

## Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Μηχανικών Πληροφορικής



Πτυχιακή Εργασία

Ηλεκτρονικό Εμπόριο

Δουλγεράκης Ιωάννης(ΑΜ:2407)

Επιβλέπων καθηγητής: Παπαδάκης Νικόλαος

Ημερομηνία παρουσίασης:

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον κύριο Παπαδάκη Νικόλαο για την ανάθεση της πτυχιακής αυτής εργασίας αλλά και της βοήθειάς του καθ' όλη την διάρκεια ανάπτυξής της έως την διεκπεραίωση της.

## **Abstract**

The purpose of this work is to develop a web shop that covers the needs of a cellar which supplies its products through the internet. For that needs we develop an e-Commerce service.

For the development and the design of this page we have used html5 language. Html is an markup language and used for web page development. Also for the product's database creation we have used the xml technology.

Each customer can create his own personal account. He can also connect to our website with it. The account allows customers to add products to their wishlist and other possibilities like special offers or newsletters.

For the need of dynamic content through the php language require a local web server. That need covered by wamp server.

We also used the php language to display the products from the xml database to our webpage. Each of the customers can add the products of his desire to the shopping cart and can promote them to order.

Finally to complete orders we used the PayPal Sandbox. The reason is that PayPal provide a fast and secure way for transactions between the store and the customers.

## Σύνοψη

Σκοπός της εργασίας αυτής είναι η ανάπτυξη μίας ιστοσελίδας ηλεκτρονικού καταστήματος που καλύπτει της ανάγκες μίας κάβας η οποία προμηθεύει τα προϊόντα της μέσω του διαδικτύου. Για τις ανάγκες αυτές έπρεπε να αναπτυχθεί μία υπηρεσία ηλεκτρονικού εμπορίου.

Για την ανάπτυξη και τον σχεδιασμό της σελίδας χρησιμοποιήθηκε η html5 γλώσσα η οποία είναι μια γλώσσα σήμανσης και χρησιμοποιείται ευρέως για την ανάπτυξη web σελίδων.Επίσης για τη δημιουργία της βάσης δεδομένων των προϊόντων που θα εμπορευόταν το κατάστημά μας χρησιμοποιήθηκε η xml τεχνολογία.

Κάθε πελάτης μπορεί να δημιουργήσει τον προσωπικό του λογαριασμό και να συνδεθεί στην σελίδα μας. Ο λογαριασμός δίνει την δυνατότητα στους πελάτες να προσθέσουν προϊόντα στη λίστα τους αλλά και άλλες δυνατότητες όπως ειδικές προσφορές και ενημερωτικά κουπόνια.

Για την ανάγκη δυναμικού περιεχομένου που ικανοποιείται μεσω της php απαιτείται ένας τοπικός web server μέσω του οποίου λειτουργεί η php γλώσσα. Η ανάγκη αυτή καλύπτεται απο τον wamp server.

Για την εμφάνιση των προϊόντων απο τη βάση στη σελίδα μας χρησιμοποιήθηκε η php γλώσσα.Κάθε πελάτης μπορεί να προσθέσει τα προϊόντα της επιθυμίας του στη κάρτα αγορών και να τα προωθήσει προς παραγγελία.

Τέλος για την ολοκλήρωση των παραγγελιών χρησιμοποιήθηκε η σελίδα της PayPal με σκοπό τις γρήγορες αλλά και ασφαλείς συναλλαγές μεταξύ των πελατών και του καταστήματος.

## Πίνακας Περιεχομένων

<i>Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης</i>	<b>1</b>
<i>Ευχαριστίες</i>	<b>2</b>
<i>Abstract</i>	<b>3</b>
<i>Σύνοψη</i>	<b>4</b>
<i>Πίνακας Περιεχομένων</i>	<b>5</b>
<i>Πίνακας Εικόνων</i>	<b>7</b>
<b>1. Εισαγωγή</b>	<b>9</b>
<b>1.1 Περίληψη</b>	<b>9</b>
<b>1.2 Στόχοι</b>	<b>10</b>
<b>1.3 Δομή Εργασίας</b>	<b>10</b>
<b>2. Χρονοδιάγραμμα</b>	<b>11</b>
<b>3. Μεθόδους και Scripts</b>	<b>12</b>
<b>4. Ανάλυση Τεχνολογιών Εφαρμογής</b>	<b>14</b>
<b>4.1 Σχεδιασμός Ιστοσελίδας</b>	<b>14</b>
<b>4.2 Ανάπτυξη Δυναμικού Περιεχομένου</b>	<b>15</b>
<b>4.3 Wamp Server</b>	<b>16</b>
<b>4.4 Τεχνολογία xml</b>	<b>17</b>
<b>5. Υλοποίηση Εφαρμογής</b>	<b>18</b>
<b>5.1 Ανάπτυξη του Login/Registration συστήματος</b>	<b>18</b>
5.1.1 Υλοποίηση Δημιουργίας Λογαριασμού	20
5.1.2 Υλοποίηση Σύνδεσης	21
5.1.3 Έλεγχος Σύνδεσης και Αποσύνδεση.	23
<b>5.2 Δημιουργία της Βάσης Δεδομένων</b>	<b>24</b>
5.2.1 XML ανάπτυξη βάσης	24
5.2.2 XML-Schema Validation	25
<b>5.3 Εμφάνιση της Βάσης Δεδομένων</b>	<b>26</b>
5.3.1 Διάβασμα του Xml από την Php	26
5.3.2 Εμφάνιση του Xml εγγράφου σε Php	26
5.3.3 Αναζήτηση Προϊόντων	28
5.3.4 Λίστα επιθυμιών	29
5.3.5 Δικαιώματα του Διαχειρηστή	30

<b>5.4</b>	<b>Προσθήκη στη Κάρτα Αγορών</b>	<b>31</b>
5.4.1	Αίτηση Κωδικού	31
5.4.2	Δημιουργία της κάρτας Αγορών	32
5.4.3	Συναρτήσεις της κάρτας Αγορών	34
5.4.4	Εμφάνιση της κάρτας Αγορών	37
<b>5.5</b>	<b>PayPal Sandbox</b>	<b>42</b>
5.5.1	Δημιουργία εικονικών λογαριασμών	42
5.5.2	Αγορά μέσω της PayPal	44
5.5.3	Ιστορικό Συναλλαγών στην PayPal	44
<b>6.</b>	<b>Αποτελέσματα</b>	<b>45</b>
<b>6.1</b>	<b>Ωφέλη</b>	<b>47</b>
<b>6.2</b>	<b>Δυσκολίες</b>	<b>48</b>
<b>6.3</b>	<b>Μελλοντικές επεκτάσεις</b>	<b>48</b>
	<b>Bibliography</b>	<b>49</b>
	<b>Παράρτημα</b>	<b>50</b>
	Παρουσίαση(PowerPoint)	51
	Περίληψη	51

## Πίνακας Εικόνων

1 Task	11
2 Χρονοδιάγραμμα	11
	14
3 Κεντρική Σελίδα	14
4 Wamp	16
5 Φόρμα Λογαριασμού	18
6 Form html	19
7 Register Function	20
8 File Counter	21
9 Login Function	22
10 Check Function	23
11 Logout	24
12 Xml Db	24
13 Xml Schema	25
14 Define Xml	26
15 Get Catalog	26
16 Εμφάνιση Καταλόγου	27
17 Εμφάνιση καταλογου Β	27
18 Search Function	28
19 Search Page	28
20 Wishlist href	29
21 AddToWishlist	29
22 ChangeQuantity	30
23 Υπερσύμδεσμος a href	31
24 Δημιουργία Κάρτας Αγορών	32
25 AddItem	33
26 Product_Exist	34
27 Get_item_Cost	34
28 Get_item_Qty	35
29 GetItemCost	35
30 getShippingCostFor	35
31 GetSubTotals	36
32 GetItemShippingCost	36
33 GetTotal	36
34 GetItemQuantity	36
35 EmptyCart	37
36 ShoppingCart	37
37 ShopCartRow Render	38
38 Render Shippings	39
39 Render Totals	39

40 Render Shopping Cart-----	40
41 Καθορισμός στοιχείων PayPal -----	41
42 Render Checkout-----	41
43 PayPal Sandbox-----	42
44 Εικονικοί λογαριασμοί-----	42
45 PayPal code-----	43
46 Bank account-----	43
47 Sandbox Payment-----	44
48 Στοιχεία καταστήματος-----	45
49 Ιστορικό παραγγελιών-----	45
50 Λεπτομέριες παραγγελίας-----	46
51 Δυνατότητα ακύρωσης-----	47



## 1. Εισαγωγή

Τα τελευταία χρόνια έχει αναπτυχθεί μια υπηρεσία για την διευκόλυνση των καταναλωτών αλλά και εμπόρων. Η υπηρεσία αυτή ονομάζεται ηλεκτρονικό εμπόριο μέσω της οποίας πωλητής και αγοραστής μπορούν να συναλλάσσονται μέσω του διαδικτύου.

Με τον όρο του ηλεκτρονικού εμπορίου αναφερόμαστε σε μια διαδικασία συναλλαγής η ανταλλαγής πίσω από έναν ηλεκτρονικό υπολογιστή διαφόρων προϊόντων η υπηρεσιών, η οποία συναλλαγή πραγματοποιείται μέσω του διαδικτύου. Με το πέρασμα των χρόνων η ταχύτητες με τις οποίες εξελίσσεται η υπηρεσία αυτή παγκοσμίως είναι ραγδαίες και υπάρχει αξιοπιστία.

Ο πιθανός καταναλωτής πλέον μπορεί να ανατρέξει σε οποιαδήποτε ιστοσελίδα της αρέσκειας του και να επιλέξει απο μια σειρά εκατομμυρίων προϊόντων η υπηρεσιών προς παραγγελία σε απομακρυσμένα καταστήματα του εσωτερικού ή και του εξωτερικού.

Επίσης με τη βοήθεια του ηλεκτρονικού εμπορίου μπορεί ο καταναλωτής να κάνει συγκρίσεις τιμών για ένα συγκεκριμένο προϊόν ή υπηρεσία μέσα απο μία σειρά δεκάδων καταστημάτων.

Το ηλεκτρονικό εμπόριο πέρα απο τις συναλλαγές μεταξύ επιχειρήσεων και καταναλωτών εξυπηρετεί και άλλους σκοπούς.Μερικά είδη του εμπορίου αυτού είναι μεταξύ καταναλωτών/πολιτών προς κυβερνητικούς φορείς(C2G), εμπόριο επιχειρήσεων προς κυβερνητικούς φορείς(B2G), όπως επίσης και επιχειρήσεις προς άλλες επιχειρήσεις.

### 1.1 Περίληψη

Σκοπός της εργασίας αυτής είναι η δημιουργία και η ανάπτυξη μιας ιστοσελίδας ώστε να μπορέσουν να καλυφθούν οι ανάγκες μιας κάβας η οποία επιθυμεί να εμπορεύεται τα προϊόντα της ηλεκτρονικά.

Για της ανάγκες της κάβας λοιπόν έπρεπε να αναπτυχθεί μια υπηρεσία ηλεκτρονικού εμπορίου ώστε να μπορεί να προμηθεύει τους πελάτες της με προϊόντα της επιθυμίας τους πίσω απο έναν ηλεκτρονικό υπολογιστή μέσω του διαδικτύου.

Για την ανάπτυξη της ιστοσελίδας αυτής χρησιμοποιήθηκαν τεχνολογίες όπως HTML5,CSS3,PHP,JS και επίσης η τεχνολογία XML για την αποθήκευση δεδομένων όπως λογαριασμοί χρηστών αλλά και η βάση δεδομένων που αποτελείται απο τα προϊόντα που παρέχει η εταιρεία. Με αυτό το τρόπο ο πελάτης μπορεί να ικανοποιήσει την παραγγελία του μέσω του διαδικτύου ανατρέχοντας μέσα απο μια σειρά προϊόντων που προμηθεύει η συγκεκριμένη κάβα.

Τέλος για την ολοκλήρωση των παραγγελιών χρησιμοποιήθηκε η σελίδα της PayPal με σκοπό τις γρήγορες αλλά και ασφαλείς συναλλαγές μεταξύ των πελατών και του καταστήματος.

## 1.2 Στόχοι

Στόχοι της ανάπτυξης της εφαρμογής αυτής είναι η διευκόλυνση των πελατών της συγκεκριμένης ή οποιασδήποτε εταιρείας να μπορούν να ερευνούν, να αναζητούν, να συναλλάσσονται ή και να συγκρίνουν τιμές προϊόντων ή υπηρεσιών γρήγορα ανατρέχοντας απλά στο διαδίκτυο.

## 1.3 Δομή Εργασίας

Στα κεφάλαια που θα ακολουθήσουν αναλύεται η ανάπτυξη του ηλεκτρονικού αυτού καταστήματος καθώς και όλες οι τεχνολογίες που απαιτήθηκαν για αυτό. Η ανάπτυξη του design της ιστοσελίδας μέσω της html, όπως επίσης και η δημιουργία της βάσης των προϊόντων του καταστήματος με τη βοήθεια της xml. Στη συνέχεια φαίνεται αναλυτικά ο τρόπος μέσω της php με τον οποίο οι χρήστες του καταστήματος μπορούν να δημιουργήσουν τον λογαριασμό τους και να συνδεθούν με αυτόν καθώς και να προσθέσουν τα προϊόντα στο καλάθι τους ή και στη λίστα επιθυμιών του.

