



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ

Τμήμα Πληροφορικής και Τηλεπικοινωνιών

ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ ΣΤΟ
ΠΕΡΙΒΑΛΛΟΝ LARAVEL - PHP



Ιακώβου Κωνσταντίνος

Επιβλέπων καθηγητής: Δημήτριος Λιαροκάπης

Άρτα, Ιούνιος, 2023

**«THE DEVELOPMENT OF A WEB APPLICATION
UTILIZING THE FRAMEWORK LARAVEL – PHP»**



Iakovou Konstantinos

Supervisor: Dimitrios Liarokapis

Arta, June, 2023

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

Άρτα, Ιούνιος 2023

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Επιβλέπων καθηγητής
Δημήτριος Λιαροκάπης,
2. Μέλος επιτροπής
Ελευθέριος Στεργίου
3. Μέλος επιτροπής
Μαργαρίτη Σπυριδούλα

© Ιακώβου, Κωνσταντίνος, 2023.

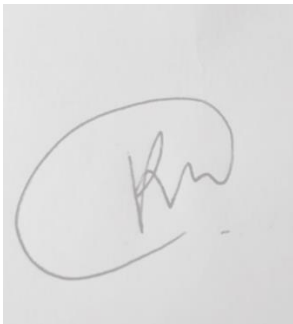
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα πτυχιακή εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Ιακώβου Κωνσταντίνος

Υπογραφή:

A square box containing a handwritten signature in dark ink. The signature is stylized and appears to be the Greek name 'Κωνσταντίνος Ιακώβου'.

ΕΥΧΑΡΙΣΤΙΕΣ

Ευχαριστώ πολύ τον πρώην συμφοιτητή μου Π.Φώτιο για την πολύτιμη βοήθεια του κατά τα ακαδημαϊκά μου έτη, τον κύριο Λιαροκάπη για τον χρόνο που αφιέρωσε επιβλέποντας της πτυχιακής μου και τέλος τον κύριο Βότη για τη μεταδοτικότητα του στο μάθημα «Τεχνικός Λόγος».

ΠΕΡΙΛΗΨΗ

Το Laravel είναι μία από τις πιο γνωστές εφαρμογές ανοικτού κώδικα πλαισίου(frameworks) ανάπτυξης ιστοσελίδων παγκόσμιου ιστού. Βασίζεται στην γλώσσα PHP τα αρχικά της οποίας σημαίνουν PHP: HyperText Preprocessor. Το Laravel χρησιμοποιεί και άλλες τεχνολογίες όπως HTML (Hypertext Markup Language), CSS (Cascade Style Sheet), JavaScript, Bootstrap, Vue, React κλπ. Ο δημιουργός του Laravel είναι ο Taylor Otwell. Πριν την εμφάνιση του Laravel υπήρχε ένα άλλο δημοφιλές περιβάλλον ανάπτυξης εφαρμογών σε γλώσσα PHP ήταν το CodeIgniter.

Στην παρούσα πτυχιακή εργασία παρουσιάζουμε τα οφέλη αλλά και τις δυσκολίες στην χρήση της συγκεκριμένης εφαρμογής προσπαθώντας να αναπτύξουμε και να παρουσιάσουμε μια διαδικτυακή εφαρμογή ηλεκτρονικού καταστήματος όπου εγγεγραμμένοι χρήστες θα μπορούν να κάνουν κράτηση μιας από τις δραστηριότητες που παρέχει η επιχείρηση όπως: ιππασία, καλλιτεχνικά και ξενάγηση στα χωριά της Κύμης Ευβοίας. Η εργασία περιλαμβάνει την περιγραφή της διαδικασίας εγκατάστασης και χρήσης των εργαλείων που χρησιμοποιήθηκαν και της λειτουργίας της εφαρμογής που αναπτύχθηκε.

Η Laravel έχει τα περισσότερα σε επέκταση έγγραφα και βίντεο εκμάθησης από άλλα μοντέρνα web εφαρμογής. Είναι back-end framework που προσφέρει όλα τα χαρακτηριστικά μιας εφαρμογής, συμπεριλαμβανομένων ισχυρών εργαλείων: JavaScript βιβλιοθηκών και frameworks όπως Vue και React, ώστε να ικανοποιεί τον προγραμματιστή ανάπτυξης εφαρμογών διαδικτύου.

