

# Παναγιώτης Ντουντούκαλης

## E18116

### 1<sup>η</sup> Ενότητα

#### **Ψηφιακές Εκτυπώσεις(Print on demand)**

Η επιχείρηση που θα προσπαθήσω να χτίσω βασίζεται στο print on demand. Το print on demand είναι ένας τρόπος που οποιοσδήποτε (είτε ιδιώτης είτε επιχείρηση) μπορεί να μπει στην ηλεκτρονική μου σελίδα και να προμηθευτεί το προϊόν που επιθυμεί με την στάμπα της προτίμησής του.

**Ανάγκη καταναλωτών:** Ανάγκη αγοραστών να κάνουν την αγορά τους από το σπίτι τους χωρίς κόπο και χωρίς περιορισμούς στην τοποθεσία ή την ώρα.

Θέλουν είτε για προσωπική χρήση(δώρο, ενθύμιο, βιβλίο) είτε για επαγγελματική χρήση(επαγγελματικές κάρτες, σήματα της εταιρίας κτλ).

**Υπάρχουσα αγορά:** Η αγορά μπορεί να βελτιωθεί πολύ γιατί έχω σκεφτεί διάφορες καινοτόμες ιδέες(διαθέσιμο προϊόν στην Ελλάδα κτλ).

**Χαρακτηριστικά προϊόντος:** Προσιτή τιμή προϊόντων, προϊόντα κατάλληλα για το περιβάλλον, χαμηλό βάρος χωρίς εξτρά χρέωση για μεταφορικά, ανθεκτικά προϊόντα.

Τρόποι βελτίωσης υπάρχοντων προϊόντων: Καινοτόμες ιδέες για αλλαγή υλικών των προϊόντων ώστε να στοχεύουμε σε όσο το δυνατόν πιο οικολογικά προϊόντα, livechat με παράπονα των καταναλωτών ώστε να υπάρχει 1-1 επικοινωνία με τον πελάτη και να συζητείται οτιδήποτε χρειάζεται βελτίωση.

## Ανταγωνιστική ανάλυση:

Τύπος		Primary	Primary	Primary	Primary	Primary	Primary	Primary
Όνομα εταιρίας	My shop	Print-easy.gr	Gelato.com	Mystis.gr	Shopify.com	Mrcopy.gr	Printhouse.gr	Wix.com
Έδρα	Ελλάδα	Ελλάδα						
Τιμές	5-50 €							
Στοχευμένο κοινό	Όλες οι ηλικιακές ομάδες	Επιχειρήσεις						
Τι προσφέρει	Μπλούζες, φούτερ, υλικά για επιχειρήσεις	Βιβλία, επαγγ. κάρτες, αφίσες, φόρμες τιμολογίων, μπλοκ αποδείξεων						
Βασικό πλεονέκτημα	Οικολογικά προϊόντα	Έδρα στην Ελλάδα, κατάλληλη για επιχειρήσεις						
Υποστήριξη	Τηλέφωνο, email, live chat	Τηλέφωνο, email						
Στρατηγική μάρκετινγκ	Διαφημίσεις στο ίντερνετ	-						

Μεταφορικά	Δωρεάν	Μόνο για Αττική 5 € Εκτός Αττικής εξτρά χρέωση
eco friendly	Ναι	-

## SWOT ανάλυση

Strengths	Weaknesses
<ol style="list-style-type: none"> <li>1. Live chat</li> <li>2. Eco friendly προϊόντα</li> <li>3. Δωρεάν μεταφορικά σε όλη την Ελλάδα</li> <li>4. Κατάλληλο και για επιχειρήσεις και για απλή χρήση</li> <li>5. Προσωπικό γνώριμο με τις νέες τεχνολογίες και τις μόδες του σήμερα</li> <li>6. Διαφημίσεις της εταιρίας μέσω άλλων σελίδων στο διαδίκτυο</li> <li>7. Χαμηλό κόστος με καλή ποιότητα</li> <li>8. Πλήρης ψηφιοποίηση της αγοράς του προϊόντος</li> <li>9. Επιστροφή χρημάτων αν ο πελάτης δεν μείνει ικανοποιημένος</li> </ol>	<ol style="list-style-type: none"> <li>1. Υπερφόρτωση των παραγγελιών και πιθανή καθυστέρηση παραλαβής</li> <li>2. Όχι μεταφορά εκτός Ελλάδας</li> <li>3. Έλλειψη πόρων για επαρκή διαφήμιση</li> <li>4. Νέα εταιρία οπότε ο κόσμος δεν μας εμπιστεύεται</li> <li>5. Πρώτες ύλες ακριβές σε σχέση με τα απλά προϊόντα</li> </ol>

<b>Opportunities</b>	<b>Threats</b>
<ol style="list-style-type: none"> <li>1. Συνεχής ενασχόληση σε οικολογικά προϊόντα</li> <li>2. Τάση για νέα μικρά brands που χρειάζονται εκτύπωση σε προσιτή τιμή</li> <li>3. Προσέγγιση σε χρήστες android και apple</li> <li>4. Eco friendly μεταφορικά</li> </ol>	<ol style="list-style-type: none"> <li>1. Νέες εταιρίες που εισέρχονται στον ανταγωνιστικό χώρο</li> <li>2. Εξέλιξη τεχνολογίας χρειάζεται πλήρη κατανόηση από τους εργαζόμενους</li> <li>3. Ολοένα και περισσότερο οικολογικά υλικά για την παραγωγή</li> <li>4. Πιθανή αύξηση κόστους πρώτων υλών</li> </ol>

## 2<sup>η</sup> Ενότητα

Όσον αφορά το επιχειρησιακό μοντέλο, θα γίνει χρήση πολλαπλών τύπων.

Σαν βασικό μοντέλο θα είναι το άμεσο ηλεκτρονικό marketing, καθώς ο καταναλωτής θα μπορεί να αγοράζει απευθείας από την εταιρία μου χωρίς κάποιον μεσάζοντα. Ταυτόχρονα έτσι δίνεται έμφαση στην ψηφιοποίηση της αγοράς με χρήση e-mail marketing.

Ταυτόχρονα θα χρησιμοποιείται έμμεσα ο καθορισμός τιμής από καταναλωτή καθώς θα υπάρχει επιλογή τιμής για αγορά ενός προϊόντος και θα εμφανίζονται τα προϊόντα που πληρούν τις προϋποθέσεις του καταναλωτή.

Σημαντικό επιχειρησιακό μοντέλο θα είναι το σπειροειδές marketing καθώς είμαστε μία νέα καινοτόμα επιχείρηση και θέλουμε ο κόσμος να μας μάθει όσο το δυνατόν γρηγορότερα. Έτσι θα χρησιμοποιήσουμε όσες διαφημίσεις μπορούμε για να γίνουμε viral και να κεντρίσουμε το ενδιαφέρον των καταναλωτών και των επιχειρήσεων.

Επίσης για τους νέους χρήστες θα χρησιμοποιήσουμε το μοντέλο των μεγάλων εκπτώσεων για να προσεγγίσουμε όσο το δυνατόν μεγαλύτερο αριθμό καταναλωτών οι οποίοι με την σειρά τους μπορεί να προτείνουν την σελίδα μας σε γνωστούς τους.

## **Καμβάς Επιχειρηματικού Μοντέλου**

### **1. Τμήματα Πελατών**

Επιχειρήσεις ή ατομικοί καταναλωτές που επιθυμούν προσιτά και οικολογικά prints ανεξαρτήτου φύλου και ηλικίας(Mass market).

### **2. Προτάσεις Αξίας**

Οικολογικά προϊόντα με προσωπικό print πάνω τους.  
Εστιάζεται σε όλη την Ελλάδα/

### **3. Κανάλια**

Μοντέρνο website με κατάλληλες λίστες και προϊόντα όλων των ειδών. Διαφημίσεις μέσω άλλων websites και μέσω προωθητικών e-mails.

#### **4. Σχέσεις Πελατών**

Live chat για κάθε πιθανή απορία, δωρεάν μεταφορικά, εγγύηση προϊόντων και επιστροφή χρημάτων σε περίπτωση που ο αγοραστής δεν μείνει ικανοποιημένος. Σελίδες στο Instagram και στο twitter για επικοινωνία πελατών και ελεύθερη κριτική.

#### **5. Ροές Εσόδων**

Άμεση πώληση στους καταναλωτές μέσω του e-shop.

#### **6. Βασικοί Πόροι**

Αγορά πρώτων υλών από συνεργαζόμενες εταιρίες, τύπωση προϊόντος από την εταιρία μας και διαχείριση προσωπικού της εταιρίας μας.

#### **7. Βασικές Δραστηριότητες**

Συνεχής ενασχόληση με τις νέες τεχνολογίες και μόδες(trends), οικολογικά προϊόντα, online marketplace, διαφήμιση στο διαδίκτυο.

#### **8. Βασικοί Συνεργάτες**

Συνεργαζόμενες εταιρίες για αγορά πρώτων υλών, websites που συνεργαζόμαστε για να διαφημιστούμε μέσω αυτών.

#### **9. Διάρθρωση Κόστους**

Αγορά πρώτων υλών, κατασκευή προϊόντων, διαφήμιση, εξοπλισμός, υπάλληλοι εταιρίας, φόροι κράτους.

## 3<sup>η</sup> Ενότητα

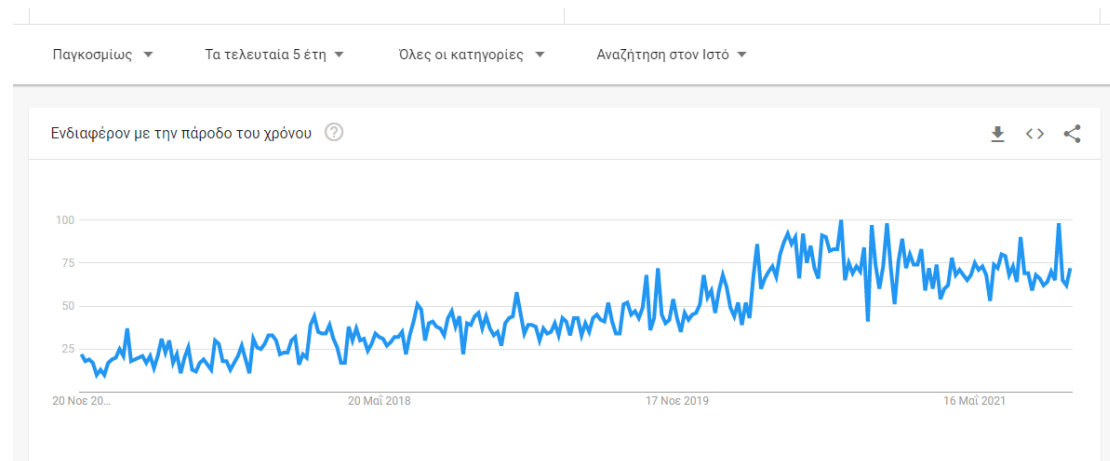
<https://trends.google.com/trends/explore?q=print%20on%20demand&geo=GR>