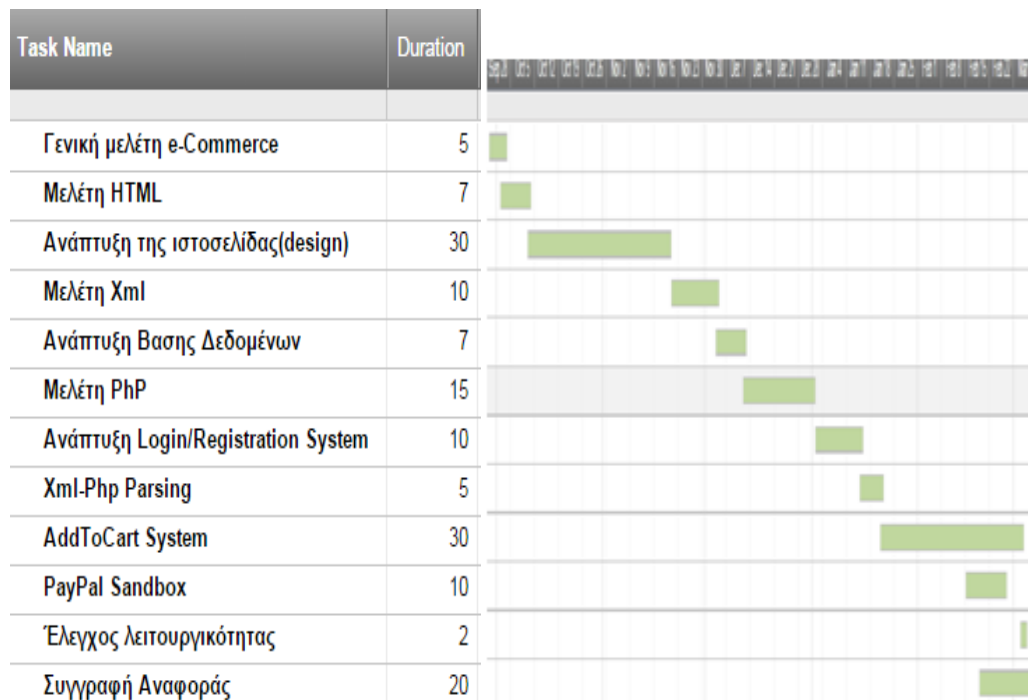
Αναφέρεται η ανάγκη για έναν διαχειριστή ο οποίος θα μπορεί να βλέπει όλους του λογαριασμούς του κάθε χρήστη καθώς και να αλλάζει τα αποθέματα του κάθε προϊόντος.

Τέλος η μεταφορά της κάρτας αγορών στη σελίδα της PayPal και η ολοκλήρωση των παραγγελιών από τους καταναλωτές. Η ανάγκη της PayPal δείχνει απαραίτητη καθώς απαιτείται ασφάλεια αλλά και ταχύτητα συναλλαγών.

## 2. Χρονοδιάγραμμα

Task Name	Duration
Γενική μελέτη e-Commerce	5
Μελέτη HTML	7
Ανάπτυξη της ιστοσελίδας(design)	30
Μελέτη Xml	10
Ανάπτυξη Βασης Δεδομένων	7
Μελέτη Php	15
Ανάπτυξη Login/Registration System	10
Xml-Php Parsing	5
AddToCart System	30
PayPal Sandbox	10
Έλεγχος λειτουργικότητας	2
Συγγραφή Αναφοράς	20

1 Task



2 Χρονοδιάγραμμα

### 3. Μεθόδους και Scripts

Μέθοδοι και Script που χρησιμοποιήθηκαν κυρίως για την ανάπτυξη του δυναμικού περιεχομένου μέσω της php.

#### *if(isset)*

Η `isset` είναι μία συνάρτηση της php η οποία ελέγχει εάν ισχύει μία συνθήκη. Δουλειά της είναι να επιστρέφει `true` όταν ακολουθείται από μία συνθήκη στην περίπτωση που η συνθήκη αυτή ισχύει.

#### *die*

Η `die` είναι κ αυτή συνάρτηση της php όπου χρησιμοποιείται στο τέλος ενός script και δηλώνει το τερματισμό/κλείσιμο ενός script.

#### *session*

Το `php session` είναι ένας μηχανισμός που μπορεί να κρατά πληροφορίες όπως για παράδειγμα να ελέγχει τα στοιχεία ενός πελάτη που έχει επισκεφτεί τη σελίδα μας και να κρατά πληροφορίες για αυτόν. Κάθε φορά που θέλουμε να κρατάμε πληροφορία για κάποιον λογαριασμό ή για κάποιο προϊόν πχ. στην κάρτα αγορών μας χρησιμοποιούμε ένα `session` το οποίο κρατά τις πληροφορίες αυτές προσωρινά στον server. Το `session` χρησιμοποιεί την τεχνολογία των cookies. Για να ξεκινήσει ένα `session` χρησιμοποιείται η εντολή `session_start()` και για να τερματιστεί η `destroy_session()`.

#### *serialize*

Η συνάρτηση `serialize` χρησιμοποιείται από την php με σκοπό να δημιουργήσει την αποθηκεύσιμη αναπαράσταση μίας αξίας. Χρησιμοποιείται για να αποθηκεύει php τιμές χωρίς να χάνεται ο τύπος των τιμών αυτών. Ο λόγος της χρήσης της είναι να αποθηκεύεται μία λίστα (σαν συμπίεση) τιμών σε ένα πίνακα εκτός της php ή υπό τη μορφή bits ή σε κάποιο `memory buffer` συμπιέζοντας έτσι το μέγεθός τους.

#### *unserialize*

Επειδή με τη `serialize` τα στοιχεία αυτά αποθηκεύονται όπως αναφέραμε εκτός php και η php δουλεύει μόνο με php τιμές και παραστάσεις, τα στοιχεία που αποθηκεύτηκαν με την `serialize` επιστρέφονται πάλι με την `unserialize` υπό μορφή php αντικείμενα.

### *simpleXMLElement*

Η μέθοδος αυτή χρησιμοποιείται και μας επιτρέπει να χειριστούμε εύκολα δεδομένα απο ένα xml έγγραφο. Μπορούμε εύκολα να πάρουμε το όνομα ενός στοιχείου ή τα χαρακτηριστικά και το περιεχόμενο των στοιχείων ενός εγγράφου εάν γνωρίζουμε τη δομή του εγγράφου αυτού. Στην ουσία μετατρέπει το xml έγγραφό μας σε έναν πίνακα αντικειμένων php.

### *file\_get\_contents()*

Από τη στιγμή που θα έχουμε πρόσβαση στο xml αρχείο μας μπορούμε να διαβάσουμε τα περιεχόμενά του. Τα περιεχόμενα αυτά μπορεί να είναι τα στοιχεία ενός xml κόμβου.

## 4. Ανάλυση Τεχνολογιών Εφαρμογής

Για την ανάπτυξη της ιστοσελίδας χρησιμοποιείται η εφαρμογή notepad++ για την δημιουργία του κώδικα. Επίσης χρειάστηκε και ένας web server(wamp) ώστε να τρέξει η σελίδα μας στον apache server. Απο τη στιγμή που η ιστοσελίδα μας είναι τύπου e-commerce(ηλεκτρονικό εμπόριο) η χρήση του wamp server είναι απαραίτητη αφού θα δουλέψει σε php γλώσσα μιας και το περιεχόμενο της θα είναι δυναμικό.

### 4.1 Σχεδιασμός Ιστοσελίδας

Αρχικά έπρεπε να αναπτυχθεί και να υλοποιηθεί ο σχεδιασμός της ιστοσελίδας και γενικότερα το θέμα πάνω στο οποίο θα δούλευε η παρουσίαση. Για τον σχεδιασμό της σελίδας λοιπόν χρησιμοποιήθηκε η HTML5 γλώσσα σήμανσης όπου δημιουργήθηκε το περιβάλλον που θα φιλοξενούσε η σελίδα μας σε συνδυασμό με την CSS3 για το στυλιστικό κυρίως κομμάτι.

Η html είναι μία γλώσσα σήμανσης και είναι η κύρια γλώσσα για την δημιουργία ιστοσελίδων. Η γλώσσα αυτή αναπτύσσεται υπό μορφή στοιχείων μέσα σε ετικέτες τα οποία στοιχεία μπορεί να είναι κείμενο, εικόνες, πίνακες κλπ. Οι ετικέτες στην html είναι προκαθορισμένες, δηλαδή για κάθε χρήση απαιτείται συγκεκριμένη ετικέτα και σχεδόν κάθε ετικέτα που ανοίγει πρέπει να κλείνει με το ίδιο όνομα ετικέτας.

Η ιστοσελίδα μας πέρα από την κεντρική αποτελείται επιπλέον από σελίδες που περιέχουν τα προϊόντα που προμηθεύεται η εταιρία καθώς επίσης και άλλες υπηρεσίες όπως την δυνατότητα δημιουργίας λογαριασμού και προσωπική λίστα επιθυμητών προϊόντων.



Κάθε σελίδα είναι χωρισμένη σε τρία κύρια μέρη που αποτελούν το σύνολο της. Τα βασικά αυτά μέρη της είναι :

- Το **header**
- Το **main section**
- Το **footer**

Το **header** όπου αναγράφεται αρχικά το όνομα της ιστοσελίδας μας, στοιχεία για την εταιρία όπως διευθύνσεις και τηλέφωνα καθώς επίσης λογαριασμός και καλάθι αγορών.

Το **main section** το οποίο είναι το κεντρικό τμήμα της σελίδας μας και περιέχει το κύριο θέμα της κάθε σελίδας ανάλογα με τα περιεχόμενα της.

Και τέλος το **footer** όπου αναφέρονται πληροφορίες σχετικά με την εταιρεία, τα καταστήματα όπως και με το λογαριασμό του κάθε χρήστη, τρόπους επαφής και επικοινωνίας. Τα τμήματα **header** και **footer** διατηρούνται όπως είναι σε κάθε σελίδα του ιστότοπού μας.

Για τη μορφοποίηση της σελίδας μας χρησιμοποιήθηκε η css3 η οποία ορίζει την εμφάνιση των στοιχείων της html αλλά και την διάταξη τους κάνοντας πιο προσιτή την εμφάνιση του html εγγράφου.

Γενικότερα ο σχεδιασμός μιας ιστοσελίδας είναι ο σημαντικότερος παράγοντας στη παρουσίαση της επιχείρησης στον χώρο του διαδικτύου. Απο αισθητικής πλευράς πρέπει να κάνει το κατάστημα πιο φιλικό προς το επισκέπτη κερδίζοντας το πλεονέκτημα της πρώτης εντύπωσης. Επίσης όσο πιο απλό είναι το περιβάλλον του καταστήματος καταστεί εύκολη την περιήγηση του επισκέπτη κάνοντας τη ηλεκτρονική επέκταση του καταστήματος μας ακόμη πιο φιλική.

## 4.2 Ανάπτυξη Δυναμικού Περιεχομένου

Η ανάγκη για δυναμικό περιεχόμενο μιάς και έχουμε να κάνουμε με e-shop ήταν αυτή που έκανε την php και τον wamp server απαραίτητους και σημαντικούς παράγοντες στην ανάπτυξη της ιστοσελίδας μας. Η php γνωστό χρησιμοποιείται για σελίδες web με δυναμικό περιεχόμενο. Πρέπει να περνούν και να επεξεργάζονται μέσα απο έναν διακομιστή του Παγκόσμιου Ιστού όπως είναι ένας apache server. Το περιεχόμενο της php σελίδας πρέπει να μπορεί να παραχθεί σε πραγματικό χρόνο και ανα πάσα στιγμή κατά την επίσκεψη μας στην σελίδα. Τα αποτελέσματα επιστρέφουν και εμφανίζονται στους επισκέπτες με την μορφή της HTML.

Η php γλώσσα χρησιμοποιείται τόσο για την ανάπτυξη του συστήματος λογαριασμού όσο και για την προσθήκη των προϊόντων στο καλάθι αλλά και την εμφάνιση τους.

### 4.3 Wamp Server

Όπως αναφέραμε, για την ανάγκη δυναμικού περιεχομένου χρειάστηκε να τρέξει ένας τοπικός web server στον υπολογιστή μας ώστε να δουλεύουν οι διαδικασίες της php σελίδας μας δυναμικά. Για την ανάγκη αυτή χρησιμοποιήθηκε ο Wamp Server. Η ονομασία του είναι από τα αρχικά Windows, Apache, mySQL, php. Ο apache είναι το σύστημα που τρέχει ο server μας.



#### 4 Wamp

Γενικά απευθύνεται σε όσους επιθυμούν να έχουν έναν web server εύκολα και γρήγορα. Είναι ένας πολύ εύκολος και βολικός τρόπος να ξεκινήσει να δημιουργεί κανείς ένα δικό του site δυναμικά. Κυριώς είναι αναγκαίος για την δημιουργία με χρήση php όπου απαιτείται δυναμικό περιεχόμενο για να μπορούν να γίνονται αλλαγές κατά τη διάρκεια που ήδη τρέχει η σελίδα μας στον τοπικό server.



