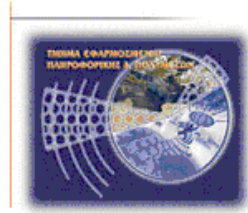




Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

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



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

με θέμα:

Ηλεκτρονικό Εμπόριο

**Σπουδαστές : Τσάφκα Βαλμίρα
Μουλματζέ Ια**

Επιβλέπων καθηγητής : Παπαδάκης Νικόλαος

Ηράκλειο - Δεκέμβριος 2014

Abstract

The goal of the dissertation is to understand basic techniques, technologies and means of implementation of an e-shop.

Why are planning to use the following languages:

1. HTML
2. CSS
3. XML
4. Javascript
5. PHP

Our final project should be an reliable online store selling products related to information technology. It should be integrated so that they could theoretically be carried out orders, but at the same time enables a user-administrator take action necessary for the proper functioning of such a shop.

Having now completed our thesis we should be able to understand the philosophy, the advantages and disadvantages of each, and the proper use of each of the aforementioned language.

Σύνοψη

Στόχος της πτυχιακής εργασίας είναι η κατανόηση βασικών τεχνικών, τεχνολογιών και μέσων υλοποίησης ενός ηλεκτρονικού ιστότοπου.

Αναλυτικότερα θα εξεταστούν και θα χρησιμοποιηθούν οι εξής τεχνολογίες:

1. HTML
2. CSS
3. XML
4. Javascript
5. PHP

Το τελικό μας αποτελέσματα θα πρέπει να είναι όσο πιο κοντά γίνεται σε ένα ηλεκτρονικό κατάστημα πωλήσεων προϊόντων, σχετιζόμενα με την πληροφορική. Θα πρέπει να είναι ολοκληρωμένο έτσι ώστε να μπορούν θεωρητικά να πραγματοποιηθούν παραγγελίες αλλά ταυτόχρονα να δίνει την δυνατότητα σε έναν χρήστη-διαχειριστή να προβαίνει σε ενέργειες απαραίτητες για την ομαλή λειτουργία ενός τέτοιου ηλεκτρονικού καταστήματος.

Έχοντας ολοκληρώσει πλέον την πτυχιακή μας εργασία θα πρέπει να είμαστε ικανοί να κατανοήσουμε την φιλοσοφία, τα πλεονεκτήματα και τα μειονεκτήματα κάθε, αλλά και την ορθή χρήση κάθε προαναφερόμενης γλώσσας.

Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε θερμά τον καθηγητή μας Κ.Παπαδάκη Νικόλαο κυρίως για την εμπιστοσύνη που μας έδειξε, και την υπομονή που έκανε κατά τη διάρκεια υλοποίησης της πτυχιακής εργασίας. Όπως επίσης και για την πολύτιμη βοήθεια και καθοδήγηση του, για την επίλυση διάφορων θεμάτων.

Θα θέλαμε επίσης να απευθύνουμε τις ευχαριστίες μας στους γονείς μας, οι οποίοι μας στήριξαν στις σπουδές μας με διάφορους τρόπους, φροντίζοντας για την καλύτερη δυνατή μόρφωση μας.

Πίνακας Περιεχομένων

Abstract	2
Σύνοψη	2
Εισαγωγή	6
<i>HTML</i>	7
<i>CSS</i>	11
<i>XML</i>	12
<i>JavaScript</i>	14
<i>jQuery</i>	14
<i>PHP</i>	15
<i>Χρήση HTML και CSS</i>	16
<i>Δομή ιστοσελίδας</i>	17
<i>Χρήση Javascript</i>	18
<i>Jquery</i>	18
<i>PHP</i>	18
<i>Ρόλος της PHP στην σελίδα</i>	18
<i>Πλεονεκτήματα χρήσης PHP στην σχεδίαση</i>	18
<i>Δυναμικότητα μέσα από την PHP</i>	19
<i>XML βάση</i>	20
<i>Δομή και σχεδιασμός Βάσης</i>	20
<i>Σύνδεση Βάσης</i>	22
<i>Πλεονεκτήματα και Μειονεκτήματα XML</i>	22
Δομή Ιστοσελίδας	23
<i>Αρχική σελίδα</i>	34
<i>Πλοήγηση Μη εγγεγραμμένου επισκέπτη</i>	35
<i>Εγγραφή πελάτη</i>	35
<i>Σύνδεση πελάτη</i>	36
<i>Πλοήγηση πελάτη</i>	36

<i>Σύνδεση Διαχειριστή</i>	39
<i>Πλοήγηση Διαχειριστή</i>	42
Παρουσίαση Ιστοσελίδας	49
<i>Αρχική σελίδα</i>	49
<i>Σύνδεση Πελάτη</i>	50
Αναζήτηση Προϊόντων	51
Καλάθι Αγορών	52
Προβολή Ιστορικό Παραγγελιών	53
<i>Σύνδεση Διαχειριστή</i>	54
Τροποποίηση Προϊόντος.....	55
Τροποποίηση Αποθεμάτων	56
Προβολή και τροποποίηση Παραγγελιών.....	57
Προβολή διαφόρων αναφορών	57
Αποτελέσματα	58
<i>Συμπεράσματα</i>	58
<i>Μελλοντικές επεκτάσεις</i>	59
Βιβλιογραφία	60

Πίνακας Εικόνων

Εικόνα 1: Online shopping	8
Εικόνα 2: HTML γλώσσα	9
Εικόνα 3: CSS γλώσσα	12
Εικόνα 4: XML Γλώσσα	13
Εικόνα 5: JavaScript logo	15
Εικόνα 6: JQuery logo	16
Εικόνα 7: PHP λειτουργία	16
Εικόνα 8: Αρχική σελίδα	51
Εικόνα 9: Φόρμα σύνδεσης Πελάτη/ Φόρμα εγγραφής πελάτη	52
Εικόνα 10: Φόρμα αναζήτησης προϊόντων	52
Εικόνα 11: Πλοήγηση κατηγοριών	53
Εικόνα 12: Καλάθι Αγορών	54
Εικόνα 13: Καλάθι αγορών μετά από αφαίρεση προϊόντος	54

Εικόνα 14: Προβολή Παραγγελιών Πελάτη	55
Εικόνα 15: Ακύρωση Παραγγελίας.....	55
Εικόνα 16: Λίστα επιλογής προϊόντων για τροποποίηση	56
Εικόνα 17: Τροποποίηση Προϊόντος.....	57
Εικόνα 18: Επεξεργασία αποθεμάτων.....	57
Εικόνα 19: Προβολή παραγγελιών από τον Admin	58
Εικόνα 20: Προβολή διαφόρων αναφορών	59

Πίνακας Πινάκων

Πίνακας 1: HTML ετικέτες	18
Πίνακας 2: CSS ιδιότητες	19
Πίνακας 3: jquery εντολές	20
Πίνακας 4: PHP εντολές.....	22

Εισαγωγή

Το ηλεκτρονικό κατάστημα είναι ένας διαδικτυακός τόπος μέσα από τον οποίο μπορούν να πραγματοποιηθούν πωλήσεις διαφόρων αγορών. Με την εισαγωγή της πληροφορικής στην καθημερινότητα μας υπήρξαν αρκετές αλλαγές τον τρόπο με τον οποίο ζούμε. Ένα από τα πράγματα που άλλαξαν ριζικά είναι και κομμάτι των εμπορικών επιχειρήσεων. Αλλαγές που μπορούν να αποτελέσουν ένα σημαντικό όπλο για τις επιχειρήσεις που επιθυμούν να ανταποκριθούν στην συνεχόμενη εξέλιξη, τον τεράστιο ανταγωνισμό και να πρωτοτυπήσουν. Μια επιχείρηση λαμβάνοντας τις σωστές στρατηγικές επιλογές στο σήμερα μπορούν να εξασφαλίσουν την επιβίωσή τους και αυτές οι επιλογές δεν μπορούν να απέχουν από την τεχνολογική πρόοδο που ζει η κοινωνία μας. Πλέον δεν αρκεί μια επιχείρηση να παρουσιάσει τον εαυτό της στο διαδίκτυο, αλλά υπάρχει και η άμεση ανάγκη να δώσει την δυνατότητα στον επισκέπτη να γίνει και πελάτης.

Το ηλεκτρονικό επιχειρείν (E-Business) αναφέρεται στην πραγματοποίηση επιχειρηματικών συναλλαγών μέσω του Internet και είναι η προσαρμογή του κλασικού επιχειρηματικού μοντέλου στην νέα ηλεκτρονική πραγματικότητα ή την ανάπτυξη νέου επιχειρηματικού μοντέλου με αντικείμενο μόνο το Διαδίκτυο.

Όπως χαρακτηριστικά έχει πει ο Μπιλ Γκέιτς: "Το Διαδίκτυο δεν είναι απλώς άλλο ένα κανάλι πωλήσεων. Θα μετασχηματίσει την επιχείρησή σας. Η μελλοντική επιχείρηση θα λειτουργεί με ένα ψηφιακό νευρικό σύστημα". Με λίγα λόγια: πωλήσεις δίχως όρια και περιορισμούς, ανεξάρτητα χρονικών και γεωγραφικών περιορισμών.

Έτσι, υπάρχει η συνεχής ανάπτυξη διεθνών ηλεκτρονικών καταστημάτων, που προσφέρουν ημερησίως μια πληθώρα προϊόντων σε αρκετά χαμηλότερες τιμές. Ανάλογα με τα είδη που υπάρχουν σε διαθεσιμότητα, ο επισκέπτης-πελάτης μπορεί να αναζητήσει και να επιλέξει το προϊόν που επιθυμεί, να μάθει την τιμή και τον χρόνο αποστολής (εάν το παραγγείλει), να δει εικόνες και παρουσιάσεις του προϊόντος, να διαβάσει κριτικές και χαρακτηριστικά αλλά και να κάνει μια πολύ πιο εύκολη και γρήγορη έρευνα αγορών. Λόγο της μη ύπαρξης κάποιων πάγιων εξόδων που έχουν τα συνηθισμένα καταστήματα, όπως αυτό του ενοικίου ή των υπαλλήλων καθώς απασχολεί πολύ λιγότερους από το συνηθισμένο αριθμό, οι τιμές που μπορούν να προσφέρουν τα ηλεκτρονικά καταστήματα είναι χαμηλότερες. Είναι ένας πωλητής που μπορεί και δουλεύει 24 ώρες την ημέρα για 365 ημέρες τον χρόνο. Οι συναλλαγές είναι πάντα ολοκληρωμένες αφού πραγματοποιούνται πλέον με μεγάλη ασφάλεια μέσα από υπηρεσίες που προσφέρουν οι τράπεζες αλλά και υπηρεσίες όπως το PayPal.



Εικόνα 1: Online shopping

HTML

Η HTML (HyperText Markup Language) είναι η βασική γλώσσα σήμανσης κειμένου για τις ιστοσελίδες.

Η HTML αποτελείται από ετικέτες (tags) που περικλείονται μέσα σε σύμβολα , μια ετικέτα(tag) αρχίζει με ένα σύμβολο '<' και τελειώνει με '>' (για παράδειγμα <p>). Οι ετικέτες αυτές συνήθως αποτελούνται από δύο τμήματα (για παράδειγμα <h2> - ετικέτα αρχής - και </h2> - ετικέτα τέλους). Ανάμεσα στις ετικέτες(tags) βρίσκεται το κείμενο(όπου υπάρχουν ειδικά tags για την μορφοποίηση κείμενου), οι πίνακες, εισαγωγή εικόνων, ήχων, δημιουργία συνδέσμων(links) με τους οποίους μπορούμε να μεταβαίνουμε αναμεσα στις σελίδες μας κλπ.

Ένα πρόγραμμα πλοήγησης διαβάζει/μεταφράζει τις ετικέτες(tags) από τα έγγραφα HTML και κατασκευάζει κατά αυτόν τον τρόπο τις σελίδες. Οι πλοηγητές δεν εμφανίζουν τις ετικέτες της HTML, αλλά τις χρησιμοποιούν για να δημιουργήσουν το περιεχόμενο της κάθε σελίδα.

Μέσω της HTML πραγματοποιείτε η ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα.

Παρακάτω μπορούμε να δούμε ένα παράδειγμα HTML εγγράφου το οποίο είναι το κλασικό παράδειγμα HELLO WORLD.



Εικόνα 2: HTML γλώσσα

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello HTML</title>
  </head>
  <body>
    <p>Hello world</p>
  </body>
</html>
```

(Το<html> και το </html> περιγράφει την ιστοσελίδα, ενώ το <body> και το περιεχόμενο του είναι το ορατό μέρος της. Το σημασμένο κείμενο '<title>Hello HTML</title>' καθορίζει τον τίτλο που θα εμφανίζεται στην μπάρα τίτλου του browser.)

Τα έγγραφα HTML αποτελούνται από στοιχεία με την γενική μορφή τους να έχουν τρία συστατικά: ένα ζεύγος από ετικέτες, την «ετικέτα εκκίνησης» και την «ετικέτα τερματισμού», μερικές ιδιότητες που ορίζουν το στοιχείο (μέσα στην ετικέτα εκκίνησης) και το κείμενο ή γενικότερα το αντικείμενο περιεχόμενο μεταξύ των ετικετών, το οποίο μπορεί να περιλαμβάνει και άλλα στοιχεία εμφωλευμένα μέσα του. Το στοιχείο HTML μπορεί να είναι οτιδήποτε ανάμεσα στις ετικέτες εκκίνησης και τερματισμού. Τέλος, κάθε ετικέτα περικλείεται σε σύμβολα «μεγαλύτερο από'<'» και «μικρότερο από'>'», δηλαδή < και >.

```
<tag attribute1="value1" attribute2="value2">content</tag>
```

Αρκετά στοιχεία HTML αναφέρονται ως άδεια στοιχεία και έχουν τη μορφή `<tag attribute1="value1" attribute2="value2" >`, δεν έχουν καθόλου περιεχόμενο. Το όνομα κάθε στοιχείου HTML είναι το ίδιο όνομα που χρησιμοποιείται στις αντίστοιχες ετικέτες. Το όνομα της ετικέτας τερματισμού ξεκινά με μια κάθετο `</>`, η οποία παραλείπεται στα άδεια στοιχεία. Τέλος, αν δεν αναφέρονται ρητά οι ιδιότητες ενός στοιχείου, τότε χρησιμοποιούνται οι προεπιλογές σε κάθε περίπτωση.

Παραδείγματα στοιχείων

Κεφαλίδα του εγγράφου HTML: `<head>...</head>`. Συνήθως περιέχει τον τίτλο, π.χ:

```
<head>
  <title>The title</title>
</head>
```

Επικεφαλίδες: οι επικεφαλίδες στην HTML ορίζονται με τις ετικέτες `<h1>` έως `<h6>`:

```
<h1>Επικεφαλίδα1</h1>
<h2>Επικεφαλίδα2</h2>
<h3>Επικεφαλίδα3</h3>
<h4>Επικεφαλίδα4</h4>
<h5>Επικεφαλίδα5</h5>
<h6>Επικεφαλίδα6</h6>
```

Παράγραφοι:

```
<p>Paragraph 1</p> <p>Paragraph 2</p>
```

Αλλαγή γραμμής: `
`. Η διαφορά ανάμεσα στο `
` και το `<p>` είναι ότι το `
` αλλάζει γραμμή χωρίς να αλλάζει την σημαντική δομή της σελίδας, ενώ το `<p>` τεμαχίζει τη σελίδα σε παραγράφους. Το `
` είναι ένα άδαιο στοιχείο, δηλαδή δεν έχει περιεχόμενο, ούτε και χρειάζεται ετικέτα τερματισμού.