Οι μεθοδολογίες που θα χρησιμοποιήσω για να επικυρώσω την επιχείρηση είναι οι εξής:

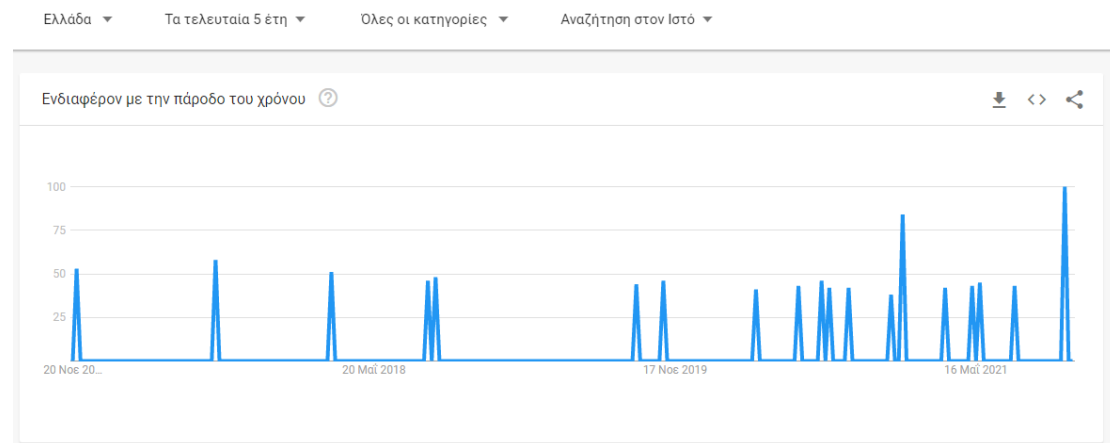
- **Έρευνα στο διαδίκτυο**

Για αρχή θα χρησιμοποιήσω την σελίδα του **Google Trends** για να δω κατά πόσο η επιχειρηματική μου ιδέα έχει τάσεις ανόδου τον τελευταίο χρόνο. Για την αναζήτηση θα χρησιμοποιήσω τον όρο **print on demand**.

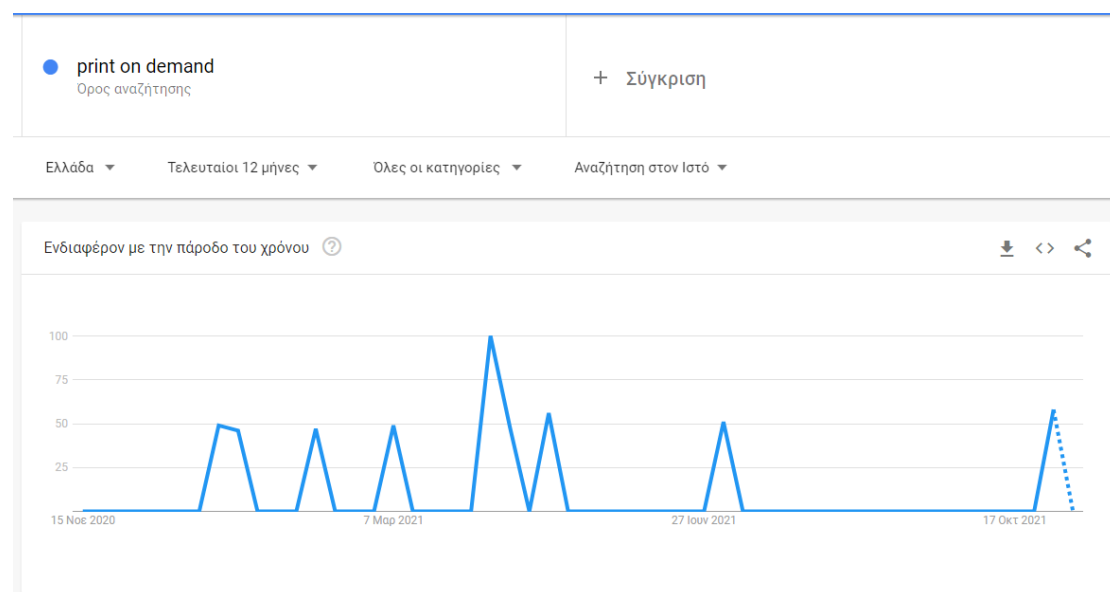
Σε παγκόσμιο επίπεδο τα τελευταία 5 έτη υπάρχει το εξής γράφημα:



Παρατηρείται μία σταθερή αύξηση μέχρι περίπου τα τέλη του 2020 και μετά μία στασιμότητα αλλά σε καλό επίπεδο μέχρι σήμερα. Οπότε σε παγκόσμιο επίπεδο η ζήτηση είναι καλή, αλλά τώρα πρέπει να ελέγξω πως είναι στην Ελλάδα αφού η επιχείρηση θα βρίσκεται μόνο στη χώρα μας.



Τα στοιχεία που είναι διαθέσιμα για την Ελλάδα δεν είναι εντελώς πλήρη, αλλά μπορούμε να κατανοήσουμε την βασική ιδέα. Παρατηρούμε ότι τα τελευταία 5 χρόνια υπάρχουν περίοδοι που η ζήτηση ανεβαίνει, με την κορύφωσή της να βρίσκεται στα τέλη του 2021. Συγκεκριμένα:



Παρατηρούμε αρκετά υψηλά επίπεδα ζήτησης, και το σημαντικό είναι ότι στο τέλος του γράφου βλέπουμε μια άνοδο που αυτό σημαίνει ότι είναι η κατάλληλη στιγμή για την επιχείρησή μας να μπει στην αγορά.

## • Δημιουργία Σελίδας Προορισμού

Σαν επόμενο βήμα είναι η δημιουργία της σελίδας προορισμού της επιχείρησης, με σκοπό να μπορούν οι πελάτες να συλλέγουν σχόλια και κριτικές σχετικά με τα προϊόντα της επιχείρησης μου και με απώτερο σκοπό να προσελκύσω νέους πελάτες.



- **Διαφημιστική Καμπάνια**

Επίσης θα χρησιμοποιήσω αρκετούς πόρους μου στην διαφημιστική καμπάνια ώστε να προσπαθήσω να κάνω viral την επιχείρησή μου και να την μάθει όσο το δυνατόν περισσότερος κόσμος γίνεται.

Οι διαφημίσεις θα είναι κυρίως σε άλλες σελίδες (Facebook, Google Ads, Instagram κτλ.) και θα συνδυαστούν με την χρήση ενός μικρού επεξηγηματικού βίντεο που θα αναλύει τον σκοπό της επιχείρησης και την οικολογική μας ευαισθησία.

## **Ονομασία: Printor**

Η ονομασία της εταιρίας μας είναι: Printor, η οποία βγαίνει από το print οπότε ο κόσμος μπορεί να καταλάβει εύκολα με τι ασχολούμαστε αν μας ακούει για πρώτη φορά.

## **Σύνθημα:**

**Printor: Οικολογική εκτύπωση, πώς ζούσατε χωρίς αυτό;**

Για το σύνθημα χρησιμοποίησα την σελίδα <https://www.shopify.com/> για την εύρεση μιας ιδέας. Ο στόχος του συγκεκριμένου συνθήματος είναι να παρακινήσω τον κόσμο στην οικολογική εκτύπωση κάνοντάς τον να κατανοήσει ότι είναι είμαστε η κατάλληλη εταιρία για αυτόν.

## Λογότυπο:



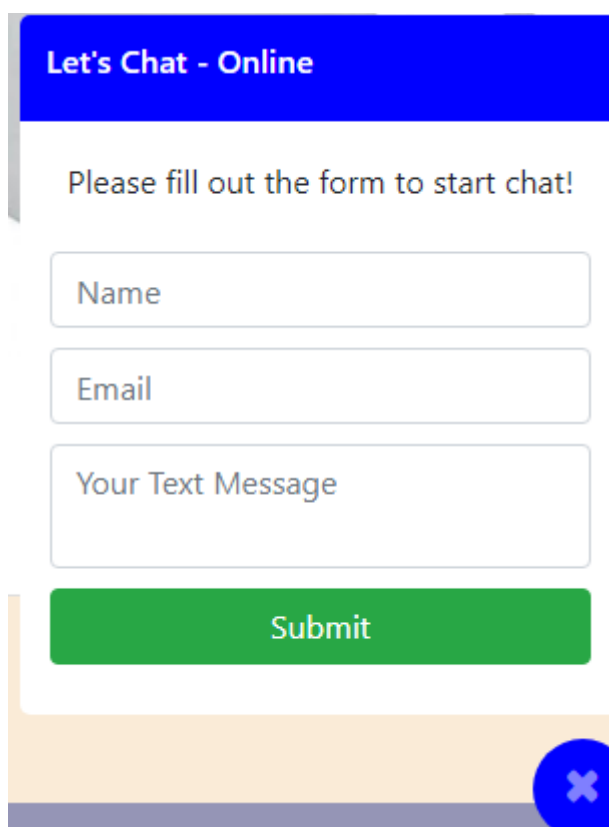
Για το λογότυπο χρησιμοποίησα μπλε και πράσινα χρώματα ώστε να συνδυάσω την οικολογική μεριά του πράσινου χρώματος και την εμπιστοσύνη του μπλε. Το σημαντικό είναι η προσθήκη του συμβόλου της οικολογίας πάνω από το P για να δείξω την στάση της επιχείρησής μου προς την οικολογία. Παράλληλα χρησιμοποίησα την γραμματοσειρά: High Tower Text η οποία δεν έχει απότομες γωνίες και συμβαδίζει με τον τόνο των χρωμάτων.

Τέλος θα πρέπει να λάβω υπόψη τις κατάλληλες άδειες που χρειάζομαι για να γίνει νόμιμη η επιχείρησή μου (Λήψη Αριθμού Φορολογικού Μητρώου, Υποβολή αίτησης για άδειες επιχειρήσεων).

## 4<sup>η</sup> Ενότητα

Από τις 12 τάσεις του ηλεκτρονικού εμπορίου, στο ηλεκτρονικό μου κατάστημα υλοποίησα τις εξής:

- **Αύξηση πράσινου καταναλωτισμού:** Η βασικότερη τάση που έχω υλοποιήσει είναι η αύξηση του πράσινου καταναλωτισμού. Η όλη ιδέα της οικολογικής εκτύπωσης είναι η αγορά ανακυκλώσιμων και φιλικών προς το περιβάλλον προϊόντων. Η κύρια ασχολία μας είναι να βρούμε τα πιο βιώσιμα προϊόντα στην καλύτερη δυνατή τιμή. Αυτό είναι κάτι που επηρεάζει πολύ θετικά τον καταναλωτή, μιας και ολοένα και περισσότεροι άνθρωποι ευαισθητοποιούνται πάνω σε αυτό το θέμα. Παρέχουμε online αποδείξεις και ενημερώσεις σε όλους τους πελάτες μας για τον τρόπο παραγωγής των προϊόντων μας και για την εξασφάλιση της ποιότητάς τους.
- **Chatbots:**



Let's Chat - Online

Please fill out the form to start chat!