#### **4.4 Τεχνολογία xml**

Στις μέρες μας η HTML είναι η κύρια γλώσσα για την ανάπτυξη και δημιουργία ιστοσελίδων. Στον Παγκόσμιο Ιστό, τα πρότυπα καθορίζονται από το οργανισμό W3C.

Η ανάπτυξη της XML είχε σκοπό να καλύψει κάποιες ελλείψεις της HTML. Η xml είναι και αυτή γλώσσα σήμανσης και χρησιμοποιεί ετικέτες. Στην xml όλες οι ετικέτες πρέπει να κλείνουν και επίσης τα xml έγγραφα είναι εύκολα προσπελάσιμα από τους υπολογιστές, αφού υπάρχει περιγραφή όλων των πληροφοριών από την ετικέτα. Αποτελεί έναν εύκολο τρόπο για τη δημιουργία μίας βάσης δεδομένων που μπορεί να χρησιμοποιηθεί σε ένα οποιοδήποτε ηλεκτρονικό κατάστημα.

## 5. Υλοποίηση Εφαρμογής

### 5.1 Ανάπτυξη του Login/Registration συστήματος

Για την καλύτερη εξυπηρέτηση των πελατών της εταιρίας αναπτύχθηκε το σύστημα λογαριασμού όπου ο κάθε πελάτης μπορεί να δημιουργήσει τον προσωπικό του λογαριασμό στη σελίδα και να συνδεθεί με αυτόν.

Η χρήση λογαριασμού προσφέρει στους πελάτες τη δυνατότητα να προσθέτουν προϊόντα στην wishlist τους καθώς επίσης και να λαμβάνουν προσφορές σχετικά με προϊόντα. Για την εταιρεία είναι ένας εύκολος τρόπος να πιστοποιεί τα στοιχεία των πελατών της όπως διευθύνσεις, τηλέφωνα και email και να μπορεί να επικοινωνήσει μαζί τους για οποιονδήποτε λόγο.

**Δημιουργία Λογαριασμού**

Όνομα

Επώνυμο

Διευθυνση

Email

Τηλεφωνο

Username

Password

Confirm Password

**Σύνδεση**

Username

Password

#### 5 Φόρμα Λογαριασμού

Για τη δημιουργία λογαριασμού ο χρήστης απαιτείται να συμπληρώσει το ονοματεπώνυμο του, τηλέφωνο, διεύθυνση, mail καθώς επίσης ένα username και password. Όλα τα στοιχεία της φόρμας είναι απαιτούμενα για την δημιουργία του λογαριασμού. Για να συνδεθεί ο χρήστης αφού έχει δημιουργήσει τον προσωπικό του λογαριασμό απαιτείται να συμπληρώσει το username και το password του.

Η φόρμα έχει αναπτυχθεί σε κώδικα html5 και όπως φαίνεται παρακάτω με το όρο required δηλώνουμε ότι πρέπει να συμπληρωθούν όλα τα στοιχεία για τη δημιουργία του λογαριασμού.

```
<form class="RegisterContentt" method="post" action="">
  <table class="format">
    <thead>
      <tr>
        <th><h3>Δημιουργία Λογαριασμού</h3></th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>
          <input class="cName" type="text" name="firstname" maxlength="10" required placeholder="Όνομα" />
        </td>
      </tr>
      <tr>
        <td>
          <input class="cName" type="text" name="lastname" maxlength="15" required placeholder="Επώνυμο" />
        </td>
      </tr>
      <tr>
        <td>
          <input class="cName" type="text" name="address" maxlength="20" required placeholder="Διεύθυνση" />
        </td>
      </tr>
      <tr>
        <td>
          <input class="cName" type="email" name="email" maxlength="20" required placeholder="Email" />
        </td>
      </tr>
      <tr>
        <td>
          <input class="cName" type="text" name="phone" maxlength="15" required placeholder="Τηλέφωνο" />
        </td>
      </tr>
      <tr>
        <td>
          <input class="cName" type="text" name="username" maxlength="15" required placeholder="Username" />
        </td>
      </tr>
      <tr>
        <td>
          <input class="cName" type="password" name="password" maxlength="15" required placeholder="Password" />
        </td>
      </tr>
      <tr>
        <td>
          <input class="cName" type="password" name="c_password" maxlength="15" required placeholder="Confirm Password"/>
        </td>
      </tr>
    </tbody>
  </table>
</form>
```

#### 6 Form html

Στη φόρμα μας χρησιμοποιείται η μέθοδος post μιάς και θέλουμε να λαμβάνουμε δεδομένα απο τον χρήστη/πελάτη μας και να στέλνονται στον εξυπηρετητή. Η μέθοδος αυτή είναι και ο ασφαλέστερος τρόπος να στέλνονται ευαίσθητα δεδομένα όπως για παράδειγμα κωδικοί πρόσβασης αλλά και άλλα δεδομένα.

Για την λειτουργικότητα του λογαριασμού θα πρέπει το περιεχόμενο της σελίδας μας να είναι δυναμικό και για τον λόγο αυτό χρειάζεται η php γλώσσα. Επίσης για την αποθήκευση των στοιχείων του χρήστη χρησιμοποιείται η xml τεχνολογία.

### 5.1.1 Υλοποίηση Δημιουργίας Λογαριασμού

Όπως αναφέρθηκε παραπάνω ο κάθε χρήστης μπορεί να δημιουργήσει τον προσωπικό του λογαριασμό. Για τη δημιουργία λογαριασμού χρησιμοποιήθηκε μία συνάρτηση η **register function**

```

<?php
include 'file_counter.php';
$errors= array();
if(isset($_POST['register'])) {
    $counter = iterator_count($fi)-2;
    $username = preg_replace ('/[^A-Za-z]/', '', $_POST['username']);
    $email = $_POST['email'];
    $password = $_POST['password'];
    $c_password = $_POST['c_password'];
    $address = $_POST['address'];
    $phone = $_POST['phone'];
    $firstname = $_POST['firstname'];
    $lastname = $_POST['lastname'];

    if(file_exists('users/' . $username . '.xml')) {
        $errors[] = 'Username already exists';
    }
    if($password != $c_password) {
        $errors[] = 'Passwords not match';
    }
    if(count($errors)== 0) {
        $counter++;
        $xml = new SimpleXMLElement('<user></user>');
        $xml->addChild('id', $counter);
        $xml->addChild('password', md5($password));
        $xml->addChild('email', $email);
        $xml->addChild('username', $username);
        $xml->addChild('firstname', $firstname);
        $xml->addChild('lastname', $lastname);
        $xml->addChild('address', $address);
        $xml->addChild('phone', $phone);
        $xml->addChild('admin', '0');

        $xml->asXML('users/' . $username . '.xml');
        header('Location: index.php');
        die;
    }
}
?>

```

7 Register Function

Με τη χρήση της php ο χρήστης συμπληρώνει τα στοιχεία του στη φόρμα τα οποία μέσω της php αποθηκεύονται σε μεταβλητές με τη μέθοδο **post**. Οι μεταβλητές αυτές αποθηκεύονται προσωρινά στον server και αποτελούν τα στοιχεία του πελάτη.

Οι λογαριασμοί των χρηστών αποθηκεύονται ως αρχεία xml σε έναν φάκελο με το όνομα users. Για μεγαλύτερη ασφάλεια ο λογαριασμός του κάθε χρήστη αποτελεί ένα ξεχωριστό αρχείο xml. Αφού λοιπόν τα στοιχεία του χρήστη αποδοθούν στις μεταβλητές και με την προϋπόθεση ότι δεν μπορεί να υπάρχουν περισσότεροι του ενός χρήστες με ίδιο username, γίνεται έλεγχος στον φάκελο users για τυχόν αρχείο με όνομα ίδιο με αυτό της μεταβλητής username. Με αυτό τον τρόπο βλέπουμε αν υπάρχει ήδη χρήστης με ίδιο username και ενημερώνει ανάλογα τον πελάτη. Επίσης ελέγχεται κατά πόσο ταιριάζουν οι κωδικοί πρόσβασης.

Από τη στιγμή που θα περάσει τους ελέγχους και δεν θα παρουσιαστεί κανένα σφάλμα ξεκινάει η διαδικασία δημιουργίας του αρχείου xml. Η διαδικασία αυτή ξεκινά με τη συνάρτηση της php, **simpleXMLElement** η οποία είναι απαραίτητη για τη δημιουργία αρχείων xml. Μέσα στις παρενθέσεις δηλώνεται όνομα του root node. Κατόπιν δημιουργούνται τα στοιχεία που θα περιέχει το xml αρχείο μας όπως id, username, όνομα χρήστη κλπ. τα οποία παίρνουν την πληροφορία από τις μεταβλητές αντίστοιχα που καταχωρήθηκαν παραπάνω. Επίσης το σύστημα υποστηρίζει και έναν administrator χρήστη το οποίο δηλώνεται με 0 ή 1. Τέλος αποθηκεύεται το αρχείο xml με όνομα το username του χρήστη.

Στόχος μας ήταν το id του κάθε χρήστη να καταχωρείται αυτόματα από το σύστημα και για τον λόγο αυτό συμπεριλήφθηκε μια μέθοδος με το όνομα **file counter**.

```
<?php $fi = new FilesystemIterator('users/', FilesystemIterator::SKIP_DOTS);
```

#### 8 File Counter

Όπως αναφέρθηκε και παραπάνω για μεγαλύτερη ασφάλεια ο κάθε λογαριασμός αποτελεί ένα ξεχωριστό αρχείο, οπότε δουλειά της μεθόδου είναι να μετρά τα αρχεία που βρίσκονται στον φάκελο users και να τα αποδίδει σε μία μεταβλητή. Τέλος η μεταβλητή αυτή είναι ένας ακέραιος που αποτελεί το id του χρήστη.

### 5.1.2 Υλοποίηση Σύνδεσης

Αφού λοιπόν ο πελάτης έχει δημιουργήσει με επιτυχία το λογαριασμό του μπορεί πλέον να συνδεθεί στη ιστοσελίδα. Για να συνδεθεί απαιτείται να πληκτρολογήσει το username και το password του.

Για τη σύνδεση αλλά και την διατήρηση του λογαριασμού του χρήστη όταν θα ανατρέχει σε άλλες σελίδες του site μας απαιτείται κάποια διαδικασία μέσω της php που εξυπηρετεί το σκοπό μας. Συμαντικό ρόλο στην διαδικασία αυτή παίζουν τα *php sessions*.

Τα **session** στην php είναι ένας σημαντικός τρόπος να αποθηκεύονται πληροφορίες (σε μεταβλητές) και οι πληροφορίες αυτές κρατούνται εκεί και μπορούν να χρησιμοποιηθούν ξανά σε όλες τις σελίδες, εφόσον το session είναι ακόμη ανοιχτό. Σε αντίθεση με τα cookies, ένα **session** κρατά τις πληροφορίες αυτές στον server.

```

<?php
$error=false;
if(isset($_POST['login'])){
    $username = preg_replace('/[^A-Za-z]/', '', $_POST['username']);
    $password =md5($_POST['password']);
    if(file_exists('users/' . $username . '.xml')){
        $xml=new SimpleXMLElement('users/' . $username . '.xml', 0, true);
        if ($password == $xml->password){
            session_start();
            $_SESSION['username'] = $username;
            header('Location: index.php!');
            die;
        }
    }
    $error=true;
}

```

9 Login Function

Ο χρήστης για να συνδεθεί πληκτρολογεί τα username και password του τα οποία με τη μέθοδο post κρατούνται στις μεταβλητές. Η php αρχικά ελέγχει εάν υπάρχει στον φάκελο users αρχείο με όνομα ίδιο με την μεταβλητή \$username και με τη βοήθεια της συνάρτησης **SimpleXMLElement** ανατρέχει το xml αρχείο με αυτό το όνομα για να πάρει το password. Εάν και το password που πληκτρολόγησε ο χρήστης είναι ίδιο με αυτό του xml τότε με την συνάρτηση session\_start() ξεκινά το session. Χρησιμοποιείται επίσης και η global μεταβλητή **\$\_SESSION[' ']** με αναγνωριστικό όνομα username και παίρνει την τιμή της μεταβλητής username.

Τέλος αφού έχει επιτευχθεί η σύνδεση ανακατευθύνει τον χρήστη στην κεντρική σελίδα και σταματά το script με τη μέθοδο die.

### 5.1.3 Έλεγχος Σύνδεσης και Αποσύνδεση.

Σκοπός της δημιουργίας λογαριασμού στη σελίδα μας ήταν το γεγονός ότι οι συνδεδεμένοι χρήστες θα μπορούσαν να έχουν περισσότερα δικαιώματα από τους ελεύθερους όπως να λαμβάνουν ενημερωτικά δελτία, να προσθέτουν επιθυμητά προϊόντα στη λίστα επιθυμιών τους αλλά και σε κάποιες περιπτώσεις να έχουν ειδικές προσφορές.