ABSTRACT

Laravel is one of the most well-known open source web development frameworks. It is based on the PHP language whose initials stand for PHP: HyperText Preprocessor. Laravel also uses other technologies such as HTML (Hypertext Markup Language), CSS (Cascade Style Sheet), JavaScript, Bootstrap, Vue, React etc. The creator of Laravel is Taylor Otwell. Before the beginning of Laravel, another popular PHP application development environment was CodeIgniter.

In this thesis we present the benefits and also the difficulties in using this application by trying to develop and presents an online e-shop application where are registered users who will be able to book one of the activities provided by the company such as: horse riding, artistic and guided tours in the villages of Kymi Evia. The work includes the description of the installation and utilization of the tools and the function of the developed application.

Laravel has the most extensive and thorough documentation and video tutorial library of any modern web application framework. Is a backend framework that provides all the features of an application, including powerful tools like JavaScript libraries and frameworks such as Vue and React, which can pleasure each web developer who use it.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΕΥΧΑΡΙΣΤΙΕΣ	iv
ΠΕΡΙΛΗΨΗ	vi
ABSTRACT	vii
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ	ix
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ	xi
1.Εισαγωγή.....	1
1.1 Σχετικά με Laravel Framework.....	2
1.2 Πλεονεκτήματα της Laravel.....	2
1.2 Μειονεκτήματα της Laravel.....	3
1.3 Η γλώσσα προγραμματισμού PHP.....	3
1.4 Frameworks.....	4
2.Οι απαιτήσεις που χρειάζονται.....	4
2.1 Γλώσσες Παγκόσμιου Ιστού	4
2.2 Composer	5
2.3 Visual Studio Code	6
2.4 MAMP.....	7
3.Ανάγκες.....	8
3.1 Φιλοξενία server.....	9
3.2 Apache server.....	10
3.3 Phpmyadmin	10

3.4 Ανάπτυξη παγκόσμιου ιστού.....	12
4.Περιγραφή.....	13
4.1 Routes.....	14
4.2 Controllers.....	15
4.3 Database: Migrations.....	17
4.4 Middleware	18
5.Υλοποίηση.....	20
5.1 Δημιουργία Laravel project.....	20
5.2 Εγκατάσταση MAMP και Composer	20
5.3 Σχετικά με το NodeJS	26
5.4 Παραμετροποίηση αρχείου .env.....	26
6.Αποτέλεσμα.....	30
6.1 Εγγραφή χρήστη.....	30
6.2 Αγορά προϊόντος μέσω stripe.....	32
6.3 Επιβεβαίωση αγοράς.....	34
6.4 Ενημέρωση στοιχείων στον λογαριασμό.	35
BIBΛΙΟΓΡΑΦΙΑ	39

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 2.2 1	6
Εικόνα 2.3.1	6
Εικόνα 2.3.2	7
Εικόνα 2.4 1	8
Εικόνα 3.1 1	9
Εικόνα 3.3 1	11
Εικόνα 4.1.1	14
Εικόνα 4.1.2	15
Εικόνα 4.2.1	16
Εικόνα 4.3.1	17
Εικόνα 4.4.1	19
Εικόνα 5.2 1	21
Εικόνα 5.2.2	22
Εικόνα 5.2.3	22
Εικόνα 5.2.4	22
Εικόνα 5.2.5	23
Εικόνα 5.2.6	24
Εικόνα 5.2.7	25
Εικόνα 5.2.8	25
Εικόνα 5.3.1	26
Εικόνα 5.4.1	27
Εικόνα 5.4.2	28
Εικόνα 5.4.3	29
Εικόνα 6.1.1	31
Εικόνα 6.1 2	32
Εικόνα 6.2 1	32
Εικόνα 6.2.2	33
Εικόνα 6.2.3	33
Εικόνα 6.2.4	34
Εικόνα 6.3 1	35
Εικόνα 6.4 1	36
Εικόνα 6.4 2	36
Εικόνα 6.4 3	37

1.Εισαγωγή

Στο κεφάλαιο αυτό αναφέρονται οι εισαγωγικές έννοιες της Laravel Framework και γενικά των frameworks, οι γλώσσες παγκόσμιου ιστού που χρησιμοποιούνται, καθώς και τα πλεονεκτήματα, όπως για παράδειγμα η εκτενής βιβλιοθήκη προγραμματισμένων λειτουργιών και μειονεκτήματα της εφαρμογής, όπως το αυξημένο κόστος της πρόσληψης προγραμματιστή που να έχει γνώσεις frameworks και γλωσσών παγκόσμιου ιστού όπως της JavaScript.