Name

Email

Your Text Message

Submit

X

Άλλη μία βασική προσθήκη του καταστήματός μου είναι η προσθήκη του chatbot. Ο χρήστης μπορεί πολύ γρήγορα να ρωτήσει οτιδήποτε τον απασχολεί και το chatbot μπορεί να του παρέχει την σωστή απάντηση άμεσα. Έτσι η συναλλαγή γίνεται πολύ πιο άμεσα και αποδοτικά. Επίσης έχουν πρόσβαση στα δεδομένα του καταστήματος, όπως απόθεμα προϊόντος και παρέχουν και πάλι άμεσα την βοήθεια τους στον διαθέσιμο πελάτη.

- **Αγορές μέσω κινητού:** Πολύ βασική είναι και η διαθεσιμότητα αγοράς μέσω κινητού. Η σελίδα είναι πλήρως responsive τόσο σε κινήτα όσο και σε tablets οπότε ο χρήστης μπορεί να εισέλθει από την συσκευή της αρεσκείας του. Η χρησιμοποίηση του κινητού για την αγορά ενός προϊόντος είναι πολύ πιο εύκολη και γρήγορη από έναν σταθερό υπολογιστή και σε συνδυασμό ότι η συντριπτική πλειοψηφία των καταναλωτών πλέον διαθέτει ένα κινητό, είναι σχεδόν απαραίτητο η σελίδα μας να αποκρίνεται στις συγκεκριμένες αγορές
- **Ανταπόκριση πελατών σε βίντεο:**

## 5<sup>η</sup> Ενότητα

### Απαιτήσεις χρηστών:

Πλευρά διαχειριστή	Πλευρά τελικών χρηστών
Ποιες τάσεις θα χρησιμοποιηθούν;	Τι φίλτρα αναζήτησης θα χρησιμοποιηθούν;
Με ποιον τρόπο θα γίνεται η εισαγωγή των πελατών;	Ποιοι είναι οι διαθέσιμοι τρόποι πληρωμής;
Με ποιον τρόπο θα γίνεται η επικύρωση των καταχωρήσεων;	Ποιες είναι οι διαθέσιμες γλώσσες;
Ποιες επιλογές προσβασιμότητας θα είναι διαθέσιμες;	Ποια media θα είναι διαθέσιμα για την διευκόλυνση της παρουσίασης των προϊόντων;
Με ποιον τρόπο θα γίνεται η καταχώρηση των hosts?	Πως γίνεται η δημιουργία ενός λογαριασμού;
	Μπορεί να γίνει διαγραφή λογαριασμού;
	Υπάρχουν διαθέσιμες κριτικές;

### User personas:

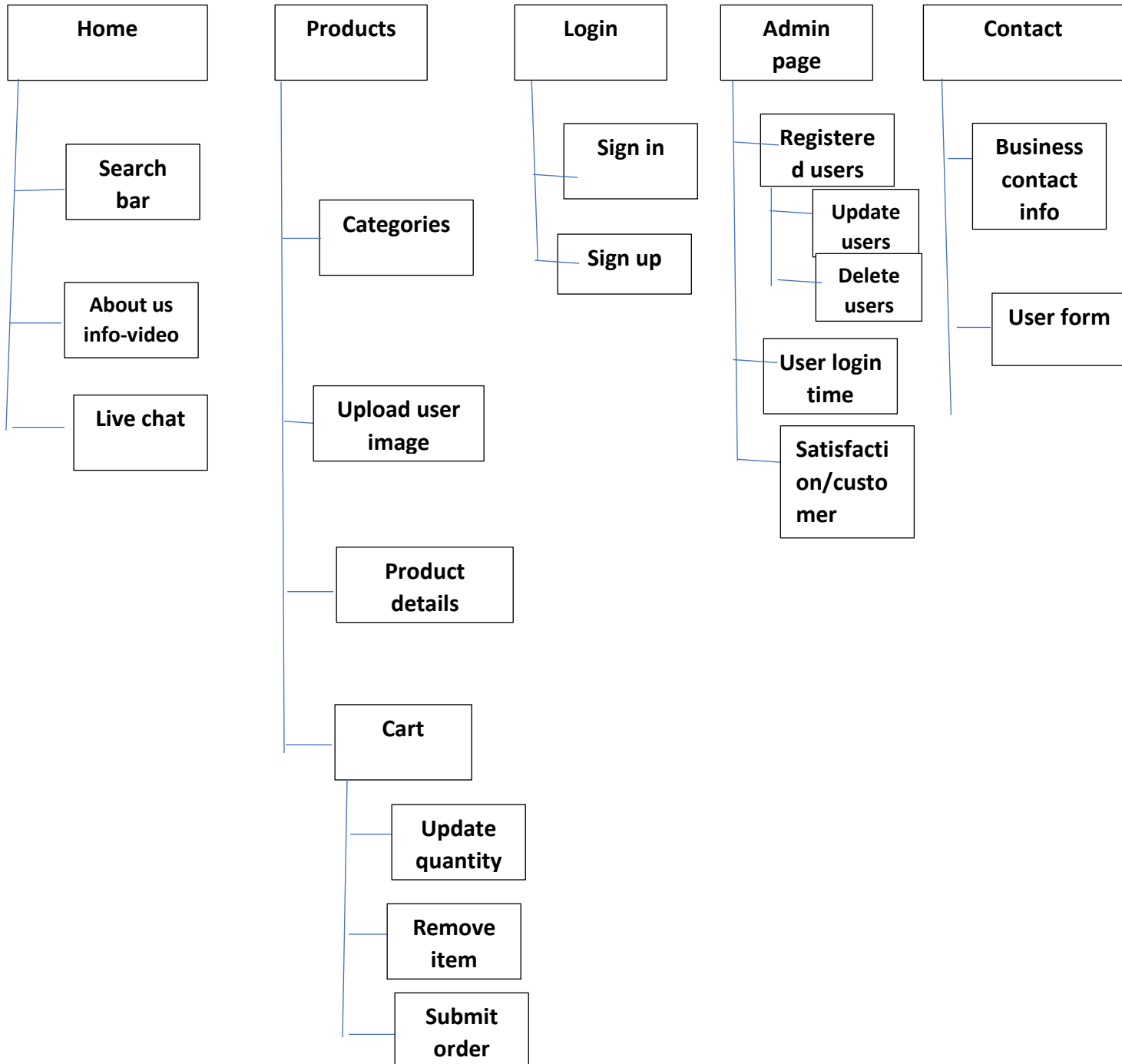
- 1) Κωνσταντίνος Ιωάννου: Είναι 43 ετών, έχει την δική του επιχείρηση που ασχολείται με online προμήθειες βασικών πρώτων υλών, επιθυμεί να τυπώνει διαφημιστικά φυλλάδια σε συμβατή τιμή και να μπορεί να επικοινωνεί άμεσα με τον πάρωχο, υποστηρίζει τα περιβαλλοντολογικά προϊόντα και αναζητεί τρόπους για να βοηθάει τον πράσινο τρόπο ζωής μέσα στην επιχείρησή του.
- 2) Δήμητρα Παύλου: Είναι 38 χρονών, ζει μαζί με τον σύζυγό της και τα δύο τους παιδιά, έχουν έναν σκύλο 2 χρονών και θέλει να τυπώνει φωτογραφίες όλης της οικογένειας σε κούπες, πίνακες, μπλούζες και καπέλα. Ακολουθεί φανατικά τα hashtags #savetheplanet #greenlife #organic και άλλα, ανακυκλώνει καθημερινά και είναι χορτοφάγος. Επιζητεί μία εταιρία που να μπορεί να τυπώνει φιλικά προς το περιβάλλον προϊόντα.
- 3) Γιώργος Παππάς: Είναι 25 ετών, ζει μόνος του, του αρέσει να βρίσκεται με τους φίλους του στο σπίτι του και να παίζουν ηλεκτρονικά παιχνίδια. Είναι λάτρης του Netflix. Θέλει να τυπώσει διάφορες αφίσες με χαρακτήρες από παιχνίδια και

σειρές. Τον ενδιαφέρει να είναι προσιτές τιμές και να έχει πρόσβαση σε ένα μοντέρνο site που να διαθέτει καινοτομίες και να ξεχωρίζει από τα υπόλοιπα.

## User stories:

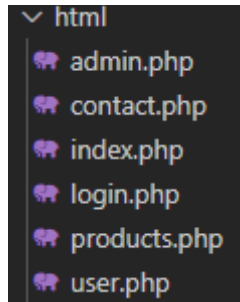
- Ως πελάτης, θέλω να μπορώ να προσθέσω την πιστωτική μου κάρτα και να πληρώνω άμεσα για το προϊόν μου
- Ως πελάτης, θέλω να βλέπω κριτικές για τα διαθέσιμα προϊόντα, αλλά και να μπορώ να καταχωρώ τις δικές μου κριτικές
- Ως πελάτης, θέλω να γνωρίζω ότι τα προσωπικά μου δεδομένα και στοιχεία αποθηκεύονται με ασφάλεια χωρίς κίνδυνο διαρροής σε τυχόν επιτιθέμενους
- Ως πελάτης, θέλω να παρουσιάζονται τα κατάλληλα πιστοποιητικά για τα προϊόντα που είναι φιλικά προς το περιβάλλον
- Ως πελάτης, θέλω να υπάρχουν εύκολα προσβάσιμα φίλτρα αναζήτησης για να γίνεται η αγορά μου πιο γρήγορη και αποδοτική
- Ως πελάτης θέλω ένα μοντέρνο eshop με διαδραστικά μέσα και παρουσίαση των προϊόντων
- Ως πελάτης θέλω να υποστηρίζεται η αγγλική γλώσσα γιατί μπορεί να μην γνωρίζω Ελληνικά
- Ως πελάτης θέλω να μπορώ να βλέπω που ακριβώς βρίσκεται η παραγγελία μου
- Ως πελάτης θέλω να ικανοποιείται η δωρεάν επιστροφή αν δεν μείνω ικανοποιημένος από το προϊόν που αγόρασα τις πρώτες 30 ημέρες

## Visual sitemap:



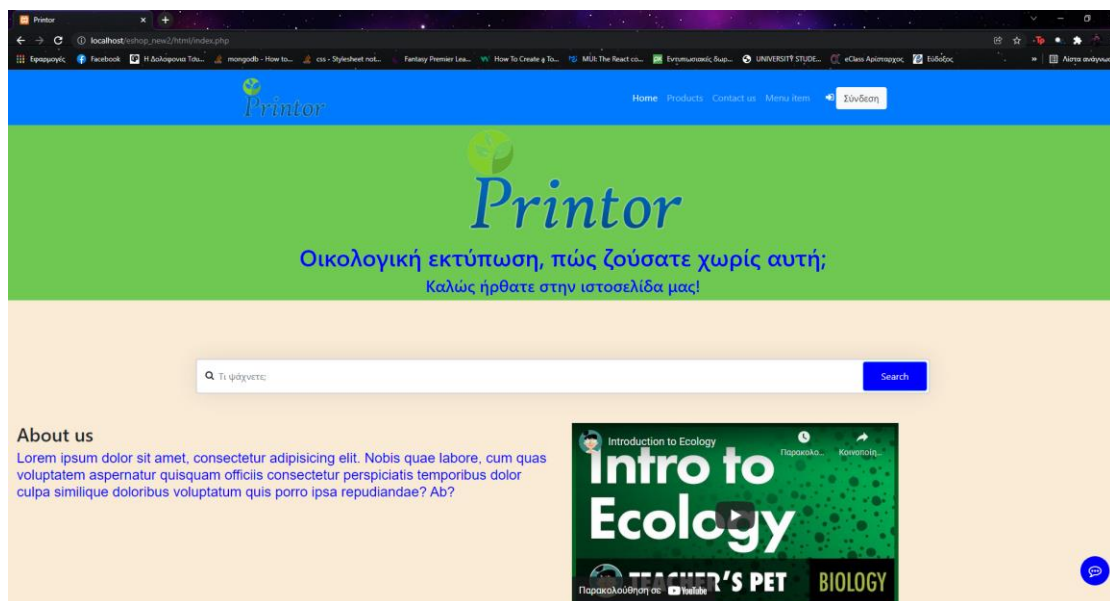
## 6<sup>η</sup> Ενότητα

Στον φάκελο html βρίσκονται όλες οι βασικές σελίδες που είναι διαθέσιμες για τον χρήστη. Έχουν μετατραπεί σε php αρχεία για να μπορεί να γραφτεί php κώδικας μέσα στην html.

