μόνο τα αξεσουάρ).

Abstract

The purpose of this thesis is to create a website which can be monitored and dynamically updated content and more specifically a web shop.

This website will be updated dynamically and changing content, in cases such as when a trader adds a new product to his store or when a product is out of stock now or unavailable.

First we will construct the site e-shop which should show all shops traders and their products. Then we will construct a multi-criteria search mechanisms by which the visitor will be able to search in all the stores and find all the products that fit the criteria and would later focus on dynamic updating of their contents. Specifically, we want the site to be updated for each new product which is entered and for each new store-trader.

It is also requested, using the Google MAPs API, to navigate to the shops and to see information about them. First we will construct a map on which stores will appear in with pins, thumbnails and some information. While later will be able to display some pins (shops) or not depending on the search (eg show all shops, only footwear, only accessories).

Περιεχόμενα

Κεφάλαιο 1	9
Εισαγωγή	9
Κεφάλαιο 2	11
Τεχνολογίες που χρησιμοποιήθηκαν	11
2.1 Η Γλώσσα Προγραμματισμού του Internet HTML	11
2.1.1 Η HTML και το Internet	11
2.1.2 Τι Είναι η HTML	11
2.1.3 Η δομή μιας HTML Σελίδας	12
2.2 XML	13
2.2.1 Προέλευση και στόχοι	14
2.3 Η Τεχνολογία CSS	14
2.3.1 Τι Είναι τα CSS	14
2.3.2 Τα Στυλ Λύνουν ένα Κοινό Πρόβλημα	15
2.3.3 Τα Φύλλα Στυλ μας Γλυτώνουν από Πολλή δουλειά	15
2.3.4 Πολλαπλά Στυλ Καταλήγουν σ' ένα	15
2.4 JavaScript	16
2.4.1 Τι είναι JavaScript?	16
2.4.2 Τι μπορούμε να κάνουμε με την javascript?	16
2.4.3 Πώς συντάσσεται?	16
2.5 Η Γλώσσα Προγραμματισμού PHP	17
2.5.1 Τι Είναι η PHP	17
2.5.2 Τι Μπορεί να Κάνει η PHP	18
2.5.3 Μια Σύντομη Ιστορία της PHP	18
2.5.4 Οι Τύποι δεδομένων της PHP	19
2.6 Η Βάση δεδομένων MySQL	19
2.6.1 Τι Είναι οι Βάσεις δεδομένων (Databases)	19
2.6.2 Τι Είναι η SQL	20
2.7 Εγκατάσταση βοηθητικών εργαλείων	20
2.7.1 Εγκατάσταση PHP - MySQL	21
2.7.2 Εγκατάσταση XAMPP	21

Κεφάλαιο 3	26
Ανάπτυξη και Περιγραφή της Εφαρμογής	26
3.1 Δομή του site	26
3.2 Χρήση αντικειμενοστρεφούς κώδικα για την ανάπτυξη της εφαρμογής	27
3.2.1 Δομή μιας κλάσης	29
3.3. Ανάπτυξη της εφαρμογής και ανάλυση του κώδικα	29
3.3.1 Αρχική σελίδα – Home	30
Κεφάλαιο 4	34
Επισκέπτες της Εφαρμογής	34
4.1 Η menu.php	34
4.2 Η katasthmata.php	36
4.3 Η proion.php	39
4.4 Η ThanksLike.php	42
Κεφάλαιο 5	44
Τα καταστήματα-Έμποροι και Admin	44
5.1 Η login.html	44
5.2 Η pistopoihsh.php	45
5.3. Τι είναι τα sessions	46
5.3.1 Πώς να δημιουργήσετε ένα session	46
5.3.2 Παράδειγμα	47
5.3.3 Καταστροφή ενός session	47
5.4 Η logout.php	48
5.5 Η Eggraph.html	48
5.6 Η Eggraph.php	50
5.7 Η katasthatarxes.php	53
5.8 Η Diathesimothta.php	56
Κεφάλαιο 6	59
Google MAPs API	59
6.1 Η Gmap.php	59
6.1 Η Gmap.php	60
Βιβλιογραφία	63

Κεφάλαιο 1

Εισαγωγή

Η απαρχή της δημιουργίας του (διαδικτύου (Internet) τοποθετείται στα 1968. Τότε το υπουργείο Άμυνας των Η.Π.Α. ανέθεσε στην Αμερικανική Υπηρεσία Προηγμένων Ερευνητικών Προγραμμάτων ARPA (Advanced Research Projects Agency) τη δημιουργία ενός αποκεντρωμένου αλλά κλειστού δικτύου του. Δημιουργήθηκε λοιπόν το ARPANET με στόχο την εξασφάλιση της αμφίδρομης επικοινωνίας μεταξύ των απομακρυσμένων στρατιωτικών μονάδων, ακόμα και έπειτα από γενικευμένη πολεμική σύρραξη. Τα επόμενα χρόνια το διαδίκτυο αυτό μετεξελίχθηκε, ευτυχώς, σε ιδανικό μέσο επικοινωνίας μεταξύ στρατιωτικών και αργότερα πανεπιστημιακών που εργάζονταν για την κυβέρνηση. Σταδιακά άρχισαν να δημιουργούνται στις Η.Π.Α. και άλλα μικρά δίκτυα στο πρότυπο του ARPANET. Το 1995 μετά και από βελτίωση των προσωπικών υπολογιστών (PC-Personal Computer) οι τεχνολογικές προϋποθέσεις θα επιτρέψουν την πραγματοποίηση του διαδικτύου. Προηγουμένως, το 1982, έχει υιοθετηθεί για πρώτη φορά ένα σαφές πρωτόκολλο για τη μετάδοση των πληροφοριών, και το 1984 δημιουργήθηκε μια κοινή ονοματοθεσία σχετικά με τους τόπους και τις διευθύνσεις του διαδικτύου (INTERNATIONALNETWORK).

Το 1990 ο Tim Berners-Lee δημιούργησε τον Παγκόσμιο Ιστό - World Wide Web (www) στο CERN της Ελβετίας για τη δικτυακή διανομή και πρόσβαση σε αρχεία υπερκειμένου και μέχρι το 1993, όταν αναπτύχθηκε ο πρώτος σελιδομετρητής ο περίφημος Mosaic για τα X Windows με γραφικό user interface από τον Marc Andreessen στο πανεπιστήμιο του Illinois των ΗΠΑ, το διαδίκτυο, ήταν κτήμα μιας μικρής επιστημονικής κοινότητας που το χρησιμοποιούσε για τον διαμοιρασμό επιστημονικών αναφορών και άρθρων. Πολλοί λίγοι θα μπορούσαν να φανταστούν τις διαστάσεις που θα έπαιρνε τα επόμενα χρόνια το διαδίκτυο τόσο σε όγκο, έκταση , εφαρμογές όσο και στην επίδραση σε διάφορους τομείς στη καθημερινότητα του ανθρώπου. Στην έκρηξη του διαδικτύου ασφαλώς συνετέλεσαν η ευρεία διάθεσή του για εμπορικούς σκοπούς επιστημονικούς σκοπούς, οι ευρυζωνικές συνδέσεις με τη συνεχιζόμενη αύξηση στις ταχύτητές τους, όπως και η ανάπτυξη κάθε είδους

περιεχομένου που αναπτύσσεται και αναρτάται σε αυτό.

Ο παγκόσμιος ιστός είναι σίγουρα η πιο διαδεδομένη υπηρεσία και το πιο δυνατό σημείο του διαδικτύου και ενοποιεί πολλά πρωτόκολλα και υπηρεσίες. Το Web είναι ένα σύστημα παράδοσης πληροφοριών, το οποίο περιλαμβάνει διάφορους τύπους στοιχείων, όπως κείμενο, φωτογραφίες, πολυμέσα και συνδέσεις που παραπέμπουν σε άλλα κείμενα και γενικότερα πληροφορίες και επιτρέπει σε κάθε επισκέπτη του (διαδικτύου να ανακτά καθορισμένες πληροφορίες με σχετικά γρήγορο και αποτελεσματικό τρόπο.

Η συνεχόμενη αύξηση των δεδομένων και η συνεχόμενη απαίτηση των χρηστών του διαδικτύου για νέες πληροφορίες, νέες με σε real -time οδήγησαν στην ανάπτυξη νέων τεχνολογιών για τον εύχρηστο τρόπο

λειτουργίας του Web και την αξιοποίησή του έτσι από την επιστημονική κοινότητα για νέες καινοτομίες και έρευνα.

Λόγω του μεγάλου αριθμού των χρηστών του διαδικτύου κρίθηκε αναγκαίο να δημιουργηθούν κάποια ηλεκτρονικά καταστήματα. Για αυτό το λόγο δημιουργήθηκαν εξειδικευμένες σελίδες αγορών-πωλείσεων στις οποίες μπορεί ο επισκέπτης να δει και να αγοράσει προϊόντα που τον ενδιαφέρουν και τα καταστήματα να δείξουν και να πουλήσουν τα προϊόντα τους. Στην παρούσα εργασία θα προσπαθήσουμε να υλοποιήσουμε μια δικτυακή εφαρμογή, πιο συγκεκριμένα μια ιστοσελίδα της οποίας το περιεχόμενο θα πρέπει να ενημερώνεται δυναμικά δηλαδή όποτε αλλάζει κάποιο προϊόν σε κάποιο κατάστημα και στην οποία οι επισκέπτες θα μπορούν να κάνουν αναζήτηση σε όλα τα προϊόντα βλέποντας κάθε φορά αυτά που ταιριάζουν στα κριτήρια της αναζήτησης τους.

Κεφάλαιο 2

Τεχνολογίες που χρησιμοποιήθηκαν

Στο κεφάλαιο αυτό θα αναφερθούμε στις τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση του site. Θα γίνει αναφορά στις HTML, CSS, Javascript, PHP και MySQL. Οι τεχνολογίες αυτές χρησιμοποιούνται σε μεγάλο βαθμό για τη δημιουργία sites στο internet και για διάφορες διαδικτυακές εφαρμογές. Επίσης, αποτέλεσαν τη βάση για άλλες τεχνολογίες (XML, Ajax, JSON) ή και για νεότερα συστήματα διαχείρισης δυναμικού Περιεχομένου (Content Management System) - CMS όπως Drupal, Joomla που χρησιμοποιούνται για τους ίδιους σκοπούς.

Επίσης, θα κάνουμε λόγο και με ποιον τρόπο θα εγκαταστήσουμε κάποια βοηθητικά εργαλεία που είναι χρήσιμα, ώστε να αξιοποιήσουμε τις παραπάνω τεχνολογίες.

2.1 Η Γλώσσα Προγραμματισμού του Internet HTML

2.1.1 Η HTML και το Internet

Κάθε σελίδα που εμφανίζεται στο Internet είναι ένα αρχείο γραμμένο με τη γλώσσα **HTML** (*HyperText Markup Language, Γλώσσα Χαρακτηρισμού Υπερ-Κειμένου*), που περιλαμβάνει το κείμενο της σελίδας, τη δομή της και τους

συνδέσμους προς άλλα έγγραφα, εικόνες ή άλλα μέσα.

Ο φυλλομετρητής (Web browser) παίρνει τις πληροφορίες από τον Web server, τις μορφοποιεί και τις εμφανίζει κατάλληλα για το σύστημά μας. Διαφορετικά προγράμματα φυλλομετρητή μπορεί να μορφοποιούν και να εμφανίζουν το ίδιο αρχείο με διαφορετικό τρόπο, ανάλογα με τις δυνατότητες του συστήματος στο οποίο τρέχουν και τις επιλογές διαμόρφωσης του προγράμματος του φυλλομετρητή.

Μια **Web σελίδα** ή **ιστοσελίδα** (*Web page*) είναι ένα μεμονωμένο στοιχείο μιας παρουσίασης για το Web και περιέχεται σ' ένα αρχείο στον δίσκο, το οποίο ανακτάται από έναν Web server και μορφοποιείται μέσω ενός φυλλομετρητή.

Η **αρχική σελίδα** (*home page*) είναι η πρώτη ή κορυφαία σελίδα μιας παρουσίασης για το Web, είναι δηλαδή το σημείο εισόδου ή εκκίνησης για τις υπόλοιπες σελίδες της παρουσίασης και η πρώτη σελίδα που θα συναντήσουν οι αναγνώστες της παρουσίασής μας. Η αρχική σελίδα περιέχει συνήθως μια σύνοψη του περιεχομένου της παρουσίασης με τη μορφή ενός πίνακα περιεχομένων ή μιας ομάδας εικονιδίων.

2.1.2 Τι Είναι η HTML

Η **HTML** είναι το ακρωνύμιο των λέξεων *HyperText Markup Language*, δηλ. *Γλώσσα Χαρακτηρισμού Υπερ-Κειμένου* και βασίζεται στη γλώσσα **SGML**, *Standard Generalized Markup Language*, που είναι ένα πολύ μεγαλύτερο σύστημα επεξεργασίας εγγράφων.

Η HTML ορίζει ένα σύνολο κοινών στυλ για τις Web σελίδες, όπως τίτλοι

(titles), επικεφαλίδες (headings), παράγραφοι (paragraphs), λίστες (lists) και πίνακες (tables). Ορίζει επίσης στυλ χαρακτήρων, όπως η έντονη γραφή (boldface) και οι ενότητες κώδικα.

Κάθε στοιχείο έχει ένα όνομα και περιέχεται μέσα στα σύμβολα <>, που αποκαλούνται **tags** (ετικέτες). Όταν γράφουμε μια Web σελίδα με την HTML, στην ουσία δίνουμε τίτλους στα διάφορα στοιχεία της σελίδας μ' αυτά τα tags.

Οι φυλλομετρητές, μαζί με τη δυνατότητά τους να ανακτούν σελίδες από το Web, λειτουργούν επίσης και σαν μορφοποιητές για την HTML. Όταν διαβάζουμε μια σελίδα γραμμένη με την HTML σ' έναν φυλλομετρητή, ο φυλλομετρητής διαβάζει (διερμηνεύει) τα tags της HTML και μορφοποιεί το κείμενο και τις εικόνες στην οθόνη.

Διαφορετικοί φυλλομετρητές, οι οποίοι τρέχουν σε διαφορετικούς υπολογιστές, μπορεί να αντιστοιχίζουν διαφορετικά στυλ σε κάθε στοιχείο μιας σελίδας. Αυτό σημαίνει ότι οι σελίδες που δημιουργούμε με την HTML μπορεί να δείχνουν εντελώς διαφορετικές από σύστημα σε σύστημα και από φυλλομετρητή σε φυλλομετρητή. δηλαδή, οι πραγματικές πληροφορίες και οι σύνδεσμοι που περιέχουν οι σελίδες μας θα είναι πάντα εκεί, αλλά η εμφάνιση των σελίδων στην οθόνη θα είναι διαφορετική.

2.1.3 Η δομή μιας HTML Σελίδας

Οι σελίδες που γράφουμε με την HTML είναι απλά αρχεία κειμένου σε μορφή ASCII, που σημαίνει ότι δεν περιέχουν πληροφορίες για κάποιο λειτουργικό σύστημα ή πρόγραμμα, αλλά μπορούν να διαβαστούν από οποιονδήποτε συντάκτη υποστηρίζει απλό κείμενο.

Τα αρχεία της HTML περιέχουν τα ακόλουθα :

- Το κείμενο της σελίδας.
- Τα tags της HTML, τα οποία υποδεικνύουν τα στοιχεία, τη δομή και τη μορφοποίηση των σελίδων, καθώς επίσης και τους συνδέσμους υπερκειμένου προς άλλες σελίδες ή προς αρχεία άλλων μορφών (πολυμέσα).