1.1 Σχετικά με Laravel Framework

Δεν μπορεί να μεταφραστεί ακριβώς στα Ελληνικά η λέξη framework αλλά ο όρος σε γενικές γραμμές αναφέρεται σε μια εφαρμογή ανάπτυξης διαδικτύου που μπορεί να την χειριστεί ο κάθε προγραμματιστής ιστού διαδικτύου. Το πιο γνωστό και φιλικό προς τον προγραμματιστή πρόγραμμα που χειρίζεται γλώσσες προγραμματισμού, HTML, CSS και scripts όπως JavaScript, είναι το Visual Studio Code, το οποίο μπορεί να εκτελέσει μέσω τερματικού εντολές σε linux, γεγονός που το κάνει πολύ εύχρηστο. Με το πρόγραμμα της Microsoft λοιπόν, μαζί με το composer που είναι ένας διαχειριστής εξαρτήσεων(dependency manager) σε επίπεδο εφαρμογής για τη γλώσσα προγραμματισμού PHP, παρέχει μια τυπική μορφή για τη διαχείριση των εξαρτήσεων του λογισμικού PHP και των απαιτούμενων βιβλιοθηκών. Καθώς το MAMP, το οποίο είναι ανοιχτού κώδικα λογισμικό που μπορεί να αναπτύξει και να προβάλει δυναμικές ιστοσελίδες και εφαρμογές και παράλληλα δίνει τη δυνατότητα στον προγραμματιστή να αναπτύξει οποιαδήποτε εφαρμογή επιθυμεί για να εξυπηρετήσει διάφορους σκοπούς, όπως για παράδειγμα τις ανάγκες ενός ξενοδοχείου ή νοσοκομείου. Τέλος, εκτός των άλλων περιλαμβάνει ένα ολοκληρωμένο σύστημα αυθεντικοποίησης και αυτόματων δοκιμών. Συμπερασματικά, η Laravel Framework είναι μια πολύτιμη και δυναμική εφαρμογή που μπορεί να υποστηρίξει ακόμα Vue and React, όπου η Vue είναι framework και η React βιβλιοθήκη, γεγονός που σημαίνει πως γίνεται ακόμα πιο συναρπαστική.

1.2 Πλεονεκτήματα της Laravel

- Απλή στην σύνταξη κώδικα: μπορεί να υποστηρίξει σύνθετες εργασίες, καθώς διαθέτει μια εκτενή βιβλιοθήκη προγραμματισμένων λειτουργιών. Η υποστήριξη αυτή κάνει την εφαρμογή πιο δυναμική.
- Επεκτασιμότητα: τα μεγέθη των έργων διαφέρουν. Επιτρέπει να αντιμετωπίζονται έργα οποιουδήποτε μεγέθους, ανάλογα με τις εκάστοτε ανάγκες του έργου.
- Ασφάλεια: από την 1^η έκδοση και μετά βελτιώθηκε το ενσωματωμένο σύστημα ελέγχου πρόσβασης, η υποστήριξη για έλεγχο ταυτότητας και η εξουσιοδότηση χρήστη. Αξίζει να αναφερθεί πως παρέχει έναν ισχυρό μηχανισμό που επιτρέπει στον προγραμματιστή να χειρίζεται εύκολα τυχόν σφάλματα ή προβλήματα.

- Εύκολη στη διαχείριση και εγκατάσταση βάσεων δεδομένων.
- Ευκολία: ένα από τα κύρια χαρακτηριστικά της Laravel Framework είναι πως παρέχει άφθονο υλικό εκπαίδευσης και εκμάθησης της εφαρμογής. Γεγονός που σημαίνει πως μπορεί κάποιος να μάθει να τη χειρίζεται χωρίς ιδιαίτερη δυσκολία.