```
<p>This <br> is a paragraph <br> with <br> line breaks</p>
```

Σχόλια:

```
<!-- This is a comment -->
```

Τα σχόλια μπορούν να βοηθήσουν στην κατανόηση της σήμανσης. Δεν εμφανίζονται στην ιστοσελίδα.

Για παράδειγμα, το `<h2>Golf</h2>` σημαίνει ότι η λέξη «Golf» θα είναι μια επικεφαλίδα δεύτερου επιπέδου. Η δομική σήμανση δεν εμπεριέχει συγκεκριμένο τρόπο εμφάνισης, αλλά οι περισσότεροι browser έχουν δικό τους προκαθορισμένο τρόπο για τη μορφοποίηση όλων των στοιχείων. Το περιεχόμενο μπορεί να μορφοποιηθεί οπτικά με τη χρήση των CSS.

Η σήμανση οπτικής μορφοποίησης περιγράφει την εμφάνιση του κειμένου, άσχετα από τον σκοπό του.

Για παράδειγμα το `έντονο κείμενο` υποδεικνύει ότι οι συσκευές που θα εμφανίσουν το κείμενο θα το κάνουν έντονο, αλλά δεν λέει τι θα κάνουν με το κείμενο οι συσκευές που δεν μπορούν να το εμφανίσουν, όπως για παράδειγμα οι συσκευές φωνητικής ανάγνωσης σελίδων. Τόσο στην περίπτωση του `έντονο` όσο και του `<i>πλάγιο</i>`, υπάρχουν άλλα στοιχεία τα οποία μπορεί οπτικά να εμφανίζονται ίδια, αλλά είναι πιο σημαντικά, όπως τα `σημαντικό κείμενο` και `κείμενο με έμφαση` αντίστοιχα. Είναι ευκολότερα να ερμηνεύσει ένας φωνητικός browser τα τελευταία δύο στοιχεία. Ωστόσο, δεν είναι ισότιμα με τα αντίστοιχα στοιχεία οπτικής μορφοποίησης. Για παράδειγμα ένα πρόγραμμα φωνητικής ανάγνωσης της σελίδας δεν θα πρέπει να προφέρει με έμφαση τον τίτλο ενός βιβλίου, αλλά όταν το ίδιο περιεχόμενο εμφανίζεται σε μια οθόνη τότε ο τίτλος θα είναι με πλάγια γράμματα. Τα περισσότερα στοιχεία οπτικής μορφοποίησης έχουν θεωρούνται ξεπερασμένα μετά την προδιαγραφή HTML 4.0, και έχουν αντικατασταθεί από τα CSS.

Η σήμανση υπερκειμένου κάνει μερικά τμήματα ενός εγγράφου να συνδέουν με άλλα έγγραφα.

Τα στοιχεία anchor δημιουργεί έναν υπερσύνδεσμο στο έγγραφο, και η ιδιότητα href ορίζει τον στόχο του συνδέσμου. Για παράδειγμα, η σήμανση HTML `Βικιπαίδεια` εμφανίζει το κείμενο «Wikipedia» ως υπερσύνδεσμο. Για την εμφάνιση μιας εικόνας ως συνδέσμου, μπορεί να εισαχθεί ένα στοιχείο `` ως περιεχόμενο του στοιχείου «a». Όπως και το `
`, το `` είναι άδαιο στοιχείο. Έχει ιδιότητες

αλλά δεν έχει περιεχόμενο, ούτε ετικέτα τερματισμού: ``.

Οι περισσότερες ιδιότητες των στοιχείων είναι ζεύγη ονομάτων και τιμών, τα οποία διαχωρίζονται με ένα «=» και γράφονται μέσα στην ετικέτα εκκίνησης ενός στοιχείου, μετά το όνομα του στοιχείου. Η τιμή μπορεί να περικλείεται σε μονά ή διπλά εισαγωγικά, παρότι τιμές που αποτελούνται από συγκεκριμένους χαρακτήρες μπορούν να γράφονται χωρίς εισαγωγικά στην HTML, αλλά όχι στην XHTML. Το να μένουν οι τιμές των ιδιοτήτων χωρίς εισαγωγικά θεωρείται ανασφαλές. Εκτός από τις ιδιότητες που γράφονται ως ζεύγη ονομάτων και τιμών, υπάρχουν και μερικές οι οποίες επηρεάζουν το στοιχείο απλά με την παρουσία τους μέσα στην ετικέτα εκκίνησης, όπως η ιδιότητα `ismap` του στοιχείου `img`.

Υπάρχουν και μερικές κοινές ιδιότητες οι οποίες εμφανίζονται σε πολλά στοιχεία:

Η ιδιότητα `id` παρέχει ένα αναγνωριστικό για ένα στοιχείο το οποίο είναι μοναδικό σε ολόκληρο το έγγραφο. Χρησιμοποιείται για να ταυτοποιεί το στοιχείο ώστε τα CSS να μπορούν να αλλάξουν τον τρόπο που αυτό εμφανίζεται, καθώς και τα σενάρια μπορούν να αλλάξουν, να μετακινήσουν ή να διαγράψουν τα περιεχόμενα ή την εμφάνισή του. Ακόμα, αν ένα `id` προστεθεί στο URL μιας σελίδας, παρέχει ένα μοναδικό αναγνωριστικό για ένα τμήμα της σελίδας, για παράδειγμα: `http://en.wikipedia.org/wiki/HTML#Attributes`

Η ιδιότητα `class` παρέχει τη δυνατότητα ταξινόμησης παρόμοιων αντικειμένων στην ίδια κλάση. Μπορεί να χρησιμοποιηθεί για να αποδώσει κάποια σημασία στο στοιχείο, ή για σκοπούς εμφάνισης. Για παράδειγμα, ένα έγγραφο HTML μπορεί να χρησιμοποιεί την επισήμανση `class="notation"` σε μερικά στοιχεία για να ξεχωρίσει από το υπόλοιπο κείμενο του εγγράφου. Κατά την εμφάνιση του εγγράφου, αυτά τα στοιχεία μπορεί -για παράδειγμα- να εμφανίζονται όλα μαζί στο τέλος της σελίδας ως υποσημειώσεις, άσχετα με την θέση που εμφανίζονται μέσα στον κώδικα. Επίσης οι ιδιότητες `class` χρησιμοποιούνται σημασιολογικά στα `microformat`. Ένα στοιχείο μπορεί να έχει πολλαπλές κλάσεις, για παράδειγμα το `class="notation important"` βάζει το στοιχείο τόσο στην κλάση «`notation`» όσο και στην «`important`».

Η ιδιότητα `style` εφαρμόζει στυλ εμφάνισης σε συγκεκριμένα στοιχεία. Θεωρείται καλύτερη τακτική να χρησιμοποιούνται οι ιδιότητες `id` ή `class` ώστε να επιλέγεται το στοιχείο μέσα σε ένα CSS, αλλά μερικές φορές μπορεί να είναι πιο απλό να ανατεθούν `style` κατευθείαν στο στοιχείο.

Η ιδιότητα `title` προσθέτει μια εξήγηση στο στοιχείο στο οποίο εφαρμόζεται. Στους περισσότερους browser αυτή η ιδιότητα εμφανίζεται ως αναδυόμενο παράθυρο βοήθειας.

Η ιδιότητα `lang` ταυτοποιεί την φυσική γλώσσα των περιεχομένων του στοιχείου, η οποία μπορεί να είναι διαφορετική από το υπόλοιπο έγγραφο. Για παράδειγμα, μέσα σε ένα αγγλικό έγγραφο μπορεί να υπάρχει το:

```
<p>Oh well, <span lang="fr">c'est la vie</span>, as they say in France.</p>
```

Το στοιχείο `abbr` μπορεί να χρησιμοποιηθεί για να επιδειξει μερικές από τις πιο πάνω ιδιότητες:

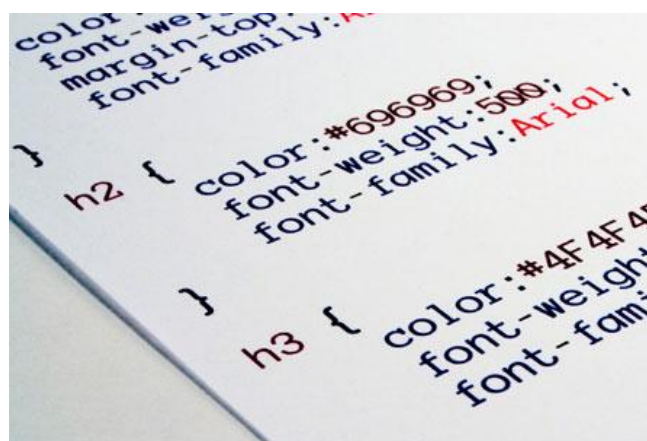
```
<abbr id="anId" class="jargon" style="color:purple;" title="Hypertext Markup Language">HTML</abbr>
```

. Αυτό το παράδειγμα εμφανίζεται ως: HTML. Στους περισσότερους browser, όταν σταθεί ο δείκτης του ποντικιού πάνω από αυτό το στοιχείο, τότε θα εμφανιστεί ένα αναδυόμενο παράθυρο με το περιεχόμενο «Hypertext Markup Language».

Τα περισσότερα στοιχεία μπορούν επίσης να δεκτούν την ιδιότητα `dir` η οποία καθορίζει την κατεύθυνση του κειμένου. Η τιμή «`rtl`» της ιδιότητας, δείχνει κείμενο που διαβάζεται από δεξιά προς τα αριστερά, όπως για παράδειγμα στα Αραβικά, στα Περσικά ή στα Εβραϊκά.

CSS

Η CSS (Cascading Style Sheets) είναι μια απλή γλώσσα υπολογιστή, η οποία μας βοηθάει να ορίσουμε με ακρίβεια την εμφάνιση και τη μορφοποίηση ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης, όπως η HTML. Η CSS καλείται να προσδιορίσει την εμφάνιση μιας ιστοσελίδας δηλαδή μπορεί να χρησιμοποιηθεί για να διαμορφώνει τα χαρακτηριστικά, τα χρώματα



Εικόνα 3: CSS γλώσσα

και το μέγεθος διαφόρων στοιχείων, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την html. Για την σωστή σχεδίαση και δόμηση μιας ιστοσελίδας θεωρείται απαραίτητη η χρήση της CSS.

Για μια σελίδα html πολύ πιθανόν να υπάρχουν περισσότερα από ένα css τα οποία είναι υπεύθυνα για την εμφάνισή της. Ένα πλεονέκτημα που έχει η χρήση του css είναι οτι αν κάνουμε μια αλλαγή σε ένα έγγραφο αυτό αυτόματα εφαρμόζεται και στις υπόλοιπες σελίδες που το χρησιμοποιούν.

Έτσι μπορούμε εύκολα να αλλάξουμε την εμφάνιση μιας ιστοσελίδας.

Το css(φύλλο στύλ) μπορεί να το δημιουργεί ο συγγραφέας της ιστοσελίδας, ο χρήστης του διαχειριστή και τέλος ο ίδιος ο διαχειριστής. Άρα θα υπάρχουν περισσότερες από μια δηλώσεις που θα αφορούν ένα συγκεκριμένο στοιχείο της ιστοσελίδας. Έτσι στη css έχει καθοριστεί μια σειρά στη οποία θα μπουν αυτές οι δηλώσεις και ποιά θα εκτελείται πρώτα.

Παρακάτω θα δείτε τον αλγόριθμο με τον οποίο θα επιτευχθεί αυτή η σειρά:

- Βρίσκεται πρώτα τις δηλώσεις που έχουν γίνει στο συγκεκριμένο στοιχείο. Αν λοιπόν το επιλέξετε οι δηλώσεις αυτές θα εφαρμοστούν στο στοιχείο.
- Ταξινομείστε με βάση τη σημασία (σημαντικά ή κανονικά) και από που προέρχεται(συγγραφέας, χρήστης, διαχειριστής χρήστη), κατά αύξουσα σειρά προτεραιότητας:
 1. Τις δηλώσεις του διαχειριστή χρήστη
 2. Τις κανονικές δηλώσεις του χρήστη
 3. Τις κανονικές δηλώσεις συγγραφέα
 4. Τις σημαντικές δηλώσεις του συγγραφέα
 5. Τις σημαντικές δηλώσεις του χρήστη
- Ταξινομείστε τις δηλώσεις που έχουν την ίδια σημασία και την ίδια προέλευση ανάλογα με τη εξειδίκευση είναι αυτός που το επιλέγει.
- Ταξινομείστε ανάλογα με τη σειρά που καθορίστηκαν δηλαδή αν δυο δηλώσεις έχουν ίδια προέλευση και εξειδίκευση, θα επικρατήσει αυτή που προσδιορίστηκε τελευταία.

Με βάση τα παραπάνω, έχουμε ως συμπέρασμα ότι θα επιλεγεί η δήλωση που θα είναι τελευταία στη σειρά αφού εφαρμοστεί μια σειρά από κανόνες που αφορούν το συγκεκριμένο στοιχείο.

XML

Η XML (Extensible Markup Language:Εκτεταμένη Γλώσσα Σήμανσης) είναι μία επεκτάσιμη γλώσσα σήμανσης(όχι γλώσσα προγραμματισμού), που την χρησιμοποιούμε για την περιγραφή μιας σημειακής γλώσσας, όπως είναι η HTML.Αναπαριστά δηλαδή την έννοια των δεδομένων. Για να δημιουργήσουμε ένα XML έγγραφο αρκεί ένας απλός κειμενογράφος π.χ Notepad ή ένα πρόγραμμα με XML επεξεργαστή, και να έχει την επέκταση .xml. Τα αρχεία xml μπορούν να ενσωματωθούν σε HTML με τη χρήση προθεμάτων xml. Η πρώτη εντολή για να δημιουργήσω ένα xml έγγραφο είναι <?xml version="1.0"?>, έτσι καταλαβαίνει το πρόγραμμα ότι ακολουθεί ένα xml έγγραφο και τι έκδοση είναι. Οι χρήστες μπορούν να καθορίσουν τις ετικέτες και τις σχέσεις αυτών στην xml.Οι ετικέτες ορίζουν το νόημα και τη δομή των δεδομένων , δηλαδή να είναι δομημένα, π.χ <onoma> andy</onoma> έχει και ετικέτα αρχής και ετικέτα τέλους.



Την xml την χρησιμοποιούμε για την απλή και εύχρηστη αποθήκευση και μετάδοση δεδομένων και πληροφοριών. Οι εφαρμογές της είναι σαν βάσεις δεδομένων περισσότερο. Με την xml περιγράφουμε οποιαδήποτε πληροφορία . Γενικά μοιάζει με την html γιατί διατηρεί την δύναμη και την ευελιξία της αλλά είναι πιο ευκολόχρηστη σαν γλώσσα.

Μέσα σε μια PHP σελίδα μπορούμε να φορτώσουμε ένα αρχείο xml με την εντολή
\$xml=Simplexml_load_file("test.xml"); και να προσπελάσουμε ένα πεδίο της
\$xml->colors->fontcolor;

Την XML 1.0 την δημιούργησε ο διεθνής οργανισμός προτύπων W3C (World Wide Web Consortium). Ένα κείμενο XML είναι μία ακολουθία χαρακτήρων. Σχεδόν κάθε χαρακτήρας Unicode μπορεί να εμφανίζεται σε ένα κείμενο XML.

Γενικά ή XML δίνει ελευθερία στην γραφή των δεδομένων γιατί τα έχει έτσι καλά δομημένα που δεν είναι απαραίτητο να διαβάζεται μόνο από προγραμματιστές αλλά και ένα απλός άνθρωπος μπορεί να το διαβάσει, οργανώνει τα δεδομένα πολύ εύκολα, σου παρέχει τη δυνατότητα εναλλακτικής παρουσίασης των δεδομένων, είναι ανεξάρτητη από υλικό και λογισμικό, είναι αναγνώσιμη από χρήστες και διαρκώς αναπτυσσόμενη και διατίθεται δωρεάν.

Παράδειγμα ενός xml εγγράφου:

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