Τα περισσότερα tags της HTML έχουν την εξής μορφή :

<ΌνομαTag> επηρεαζόμενο κείμενο </ΌνομαTag>

Τα tags της HTML έχουν γενικά ένα tag αρχής και ένα tag τέλους ή ένα tag ανοίγματος και ένα tag κλεισίματος, τα οποία περικλείουν το κείμενο που επηρεάζουν. Το tag αρχής ενεργοποιεί μια λειτουργία ή ένα χαρακτηριστικό, όπως είναι για παράδειγμα η έντονη γραφή, ενώ το tag τέλους την απενεργοποιεί. Τα tags τέλους έχουν το ίδιο όνομα με τα tags αρχής, αλλά με πρόθεμα τον χαρακτήρα /. Δεν αποτελούν ζευγάρι όλα τα tags της HTML, καθώς ορισμένα είναι "μονομελή", ενώ άλλα περιέχουν επιπλέον πληροφορίες και κείμενο μέσα στα σύμβολα <>. Ακόμη, όλα τα tags της HTML δεν κάνουν διάκριση μεταξύ κεφαλαίων και πεζών γραμμάτων, δηλ. μπορούμε να τα γράφουμε είτε με κεφαλαίους είτε με πεζούς χαρακτήρες είτε με οποιονδήποτε

συνδυασμό τους. Συνήθως τα γράφουμε κεφαλαία για να μπορούμε να τα ξεχωρίζουμε ευκολότερα από το κείμενο της σελίδας.

Τα αρχεία της HTML πρέπει να τα γράφουμε σε συντάκτες κειμένων, δηλ. προγράμματα που μπορούν να αποθηκεύσουν αρχεία κειμένου σε μορφή ASCII. Τέτοια προγράμματα είναι τα Notepad, Wordpad και Write των Windows, το Edit του DOS και τα vi, emacs και pico του UNIX.

Το όνομα του αρχείου θα πρέπει να έχει επέκταση **.html** ή **.htm** σε συστήματα DOS ή Windows που επιτρέπουν επεκτάσεις μόνο τριών χαρακτήρων.

Η HTML χρησιμοποιεί τρία tags για την περιγραφή της συνολικής δομής μιας σελίδας, τα οποία παρέχουν ορισμένες απλές πληροφορίες κεφαλίδας. Αυτά τα tags προσδιορίζουν τη σελίδα μας στους φυλλομετρητές και παρέχουν επίσης απλές πληροφορίες για τη σελίδα, όπως τον τίτλο ή τον συγγραφέα της, πριν από τη φόρτωση ολόκληρης της σελίδας.

2.2 XML

Σε ένα κόσμο όπου οι πληροφορίες παρέχονται μέσω του παγκόσμιου διαδικτύου, τα έγγραφα πρέπει να είναι εύκολα προσβάσιμα, μεταφέρσιμα και ευέλικτα. Πρέπει επίσης να είναι ανεξάρτητα οποιουδήποτε συστήματος και περιεχομένου. Οι γενικευμένες γλώσσες έχουν τέτοια χαρακτηριστικά, παρέχοντας στα έγγραφα αυτά μια δυνατότητα η οποία δεν υπάρχει σε άλλες γλώσσες περιγραφής εγγράφων. Η HTML είναι προβληματική και περιοριστική γλώσσα. Η XML (Extensible Markup Language) έλυσε πολλά από τα προβλήματα που αντιμετώπισαν οι σχεδιαστές του web και είναι υπεύθυνη για την XHTML, μια ανασχεδιασμένη HTML. Θα χρησιμοποιείται για πολλά χρόνια επειδή προσφέρει αποτελεσματικές και δυναμικές πολυμεσικές λύσεις.

Η XML σχεδιάστηκε να ικανοποιήσει πολλές ανάγκες δίνοντας στα έγγραφα ένα μεγαλύτερο επίπεδο προσαρμοστικότητας στο στυλ και τη δομή από αυτό που υπήρχε παλαιότερα στην HTML. Η XML προσφέρει στους σχεδιαστές της HTML τη δυνατότητα να προσθέτουν περισσότερα στοιχεία στη γλώσσα. Δεν αναφέρεται μονάχα στους σχεδιαστές του web αλλά σε οποιονδήποτε ασχολείται με εκδόσεις.

Στην πραγματικότητα, η XML είναι markup γλώσσα για έγγραφα που περιέχουν δομημένες πληροφορίες. Markup γλώσσα είναι ένας μηχανισμός που καθορίζει δομές σε ένα έγγραφο. Οι δομημένες πληροφορίες περιλαμβάνουν περιεχόμενο και κάποιες διευκρινίσεις για το ρόλο που παίζει το περιεχόμενο σχεδόν όλα τα έγγραφα έχουν την ίδια δομή.

Η XML είναι κάτι περισσότερο από markup language είναι metalanguage, δηλαδή μια γλώσσα που χρησιμοποιείται για να καθορίσει νέες markup γλώσσες. Η XML συμπληρώνει και δεν αντικαθιστά την HTML. Ενώ η HTML χρησιμοποιείται στη διατύπωση και την εμφάνιση των δεδομένων η XML αναπαριστά τη συναφή έννοια των δεδομένων. Στην HTML τα tags είναι προκαθορισμένα ενώ η XML παρέχει τη δυνατότητα να καθορίζουν οι χρήστες τα tags και τις δομημένες μεταξύ τους σχέσεις.

Τα XML έγγραφα δεν είναι πολύπλοκα αλλά απλά και πολύ αποτελεσματικά. Το διδακτικό υλικό της well-formed XML αναλύει τη δημιουργία των XML εγγράφων, η οποία είναι κατά κάποιο τρόπο ίδια με την HTML καθώς επιτρέπει τη μη δομημένη δημιουργία εγγράφου. Η valid XML

είναι πιο σύνθετη. Απαιτεί την ύπαρξη ενός Document Type Definition ή ενός XML Schema πριν να γραφεί το έγγραφο, που παρέχει μια γενική δομή με βάση την οποία τη δημιουργούμε.

Η γλώσσα προγραμματισμού XML περιγράφει μια κατηγορία πληροφοριών (data objects) που καλούνται XML έγγραφα (documents) καθώς επίσης περιγράφει τμηματικά τη συμπεριφορά των προγραμμάτων που τα επεξεργάζονται.

Τα XML έγγραφα αποτελούνται από μονάδες αποθήκευσης που καλούνται entities (οντότητες), οι οποίες περιέχουν πληροφορίες αναλυμένες ή μη. Οι αναλυμένες πληροφορίες αποτελούνται από χαρακτήρες (characters) οι οποίοι συνθέτουν character data και άλλοι οι οποίοι συνθέτουν markup. Η μορφή markup κωδικοποιεί την περιγραφή της τελικής αποθήκευσης του εγγράφου καθώς και τη λογική δομή.

Ένα λογισμικό μοντέλο που καλείται επεξεργαστής XML χρησιμοποιείται να διαβάσει XML έγγραφα και παρέχει πρόσβαση στο περιεχόμενο και τη δομή τους. Υποτίθεται ότι ο επεξεργαστής XML λειτουργεί εκ μέρους ενός άλλου μοντέλου που καλείται application (εφαρμογή). Αυτή η προδιαγραφή περιγράφει την απαιτούμενη συμπεριφορά του επεξεργαστή και συγκεκριμένα πως θα πρέπει να διαβάσει τα XML δεδομένα και ποιες πληροφορίες πρέπει να παρέχει στην εφαρμογή.

2.2.1 Προέλευση και στόχοι

Η γλώσσα XML αναπτύχθηκε από μια Ομάδα Εργασίας της XML κάτω από την καλή κηδεμονία του διεθνούς οργανισμού World Wide Web Consortium (W3C) το 1996. Εδραιώθηκε από τον John Bosak της Sun Microsystems με την ενεργή συμμετοχή μιας XML Ομάδας Ειδικού Ενδιαφέροντος (που οργανώθηκε από τον οργανισμό W3C).

Οι προσχεδιασμένοι στόχοι της XML είναι:

1. Η XML πρέπει να είναι εύχρηστη στο Internet.
2. Η XML πρέπει να υποστηρίζει μεγάλη ποικιλία από εφαρμογές
3. Η XML πρέπει να είναι συμβατή με την SGML.
4. Θα είναι εύκολο να γράφονται προγράμματα που επεξεργάζονται XML έγγραφα.
5. Ο αριθμός των προαιρετικών χαρακτηριστικών στην XML θα είναι όσο το δυνατόν πιο μικρός, ιδανικό επίπεδο το μηδέν.
6. Τα XML έγγραφα θα πρέπει να είναι ευανάγνωστα
7. Ο σχεδιασμός XML θα πρέπει να προετοιμάζεται γρήγορα.
8. Ο σχεδιασμός XML θα πρέπει να είναι τυπικός και περιεκτικός.
9. Τα XML έγγραφα θα πρέπει να δημιουργούνται εύκολα.
10. Η περιεκτικότητα στον XML συμβολισμό είναι μικρής σημασίας.

2.3 Η Τεχνολογία CSS

2.3.1 Τι Είναι τα CSS

Ο όρος CSS σημαίνει *Cascading Style Sheets*, δηλ. *διαδοχικά Φύλλα Στυλ*. Τα στυλ ορίζουν τον τρόπο εμφάνισης των στοιχείων της HTML και αποθηκεύονται σε Φύλλα Στυλ (Style Sheets). Τα Στυλ προστέθηκαν στην

HTML 4.0 για να λύσουν ορισμένα προβλήματα και τα Εξωτερικά Φύλλα Στυλ (External Style Sheets) μπορούν να μας γλυτώσουν από πολλή δουλειά., αφού πολλαπλοί ορισμοί στυλ μπορούν να συμπεριληφθούν σ' έναν.

2.3.2 Τα Στυλ Λύνουν ένα Κοινό Πρόβλημα

Τα tags της HTML σχεδιάστηκαν αρχικά για να ορίσουν το περιεχόμενο ενός εγγράφου. Η δουλειά τους ήταν να έλεγαν "Αυτή είναι μια επικεφαλίδα", "Αυτή είναι μια παράγραφος", "Αυτός είναι ένας πίνακας", χρησιμοποιώντας tags όπως <h1>, <p>, <table> κοκ. Η διάταξη (layout) του εγγράφου ήταν υπόθεση του φυλλομετρητή, χωρίς τη χρήση tags μορφοποίησης (formatting tags).

Καθώς οι δύο κύριοι φυλλομετρητές, ο Netscape και ο Internet Explorer,

συνέχισαν να προσθέτουν νέα HTML tags και χαρακτηριστικά (attributes), όπως το tag και το attribute color, στις αρχικές προδιαγραφές της HTML, γινόταν ολοένα και δυσκολότερη η δημιουργία Web sites όπου το περιεχόμενο των εγγράφων HTML να μπορεί να ξεχωρίζει καθαρά από τη διάταξη παρουσίασης του εγγράφου.

Για να μπορέσει να λύσει αυτό το πρόβλημα, το *World Wide Web Consortium (W3C)* – το μη κερδοσκοπικό consortium ορισμού στάνταρτς που είναι υπεύθυνο για την στάνταρταποίηση της HTML – δημιούργησε τα ΣΤΥΛ (STYLES) σαν προσθήκη στην HTML 4.0.

Και ο Netscape 4.0 και ο Internet Explorer 4.0 υποστηρίζουν τα διαδοχικά Φύλλα Στυλ (Cascading Style Sheets).

2.3.3 Τα Φύλλα Στυλ μας Γλυτώνουν από Πολλή Δουλειά

Τα στυλ στην HTML 4.0 ορίζουν το πώς εμφανίζονται τα HTML στοιχεία,

όπως ακριβώς το tag font και το attribute color στην HTML 3.2. Τα στυλ αποθηκεύονται συνήθως σε αρχεία που είναι εξωτερικά (external) στα HTML έγγραφά μας.

Τα εξωτερικά φύλλα στυλ (external style sheets) μάς δίνουν τη δυνατότητα να αλλάξουμε την εμφάνιση και τη διάταξη όλων των σελίδων στο δικό μας Web site, με απλή επεξεργασία ενός μόνου CSS εγγράφου. Αν έχουμε ποτέ προσπαθήσει να αλλάξουμε τη γραμματοσειρά (font) ή το χρώμα (color) όλων των επικεφαλίδων (headings) σ' όλες τις ιστοσελίδες μας, θα καταλάβουμε γιατί τα CSS μπορούν να μας γλυτώσουν από πολλή δουλειά.

Τα CSS αποτελούν μια μεγάλη επιτυχία στον σχεδιασμό του Web (Web design) επειδή δίνουν τη δυνατότητα στους developers να ελέγξουν το στυλ και τη διάταξη πολλών ιστοσελίδων μονομιάς. Σαν Web developer μπορούμε να ορίσουμε ένα στυλ για κάθε HTML στοιχείο και να το εφαρμόσουμε σ' όσες ιστοσελίδες θέλουμε. Για να κάνουμε μια καθολική (global) αλλαγή, απλά αλλάζουμε το στυλ μία φορά και όλα τα στοιχεία του Web ενημερώνονται αυτόματα.

2.3.4 Πολλαπλά Στυλ Καταλήγουν σ' ένα

Τα Φύλλα Στυλ επιτρέπουν τον ορισμό των πληροφοριών στυλ με πολλούς τρόπους. Τα στυλ μπορούν να ορισθούν μέσα σ' ένα μόνο HTML στοιχείο, μέσα στο στοιχείο <head> μιας HTML σελίδας ή σ' ένα εξωτερικό αρχείο CSS. Πολλά εξωτερικά Φύλλα Στυλ μπορούν να χρησιμοποιούνται μέσα απ' ένα μόνο HTML έγγραφο.

Ποιο στυλ θα χρησιμοποιηθεί όταν υπάρχουν περισσότερα από ένα καθορισμένα στυλ για ένα HTML στοιχείο ;

Σε γενικές γραμμές μπορούμε να πούμε ότι όλα τα στυλ θα καταλήξουν (cascade) σ' ένα νέο εικονικό (virtual) Φύλλο Στυλ σύμφωνα με τους παρακάτω κανόνες, όπου ο μεγαλύτερος αριθμός έχει και την υψηλότερη προτεραιότητα :

1. Προεπιλογή του φυλλομετρητή.
2. Εξωτερικό Φύλλο Στυλ (External Style Sheet).
3. Εσωτερικό Φύλλο Στυλ (Internal Style Sheet), μέσα στο τμήμα header . του εγγράφου.
4. Inline Style, μέσα στο HTML στοιχείο.

2.4 JavaScript

2.4.1 Τι είναι JavaScript?

Η javascript είναι μια scripting γλώσσα η οποία έχει σχεδιαστεί και χρησιμοποιείται για να εισάγουμε την έννοια της διαδραστικότητας στις html σελίδες. Είναι μια ερμηνευτική γλώσσα,δηλαδή το script εκτελείται χωρίς να έχει περάσει από τη διαδικασία της σύνταξης. Η javascript και η java είναι δύο εντελώς διαφορετικές γλώσσες.

2.4.2 Τι μπορούμε να κάνουμε με την javascript?