1.2 Μειονεκτήματα της Laravel

- Ελαφρότητα: θεωρείται ένα ελαφρύ framework για το λόγο ότι διαθέτει μια ολοκληρωμένη λίστα με ενσωματωμένες δυνατότητες που έχουν σχεδιαστεί για να κάνουν την ανάπτυξη ιστού ευκολότερη, αλλά μπορεί ταυτόχρονα να κουράσει γρήγορα τον χρήστη κατά την υλοποίηση μια ισχυρής εφαρμογής.
- Κόστος: για να μπορεί να υλοποιηθεί μια εφαρμογή απαιτείται ο προγραμματιστής να κατέχει εμπειρία και γνώσεις πάνω στην Laravel γεγονός που κοστίζει πολύ.
- Υποστήριξη: το γεγονός ότι η υποστήριξη για το λογισμικό δεν είναι άμεσα διαθέσιμο μπορεί να προκαλέσει προβλήματα στην περίπτωση διακοπής λειτουργίας του συστήματος.
- Ενημερώσεις: ένα από τα βασικότερα μειονεκτήματα του framework είναι πως οι παλαιότερες εκδόσεις μπορούν να παρουσιάσουν διάφορα σφάλματα και δυσκολίες σε μια εφαρμογή.
- Καθυστέρηση: η ανάπτυξη ιστού είναι αρκετά αργή σε σχέση με άλλα frameworks, διότι όπως προαναφέρθηκε, η Laravel έχει πολλά ενσωματωμένα εργαλεία.

1.3 Η γλώσσα προγραμματισμού PHP

Όπως αναφέρθηκε και στην εισαγωγή η Laravel Framework βασίζεται στην PHP καθώς είναι η πιο διαδεδομένη γλώσσα προγραμματισμού, στην οποία γράφονται καθημερινά χιλιάδες σενάρια. Αντί λοιπόν για πολλές εντολές εξόδου HTML (όπως στη C ή την Perl), οι σελίδες PHP περιέχουν HTML με ενσωματωμένο κώδικα. Ο κώδικας PHP περικλείεται σε ειδικές οδηγίες έναρξης και λήξης επεξεργασίας “<?php και ?>” που επιτρέπουν τη μετάβαση και έξοδο από το σενάριο κώδικα.

Ένα από τα μεγαλύτερα πλεονεκτήματα, είναι της PHP ότι είναι εξαιρετικά εύληπτη από νέο προγραμματιστή, αλλά συγχρόνως προσφέρει και πολλές προηγμένες δυνατότητες

σε έναν έμπειρο προγραμματιστή. Υπάρχει μεγάλη λίστα με τις δυνατότητες της PHP, ώστε σε σύντομο χρονικό διάστημα να μπορούν να συνταχτούν απλά scripts.

Η PHP εκτελείται στο διακομιστή δημιουργώντας HTML που στη συνέχεια αποστέλλεται στον πελάτη σε σχέση με την JavaScript.

1.4 Frameworks

Όπως τα διάσημα frameworks applications: Vue, Angular, jQuery, έτσι και η Laravel στον προγραμματισμό υπολογιστών είναι ένα σύνολο αφηρημένων εννοιών, στην οποία το λογισμικό μπορεί να αλλάξει επιλεκτικά με πρόσθετο κώδικα γραμμένο από τον προγραμματιστή παρέχοντας το έτσι για συγκεκριμένη κάθε φορά εφαρμογή. Παρέχει έναν τυπικό τρόπο δημιουργίας και ανάπτυξης εφαρμογών και είναι ένα καθολικό, επαναχρησιμοποιήσιμο περιβάλλον λογισμικού, που αποτελεί μια υποκατηγορία μιας μεγαλύτερης πλατφόρμας, με ιδιαίτερη λειτουργικότητα για τη διευκόλυνση της ανάπτυξης εφαρμογών λογισμικού, προϊόντων και λύσεων. Μπορεί να περιλαμβάνουν προγράμματα υποστήριξης, μεταγλωττιστές, βιβλιοθήκες κώδικα, σύνολα εργαλείων και διεπαφές προγραμματισμού εφαρμογών (API) που συγκεντρώνουν όλα τα διαφορετικά στοιχεία για να επιτρέψουν την ανάπτυξη ενός έργου ή συστήματος.