```
<?xml version="1.0"?>
<painting>
  
  <caption>mia poly foto by photografer
  <date>1321</date>.</caption>
</painting>
```

Υπάρχουν πέντε στοιχεία σε αυτό το κείμενο του παραδείγματος: τα painting, img, caption, και ένα date. Το στοιχείο date, είναι παιδί του στοιχείου caption, το οποίο είναι παιδί του στοιχείου-ρίζας painting.

JavaScript

Η JavaScript (JS) είναι μια γλώσσα προγραμματισμού (scripting) για ηλεκτρονικούς υπολογιστές. Αυτή η γλώσσα είναι client side (δηλαδή από την πλευρά του πελάτη) που σημαίνει ότι το αποτέλεσμα εκτελείται και παράγεται στον φυλλομετρητή του πελάτη. Με αποτέλεσμα να μπορούν να επικοινωνούν, να ανταλλάσσουν δεδομένα με τον χρήστη και επιπλέον να αλλάζουν το περιεχόμενο ενός εγγράφου.



JavaScript

Εικόνα 5: JavaScript logo

Η JavaScript δεν έχει σχέση με την java παρόλο που αντιγράφει ονοματα και συμβάσεις απο αυτήν, λειτουργεί δηλαδή σαν συμπληρωματική σε μια ιστοσελίδα και όχι στη δημιουργία της εξ'ολοκλήρου. Βασίζεται στα πρότυπα (prototype-based), κάνει κάποια πράγματα δυναμικά σε μια ιστοσελίδα, έχει αδύναμους τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξή της είναι επηρεασμένη από τη C, ενώ είναι αντικειμενοστραφή γλώσσα.

Η JavaScript σχεδιάστηκε με τις βασικές αρχές από τις γλώσσες προγραμματισμού Self και Scheme. Συνεπάγεται ότι είναι μια γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), που υποστηρίζει αντικειμενοστρεφή, προστακτική και συναρτήσεις προγραμματισμού.

Η JavaScript χρησιμοποιείται και σε έγγραφα PDF, σε εξειδικευμένους φυλλομετρητές και σε μικρές εφαρμογές της επιφάνειας εργασίας. Η χρήση της JavaScript στη πλευρά του διακομιστή (server-side) εμφανίζεται με το Node.js όπου είναι ένα μοντέλο προγραμματισμού που βασίζεται στα events (γεγονότα). Έτσι γίνεται πιο δημοφιλής στην ανάπτυξη εφαρμογών.

Το πρότυπο της γλώσσας ονομάζεται ECMAScript με βάση τον οργανισμό τυποποίησης ECMA.

JQuery

Η jQuery είναι μια βιβλιοθήκη JavaScript σχεδιασμένη να απλοποιήσει τη υλοποίηση σεναρίων (scripting) στη πλευρά του πελάτη (client-side) της HTML και είναι συμβατή με πολλαπλούς φυλλομετρητές Ιστού (browsers). Κυκλοφόρησε τον Ιανουάριο του 2006 από τον Τζον Ρέριγκ (John Resig). Έχει γίνει πολύ δημοφιλής αφού χρησιμοποιείται σε πάνω από το 65% των 10.000 ιστοτόπων με τη μεγαλύτερη επισκεψιμότητα και επίσης η google την χρησιμοποιεί στην αναζήτησή της στο youtube, adsense κτλ .



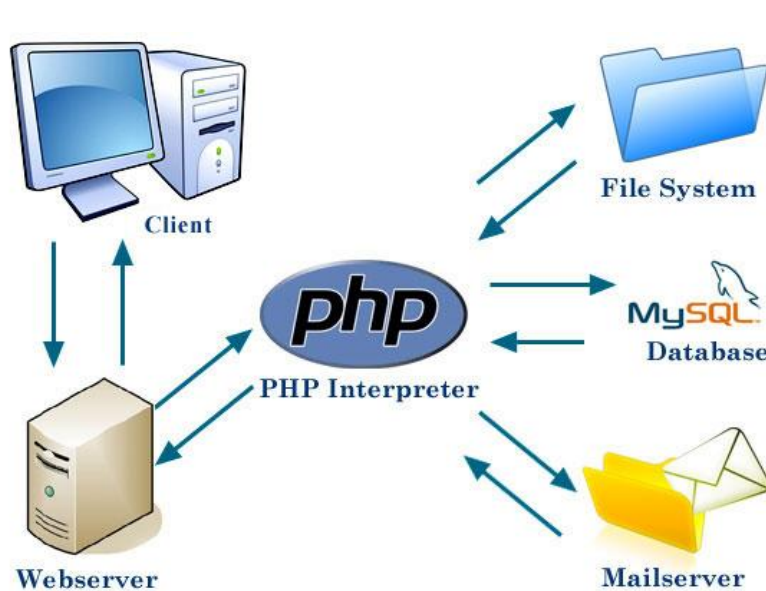
Εικόνα 6: JQuery logo

η jQuery είναι ελεύθερο λογισμικό, με άδεια MIT και είναι ένα πολύ δυνατό εργαλείο, που επιλέγουν οι περισσότερες ιστοσελίδες γιατί μας επιτρέπει να κάνουμε πολλά σύγχρονα πράγματα όταν κατασκευάζουμε

μια ιστοσελίδα. Είναι απαραίτητη για να πραγματοποιήσουμε μια ευχάριστη και χρήσιμη ιστοσελίδα. Έχει συμβατότητα σε όλους τους browsers. Σε μια ιστοσελίδα με τη χρήση της jQuery μπορούμε να δημιουργήσουμε κίνηση, animation, αλλάξουμε ιδιότητες html, ajax requests κτλ.

PHP

Η PHP (Hypertext PreProcessor) είναι μια γλώσσα προγραμματισμού ανοιχτού κώδικα (ελεύθερο να το χρησιμοποιήσει οποιοσδήποτε) που χρησιμοποιείται στην δημιουργία ψηφιακών δυναμικών σελίδων. Όταν αποστέλουμε μια σελίδα PHP περνάει πρώτα από έναν διακομιστή (π.χ. Apache) για επεξεργασία, ώστε το τελικό περιεχόμενο να παραχθεί σε πραγματικό χρόνο, ενσωματώνεται μέσα στον κώδικα της html και εκτελείται στον server.



Εικόνα 7: PHP λειτουργία

Ένα αρχείο που έχει κώδικα PHP έχει την επέκταση *.php αλλά υπάρχουν και άλλες εκτάσεις όπως *.php4, *.phtml.

Ένας server πρέπει να είναι ρυθμισμένος για να μπορεί να επεξεργάζεται και να μεταγλωτίζει τον php κώδικα που θα περιέχουν τα php αρχεία. Η php εκτελείται μέσω ενός ειδικού προγράμματος που είναι εγκατεστημένο στον server, τον apache.

Κάποια χαρακτηριστικά της php είναι: ανοιχτού

κώδικα, λειτουργεί σε περιβάλλον LINUX, εξελίσσεται συνέχεια, έχει πολλές έτοιμες συναρτήσεις που μας βοηθάνε και προγραμματίζουμε εύκολα με αυτήν. Οι ετικέτες ανοίγματος είναι <?php και οι ετικέτες κλεισίματος ?>. Μέσα σε αυτές τις ετικέτες γράφουμε εντολές (για παράδειγμα echo "helloo");, με το ελληνικό ερωτηματικό τερματίζεται μια εντολή στην php.

Μεθοδολογία Υλοποίησης

Χρήση HTML και CSS

Η html πλέον δίνει πολλές περισσότερες δυνατότητες στον προγραμματιστή. Δεν είναι ένα μόνο μέσω διαμόρφωσης της σελίδας αλλά κάτι πολύ παραπάνω, καθώς η html5 αυξάνει της λειτουργίες της με τέτοιο τρόπο που μας απαλλάσσει από περιττό κώδικα.

Μεγαλύτερο και δυνατότερο παράδειγμα είναι η χρήση του required μέσα στο input αλλά και ο προσδιορισμός του input πεδίου(text, number, email και τα λοιπά) έτσι ώστε πριν γίνει submit η φόρμα όχι μόνο να δει εάν έχουν δοθεί τιμές στα πεδία, αλλά να επαληθεύσει ότι και οι τιμές είναι στα πρότυπα που ζητάμε.

html	Δήλωση html αρχείου
head	Δήλωση head στοιχείου
body	Δήλωση body στοιχείου
p	Παράγραφος
table	Πίνακας
tr	Γραμμή πίνακα
td	Στήλη πίνακα
h1,h2,h3,h4	Επικεφαλίδες
span	Ομαδοποίηση κειμένου
div	Ομαδοποίηση στοιχείων που υπάγονται σε αυτό

a	Σύνδεσμος
script	Κώδικας javascript
form	Φόρμα
input	Input φόρμας
br	Αλλαγή σειράς
img	Εικόνα
b	Έντονα γράμματα
label	Ετικέτα input φόρμας

Πίνακας 1: HTML ετικέτες

padding
Margin
width
height
float
border
background
font-size
font-weight
line-height
display
color
border-radius
box-shadow
position
min-height
text-align
text-decoration

Πίνακας 2: CSS ιδιότητες

Δομή ιστοσελίδας

Η σελίδα μας βασίζεται μόνο στην αρχική της. Οι παράμετροι που δίνονται είναι αυτοί που ορίζουν και τα στοιχεία που θα χρησιμοποιηθούν για τον σχεδιασμό της αρχικής σελίδας. Το index.php αναζητά σε κάθε φόρτωση εάν υπάρχουν μεταβλητές και ανάλογα εάν υπάρχουν ή όχι βλέπει ποια μικρότερα στοιχεία πρέπει να εισάγει αλλά και ποια πρέπει να αφαιρέσει. Αυτό γίνεται με την χρήση της PHP και την εισαγωγή php αρχείων μέσα από την εντολή include

Χρήση Javascript

Η javascript που χρησιμοποιείται είναι αυτή που μας βοηθάει να βελτιώσουμε την εμφάνιση της σελίδας. Μέσα στο site γίνεται χρήση της jquery βιβλιοθήκης την οποία και θα τα αναλύσουμε παρακάτω. Η javascript είναι ένα βασικό κομμάτι των σημερινών σελίδων αλλά η χρήση τους πρέπει να είναι περιορισμένη σε κάθε σελίδα. Δεν θεωρείται από τους καλύτερους τρόπους να αναπτύξει μια σελίδα καθώς μπορεί να καθορίσει και μάλιστα αρνητικά την ταχύτητα ενός site. Υπάρχουν φυσικά τρόποι να χρησιμοποιηθεί σωστά αυτή η γλώσσα έτσι ώστε να επιρρεάσει όσο το δυνατόν λιγότερο. Τα καλά κομμάτι κώδικα javascript δεν επιρρεάζουν σε πρώτο χρόνο την εμφάνιση και την φόρτωση της σελίδας να υλοποιούνται στο τέλος κάθε σελίδας, διότι κατά την φόρτωση ενός κώδικα javascript ο πλοηγτής μας σταματάει να ζητάει παραπάνω αρχεία.

Jquery

Όπως προαναφέραμε για την υλοποίηση της σελίδας μας κάναμε χρήση της jquery βιβλιοθήκης. Η jquery είναι μια εξέλιξη της javascript μέσα από την οποία οι δυνατότητες που υπάρχουν πλέον στον σχεδιασμό των σελίδων γίνονται ολοένα και περισσότερες. Μέσα από τις βιβλιοθήκες που υπάρχουν μπορούν επιτευχθούν πολλά πράγματα όπως είναι το lightbox, animation μέσα στην σελίδα αλλά και διαμόρφωση της σελίδας με πολύ πιο εντυπωσιακό τρόπο από ότι θα γινόταν με την απλή javascript.

datepicker	Συνάρτηση για προβολή επιλογής ημερομηνίας
colorbox	Αντίστοιχο lightbox
click	Έλεγχος γεγονότος click πάνω σε στοιχείο
toggle	Αλλαγή κατάσταση εμφάνισης

Πίνακας 3: jquery εντολές

PHP

Ρόλος της PHP στην σελίδα

Η PHP είναι ένα από τα καλύτερα εργαλεία που μπορεί να έχει στα χέρια του κάποιος που θέλει να φτιάξει μια δυναμική σελίδα. Είναι ένα μέσω να δίνει την δυνατότητα στον χρήστη να επιλέγει ενέργειες τέτοιες ώστε να αλλάζει η δομή του site. Η php μπορεί να μετατρέψει ένα απλό site με μια σύνθετη και πολύπλοκη εφαρμογή. Μπορεί να συνδεθεί με μια βάση δεδομένων, να αλλάξει στοιχεία, να στείλει δεδομένα στον server και γενικότερα να μπορέσει να κάνει πράγματα που από μόνη της μια σελίδα δεν θα μπορούσε να κάνει.

Πλεονεκτήματα χρήσης PHP στην σχεδίαση

Η PHP είναι open source, πράγμα που σημαίνει ότι ο καθένας έχει ελεύθερη πρόσβαση στον πηγαίο κώδικα και μπορεί να τον χρησιμοποιήσει για τους δικούς τους σκοπούς ανάπτυξης. Ένα βασικό πλεονέκτημα του μοντέλου ανοιχτού κώδικα είναι ότι μπορεί να αποφευχθεί το κλείδωμα από τους προμηθευτές.

Η PHP επιτρέπει τη γρήγορη υλοποίηση σύνθετων λύσεων. Το όφελος; Εφαρμογές και σελίδες μπορούν να μπαίνουν μέσα στην αγορά σε πολύ πιο σύντομο χρονικό διάστημα, το κόστος παραγωγής αυτόματα μειώνεται καθώς μειώνεται και ο χρόνος κατασκευής.

Ένα από τα μεγαλύτερα πλεονεκτήματα της PHP είναι η συμβατότητα της σε σχεδόν κάθε πλατφόρμα. Η PHP είναι ένας κώδικας που εκτελείται από την μεριά του server οπότε ο χρήστης δεν έχει την δυνατότητα να δει και

να επέμβει στον κώδικα, αποτέλεσμα αυτού είναι η μεγαλύτερη ασφάλεια των συστημάτων που χρησιμοποιούν την PHP. Επιπλέον, επειδή η PHP συνεργάζεται άψογα με άλλες τεχνολογίες (π.χ. Java), μπορούμε να επαναχρησιμοποιεί τρέχοντα στοιχεία του λογισμικού σας. Αυτό είναι ένα σημαντικό πλεονέκτημα, καθώς δεν απαιτείται εκ νέου ανάπτυξη για το υπάρχον λογισμικό.

Προσφέρει επίσης μεγάλη ευελιξία κατά τη διάρκεια και μετά το αρχικό σχέδιο. Αυτό είναι σημαντικό, δεδομένου ότι η λειτουργικότητα αλλάζει συχνά κατά τη διάρκεια της ζωής του έργου. Ένα μεγάλο πράγμα για την PHP είναι ότι μπορούμε να υλοποιήσουμε τις αλλαγές, ακόμη και μετά την έναρξη της ανάπτυξης, χωρίς να χάσει πολύτιμο χρόνο.

Δυναμικότητα μέσα από την PHP

Μέσα από την PHP μας δίνεται η δυνατότητα να αναπτύξουμε ένα αξιόπιστο σύστημα μεγάλης ασφάλειας, που παράγει δυναμικά το περιεχόμενο της σελίδας. Μπορεί να δημιουργούνται σελίδες νέες χωρίς την επέμβαση του σχεδιαστή. Μέσα από την αλλαγή της βάσης δεδομένων που έχει το site μπορεί να αναπτυχθεί ένα ολόκληρο site.

Θέτοντας αρχικούς κανόνες και μορφές (templates) για την εμφάνιση του site, αυτό όσο μεγαλώνει η βάση μπορεί να μεγαλώνει χωρίς να αυξάνονται τα αρχεία που απαιτούνται για την εμφάνιση του.

Παρακάτω μπορούμε να δούμε τις βασικές εντολές της PHP που γίνονται χρήση μέσα στην σελίδα μας.

include	Εισαγωγή εξωτερικού αρχείου
echo	Τύπωση στο αρχείο
foreach	Επανάληψη
if	Έλεγχος συνθήκης
simplexml_load_file	Φόρτωση XML αρχείου
addChild	Προσθήκη πεδίου σε xml πεδίο
fopen	Άνοιγμα αρχείο
fwrite	Εγγραφή σε αρχείο
fclose	Κλείσιμο αρχείου
isset	Έλεγχος εάν ορίζεται μεταβλητή
break	Διακοπή επανάληψης
asXml	Αποθήκευση XML αρχείου
array_push	Προσθήκη στοιχείου σε πίνακα
serialize	Μετατροπή array σε μοναδική μεταβλητή
unserialize	Μετατροπή μοναδικής μεταβλητής σε array
count	Άθροισμα στοιχείων πίνακα
array	Δήλωση πίνακα
unset	Αποδέσμευση μεταβλητής
date	Προβολή ημερομηνίας
in_array	Έλεγχος ύπαρξης μέσα σε πίνακα
intval	Μετατροπή μεταβλητής σε integer

(string)	Μετατροπή μεταβλητής σε string
explode	Προσθήκη στοιχείων string μέσα σε πίνακα

Πίνακας 4: PHP εντολές

XML βάση

Δομή και σχεδιασμός Βάσης

Η δομή της βάσης μας δεν έχει καμία διαφορά από την βάση που θα κάναμε εάν χρησιμοποιούσαμε mysql. Έχουμε δημιουργήσει 5 πίνακες – users, stores, stocks, products και orders – και κάθε πίνακας αποτελείται από της εγγραφές που χρησιμοποιούμε μέσα στον κώδικα μας.

Παρακάτω μπορούμε να δούμε ένα δείγμα της βάσης που χρησιμοποιούμε στην σελίδα μας.