- Με την javascript μπορούμε να εκτελέσουμε κάποια πράγματα όταν συμβαίνει ένα γεγονός,για παράδειγμα όταν ο χρήστης κλικάρει σε ένα html στοιχείο,να εκτελείται κάποιο script και λαμβάνουμε τα αντίστοιχα αποτελέσματα.
- Η javascript μπορεί να διαβάσει και να αλλάξει τα περιεχόμενα ενός html στοιχείου.
- Μπορούμε να την χρησιμοποιήσουμε για να επικυρώσουμε δεδομένα μιας φόρμας(validate) πρώτου να υποβληθούν στον server (γλυτώνουμε και τον server απο επιπλέον έλεγχο).
- Με την βοήθεια της javascript μπορούμε να εντοπίσουμε τον browser του επισκέπτη και ανάλογα με τον browser να φορτώσουμε την αντίστοιχη σελίδα που είναι φτιαγμένη για αυτόν τον browser.
- Μπορούμε να δημιουργήσουμε cookies (να αποθηκεύουμε και να λαμβάνουμε πληροφορίες στον υπολογιστή του επισκέπτη).

2.4.3 Πώς συντάσσεται?

Για να εισάγουμε javascript σε ένα html αρχείο χρησιμοποιούμε την ετικέτα <script> και μέσα σε αυτή την ετικέτα χρησιμοποιούμε το όρισμα "type" για να ορίσουμε την scripting γλώσσα που θα χρησιμοποιήσουμε.

Οι ετικέτες <script>...</script> μας δηλώνουν που αρχίζει και που τελειώνει η javascript.

Ας δούμε ένα παράδειγμα για να καταλάβουμε καλύτερα :

```
<html>
< body>
< p>Μια παράγραφος</p>
<script type="text/javascript">
... κώδικας javascript ...
< /script>
</body>
< /html>
```

Ό,τι είναι μέσα στις ετικέτες <script> εκτελείται από τον browser. Αν δεν βάλουμε τον κώδικα javascript μέσα σε ετικέτες script ο browser τον αντιλαμβάνεται σαν απλό κείμενο και δεν τον εκτελεί.

2.5 Η Γλώσσα Προγραμματισμού PHP

2.5.1 Τι Είναι η PHP

Η *PHP*, όπου τα αρχικά σημαίνουν *Hypertext PreProcessor*, είναι μια γλώσσα συγγραφής σεναρίων (scripting language) που ενσωματώνεται μέσα στον κώδικα της HTML και εκτελείται στην πλευρά του server (server-side scripting).

Ανταγωνιστικές της τεχνολογίας PHP είναι οι εξής γλώσσες προγραμματισμού: *ASP (Active Server Pages)* της εταιρείας Microsoft, *CFML (ColdFusion Markup Language)* της εταιρείας Allaire και *JSP (JavaServer Pages)* της εταιρείας Sun.

Το μεγαλύτερο μέρος της σύνταξής της, η PHP το έχει δανειστεί από την C, την Java και την Perl και διαθέτει και μερικά δικά της μοναδικά χαρακτηριστικά. Ο σκοπός της γλώσσας είναι να δώσει τη δυνατότητα στους web developers να δημιουργούν δυναμικά παραγόμενες ιστοσελίδες.

Ακολουθεί ένα εισαγωγικό παράδειγμα :

```
<html>
<head>
<title> Παράδειγμα </title>
</head>
<body>
<?php echo "Γεια σας, είμαι ένα script της PHP!"; ?>
</body>
</html>
```

Προσέξτε πόσο διαφέρει από ένα CGI script που γράφεται σ' άλλες γλώσσες, όπως η Perl ή η C, όπου αντί να γράψουμε ένα πρόγραμμα με πολλές εντολές για να δημιουργήσουμε κώδικα HTML, γράφουμε ένα HTML

script με κάποιον ενσωματωμένο κώδικα για να κάνει κάτι, όπως στην συγκεκριμένη περίπτωση να εμφανίσει κάποιο κείμενο (μήνυμα). Ο κώδικας της PHP περικλείεται με ειδικά tags αρχής και τέλους για να μπορούμε να εισερχόμαστε και να εξερχόμαστε από το PHP mode.

Αυτό που ξεχωρίζει την PHP από μια γλώσσα όπως η JavaScript, η οποία εκτελείται στην πλευρά του χρήστη (client-side), είναι ότι ο κώδικας της εκτελείται στον server. Αν είχαμε σ' έναν server ένα script παρόμοιο με το παραπάνω, ο χρήστης (client) θα λάμβανε το αποτέλεσμα της εκτέλεσης αυτού του script, χωρίς να είναι σε θέση να γνωρίζει ποιος μπορεί να είναι ο αρχικός κώδικας.

Μπορούμε ακόμη να ρυθμίσουμε (configure) τον web server ώστε να επεξεργάζεται όλα τα HTML αρχεία με την PHP και τότε δεν θα υπάρχει πράγματι κανένας τρόπος να μάθουν οι χρήστες τον κώδικά μας.

2.5.2 Τι Μπορεί να Κάνει η PHP

Στο πιο βασικό επίπεδο, η PHP μπορεί να κάνει ό,τι και τα άλλα προγράμματα της τεχνολογίας CGI, όπως επεξεργασία των δεδομένων μιας φόρμας, δημιουργία δυναμικού περιεχομένου ιστοσελίδων ή αποστολή και λήψη cookies.

Ίσως το δυνατότερο και πιο σημαντικό χαρακτηριστικό της PHP είναι η υποστήριξη που παρέχει σε μια ευρεία γκάμα από βάσεις δεδομένων. Έτσι, το να δημιουργήσουμε μια ιστοσελίδα που να παρέχει υποστήριξη σε βάσεις δεδομένων είναι απίστευτα απλό. Υποστηρίζει τις εξής βάσεις δεδομένων :

Adabas D	dBase	Empress	FilePro	Informix	InterBase	mSQL
MySQL	Oracle	PostgreSQL	Solid	Sybase	Velocis	Unix dbm

Η PHP παρέχει επίσης υποστήριξη για συνομιλία μ' άλλες υπηρεσίες, χρησιμοποιώντας πρωτόκολλα όπως τα IMAP, SNMP, NNTP, POP3 ή και το HTTP.

2.5.3 Μια Σύντομη Ιστορία της PHP

Η ιδέα για την δημιουργία της PHP ελήφθη το φθινόπωρο του 1994 από τον *Rasmus Lerdorf*. Οι πρώτες ανεπίσημες εκδόσεις (versions) της PHP χρησιμοποιήθηκαν στην αρχική του σελίδα (home page) για να μπορεί να παρακολουθεί αυτούς που έμπαιναν στην σελίδα. Η πρώτη έκδοση που δόθηκε για χρήση στο κοινό ήταν διαθέσιμη στις αρχές του 1995 με το όνομα *Personal Home Page Tools*.

Αποτελείτο από μια πολύ απλοϊκή μηχανή ανάλυσης (parser engine) η οποία καταλάβαινε λίγες μόνο ειδικές μακροεντολές (macros) και έναν αριθμό από utilities που βρίσκονταν σε κοινή χρήση στις home pages εκείνη την εποχή. Ένα guestbook, ένας μετρητής (counter) και κάποιο άλλο υλικό. Ο αναλυτής (parser) ξαναγράφηκε στα μέσα του 1995 και ονομάστηκε *PHP/FI Version 2*.

Το όνομα FI προέρχεται από ένα άλλο πακέτο που είχε γράψει ο Rasmus και το οποίο διερμήνευε (interpreted) τα δεδομένα από φόρμες της HTML. Συνδύασε τα εργαλεία scripts της Personal Home Page με τον *Form Interpreter* και πρόσθεσε υποστήριξη για mSQL. Έτσι γεννήθηκε η *PHP/FI*, η οποία αναπτύχθηκε αλματωδώς και διάφοροι χρήστες άρχισαν να συνεισφέρουν κώδικα σ' αυτήν.

Υπολογίζεται ότι μέχρι τα τέλη του 1996, η PHP/FI χρησιμοποιείτο σε τουλάχιστον 15.000 web sites σ' όλον τον κόσμο και στα μέσα του 1997 αυτός ο αριθμός είχε ξεπεράσει τις 50.000. Στα μέσα του 1997 είχαμε επίσης μια αλλαγή στην ανάπτυξη της PHP.

Σταμάτησε να αποτελεί το αγαπημένο αντικείμενο του Rasmus και έγινε ο στόχος μιας πιο καλά οργανωμένης ομαδικής εργασίας. Ο αναλυτής (parser) ξαναγράφηκε από την αρχή από τους Zeev Suraski και Andi Gutmans και αυτός ο νέος parser αποτέλεσε τη βάση για την PHP Version 3. Ένα μεγάλο μέρος του utility code μεταφέρθηκε από την PHP/FI στην PHP3 και ένα μεγάλο μέρος του ξαναγράφηκε από την αρχή.

Σήμερα, η PHP/FI και η PHP3 έρχονται μ' έναν αριθμό εμπορικών προϊόντων όπως ο web server C2 StrongHold και το RedHat Linux. Σύμφωνα με μια συντηρητική εκτίμηση, η PHP χρησιμοποιείται από περισσότερα από 150.000 sites σ' όλον τον κόσμο.

2.5.4 Οι Τύποι Δεδομένων της PHP

Η PHP υποστηρίζει τους εξής τύπους δεδομένων :

- array
- floating-point numbers
- integer
- object
- string

Ο τύπος δεδομένων μιας μεταβλητής δεν ορίζεται συνήθως από τον προγραμματιστή αλλά αποφασίζεται την ώρα εκτέλεσης (runtime) από την PHP ανάλογα με το περιβάλλον (context) στο οποίο χρησιμοποιείται η μεταβλητή. Αν θέλουμε να κάνουμε μια μεταβλητή να μετατραπεί σ' έναν συγκεκριμένο τύπο, μπορούμε είτε να *μετατρέψουμε (cast)* τη μεταβλητή ή να χρησιμοποιήσουμε τη συνάρτηση *settype()* σ' αυτή.

Πρέπει να έχουμε υπόψη μας ότι μια μεταβλητή μπορεί να συμπεριφερθεί διαφορετικά σε συγκεκριμένες καταστάσεις, ανάλογα με το τι τύπο δεδομένων έχει εκείνη την στιγμή.

2.6 Η Βάση Δεδομένων MySQL

2.6.1 Τι Είναι οι Βάσεις Δεδομένων (Databases)

Μια **βάση δεδομένων (database)** αποτελείται από έναν ή περισσότερους **πίνακες (tables)**, ο καθένας από τους οποίους περιέχει μια λίστα από κάποια πράγματα. Για μια βάση δεδομένων πελατών (clients), είναι φυσικό να ξεκινήσουμε μ' έναν πίνακα με όνομα *clients* που θα περιέχει μια λίστα από στοιχεία πελατών.

Ο κάθε πίνακας σε μια βάση δεδομένων περιέχει μια ή περισσότερες **στήλες** (*columns*) ή **πεδία** (*fields*), όπου η κάθε στήλη περιέχει μια συγκεκριμένη πληροφορία για τον κάθε πελάτη που υπάρχει στην βάση δεδομένων (*database*).

Ο πίνακας *clients* μπορεί να περιέχει στήλες για τον κωδικό ενός πελάτη (*ID*), για το όνομά του (*Name*) καθώς και για την ημερομηνία γέννησής του (*Date*). Ο κάθε πελάτης που αποθηκεύουμε σ' αυτόν τον πίνακα λέμε ότι αποτελεί μια **γραμμή** (*row*) ή μια **εγγραφή** (*record*) του πίνακα. Για παράδειγμα, ας δούμε τον παρακάτω πίνακα :

ID	Name	Date
1	Αντωνιάδης	1970-04-01
2	Παπαδόπουλος	1968-02-22

Εκτός από τις στήλες για το όνομα του πελάτη (*Name*) και την ημερομηνία γέννησής του (*Date*), υπάρχει και μια στήλη με όνομα *ID*, ο σκοπός της οποίας είναι να εκχωρήσει έναν μοναδικό αριθμό στον κάθε πελάτη έτσι ώστε να έχουμε έναν εύκολο τρόπο αναφοράς σ' αυτόν και να μπορούμε να τον ξεχωρίσουμε από τους άλλους πελάτες.

Σαν επισκόπηση, το παραπάνω είναι ένας πίνακας τριών στηλών που περιέχει δύο γραμμές ή καταχωρήσεις. Η κάθε γραμμή του πίνακα περιέχει έναν κωδικό (*ID*) αναγνώρισης του πελάτη, το όνομά του (*text*) καθώς και την ημερομηνία γέννησής του (*date*). Με βάση αυτήν την βασική ορολογία, είμαστε έτοιμοι να αρχίσουμε να χρησιμοποιούμε την MySQL.

2.6.2 Τι Είναι η SQL

Το σύνολο των εντολών που θα χρησιμοποιούμε από δω και πέρα για να λέμε στην MySQL τι να κάνει, αποτελεί μέρος ενός *standard* που αποκαλείται *δομημένη Γλώσσα Ερωτημάτων* (*Structured Query Language*) ή **SQL**. Οι εντολές της SQL αποκαλούνται επίσης και *ερωτήματα* (*queries*).

Η SQL αποτελεί την *standard* γλώσσα για αλληλεπίδραση με τις περισσότερες βάσεις δεδομένων, έτσι ακόμα κι αν αλλάξουμε στο μέλλον από την MySQL σε μια βάση δεδομένων όπως την *Microsoft SQL Server*, θα διαπιστώσουμε ότι οι περισσότερες από τις εντολές είναι ολόιδιες.

Δεν πρέπει να συγχέουμε την SQL με την MySQL. Η MySQL είναι το λογισμικό του *διακομιστή βάσεων δεδομένων* (*database server software*) που χρησιμοποιούμε, ενώ η SQL είναι η γλώσσα που χρησιμοποιούμε για να αλληλεπιδράσουμε με την βάση δεδομένων.

2.7 Εγκατάσταση βοηθητικών εργαλείων

Για τον σχεδιασμό και την ομαλή λειτουργία του site στον υπολογιστή πρέπει να εγκαταστήσουμε τις βιβλιοθήκες για PHP και MySQL και ένα τοπικό *web server*. Για τη συγγραφή κώδικα HTML έχουμε τη δυνατότητα να χρησιμοποιήσουμε από ένα απλό *text editor* όπως το *notepad* των *windows* ως πιο εξειδικευμένες εφαρμογές που μπορεί να βρει κανείς αναζητώντας στο

διαδίκτυο δωρεάν ή επί πληρωμή (Dreamweaver) τόσο σε περιβάλλον windows όσο σε περιβάλλον Linux ή και σε Mac. Η εγκατάσταση βιβλιοθηκών για PHP κτλ είναι πιο απαιτητική διαδικασία και συνήθως καταφεύγουμε σε πιο "γρήγορες" λύσεις που θα μας διευκολύνουν.

2.7.1 Εγκατάσταση PHP – MySQL

Για να μπορέσουμε να εργαστούμε και να σχεδιάσουμε δυναμικές σελίδες με τη χρήση της PHP και της MySQL απαιτείται το ανάλογο λογισμικό και ένας Web server . Θα χρησιμοποιήσουμε σαν server τον Apache Web Server και για τις βάσεις δεδομένων μας θα χρησιμοποιήσουμε τη MySQL. Τις 2 αυτές υπηρεσίες θα τις εγκαταστήσουμε μέσω του XAMPP όπου περιέχει δωρεάν όλες τις απαραίτητες βιβλιοθήκες για την ιστοσελίδα μας.

Το XAMPP είναι ένα πρόγραμμα ελεύθερου λογισμικού με μηδενικό κόστος. Αποτελείται κυρίως από τον Apache HTTP Server, MySQL βάση δεδομένων και μεταγλωττιστές σεναρίων γραμμένους στις γλώσσες προγραμματισμού PHP και Perl.