2.Οι απαιτήσεις που χρειάζονται

Όπως προαναφέρθηκε στα μειονεκτήματα, για να μπορέσει να χειριστεί κάποιος προγραμματιστής μια δυναμική ή απλή εφαρμογή απαιτούνται πολλές γνώσεις, όπως HTML, CSS, JavaScript και άλλα frameworks όπως Vue, Angular, jQuery, στη συγκεκριμένη περίπτωση της μικρής επιχείρησης υπάρχουν αρκετές απαιτήσεις της εφαρμογής, που αφορούν τι μπορεί να κάνει ο χρήστης με την εφαρμογή. Σε αυτό το κεφάλαιο θα παρουσιαστούν - αναλυθούν οι έννοιες αυτές καθώς για το Composer και MAMP.

2.1 Γλώσσες Παγκόσμιου Ιστού

- HTML(HyperText Markup Language): τυπική γλώσσα σήμανσης για έγγραφα που έχουν σχεδιαστεί για εμφάνιση σε πρόγραμμα περιήγησης ιστού.

Υποστηρίζεται από τεχνολογίες όπως Cascading Style Sheets (CSS) και γλώσσες δέσμης ενεργειών όπως η JavaScript.

- Cascading Style Sheets: περιγράφει πως πρέπει να εμφανίζονται τα στοιχεία HTML και χρησιμοποιείται για την μορφοποίηση ενός εγγράφου HTML.
- JavaScript: ως συντόμευση JS που είναι μια από τις βασικές τεχνολογίες του Παγκόσμιου Ιστού, μαζί με την HTML και το CSS. Περιλαμβάνει στοιχεία της γλώσσας προγραμματισμού JAVA και είναι η πιο διάσημη γλώσσα προγραμματισμού παγκοσμίως.
- PHP: Επίσης διάσημη γλώσσα προγραμματισμού γενικής χρήσης και προσανατολισμένη στην ανάπτυξη ιστού.

2.2 Composer

Το Composer είναι ένα εργαλείο διαχείρισης που υλοποιήθηκε με PHP και επιτρέπει τον καθορισμό μιας σειράς εντολών στο project που εξαρτάται από αυτό και διαχειρίζεται την αναβάθμιση και εγκατάσταση. Μπορεί εύκολα να γίνει λήψη και να ενσωματωθούν χρήσιμα πακέτα μέσω του τερματικού(Terminal). Η λήψη μπορεί να γίνει είτε από getcomposer.org, είτε με εντολές σε λογισμικά, όπως linux/Unix/MacOS/WindowsOS. Στην τελευταία του έκδοση απαιτεί PHP 7.2.5 για να εκτελεστεί. Απαιτούνται επίσης κάποιες ευαίσθητες ρυθμίσεις, όπως για παράδειγμα στο αρχείο `.env`, στο οποίο περιέχονται κάποιες κοινές ρυθμίσεις που διαφέρουν ανάλογα με το αν το project εκτελείται τοπικά ή σε διακομιστή. Οι μεταβλητές αυτές ανακτώνται από διάφορα αρχεία διαμόρφωσης Laravel στον κατάλογο αρχείων. Αυτό φαίνεται στην εικόνα 2.2.1 όπου `APP_URL` αν εκτελείται σε διακομιστή της εφαρμογής όπου εισάγεται ο σύνδεσμος τις ιστοσελίδας, αλλιώς τοπικά `http://localhost`. Επίσης πολύ σημαντικό ρόλο κατέχουν οι παράμετροι που πρέπει να συνδεθούν στη βάση δεδομένων μέσω `myrhadmin`.