```
<database>

  <users>

    <user id="0">
      <name>admin</name>
      <username>admin</username>
      <password>admin</password>
    </user>

    <user id="4">
      <name>hj</name>
      <adress>mjjhbk</adress>
      <telephone>bjhfhjggbjftgj</telephone>
      <creditcard>ikkg</creditcard>
      <username>dani</username>
      <password>dani</password>
      <email>andriana_27-12@hotmail.com</email>
    </user>

  </users>

  <stores>

    <store id="1">
      <adress>G. Chalkiadaki</adress>
      <manager>Giorgos Kav</manager>
      <telephone>6957352546</telephone>
    </store>

    <store id="2">
      <adress>Rolen 14</adress>
      <manager>Apostolos</manager>
      <telephone>2810225968</telephone>
    </store>
  </stores>

  <products>

    <product id="45887952">
```

```

        <price>400</price>
        <name>Playstation 4</name>
        <category>games</category>
        <company>Sony</company>
        <image>ps4.jpg</image>
    </product>

    <product id="45887953">
        <price>350</price>
        <name>XBox 360</name>
        <category>games</category>
        <company>Microsoft</company>
        <image>xbox.jpg</image>
    </product>

</products>

<orders>

    <order id="3">
        <userid>andriana</userid>
        <date>26/11/2014</date>
        <checkdate>0</checkdate>
        <store>1</store>
        <checkdate>0</checkdate>
        <product>
            <code/>
            <quantity>1</quantity>
        </product>
    </order>

</orders>

<stocks>

    <stock>
        <codeproduct>45887952</codeproduct>
        <quantity>4</quantity>
        <storecode>1</storecode>
    </stock>

    <stock>
        <codeproduct>45887952</codeproduct>
        <quantity>5</quantity>
        <storecode>2</storecode>
    </stock>

</stocks>

</database>

```

Σύνδεση Βάσης

Η βάση μας συνδέεται με την σελίδα μας μέσα από την εντολή `simplexml_load_file`. Μέσα από αυτήν την εντολή φορτώνεται η βάση και γίνεται προσπελάσιμη από την PHP περίπου με την ίδιο τρόπο που χειριζόμαστε έναν πίνακα.

Έστω ότι έχουμε ένα στοιχείο `stocks` με παιδιά στοιχεία `stock` με την σειρά τους πεδίο `name`. Τότε μπορούμε να φτάσουμε την τιμή του κάθε `stock` απλά ακολουθώντας το μονοπάτι: `stocks->stock->name`. Περισσότερα παραδείγματα θα δείτε στην συνέχεια της αναφοράς.

Πλεονεκτήματα και Μειονεκτήματα XML

Η xml βάση είναι μια εύκολη, γρήγορη και οικονομική λύση για να δημιουργήσουμε μια βάση δεδομένων. Είναι εύκολη η δημιουργία καθώς δεν χρειάζεται να γίνει σύνδεση των πινάκων, να οριστούν σχέσης, μοναδικά κλειδιά και όλα αυτά τα τυπικά λάθη που μπορούν να γίνουν κατά την δημιουργία μας βάσης mysql.

Αυτό φυσικά εγγυμονεί και μεγάλους κινδύνους καθώς μπορεί εύκολα να γίνουν διπλότυπες εγγραφές εκεί που δεν πρέπει να γίνουν, η ασφάλεια εισαγωγής είναι πολύ μικρότερη από αυτή που παρέχει η mysql και πολλές φορές η βάση είναι προσβάσιμη από ανεπιθύμητους επισκέπτες, πράγμα που μπορεί να αποφευχθεί φυσικά με σωστό προγραμματισμό. Αυτό όμως απαιτεί την μεγαλύτερη προσοχή κατά την δημιουργία, επεξεργασία και αποθήκευση της βάσης xml. Επίσης ο προσπέλαση της xml βάσης δεν γίνονται με ερωτήματα αλλά με συνεχείς επαναλήψεις και έλεγχους συνθηκών μέχρι να φτάσουμε στο ζητούμενο στοιχείο, πράγμα που μπορεί να μας στοιχίσει πολύ περισσότερες γραμμές κώδικα.

Δομή Ιστοσελίδας

Κάθε επιλογή που κάνει ο επισκέπτης έχει σαν αποτέλεσμα την αποστολή μεταβλητών, είτε μέσω της μεθόδου POST, είτε μέσω της μεθόδου GET αλλά ακόμα και την SESSION. Ανάλογα την προστασία αλλά και την χρήση που θέλουμε να κάνουμε κάθε μεταβλητής τότε επιλέγουμε την κατάλληλη μεταβλητή.

Κάθε μεταβλητή είναι ένας παράγοντας διαμόρφωσης της παρουσίασης της σελίδας, αλλά και τον δυνατοτήτων που δίνονται στον επισκέπτη. Για παράδειγμα εάν υπάρξει η μεταβλητή `$_SESSION['user']`, τότε δίνονται επιπλέον επιλογές που και πάλι διαφέρουν, ανάλογα το ποιος είναι ο χρήστης – διαχειριστής ή επισκέπτης.

Η ιστοσελίδα βασίζεται σε ένα μόνο αρχείο που εμφανίζεται στο url, το `index.php`. Από εκεί και πέρα το περιεχόμενο αυτής της σελίδας δημιουργείτε με συνθήκες και ελέγχους των μεταβλητών που μας δίνονται και την ανάλογη εισαγωγή αντίστοιχων αρχείων. Σε πολλές σελίδες υπάρχουν αρχεία που είναι κοινά οπότε έτσι γίνεται η εισαγωγή των ίδιων αρχείων.

Γιατί όμως χρησιμοποιούμε αυτήν την μέθοδο; Αυτή η μέθοδος δημιουργίας ιστοσελίδων είναι αγαπημένος τρόπος ανάπτυξης μεγάλων και δυναμικών ιστοσελίδων καθώς δεν επαναλαμβάνεται ο ίδιος κώδικας για όλες τις σελίδες. Ο προγραμματιστής γράφει μια φορά βασικά κομμάτια της σελίδας, όπως το header, το footer αλλά και κάποια μενού που είναι κοινά για όλες τις υποσελίδες. Έτσι όταν είναι να γίνει μια αλλαγή, χρειάζεται μόνο η αλλαγή ενός αρχείου και η αλλαγή εφαρμόζεται σε όλες τις υποσελίδες. Με αυτόν τον τρόπο ο προγραμματισμός γίνεται πιο εύκολα, οργανωμένα και με μεγαλύτερη αποφυγή λαθών. Ο έλεγχος κάποιων δυσλειτουργιών είναι εξίσου εύκολος καθώς τα κομμάτια αυτά γίνονται σταδιακά και κάθε κομμάτι δοκιμάζεται πριν εισαχθεί μέσα στην βασική σελίδα.

Η εισαγωγή των αρχείων περιεχομένων της σελίδας γίνεται με την εντολή της PHP `include`. Τα βασικά κομμάτια που γίνονται εισαγωγή σε κάθε υποσελίδα του ηλεκτρονικού μας καταστήματος είναι:

- Η επικεφαλίδα (header της html)

Το header της σελίδας παραμένει σταθερό για όλες τις υποσελίδες του ηλεκτρονικού μας καταστήματος καθώς όλα τα περιεχόμενα του είναι χρήσιμα για αυτές

`<head>`

```
<?php header('Content-Type: text/html; charset=UTF-8');?>
```

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

```
<link rel="stylesheet" href="css/colorbox.css" />
```

```
<link href="//maxcdn.bootstrapcdn.com/font-awesome/4.2.0/css/font-awesome.min.css" rel="stylesheet">
```

```
<link rel="stylesheet" href="//code.jquery.com/ui/1.11.2/themes/smoothness/jquery-ui.css">
```

```
<script type="text/javascript" src="http://code.jquery.com/jquery-1.11.1.min.js" ></script>
```

```
<script src="//code.jquery.com/ui/1.11.2/jquery-ui.js"></script>
```

```
<script src="js/jquery.colorbox.js"></script>
```

```
</head>
```

- Το αρχείο user.php

Το αρχείο user.php είναι αυτό που ελέγχει για απόπειρα εισαγωγής χρήστη, εάν έχει ήδη γίνει εισαγωγή αλλά και κάποιες άλλες βασικές λειτουργίες της σελίδας.

```
session_start();  
  
if(isset($_GET['action']) && $_GET['action'] == 'logout'){  
  
    session_destroy();  
  
    unset($_SESSION);  
  
}
```

Σε αυτό το κομμάτι ελέγχει εάν υπάρχει ή όχι η εντολή προς την σελίδα να αποσυνδέσει τον χρήστη. Εάν υπάρχει αυτή η εντολή τότε καταστρέφει την SESSION που έχει οριστεί στον server και αποδεσμεύει τις μεταβλητές της.

```
if(isset($_POST['login'])){  
    $xml=simplexml_load_file("data.xml");  
    foreach($xml->users->user as $user){  
        if($user->username == $_POST['username'] && $user->password ==  
$_POST['password']) {  
            $userLog = (string)$user->name;  
            $_SESSION['user'] = $userLog;  
            break;  
        }else{  
            $error = 'Login Error: wrong username or password.';  
        }  
    }  
}
```

Αυτό το κομμάτι του κώδικα είναι υπεύθυνο εάν έχει γίνει submit η φόρμα σύνδεσης να ελέγξει το xml για τον χρήστη που προσπαθεί να συνδεθεί. Εάν ο χρήστης υπάρχει και ο κωδικός που δίνει είναι ο ίδιος με αυτόν που είναι αποθηκευμένος μέσα στην βάση τότε τον συνδέει και

αποθηκεύει το όνομα του χρήστη στον σέρβερ έτσι ώστε να μείνει συνδεδεμένος για όλη την διάρκεια της παραμονής του στην σελίδα

```
if(isset($_POST['register'])){\n    $xml=simplexml_load_file("data.xml");\n    foreach($xml->users->user as $user){\n        if($user->username == $_POST['username']) {\n            $error= 'Username is already exist.';\n            break;\n        }\n    }\n    if($error== 'none'){ \n        $newuser = $xml->users->addChild('user');\n        $newuser->addChild('name', $_POST['name']);\n        $newuser->addChild('adress', $_POST['adress']);\n        $newuser->addChild('telephone', $_POST['telephone']);\n        $newuser->addChild('creditcard', $_POST['credit']);\n        $newuser['id'] = count($xml->users->user)+1;\n        $newuser->addChild('username', $_POST['username']);\n        $newuser->addChild('password', $_POST['passwordReg']);\n        $newuser->addChild('email', $_POST['email']);\n        $xml->asXML("data.xml");\n        $message="Registration was successful. Please login";\n    }\n}
```

Το παραπάνω κομμάτι του κώδικα είναι υπεύθυνο να ελέγχει εάν έχει γίνει αίτηση εγγραφής νέου χρήστη. Εάν το κάτι τέτοιο ισχύει αναζητά την βάση για τυχόν ήδη υπάρχων username με αυτό που δόθηκε. Εάν βρει τότε διακόπτει την εγγραφή. Εάν όμως δεν βρει τότε καταγράφει τον χρήστη στην βάση μαζί με τα ζητούμε στοιχεία.