Το όνομα XAMPP είναι ακρωνύμιο και σημαίνει:

- X (σημαίνει πως είναι το λειτουργικό σύστημα X για Windows, L για Linux M για Mac)
- Apache HTTP Server
- MySQL
- PHP
- Perl

Το πρόγραμμα έχει δημιουργηθεί κάτω από την άδεια χρήσης GNU (General Public License) και συμπεριφέρεται σαν ένας web server χωρίς κόστος ικανός να φιλοξενεί πολλές δυναμικές ιστοσελίδες. Το XAMPP είναι διαθέσιμο στα Microsoft Windows, Linux, Solaris, και Mac OS X, και είναι κυρίως χρήσιμο για ανάπτυξη web projects.

2.7.2 Εγκατάσταση XAMPP

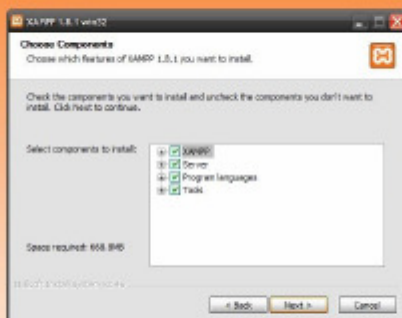
Το xampp μπορούμε να το βρούμε δωρεάν και να το «κατεβάσουμε» δωρεάν από την ιστοσελίδα <http://www.apachefriends.org/en/xampp.html> και να επιλέξουμε σε ποια πλατφόρμα θέλουμε να το εγκαταστήσουμε π.χ σε Linux, Windows, Mac OS και Solaris. Έχουμε κατεβάσει το xampp-win32-1.8.1-VC9-installer.exe και κάνουμε διπλό κλικ για να το εγκαταστήσουμε. Και εμφανίζεται η παρακάτω εικόνα:

XAMPP 1.8.1 win32



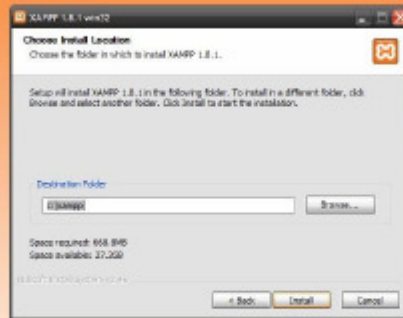
Επιλέγουμε Next

XAMPP 1.8.1 win32



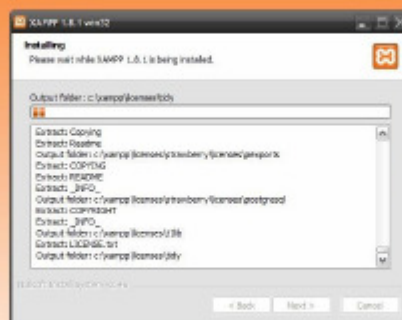
Next

XAMPP 1.8.1 win32

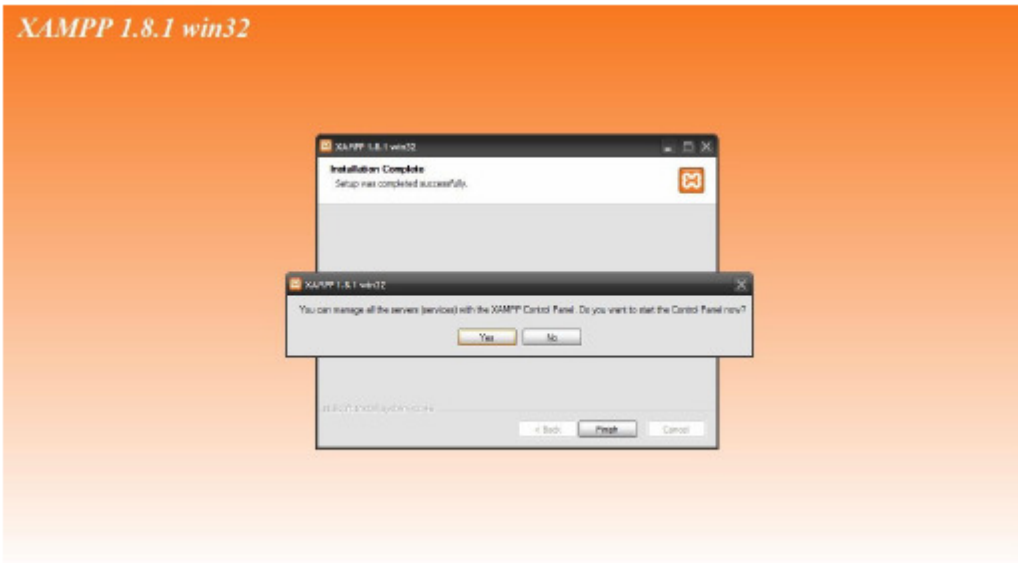


Πατάμε Install

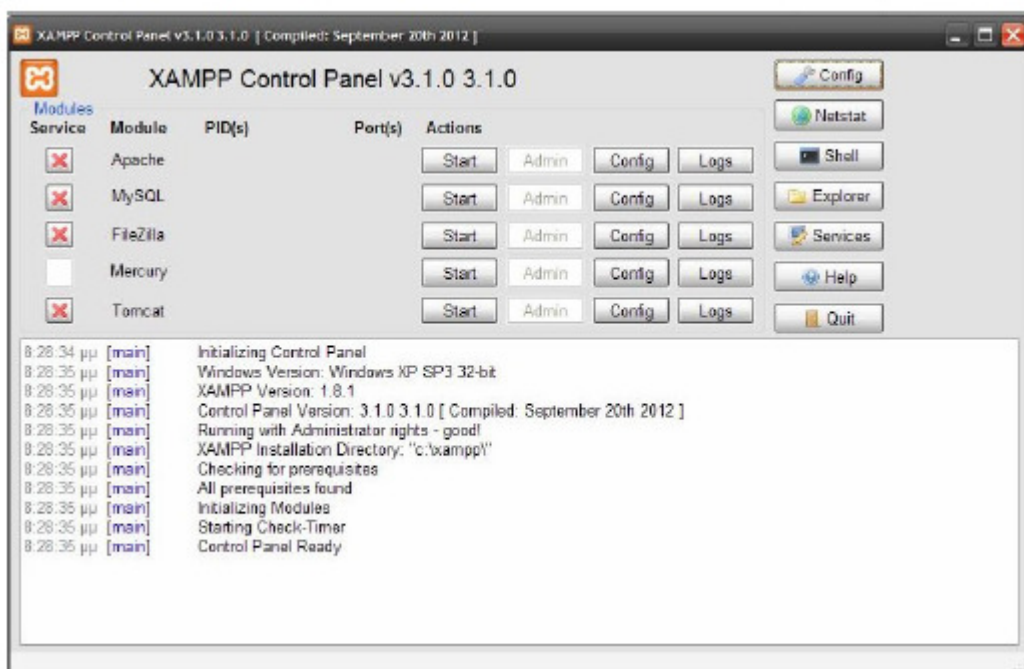
XAMPP 1.8.1 win32



Και όταν ολοκληρωθεί η εγκατάσταση επιλέγουμε finish και εμφανίζεται το παρακάτω παράθυρο



Αν επιλέξουμε "Yes" ανοίγει το Control Panel



Και κάθε φορά που θα θέλουμε να χρησιμοποιήσουμε Apache Server και τις βάσεις MySQL θα ανοίγουμε το XAMPP που έχει εγκατασταθεί σαν πρόγραμμα πλέον στο pc και θα πατάμε Start στις επιλογές Apache και MySQL, όπως φαίνεται και στην εικόνα παραπάνω.

Αφού γίνει σωστά η εγκατάσταση αν ανοίξουμε έναν browser και πληκτρολογήσουμε στη μπάρα διεύθυνσης <http://localhost> ή <http://127.0.0.1> τότε ανοίγει η σελίδα <http://localhost/xampp> ή <http://127.0.0.1/xampp> αντίστοιχα.

XAMPP for Windows

Welcome to XAMPP for Windows!

Congratulations:
You have successfully installed XAMPP on this system!

++++ +++++ a great thank you to Ackettock142 for this new fine Control Panel! +++++ +++++

XAMPP-Status

Alle diese Dienste laufen nun neben der XAMPP-Installation gestartet und bzw. welche Funktionen, welche Werte an der Konfiguration von XAMPP übergeben werden, sind hier, CPU und RAM-Auslastung.

MySQL - MySQLdb **Running** **MySQL**

PHP - PHP5 **Running** **PHP5**

Perl - Perl **Running** **Perl**

Tomcat - Tomcat **Running** **Tomcat**

Index	Status	PROZESS	PORTEN	ADMIN	ADMIN	ADMIN	ADMIN	ADMIN	ADMIN
1	Running	MySQL	4329	3306	Admin	Admin	Admin	Admin	Admin
2	Running	PHP5	4372	3306	Admin	Admin	Admin	Admin	Admin
3	Running	Perl	4329	3306	Admin	Admin	Admin	Admin	Admin
4	Running	Tomcat	4329	8080, 8005, 8006	Admin	Admin	Admin	Admin	Admin

Κεφάλαιο 3

Ανάπτυξη και Περιγραφή της Εφαρμογής

Στο μέρος αυτό θα δούμε πως δομείται το site, τις ιστοσελίδες που το αποτελούν, το περιεχόμενο κάθε μιας και τον κώδικα που αποτελεί κάθε μια. Θα δούμε τι μπορεί να κάνει ο διαχειριστής- administrator του site στο περιεχόμενο και με ποιο τρόπο. Τα βασικά θέματα αυτού του μέρους είναι :

- Ενσωμάτωση PHP στην HTML
- Επαναχρησιμοποίηση κώδικα και σύνταξη συναρτήσεων
- Πρόσθεση δυναμικού περιεχομένου
- Αναγνωριστικά (id)
- Μεταβλητές που δηλώνονται από τον χρήστη
- Τύποι μεταβλητών
- Εκχώρηση τιμών σε μεταβλητές
- Αποφάσεις με τις if , else
- Πίνακες
- Χρήση Sessions

3.1 Δομή του site

Το site είναι δομημένο ως εξής:

- Μια κοινή αρχική σελίδα στην οποία υπάρχουν διάφοροι τύποι μενού επιλογών (π.χ. drop down menu) για να κάνουμε τις αναζητήσεις(π.χ. κατηγορία, περιοχή, τιμή, uploading date, είδος) και να μπορεί ο επισκέπτης να βλέπει συγκεκριμένο υποσύνολο των προϊόντων.
- Υπάρχουν 3 κατηγορίες χρηστών, Καταστήματα-εμποροι, επισκέπτες, admin. Εκτός από τους επισκέπτες οι άλλοι χρήστες πρέπει να κάνουν login για να ταυτοποιηθεί ποιοι είναι κ για να έχουν κάποια προνόμια σε σχέση με τους επισκέπτες.
- Οι επισκέπτες δεν έχουν δυνατότητα να αλλάξουν κάτι στο e-κατάστημα ή στα προϊόντα. Μπορούν όμως να αφήσουν κάποιο σχόλιο ή like στα προϊόντα κ να στείλουν κ κάποιο σύντομο μήνυμα στους εμπορους. Ακόμα υπάρχει φόρμα για να στέλνουν μήνυμα στους εμπορους
- Τα Καταστήματα-εμποροι έχουν την δυνατότητα να κάνουν κάποιες αλλαγές στο e-κατάστημα τους(π.χ. διαθεσιμότητα, τιμή προϊόντων). Για την δημιουργία e-καταστήματος χρειάζεται εγγραφή με χρήση διάφορων πεδίων. Υπάρχουν πεδία σύνδεσης των εμπόρων(όνομα, pass) για να μπορούν να συνδεθούν και αν συνδεθούν επιτυχώς να μπορούν να κάνουν αλλαγές στο e-κατάστημα τους.
- Υπάρχουν διάφοροι τύποι μενού επιλογών (π.χ. drop down menu) για να κάνουμε τις αναζητήσεις(π.χ. κατηγορία, περιοχή, τιμή, uploading

date, ειδος) και να μπορεί ο επισκέπτης να βλέπει συγκεκριμένο υποσύνολο των προϊόντων.

- Γίνετε χρήση Google MAPs API με πλοήγηση, και με πινέζες με μικρογραφίες και πληροφορίες. Επίσης υπάρχει η δυνατότητα να εμφανίζονται κάποιες πινέζες (καταστήματα) ή όχι ανάλογα με την αναζήτηση (π.χ. δείξε όλα τα καταστήματα, μόνο τα υποδήματα, μόνο τα αξεσουάρ).
- Αποτελείται από κάποιους πίνακες στη βάση δεδομένων που περιέχουν τα καταστήματα-εμπόρους και όλα τα στοιχεία για αυτά καθώς και όλα τους τα προϊόντα με διάφορες πληροφορίες για το καθένα ξεχωριστά, με το περιεχόμενό τους να διαχειρίζεται μόνο ο administrator του site ή τα καταστήματα-έμποροι αντίστοιχα.

3.2 Χρήση αντικειμενοστρεφούς κώδικα για την ανάπτυξη της εφαρμογής

Για την ανάπτυξη του site χρησιμοποιήθηκε και αντικειμενοστρεφής (object oriented -oop) προγραμματισμός με PHP. Οι μοντέρνες γλώσσες προγραμματισμού υποστηρίζουν ή και ακόμα και απαιτούν αντικειμενοστρεφή προσέγγιση στην ανάπτυξη προγραμμάτων, εφαρμογών και ιστοσελίδων. Η αντικειμενοστρεφής ανάπτυξη προσπαθεί να χρησιμοποιεί κλάσεις, σχέσεις και ιδιότητες αντικειμένων του συστήματος σαν βοήθημα στην ανάπτυξη εφαρμογών, projects κτλ.

Το αντικειμενοστρεφές λογισμικό έχει σχεδιαστεί και δημιουργηθεί ως ένα σύνολο από αυτόνομα αντικείμενα με ιδιότητες και λειτουργίες, που αλληλεπιδρούν προκειμένου να ανταποκριθούν στις ανάγκες για τις οποίες κατασκευάστηκε. Η αντικειμενοστρεφής προσέγγιση μπορεί να βοηθήσει στη διαχείριση της πολυπλοκότητας ενός «έργου», να αυξήσει την επαναχρησιμοποίηση του κώδικα και έτσι να περιορίσει και το κόστος συντήρησης.

Τα αντικείμενα μπορούν να ομαδοποιηθούν σε κλάσεις. Οι κλάσεις αντιπροσωπεύουν ένα σύνολο από αντικείμενα που μπορεί να είναι μόνο ατομικά, αλλά πρέπει να έχουν και κάποια κοινά στοιχεία. Μια κλάση περιέχει αντικείμενα που έχουν όλα τις ίδιες λειτουργίες, που συμπεριφέρονται με τον ίδιο τρόπο και οι ίδιες ιδιότητες αντιπροσωπεύουν τα ίδια πράγματα, αν και οι τιμές αυτών των ιδιοτήτων διαφέρουν από αντικείμενο σε αντικείμενο.

Παράδειγμα το ποδήλατο μπορεί να είναι μια κλάση από αντικείμενα, που περιγράφουν πολλά διαφορετικά ποδήλατα, με κοινά χαρακτηριστικά ή ιδιότητες όπως δυο τροχούς, χρώμα, μέγεθος και λειτουργίες όπως κίνηση.