Για την εταιρία ο λογαριασμός του χρήστη είναι πολλές φορές σημαντικός ώστε να πιστοποιεί τα στοιχεία του χρήστη και να μπορεί να επικοινωνεί μαζί του. Επίσης η σελίδα μας μέσω της δημιουργίας λογαριασμού θα μπορούσε να ξεχωρίζει το administrator χρήστη απο τους χρήστες/πελάτες.

Για τους λόγους αυτούς χρησιμοποιήθηκε μια συνάρτηση η **check loggedin()**

```

<?php

function loggedin() {

    if(isset($_SESSION['username']) && !empty($_SESSION['username'])) {
        return true;
    } else{
        return false;
    }
}
?>

```

10 Check Function

Η συνάρτηση αυτή συμπεριλαμβάνεται σε όλες τις σελίδες του site και σκοπός της είναι να ελέγχει εάν κάποιος χρήστης είναι συνδεδεμένος με σκοπό να του δίνει τα ανάλογα δικαιώματα. Αυτό που κάνει στην ουσία είναι να εξετάζει εάν υπάρχει ανοιχτό session με αναγνωριστικό username και ότι το περιεχόμενο του session αυτού δεν είναι άδειο. Αν ισχύουν τα παραπάνω επιστρέφει true που σημαίνει ότι χρήστης με το username αυτό είναι συνδεδεμένος τη δεδομένη χρονική στιγμή, αλλιώς επιστρέφει false.

Τέλος η συνάρτηση **logout()** η οποία καταστρέφει όλα τα sessions που είναι ανοιχτά τη δεδομένη χρονική στιγμή αποσυνδέοντας έτσι τον χρήστη.

```
<?php
session_start();
session_destroy();
header('Location: index.php');
?>
```

11 Logout

## 5.2 Δημιουργία της Βάσης Δεδομένων

Για την δημιουργία της Βάσης των προϊόντων που προσφέρει το κατάστημα χρησιμοποιήθηκε η xml τεχνολογία.

### 5.2.1 XML ανάπτυξη βάσης

Για την ανάπτυξη της βάσης όπως αναφέρθηκε χρησιμοποιήθηκε η xml, με την οποία δημιουργήσαμε ένα αρχείο με το όνομα catalog όπου αποθηκεύονται όλα μας τα προϊόντα.

```
items xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
xsi:noNamespaceSchemaLocation="product.xsd">
  <product gerne="wine">
    <id>WHTWN001</id>
    <name>Belmont Sauvignon Blanc 2010</name>
    <producer>Bordeaux</producer>
    <price>15.35</price>
    <category>white</category>
    <quantity>28</quantity>
    <img>images/white_wine.jpg</img>
  </product>
  <product gerne="wine">
    <id>WHTWN002</id>
    <name>Saint Cosme</name>
    <producer>Southern Rhone</producer>
    <price>20.79</price>
    <category>white</category>
    <quantity>30</quantity>
    <img>images/saintcos.jpg</img>
  </product>
```

12 Xml Db



Το αρχείο μας έχει root node με το όνομα items και μέσα περιέχει τα sub nodes με όνομα product το κάθε ένα από τα οποία αποτελεί ένα προϊόν. Με το χαρακτηριστικό gerne δηλώνουμε το είδος του προϊόντος. Κάθε προϊόν περιέχει τα στοιχεία που το προσδιορίζουν όπως όνομα, παραγωγό, την κατηγορία στην οποία ανήκει, έναν μοναδικό αναγνωριστικό κωδικό, καθώς επίσης την τιμή του και την ποσότητα αποθέματος που υπάρχει.

### 5.2.2 XML-Schema Validation

Η χρήση του Xml-Schema είναι απαραίτητη και χρησιμοποιείται με σκοπό να οριστεί η δομή του xml εγγράφου μας και να δηλωθούν οι τύποι των δεδομένων που θα χρησιμοποιηθούν.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="items">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="product" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="id" type="xsd:string"/>
              <xsd:element name="name" type="xsd:string"/>
              <xsd:element name="producer" type="xsd:string"/>
              <xsd:element name="price" type="xsd:decimal"/>
              <xsd:element name="category" type="xsd:string"/>
              <xsd:element name="quantity" type="xsd:integer"/>
              <xsd:element name="img" type="xsd:string"/>
            </xsd:sequence>
            <xsd:attribute name="gerne" use="required" type="xsd:string"/>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

13 Xml Schema

Με το όρο element name και type καθορίζουμε το όνομα και τον τύπο των στοιχείων μας και με τον όρο attribute αντίστοιχα το όνομα και τον τύπο των χαρακτηριστικών τους. Κάθε sequence ορίζει ένα ξεχωριστό node του αρχείου μας και τέλος, με το χαρακτηριστικό maxOccurs βάζουμε έναν περιορισμό πληθικότητας των προϊόντων μας. Στη συγκεκριμένη περίπτωση δεν έχει κάποιο συγκεκριμένο όριο μας και μπορούν να προστεθούν ακόμη περισσότερα προϊόντα.

### 5.3 Εμφάνιση της Βάσης Δεδομένων

Για την εμφάνιση των προϊόντων της βάσης μας αλλά και των στοιχείων τους χρησιμοποιείται η php η οποία μεταφράζει τα δεδομένα του εγγράφου και τα εμφανίζει στη σελίδα μας.

#### 5.3.1 Διάβασμα του Xml από την Php

Αρχικά καθορίστηκε το xml έγγραφο μας σε μία global μεταβλητή ώστε να είναι προσβάσιμο από όλες τις σελίδες μας μέσω της μεταβλητής αυτής χωρίς να χρειάζεται κάθε φορά να ανατρέχουμε στο αρχείο αυτό καθ' αυτό. Η μεταβλητή αυτή ονομάζεται **STORE\_XML\_FILE** και ο καθορισμός γίνεται με τη χρήση της **define()**.

```
<code><?php
/** DEFINE GLOBALS**/
define('STORE_XML_FILE' , 'catalog.xml');
```

#### 14 Define Xml

Έπειτα δημιουργήθηκε μία συνάρτηση η **get\_xml\_catalog** η οποία μπορεί να προσπελάσει ολόκληρο το έγγραφο και να διαβάσει όλα τα περιεχόμενα του χρησιμοποιώντας την συνάρτηση της php την **SimpleXMLElement** και με την **file\_get\_contents** τρέχει τα δεδομένα του xml αρχείου.Όπως φαίνεται τα περιεχόμενα του εγγράφου επιστρέφουν στη συνάρτηση αυτή.

```
<code>function get_xml_catalog()
{
return new SimpleXMLElement(file_get_contents(STORE_XML_FILE));
}
```

#### 15 Get Catalog

Η συνάρτηση αυτή χρησιμοποιείται κάθε φορά που θέλουμε να εμφανίσουμε κάποιο από τα προϊόντα μας ή θέλουμε να πάρουμε κάποιο από τα στοιχεία των προϊόντων.

#### 5.3.2 Εμφάνιση του Xml εγγράφου σε Php

Με τον τρόπο λοιπόν αυτό είμαστε σε θέση με την βοήθεια της php και των συναρτήσεων της να εμφανίσουμε τον κατάλογο μας ολόκληρο ή κάποια από τα προϊόντα του ανάλογα με τα φίλτρα που εμείς καθορίζουμε.





```

foreach(get_xml_catalog() as $product)
{
    if($product->category=='beer') {
        echo'
            <td>
                
            </td>';
        echo'
            <td>
                '. $product->name.'
            </td>';
        echo'
            <td>
                '. $product->producer.'
            </td>';
        echo'
            <td>
                <div class="price">
                    <euro> '. $product->price.'
                </div>
            </td>';
    }
}

```

16 Εμφάνιση Καταλόγου

Για την εμφάνιση των προϊόντων μας ανάλογα με το στοιχείο κατά το οποίο θέλουμε να τα εμφανίσουμε χρησιμοποιείται ο βρόχος `foreach()`. Με τη βοήθεια του βρόχου αυτού σε συνδυασμό με την συνάρτηση `get_xml_catalog` για κάθε sub node(προϊόντος) του xml εγγράφου μας που πληροί τις προϋποθέσεις που οι ίδιοι του βάλαμε να εμφανίσει υπο τη μορφή σειράς πίνακα τα στοιχεία του προϊόντος που επιθυμούμε με την εντολή `echo`. Η συνθήκη `if` που εμφανίζεται παραπάνω δηλώνει ότι θα εμφανιστούν μόνο τα προϊόντα που ανήκουν στη κατηγορία beer.

	Chimay Blue 0.75L	Bieres de Chimay	€ 8.00	add to cart	Normal Stock
	Chimay Red 0.75L	Bieres de Chimay	€ 6.46	add to cart	Low Stock
	Deus	Brouwerij Bosteels	€ 28.24	add to cart	Out Of Stock
	Duvel 0.75L	MoortGat Brewery	€ 7.73	add to cart	Low Stock

17 Εμφάνιση καταλογου B

### 5.3.3 Αναζήτηση Προϊόντων

Για την αναζήτηση προϊόντων τόσο ανά όνομα όσο και ανά κατηγορία ή παραγωγό δημιουργήσαμε μία ακόμη συνάρτηση την **search function**

```
<?php
if(isset($_POST['search'])) {
    $searchfield = $_POST['searchfield'];

    if(!session_id()) session_start();
    $_SESSION['searchfield'] = $searchfield;
    header('Location: search.php');
    die;
}
```

18 Search Function

Με την συνάρτηση αυτή ο χρήστης πληκτρολογεί το προϊόν, παραγωγό ή την κατηγορία που επιθυμεί να αναζητήσει. Το όνομα που πληκτρολόγησε καταχωρείται σε μία μεταβλητή με τη μέθοδο **post** και ξεκινά το **session** της αναζήτησης. Στο session αυτό αποθηκεύεται η τιμή που υπάρχει στην μεταβλητή **\$searchfield**.

```
foreach(get_xml_catalog() as $product)
{
    if($product->category == $_SESSION['searchfield'] ||
    $product->name == $_SESSION['searchfield'] ||
    $product->producer == $_SESSION['searchfield'] ){
        echo'
        <td>
            
        </td>';
        echo'
        <td>
            '. $product->name.'
        </td>';
        echo'
        <td>
            '. $product->producer.'
        </td>';
        echo'
        <td>
            <div class="price">
                €euro; '. $product->price.'
            </div>
        </td>';
    }
```

19 Search Page

Τέλος η συνάρτηση ανακατευθύνει τον χρήστη στην σελίδα αναζήτησης όπου εμφανίζονται τα αποτελέσματα της αναζήτησης του.

Όπως αναφέρθηκε παραπάνω για την εμφάνιση των προϊόντων χρησιμοποιείται ο βρόχος **foreach** και η εντολή **echo** για κάθε στοιχείο του προϊόντος που θα εμφανίσει. Η συνθήκη που πρέπει να τηρούν τα προϊόντα που θα εμφανιστούν φαίνεται στην **if** όπου για κάθε προϊόν του καταλόγου να ισχύει η κατηγορία, το όνομα ή ο παραγωγός να έχει το ίδιο όνομα με αυτό που έχει αποθηκευτεί στο **session** και έτσι εμφανίζει να ανάλογα αποτελέσματα.

### 5.3.4 Λίστα επιθυμιών

Ο χρήστης επίσης μπορεί να προσθέσει τα προϊόντα της επιθυμίας του στη λίστα επιθυμιών. Αυτό γίνεται εύκολα με τη συνάρτηση **addToWishlist** μέσω της **php** και της **xml** τεχνολογίας. Ο χρήστης επιλέγει ένα προϊόν με βάση τον κωδικό του ο οποίος αποθηκεύεται στη μεταβλητή **\$product\_id** με τη μέθοδο **request** χρησιμοποιώντας τον υπερσύνδεσμο **a href**.

```
<a href="addToWishlist.php?id='.$product->id.'">wishlist</a>
```

#### 20 Wishlist href

Έπειτα ακολουθεί η διαδικασία δημιουργίας του **xml** εγγράφου με τα στοιχεία του επιθυμητού προϊόντος και του ονόματος του πελάτη που έκανε το αίτημα.

```
if(isset($_POST['addToWish'])){
    foreach(get_xml_catalog() as $product)
    {
        if($product->id == $product_id){
            $prodID = $product->id;
            $prodName = $product->name;
            $prodProducer = $product->producer;
            $prodPrice = $product->price;
            $prodImg = $product->img;
        }
    }
    if(count($errors) == 0){
        $xml = new SimpleXMLElement('<wish></wish>');
        $xml->addChild('client', $_SESSION['username']);
        $xml->addChild('id', $prodID);
        $xml->addChild('name', $prodName);
        $xml->addChild('producer', $prodProducer);
        $xml->addChild('price', $prodPrice);
        $xml->addChild('img', $prodImg);

        $xml->asXML('wishlist/' . $_SESSION['username'] . $prodID . '.xml');
        header('Location: wishlist.php');
        die;
    }
}
```

#### 21 AddToWishlist

Το σύστημα απαιτεί να είναι κάποιος χρήστης συνδεδεμένος ώστε να μπορεί να προστεθεί κάποιο προϊόν στη wishlist.