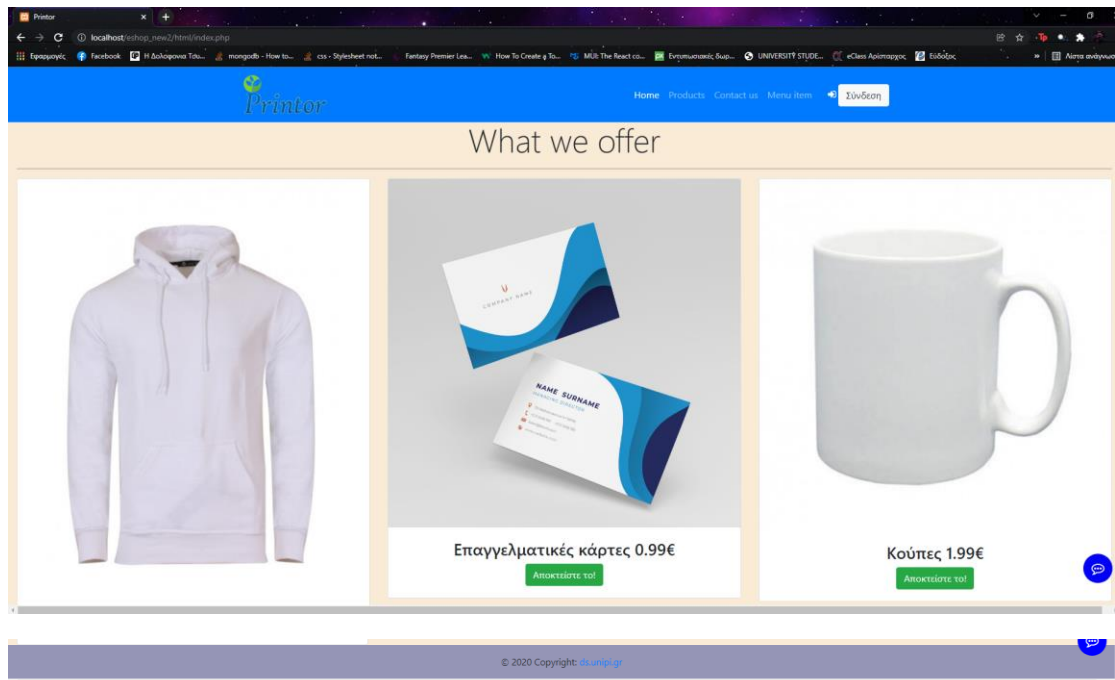


### Index.php

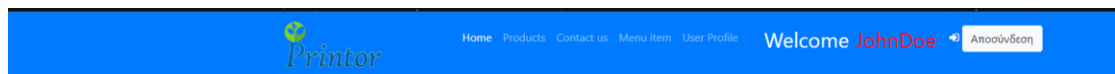
Η βασική σελίδα του eshop όπου εκεί μπαίνει ο χρήστης όταν εισέρχεται στο σαιτ:







Στην κορυφή της σελίδας υπάρχει ένα sticky navigation bar με τις βασικές σελίδες που μπορεί να πάει ο χρήστης. Σε περίπτωση που κάνει σύνδεση και είναι επιτυχής, φαίνεται το εξής:



Παρατηρούμε ότι έχει προστεθεί στο navigation bar η σελίδα User profile και ένα μήνυμα καλωσορίσματος με το username του χρήστη. Τέλος αντί για το κουμπί της σύνδεσης βλέπουμε το κουμπί της αποσύνδεσης. Στον κώδικα η υλοποίηση έγινε ως εξής:

```

39 <!-- Navigation -->
40 <nav id="navbar_top" class="navbar navbar-expand-lg navbar-dark bg-primary sticky-top">
41 <div class="container">
42 <a class="navbar-brand" href="#"></a>
43 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#main_nav">
44 <span class="navbar-toggler-icon"></span>
45 </button>
46 <div class="collapse navbar-collapse" id="main_nav">
47 <ul class="navbar-nav ml-auto"><!--ml-auto για να είναι τα styloixeia stin deksia meria-->
48 <li class="nav-item active"><a class="nav-link" href="#"> Home </a></li>
49 <li class="nav-item"><a class="nav-link" href="/products.php"> Products </a></li>
50 <li class="nav-item"><a class="nav-link" href="/contact.php"> Contact us </a></li>
51 <li class="nav-item"><a class="nav-link" href="#"> Menu item </a></li>
52
53 <?php
54 if (isset($_SESSION["username"]) && isset($_SESSION["password"])){ ?>
55 <li class="nav-item user-nav-item"><a class="nav-link" href="user.php"> User Profile </a></li>
56 <p class="session_welcome">Welcome <div class="session_username_welcome"><?php echo $_SESSION["username"]></div></p>
57 <div class="login_button">
58 <i class="fa fa-sign-in" aria-hidden="true"></i>
59 <a href="../../../php/logout.php" class="btn btn-light" role="button" aria-pressed="true">Αποσύνδεση</a>
60 </div>
61 <?php }
62 else{ ?>
63 <div class="login_button">
64 <i class="fa fa-sign-in" aria-hidden="true"></i>
65 <a href="/login.php" class="btn btn-light" role="button" aria-pressed="true">Σύνδεση</a>
66 </div>
67 <?php }
68 ?>
69
70 </ul>
71 </div>
72 </div>
73 </nav>

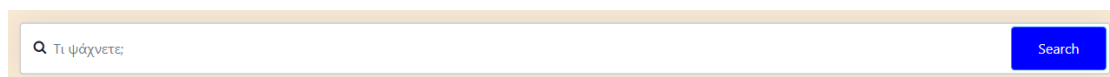
```

Στην γραμμή 53 ελέγχω αν έχει οριστεί το Session του username και το password(τα οποία ορίζονται κατά το login) και αν έχουν οριστεί εμφανίζω τα εξτρά περιεχόμενα που ανέφερα πιο πάνω.



```
<!--Welcome-->
<div class="welcome container-fluid padding">
  <div class="row text-center ">
    <div class="col-12">
      
      <h1>Οικολογική εκτύπωση, πώς ζούσατε χωρίς αυτή;</h1>
    </div>
    <div class="col-12">
      <h3>Καλώς ήρθατε στην ιστοσελίδα μας!</h3>
    </div>
  </div>
</div>
```

Στην συνέχεια έχω ένα welcome με την εικόνα της εταιρίας μου και με το slogan που δημιούργησα νωρίτερα.



```
<!--search bar-->
<div class="row height d-flex justify-content-center align-items-center">
  <div class="col-md-8">
    <div class="search"> <i class="fa fa-search"></i> <input type="text" class="form-control" placeholder="Τι ψάχνετε;"> <button class="btn btn-primary">Search</button> </div>
  </div>
</div>
```

Έπειτα υλοποίησα ένα απλό search bar στο οποίο ο χρήστης μπορεί να αναζητήσει για συγκεκριμένα προϊόντα.

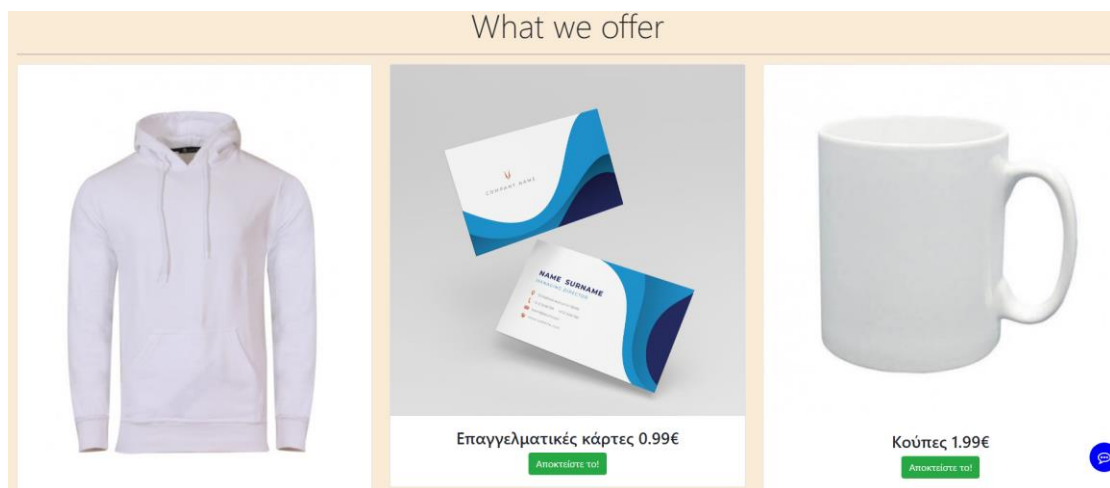
## About us

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nobis quae labore, cum quas voluptatem aspernatur quisquam officiis consectetur perspicatis temporibus dolor culpa similique doloribus voluptatum quis porro ipsa repudiandae? Ab?



```
<div class="container-fluid padding">
  <div class="row padding">
    <div class="col-lg-6">
      <h2>About us</h2>
      <div class="about_us_content">
        <h4 class="about_us_content">
          Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nobis quae labore, cum quas voluptatem aspernatur quisquam officiis consectetur perspicatis temporibus dolor culpa similique doloribus voluptatum quis porro ipsa repudiandae? Ab?</h4>
        <br>
      </div>
    </div>
    <div class="col-lg-6">
      <iframe width="560" height="315" src="https://www.youtube.com/embed/GInFyldVH4" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>
    </div>
  </div>
</div>
```

Έπειτα υλοποίησα ένα about us page και ένα intro video για την εταιρία μου και τα έβαλα στην ίδια γραμμή με την χρήση των col.