Μια αντικειμενοστρεφής γλώσσα προγραμματισμού μπορεί να υποστηρίξει πολυμορφισμό, που σημαίνει ότι διαφορετικές κλάσεις μπορούν να έχουν διαφορετικές συμπεριφορές για την ίδια λειτουργία. Για παράδειγμα, μια κλάση αυτοκινήτου και μια κλάση ποδηλάτου έχουν και οι δυο διαφορετικές λειτουργίες κίνησης.

Η κληρονομικότητα επιτρέπει τη δημιουργία μιας ιεραρχικής σχέσης μεταξύ κλάσεων χρησιμοποιώντας υποκλάσεις. Μια υποκλάση κληρονομεί ιδιότητες και λειτουργίες από την υπερκλάση της. Για παράδειγμα, το

αυτοκίνητο και το ποδήλατο έχουν κάποια κοινά. Θα μπορούσε να υπάρχει μια κλάση "όχημα" που περιέχει την ιδιότητα του χρώματος και μια λειτουργία κίνησης, την οποία έχουν όλα τα οχήματα και την οποία κληρονομούν τα αυτοκίνητα και τα ποδήλατα.

Κεντρική ιδέα στον αντικειμενοστρεφή προγραμματισμό είναι η **κλάση** (class), μία αυτοτελής αναπαράσταση κάποιας κατηγορίας αντικειμένων, είτε φυσικών αντικειμένων του πραγματικού κόσμου είτε νοητών, εννοιολογικών αντικειμένων, σε ένα περιβάλλον προγραμματισμού. Πρακτικώς, είναι ένας τύπος δεδομένων, ή αλλιώς το προσχέδιο μίας δομής δεδομένων με δικά της περιεχόμενα, τόσο μεταβλητές όσο και διαδικασίες. Τα περιεχόμενα αυτά δηλώνονται είτε ως *δημόσια* (public) είτε ως *ιδιωτικά* (private), με τα ιδιωτικά να μην είναι προσπελάσιμα από κώδικα εκτός της κλάσης. Οι διαδικασίες των κλάσεων συνήθως καλούνται *μέθοδοι* (methods) και οι μεταβλητές τους *γνωρίσματα* (attributes) ή *πεδία* (fields). Μία κλάση πρέπει ιδανικά να είναι εννοιολογικά αυτοτελής, να περιέχει δηλαδή μόνο πεδία τα οποία περιγράφουν μία κατηγορία αντικειμένων και δημόσιες μεθόδους, οι οποίες επενεργούν σε αυτά όταν καλούνται από το εξωτερικό πρόγραμμα, χωρίς να εξαρτώνται από άλλα δεδομένα ή κώδικα εκτός της κλάσης, και να είναι επαναχρησιμοποιήσιμη.

Αντικείμενο (object) είναι το στιγμιότυπο μίας κλάσης, δηλαδή η δομή δεδομένων βασισμένη στο «καλούπι» που προσφέρει η κλάση. Παραδείγματος χάρη, σε μία αντικειμενοστραφή γλώσσα προγραμματισμού θα μπορούσαμε να ορίσουμε κάποια κλάση ονόματι BankAccount, η οποία αναπαριστά έναν τραπεζικό λογαριασμό, και να δηλώσουμε ένα αντικείμενο της με όνομα MyAccount. Το αντικείμενο αυτό θα έχει δεσμεύσει χώρο στη μνήμη με βάση τις μεταβλητές και τις μεθόδους που περιγράψαμε όταν δηλώσαμε την κλάση. Έτσι, στο αντικείμενο θα μπορούσε να περιέχεται ένα γνώρισμα Balance (=υπόλοιπο) και μία μέθοδος GetBalance (=επίστρεψε το υπόλοιπο). Ακολούθως, θα μπορούσαμε να δημιουργήσουμε ακόμα ένα ή περισσότερα αντικείμενα της ίδιας κλάσης τα οποία θα είναι διαφορετικές δομές δεδομένων (διαφορετικοί τραπεζικοί λογαριασμοί στο παράδειγμα). Ας σημειωθεί εδώ πως τα αντικείμενα μίας κλάσης μπορούν να προσπελάσουν τα ιδιωτικά περιεχόμενα άλλων αντικειμένων της ίδιας κλάσης.

Κληρονομικότητα ονομάζεται η ιδιότητα των κλάσεων να επεκτείνονται σε νέες κλάσεις, ρητά δηλωμένες ως κληρονόμους (*υποκλάσεις* ή 'θυγατρικές κλάσεις'), οι οποίες μπορούν να επαναχρησιμοποιήσουν τις μεταβιβάσιμες μεθόδους και ιδιότητες της γονικής τους κλάσης, αλλά και να προσθέσουν δικές τους. Στιγμιότυπα των θυγατρικών κλάσεων μπορούν να χρησιμοποιηθούν όπου απαιτούνται στιγμιότυπα των γονικών (εφόσον η θυγατρική είναι κατά κάποιον τρόπο μία πιο εξειδικευμένη εκδοχή της γονικής), αλλά το αντίστροφο δεν ισχύει. Παράδειγμα κληρονομικότητας είναι μία γονική κλάση Vehicle (=Όχημα) και οι δύο πιο εξειδικευμένες υποκλάσεις της Car (=Αυτοκίνητο) και Bicycle (=Ποδήλατο), οι οποίες λέμε ότι "κληρονομούν" από αυτήν. Πολλαπλή κληρονομικότητα είναι η δυνατότητα που προσφέρουν ορισμένες γλώσσες προγραμματισμού μία κλάση να κληρονομεί ταυτόχρονα από περισσότερες από μία γονικές. Από μία υποκλάση μπορούν να προκύψουν νέες υποκλάσεις που κληρονομούν από αυτήν, με αποτέλεσμα μία ιεραρχία κλάσεων που συνδέονται μεταξύ τους "ανά γενιά" με σχέσεις κληρονομικότητας.

3.2.1 Δομή μιας κλάσης

Ένας απλός ορισμός κλάσης μπορεί να είναι κ ο εξής:

```
class classname
```

```
{  
}
```

Αν τώρα θέλουμε να ορίσουμε μια κλάση με όνομα `classname1` με δυο ιδιότητες, τις `$attribute1`, `$attribute2` με την πρώτη να είναι δημόσια ορατή και τη δεύτερη όχι, τότε γράφουμε:

```
class classname1
```

```
{  
  
    public $attribute1;  
  
    private $attribute2;  
  
}
```

Αν θέλουμε να δημιουργήσουμε και λειτουργίες (συναρτήσεις) σε μια κλάση με όνομα `classname2`, όπου η μια `operation1()` δεν παίρνει παραμέτρους και η `operation2()` παίρνει δύο, γράφουμε:

```
class classname2
```

```
{  
  
    function operation1()  
  
    {  
  
    }  
  
    function operation2($param1, $param2) { }  
  
}
```

Η δημιουργία σελίδων μέσω PHP κλάσεων έχει προφανή πλεονεκτήματα. Έχοντας μια κλάση που περιέχει τις περισσότερες συναρτήσεις και αντικείμενα για έναν ιστότοπο, χρειάζεται λιγότερη δουλειά στη δημιουργία

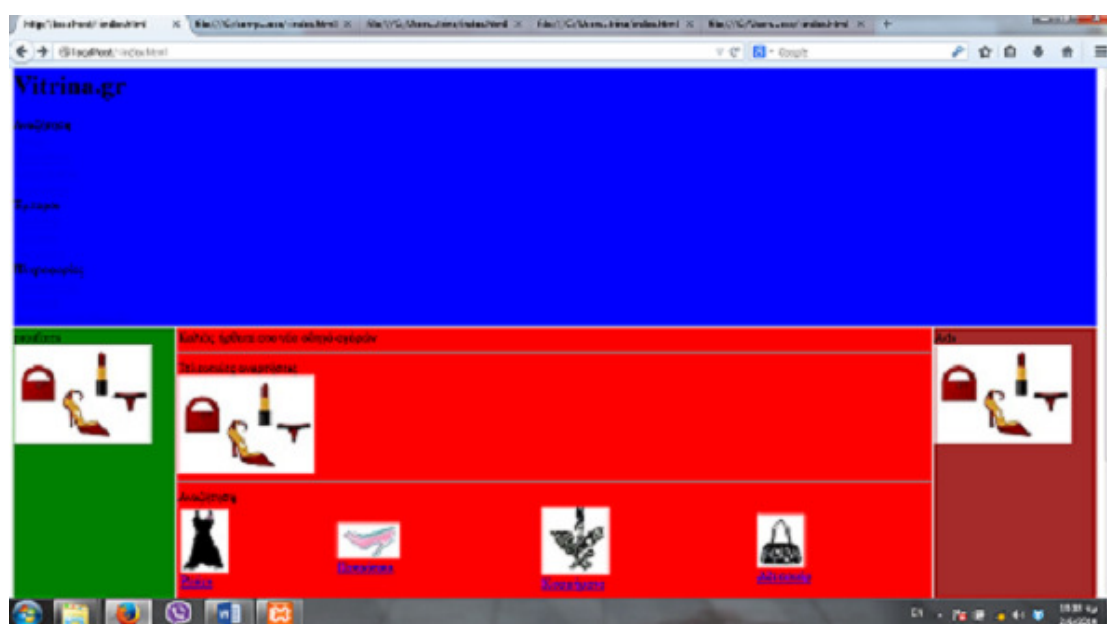
νέας σελίδας. Ενημερώνοντας την κλάση ενημερώνονται όλες οι σελίδες ταυτόχρονα. Χρησιμοποιώντας κληρονομικότητα, παράγονται διαφορετικές εκδόσεις της αρχικής κλάσης χωρίς να χάνονται τα πλεονεκτήματα των κλάσεων.

Με αυτό τον τρόπο θα χρησιμοποιήσουμε για τις περισσότερες σελίδες του ιστοτόπου κοινό κώδικα χρησιμοποιώντας αντικειμενοστρεφή προγραμματισμό για την PHP.

3.3. Ανάπτυξη της εφαρμογής και ανάλυση του κώδικα

3.3.1 Αρχική σελίδα – Home

Η πρώτη εικόνα που θα δει ο χρήστης που επισκέπτεται το site είναι η παρακάτω:



Αρχική σελίδα – index.php

Από την αρχική σελίδα όπως βλέπουμε μπορούμε να πλοηγηθούμε σε όλο το site και να αναζητήσουμε προϊόντα αναμεσα σε όλα τα καταστήματα-έμποροι.

Ο κώδικας της αρχικής σελίδας είναι :

```
<html>
<head>
<script type="text/javascript">
<!--
/* we'll only allow DOM browsers to simplify things*/
(document.getElementById ? DOMCapable = true : DOMCapable = false);

function hide(menuName)
{
```

```

    if (DOMCapable)
    {
        var theMenu = document.getElementById(menuName+"choices");
        theMenu.style.visibility = 'hidden';
    }
}

function show(menuName)

{
    if (DOMCapable)
    {
        var theMenu = document.getElementById(menuName+"choices");
        theMenu.style.visibility = 'visible';
    }
}

```

//-->

Με αυτό το javascript όταν ο χρήστης περνά το ποντίκι πάνω σε ένα html στοιχείο, εκτελείται το script και από κρυφό το μενού γίνεται φανερό. Συγκεκριμένα η συνάρτηση hide(menuName) είναι αυτή που κρύβει το μενού και η συνάρτηση show(menuName) είναι αυτή που το φανερώνει.

```

</script>
</head>
<body>

```

```

<table width="100%" Height="100%" border="0">
<tr>  <!-- Top -->

```

```

    <td colspan="3" style="background-color:blue;" width="100%">
    <h1>Vitrina.gr</h1>

```

```

<div id="menu1" class="menu" onmouseover="show('menu1');"
onmouseout="hide('menu1');">
<div class="menuHead">Αναζήτηση</div>
<div id="menu1choices" class="menuChoices">
<a href="Roux.html">Ρούχα</a><br />
<a href="Shoes.html">Παπούτσια</a><br />
<a href="Kosmimata.html">Κοσμήματα</a><br />
<a href="Accesories.html">Accesories</a><br />
</div>
</div>
<div id="d" class="menu" onmouseover="show('menu2');"
onmouseout="hide('menu2');">
<div class="menuHead">Έμποροι</div>
<div id="menu2choices" class="menuChoices">

```

```

<a href="Login.html">Συνδεση</a><br />
<a href="Emporoi.html">Εμποροι</a><br />
<a href="Map.html">Στο χάρτη</a><br />
</div>
</div>
<div id="menu3" class="menu" onmouseover="show('menu3');"
onmouseout="hide('menu3');">
<div class="menuHead">Πληροφορίες</div>
<div id="menu3choices" class="menuChoices">
<a href="Contact.html">Επικοινωνία</a><br />
<a href="SignIn.html">Εγγραφή</a><br />
<a href="Fee.html">Ανανέωση συνδρομής</a><br />
</div>
</div>

```

Τα πιο πάνω μενού (menu1, menu2, menu3) Είναι μενού που για όσο χρονικό διάστημα το ποντίκι δεν περνά από πάνω τους είναι κρυφά ενώ όταν περνά το ποντίκι από πάνω τους γίνονται φανερά. Αυτό επιτυγχάνεται με κλήση της hide και της show συνάρτησης javascript αντίστοιχα.

```

</td>
</tr>
<tr valign="top">
<!-- Prosfores -->
<td style="background-color:green;width:50px;text-align:top;">
prosfores <br> 
</td>

<!-- Vitrina -->
<td valign="left" style="background-
color:red;height:500px;width:70%;text-align:top;">
<div> Καλώς ήρθατε στο νέο οδηγό αγόρων</div>
<HR>
<div> Τελευταίες αναρτήσεις <br>
 </div>
<HR>
<div> Αναζήτηση <br>
<table width="100%" border="0" valign="left">
<tr><td><a href="Rouxa.html">
</a><br clear="all"><a href="Rouxa.html">
Πούχα</a></td>
<td><a href="Shoes.html"> </a><br
clear="all"><a href="Shoes.html"> Παπούτσια</td>
<td><a href="Kosmimata.html"> </a><br clear="all"><a
href="Kosmimata.html"> Κοσμήματα </td>
<td><a href="Accesories.html"> </a><br clear="all"><a
href="Accesories.html"> Αξεσουάρ </td></tr>

```



```

        </table></div>
    </td>

    <!-- Ads -->
    <td style="background-
color:brown;height:500px;width:100px;text-align:top;">
    Ads <br> 
    </td>
</tr>

<tr>

    <!-- Bottom -->
    <td colspan="3" style="background-color:orange;text-
align:center;">
    Copyright © Xristos
    </td>
</tr>
</table>

<script type="text/javascript">
<!--
/* Don't hide menus for JS off and older browsers */
if (DOMCapable)
{
    hide("menu1");
    hide("menu2");
    hide("menu3");
}
//-->

Αυτό το script χρειάζεται έτσι ώστε όταν το ποντίκι περάσει κ φύγει
από τα μενού αυτά να γίνουν κ πάλι κρυφά.

</script>

</body>
</html>

```

Κεφάλαιο 4

Επισκέπτες της Εφαρμογής

Στο μέρος αυτό θα δούμε πως μπορεί ο επισκέπτης του site να βλέπει συγκεκριμένο υποσύνολο των προϊόντων κάνοντας διάφορες αναζητήσεις, χρησιμοποιώντας διάφορους τύπους μενού επιλογών. Ακόμα θα δούμε πως μπορούν να αφήσουν κάποιο σχόλιο ή like στα προϊόντα και να στείλουν και κάποιο σύντομο μήνυμα στους εμπόρους. Επίσης υπάρχει φόρμα για να στέλνουν μήνυμα στους εμπόρους