```
if(isset($_SESSION['order'])){\n    if(isset($_POST['removeItem'])){\n        $order = unserialize($_SESSION['order']);\n        foreach($order as $prod=>$p){\n            if($p['cod'] == $_POST['removeItem']){\n                unset($order[$prod]);\n                $_SESSION['order'] = serialize($order);\n            }\n        }\n    }\n    $order = unserialize($_SESSION['order']);\n}else{\n    $order = array();\n}
```

```
}
```

Ο παραπάνω κώδικας είναι αυτός που ελέγχει εάν έχει δοθεί στον σέρβερ το λεγόμενο καλάθι αγορών. Εάν ο χρήστης έχει προσθέσει μέχρι στιγμής προϊόντα στο καλάθι αγορών τότε αυτά στέλνονται στην μεταβλητή \$_SESSION['order'] στον σέρβερ. Επίσης γίνεται έλεγχος εάν έχει ζητηθεί από τον χρήστη η διαγραφή κάποιου προϊόντος από το καλάθι αγορών και αυτό ανανεώνεται στον σέρβερ.

```
if(count($order) > 0 && isset($_GET['action']) && $_GET['action'] == 'checkout'){
    $xml=simplexml_load_file("data.xml");
    $max = 0;
    foreach($xml->orders->order as $ord){
        if($max<$ord['id'])
            $max=$ord['id'];
    }
    $storeCode = null;
    foreach($xml->stores->store as $store){
        $storestock = true;
        foreach($order as $prod){
            foreach($xml->stocks->stock as $stock){
                if($stock->storecode == $store['id'] && $prod['cod'] ==
$stock->codeproduct){
                    if($stock->quantity < $prod['quant'] ) {
                        $storestock = false;
                    }
                }
            }
        }
        if($storestock == true) {
            $storeCode = $store['id'];
            foreach($order as $prod){
                foreach($xml->stocks->stock as $stock){
                    if((string)$stock->storecode === (string)$storeCode
&& (string)$prod['cod'] === (string)$stock->codeproduct){
                        $stock->quantity = intval($stock->quantity) -
intval($prod['quant']);
                    }
                }
            }
            break;
        }
    }
    $neworder = $xml->orders->addChild('order');
    $neworder->addAttribute('id', $max +1);
    $neworder->addChild('userid', $_SESSION['user']);
    $neworder->addChild('date', date('d/m/Y'));
```

```

$neworder->addChild('checkdate', 0);
if($storeCode == null) {
    $neworder->addChild('store', 'N/A');
}else{
    $neworder->addChild('store', $storeCode);
}
$neworder->addChild('checkdate', 0);
foreach($sorder as $prod){
    $product = $neworder->addChild('product');
    $product->addChild('code', $prod['cod']);
    $product->addChild('quantity', $prod['quant']);
}
$xml->asXML("data.xml");
$message="Thank you for your order";
unset($sorder);
unset($_SESSION['order']);
}

```

Αυτό το κομμάτι είναι υπεύθυνο για την ολοκλήρωση της κάθε παραγγελίας. Αρχικά ελέγχει την διαθεσιμότητα όλων των προϊόντων στα καταστήματα ανά σειρά. Εάν έστω και ένα προϊόν δεν υπάρχει σε κάποιο κατάστημα σταματά τον έλεγχο για αυτό το κατάστημα και ξεκινά να ελέγχει το επόμενο. Εάν αυτό επαναληφθεί για όλα τα καταστήματα χωρίς το επιθυμητό αποτέλεσμα τότε απλά δεν ορίζεται κατάστημα παραλαβής.

Κατά την αποθήκευση εάν τελικά έχει βρεθεί κατάστημα που μπορεί να εξυπηρετήσει την παραγγελία τότε αφαιρούνται από οι ποσότητες που ζητήθηκαν από κάθε προϊόν από το απόθεμα τους για το συγκεκριμένο κατάστημα.

- Το μεγαλύτερο κομμάτι του body

Βασικά τέτοια κομμάτια της σελίδας που παραμένουν σχεδόν σταθερά είναι τα εξής:

```
<div class="top">
```

```

<div class="date">
    <?php if(isset($_SESSION['user'])) { echo 'Welcome ' . $_SESSION['user'] . ' - '; } ?>
    &nbsp;
    <?php echo date('F j, Y'); ?>
</div>

```

```
<div class="topMenu">
```

```

    <?php if(isset($_SESSION['user'])) { ?>
    <a href="index.php?action=logout">Log out</a>

```



```

        <?php }else{ ?>

        <a class='login' href="#login_content">Log in</a>

        <?php } ?>

    </div>

</div>

```

Στην div κλάσης top υπάρχει η ημερομηνία, το καλωσόρισμα εάν είναι συνδεδεμένος κάποιος χρήστης αλλά και το μενού σύνδεσης/αποσύνδεσης

Με την επιλογή σύνδεσης μέσω του colobox, μιας συνάρτησης jquery ανοίγει ένα lightbox παράθυρο στο οποίο χρήστης έχει την επιλογή είτε να συνδεθεί είτε να κάνει εγγραφή. Το παράθυρο αυτό αποτελείται από τα εξής στοιχεία:

```

<div style='display:none'>

<div id='login_content' style='padding:10px; background:#fff;'>

<div class="left w45" style="border-right: 1px solid #cdcdcd; margin-right: 15px;">

<h3>Log In</h3>

<form method="POST" name="login" action="index.php">

<label>Username</label><br>

<input required type="text" value="<?php if(isset($_POST['username'])){echo $_POST['username'];} ?>" name="username" /> <br>

<label>Password</label><br>

<input required type="password" name="password" /><br>

<br>

<input type="submit" value="Login" name="login"/>

</form>

</div>

<div class="left w45">

```

```

<h3>Register</h3>

<form method="POST" name="register" action="index.php">

<label>Name</label><br>

<input required type="text" value="<?php if(isset($_POST['name'])){echo $_POST['name'];}
?>" name="name" /> <br>

<label>Address</label><br>

<input required type="text" value="<?php if(isset($_POST['address'])){echo $_POST['address'];}
?>" name="address" /> <br>

<label>Telephone</label><br>

<input required type="text" value="<?php if(isset($_POST['telephone'])){echo
$_POST['telephone'];} ?>" name="telephone" /> <br>

<label>Email</label><br>

<input required type="email" value="<?php if(isset($_POST['email'])){echo $_POST['email'];}
?>" name="email" /> <br>

<label>Credit Card</label><br>

<input required type="text" value="<?php if(isset($_POST['credit'])){echo $_POST['credit'];}
?>" name="credit" /> <br>

<label>Username</label><br>

<input required type="text" name="username" /> <br>

<label>Password</label><br>

<input required type="password" value="<?php if(isset($_POST['passwordReg'])){echo
$_POST['passwordReg'];} ?>" name="passwordReg" /> <br>

<br>

<input type="submit" value="Join" name="register"/>

</form>

</div>

```

```
</div>
```

```
</div>
```

Τα πεδία input χρησιμοποιούν το required που είναι στοιχείο της HTML5. Έτσι πατώντας την αποστολή στοιχείων, η HTML ελέγχει εάν έχουν γίνει εισαγωγή των απαιτούμενων στοιχείων. Εάν δεν έχουν γίνει τότε δεν αφήνει την φόρμα να αποσταλεί.

```
<div class="header">
```

```
<div class="logo"><h1><i class="fa fa-truck"></i> E-Shop</h1></div>
```

```
<div class="right">&nbsp;<i class="fa fa-search" id="searchButton"></i></div>
```

```
<div class="right">
```

```
<form id="search" method="POST" action="index.php?action=search" >
```

```
<input type="text" name="searchfor" placeholder="Search" />
```

```
</form>
```

```
</div>
```

```
<?php include('menu.php'); ?>
```

```
</div>
```

Το header.php είναι το βασικό κομμάτι της σελίδας που περιέχει το λογότυπο και το μενού. Αυτό γίνεται include μέσα στην index.php. Το μενού υπάρχει και αυτό μέσα στο menu.php:

```
<div class="menu">
```

```
<?php if (isset($_SESSION['user']) && $_SESSION['user'] == 'admin'){
```

```
?>
```

```
<a href="index.php?action=modify" class="first">Modify a Product</a><a
```

```
href="index.php?action=stock">Show Stock</a><a
```

```
href="index.php?action=ordersAdmin">Show Orders</a><a href="index.php?action=stores"
```

```
class="last">Show Stores</a>
```

```
<?php
```

```
}else { ?>
```

```
<a href="index.php" class="first">Home</a><a  
href="index.php?category=computers">Computers</a><a  
href="index.php?category=games">Games</a><a  
href="index.php?category=mobiles">Mobiles</a><a class="last"  
href="index.php?file=contact">Contact</a>
```

```
<?php } ?>
```

```
</div>
```

Το μενού βλέπουμε ότι αλλάζει εάν υπάρχει συνδεδεμένος διαχειριστής. Δίνονται διαφορετικές επιλογές από τον επισκέπτη και τον πελάτη

Το footer.php που γίνεται include στο τέλος τις σελίδας είναι αυτό που περιέχει το υποσέλιδο της σελίδας και παραμένει σταθερό για όλες της σελίδας

```
<div id="footer">
```

```
  <div class="left social">
```

```
    <h3>Social Networks</h3>
```

```
    <p>Follow us and join us on our social networks.</p>
```

```
    <i class="fa fa-facebook-square"></i>&nbsp;
```

```
    <i class="fa fa-google-plus-square"></i>&nbsp;
```

```
    <i class="fa fa-twitter-square"></i>&nbsp;
```

```
    <i class="fa fa-pinterest-square"></i>&nbsp;
```

```
  </div>
```

```
  <div class="right" style="margin-right: 50px">
```

```
    <h2><i class="fa fa-truck"></i> E-Shop</h2>
```

```
    <b>Adress:</b> Estayromenos<br>
```

```
    <b>Email:</b> info@e-shop.com<br>
```

```
    <b>Telephone:</b> 123-4567890
```

```
  </div>
```

```

<div class="copyright">&copy; E-Shop <?php echo date('Y'); ?></div>

<script type="text/javascript" >

$("#login").colorbox({inline:true, width:"50%"});

$( "#searchButton" ).click(function() {

    $( "#search" ).toggle("slow");

});

</script>
</div>

<?php
if(!isset($_SESSION['user'])){

    include('login.php');

} ?>

```

Σε αυτό το σημείο έχοντας αναλύσει τα βασικά στοιχεία τις σελίδας που παραμένουν στο μεγαλύτερο σημείο σταθερά θα θεωρήσουμε δεδομένα τα εξής στην συνέχεια της αναφοράς. Κάθε μεγάλη αλλαγή στις υπόλοιπες σελίδες εμφανίζονται στην div με id = content, η οποία φαίνεται παρακάτω.

```

<div id="content">

<?php if(isset($_SESSION['user']) && $_SESSION['user'] == 'admin'){

    include('admin.php');

} else{ ?>

<div id="left">

<?php include('left.php'); ?>

</div>

<div id="right">

<?php

```

```

if(isset($_GET['category'])){

    include('category.php');

}else if(isset($_GET['action']) && $_GET['action'] == 'cart'){

    include('cart.php');

    }else if(isset($_GET['action']) && $_GET['action'] == 'orders'){

    include('orders.php');

    }else if(isset($_GET['action']) && $_GET['action'] == 'checkout'){

    include('checkout.php');

    }else if(isset($_GET['action']) && $_GET['action'] == 'search'){

    include('search.php');

    }else {

    include('home.php')    ;

    }

?>

</div>

<?php } ?>

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

</div>

```

Παρατηρούμε ότι εάν γίνει σύνδεση του διαχειριστή κάνει include το αρχείο admin.php ενώ για απλούς χρήστες ή μη εγγεγραμμένους επισκέπτες λαμβάνει φορτώνει τα ίδια αρχεία, πάντα ανάλογα την μεταβλητή \$_GET['action'] που δίνεται μέσω το url.

Αρχική σελίδα

Στην αρχική σελίδα γίνεται η εμφάνιση των 4 πιο πρόσφατων προϊόντων.

```

<form name="order" id="order" method="POST" action="">
    <input value="" name="add" id="add" type="hidden" />
</form>
<?php
$xml=simplexml_load_file("data.xml");
$i = 0;
foreach($xml->products->product as $product){
    $i++;
    ?>
    <div class="product">
        <div class="proImg"></div>
        <div class="desc">
            <h2><?php echo $product->name;?> <span class="category" style="font-
size:13px;"><?php echo $product['id'];?></span></h2>
            <div class="desc1">
                <span class="category">Category: <i><?php echo $product->category;?></i></span>
            <br>
            <p>Company: <?php echo $product->company;?></p>
        </div>
            <div class="desc2">
                <span class="price"><?php echo $product->price;?> &euro;</span><br>

```

```
<span class="buy" onclick="<?php if(!isset($_SESSION['user'])) { ?>javascript:
alert('Please login first!');<?php }else{ ?> javascript:
document.getElementById('add').value='<?php echo $product['id'];?>';
document.getElementById('order').submit(); <?php } ?>">Buy it!</span>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<?php
if($i == 4){break;}

}

?>
```

Με την επιλογή αγοράς κάποιου προϊόντος γίνεται έλεγχος εάν υπάρχει συνδεδεμένος χρήστης. Εάν ναι γίνεται submit η φόρμα order με την οποία και προστίθεται το προϊόν στο καλάθι. Εάν το προϊόν υπάρχει ήδη στο καλάθι τότε απλά αυξάνεται η ποσότητα του.

Πλοήγηση Μη εγγεγραμμένου επισκέπτη

Ο μη εγγεγραμμένος επισκέπτης έχει την δυνατότητα να πλοηγηθεί κανονικά στα προϊόντα αλλά δεν έχει την δυνατότητα να προσθέσει στο καλάθι αγορών κάποιο προϊόν. Δεν του είναι ορατό το μενού για το καλάθι αγορών αλλά ούτε και η εμφάνιση των παραγγελιών που έχουν γίνει. Το μενού αυτό δίνεται στην div left η οποία δεν εμφανίζεται εάν δεν υπάρχει χρήστης συνδεδεμένος.

Παρακάτω θα δούμε τον κώδικα για της σελίδες που μπορεί πλοηγηθεί ο επισκέπτης, καθώς είναι ίδιες με αυτές του εγγεγραμμένου πελάτη.

Εγγραφή πελάτη

Η εγγραφή του κάθε επισκέπτη μπορεί να γίνει από όλες τις σελίδες. Εάν κάποιος επισκέπτης έχει ήδη συνδεθεί δεν μπορεί να κάνει νέα εγγραφή, όχι τουλάχιστον όσο παραμένει συνδεδεμένος. Δεν υπάρχει κάποια επιπλέον σελίδα για την εγγραφή καθώς όπως προαναφέραμε στην αρχή του κεφαλαίου είναι ένα από τα δυναμικά μενού που υπάρχει σε όλες τις υποσελίδες.

Σύνδεση πελάτη

Ο επισκέπτης μπορεί επίσης να συνδεθεί ανά πάσα στιγμή χωρίς να χρειάζεται να αλλάξει σελίδα. Μέσα από το αναδυόμενο παράθυρο μπορεί να κάνει σύνδεση και αφού είναι πετυχημένη η σύνδεση να βρεθεί στην ίδια υποσελίδα, συνεχίζοντας κανονικά την πλοήγηση του και τις αγορές του. Για τον κώδικα δείτε την αρχή του κεφαλαίου.

Πλοήγηση πελάτη

Ο πελάτης μέσα από το μενού στο header μπορεί να πλοηγηθεί στις κατηγορίες, να επανέλθει στην αρχική ή να πάει στην σελίδα επικοινωνίας. Επίσης μπορεί να αναζητήσει κάποιο προϊόν βάση το όνομα, τον κατασκευαστή ή την κατηγορία, δίνοντας το όνομα στην φόρμα αναζήτησης που βρίσκεται μόνιμα πάνω δεξιά στο site.

Όσο παραμένει συνδεδεμένος ο πελάτης μπορεί να δει και το καλάθι αγορών του αλλά και τις παραγγελίες που έχει ήδη καταχωρήσει. Αυτές οι επιλογές δίνονται μέσα στην div left που όπως προαναφέραμε είναι ορατή μόνο σε αυτόν.

Στο καλάθι των αγορών ο πελάτης έχει την δυνατότητα να αφαιρέσει κάποιο προϊόν από το καλάθι αλλά και να καταχωρήσει την παραγγελία του.