Αφού λοιπόν ελεγχθεί ο κωδικός του προϊόντος και είναι ίδιος με αυτόν του καταλόγου μας ξεκινά η διαδικασία καταχώρησης των στοιχείων του προϊόντος του εγγράφου σε php μεταβλητές στον server όπου από εκεί με την βοήθεια της συνάρτησης **SimpleXMLElement** δημιουργώ το αρχείο με όνομα το username του χρήστη και το id του προϊόντος.

### 5.3.5 Δικαιώματα του Διαχειριστή

Ο διαχειριστής της σελίδας όσον αφορά τα προϊόντα έχει το δικαίωμα να αλλάζει τα αποθέματα των προϊόντων καθώς αυτά εξαντλούνται, όπως επίσης και όταν έρχονται νέα στο κατάστημα μας. Για την ανανέωση του καταλόγου όσον αφορά το απόθεμα προϊόντος ακολουθείται η παρακάτω συνάρτηση.

```

if(isset($_POST['change'])){
    $ChangeQty = $_POST['changeQty'];

    $items = new SimpleXMLElement(file_get_contents('catalog.xml'));

    foreach($items as $prod){
        if($prod->id == $product_id)
            $prod->quantity = $ChangeQty;
    }

    $items->asXML('catalog.xml');
}

```

#### 22 ChangeQuantity

Επιλέγοντας το προϊόν με το id του ο διαχειριστής πληκτρολογεί στην input form με το όνομα **changeQty** τον αριθμό αποθέματος του συγκεκριμένου προϊόντος. Με τη χρήση της συνάρτησης **SimpleXMLElement** της **file\_get\_content** ανατρέχει όλα τα στοιχεία του xml καταλόγου μας. Όταν συναντήσει το προϊόν με id ίδιο με αυτό που έκανε αίτηση στον server ο διαχειριστής, παίρνει την ποσότητα του προϊόντος και την κάνει ίδια με αυτή που πληκτρολόγησε μέσω της μεθόδου **post**. Τέλος ανανεώνεται και αποθηκεύεται το xml έγγραφό μας.

## 5.4 Προσθήκη στη Κάρτα Αγορών

Ένα από τα σημαντικότερα μέρη ενός ηλεκτρονικού καταστήματος είναι η δυνατότητα να μπορεί ο πελάτης να προσθέτει τα προϊόντα της επιθυμίας του στο καλάθι αγορών. Αυτό πραγματοποιείται μέσω της `php` και των **php sessions** που θα κρατούν την πληροφορία στον server.

### 5.4.1 Αίτηση Κωδικού

Το κυριότερο αναγνωριστικό για την επιλογή ενός προϊόντος είναι ο αναγνωριστικός κωδικός (`id`) που το προσδιορίζει. Κύριο μέλημα μας λοιπόν είναι να βρεθεί ένας τρόπος με τον οποίο να γίνεται μία αίτηση στον server μας που να ζητείτε ο κωδικός αυτός. Αυτό το καταφέραμε με την βοήθεια της `php`.

Αρχικά έπρεπε να δημιουργηθεί ένας σύνδεσμος με ετικέτα `<a>` και την ιδιότητα της ετικέτας `href` στην οποία περιλαμβάνεται το `url` ενός αρχείου ή μιας σελίδας, όπως και το `php` αρχείο `addToCart` που είναι η σελίδα μας. Ο υπερσύνδεσμος αυτός περιλαμβάνεται σε κάθε προϊόν που εμφανίζεται στη σελίδα μας μέσω του βρόχου `foreach()`.

```
foreach(get_xml_catalog() as $product)
{
    echo '
        <td>
            <div class="addToCart">
                <a href="addToCart.php?id='.$product->id.'">add to cart</a><br>
            </div>
        </td>';
}
```

#### 23 Υπερσύνδεσμος a href

Σκοπός του συνδέσμου είναι πατώντας τον σε κάθε προϊόν να μας οδηγεί στην σελίδα **addToCart.php** δίνοντας μας τόν κωδικό του συγκεκριμένου προϊόντος που επιλέξαμε. Αυτό επιτυγχάνεται χρησιμοποιώντας την `get_xml_catalog()` απο την οποία μέσω της μεταβλητής `$product->id` παίρνουμε το επιθυμητό `id`. Αφού λοιπόν ο πελάτης πατήσει τη προσθήκη στη κάρτα του κάθε προϊόντος ο σύνδεσμός μας μετατρέπεται υπό τη μορφή που δημιουργήσαμε και όπου `id= $product->id` εμφανίζεται ο κωδικός προϊόντος. Κάθε προϊόν ζητείται από τον κωδικό του με τη μέθοδο `request['id']` ο οποίος κωδικός αποδίδεται σε μία μεταβλητή `$product_id` ώστε να μπορεί να χρησιμοποιηθεί απο την `php`.

```

<?php
$product_id = $_REQUEST['id'];

if (product_exists($product_id))
{
    $shopping_cart->AddItem($product_id);
}

set_shopping_cart($shopping_cart);
?>

```

Στη συνέχεια γίνεται έλεγχος για το εάν υπάρχει ο κωδικός στο κατάλογο με τα προϊόντα μας με συνθήκη και τη μέθοδο `product_exist()`.

#### 5.4.2 Δημιουργίας της κάρτας Αγορών

Από τη στιγμή που ο πελάτης κάνει αίτηση του κωδικού ενός προϊόντος προς τον server για να το προσθέσει στη κάρτα αγορών του ξεκινά μια διαδικασία μέσω της php για να επιτευχθεί το αίτημα του.

Η δημιουργία της κάρτας αγορών γίνεται με τη βοήθεια ενός ακόμα php session, το `session(cart)` το οποίο κρατά πληροφορίες για τα προϊόντα που προσθέτουμε στο καλάθι μας.

```

function get_shopping_cart()
{
    if (!isset($_SESSION['cart']))
        return new ShoppingCart();
    else
        return unserialize($_SESSION['cart']);
}

function set_shopping_cart($cart)
{
    $_SESSION['cart'] =serialize($cart);
}

```

24 Δημιουργία Κάρτας Αγορών

Αφού ο πελάτης επιλέξει ένα προϊόν στο καλάθι του ξεκινά το `session(cart)` ώστε να προστεθεί το προϊόν στη κάρτα αγορών.



Αρχικά μέσω της php γίνεται ο έλεγχος για το εάν η κάρτα(το session) έχει ανοιχτεί ήδη μέσω ενός άλλου προϊόντος, δηλαδή να υπάρχει ήδη κάποιο προϊόν στο καλάθι ώστε να αποθηκευτεί το προϊόν στο ίδιο session ή στην περίπτωση που δεν έχει ανοίξει ακόμη να ορίσει ένα καινούριο **ShoppingCart()**. Αυτό γίνεται με την συνάρτηση **get\_shopping\_cart()**.

Επειδή τα δεδομένα της database όταν καλούνται απο την php αρχικά αποθηκεύονται σε ένα array μέσω της συνάρτησης **addItem** και στην συνέχεια επιστρέφονται πίσω στη κάρτα μας ώστε να εμφανιστούν, γίνεται η χρήση δύο συναρτήσεων, της **serialize()** η οποία αποθηκεύει τα δεδομένα της κάρτας μέσα στο **session(cart)** σαν συμπίεση υπο τη μορφή πίνακα. Η λειτουργία της **serialize()** γίνεται μέσω της συνάρτησης **set\_shopping\_cart**. Η δεύτερη συνάρτηση είναι η **unserialize()** η οποία μέσω της **get\_shopping\_cart** επιστρέφει τα δεδομένα που αποθηκεύτηκαν στο array απο την **serialize()** ως **php objects**.

```
protected $items=array();

public function addItem($product_id)
{
    if(array_key_exists($product_id,$this->items)){
        $this->items[$product_id] = $this->items[$product_id] +1;
    }
    else {
        $this->items[$product_id] = 1;
    }
}
```

25 AddItem

Για την προσθήκη των στοιχείων του κάθε προϊόντος με βάση το id του στον πίνακα δημιουργήθηκε αρχικά το αρχείο **shoppingCart.php** που περιέχει τις συναρτήσεις για τη δημιουργία της κάρτας.Χρησιμοποιείται η συνάρτηση **addItem**.Κάθε φορά που προσθέτει ο χρήστης ένα προϊόν γίνεται έλεγχος για το εάν το key με τον κωδικό αυτό υπάρχει ήδη στο array.Εαν υπάρχει ήδη αυξάνεται κατά ένα η ποσότητα.

### 5.4.3 Συναρτήσεις της κάρτας Αγορών

Για την δημιουργία της κάρτας αγορών απαιτούνται αρκετές συναρτήσεις πέρα από τη δημιουργία της κάρτας, καθώς και της αποθήκευσης προσωρινά των προϊόντων σε αυτή. Για την χρήση της κάρτας δημιουργήθηκαν δύο php αρχεία το **shoppingCart.php** και το **functions.php**. Τα αρχεία αυτά συμπεριλαμβάνονται σε όλες τις σελίδες που περιέχονται τα προϊόντα μας καθώς και στην σελίδα της κάρτας αγορών.

Παρακάτω ακολουθούν οι συναρτήσεις που απαιτούνται για την δημιουργία και εμφάνιση της κάρτας.

```
function product_exists($product_id)
{
    foreach(get_xml_catalog() as $product)
    {
        if($product->id == $product_id)
            return true;
    }
    return false;
}
```

26 Product\_Exist

Η **product\_exist()** ελέγχει εάν υπάρχει προϊόν με το κωδικό που ζητήθηκε στο xml έγγραφό μας και ανάλογα επιστρέφει true ή false

```
function get_item_cost($product_id)
{
    foreach(get_xml_catalog() as $product)
    {
        if($product_id == $product->id)
            return $product->price;
    }
    throw new Exception('item not found:' . $product_id);
}
```

27 Get\_item\_Cost

Η **get\_item\_cost** η οποία ελέγχει το προϊόν μέσα στο xml αρχείο με τον κωδικό που ζητήθηκε και επιστρέφει την τιμή του.

```
function get_item_qty($product_id)
{
    foreach(get_xml_catalog() as $product)
    {
        if($product_id == $product->id)
            return $product->quantity;
    }
    throw new Exception('item not found:' . $product_id);
}
```

#### 28 Get\_item\_Qty

Η **get\_item\_qty** όπου παίρνει την ποσότητα του συγκεκριμένου προϊόντος με βάση τον κωδικό που ζητήθηκε.

```
public function GetItemCost($product_id)
{
    $cost_string = get_item_cost($product_id);
    $cost_float = "$cost_string" + 0;

    return $cost_float *$this->GetItemQuantity($product_id);
}
```

#### 29 GetItemCost

Η **GetItemCost()** η οποία παίρνει την τιμή της **get\_item\_cost** για το συγκεκριμένο id, την μετατρέπει σε float και επιστρέφει τον αριθμό(τιμή) του προϊόντος επί την ποσότητα του προϊόντος που υπάρχει στην κάρτα.

```
private function getShippingCostFor($price)
{
    if ($price < 10)
        return 0.12;
    else
        if ($price <30)
            return 0.08;
        else
            return 0.00;
}
```

#### 30 getShippingCostFor

Η **getShippingCostFor** η οποία για κάθε τιμή κάτω του 10 επιστρέφει 0.12, κάτω του 30 επιστρέφει 0.08 και εκτός συνθηκών επιστρέφει 0.00 και χρησιμοποιείται για τα μεταφορικά ανάλογα της τιμής του προϊόντος.

```
public function GetSubTotal()
{
    $total = 0;
    foreach ($this->items as $product_id => $quantity)
        $total = $total + $this->GetItemCost($product_id);

    return $total;
}
```

### 31 GetSubTotals

Η **GetSubTotal()** όπου για κάθε item μέσα στο array με βάση την ποσότητα του κάθε προϊόντος επιστρέφει το επιστρέφει το συνολικό ποσό που παίρνει για κάθε id μέσω της **GetItemCost()**.

```
public function GetItemShippingCost($product_id)
{
    return $this->GetItemQuantity($product_id) * $this->getShippingCostFor(get_item_cost($product_id));
}
```

### 32 GetItemShippingCost

Η **GetItemShippingCost()** που για το id που ζητήθηκε επιστρέφει τα μεταφορικά ανάλογα με την τιμή του προϊόντος επί την ποσότητα του προϊόντος που έχει ζητηθεί και βρίσκεται στο array.

```
public function GetTotal()
{
    return $this->GetSubTotal() + $this->GetShippingCost();
}
```

### 33 GetTotal

Η **GetTotal()** η οποία επιστρέφει το συνολικό ποσό επί τα μεταφορικά των προϊόντων.

```
public function GetItemQuantity($product_id)
{
    return intval($this->items[$product_id]);
}
```

### 34 GetItemQuantity

Η **GetItemQuantity()** που επιστρέφει για το συγκεκριμένο id έναν ακέραιο αριθμό ο οποίος αντιπροσωπεύει το πόσες φορές έχει εισαχθεί το id αυτό στο array και είναι η ποσότητα του προϊόντος στην κάρτα.

```
public function EmptyCart()
{
    $this->items = array();
}
```

35 EmptyCart

Τέλος η **EmptyCart()** η οποία αναδημιουργεί το array και στην ουσία διαγράφει την κάρτα μας.