4.1 Η menu.php

Υπάρχουν κάποια μενού στην menu.php από τα οποία ο επισκέπτης μπορεί να κάνει τις αναζητήσεις του καθορίζοντας κάθε φορά πιο υποσύνολο των προϊόντων θέλει να δει μέσω πολυκριτηριακής αναζήτησης. Συγκεκριμένα ο κώδικας της menu.php είναι ο πιο κάτω:

```
<html>
<head>
</head>
<body>

<form name="input" action="katasthmata.php" method="post">

<input type="radio" name="Sex" value="male" checked/> Male1<br
/>

<input type="radio" name="Sex" value="female" /> Female<br /><br
/>

Choose kathgoria...:

<SELECT NAME="Kathgoria">
<OPTION VALUE="ola"           >ola
<OPTION VALUE="Rouxa"        >Rouxa
<OPTION VALUE="Papoutsia"    >Papoutsia
<OPTION VALUE="kosmhmata"    >kosmhmata
<OPTION VALUE="Ajesoyar"     >Ajesoyar
</SELECT>

</br>

Choose Perioxh...:
```

```
<SELECT NAME="Perioxh">
<OPTION VALUE="ola"           >ola
<OPTION VALUE="Ermou"        >Suntagma-Ermou
<OPTION VALUE="Aioly"        >A8hna-Aioly
<OPTION VALUE="Per"          >Petrupolh-Petrupolews
</SELECT>
```

</br>

Choose Timi apo...:

```
<SELECT NAME="TimiMin">
<OPTION VALUE="0"           >ola
<OPTION VALUE="5"           >5
<OPTION VALUE="20"          >20
<OPTION VALUE="50"          >50
<OPTION VALUE="100"         >100
</SELECT>
```

</br>

Choose Timi ews...:

```
<SELECT NAME="TimiMax">
<OPTION VALUE="1000000"     >ola
<OPTION VALUE="5"           >5
<OPTION VALUE="20"          >20
<OPTION VALUE="50"          >50
<OPTION VALUE="100"         >100
</SELECT>
```

</br>

```
Typos: <input type="text" NAME="Stoixeia" />
<input type="submit" value="Submit1" />
```

</form>

```
</body>
<head>
```

Στην ουσία η menu.php είναι μια φόρμα. Οι φόρμες δημιουργούνται για την παροχή διάφορων πεδίων στα οποία ο χρήστης μπορεί να επιλέξει ανάμεσα σε προκαθορισμένες επιλογές ή να εισάγει πληροφορίες. Στην menu.php όπως είδαμε έχουμε μια επιλογή τύπου radio, τέσσερες επιλογές τύπου select, δηλαδή με λίστα επιλογών, και μία επιλογή τύπου text, δηλαδή με εισαγωγή κειμένου.

Ανοίγοντας την φόρμα βλέπουμε την ετικέτα:

```
<form name="input" action="katasthmata.php" method="post">
```

Όταν χρησιμοποιούμε την ετικέτα <form> καθορίζουμε επίσης και τον τρόπο αποστολής των δεδομένων προς τον εξυπηρετητή με την χρήση των χαρακτηριστικών method και action της ετικέτας.

Εδώ **action="katasthmata.php"** που στην ουσία δείχνει, με μορφή URL το πρόγραμμα που θα δεχθεί τις πληροφορίες που θα αποσταλούν και θα μπορεί να τις επεξεργαστεί, στην περίπτωση μας την katasthmata.php.

Επίσης το χαρακτηριστικό **method="post"** που σημαίνει ότι οι πληροφορίες θα σταλούν με την μέθοδο post, που είναι η συνηθέστερη και στέλνει τα δεδομένα της φόρμας χωριστά από το URL.

4.2 Η katasthmata.php

Τα δεδομένα που θα σταλούν με την post από την menu.php θα πάνε στην katasthmata.php. Αναλυτικά ο κώδικας της katasthmata.php είναι ο πιο κάτω:

```
<html>
<head>
<title>Vitrina.gr</title>

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

    function jumpto(){

        if (document.forma.jumpmenu.value != "null") {
            document.location.href = 1ed.php
        }
    }

    function DoNav()
    {
        document.location.href = 'http://www.yahoo.com/';
    }

</script>
</head>
<body>
```

```

<?php

$Stoixeia=$_POST['Stoixeia'];
$con = mysql_connect("localhost","root","");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}

mysql_select_db("vitrina", $con);

$result=mysql_query("SELECT * FROM eshop WHERE Stoixeia LIKE
'%$Stoixeia%'");

echo "<table border='1'>
    <tr>
        <th>eikona</th><th>stoixeia</th><th>katasthma</th><th>perioxh
</th><th>timi</th>
    </tr>";

$row = mysql_fetch_array($result);
while($row)
{
    if($row['Dia8esimothta']== '1'
    && $_POST['Perioxh']== $row['Perioxi']
    && $_POST['Sex'] == $row['Sex']
    && $_POST['Kathgoria']==$row['Kathgoria']
    && $_POST['TimiMin'] <= $row['Timi']
    && $_POST['TimiMax'] >= $row['Timi'])
    {
        echo "<tr>";
        echo "    <td><a href= Proion.php?ID=" .
urlencode($row['ID'])."> </a></td>";
        echo "<td>" . $row['Stoixeia'] . "</a></td>";
        echo "<td>" . $row['Katasthma'] . "</td>";
        echo "<td>" . $row['Perioxi'] . "</td>";
        echo "<td>" . $row['Timi'] . "</td>";
        echo "</tr>";
    }

    if($row['Dia8esimothta']== '1'
    && $_POST['Perioxh']== "ola"
    && $_POST['Sex'] == $row['Sex']
    && $_POST['Kathgoria']==$row['Kathgoria']
    && $_POST['TimiMin'] <= $row['Timi']
    && $_POST['TimiMax'] >= $row['Timi'])

```

```

        {
            echo "<tr>";
            echo "<td><a href= Proion.php?ID="
urlencode($row['ID'])."> </a></td>";
            echo "<td>" . $row['Stoixeia'] . "</td>";
            echo "<td>" . $row['Katasthma'] . "</td>";
            echo "<td>" . $row['Perioxi'] . "</td>";
            echo "<td>" . $row['Timi'] . "</td>";
            echo "</tr>";
        }

        if($row['Dia8esimothta']== '1'
        && $_POST['Perioxi']== $row['Perioxi']
        && $_POST['Sex'] == $row['Sex']
        && $_POST['Kathgoria']=="ola"
        && $_POST['TimiMin'] <= $row['Timi']
        && $_POST['TimiMax'] >= $row['Timi'])

        {
            echo "<tr>";
            echo "<td><a href= Proion.php?ID="
urlencode($row['ID'])."> </a></td>";
            echo "<td>" . $row['Stoixeia'] . "</td>";
            echo "<td>" . $row['Katasthma'] . "</td>";
            echo "<td>" . $row['Perioxi'] . "</td>";
            echo "<td>" . $row['Timi'] . "</td>";
            echo "</tr>";
        }

        if($row['Dia8esimothta']== '1'
        && $_POST['Perioxi']=="ola"
        && $_POST['Sex'] == $row['Sex']
        && $_POST['Kathgoria']=="ola"
        && $_POST['TimiMin'] <= $row['Timi']
        && $_POST['TimiMax'] >= $row['Timi'])

        {
            echo "<tr>";
            echo "<td><a href= Proion.php?ID="
urlencode($row['ID'])."> </a></td>";
            echo "<td>" . $row['Stoixeia'] . "</td>";
            echo "<td>" . $row['Katasthma'] . "</td>";
            echo "<td>" . $row['Perioxi'] . "</td>";
            echo "<td>" . $row['Timi'] . "</td>";
            echo "</tr>";
        }

```

```
    $row=mysql_fetch_array($result);  
  }  
  echo "</table>";
```

```
mysql_close($con);
```

```
?>
```

```
</body>  
</html>
```

Όπως βλέπουμε η `katasthmata.php` είναι ένας πίνακας στον οποίο επιστρέφονται τα προϊόντα που πληρούν τις προϋποθέσεις της πολυκριτηριακής αναζήτησης.

Πιο αναλυτικά το πρώτο πράγμα που πρέπει να γίνει είναι η σύνδεση της Php με την MySQL. Αυτό γίνεται με την `mysql_connect`.

```
$con = mysql_connect("localhost","root","");
```

Το `localhost` είναι το όνομα του server που συνδεόμαστε και `root` είναι το όνομα του χρήστη για την Mysql. Κωδικός χρήστη δεν υπάρχει. Η `mysql_connect` επιστρέφει `true` ή `false`. Η `if` κάνει έλεγχο και σε περίπτωση που αποτύχει η σύνδεση με την βάση δεδομένων επιστρέφει `false`, δηλαδή `mysql error`.

```
if (!$con)  
{  
    die('Could not connect: ' . mysql_error());  
}
```

Μετά επιλέγουμε την βάση δεδομένων που θα χρησιμοποιήσουμε, στην περίπτωση μας την βάση `"vitrina"` :

```
mysql_select_db("vitrina", $con);
```

Κάνουμε μια αναζήτηση στον πίνακα `eshop` και ξεχωρίζουμε τα προϊόντα που στα στοιχεία περιέχουν το κείμενο που έχουμε γράψει στη `text menu.php`

Μετά με τις `if` εμφανίζουμε τα προϊόντα που πληρούν όλα τα κριτήρια της αναζήτησης

Τέλος με την `mysql_close` κλείνουμε την ανοιχτή σύνδεση με την MySQL.

```
mysql_close($con);
```

4.3 H `proion.php`

Εάν ο επισκέπτης θέλει να δει πιο αναλυτικά κάποιο από τα προϊόντα που εμφανίζονται στον πίνακα μπορεί να κάνει κλικ επάνω του.

```

echo "<td>
<a href= Proion.php?ID=" . urlencode($row['ID']).">
 </a></td>";

```

Στην περίπτωση αυτή βλέπουμε ότι αν κάνουμε κλικ πάνω στην εικόνα του προϊόντος καλούμε την proion.php . Ο κώδικας της proion.php είναι:

```

<html>
<head>

    <script type="text/javascript">
    function OpenNewWindow(bigurl, width, height)
    {
        var newWindow = window.open("", "pictureViewer",
            "location=no, directories=no, fullscreen=no, menubar=no,
status=no, toolbar=no, width=" + width + ", height=" + height + ",
scrollbars=no");
        newWindow.document.writeln("<html>");
        newWindow.document.writeln("<body style='margin: 0 0 0 0;'>");
        newWindow.document.writeln("<a
href='javascript:window.close();'>");
        newWindow.document.writeln("");
        newWindow.document.writeln("</a>");
        newWindow.document.writeln("</body></html>");
        newWindow.document.close();
    }

    </script>

</head>
<body>

<?php

echo "<p>";
echo $_GET['ID'];

echo "</p>";

$ID=$_GET['ID'];

$con = mysql_connect("localhost","root","");

if (!$con)
{
    die('Could not connect: ' . mysql_error());

```



```

}

mysql_select_db("test", $con);

$result=mysql_query("SELECT * FROM proionta WHERE ID ='$ID'");
$row = mysql_fetch_array($result);

if($row)
{
    $Foto=$row['Foto'];
    $Stoixeia=$row['Stoixeia'];
    $Katasthma=$row['Katasthma'];
    $Perioxi=$row['Perioxi'];
    $Timi=$row['Timi'];
    $Kathgoria=$row['Kathgoria'];
    $Xrwma=$row['Xrwma'];
    $Yliko=$row['Yliko'];
    $Like=$row['Like'];

    echo "<p><a href=\"\#\" onclick=\"OpenNewWindow('\" . $Foto. \"', 600,
450); return true;\><img src=\"\" . $Foto . \" width=150 height=150
align=left> </a>
timi:\" . $Timi . \"euro <br/> kathgoria:\" . $Kathgoria .\"<br/> xrwma:\" .
$Xrwma .\" <br/> uliko:\" . $Yliko . \" <br/> stoixeia:\" . //
$Stoixeia .\" <br/> likes:\" . $Like. \"</p>";

    echo " <button onclick=\"location.href='1ThanksLike.php?ID= \" .
urlencode($row['ID']). \" \" >I like it!</button> ";

    echo "<br clear=all> <p> katasthma:\" . $Katasthma . \" </p>";
    echo " <button onclick=\"location.href='1map.php?Katasthma= \" .
urlencode($row['Katasthma']). \" \" > Pou Briskomaste</button>";
    echo "<p>epikoinwnia:</p>";

    $row=mysql_fetch_array($result);
}
mysql_close($con);

?>

<form method="post" action="1sendmail.php">

Email: <input name="email" type="text" /><br />
Message:<br />
<textarea name="message" rows="15" cols="40">

```

```
</textarea><br />
<input type="submit" value="Αποστολη" />
```

```
</form>
```

```
</body>
```

```
</html>
```

Αυτό που κάνει η proioin.php είναι να μας δείχνει το προϊόν που επιλέξαμε αναλυτικότερα και να μας δίνει μερικές δυνατότητες. Οι δυνατότητες είναι κάνοντας κλικ πάνω στην εικόνα να ανοίγει σε καινούριο παράθυρο μεγαλύτερη εικόνα του προϊόντος. Αν θέλουμε μπορούμε να κάνουμε like και μπορούμε να δούμε πόσα like έχουν γίνει στο προϊόν. Ακόμα μπορούμε να στείλουμε κάποιο μήνυμα στον καταστηματούχο.

Η proioin.php ανοίγει το σωστό προϊόν από την βάση, γιατί της περνάμε από την katasthmata την ID του προϊόντος.

Όταν κάνουμε κλικ πάνω στην φωτογραφία του προϊόντος

```
<a href="#" onclick="OpenNewWindow(' . $Foto . ', 600, 450);
return true;">
 </a>
```

Κάνουμε κλίση στην συνάρτηση OpenNewWindow η οποία ανοίγει την φωτογραφία σε καινούριο παράθυρο. Αν κάνουμε πάλι κλικ πάνω στο καινούριο παράθυρο η φωτογραφία κλείνει.

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

```
function OpenNewWindow(bigurl, width, height)
{
var newWindow = window.open("", "pictureViewer",
"location=no, directories=no, fullscreen=no, menubar=no,
status=no, toolbar=no, width=" + width + ", height=" + height + ",
scrollbars=no");
newWindow.document.writeln("<html>");
newWindow.document.writeln("<body style='margin: 0 0 0 0;'>");
newWindow.document.writeln("<a
href='javascript:window.close();'>");
newWindow.document.writeln("");
newWindow.document.writeln("</a>");
newWindow.document.writeln("</body></html>");
newWindow.document.close();
}
```

```
</script>
```

4.4 Η ThanksLike.php

Όταν θέλουμε να κάνουμε like πατάμε πάνω στο “ I like it” κουμπί και κάνει καλείται η ThanksLike.php η οποία αυξάνει κατά ένα τα like που έχουν γίνει στο συγκεκριμένο προϊόν, το αναγνωρίζει από την ID του, και μας ευχαριστεί.

```
echo " <button onclick=\"location.href='ThanksLike.php?
ID= ". urlencode($row['ID']). " \" > I like it!</button> ";
```

Ο κώδικας της ThanksLike.php είναι:

```
<?php
```

```
echo "<br/>";
```

```
$ID=$_GET['ID'];
echo $ID;
```

```
$con = mysql_connect("localhost","root","");
if (!$con)
```

```
{
    die('Could not connect: ' . mysql_error());
}
```

```
mysql_select_db("vitrina", $con);
```

```
mysql_query("UPDATE eshop SET Like= Like++ WHERE ID ='$ID'");
```

```
echo "<p> ευχαριστούμε που δείξατε την προτίμηση σας <br/> </p>";
mysql_close($con);
```

```
echo "<meta http-equiv=\"Refresh\" content=\"2;url=Proion.php?ID=
". urlencode($ID). \" \">>";
```

```
?>
```

Κεφάλαιο 5

Τα Καταστήματα- έμποροι και admin

Τα καταστήματα-έμποροι και ο admin έχουν κάποια προνόμια σε σχέση με τους επισκέπτες και για αυτό το λόγο πρέπει αρχικά να τακτοποιηθεί ότι είναι αυτοί για να χρησιμοποιήσουν τις δυνατότητες τους. Δηλαδή να κάνουν κάποιες αλλαγές στο e-κατάστημα τους(π.χ. διαθεσιμότητα, τιμή προϊόντων).