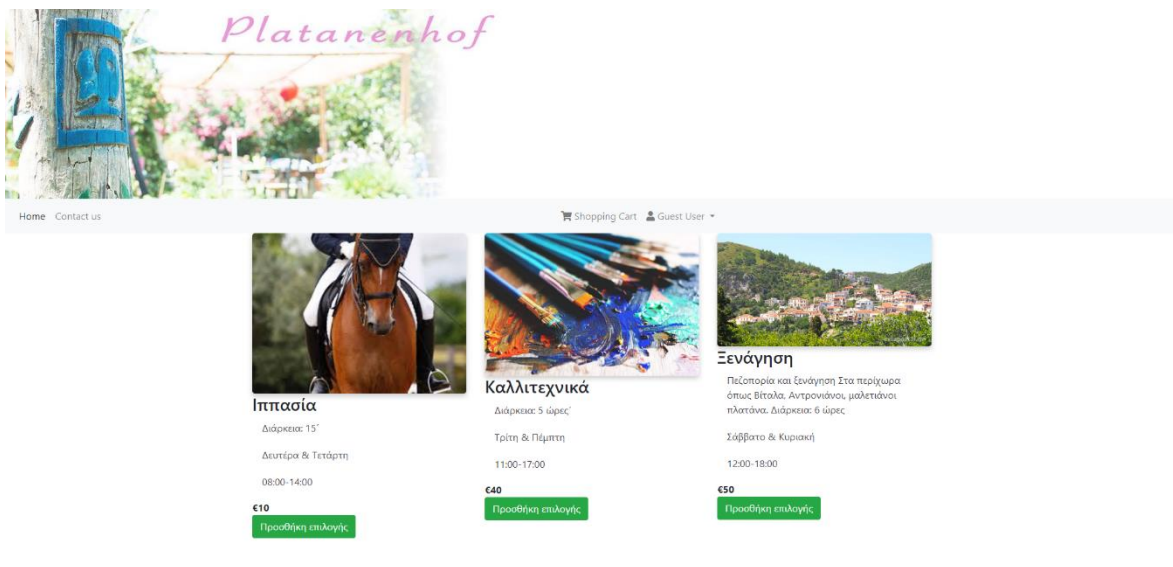
```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:2WLuKXcwDiXTI3II1X30Pwq7XqIQwEebKlZ70ydGqfU=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8
9 DB_CONNECTION=mysql
10 DB_HOST=localhost
11 DB_PORT=3306
12 DB_DATABASE=laravel
13 DB_USERNAME=root
14 DB_PASSWORD=root
15
```

Εικόνα 2.2.1

2.3 Οι απαιτήσεις της εφαρμογής ως προς την χρήση

Η έννοια υλοποίησης της εφαρμογής βασίζεται σε ένα απλό και λειτουργικό ηλεκτρονικό κατάστημα, που αφορά μια μικρή επιχείρηση η οποία μπορεί να παρέχει στους πελάτες 3 κατηγορίες δραστηριοτήτων. Οι κατηγορίες αυτές συγκεκριμένα είναι η ιππασία, τα καλλιτεχνικά και τέλος η ξενάγηση στα περίχωρα της περιοχής της Κύμης στην Εύβοια.

Οι λειτουργίες που υπάρχουν σε αυτή την εφαρμογή απαιτούν σαν προϋπόθεση ο χρήστης να κάνει εγγραφή, μέσω του μενού sign up που φαίνεται στην εικόνα 2.3.1.



Εικόνα 2.3.1

Εφόσον θα έχει ήδη εγγραφεί, θα παρουσιαστεί μια επιπλέον επιλογή που θα αφορά την εμφάνιση προσωπικών στοιχείων του πελάτη και τη δυνατότητα επεξεργασίας των δεδομένων αυτών όπως φαίνεται στην εικόνα 2.3.2

Home Contact us Shopping Cart Guest User

Sign Up

Name:

Surname:

Password: Show Password
Your password must be more than 8 characters long, should contain at-least 1 Uppercase, 1 Lowercase, 1 Numeric and 1 special character.