**5.4.4 Εμφάνιση της κάρτας Αγορών**

Μετά την προώθηση προϊόντων στην κάρτα αγορών τα προϊόντα μας πρέπει να εμφανιστούν σε αυτή. Για την εμφάνιση των προϊόντων όπως αναφέραμε στη κάρτα αγορών αναγράφεται ο κωδικός του προϊόντος καθώς και τα στοιχεία όπως τιμή, ποσότητα, σύνολο και μεταφορικά.

ProductID	Quantity	Amount
TQL001	1	€ 22.9
RSWN003	1	€ 6.5
TQL004	2	€ 47.8
RDWN001	1	€ 124.24
	Shipping	€ 0.36
	Total	€ 201.8

36 ShoppingCart

Για την εμφάνιση των προϊόντων μας στην κάρτα αγορών μέσω της php έγινε η χρήση πίνακα. Δημιουργήθηκε αρχικά μία μέθοδος με το όνομα **render\_shopping\_cart\_row** με μεταβλητές την κάρτα που έχουμε ανοίξει σε session, τον κωδικό του προϊόντος και μίας ακόμη μεταβλητής της **\$line\_item\_counter** της οποίας η χρήση είναι κυρίως για το site της PayPal και μετρά κάθε γραμμή ξεχωρίζοντας το κάθε προϊόν απο τα υπόλοιπα.

Όπως αναφέραμε παραπάνω δημιουργήθηκαν αρκετές συναρτήσεις.php για την δημιουργία και εμφάνιση της κάρτας. Η συνάρτηση αυτή περιέχει ακόμη κάποιες μεταβλητές αρκετά χρήσιμες για την εμφάνιση της κάρτας όπως την \$quantity η οποία μέσω της συνάρτησης **GetItemQuantity()** μας δίνει την ποσότητα του κάθε προϊόντος προς την κάρτα και το \$amount το οποίο κάθε φορά μας δίνει την τιμή επί την ποσότητα του κάθε προϊόντος. Η συνάρτηση αυτή έχει ως επιστροφή υπό την μορφή πίνακα τον κωδικό του κάθε προϊόντος, την ποσότητα του και την τιμή επί την ποσότητα.

```
function render_shopping_cart_row(ShoppingCart $shopping_cart , $product_id , $line_item_counter)
{
    $quantity = $shopping_cart->GetItemQuantity($product_id);
    $amount = $shopping_cart->GetItemCost($product_id);
    $unit_cost = get_item_cost($product_id);

    $shipping_ammount = $shopping_cart->GetItemShippingCost($product_id);

    .....

    return "
        <tr>
            <td>
                $product_id
                <input type='hidden' name='item_name_$line_item_counter' value='$product_id' />
            </td>
            <td>
                $quantity
                <input type='hidden' name='quantity_$line_item_counter' value='$quantity' />
            </td>
            <td>
                €$amount
                <input type='hidden' name='amount_$line_item_counter' value='$unit_cost' />
                <input type='hidden' name='shipping_$line_item_counter' value='$shipping_ammount' />
            </td>
        </tr>";
}
```

### 37 ShopCartRow Render

Για την εμφάνιση των μεταφορικών δημιουργήθηκε η συνάρτηση **render\_shopping\_cart\_shipping\_row** η οποία επιστρέφει το σύνολο των μεταφορικών μέσω της μεθόδου **getShippingCost** ανάλογα με τα προϊόντα που επιλέξαμε.

```
function render_shopping_cart_shipping_row(ShoppingCart $shopping_cart)
{
    return "
    <tr>
        <td>
        </td>
        <td>
            Shipping
        </td>
        <td>
            €euro: ".$shopping_cart->GetShippingCost()."
        </td>
    </tr>
    ";
}
```

### 38 Render Shippings

Στο τέλος θα έπρεπε να εμφανίζεται και το συνολικό ποσό της κάρτας συμπεριλαμβανομένων των τιμών αλλά και των μεταφορικών όλων των προϊόντων. Αυτό επιτυγχάνεται μέσω της συνάρτησης `render_shopping_cart_total` και της μεθόδου `getTotal()` παίρνοντας το συνολικό ποσό της κάρτας.

```
function render_shopping_cart_total_row(ShoppingCart $shopping_cart)
{
    return "
    <tr>
        <td>
        </td>
        <td>
            Total
        </td>
        <td>
            €euro: ".$shopping_cart->GetTotal()."
        </td>
    </tr>
    ";
}
```

### 39 Render Totals

Για την εμφάνιση όλων των παραπάνω στοιχείων για να πάρουμε το τελικό αποτέλεσμα της κάρτας μας χρησιμοποιήθηκε η συνάρτηση `render_shopping_cart` η οποία εμφανίζει όλες τις τιμές που επιστρέφουν από τις παραπάνω συναρτήσεις υπό μορφή πίνακα παίρνοντας το τελικό αποτέλεσμα της κάρτας μας.

```
function render_shopping_cart(ShoppingCart $shopping_cart)
{
    $output = ""

    <table class='shoppingCart'>
        <tr>
            <th>
                ProductID
            </th>
            <th>
                Quantity
            </th>
            <th>
                Amount
            </th>
        </tr>";

        $line_item_counter = 1;
        foreach($shopping_cart->GetItems() as $product_id)
        {
            $output .= render_shopping_cart_row($shopping_cart, $product_id, $line_item_counter);
            $line_item_counter++;
        }

        $output .= render_shopping_cart_shipping_row($shopping_cart);

        $output .= render_shopping_cart_total_row($shopping_cart);

        $output .= "</table>";

        return $output;
    }
}
```

#### 40 Render Shopping Cart

Για κάθε προϊόν που εισάγεται στην κάρτα αγορών με βάση το id του εμφανίζει τα στοιχεία όπως είπαμε, ποσότητα και συνολικό ποσό και επίσης στις δύο τελευταίες γραμμές τα μεταφορικά και το συνολικό κόστος.

Τέλος για την προώθηση της κάρτας προς ολοκλήρωση αγοράς στο site της PayPal έπρεπε να καθορίσουμε δύο global μεταβλητές.

Οι μεταβλητές αυτές δηλώνουν το όνομα της εταιρίας μας, για την ακρίβεια το mail της εταιρίας μας και επίσης τη σελίδα της PayPal Sandbox όπου θα καθοδηγηθεί ο πελάτης του καταστήματος.



```
define('PAYPAL_BUSINESS_VALUE' , 'Doulge_R@theCellar.com');
define('PAYPAL_FORM_URL' , 'https://www.sandbox.paypal.com/cgi-bin/webscr');
```

#### 41 Καθορισμός στοιχείων PayPal

Επίσης θα χρειαστεί ακόμη μία τελευταία συνάρτηση, η **render\_paypal\_checkout** η οποία περιέχει όλα τα απαραίτητα στοιχεία, για την ακρίβεια την απαραίτητη φόρμα που δίνει όλα τα στοιχεία ώστε να μεταφέρει ο πελάτης την κάρτα του στη σελίδα της PayPal.

```
]function render_paypal_checkout(ShoppingCart $shopping_cart){
    .....
    .....

    return "
    <form action='". PAYPAL_FORM_URL .' method='post'>
        <input type='hidden' name='business' value='". PAYPAL_BUSINESS_VALUE .' />
        <input type='hidden' name='cmd' value='_cart' />
        <input type='hidden' name='upload' value='1' />
        <input type='hidden' name='currency_code' value='EUR' />
        <input type='hidden' name='lc' value='GR' />

        ". render_shopping_cart($shopping_cart) ."

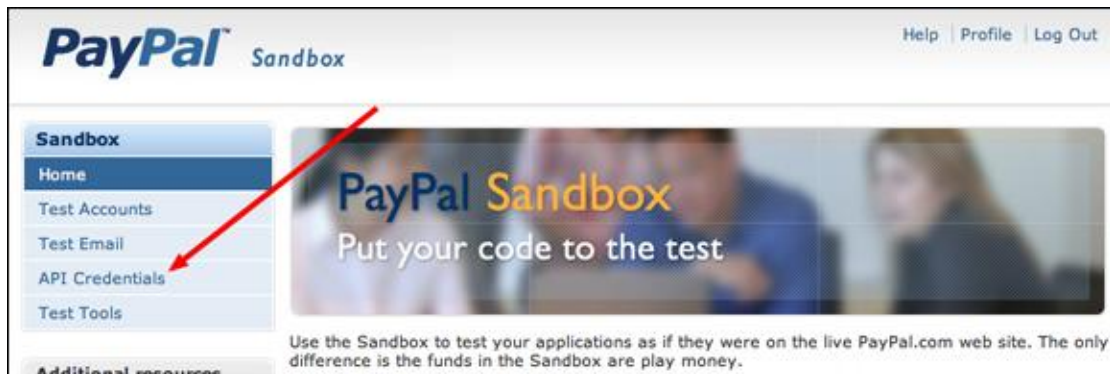
        <input class='centerBtn' type='image' name='submit'
        src='https://www.paypal.com/en_us/i/btn/btn_buynowcc_lg.gif' />
    </form>";
```

#### 42 Render Checkout

Στη φόρμα αναγράφονται όλα τα στοιχεία όπως το action της φόρμας που είναι το url του sandbox της PayPal, το όνομα της εταιρίας, το συνάλλαγμα που χρησιμοποιείται καθώς επίσης και την κάρτα αγορών.

## 5.5 PayPal Sandbox

Τελικός στόχος της δημιουργίας του ηλεκτρονικού μας καταστήματος δεν είναι άλλο από την ηλεκτρονική αγορά των προϊόντων μας από τους πελάτες του καταστήματος. Για να επιτευχθεί αυτό και σκεπτόμενοι την ασφαλέστερη συναλλαγή του καταστήματος με τους πελάτες χρησιμοποιήθηκε η PayPal και για την ακρίβεια μία εφαρμογή της το PayPal Sandbox.



### 43 PayPal Sandbox

Το PayPal Sandbox είναι όπως αναφέρθηκε μία εφαρμογή της PayPal η οποία δουλεύει σε συνδυασμό με το PayPal for Developers και σκοπό έχει να δημιουργήσει ένα εικονικό περιβάλλον ασφαλών συναλλαγών ενός η περισσότερων καταστημάτων για την εξυπηρέτηση των πελατών τους.

### 5.5.1 Δημιουργία εικονικών λογαριασμών

Η δημιουργία ενός πραγματικού λογαριασμού είναι απαραίτητη μέσω του οποίου δουλεύει η εφαρμογή. Μέσω του PayPal Developers δημιουργούνται όλοι αυτοί οι λογαριασμοί (εικονικοί) που θα χρησιμοποιηθούν για τις συναλλαγές του καταστήματος και των πελατών του.

<input type="checkbox"/>	Email address	Type	Country
<input type="checkbox"/>	MDragon@mail.com <a href="#">Profile</a>   <a href="#">Notifications</a>	Premier	GR
<input type="checkbox"/>	▶ avasGD@mail.com	Premier	GR
<input type="checkbox"/>	▶ Doulge_R@theCellar.com	Business	GR

### 44 Εικονικοί λογαριασμοί

Μπορούμε να δημιουργήσουμε λογαριασμό business που αποτελεί την επιχείρηση ή πελάτη.

Για την δημιουργία του κάθε λογαριασμού συμπληρώνουμε στοιχεία όπως mail, ονοματεπώνυμο, διεύθυνση και όλα όσα απαιτεί η φόρμα για τη δημιουργία του. Επίσης ζητείται να δημιουργήσουμε έναν εικονικό τραπεζικό λογαριασμό ο οποίος γίνεται αυτόματα και έναν 16-digit PayPal κωδικό για MasterCard ή Visa ανάλογα την επιλογή του πελάτη. Οσον αφορά τον λογαριασμό του καταστήματος ή του πωλητή, δίνεται η δυνατότητα ακόμη να επιλέξουν ονομασία του καταστήματος καθώς και τον τύπο του, τα είδη που εμπορεύεται και άλλα.

#### 45 PayPal code

Η δημιουργία κωδικού τραπεζής αλλά και PayPal κάρτας, αν και εικονικοί λογαριασμοί είναι απαραίτητη για την επαλήθευση των στοιχείων και του λογαριασμού από την PayPal.

**Bank account** [Back to My Profile](#)

Select	Bank Account	Status
<input checked="" type="radio"/>	Rabobank Nederland x-1236 Greece	Primary

Remove Add

#### 46 Bank account

### 5.5.2 Αγορά μέσω της PayPal

Ο χρήστης/πελάτης αφού επιλέξει τα επιθυμητά προϊόντα από το site μας στο καλάθι αγορών προωθεί την παραγγελία του στη σελίδα της PayPal ώστε να την ολοκληρώσει.