Έπειτα παρουσιάζω στον χρήστη τι προσφέρουμε σε μία γραμμή με τρεις ίσες στήλες. Αν ο χρήστης επιθυμεί να αποκτήσει ένα προϊόν μπορεί να πατήσει στο πράσινο κουμπί κάτω από το προϊόν και να μεταφερθεί στην σελίδα των προϊόντων.

Στο τέλος της σελίδας έχω υλοποιήσει ένα απλό footer που παρουσιάζει τα βασικά στοιχεία της εταιρίας.

Let's Chat - Online

Please fill out the form to start chat!

Your Text Message

Submit

✕

```

<!--live chat-->
<input type="checkbox" id="check"> <label class="chat-btn" for="check"> <i class="fa fa-commenting-o comment"></i> <i class="fa fa-close close"></i> </label>
<div class="wrapper">
  <div class="header">
    <h6>Let's Chat - Online</h6>
  </div>
  <div class="text-center p-2"> <span>Please fill out the form to start chat!</span> </div>
  <div class="chat-form"> <input type="text" class="form-control" placeholder="Name"> <input type="text" class="form-control" placeholder="Email"> <textarea
class="form-control" placeholder="Your Text Message"></textarea> <button class="btn btn-success btn-block">Submit</button> </div>
</div>

```

Μία ακόμα λειτουργία του σάιτ είναι το livechat. Για την υλοποίηση του χρησιμοποίησα τα εξής css:

```

.chat-btn {
  position: fixed;
  right: 14px;
  bottom: 30px;
  cursor: pointer
}

.chat-btn .close {
  display: none
}

#check:checked~.chat-btn i {
  display: block;
  pointer-events: auto;
  transform: rotate(180deg)
}

#check:checked~.chat-btn .comment {
  display: none
}

.chat-btn i {
  font-size: 22px;
  color: ■rgb(255, 255, 255) !important
}

```

Όπως παρατηρούμε στην αρχή έχω όλο το περιεχόμενο κρυφό στον χρήστη μέχρι να πατήσει πάνω το κουμπί και όταν το πατήσει εμφανίζεται.

## Products.php

# Upload image

Επιλογή αρχείου

Δεν επιλέχθηκε κανένα αρχείο.

Αρχικά επιτρέπω στον χρήστη να επιλέξει την φωτογραφία του που θέλει να τυπώσω στο προϊόν του. Η υλοποίηση γίνεται με την βοήθεια της JavaScript:

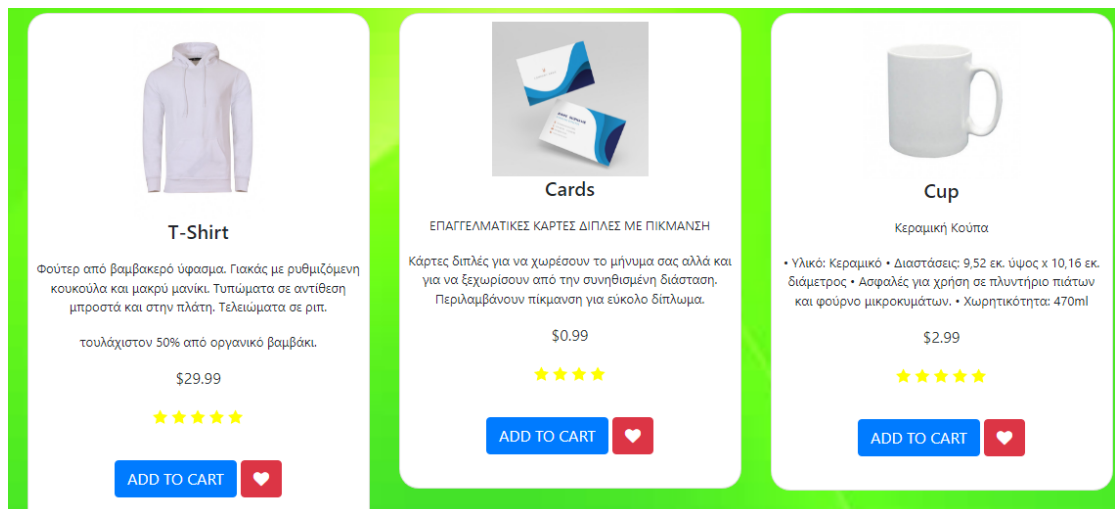
```

<h1 class="text-center">Upload image</h1>
<!--make user upload his photo to print-->
<div class="upload_image mx-auto">
  <input type="file" id="image_input" accept="image/png, image/jpeg">

  <div id="display_image"></div>
</div>

<script>
  const image_input = document.querySelector("#image_input");
  image_input.addEventListener("change", function() {
    const reader = new FileReader();
    reader.addEventListener("load", () => {
      const uploaded_image = reader.result;
      document.querySelector("#display_image").style.backgroundImage = `url(${uploaded_image})`;
    });
    reader.readAsDataURL(this.files[0]);
  });
</script>

```



Έπειτα παρουσιάζω τα προϊόντα του καταστήματός μου, με τις βασικές πληροφορίες τους, όπως το όνομα τους, τα βασικά τους χαρακτηριστικά, τα υλικά που είναι φτιαγμένα, την τιμή τους, τα αστέρια αξιολόγησης των χρηστών, την προσθήκη στα αγαπημένα και την προσθήκη στο καλάθι.

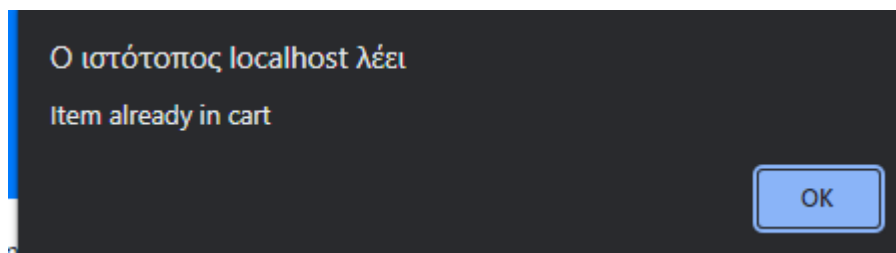
Όταν ο χρήστης πατήσει το κουμπί ADD TO CART τότε γεμίζει το καλάθι του. Αρχικά το καλάθι είναι άδειο:



Με την προσθήκη του χρήστη:



Όπου μπορεί να επιλέξει τα συνολικά προϊόντα που θέλει να αγοράσει και η τιμή ανανεώνεται κάθε φορά που προσθέτει ένα προϊόν. Αν πατήσει το κουμπί remove το αντίστοιχο προϊόν διαγράφεται από το καλάθι. Δεν αφήνω τον χρήστη να βάλει τιμή κάτω του 1 στην ποσότητα και σε περίπτωση που βάλει ανανεώνεται αυτόματα η τιμή πίσω στο 1. Αν πατήσει προσθήκη στο καλάθι ένα προϊόν το οποίο βρίσκεται ήδη στο καλάθι, ενημερώνεται με το αντίστοιχο μήνυμα:



Η υλοποίηση του καλαθιού έγινε με την χρήση JavaScript στο αρχείο store.js.

## Store.js

Για το remove button έγινε η εξής υλοποίηση:“

```
var removeCartItemButtons = document.getElementsByClassName('btn-danger')
console.log(removeCartItemButtons)

for(var i=0; i<removeCartItemButtons.length; i++){
    var button = removeCartItemButtons[i]
    button.addEventListener('click', removeCartItem)
}

function removeCartItem(event){
    var buttonClicked = event.target
    buttonClicked.parentElement.parentElement.remove() /*remove div named cart-row */
    updateCartTotal()
}
```

Όπου αποθηκεύω στην μεταβλητή `removeCartItemButtons` το `btn-danger` που παίρνω από το `phr` αρχείο και με επανάληψη καλώ την συνάρτηση `removeCartItem` με μεταβλητή την θέση του `button` που αποθήκευσα. Έπειτα κάνω `remove` όλο το `div` και ανανεώνω το συνολικό ποσό της αγοράς του χρήστη καλώντας το `updateCartTotal`:

```
function updateCartTotal(){
    var cartItem = document.getElementsByClassName('cart-items')[0] /*get only first item instead of whole array*/
    var cartRows = cartItem.getElementsByClassName('cart-row')
    var total=0
    for(var i=0; i<cartRows.length; i++){
        var cartRow = cartRows[i]
        var priceElement = cartRow.getElementsByClassName('cart-price')[0]
        var quantityElement = cartRow.getElementsByClassName('cart-quantity-input')[0] /*get price and quantity spans from products.html */

        var price = parseFloat(priceElement.innerText.replace('$', '')) /*get value from price and remove dollar sign */
        var quantity = quantityElement.value
        total = total + ( price*quantity )
    }
    total = Math.round(total*100) / 100 /*make total have only two decimal numbers */
    /*get element of total price from products.html and update it */
    document.getElementsByClassName('cart-total-price')[0].innerText = '$' + total
}
```

Αρχικά αποθηκεύω τις μεταβλητές `cartItem` και `cartRows` οι οποίες παίρνουν τις αντίστοιχες τιμές του `phr` αρχείου και την μεταβλητή `total` η οποία θα κρατάει τη συνολική τιμή. Επαναληπτικά για κάθε προϊόν που βρίσκεται στο καλάθι παίρνω τα δεδομένα που χρειάζονται (την τιμή και την ποσότητα) και υπολογίζω το `total` και το εμφανίζω στον χρήστη.

```
/*change price of cart element when changing quantity */
var quantityInputs = document.getElementsByClassName('cart-quantity-input')
for(var i=0; i<quantityInputs.length; i++){
    var input = quantityInputs[i]
    input.addEventListener('change', quantityChanged)
}

function quantityChanged(event){
    var input = event.target
    /*check if quantity input is valid */
    if( isNaN(input.value) || input.value<=0 ){
        input.value=1
    }
    updateCartTotal()
}
```

Για να αλλάζει η συνολική τιμή όταν ο χρήστης αλλάζει την ποσότητα του προϊόντος που επιθυμεί να αγοράσει υλοποίησα τις δύο παραπάνω συναρτήσεις. Εδώ γίνεται ο έλεγχος αν η τιμή που βάζει ο χρήστης είναι κάτω του 1 και την ανανεώνει με το 1 και ενημερώνει την νέα τιμή με την συνάρτηση `updateCartTotal`.