```
<table><tr><td style="width: 15%">Name</td><td style="width: 5%">Quantity</td><td style="width: 15%">Price</td><td style="width: 5%">Cancel</td></tr>
```

```
<?php
```

```
$xml=simplexml_load_file("data.xml");
```

```
foreach($order as $prod){
```

```
    foreach($xml->products->product as $product){
```

```
        if($product['id'] == $prod['cod']) {
```

```
            echo '<tr><td>'.$product->name.'</td><td>'.$prod['quant'].'</td><td>'.$product->price * $prod['quant'].'</td><td></td></tr>';
```

```
        }
```

```
    }
```

```
}
```

```
?>
```

```
</table>
```

```
<div class="right button"><a href="index.php?action=checkout">Order</a></div>
```

```
<form name="remove" id="remove" method="POST" action="">
```

```
    <input value="" name="removeItem" id="removeItem" type="hidden" />
```

```
</form>
```

Η ολοκλήρωση της παραγγελίας γίνεται μέσα από url το οποίο δίνει την μεταβλητή `$_GET['action'] = checkout` όπου και ελέγχεται στην αρχή της σελίδας – από το αρχείο `user.php` – έτσι ώστε να γίνει σωστά η καταχώρηση της, να οριστεί το σωστό κατάσταση εξυπηρέτησης, να αφαιρεθούν τα προϊόντα από τα αποθέματα και να περιμένει την έγκριση του διαχειριστή.

Επίσης ο πελάτης μπορεί να ακυρώσει μια παραγγελία εάν δεν έχει εγκριθεί από τον διαχειριστή. Επιλέγοντας να δει τις παραγγελίες του, ο κώδικας ελέγχει εάν έχει εγκριθεί ή όχι και ανάλογα το αποτέλεσμα δίνει την δυνατότητα ή όχι στον πελάτη να την ακυρώσει.

```
<h2>Yours orders</h2>
```

```
<form name="removeOrder" id="removeOrder" method="POST" action="">
```

```
    <input value="" name="remove" id="remove" type="hidden" />
```

```
</form>
```

```
<?php
```

```
$xml= simplexml_load_file('data.xml');
```

```
if(isset($_POST['remove'])) {
```

```
    $i=0;
```

```
    foreach($xml->orders->order as $order){
```

```

        if($order['id'] == $_POST['remove']) {
            break;
        }
        $i++;
    }
    unset($xml->orders->order[$i]);
    $xml->asXml('data.xml');
}

```

```

foreach($xml->orders->order as $orderID){

```

```

    if($orderID->userid == $_SESSION['user']){
        ?>
        <h3>Order <?php echo $orderID['id'];?></h3>

        <?php
            echo 'Store: '.$orderID->store.<br>';
            foreach($orderID->product as $pro){
                echo 'Product code: '.$pro->code.', quantity: '.$pro-
>quantity.<br>';
            }
            echo<br><br>';

```

```

        if($orderID->checkdate != 0){echo 'You can not cancel this
order;'}else{echo '<span class="button" onclick="javascript:
document.getElementById(\'remove\').value=\'".$orderID[\'id\'].\'';
document.getElementById(\'removeOrder\').submit();" >Cancel Order</span>'}

    }

}

?>

```

Σύνδεση Διαχειριστή

Όταν συνδεθεί ο διαχειριστής τότε αρχικά του εμφανίζεται μια σελίδα με όλα τα προϊόντα τα οποία έχουν την επιλογή να το επεξεργαστεί.

```

<form name="modify" id="modify" method="POST" action="">
    <input value="" name="modifyCode" id="modifyCode" type="hidden" />
</form>

<?php

if(isset($_POST['modifyCode'])){

    $xml= simplexml_load_file('data.xml');

    $i=0;

foreach($xml->products->product as $product){

    if($product['id'] == $_POST['modifyCode']){break;}

```

```

    $i++;
}

?>

<form name="modified" id="modified" method="POST" action="">

    <input value="<?php echo $xml->products->product[$i]->name;?>"
name="productName" id="productName" type="text" />

    <input value="<?php echo $xml->products->product[$i]->category;?>"
name="productCat" id="productCat" type="text" />

    <input value="<?php echo $xml->products->product[$i]['id'];?>" name="productCode"
id="productCode" type="text" readonly/>

    <input value="<?php echo $xml->products->product[$i]->company;?>"
name="productCompany" id="productCompany" type="text" />

    <input value="<?php echo $xml->products->product[$i]->price;?>"
name="productPrice" id="productPrice" type="text" />

    <input value="<?php echo $xml->products->product[$i]->image;?>"
name="productImage" id="productImage" type="text" />

    <input value="<?php echo $i;?>" name="pos" id="pos" type="hidden" />

    <input name="save" id="save" type="submit" value="Save" class="button" />

</form>

<?php
}else{

    $xml= simplexml_load_file('data.xml');

```

```

        if(isset($_POST['save'])){
            foreach($xml->
>xpath("products/product[@id='".$_POST['productCode']."'"]) as $change){
                $change->category = $_POST['productCat'];
                $change->name = $_POST['productName'];
                $change->company = $_POST['productCompany'];
                $change->price = $_POST['productPrice'];
                $change->image = $_POST['productImage'];
            }
            $xml->asXML('data.xml');
        }
    }

    foreach($xml->products->product as $product){
        ?>
        <div class="product">
            <div class="proImg"></div>
            <div class="desc">
                <h2><?php echo $product->name;?> <span class="category" style="font-
size:13px;"><?php echo $product['id'];?></span></h2>
                <div class="desc1">
                    <span class="category">Category: <i><?php echo $product->category;?></i></span>
                <br>

```

```

<p>Company: <?php echo $product->company;?></p>

</div>

<div class="desc2">

<span class="price"><?php echo $product->price;?> &euro;</span><br>

<span class="buy" onclick="javascript:
document.getElementById('modifyCode').value='<?php echo $product['id'];?>';
document.getElementById('modify').submit();">Edit</span>

</div>

</div>

</div>

</div>

                <?php
}

}

?>

```

Εάν επιλεγεί εάν προϊόν τότε φορτώνεται πάλι η ίδια σελίδα αλλά έχοντας διαφορετικά δεδομένα πλέον δίνονται τα πεδία του προϊόντος στον διαχειριστή μέσα σε μια φόρμα, τα οποία και μπορεί να αλλάξει και να αποθηκεύσει.

Πλοήγηση Διαχειριστή

Ο διαχειριστής έχει την δυνατότητα να δει και να τροποποιήσει την διαθεσιμότητα κάθε προϊόντος σε κάθε κατάσταση. Δίνεται ένα πίνακας με τους κωδικούς των προϊόντων, τον κωδικό του καταστήματος αλλά και ένα input που περιέχει το τωρινό απόθεμα. Αλλάζοντας αυτόν τον αριθμό και αποθηκεύοντας τότε αλλάζει και το απόθεμα. Αρχικά ο πίνακας δείχνει όλα τα προϊόντα, σε όλα τα υποκαταστήματα και όλες της κατηγορίες. Στα δεξιά του πίνακα υπάρχουν 3 select στα οποία ο διαχειριστής μπορεί να φιλτράρει το ακριβώς προϊόντα που θέλει να δει – είτε ανά κατάσταση, είτε ανά κατηγορία ή ακόμα και προϊόν.

```
<form name="quantity" class="left w50" id="quantity" method="POST" action="" >
```

```

<?php
$xml = simplexml_load_file('data.xml');
if(isset($_POST['save'])){

    foreach($_POST as $productcode=>$quantity) {
        $product = explode("@", $productcode);
        foreach($xml->stocks->stock as $stock){
            if($product[0] == $stock->codeproduct && $product[1] == $stock-
>storecode) {
                if($stock->quantity != $_POST[$productcode]){
                    $stock->quantity = $_POST[$productcode];
                    $xml->asXML('data.xml');
                }
            }
        }
    }
}

if(isset($_SESSION['user']) && $_SESSION['user'] == 'admin'){

    echo '<table style="width:100%"><tr style="background: #eee; color: #3E5363; font-
weight: bold"><td>Store</td><td>Product</td><td>Quantity</td><td></td></tr>';

```



```

if (isset($_POST['stores']) && $_POST['stores'] !== ""){

    foreach($xml->stocks->stock as $stock){

        if($stock->storecode == $_POST['stores']) {

            echo '<tr><td>'.$stock->storecode.'</td><td>'.$stock-
>codeproduct.'</td><td><input type="number" name="'. $stock->codeproduct.'@'. $stock-
>storecode.'" value="'. $stock->quantity.'" /></td><td></td></tr>';

        }

    }

}

}else if (isset($_POST['products']) && $_POST['products'] !== ""){

    foreach($xml->stocks->stock as $stock){

        if($stock->codeproduct == $_POST['products']) {

            echo '<tr><td>'.$stock->storecode.'</td><td>'.$stock-
>codeproduct.'</td><td><input type="number" name="'. $stock->codeproduct.'@'. $stock-
>storecode.'" value="'. $stock->quantity.'" /></td><td></td></tr>';

        }

    }

}

}else if (isset($_POST['categories']) && $_POST['categories'] !== ""){

    $array = array();

    foreach($xml->products->product as $product){

        if($product->category == $_POST['categories']) {

            array_push($array, (string)$product['id']);

        }

    }

    foreach($xml->stocks->stock as $stock){

        if(in_array($stock->codeproduct, $array)) {

```

```

        echo '<tr><td>'.$stock->storecode.'</td><td>'.$stock-
>codeproduct.'</td><td><input type="number" name="'.$stock->codeproduct.'@'.$stock-
>storecode.'" value="'.$stock->quantity.'" /></td><td></td></tr>';

    }

}

}else{

    foreach($xml->stocks->stock as $stock){

        echo '<tr><td>'.$stock->storecode.'</td><td>'.$stock-
>codeproduct.'</td><td><input type="number" name="'.$stock->codeproduct.'@'.$stock-
>storecode.'" value="'.$stock->quantity.'" /></td><td></td></tr>';

    }

}

echo '</table>';

}

```

```

?>

<input type="submit" class="button" name="save" value="Save"/>

</form>

<div class="right">

<form action="" method="POST" name="showOrder" id="showOrder">

<label>Store</label>

    <select name="stores" id="stores" onchange="javascript:
document.getElementById('products').value=''; document.getElementById('categories').value='';
document.getElementById('showOrder').submit();">

        <option value="">All</option>

    <?php

        foreach($xml->stores->store as $store){

```

```

        echo '<option value="'. $store['id']. '""';

        if (isset($_POST['stores']) && $_POST['stores'] ==
$store['id']){echo ' selected';}

        echo '>Store ' . $store['id']. ' </option>';

    }

?>

</select>

<label>Product</label>

    <select name="products" id="products" onchange="javascript:
document.getElementById('stores').value=""; document.getElementById('categories').value="";
document.getElementById('showOrder').submit();">

        <option value="">All</option>

        <?php

            foreach($xml->products->product as $product){

                echo '<option value="'. $product['id']. '""';

                if (isset($_POST['products']) && $_POST['products']
== $product['id']){echo ' selected';}

                echo '>' . $product->name. ' </option>';

            }

        ?>

    </select>

<label>Category</label>

    <select name="categories" id="categories" onchange="javascript:
document.getElementById('stores').value=""; document.getElementById('products').value="";
document.getElementById('showOrder').submit();">

        <option value="">All</option>

```

```

        <option value="games" <?php if (isset($_POST['categories']) &&
$_POST['categories'] == 'games'){echo ' selected';} ?>>Games</option>

        <option value="computers" <?php if (isset($_POST['categories']) &&
$_POST['categories'] == 'computers'){echo ' selected';} ?>>Computers</option>

        <option value="mobile" <?php if (isset($_POST['categories']) &&
$_POST['categories'] == 'mobile'){echo ' selected';} ?>>Mobiles</option>

</select>

<br>

<br>

</form></div>

```

Επιλέγοντας από το μενού να δει τις παραγγελίες βλέπει αρχικά όλες τις παραγγελίες που έχουν γίνει μέχρι τώρα. Ένας πίνακας εμφανίζεται με το όνομα του πελάτη, τον κωδικό της παραγγελίας, την ημερομηνία καταχώρησης παραγγελίας, την ημερομηνία έγκρισης εάν έχει γίνει, αλλιώς ένα κουμπί με το οποίο δίνει έγκριση, τα προϊόντα και την ποσότητα κάθε προϊόντος και το συνολικό ποσό που πρέπει να εισπραχθεί.

Στα δεξιά του πίνακα υπάρχει μια φόρμα μέσα από την οποία ο διαχειριστής μπορεί να εφαρμόσει κάποια φίλτρα ώστε να μπορέσει να γίνει πιο εύκολη η διαχείριση των παραγγελιών. Μέσα από αυτήν την φόρμα μπορεί να δει τις παραγγελίες που έγιναν ανά κατάσταση, ανά προϊόν, ανά κατηγορία ή ανά κατασκευαστή και όλα αυτά μέσα σε ένα εύρος ημερομηνιών. Εάν δεν δοθούν ημερομηνίες τότε εμφανίζει τα παραπάνω από την αρχή του χρόνου για την σελίδα.

Τελευταία επιλογή του διαχειριστή είναι να δει τα καταστήματα που υπάρχουν και τις βασικές πληροφορίες. Μέσα από αυτήν την σελίδα ο διαχειριστής μπορεί να δει τον κωδικό κάθε καταστήματος, το όνομα του υπεύθυνου, την οδό του, το τηλέφωνο αλλά και τον μέσο όρο που χρειάζεται μια παραγγελία για να εγκριθεί.