#### To Kelari

The screenshot shows the PayPal checkout interface. On the left, there is a 'Your order summary' box containing a table with the following data:

Descriptions	Amount
CMGN003 Item price: €186.69 Quantity: 1	€186.69
<b>Item total</b>	<b>€186.69</b>
<b>Total €186.69 EUR</b>	

On the right, under the heading 'Choose a way to pay', there are two main options:

- Pay with my PayPal account**: Includes a sub-heading 'Log in to your PayPal account to complete the purchase'. It features input fields for 'Email' (containing 'doulgecellar@gmail.com') and 'PayPal password'. There is a checkbox for 'This is a private computer. What's this?' and a 'Log In' button. A link for 'Forgot your email or password?' is also present.
- Pay with my credit or debit card**: Includes a sub-heading '(Optional) Sign up for PayPal for faster future checkout'.

#### 47 Sandbox Payment

Αφού μεταφερθεί στη σελίδα της PayPal όπου φαίνεται το όνομα του καταστήματος, επίσης εμφανίζονται τα στοιχεία της παραγγελίας όπως ο κωδικός του προϊόντος, η τιμή του και η ποσότητα. Για να προωθηθεί η παραγγελία ζητείται από τον χρήστη να εισάγει το mail και τον κωδικό πρόσβασης για να συνδεθεί ή να δημιουργήσει νέο λογαριασμό και κατόπιν να πιστοποιήσει την παραγγελία του προς το κατάστημα.

### 5.5.3 Ιστορικό Συναλλαγών στην PayPal

Μετά την επιτυχή ολοκλήρωση μίας συναλλαγής η παραγγελία εμφανίζεται τόσο στον προσωπικό λογαριασμό του κάθε πελάτη όσο και σε αυτόν του πωλητή. Ένας administrator ο οποίος έχει πρόσβαση στον λογαριασμό του καταστήματος θα πρέπει να ελέγχει σε τακτά χρονικά διαστήματα για τυχόν παραγγελίες που γίνονται και για άλλες που περιμένουν και δεν έχουν εκπληρωθεί ακόμη.

## 6. Αποτελέσματα



**PayPal**

My Account | Send Money | Request Money | Merchant Services | Auction Tools

Overview | Top Up | Withdraw | History | Resolution Center | Profile

**Welcome, Giannis Doulge**

Business Name: To Kelari | Account Type: Business | Status: [Verified](#) | Account Limits: [View Limits](#)

**PayPal balance: €260.61 EUR**

Available balance in EUR (primary): €128.78 EUR  
 Pending balance: €131.83 EUR

Total balance (all currencies, available and pending) converted to EUR: €260.61 EUR [Hide](#)

Currency	Available	Pending	Total
EUR (Primary)	€128.78 EUR	<a href="#">€131.83 EUR</a>	€260.61 EUR
USD	\$0.00 USD	\$0.00 USD	\$0.00 USD

### 48 Στοιχεία καταστήματος

Στον λογαριασμό του administrator κάθε φορά που γίνεται κάποια παραγγελία εμφανίζεται το ποσό που εκκρεμεί δηλώνοντας ότι υπάρχουν παραγγελίες ανεκπλήρωτες. Επίσης εμφανίζονται σαν ιστορικό όλες οι παραγγελίες που εκκρεμούν (under review) καθώς και αυτές που έχουν ολοκληρωθεί.

My recent activity - Last 7 days (Feb 12, 2015-Feb 19, 2015)

[Archive](#) [What's this](#) [Payment status glossary](#)

<input type="checkbox"/>	Date	Type	Name/Email	Payment status	Details	Order status/Actions	Gross
<input type="checkbox"/>	Feb 18, 2015	Payment From	Maria Dragon	Under Review	<a href="#">Details</a>	<a href="#">Issue refund</a>	€86.10 EUR
<input type="checkbox"/>	Feb 17, 2015	Payment From	Giorgos Papadakis	Under Review	<a href="#">Details</a>	<a href="#">Issue refund</a>	€24.36 EUR
<input type="checkbox"/>	Feb 15, 2015	Payment From	Maria Dragon	Under Review	<a href="#">Details</a>	<a href="#">Issue refund</a>	€27.10 EUR
<input type="checkbox"/>	Feb 14, 2015	Payment For	Giorgos Papadakis	Completed	<a href="#">Details</a>		-€100.00 EUR
<input type="checkbox"/>	Feb 14, 2015	Payment From	Giorgos Papadakis	Completed	<a href="#">Details</a>	<a href="#">Issue refund</a>	€39.60 EUR
<input type="checkbox"/>	Feb 13, 2015	Fee Reversal From	PayPal	Completed	<a href="#">Details</a>		€0.92 EUR

### 49 Ιστορικό παραγγελιών

Επίσης ο διαχειριστής θα πρέπει συχνά να ελέγχει την πληρότητα των αποθεμάτων του καταστήματος (stock) και ανάλογα το αν μπορεί να προωθήσει μια παραγγελία ή να την ακυρώσει λόγω μη πληρότητας ενημερώνοντας έτσι τον πελάτη επιστρέφοντας το ποσό που δεσμεύτηκε ο πελάτης.

Από την πλευρά του πελάτη αφού πραγματοποιήσει την παραγγελία του, τα στοιχεία αυτής εμφανίζονται στον προσωπικό του λογαριασμό. Του παρέχεται η επιλογή να ελέγξει λεπτομερώς ολόκληρη την παραγγελία, κωδικός παραγγελίας, ποσό και οποιαδήποτε λεπτομέρεια.

**Shopping Cart Payment Sent** (Unique Transaction ID #941796899S358383A)

Original Transaction				
Date	Type	Status	Details	Amount
Feb 18, 2015	Payment To To Kelari	Completed	...	-€86.10 EUR

Related Transaction				
Date	Type	Status	Details	Amount
Feb 18, 2015	Charge From Credit Card	<a href="#">Completed</a>	<a href="#">Details</a>	€86.10 EUR

**Business Name:** To Kelari (The recipient of this payment is **Non-U.S. - Verified**)

**Email:** Doulge\_R@theCellar.com

**Shopping Cart Contents**

Qty	Item	Options	Price
1	CMGN001		€86.10 EUR
<b>Amount</b>			<b>€86.10 EUR</b>

**Order Description:** Shopping Cart

**Item Total:** €86.10 EUR

**Sales Tax:** €0.00 EUR

**Shipping:** €0.00 EUR

**Handling:** €0.00 EUR

**Total amount:** -€86.10 EUR

**Fee amount:** €0.00 EUR

**Net amount:** -€86.10 EUR

**Date:** Feb 18, 2015

**Time:** 17:21:11 GMT+02:00

**Status:** Completed

**Shipping Address:** Maria Dragon  
Kalokairinou 66  
Heraklion, Crete

50 Λεπτομέρειες παραγγελίας

Τέλος το σύστημα του παρέχει τη δυνατότητα να ακυρώσει την παραγγελία εφόσον αυτή δεν έχει εκπληρωθεί μέσα σε ένα χρονικό διάστημα, επιστρέφοντας του το ποσό που δεσμεύτηκε.

.....

**Business Contact Information**

Customer Service Email: [Doulge\\_R@theCellar.com](mailto:Doulge_R@theCellar.com)

Customer Service Phone: +30 2815211233

.....

**Funding Type:** Credit Card

**Funding Source:** €86.10 EUR - MasterCard Card XXXX-XXXX-XXXX-4444

This credit card transaction will appear on your bill as "PAYPAL \*KELARI".

.....

**Description:** Shopping Cart

.....

Having problems with this transaction? Please talk directly to the seller and try to resolve the issue with them.

You can still report the problem in the [Resolution Centre](#) within 180 days.

.....

**Sandbox Test:**  
Payment Review

[Accept](#) [Deny](#)

.....

**51 Δυνατότητα ακύρωσης**

**6.1 Ωφέλη**

Η ανάπτυξη της πτυχιακής αυτής από πλευράς εργασίας εξυπηρετεί κάποιον που θέλει να αναπτύξει μια εφαρμογή/ιστοσελίδα ηλεκτρονικού εμπορίου όπως κάποιο κατάστημα. Τα οφέλη προς τον καταναλωτή είναι η εύκολη και γρήγορη αγορά ή σύγκριση τιμών απλά και γρήγορα μέσω του διαδικτύου.

Ο σκοπός της εκπόνησης της εργασίας αυτής είναι εκπαιδευτικός ως προς την καλύτερη κατανόηση των γλωσσών και τεχνολογιών που αναπτύσσονται κατά την υλοποίηση της. Παράλληλα όμως και αναπτυξιακός ως μία πολύ χρήσιμη εφαρμογή που αναπτύσσεται ραγδαία τα τελευταία χρόνια.

## **6.2 Δυσκολίες**

Η κυριότερη δυσκολία που αντιμετωπίστηκε ήταν ως προς την μεταφορά των προϊόντων προς τη κάρτα των αγορών μιάς και χρησιμοποιήθηκαν αρκετές συναρτήσεις αλλά και γενικότερα ο τρόπος με τον οποίο δουλεύει και επικοινωνεί η rhr και η αλληλεπίδραση με ένα array για την προσωρινή αποθήκευση των τιμών και την επιστροφή των δεδομένων πίσω ξανα ως rhr αντικείμενα.

## **6.3 Μελλοντικές επεκτάσεις**

Όσον αφορά της μελλοντικές επεκτάσεις σκοπός είναι το σύστημά μας να υποστηρίζει και άλλων ειδών τρόπους πληρωμής.Επίσης την επιστροφή των δεδομένων και των πληροφοριών για κάθε παραγγελία που γίνεται μέσω της σελίδας της PayPal πίσω στην κεντρική μας σελίδα.



## Bibliography

- Developers, PayPal. "https://developer.paypal.com/" *https://developer.paypal.com/*. PayPal.
- html5. "http://www.w3schools.com/html/default.asp." *http://www.w3schools.com/*. W3Shools.
- JavaScript. "http://www.w3schools.com/js/default.asp." *http://www.w3schools.com/*. W3Shools.
- PayPal. "https://www.paypal.com/gr/webapps/mpp/home." *https://www.paypal.com*. PayPal.
- php. "http://php.net/manual/en/book.xsl.php." *http://php.net/manual/en/book.xsl.php*. php.net.
- . "http://www.w3schools.com/php/default.asp." *http://www.w3schools.com/*. W3Shools.
- php\_sessions. "http://www.tutor.edu.gr/index.php/php/phpii/php-sessions." *http://www.tutor.edu.gr/*. Tutor.
- php-arrays. "http://dide.flo.sch.gr/Plinet/Tutorials-PHP-Bible/Tutorials-PHP-Bible-4.html." *http://dide.flo.sch.gr/*. dide.
- php-parsing. "http://stackoverflow.com/questions/24269817/php-xpath-error-in-foreach-statement." *http://stackoverflow.com/*. stackoverflow.
- Sandbox, PayPal. "https://www.sandbox.paypal.com/gr/webapps/mpp/home." *https://www.sandbox.paypal.com*. PayPal.
- Tags, Html. "http://www.w3schools.com/tags/tag\_html.asp." *http://www.w3schools.com/*. W3Shools.
- TeiOfCrete. "xml." In *Xml έγγραφα*, by Papadakis.
- Xml. "http://www.w3schools.com/xml/default.asp." *http://www.w3schools.com/*. W3Shools.
- xml-validate. "http://www.freeformatter.com/xml-validator-xsd.html." *http://www.freeformatter.com/*. freeformatter.

## Παράρτημα

ΕΝΟΤΗΤΑ	ΚΑΤΗΓΟΡΙΑ	ΤΕΧΝΟΛΟΓΙΑ
<b>1</b>		
Ανάπτυξη Σελίδας	Design	Html5/Css3
<b>2</b>		
DataBase	Ανάπτυξη	Xml
	Validation	Xml-Schema
<b>3</b>		
Logi/Reg System	Registration	Php/Xml
	Login	Php/Xml
	Connection Check	Php
<b>4</b>		
DataBase Parsing	Διάβασμα Xml Db	Php/Xml
	Εμφάνιση Xml Db	Php/Xml
	Αναζήτηση Προϊόντων	Php
	Αλλαγή Αποθεμάτων	Php
<b>5</b>		
addToCart	Αίτηση Κωδικού	Php
	Δημιουργία Κάρτας	Php
	Συναρτήσεις Κάρτας	Php
	Εμφάνιση Κάρτας	Php
<b>6</b>		
PayPal	Δημιουργία Λογαριασμών	PayPal Sandbox
	Αγορά μέσω PayPal	PayPal Sandbox
	Ιστορικό Συναλλαγών	PayPal Sandbox

## Παρουσίαση(PowerPoint)

[Ηλεκτρονικο εμποριο.pptx](#)

### Περίληψη

Τα τελευταία χρόνια έχει αναπτυχθεί μια υπηρεσία για την διευκόλυνση των καταναλωτών αλλά και εμπόρων. Η υπηρεσία αυτή ονομάζεται ηλεκτρονικό εμπόριο μέσω της οποίας πωλητής και αγοραστής μπορούν να συναλλάσσονται μέσω του διαδικτύου.

Με τον όρο του ηλεκτρονικού εμπορίου αναφερόμαστε σε μια διαδικασία συναλλαγής η ανταλλαγής πίσω από έναν ηλεκτρονικό υπολογιστή διαφόρων προϊόντων η υπηρεσιών, η οποία συναλλαγή πραγματοποιείται μέσω του διαδικτύου. Με το πέρασμα των χρόνων η ταχύτητες με τις οποίες εξελίσσεται η υπηρεσία αυτή παγκοσμίως είναι ραγδαίες και υπάρχει αξιοπιστία.