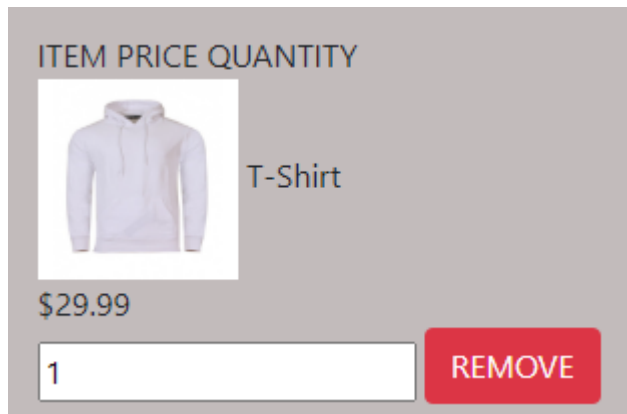
```

/*add to cart button */
var addToCartButtons = document.getElementsByClassName('shop-item-button')
for(var i=0; i<addToCartButtons.length; i++){
    var button = addToCartButtons[i]
    button.addEventListener('click', addToCartClicked)
}

function addToCartClicked(event){
    var button = event.target
    var shopItem = button.parentElement.parentElement
    var title = shopItem.getElementsByClassName('item-name')[0].innerText
    var price = shopItem.getElementsByClassName('item-price')[0].innerText
    var image = shopItem.getElementsByClassName('item-image')[0].src
    console.log(title, price, image)

    addItemToCart(title, price, image)
    updateCartTotal()
}

```



Για να δείχνω στον χρήστη τις παραπάνω πληροφορίες υλοποιώ τις παραπάνω συναρτήσεις όπου παίρνω και πάλι τα κατάλληλα δεδομένα από το ρηρ και τα εμφανίζω στο καλάθι.

```

/*add item on cart function */
function addItemToCart(title, price, image){
    var cartRow = document.createElement('div')
    var cartItems = document.getElementsByClassName('cart-items')[0]

    /*check if cart-item is already in cart */
    var cartItemNames = cartItems.getElementsByClassName('cart-item-title')
    for(var i=0; i<cartItemNames.length; i++){
        if (cartItemNames[i].innerText == title) {
            alert("Item already in cart")
            return
        }
    }

    var cartRowContents = `<div class="cart-item cart-column">
        
        <span class="cart-item-title">${title}</span>
    </div>
    <span class="cart-price cart-column">${price}</span>
    <div class="cart-quantity cart-column">
        <input class="cart-quantity-input" type="number" value="1">
        <button class="btn btn-danger" type="button">REMOVE</button>
    </div>`

    cartRow.innerHTML = cartRowContents
    cartItems.append(cartRow) /*puts it at the end of cartItems */

    cartRow.getElementsByClassName('btn-danger')[0].addEventListener('click', removeCartItem) /*for new items that weren't there when the page loaded */
    cartRow.getElementsByClassName('cart-quantity-input')[0].addEventListener('change', quantityChanged)
}

```

Τέλος για να προσθέσω το προϊόν στο καλάθι δημιουργώ το κενό div cartRow όπου μέσα προσθέτω την μεταβλητή cartRowContents στην οποία γράφω HTML κώδικα και το προσθέτω στο div.

Για την αξιολόγηση του καταστήματος υλοποίησα ένα modal το οποίο είναι ένα popup που εμφανίζεται όταν ο χρήστης πατήσει στο κουμπί PURCHASE μέσα στο καλάθι. Τότε του εμφανίζεται το εξής:

## Ευχαριστούμε για την προτίμησή σας! ✕

Πριν φύγετε, θα θέλαμε να μας πείτε την εμπειρία σας με την σελίδα μας!

Πόσο ευχαριστημένος μείνατε με την σελίδα μας;



Όπου μπορεί να πατήσει όποιο κουμπί του ταιριάζει περισσότερο για την εμπειρία του με το κατάστημά μου. Τα δεδομένα αυτά περνάνε στο admin.php όπου μπορεί να δει τους συνολικούς χρήστες για το κάθε κουμπί. Η υλοποίηση έγινε ως εξής:

```
<button class="btn btn-primary btn-purchase button_popup" type="button" data-target="#popup_modal" data-toggle="modal">PURCHASE</button>
<!--modal-->
<div class="modal" id="popup_modal">
  <div class="popup-header">
    <h1>Ευχαριστούμε για την προτίμησή σας!</h1>
    <button data-dismiss="modal">&times</button>
  </div>
  <h3>Πριν φύγετε, θα θέλαμε να μας πείτε την εμπειρία σας με την σελίδα μας!</h3>
  <h4>Πόσο ευχαριστημένος μείνατε με την σελίδα μας;</h4>
  <div class="row">
    <form action="../../php/modal.php" method="post">
      <div class="col-sm">
        <button name="happy_user" type="submit" class="btn btn-success"><i class="fa fa-smile-o" aria-hidden="true"></i></button>
      </div>
      <div class="col-sm">
        <button name="mediocre_user" type="submit" class="btn btn-warning"><i class="fa fa-meh-o" aria-hidden="true"></i></button>
      </div>
      <div class="col-sm">
        <button name="sad_user" type="submit" class="btn btn-danger"><i class="fa fa-frown-o" aria-hidden="true"></i></button>
      </div>
    </form>
  </div>
</div>
```

Βλέπουμε ότι μέσα στο modal έχω μία φόρμα με 3 κουμπιά η οποία στέλνεται στο αρχείο modal.php. Με την βοήθεια του data-toggle μπορώ να χειρίζομαι το αν είναι

ενεργοποιημένο το modal ή όχι. Αν ενεργοποιηθεί τότε έχω φτιάξει κατάλληλα το CSS για να φαίνεται όμορφα στον χρήστη.

## Modal.php

```
$collection = $db->modal;
echo "Collection selected successfully<br><br>";

$cursor = $collection->find();
foreach($cursor as $document){

    if ( isset( $_POST['happy_user'] )) {
        $total_users=$document["happy_users"]+1;
        echo $total_users;
        $collection->updateOne(array("happy_users"), array('$set'=>array("happy_users"=>$total_users)));
    }

    else if ( isset( $_POST['mediocre_user'] )) {
        $total_users=$document["mediocre_users"]+1;
        $collection->updateOne(array('$set'=>array("mediocre_users"=>$total_users)));
    }

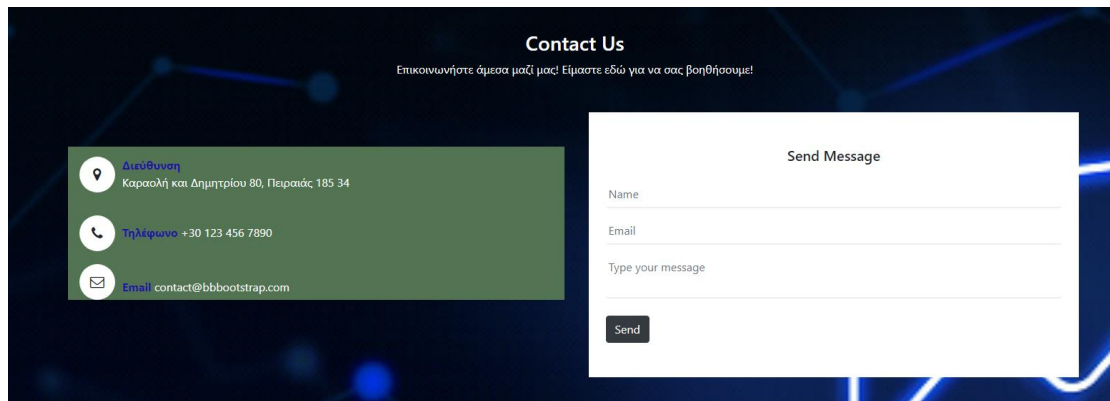
    else if ( isset( $_POST['sad_user'] )) {
        $total_users=$document["sad_users"]+1;
        $collection->updateOne(array('$set'=>array("sad_users"=>$total_users)));
    }

}

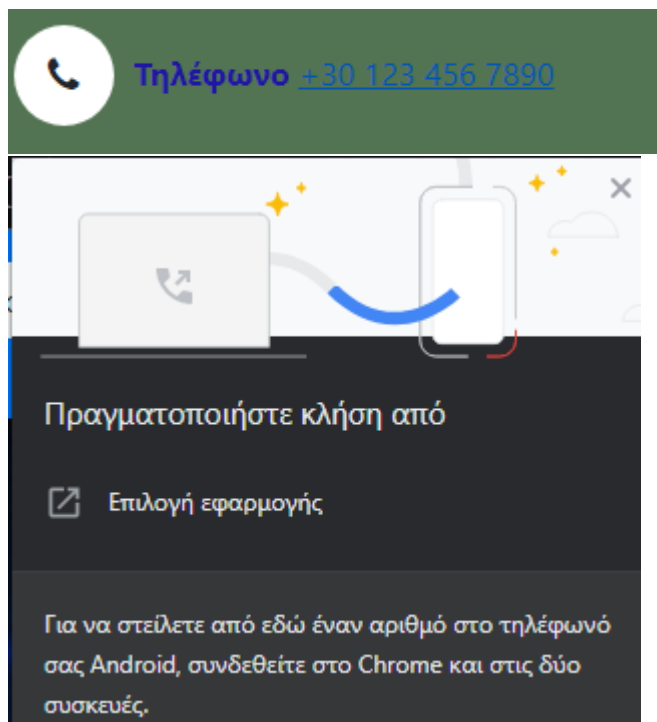
header('Location: ../html/products.php');
```

Τώρα στην βάση δεδομένων δημιουργώ το collection modal και μέσα στο modal κάνω το query find για να βρω ποιο κουμπί έχει πατηθεί από τον χρήστη και ανάλογα με την επιλογή του αυξάνω κατά ένα την κατάλληλη στήλη. Τέλος τον ανακατευθύνω στο products.php. Τα δεδομένα αυτά μπορεί να τα δει ο admin στο admin.php.

## Contact.php



Στο contact.php έχω στην αριστερή στήλη την διεύθυνση, το τηλέφωνο και το email της επιχείρησής μου. Ο χρήστης μπορεί να επικοινωνήσει άμεσα μαζί μας αν πατήσει το τηλέφωνο ή το email:



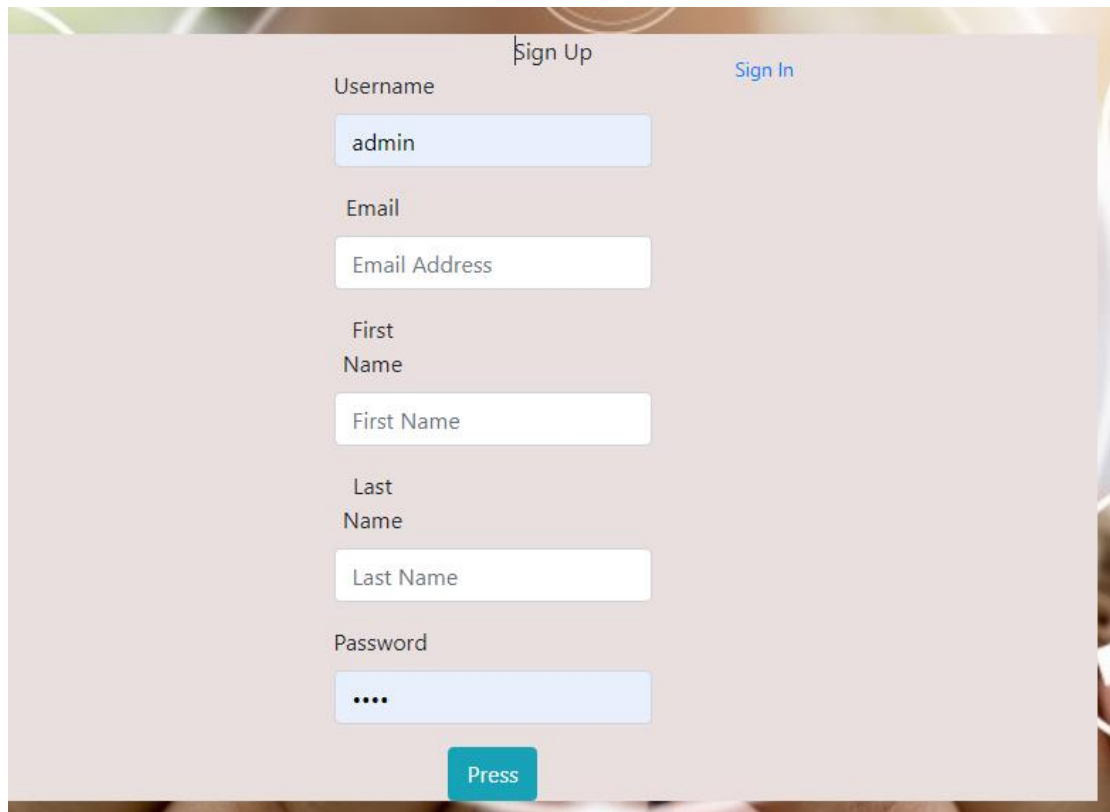
The image shows a dark-themed email composition window. At the top is a ribbon with tabs: 'Format' (selected), 'Insert', 'Draw', and a dropdown arrow. To the right are icons for 'Discard' and 'Send'. Below the ribbon is a toolbar with a font size selector (showing 'A'), a text alignment icon, a dropdown menu currently showing 'Heading 1', a settings gear icon, and an 'Undo' button. The email header fields are as follows: 'From: panosntountou7@hotmail.com' with an external link icon; 'To: contact@bbbootstrap.com;' with a person icon and 'Cc & Bcc' text; and an empty 'Subject' field. At the bottom left, it says 'Sent from Mail for Windows'.

Στην δεξιά στήλη έχω μια απλή φόρμα που ο χρήστης μπορεί να στείλει το μήνυμα που επιθυμεί στην σελίδα μας.

## Login.php

Όταν ο χρήστης πατήσει στο login button μεταφέρεται αυτόματα στην σελίδα login.php. Εδώ έχει δύο επιλογές, είτε να κάνει sign in αν έχει ήδη προφίλ είτε να κάνει sign up αν είναι νέος χρήστης:

The image shows a login form with a light pink background. At the top, it says 'Sign In' in black and 'Forgot password?' in blue. There are two input fields: the first is labeled 'username' with a person icon to its left, and the second is labeled 'password' with a lock icon to its left. Below these fields is a checkbox labeled 'Remember me'. A green button with the text 'Press' is centered below the checkbox. At the bottom, there is a horizontal line and the text 'Don't have an account! Sign Up Here' in blue.



Η εναλλαγή στις δύο φόρμες γίνεται με τον εξής τρόπο:

Έχω δύο divs, το **loginbox** και το **signupbox** για το **login** και για το **sign up** αντίστοιχα.

Το **signupbox** είναι αρχικά κρυφό με την ιδιότητα `display:none` και όταν ο χρήστης πατήσει πάνω στο link **Sign up here** τότε εκτελείται η παρακάτω εντολή:

```
<a href="#" onClick="$('#loginbox').hide(); $('#signupbox').show()">>  
  Sign Up Here  
</a>
```

Η οποία κρύβει το **loginbox** και εμφανίζει το **signupbox**.

Με όμοιο τρόπο γίνεται η αλλαγή από το **sign up** στο **sign in**.

Όσον αφορά την διαδικασία του **sign up**. Μέσα στο div του **signupbox** έχω δημιουργήσει μία φόρμα με την οποία στέλνω τα δεδομένα με **POST** στο αρχείο **signup.php**.

## Signup.php

Στο **signup.php** δημιούργησα την πρώτη φορά την βάση μου με όνομα **signedUsers** και το **collection users**, μέσα στο οποίο θα αποθηκεύονται όλοι ο εγγεγραμμένοι χρήστες.

```

11 // $collection = $db->createCollection("users");
12 //echo "Collection created successfully<br><br>";
13
14 $collection = $db->users;
15 echo "Collection selected successfully<br><br>";

```

Τώρα κάθε φορά που ένας χρήστης θα στέλνει την φόρμα του στο signup παίρνω τα στοιχεία του και τα αποθηκεύω στις αντίστοιχες μεταβλητές:

```

18 $username = $_POST['signup_username'];
19 $email = $_POST['signup_email'];
20 $firstName = $_POST['firstname'];
21 $lastName = $_POST['lastname'];
22 $password = $_POST['passwd'];
23

```

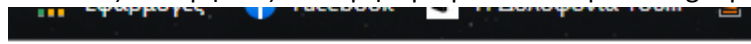
Έπειτα εκτελώ το κατάλληλο query για την εισαγωγή των στοιχείων στο collection:

```

$document = array(
    "username" => "$username",
    "email" => "$email",
    "firstName" => "$firstName",
    "lastName" => "$lastName",
    "password" => "$password" );
$collection->insertOne($document);

```

Και τέλος του εμφανίζω ένα μήνυμα με ένα link για το login.php



Connected successfully  
 Database selected successfully  
  
 Collection selected successfully  
  
 Inserted  
  
[Back to login](#)

Όσον αφορά τώρα το login div. Έχω πάλι μία φόρμα η οποία στέλνεται στο αρχείο login\_authentication.php

## Login\_authentication.php

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

Εδώ αρχικά ορίζω το session\_start για να μπορέσω να αποθηκεύσω τα στοιχεία του χρήστη και να μπορώ να τα εμφανίσω και στις υπόλοιπες σελίδες στην συνέχεια. Στην συνέχεια και αφού κάνω τις συνδέσεις για να επιλέξω το collection users, αποθηκεύω τα inputs του χρήστη στα κατάλληλα SESSIONS και την μεταβλητή signedUsers η οποία θα χρησιμοποιηθεί για τον έλεγχο του αν βρέθηκε ο αντίστοιχος χρήστης:

```
19 $_SESSION["username"] = $_POST['username'];  
20 $_SESSION["password"] = $_POST['password']; //check signup.php for other sessions  
21 $signedUser = true;
```

Στην συνέχεια εκτελώ το query για να ψάξω στην βάση των δεδομένων και να βρω αν υπάρχει χρήστης με αυτό το username και password.

```
$cursor = $collection->find( array('username'=>$_SESSION["username"],  
                                     'password'=> $_SESSION["password"]) );
```

```
foreach ($cursor as $document){  
    $signedUser = false;  
}
```

Επαναληπτικά σε όλα τα documents του collection αν βρεθεί αντιστοιχεία αλλάζω το signedUser σε false.


Αν η μεταβλητή μετά την επανάληψη δεν έχει αλλάξει, τότε εμφανίζω στον χρήστη ότι έβαλε λάθος στοιχεία:

```
Connected successfully  
Database selected successfully  
  
Collection selected successfully  
Wrong inputs  
Back to Login Page
```

Αλλιώς ελέγχω αν ο χρήστης είναι ο admin ή όχι για να τον μεταφέρω στο admin.php(το οποίο δεν είναι προσβάσιμο από κανέναν απλό χρήστη) ή πίσω στο index.php. Επίσης προσθέτω έναν timer που ελέγχει τον συνολικό χρόνο σε λεπτά που ο admin είναι συνδεδεμένος στο σύστημα:

```
$_SESSION ['timer'] = time(); //for user to show time
```

Έστω ότι συνδέεται ο χρήστης Panos με κωδικό panos στο σύστημα:

 [Home](#) [Products](#) [Contact us](#) [Menu item](#) [User Profile](#) Welcome **Panos** [Αποσύνδεση](#)



Μεταφέρεται στο index.php και βλέπουμε ότι έχει εμφανιστεί η επιλογή για το user profile του, το μήνυμα welcome Panos και το κουμπί Αποσύνδεση. Αν πατήσει στο κουμπί μεταφέρεται στο αρχείο logout.php:

```
<?php
    session_start();
    session_destroy();
    echo 'You have been logged out. <a href="../html/index.php">Go back</a>';
?>
```

Το οποίο τελειώνει το session και εμφανίζει το αντίστοιχο μήνυμα στον user.

---

You have been logged out. [Go back](#)

## Admin.php

Όταν ο χρήστης βάλει το username του admin("admin") και τον κωδικό του admin("pass") μεταφέρεται στην αρχική σελίδα του admin, η οποία είναι η εξής:

hello admin

Registered users:

?>

Username	Password	First name	Last name	Email	Delete User	Edit User
Username: admin	Password: pass	First name: admin	Last name: admin	Email: admin@test.com	DELETE	EDIT
Username: myusername	Password: mypass	First name: myfirst	Last name: mylast	Email: myemail	DELETE	EDIT
Username: a	Password: a	First name: a	Last name: a	Email: a	DELETE	EDIT
Username: JohnDoe	Password: JohnDoe	First name: John	Last name: Doe	Email: johndoe@test.com	DELETE	EDIT
Username: aaa	Password: aaa	First name: aaa	Last name: aaa	Email: aaa	DELETE	EDIT
Username: Panos	Password: panos	First name: Παναγιώτης	Last name: Ντουντούκας	Email: panosntountou7@hotmail.com	DELETE	EDIT

Happy users: 0

Mediocre users: 0

Sad users: 0

Last user logged in for: 0.05 minutes

Εδώ του παρουσιάζεται ένας πίνακας με όλους τους εγγεγραμμένους χρήστες του eshop, όπου έχει πρόσβαση σε όλα τα στοιχεία που έχουν συμπληρώσει στην φόρμα τους κατά την εγγραφή τους. Η υλοποίηση έχει γίνει ως εξής:

```

28     <table name = "userDetails">
29         <tr>
30             <th>Username</th>
31             <th>Password</th>
32             <th>First name</th>
33             <th>Last name</th>
34             <th>Email</th>
35             <th>Delete User</th>
36             <th>Edit User</th>
37         </tr>
38     <?php
39     echo '<h2>Registered users:</h2><br>';
40
41     require '../vendor/autoload.php';
42     $m = new MongoDB\Driver\ MongoClient ("mongodb://127.0.0.1/");
43
44     $db = $m->signedUsers;
45
46     $collection = $db->users;
47
48     $cursor = $collection->find();
49     foreach ($cursor as $document){
50         ?>
51
52         <tr>
53             <td>
54                 Username: <?php echo $document["username"];?>
55             </td>
56             <td>
57                 Password: <?php echo $document["password"];?>
58             </td>
59             <td>
60                 First name: <?php echo $document["firstName"];?>
61             </td>
62             <td>
63                 Last name: <?php echo $document["lastName"];?>
64             </td>
65             <td>
66                 Email: <?php echo $document["email"];?>
67             </td>

```

Αρχικά γίνονται όλες οι συνδέσεις ώστε να χρησιμοποιηθεί το κατάλληλο collection για την εμφάνιση των δεδομένων μέσω της MongoDB. Τα δεδομένα αυτά παρουσιάζονται μέσα στον πίνακα επαναληπτικά(με την χρήση της γραμμής 49) και εμφανίζονται δυναμικά με την χρήση της php μέσα στην html(όπως για παράδειγμα η γραμμή 54).