```
<form name="quantity" class="left w70" id="quantity" method="POST" action="">
```

```
<?php
```

```
$xml = simplexml_load_file('data.xml');
```

```
if(isset($_SESSION['user']) && $_SESSION['user'] == 'admin'){
```

```
    echo '<table style="width:100%"><tr style="background: #eee; color: #3E5363; font-weight: bold"><td>Store</td><td>Manager</td><td>Adress</td><td>Telephone</td><td>Avarage Days Per Order</td></tr>';
```

```
        foreach($xml->stores->store as $store){
```

```
            $or = 0;
```

```
            foreach($xml->orders->order as $order){
```

```
                if((string)$order->store == $store['id']){
```

```
                    if($order->checkdate == "0"){
```

```
                        $day = $day + (date("d/m/Y") - date($order->date));
```

```
                    }else{
```

```
                        $day = $day + (date($order->checkdate) - date($order->date));
```

```
                    }
```

```
                    $or++;
```

```
                }
```

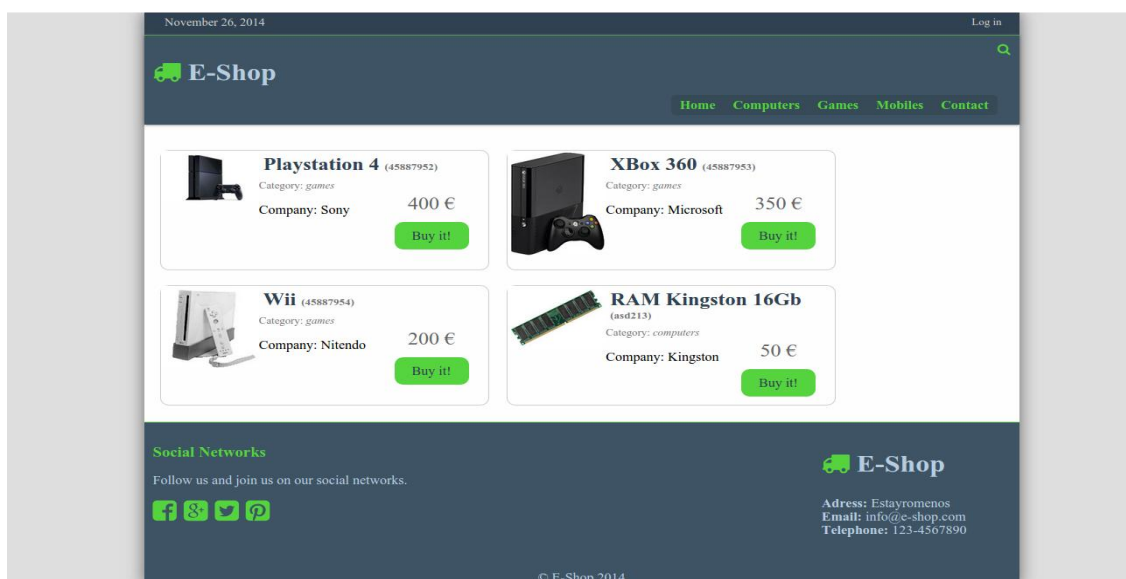
```
            }
```