Password Confirmation: Show Password

E-Mail Address:

Confirm Email:

Phone Number:

Σταθερό τηλέφωνο(προαιρετικό):

Present Address:

City:

Zip:

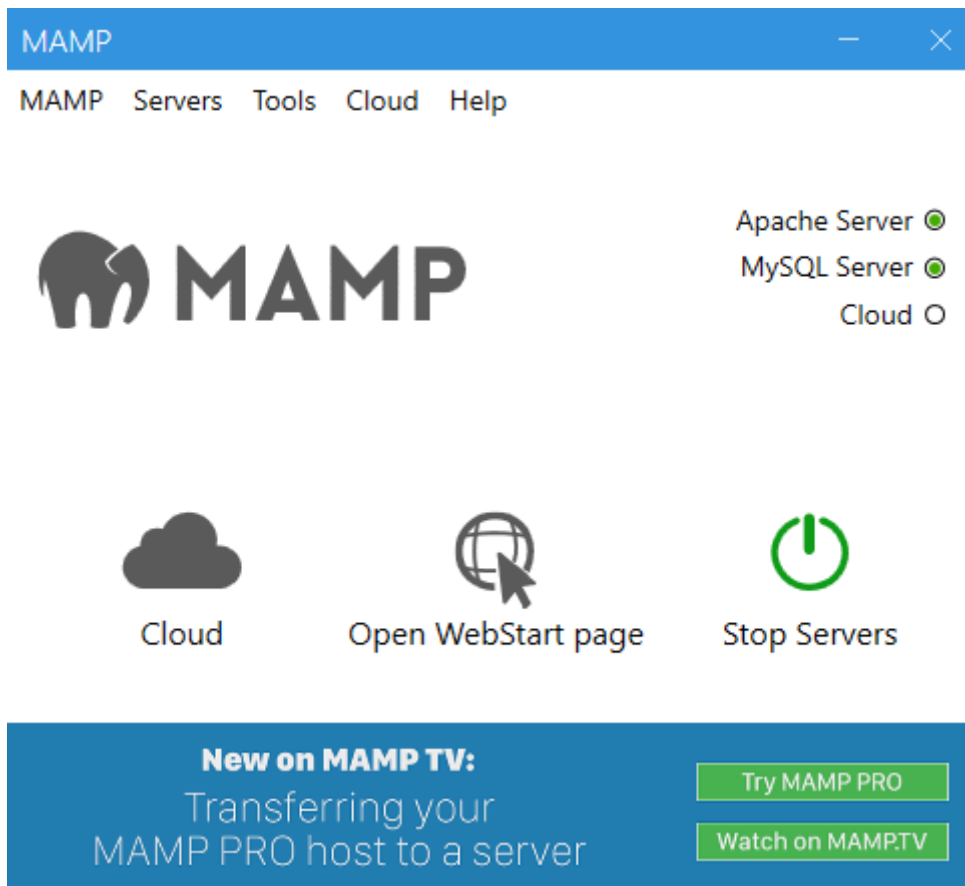
Vatnumber:

Εικόνα 2.3.2

Πρέπει να αναφερθεί πως, αν δε έχει κάνει εγγραφή ο χρήστης, δε θα μπορεί να έχει πρόσβαση σε αυτή τη λειτουργία. Τέλος ο χρήστης, όπως θα αναφερθεί και στα επόμενα κεφάλαια, θα μπορεί να επιλέξει μια από τις τρεις κατηγορίες οι οποίες αναφέρουν το εβδομαδιαίο πρόγραμμα που θα μπορεί να παρακολουθήσει ο χρήστης, το οποίο φαίνεται στην προηγούμενη εικόνα 2.3.1.

2.4 MAMP

Πρόγραμμα το οποίο μπορεί να προβάλει σε τοπικό υπολογιστή μια δυναμική ή απλή ιστοσελίδα ή διαδικτυακή εφαρμογή. Είναι μια στοίβα λύσεων που αποτελείται από ελεύθερο, ανοιχτού κώδικα και αποκλειστικό εμπορικό λογισμικό που χρησιμοποιούνται μαζί για την ανάπτυξη και τη εκτέλεση δυναμικών ιστοσελίδων. Μπορεί να εγκατασταθεί σε MacOS ή WindowsOS δωρεάν. Το πρόγραμμα αυτό μπορεί να το χρησιμοποιήσει ο προγραμματιστής για να υλοποιήσει και να επεξεργαστεί το έργο του. Συνδέεται με την βάση δεδομένων μέσω myhradmin, καθώς σε cloud υπηρεσίες και το γραφικό περιβάλλον του και η χρήση του το κάνει πιο ευχάριστο στον χρήστη και αρεστό σε σχέση με άλλα παρόμοια προγράμματα, όπως το XAMP. Στην εικόνα 2.4.1 φαίνεται πως εκτελεί τον apache και Mysql Server.