Ο πιθανός καταναλωτής πλέον μπορεί να ανατρέξει σε οποιαδήποτε ιστοσελίδα της αρέσκειας του και να επιλέξει απο μια σειρά εκατομμυρίων προϊόντων η υπηρεσιών προς παραγγελία σε απομακρυσμένα καταστήματα του εσωτερικού ή και του εξωτερικού.

Επίσης με τη βοήθεια του ηλεκτρονικού εμπορίου μπορεί ο καταναλωτής να κάνει συγκρίσεις τιμών για ένα συγκεκριμένο προϊόν ή υπηρεσία μέσα απο μία σειρά δεκάδων καταστημάτων.

Το ηλεκτρονικό εμπόριο πέρα απο τις συναλλαγές μεταξύ επιχειρήσεων και καταναλωτών εξυπηρετεί και άλλους σκοπούς.Μερικά είδη του εμπορίου αυτού είναι μεταξύ καταναλωτών/πολιτών προς κυβερνητικούς φορείς(C2G), εμπόριο επιχειρήσεων προς κυβερνητικούς φορείς(B2G), όπως επίσης και επιχειρήσεις προς άλλες επιχειρήσεις.

Σκοπός της εργασίας αυτής είναι η δημιουργία και η ανάπτυξη μιας ιστοσελίδας ώστε να μπορέσουν να καλυφθούν οι ανάγκες μιας κάβας η οποία επιθυμεί να εμπορεύεται τα προϊόντα της ηλεκτρονικά.

Για της ανάγκες της κάβας λοιπόν έπρεπε να αναπτυχθεί μια υπηρεσία ηλεκτρονικού εμπορίου ώστε να μπορεί να προμηθεύει τους πελάτες της με προϊόντα της επιθυμίας τους πίσω απο έναν ηλεκτρονικό υπολογιστή μέσω του διαδικτύου.

Για την ανάπτυξη της ιστοσελίδας αυτής χρησιμοποιήθηκαν τεχνολογίες όπως HTML5,CSS3,PHP.

Επίσης η τεχνολογία XML για την αποθήκευση δεδομένων όπως λογαριασμοί χρηστών αλλά και η βάση δεδομένων που αποτελείται από τα προϊόντα που παρέχει η εταιρεία.

Με αυτό το τρόπο ο πελάτης μπορεί να ικανοποιήσει την παραγγελία του μέσω του διαδικτύου ανατρέχοντας μέσα από μια σειρά προϊόντων που προμηθεύει η συγκεκριμένη κάβα. Για την δημιουργία της σελίδας αυτής έπρεπε αρχικά να μελετηθούν όλες οι πιθανές τεχνολογίες που θα μπορούσαν να χρησιμοποιηθούν.

Ο σχεδιασμός της σελίδας έγινε με τη χρήση της html5 και css3. Η html είναι μία γλώσσα σήμανσης και είναι η κύρια γλώσσα για την δημιουργία ιστοσελίδων. Η γλώσσα αυτή αναπτύσσεται υπό μορφή στοιχείων μέσα σε ετικέτες τα οποία στοιχεία μπορεί να είναι κείμενο, εικόνες, πίνακες κλπ. Οι ετικέτες στην html είναι προκαθορισμένες, δηλαδή για κάθε χρήση απαιτείται συγκεκριμένη ετικέτα και σχεδόν κάθε ετικέτα που ανοίγει πρέπει να κλείνει με το ίδιο όνομα ετικέτας.

Αρχικά δημιουργήθηκε ο κεντρικός κορμός της ιστοσελίδας οποίος αποτελείται από τρία βασικά μέρη, το header το οποίο περιγράφει το όνομα της σελίδας μας καθώς και πληροφορίες για την εταιρία όπως τηλέφωνο και διεύθυνση καθώς επίσης και τον λογαριασμό του χρήστη όπως και το καλάθι των αγορών. Επίσης το κεντρικό μέρος το οποίο ουσιαστικά είναι η βάση κάθε σελίδας μας και τέλος το footer όπου περιλαμβάνονται διάφορες πληροφορίες και σύνδεσμοι εξυπηρέτησης για τους χρήστες μας.

Για τη μορφοποίηση της σελίδας μας χρησιμοποιήθηκε η css3 η οποία ορίζει την εμφάνιση των στοιχείων της html αλλά και την διάταξη τους κάνοντας πιο προσιτή την εμφάνιση του html εγγράφου.

Για την καλύτερη εξυπηρέτηση των πελατών της εταιρίας αναπτύχθηκε το σύστημα λογαριασμού όπου ο κάθε πελάτης μπορεί να δημιουργήσει τον προσωπικό του λογαριασμό στη σελίδα και να συνδεθεί με αυτόν.

Η χρήση λογαριασμού προσφέρει στους πελάτες τη δυνατότητα να προσθέτουν προϊόντα στην wishlist τους καθώς επίσης και να λαμβάνουν προσφορές σχετικά με προϊόντα. Για την εταιρεία είναι ένας εύκολος τρόπος να πιστοποιεί τα στοιχεία των πελατών της όπως διευθύνσεις, τηλέφωνα και email και να μπορεί να επικοινωνήσει μαζί τους για οποιοδήποτε λόγο.

Για τη δημιουργία λογαριασμού ο χρήστης απαιτείται να συμπληρώσει το ονοματεπώνυμο του, τηλέφωνο, διεύθυνση, mail καθώς επίσης ένα username και password. Όλα τα στοιχεία της φόρμας είναι απαιτούμενα για την δημιουργία του λογαριασμού. Για να συνδεθεί ο χρήστης αφού έχει δημιουργήσει τον προσωπικό του λογαριασμό απαιτείται να συμπληρώσει το username και το password του.

Για την λειτουργικότητα του λογαριασμού θα πρέπει το περιεχόμενο της σελίδας μας να είναι δυναμικό και για τον λόγο αυτό χρειάζεται η php γλώσσα. Επίσης για την αποθήκευση των στοιχείων του χρήστη χρησιμοποιείται η xml τεχνολογία.

Η xml τεχνολογία χρησιμοποιήθηκε επίσης για τη δημιουργία και ανάπτυξη της βάσης δεδομένων. Όλα μας τα προϊόντα αποθηκεύονται σε ένα και μοναδικό xml έγγραφο μέσω του οποίου το κάθε προϊόν περιγράφεται απο τα στοιχεία που το προσδιορίζουν. Η xml όπως και η html είναι και αυτή γλώσσα σήμανσης. Η ανάπτυξη της είχε σκοπό να καλύψει κάποιες ελλείψεις της HTML. Και αυτή η γλώσσα χρησιμοποιεί ετικέτες. Στην xml όλες οι ετικέτες πρέπει να κλείνουν και επίσης τα xml έγγραφα είναι εύκολα προσπελάσιμα απο τους υπολογιστές, μιας και περιγράφονται όλες οι πληροφορίες απο την ετικέτα. Αποτελεί έναν εύκολο τρόπο για τη δημιουργία μίας βάσης δεδομένων που μπορεί να χρησιμοποιηθεί σε ένα οποιοδήποτε ηλεκτρονικό κατάστημα.

Επίσης χρησιμοποιήσαμε και το xml schema το οποίο λειτουργεί με σκοπό να ορίσει τη δομή του xml εγγράφου μας και να δηλωθούν οι τύποι των δεδομένων που χρησιμοποιούνται.

Η μετάφραση και εμφάνιση του xml εγγράφου μας πραγματοποιείται με τη βοήθεια της php. Η php είναι γλώσσα δημιουργίας ιστοσελίδων με δυναμικό περιεχόμενο η οποία επεξεργάζεται από έναν server (apache) ώστε το περιεχόμενο να παράγεται σε πραγματικό χρόνο και εμφανίζεται στην ιστοσελίδα υπό μορφή html. Για τη λειτουργία της php στην ιστοσελίδα κάθε αρχείο απαιτείται να έχει επέκταση .php και έναν τοπικό server μέσω του οποίου θα τρέξει. Ο apache server χρησιμοποιείται πλέον ευρέως απο το διαδίκτυο και υποστηρίζει εξ ορισμού την php γλώσσα.

Ως τοπικός server χρησιμοποιήθηκε ο wamp server ο οποίος υποστηρίζει τον apache όπως και την php για σελίδες δυναμικού περιεχομένου. Τα αρχικά του σημαίνουν windows apache mySQL php και είναι μία εύκολη και φτηνή λύση να δημιουργήσει κανείς δυναμικού περιεχομένου σελίδες.

Χρησιμοποιούνται διάφορες συναρτήσεις της για να μπορέσει να διαβαστεί το έγγραφο του καταλόγου και για να εμφανιστούν τα στοιχεία αυτά υπό τη μορφή πίνακα στη σελίδα μας. Τα προϊόντα εμφανίζονται με συνθήκες πάνω στις οποίες επιλέγουμε εμείς τον τρόπο που θα εμφανιστούν με βάση κάποιο όνομα ή τη κατηγορία που ανήκει κάποιο προϊόν.

Ένα απο τα σημαντικότερα μέρη ενός ηλεκτρονικού καταστήματος είναι η δυνατότητα να μπορεί ο πελάτης να προσθέτει τα προϊόντα της επιθυμίας του στο καλάθι αγορών. Αυτό πραγματοποιείται μέσω της `php` και των `php sessions` που θα κρατούν την πληροφορία στον `server`.

Το `php session` είναι ένας μηχανισμός που μπορεί να κρατά πληροφορίες στον `server` όπως για παράδειγμα να ελέγχει την ταυτότητα ενός πελάτη που έχει επισκεφτεί τη σελίδα μας και να κρατά πληροφορίες για αυτόν.

Κάθε φορά που θέλουμε να κρατάμε πληροφορία για κάποιον λογαριασμό ή για κάποιο προϊόν πχ. στην κάρτα αγορών μας χρησιμοποιούμε ένα `session` το οποίο κρατά τις πληροφορίες αυτές προσωρινά στον `server`. Το `session` χρησιμοποιεί την τεχνολογία των `cookies`.

Αφού λοιπόν ο πελάτης επιλέξει από μία σειρά δεκάδων προϊόντων που προμηθεύει το κατάστημά μας έχει τη δυνατότητα να προσθέσει τα προϊόντα αυτά στο καλάθι αγορών του με τελικό προορισμό την παραγγελία και αγορά αυτών.

Τελικός στόχος της δημιουργίας του ηλεκτρονικού μας καταστήματος δεν είναι άλλο απο την ηλεκτρονική αγορά των προϊόντων μας από τους πελάτες του καταστήματος. Για να επιτευχθεί αυτό και σκεπτόμενοι την ασφαλέστερη συναλλαγή του καταστήματος με τους πελάτες χρησιμοποιήθηκε η `PayPal` και για την ακρίβεια μία εφαρμογή της το `PayPal Sandbox`.

Το `PayPal Sandbox` είναι όπως αναφέρθηκε μία εφαρμογή της `PayPal` η οποία δουλεύει σε συνδυασμό με το `PayPal for Developers` και σκοπό έχει να δημιουργήσει ένα εικονικό περιβάλλον ασφαλών συναλλαγών ενός η περισσότερων καταστημάτων για την εξυπηρέτηση των πελατών τους.

Η δημιουργία ενός πραγματικού λογαριασμού είναι απαραίτητη μέσω του οποίου δουλεύει η εφαρμογή. Μέσω του `PayPal Developers` δημιουργούνται όλοι αυτοί οι λογαριασμοί (εικονικοί) που θα χρησιμοποιηθούν για τις συναλλαγές του καταστήματος και των πελατών του.

Για την δημιουργία του κάθε λογαριασμού συμπληρώνουμε στοιχεία όπως `mail`, `ονοματεπώνυμο`, `διεύθυνση` και όλα όσα απαιτεί η φόρμα για τη δημιουργία του. Επίσης ζητείται να δημιουργήσουμε έναν εικονικό τραπεζικό λογαριασμό ο οποίος γίνεται αυτόματα και έναν `16-digit PayPal` κωδικό για `MasterCard` ή `Visa` ανάλογα την επιλογή του πελάτη. Οσον αφορά τον λογαριασμό του καταστήματος ή του πωλητή, δίνεται η δυνατότητα ακόμη να επιλέξουν ονομασία του καταστήματος καθώς και τον τύπο του, τα είδη που εμπορεύεται και άλλα.

Μετά την επιτυχή ολοκλήρωση μίας συναλλαγής η παραγγελία εμφανίζεται τόσο στον προσωπικό λογαριασμό του κάθε πελάτη όσο και σε αυτόν του πωλητή. Ένας administrator ο οποίος έχει πρόσβαση στον λογαριασμό του καταστήματος θα πρέπει να ελέγχει σε τακτά χρονικά διαστήματα για τυχόν παραγγελίες που γίνονται και για άλλες που περιμένουν και δεν έχουν εκπληρωθεί ακόμη. Με αυτόν τον τρόπο ολοκληρώνονται οι συναλλαγές των καταναλωτών του καταστήματος μας με ασφάλεια και ταχύτητα με τη βοήθεια του διαδικτύου.