```
            $sav = $day/$or;
```

```
        echo '<tr><td>'.$store['id'].'</td><td>'.$store-
>manager.'</td><td>'.$store->adress.'</td><td>'.$store-
>telephone.'</td><td>'.intval($av).'
```

Παρουσίαση Ιστοσελίδας

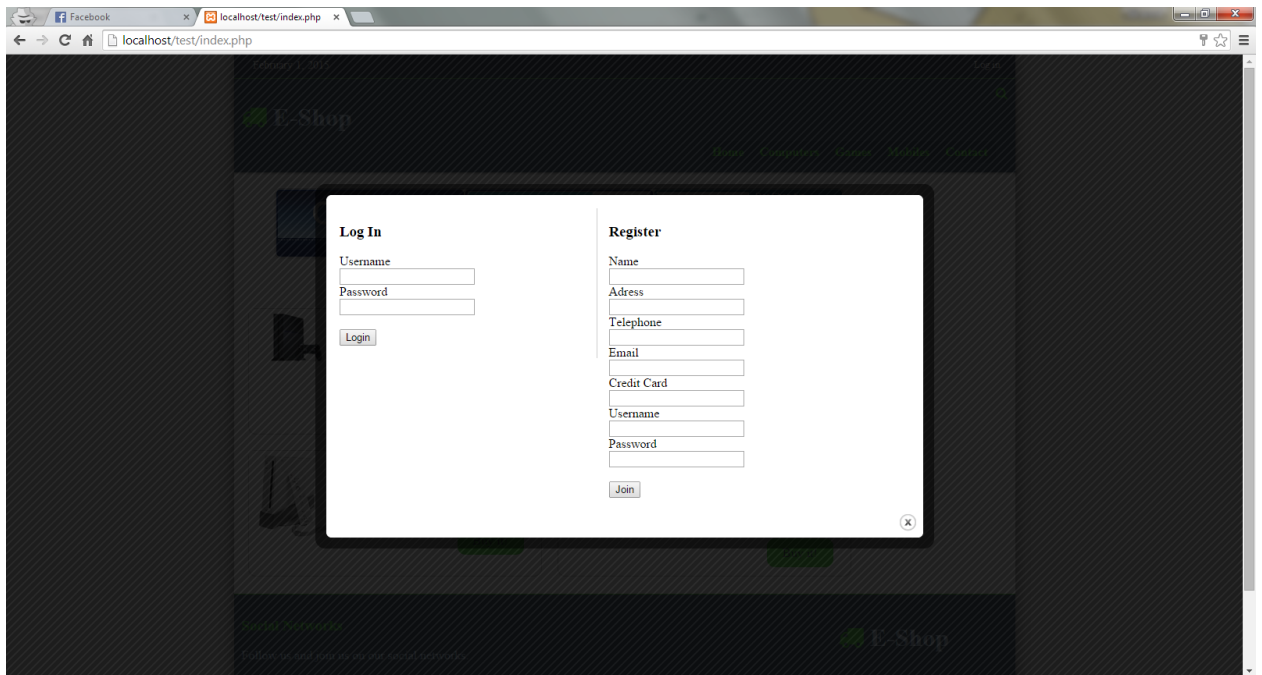
Αρχική σελίδα



Εικόνα 8: Αρχική σελίδα

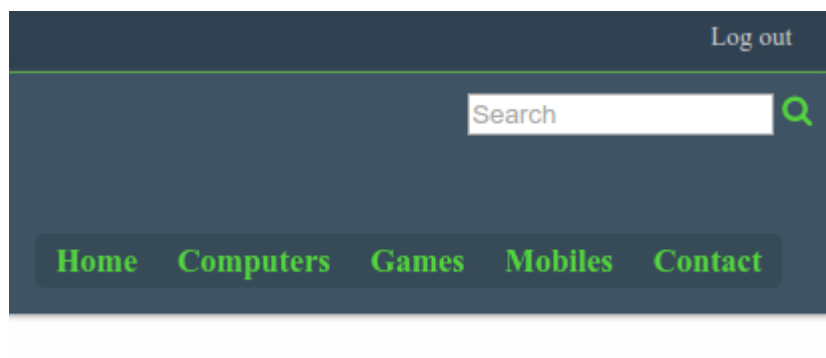
Η αρχική σελίδα όπως μπορούμε να δούμε στην παραπάνω εικόνα χωρίζεται σε 3 βασικά σημεία. Το πρώτο που είναι το header περιέχει το λογότυπο της σελίδας, το μενού πλοήγησης, το κουμπί αναζήτησης που εμφανίζεται αφού κάνουμε click πάνω στο εικονίδιο της αναζήτησης – το οποίο χρησιμοποιούμε jquery για να το πετύχουμε – και το μενού πάνω από αυτήν.

Σύνδεση Πελάτη

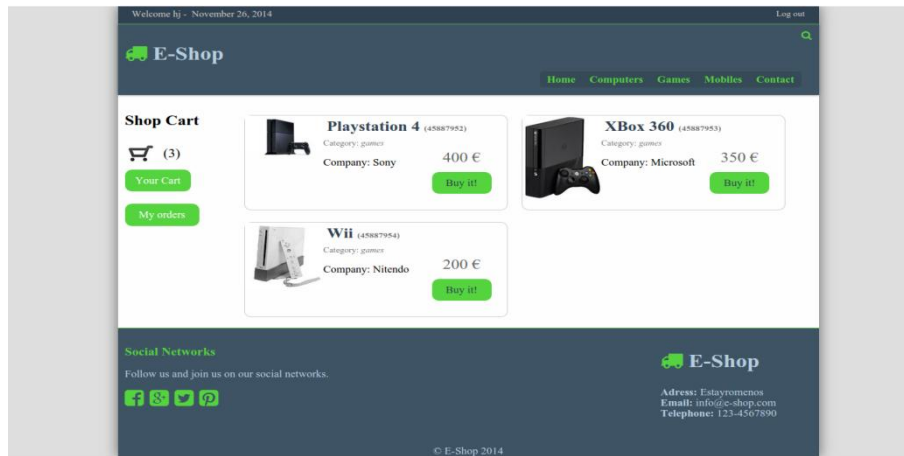


Εικόνα 9: Φόρμα σύνδεσης Πελάτη/ Φόρμα εγγραφής πελάτη

Αναζήτηση Προϊόντων



Εικόνα 10: Φόρμα αναζήτησης προϊόντων

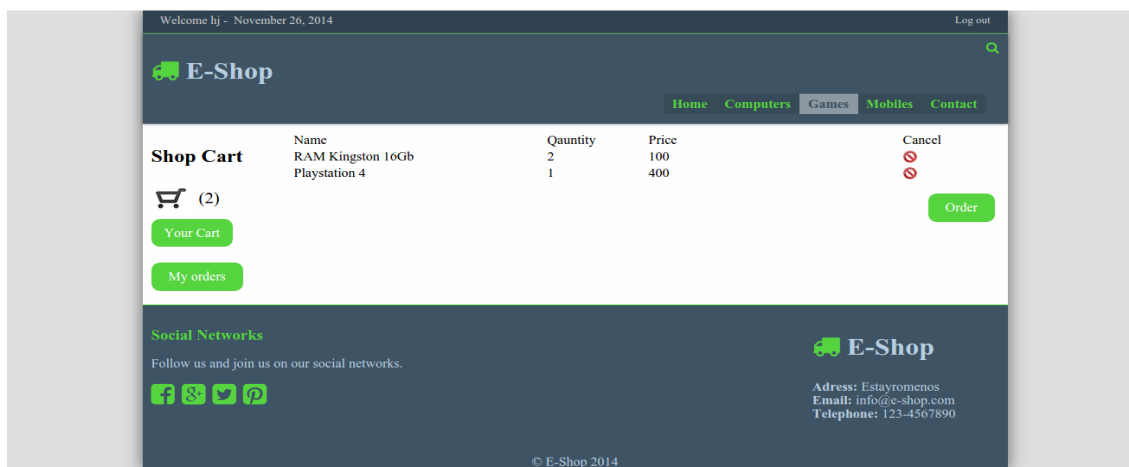


Εικόνα 11: Πλοήγηση κατηγοριών

Καλάθι Αγορών

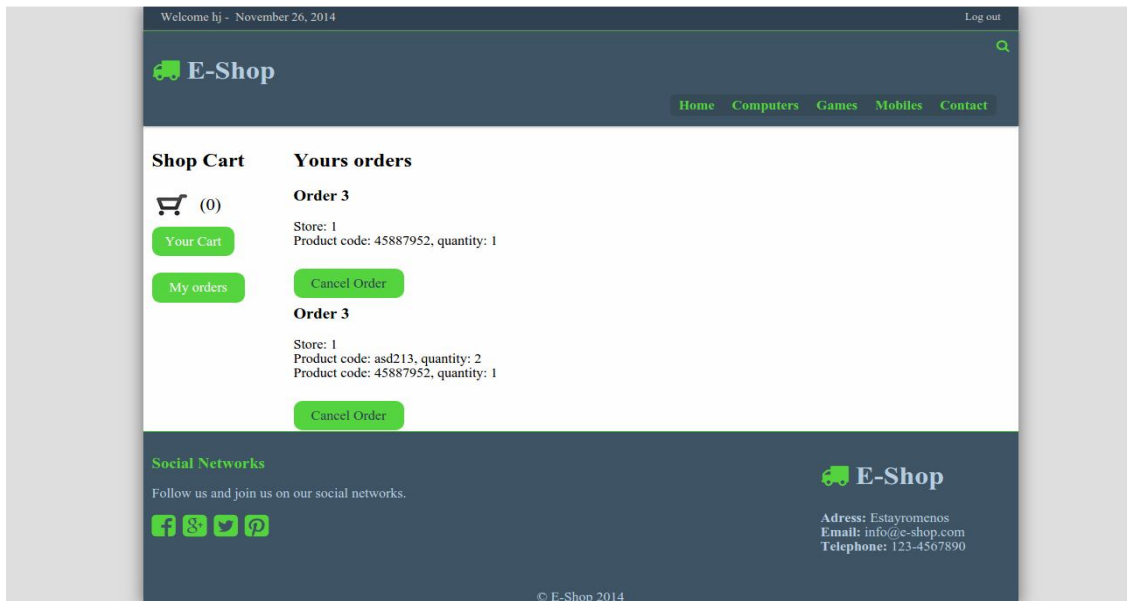


Εικόνα 11: Καλάθι Αγορών

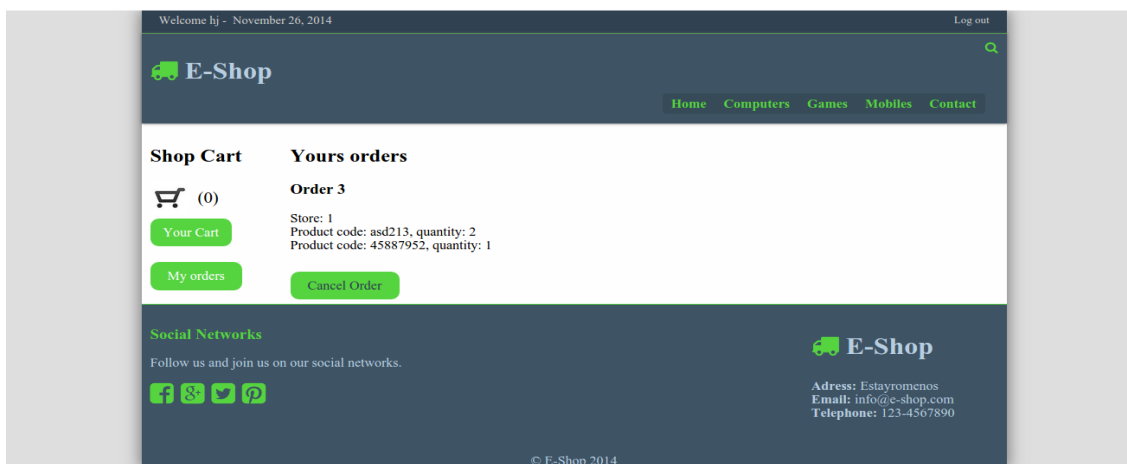


Εικόνα 12: Καλάθι αγορών μετά από αφαίρεση προϊόντος

Προβολή Ιστορικό Παραγγελιών



Εικόνα 13: Προβολή Παραγγελιών Πελάτη









Εικόνα 14: Ακρόρωση Παραγγελίας

Σύνδεση Διαχειριστή

Welcome admin - November 26, 2014 Log out

E-Shop Modify a Product Show Stock Show Orders Show Stores

 <p>Playstation 4 (45887952) Category: games Company: Sony 400 € Edit</p>	 <p>Xbox 360 (45887953) Category: games Company: Microsoft 350 € Edit</p>
 <p>Wii (45887954) Category: games Company: Nintendo 200 € Edit</p>	 <p>RAM Kingston 16Gb (488213) Category: computers Company: Kingston 50 € Edit</p>
 <p>USB hard drive (4814132) Category: computers Company: Sandisk 10 € Edit</p>	 <p>LED 24" HD Monitor (8993423) Category: computers Company: LG 180 € Edit</p>
<p>(45887955) Category: Company: 54 € Edit</p>	<p>(45887956) Category: Company: 48 € Edit</p>
<p>(45887957) Category: Company: 87 € Edit</p>	<p>Samsung S5 (45887958) Category: mobiles Company: Samsung 789 € Edit</p>
<p>iPhone 5s (45887959) Category: mobiles Company: Apple 879 € Edit</p>	<p>(45887960) Category: Company: 15 € Edit</p>
<p>(45887961) Category: Company: 90 € Edit</p>	<p>(45887962) Category: Company: 3 € Edit</p>
<p>(45887963) Category: Company: 64 € Edit</p>	<p>(45887964) Category: Company: 23 € Edit</p>
<p>12 (45887965) Category: Company: € Edit</p>	<p>(45887966) Category: Company: 23 € Edit</p>
<p>(45887967) Category: Company: 4 € Edit</p>	

Social Networks
Follow us and join us on our social networks.

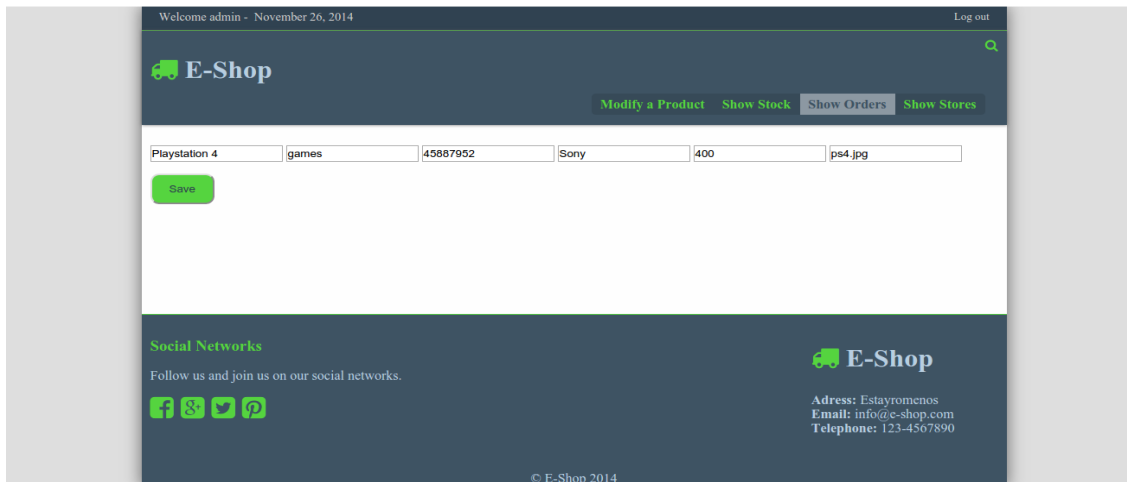
[f](#) [t](#) [g+](#) [p](#)

E-Shop
Address: Εστιατοριου
Email: info@e-shop.com
Telephone: 123-4567890

© E-Shop 2014

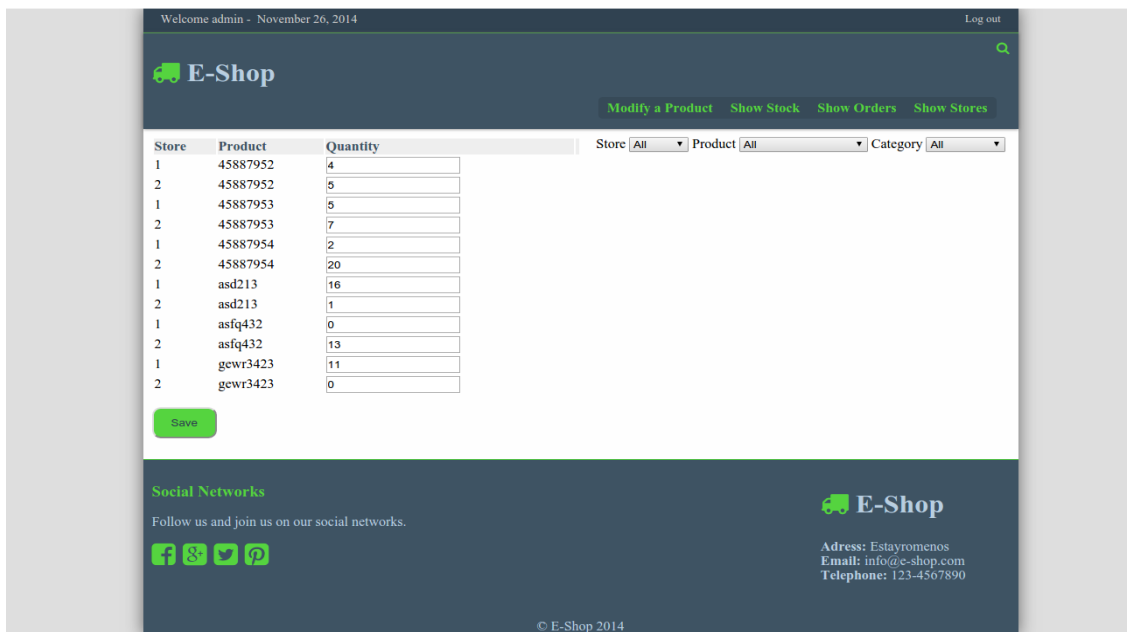
Εικόνα 15: Λίστα επιλογής προϊόντων για τροποποίηση

Τροποποίηση Προϊόντος



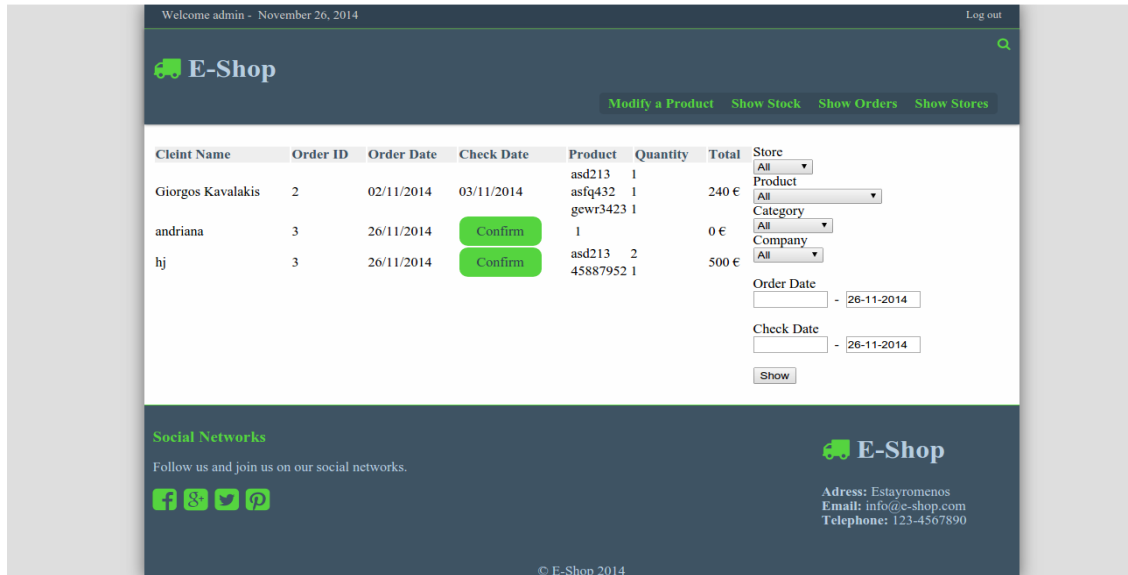
Εικόνα 16: Τροποποίηση Προϊόντος

Τροποποίηση Αποθεμάτων



Εικόνα 17: Επεξεργασία αποθεμάτων

Προβολή και τροποποίηση Παραγγελιών



Welcome admin - November 26, 2014 Log out

E-Shop Modify a Product Show Stock Show Orders Show Stores

Client Name	Order ID	Order Date	Check Date	Product	Quantity	Total	Store
Giorgos Kavalakis	2	02/11/2014	03/11/2014	asd213	1	240 €	All
				asfq432	1		All
				gewr3423	1		All
andriana	3	26/11/2014	<input type="button" value="Confirm"/>		1	0 €	All
hj	3	26/11/2014	<input type="button" value="Confirm"/>	asd213	2	500 €	All
				45887952	1		All

Order Date: - 26-11-2014
Check Date: - 26-11-2014

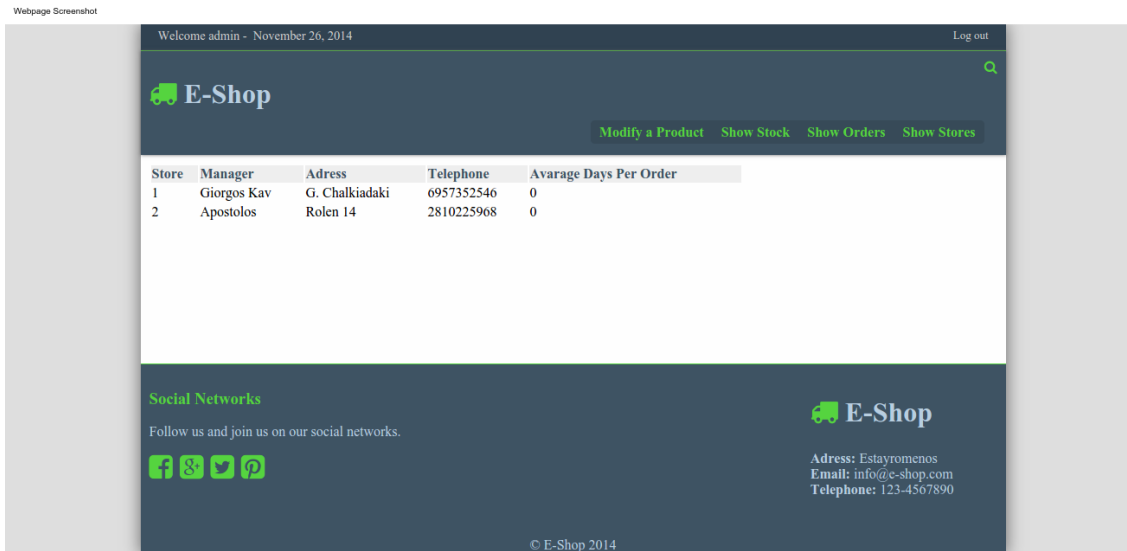
Social Networks
Follow us and join us on our social networks.
[f](#) [g+](#) [t](#) [p](#)

E-Shop
Address: Estayromenos
Email: info@e-shop.com
Telephone: 123-4567890

© E-Shop 2014

Εικόνα 18: Προβολή παραγγελιών από τον Admin

Προβολή διαφόρων αναφορών



Εικόνα 19: Προβολή διαφόρων αναφορών

Αποτελέσματα

Η ιστοσελίδα μας κάνει την χρήση νέων τεχνολογιών όπως είναι η jquery αλλά και html5 και εξωτερικών css αρχείων τα οποία μας δίνουν πολλές δυνατότητες. Με την χρήση της jquery μπορούμε να δημιουργήσουμε μια πολύ ελκυστική σελίδα η οποία κάνει χρήση σύγχρονων μέσων, όπως είναι το jqueryui datpickier που δίνει την επιλογή στον χρήστη να επιλέξει ημερομηνίες όπου εκεί χρειάζεται αλλά γίνεται και με ελάχιστες γραμμές κώδικα, κάνοντας ευκολότερη την υλοποίηση του.

Το ηλεκτρονικό κατάστημα είναι σε θέση να δεχθεί παραγγελίες από επισκέπτες χωρίς να υπάρχει κάποιο πρόβλημα κατά την παραμονή του στην σελίδα, δίνοντας μια ευχάριστη εμπειρία στον χρήστη μας και κάνοντας εύκολη την πλοήγηση του αλλά και την ολοκλήρωση των αγορών.

Συμπεράσματα

Ένα ηλεκτρονικό κατάστημα είναι αρκετά εύκολο στο να κατασκευαστεί από την στιγμή που η ασφάλεια του συστήματος δεν χρειάζεται να είναι ιδιαίτερα αυστηρή. Χρειάζονται βασικές αρχές ασφάλειας ώστε να εξασφαλιστεί η ομαλή λειτουργία του ιστότοπου. Καλό θα ήταν η βάση μας να μην είναι προσβάσιμη από τρίτους καθώς αυτό εγκυμονεί κινδύνους για το ηλεκτρονικό μας κατάστημα. Ένας τρόπος που θα μπορούσε να γίνει αυτό είναι να κρυπτογραφηθεί ή να απαγορευθεί η προβολή του μέσα από το .htaccess αρχείο.

Παρατηρούμε πως η χρήση έτοιμων βιβλιοθηκών όπως αυτή της jquery και jqueryui αλλά και css όπως αυτό του Awesome Fonts μας εξασφαλίζει την ομοιόμορφη παρουσίαση του περιεχομένου του ιστότοπου μας, ανεξαρτήτως του πλοηγητή ή το λειτουργικό σύστημα του επισκέπτη. Η χρήση τους συνιστάτε να γίνεται με κλήση από CDN (Content Delivery Network)

καθώς η παράδοση τους είναι πολύ πιο γρήγορη από την εισαγωγή τους από τον δικό μας server.

Η σωστή δημιουργία της βάσης δεδομένων είναι το άλφα και το ωμέγα για την γρήγορη και ορθή λειτουργία ενός τέτοιου καταστήματος. Όσο περισσότερα λάθη υπάρχουν στην βάση μας τόσο μεγαλύτερα μπορεί να είναι τα λάθη που θα δημιουργηθούν μελλοντικά. Καθώς και σωστή δομή της βάσης είναι κρίσιμη για την ανάπτυξη ενός γρήγορου και καθαρού κώδικα.

Μελλοντικές επεκτάσεις

Μελλοντικά θα μπορούσε να γίνει χρήση του xpath στην PHP ώστε να μπορέσει να αφαιρεθεί ένα μεγάλο κομμάτι κώδικα που αφορά την αναζήτηση και χρήση μέσα στον xml αρχείο. Αυτό πιθανόν θα μείωνε τον απαιτούμενο χρόνο για την ολοκλήρωση κάποιου ερωτήματος προς την βάση αλλά θα μίκραινε και κατά μεγάλο ποσοστό ο κώδικας, κάνοντας τον πιο ευκολοδιάβαστο και κατανοητό προς τρίτους.

Ένα τέτοιο site δίνει πολλές δυνατότητες για επεκτάσεις. Μικρές λειτουργίες που μπορούν να υλοποιηθούν είναι πολλές που μπορούν να διευκολύνουν ακόμα περισσότερο την πλοήγηση του χρήστη. Ταυτόχρονα θα μπορούσε να δημιουργηθεί webservice ώστε να παρέχει και σε άλλα site προϊόντα για πώληση αυξάνοντας το αγοραστικό κοινό και την προβολή των προϊόντων του.

Ο χώρος των ηλεκτρονικών σελίδων είναι ένας χώρος γεμάτος με νέες ιδέες, ένας κλάδος που καθημερινά αναπτύσσεται και δημιουργεί νέες τεχνολογικές δυνατότητες.

Βιβλιογραφία

1. <http://en.wikipedia.org/>
2. <http://php.net/>
3. <http://jquery.com/>
4. <http://fontawesome.github.io/Font-Awesome/>
5. <http://www.w3schools.com/>
6. <http://jqueryui.com/>