Πέρα από την εμφάνιση των δεδομένων των χρηστών, ο admin έχει την επιλογή δύο κουμπιών, το DELETE και το EDIT.

### DELETE:

```
<form name="delete_form" method="post" action="../php/delete_user.php">
  <input type="hidden" name="delete_user" value="<?php echo $document["username"] ?>" />
  <button type="submit" name="delete_btn" class="btn btn-danger">DELETE</button>
</form>
```

Για να μπορεί ο χρήστης να διαγράψει κάποιον χρήστη μπορεί να πατήσει το κουμπί delete της γραμμής που επιθυμεί. Στην html σελίδα έχω δημιουργήσει μία φόρμα όπου μέσα έχει δύο στοιχεία, το button που πατάει ο admin και ένα κρυφό στον admin input με τιμή το username του κάθε χρήστη. Ο λόγος που χρησιμοποιείται αυτό το input είναι για να μπορώ να παίρνω την κατάλληλη γραμμή και να σβήνω μόνο τον συγκεκριμένο χρήστη. Αφού ο admin πατήσει το κουμπί το πρόγραμμα αυτόματα μεταφέρεται στο delete\_user.php το οποίο είναι το εξής:

```
<?php
require '../vendor/autoload.php';
$m = new MongoClient("mongodb://127.0.0.1/");

$db = $m->signedUsers;

$collection = $db->users;

if ( isset( $_POST['delete_btn'] ) ) {
    $id = $_POST['delete_user'];

    $collection->deleteMany(array("username"=>$id));
    header('Location: ../html/admin.php');
}

?>
```

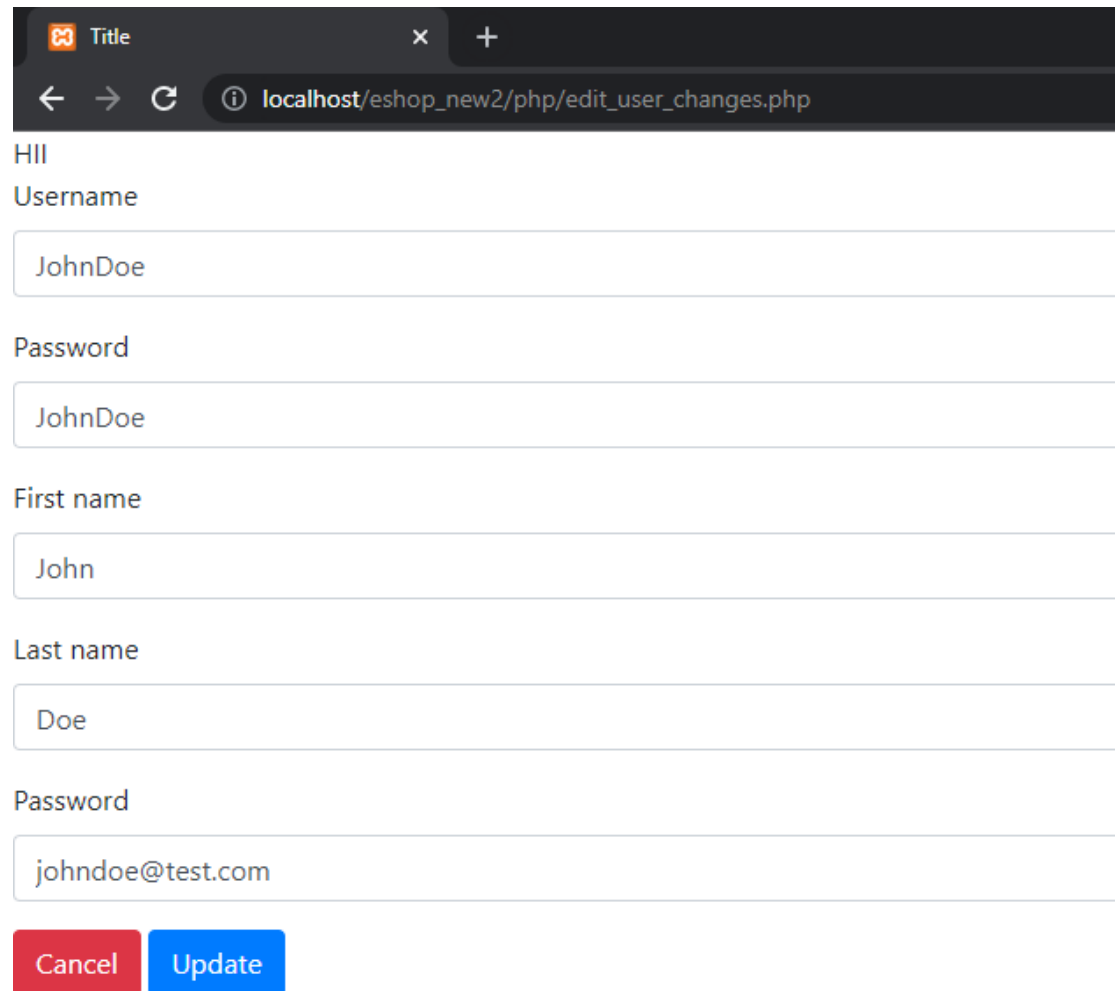
Στην ουσία εδώ γίνεται και πάλι η σύνδεση στο collection και εφόσον το button του delete έχει πατηθεί ορίζω στην μεταβλητή id το κρυφό input που πέρασα από το admin.php. Τέλος κάνω delete από το collection στην γραμμή που το id είναι το ίδιο με το username της βάσης και ανακατευθύνω αυτόματα τον admin στο admin.php.

### EDIT:

Το edit κουμπί έχει παρόμοια λογική και ο admin μπορεί να το χρησιμοποιήσει για να αλλάξει ένα ή περισσότερα δεδομένα οποιουδήποτε εγγεγραμμένου χρήστη.

```
<form name="change_form" method="post" action="../../php/edit_user_changes.php">
  <input type="hidden" name="edit_user" value="<?php echo $document["username"] ?>" />
  <button type="submit" name="edit_btn" class="btn btn-primary">EDIT</button>
</form>
```

Και πάλι έχω ένα form με την ίδια λογική του delete. Ο admin τώρα ανακατευθύνεται στο αρχείο edit\_user\_changes, το οποίο στον browser φαίνεται ως εξής:



Browser address bar: localhost/eshop\_new2/php/edit\_user\_changes.php

Form fields:

- Username: JohnDoe
- Password: JohnDoe
- First name: John
- Last name: Doe
- Password: johndoe@test.com

Buttons: Cancel, Update

(Τα δεδομένα που φαίνονται είναι τα δεδομένα του χρήστη που ο admin επέλεξε να κάνει edit). Εδώ παρουσιάζεται ένα form με όλα τα στοιχεία του συγκεκριμένου χρήστη. Ο

κώδικας είναι ο εξής:

```
require '../vendor/autoload.php';
$m = new MongoClient ("mongodb://127.0.0.1/");

$db = $m->signedUsers;

$collection = $db->users;

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

    $id = $_POST['edit_user'];
    $cursor = $collection->find(array('username'=>$id) );
    //$collection->deleteMany(array("username"=>$id));
    //header('Location: ../html/admin.php');
    //echo $collection;
    foreach ($cursor as $document)
    {
        |>

<form action="update_user.php" method="POST">

    <input type="hidden" name="edit_id" value="<?php echo $document["username"]?>">
    <div class="form-group">
        <label>Username </label>
        <input type="text" name="edit_username" value="<?php echo $document["username"]?>" class="form-control" placeholder="Enter Username">
    </div>
    <div class="form-group">
        <label>Password</label>
        <input type="text" name="edit_password" value="<?php echo $document["password"]?>" class="form-control" placeholder="Enter Password">
    </div>
    <div class="form-group">
        <label>First name</label>
        <input type="text" name="edit_firstname" value="<?php echo $document["firstName"]?>" class="form-control" placeholder="Enter First name">
    </div>
    <div class="form-group">
        <label>Last name</label>
        <input type="text" name="edit_lastname" value="<?php echo $document["lastName"]?>" class="form-control" placeholder="Enter Last name">
    </div>

    <div class="form-group">
        <label>Password</label>
        <input type="text" name="edit_email" value="<?php echo $document["email"]?>" class="form-control" placeholder="Enter Email">
    </div>

    <a href="../html/admin.php" class="btn btn-danger">Cancel</a>
    <button type="submit" name="update_btn" class="btn btn-primary">Update</button>


```

Εδώ μετά την σύνδεση στο collection ελέγχω αν έχει πατηθεί το edit button από το admin.php και αν έχει πατηθεί τότε θέτω στην μεταβλητή id το αποτέλεσμα του κρυφού input(που περιέχει το username του χρήστη). Έπειτα με χρήση του foreach εμφανίζω όλα τα στοιχεία του μέσα σε μία φόρμα. Αν ο admin πατήσει cancel τότε μεταφέρεται απευθείας πίσω στο admin.php, αλλιώς μετατρέπει όποια inputs της φόρμας επιθυμεί και αφού πατήσει update αυτά ανανεώνονται στις νέες του επιλογές. Για να το υλοποιήσω αυτό στέλνω την φόρμα σε ένα αρχείο update\_user.php το οποίο είναι το εξής:

```

<?php

require '../vendor/autoload.php';
$m = new MongoClient ("mongodb://127.0.0.1/");

$db = $m->signedUsers;

$collection = $db->users;

//if ( isset( $_POST['delete_btn'] ) ) {

    //$id = $_POST['delete_user'];

    //$collection->deleteMany(array("username"=>$id));
    //header('Location: ../html/admin.php');
    //echo $collection;
    //}

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

    $id=$_POST['edit_id']; //get selected user to update like i did in delete user

    $username = $_POST['edit_username'];
    $password = $_POST['edit_password'];
    $firstname = $_POST['edit_firstname'];
    $lastname = $_POST['edit_lastname'];
    $email = $_POST['edit_email'];

    $collection->updateOne(array( "username"=>$id ), array( '$set'=>array(
        "username" => "$username",
        "password" => "$password",
        "firstName" => "$firstname",
        "lastName" => "$lastname",
        "email" => "$email" ) ) );

    header('Location: ../html/admin.php');

}

}

```

Εδώ μετά τον έλεγχο του admin να πατήσει το κουμπί του update ορίζω και πάλι την μεταβλητή id για να κάνω τις αλλαγές μόνο για τον συγκεκριμένο χρήστη. Ορίζω τις 5 μεταβλητές με τις τιμές που πέρασε ο admin στο edit\_user\_changes.php και στις γραμμές 53-58 κάνω το update των δεδομένων.

Στη γραμμή 109 του admin.php του εμφανίζω την συνολική ώρα που είναι συνδεδεμένος στο σύστημα:

```

<?php $timeSinceLogin =( time () - $_SESSION ['timer']) /60; //time in minutes.
echo "Last user logged in for: " . $timeSinceLogin . " minutes";
?>

```

Last user logged in for: 2.35 minutes