Για την δημιουργία e-καταστήματος χρειάζεται εγγραφή με χρήση διάφορων πεδίων. Υπάρχουν πεδία σύνδεσης των εμπόρων(όνομα, pass) για να μπορούν να συνδεθούν και αν συνδεθούν επιτυχώς να μπορούν να κάνουν αλλαγές στο e-κατάστημα τους.

5.1 Η login.html

Η ταυτοποίηση γίνεται μέσω της login.html της οποίας ο κώδικας είναι ο πιο κάτω:

```
<div id="content">
<h1>Login</h1>

<form id="form2" name="form2" action = "Pistopoihsh.php"
method="post">

<table id="tab2">
<tr><td align="left">
<p><strong>Username: </strong>
<input name="username1" type="text"
id="username1"/></p></td></tr>
<tr><td align="left">
<p><strong>Password: </strong>
<input name="pass1" type="password" id="pass1"/></p></td></tr>
<tr><td align="center"><p>
<input type="submit" name="Submit" value="Είσοδος"/>
</p>
</td></tr>
</table>
</form>
</div>
</div>
```

Όπως βλέπουμε η login.html είναι μια φόρμα η οποία παίρνει τα δεδομένα που εισάγει ο χρήστης, δηλαδή το username και το password, και μέσω της μεθόδου post τα περνάει στην pistopoihsh.php

5.2 Η pistopoihsh.php

Αυτό που κάνει η pistopoihsh.php είναι να πηγαίνει στην βάση δεδομένων και να κάνει αναζήτηση αν υπάρχει χρήστης, κατάστημα-έμπορος, με αυτά τα username-password.

Ο κώδικας της pistopoihsh.php είναι:

```
<?php

session_start();
$username=$_POST["username1"];
$password=$_POST["pass1"];

mysql_select_db("Katasthmata",
mysql_connect("localhost","root",""));
$user = mysql_real_escape_string($username);
$pass = mysql_real_escape_string($password);
$result = mysql_query("SELECT username FROM admin WHERE
username ='$user' AND password = '$pass'");
$row = mysql_fetch_array($result);

if ($row)

{
//session_start(); $_SESSION['user']=$username;
header('Location: insert.php');
}

else

{
echo "Wrong username/password ";
header('Location: login.html');
}

?>
```

Η αναζήτηση στον πίνακα της βάσης δεδομένων γίνεται με την:

```
$result = mysql_query("SELECT username FROM admin WHERE
username ='$user' AND password = '$pass'");
```

Αν ο χρήστης έχει δώσει σωστά username και password γίνεται κλήση της μεθόδου session_start() και ξεκινάει ένα session.

```
if ($row)
```

```
{
//session_start(); $_SESSION['user']=$username;
header('Location: insert.php');
}
```

Αν τα username και password που έδωσε ο χρήστης είναι λάθος εμφανίζεται η ένδειξη “wrong username/password” και ο χρήστης ανακατευθύνεται στην login.php για να προσπαθήσει ξανά.

else

```
{
echo "Wrong username/password ";
header('Location: login.html');
}
```

5.3 Τι είναι τα sessions

Τα sessions είναι ένας μηχανισμός με τον οποίο μπορείτε να ελέγξετε την ταυτότητα του επισκέπτη όταν αυτός περιηγείται στις σελίδες ενός ιστότοπου. Με την php μπορείτε να δημιουργείτε sessions για τους επισκέπτες και έτσι να ελέγχετε την ταυτότητά τους και κατά συνέπεια την πρόσβαση αυτών σε ευαίσθητες σελίδες.

Τα session χρησιμοποιούν την τεχνολογία των cookies. Δηλαδή κάθε επισκέπτης δέχεται ένα cookie στο μηχάνημά του το οποίο περιέχει έναν αριθμό (id) το οποίο είναι και το αναγνωριστικό του. Ελέγχοντας εσείς το αναγνωριστικό του επισκέπτη μπορείτε να δείτε αν ο επισκέπτης είναι σε session ή όχι.

5.3.1 Πως να δημιουργήσετε ένα session

Για την δημιουργία ενός session στο μηχάνημα ενός χρήστη χρησιμοποιείτε την εντολή: session_start() η οποία πρέπει να βρίσκεται πριν από την ετικέτα <html>.

Η session_start() δημιουργεί ένα cookie στο μηχάνημα του επισκέπτη με όνομα: **PHPSESSID** και μια τυχαία αλφαριθμητικά τιμή της μορφής: **mhjr1dgss5itb12vqqi6j91gr3**

Για να διαβάσετε την τιμή του αναγνωριστικού (δηλαδή του cookie με όνομα: PHPSESSID), χρησιμοποιείτε την εντολή: session_id()

Εάν επομένως ξεκινάτε ως επισκέπτης από τη σελίδα first.php, εκεί δημιουργείται το αναγνωριστικό και με την session_id() μπορείτε να διαβάσετε την τιμή του και να την αποθηκεύσετε σε μια μεταβλητή.

Εάν τώρα μεταφερθείτε σε μια άλλη σελίδα έστω second.php η μεταβλητή και η τιμή που περιέχει χάνεται και έτσι η second.php δεν γνωρίζει αν ο επισκέπτης είναι σε session ή όχι.

Για τον παρπάνω λόγο η php χρησιμοποιεί την global μεταβλητή \$_SESSION στην οποία αποθηκεύεται η τιμή του αναγνωριστικού αλλά παράλληλα μπορείτε να αποθηκεύσετε και άλλα δεδομένα για τον επισκέπτη.

Το αποθηκευμένο αναγνωριστικό διατηρείτε στην μεταβλητή \$_SESSION και δεν χάνεται στη μεταφορά από μια σελίδα σε άλλη.

5.3.2 Παράδειγμα

Έστω η σελίδα first.php η οποία θα πρέπει να αρχίζει με τις δύο παρακάτω γραμμές.

```
<?php
session_start();
$_SESSION['visitor']=session_id();
?>
```

Για να μπορείτε να ελέγχετε τον επισκέπτη στη σελίδα second.php θα πρέπει να υπάρχει ο παρακάτω κώδικας.

```
<?php
session_start();
if(isset($_SESSION['visitor']) && $_SESSION['visitor']==session_id())
{
    echo "welcome";
}
else
{
    echo "sorry not in session";
}
?>
```

5.3.3 Καταστροφή ενός session

Για να καταστρέψετε ένα session χρησιμοποιείτε την εντολή unset();. Για παράδειγμα unset(\$_SESSION['visitor']);. Εναλλάκτικά μπορείτε να χρησιμοποιήσετε την εντολή session_destroy();

Παρατηρήσεις

Όλα τα αρχεία που εμπλέκονται με sessions θα πρέπει να αρχίζουν με την session_start()

Θα πρέπει το μηχανήμα του χρήστη να δέχεται cookies.

Με την session_destroy(); θα διαγράψετε όλα τα αποθηκευμένα δεδομένα.

5.4 Η logout.php

Όταν ο χρήστης θέλει να αποσυνδεθεί, είτε είναι κατάστημα-έμπορος είτε είναι admin, χρησιμοποιούμε την logout.php.

Ο κώδικας της logout.php είναι:

```
<?php
session_start(); session_destroy();
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>
LOGOUT
</title>
</head>

<body>
<p align="center">Ευχαριστούμε που μας επισκεφθήκατε.</p>
<p align="center">Πατήστε <a href="index.php" </a> για επιστροφή
στην αρχική σελίδα </p>
</body>

</html>
```

Αυτό που κάνει η logout.php είναι να καλέσει την session_destroy που κλείνει καταστρέφει το ανοιχτό session και όλα τα άλλα δεδομένα που έχουν αποθηκευτεί στην μεταβλητή \$_session.

Επίσης μετά μας εμφανίζει την ένδειξη 'Ευχαριστούμε που μας επισκεφτήκατε' και μας ανακατευθύνει στην αρχική σελίδα (index.php).

5.5 Η Eggraph.html

Η αρχική εγγραφή των χρηστών, καταστήματα-έμποροι, γίνεται μέσω της Eggraph.html . Η Eggraph.html ζητάει κάποια στοιχεία από τους νέους χρήστες για να τους εγγράψει στις βάσεις δεδομένων του site.

Ο κώδικας της Eggraph.html είναι:

```
<html>

<body>
<title>
test
```



```
</title>
```

```
<form  
id="form1" name="form1" action="eggraph.php" method="post">
```

```
<table>
```

```
<tr>
```

```
<td><strong>Όνομα Καταστήματος:</strong></td>
```

```
<td><input name="Katasthma" type="text" id="Katasthma"/>  
<BR></td>
```

```
</tr>
```

```
<tr>
```

```
<td><strong>Διευθυνση:</strong></td>
```

```
<td><input name="Address" type="text" id="Address"/> <BR></td>  
</tr>
```

```
<tr>
```

```
<td> <input type="radio" name="Perioxi" value="Ermou"  
><strong> Suntagma-Ermou</strong><br /></td>
```

```
<td><input type="radio" name="Perioxi" value="Aioly"  
><strong> A8hna-Aioly</strong><br /></td>
```

```
<td><input type="radio" name="Perioxi" value="Per" /><strong>  
Petroupolh-Petroupolews</strong><br /></td>
```

```
</tr>
```

```
<tr>
```

```
<td><strong>username:</strong></td>
```

```
<td><input  
name="username" type="text" id="username"/><BR></td>
```

```
</tr>
```

```

<tr>

    <td><strong>password:</strong></td>

    <td><input
name="password"type="password"id="password"/><BR></td>

</tr>

<tr>

    <td><strong>confirm password:</strong></td>

    <td><input
name="cpassword"type="password"id="cpassword"/><BR></td>

</tr>

<tr>

    <td><strong>e-mail:</strong></td>

    <td><input name="email"type="text"id="email"/><BR></td>

</tr>
</table>

    <input type="submit"name="Submit"value="Εγγραφή"/>

</form>
</body>
</html>

```

Όπως βλέπουμε εδώ ζητά από τον χρήστη να δώσει κάποιες πληροφορίες-στοιχεία. Τα στοιχεία που θα δώσει στην φόρμα η Eggraph.html θα τα στείλει με την μέθοδο post στην Eggraph.php.

5.6 Η Eggraph.php

Η Eggraph.php όταν θα πάρει τα στοιχεία αφού κάνει κάποιο έλεγχο θα κάνει μια νέα εγγραφή στις βάσεις του site που θα περιέχει τα στοιχεία του νέου καταστήματος-εμπόρου.

Ο κώδικας της Eggraph.php είναι:

```
<?php
```

```
$Katasthma=$_POST["Katasthma"];
```

```

$Address=$_POST["Address"];
$Perioxi=$_POST["Perioxi"];
$username=$_POST["username"];
$password=$_POST["password"];
$cpassword=$_POST["cpassword"];
$email=$_POST["email"];

$con=mysql_connect("localhost","root","");

if ($password !=$cpassword)
{
    header( 'Location: eggraphWP.html' ) ;
    break;
}
if (!$con)
{
    echo "problem".mysql_error();
}
else
{
    mysql_select_db("test",$con);
    mysql_query("INSERT INTO eshops (Katasthma,
Address, Perioxi, Username, Password, Email, Katastash)
VALUES('$Katasthma','$Address','$Perioxi','$username',
'$password','$email', '1')",$con);
}
mysql_close($con);

header( 'Location: Kalwshrthes.html' ) ;

?>

```

Βλέπουμε πως η Eggraph.php ελέγχει αν τα password είναι ίδια και αν δεν είναι καλεί την EggraphWP.html

```

if ($password !=$cpassword)
{
    header( 'Location: eggraphWP.html' ) ;
    break;
}

```

Η eggraphWP.html δεν διαφέρει σε τίποτα από την Eggraph.html εκτός του ότι ζητά από την χρήστη να συμπληρώσει στην φόρμα με το ίδιο password και τις δύο φορές.

Δηλαδή στον κώδικα έχει αλλάξει μόνο αυτό το κομμάτι

```

<tr>

    <td><strong>password:</strong></td>

    <td><input
name="password"type="password"id="password"/><BR></td>
    <td> Παρακαλώ εισαγετε σωστό Password </td>

</tr>

```

Αν η Eggraph.php κάνει τον έλεγχο της και δεν βρει κάποιο λάθος, τότε θα εκτελεστεί το πιο κάτω κομμάτι κώδικα

```

    else
    {
        mysql_select_db("test",$con);
        mysql_query("INSERT INTO eshops
(Katasthma, Address, Perioxi, Username, Password, Email, Katastash)
VALUES
('$Katasthma','$Address','$Perioxi','$username','$password','$email',
'1')",$con);
    }

    mysql_close($con);

    header( 'Location: Kalwshrthes.html' ) ;

```

δηλαδή θα γίνει μια εγγραφή στην βάση δεδομένων με τον νέο χρήστη και τα στοιχεία που μας έδωσε. Μετά θα ανοίξει την Kalwshrthes.html

Ο κώδικας της Kalwshrthes.html είναι ο πιο κάτω:

```

<html>

<head>
</head>

<body>

Kalws hrthes sto Vitrina.gr </br>
An 8es na deis to katasthma sou pata <a href="katasthmatarxes.php">
edw </a>

</body>
</html>

```

Στην ουσία η Kalwshrthes.html δεν κάνει τίποτα άλλο από το να κάνει ένα χαιρετισμό στο νέο κατάστημα-έμπορο και να του λέει αν θέλει να μεταφερθεί στην προσωπική του σελίδα να πατήσει το 'εδώ' έτσι ώστε να καλέσει την `katasthmatarxes.php`

5.7 Η `katasthmatarxes.php`

Η προσωπική σελίδα του κάθε καταστήματος-εμπόρου είναι η `katasthmatarxes.php`. Μέσα από αυτήν μπορεί να δει ο χρήστης το σύνολο των προϊόντων του, να προσθέσει ή να αφαιρέσει κάποιο προϊόν και να ορίσει κάποιο προϊόν σαν διαθέσιμο ή μη.

Αναλυτικά ο κώδικας της `katasthmatarxes.php` είναι:

```
<html>
<head>
<title>Vitrina.gr</title>

</head>
<body>

<?php

$Katasthma="bkat" ; //////////////////////////////////
$con = mysql_connect("localhost","root","");

if (!$con)
{
    die('Could not connect: ' . mysql_error());
}

mysql_select_db("test", $con);

$result=mysql_query("SELECT * FROM proionta WHERE Katasthma =
'$Katasthma'");
echo      '<form      id="forma1"name="forma1"      method="post"
action="Diathesimothta.php" >';

echo "diathesima Proionta";
echo "<table border='1'>
    <tr>
        <th>eikona</th><th>stoixeia</th><th>katasthma</th><th>perioxh
</th><th>timi</th><th>Diathesimothta</th>
    </tr>";

$row = mysql_fetch_array($result);
```

```

while($row)
{
    if($row['Dia8esimothta']== '1')
    {
        echo "<tr>";
        echo "<td>
<a href= Proion.php?ID=" . urlencode($row['ID'])."> </a></td>";
        echo "<td>" . $row['Stoixeia'] . "</a></td>";
        echo "<td>" . $row['Katasthma'] . "</td>";
        echo "<td>" . $row['Perioxi'] . "</td>";
        echo "<td>" . $row['Timi'] . "</td>";
        echo "<td>" .
<input type="radio" name="Epilogi" value='.$row['ID'].>' .<input
type="submit" value=DIATHESIMO>' . "</td>";

        echo "</tr>";
    }

    if($row['Dia8esimothta']== '0')
    {
        echo "<tr>";
        echo "<td>
<a href= Proion.php?ID=" . urlencode($row['ID'])."> </a></td>";
        echo "<td>" . $row['Stoixeia'] . "</a></td>";
        echo "<td>" . $row['Katasthma'] . "</td>";
        echo "<td>" . $row['Perioxi'] . "</td>";
        echo "<td>" . $row['Timi'] . "</td>";
        echo "<td>" .<input type="radio" name="Epilogi"
value='.$row['ID'].>' .<input type="submit" value=MH_DIATHESIMO>' .
"</td>";

        echo "</tr>";
    }

    $row=mysql_fetch_array($result);
}
echo "</table>";

mysql_close($con);

```

?>

```
</body>
</html>
```

Όπως βλέπουμε αρχικά ανοίγουμε την βάση δεδομένων και μετα κάνουμε αναζήτηση για να προβάλλουμε όλα τα προϊόντα του συγκεκριμένου καταστήματος-έμπορου.

```
$con = mysql_connect("localhost","root","");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}
```

```
mysql_select_db("test", $con);
```

```
$result=mysql_query("SELECT * FROM proionta WHERE Katasthma = '$Katasthma'");
```

Σε ένα πίνακα προβάλλουμε όλα τα προιοντα του χρήστη, πρωτα τα διαθέσιμα και μετα τα μη διαθέσιμα

```
while($row)
{
    if($row['Dia8esimothta']== '1')
    {
        echo "<tr>";
        echo "<td>";
        <a href= Proion.php?ID=" . urlencode($row['ID'])."> </a></td>";
        echo "<td>" . $row['Stoixeia'] . "</a></td>";
        echo "<td>" . $row['Katasthma'] . "</td>";
        echo "<td>" . $row['Perioxi'] . "</td>";
        echo "<td>" . $row['Timi'] . "</td>";
        echo "<td>" . <input type="radio" name="Epilogi"
        value='.$row['ID'].>' . <input type="submit" value=DIATHESIMO>' .
        "</td>";
        echo "</tr>";
    }

    if($row['Dia8esimothta']== '0')
    {
        echo "<tr>";
        echo "<td>";
        <a href= Proion.php?ID=" . urlencode($row['ID'])."> </a></td>";
```

```

        echo "<td>" . $row['Stoixeia'] . "</a></td>";
        echo "<td>" . $row['Katasthma'] . "</td>";
        echo "<td>" . $row['Perioxi'] . "</td>";
        echo "<td>" . $row['Timi'] . "</td>";
        echo "<td>" . '<input type="radio" name="Epilogi"
value='. $row['ID']. '>' . '<input type="submit" value="MH_DIATHESIMO">' .
"</td>";
        echo "</tr>";
    }

    $row=mysql_fetch_array($result);
}

```

Πατώντας στο 'DIATHESIMO' ή "MH_DIATHESIMO" ο χρήστης καλεί την Diathesimothta.php η οποία αλλάζει το προϊόν από διαθέσιμο σε μη διαθέσιμο ή το ανάποδο.

```

echo '<form id="forma1" name="forma1" method="post"
action="Diathesimothta.php" >';

```

5.8 Η Diathesimothta.php

Ο κώδικας της Diathesimothta.php είναι:

```
<?php
```

```

$ID=$_POST["Epilogi"];
$i=1;
echo $ID ;
$con=mysql_connect("localhost","root", "");

if (!$con)
{
    echo "problem".mysql_error();
}
else
{
    mysql_select_db("test",$con);

    $result = mysql_query("SELECT * FROM proionta
WHERE ID='$ID'");

```



```

        while($row = mysql_fetch_array($result))
        {
            if ($row['Dia8esimothta']==1)
            {
                mysql_query("UPDATE eshop SET
Dia8esimothta=0 WHERE ID='$ID' && Dia8esimothta=1 ");
            }
            else
            {
                mysql_query("UPDATE eshop SET
Dia8esimothta=1 WHERE ID='$ID' && Dia8esimothta=0 ");
            }
        }

    }
    mysql_close($con);
    echo "<p> H allagh engine</p> ";

    echo "<meta http-equiv=\"Refresh\"
content=2;url=katasthmatarxes.php>";

    ?>

```

Όπως βλέπουμε η Diathesimothta.php κάνει αναζήτηση μέσα στην βάση με όλα τα προϊόντα με κριτήριο την ID του προϊόντος που είναι μοναδική για κάθε προϊόν

```

mysql_select_db("test",$con);

$result = mysql_query("SELECT * FROM proionta WHERE
ID='$ID'");

while($row = mysql_fetch_array($result))
    Στο προϊόν που θα έχει αυτή την ID κάνει έλεγχο αν η diathesimothta
του είναι 1 ή 0 και την αντιστρέφει, Μετατρέποντας το έτσι από διαθέσιμο σε
μη διαθέσιμο και αντίθετα ανάλογα με το την επιλογή του χρήστη.

    if ($row['Dia8esimothta']==1)
    {

        mysql_query("UPDATE eshop SET Dia8esimothta=0 WHERE
ID='$ID' && Dia8esimothta=1 ");
    }

```

```
else
{

mysql_query("UPDATE eshop SET Dia8esimothta=1 WHERE
ID='$ID' && Dia8esimothta=0 ");
}
```

Μόλις γίνει η αλλαγή μας ενημερώνει πως έγινε και αμέσως μετα κάνει ανανέωση στην σελίδα [katasthmatarxes.php](#)

```
mysql_close($con);
echo "<p> H allagh egine</p> ";

echo "<meta http-equiv=
\"Refresh\"content=2;url=katasthmatarxes.php>";
```

Κεφάλαιο 6

Google MAPs API

Το Google MAPs API είναι μια προγραμματιστική διεπαφή (API Application Programming Interface) για την ενσωμάτωση χαρτών Google (Google Maps) σε ιστοσελίδες με χρήση Javascript. Το API περιλαμβάνει ένα σύνολο λειτουργικοτήτων και υπηρεσιών που παρέχουν τη δυνατότητα για χειρισμό χαρτών και τον εμπλουτισμό αυτών με περιεχόμενο. Ο στόχος της χρήσης του API είναι η δημιουργία και παροχή προηγμένων και βασισμένων σε χάρτες εφαρμογών διαδικτύου.

Για να δημιουργήσουμε μια εφαρμογή με Google MAPs πρέπει να αποκτήσουμε ένα κλειδί (Google MAPs API key). Για να αποκτήσουμε αυτό το κλειδί είναι προϋπόθεση ο χρήστης να έχει λογαριασμό στο gmail αλλιώς πρέπει να δημιουργήσει. Για να αποκτήσουμε το κλειδί πάμε στο <http://code.google.com/apis/maps/signup.html> . Δεν υπάρχει κανένας περιορισμός όσον αφορά τα «κλειδιά» που μπορεί να ανακτήσει ένας χρήστης (δηλ. ένας λογαριασμός στο gmail μπορεί να ανακτήσει η «κλειδιά»). Ο μόνος περιορισμός είναι η αντιστοίχιση «1-1» κάθε «κλειδιού» προς κάθε domain.

6.1 Η Gmap.php

Η Gmap.php είναι ανοίγει μια σελίδα που μας δείχνει δυο καταστήματα στον χάρτη, ο κώδικάς της είναι:

```
<html>
<head>
<title> Xarths </title>
<script
src="http://maps.google.com/maps?file=api&v=2&sensor=true
&key=ABQIAAAD3pTC3C_r3fRkBZNpC4PzhT2yXp_ZAY8_ufC3CF
XhHIE1NvwkxTHs43oGTmhkwJHkZMx2LKUIJdb4w"
type="text/javascript"></script>
```

```
<script type="text/javascript">
var map = null;
var geocoder = null;
```

```
function initialize() {
    if(GBrowserIsCompatible()) {
        var point;
        var map=new GMap2(document.getElementById("map"));
        point= new GLatLng(37.4034,22.1401);
        map.setCenter(point,7); //arxiko zoom level=7
    }
}
```

```

var point = new GLatLng(38.27628,23.6696);
var marker1 = new GMarker(point);
map.addOverlay(marker1);
GEvent.addListener(marker1, "click",
function(){marker1.openInfoWindowHtml('Stoixeia katasthmatos ');});

var point = new GLatLng(39.377496,22.933089);
var marker2 = new GMarker(point);
map.addOverlay(marker2);
GEvent.addListener(marker2, "click",
function(){marker2.openInfoWindowHtml('Stoixeia katasthmatos2 
</head>
<body onload="initialize()" onunload="GUnload()" style=" background-
color:transparent">
    <div id="map" style="width: 900px; height: 500px"></div>

</body>
</html>

```

6.2 Η GmapNEW.php

Με την GmapNEW.php παίρνουμε μια διεύθυνση, στην εφαρμογή μας την διεύθυνση ενός καταστήματος, και την βρίσκουμε στον χάρτη βάζοντας πινέξες.

Ο κώδικας της GmapNEW.php είναι:

```

<html>
<head>
<title> Xarths </title>
<script
src="http://maps.google.com/maps?file=api&v=2&sensor=true
&key=ABQIAAAD3pTC3C_r3fRkBZNpC4PzhT2yXp_ZAY8_ufC3CF
XhHIE1NvwkxTHs43oGTmhkwJHkZMx2LKUIJdb4w"
type="text/javascript"></script>

<script type="text/javascript">

var map = new GMap2();;

```

```

map.setCenter(new GLatLng(34, 0), 1);
var geocoder = new GClientGeocoder();

function addAddressToMap(response) {
  map.clearOverlays();
  if (!response || response.Status.code != 200) {
    alert("\\" + address + "\" not found");
  } else {
    place = response.Placemark[0];
    point = new GLatLng(place.Point.coordinates[1],
      place.Point.coordinates[0]);
    marker = new GMarker(point);
    map.addOverlay(marker);
    marker.openInfoWindowHtml(place.address + '<br>' +
      '<b>Country code:</b>' +
      place.AddressDetails.Country.CountryNameCode);
  }
}

// showLocation() kaleitai molis patiseis to Search
// sthn forma. Metatrepei thn dieuyynsh ths formas se
suntetagmenes
// kai bazei marker sthn 8esh toy katasthmatos ston xarth.
function showLocation() {
  var address = document.forms[0].q.value;
  geocoder.getLocationsAsync(address, addAddressToMap);
}

</script>
<!-- Dhmiourgei mia input gia na baleis thn dieuyynsh
pou 8a anazhthsei otan patiseis to submit sth forma. -->
<form action="#" onsubmit="showLocation(); return false;">
  <p>
    <b>Αναζήτηση Διευθυνσης:</b>
    <input type="text" name="q" value="" class="address_input"
size="40" />
    <input type="submit" name="find" value="Anazhthsh" />
  </p>
</form>

</script>
</head>
<body>

```

```
</body>
</html>
```

Όπως βλέπουμε πρόκειται για μια φόρμα που συμπληρώνεις την διεύθυνση και κάνει αναζήτηση μόλις πατήσεις το κουμπί 'αναζήτηση'.

```
<form action="#" onsubmit="showLocation(); return false;">
  <p>
    <b>Αναζήτηση Διευθυνσης:</b>
    <input type="text" name="q" value="" class="address_input"
size="40" />
    <input type="submit" name="find" value="Anazhthsh" />
  </p>
</form>
```

Η αναζήτηση γίνεται από την showLocation() η οποία καλείται μόλις πατήσουμε το κουμπί 'αναζήτηση' στην φόρμα. Αυτό που κάνει είναι να μετατρέπει την διεύθυνση που έχουμε εισαγάγει σε συντεταγμένες και να βάζει σε αυτή την θέση πινέζα (marker). Αυτή η διεύθυνση θα είναι η θέση του καταστήματος στον χάρτη. Την διεύθυνση θα πρέπει να την βρίσκει από τα στοιχεία του καταστήματος-εμπόρου που έχει το site στη βάση δεδομένων του.

```
function showLocation()
{
  var address = document.forms[0].q.value;
  geocoder.getLocationsAsync(address, addAddressToMap);
}
```

Βιβλιογραφία

- [1] Μπουζινόπουλος Ιωάννης, Πτυχιακή Εργασία «Σχεδίαση και Ανάπτυξη Διαδικτυακού Περιβάλλοντος Ενημέρωσης, Παρακολούθησης και Διαχείρισης Δυναμικού Περιεχομένου Διαδικτυακής Εφαρμογής
- [2] Παναγιώτης Φυταμπάνης, Πτυχιακή Εργασία «Σχεδίαση και Ανάπτυξη διαδικτυακού Περιβάλλοντος Ενημέρωσης, Παρακολούθησης και διαχείρισης δυναμικού Περιεχομένου διαδικτυακής Εφαρμογής», 2012 τμήμα Επιστήμης και Τεχνολογίας Τηλεπικοινωνιών του Πανεπιστημίου Πελοποννήσου
- [3] Νικόλαος Τσελίκας, Σημειώσεις στα πλαίσια του μαθήματος "Δικτυακές Υπηρεσίες & Ηλεκτρονικό Εμπόριο", για το μεταπτυχιακό πρόγραμμα του τμήματος Επιστήμης και Τεχνολογίας Τηλεπικοινωνιών του Πανεπιστημίου Πελοποννήσου
- [4] Luke Welling, Laura Thomson , Ανάπτυξη Web Εφαρμογών με PHP και MySQL (4η έκδοση), Εκδόσεις Μ. Γκιούρδας
- [5] W. Jason Gilmore , Beginning PHP and MySQL: From Novice to Professional (Expert's Voice in Web Development) , Apress
- [6] Robin Nixon , Learning PHP, MySQL, and JavaScript: A Step-By-Step Guide to Creating Dynamic Websites
- [7] Stefan Mischook, Learn Object Oriented Programming (OOP) in PHP , September 07 2007
- [8] <http://www.google.com>
- [9] <http://www.php.net>
- [10] <http://net.tutsplus.com>
- [11] <http://www.w3schools.com>
- [12] <http://httpd.apache.org>
- [13] <http://htmlparser.sourceforge.net>
- [14] <http://www.the-art-of-web.com>
- [15] <http://greetuts.net>
- [16] <http://www.wikipedia.com>
- [17] <http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-HTML.html>
- [18] <http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-CSS.html>
- [19] <http://studentguru.gr/w/tutorials/a0-javascript-javascript-html.aspx>
- [20] <http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-Php-Analytical.html>
- [21] <http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-MySQL.html>