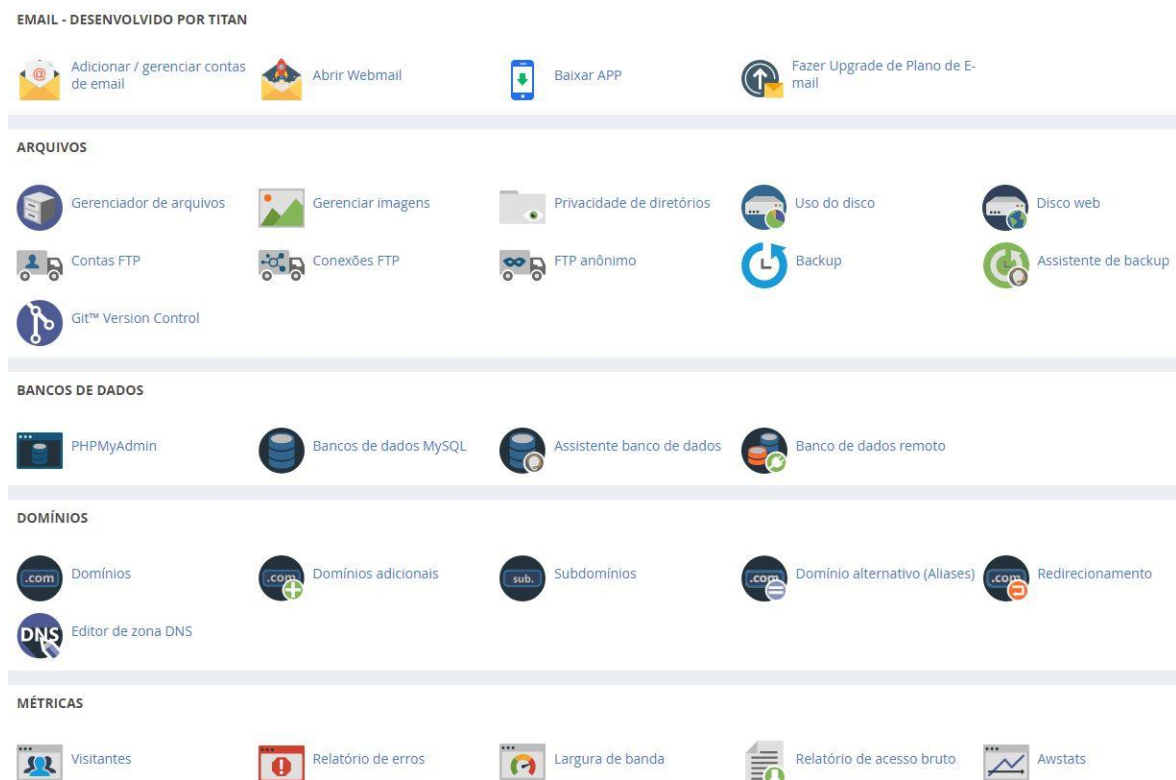
Εικόνα 2.4 1

3.Ανάγκες

Σε αυτό το κεφάλαιο αναφέρονται οι προϋποθέσεις που απαιτούνται, ώστε μέσω του framework να μπορεί να εξελιχτεί το έργο. Μία από τις προϋποθέσεις αυτές είναι η φιλοξενία σε διακομιστή, συγκεκριμένα στο HostGator. Κάθε διακομιστής μπορεί να έχει διαφορετικές ανάγκες, όπως η σύνδεση με τον server μέσω SSH ή Secure Shell. Η ύπαρξη του apache server, που χρησιμοποιείται στην ανάπτυξη εφαρμογών παγκόσμιου ιστού, καθώς και η διαχείριση της πλατφόρμας «phpMyAdmin» από τον χρήστη μέσα σε αυτή την εφαρμογή, αποτελεί ένα σημαντικό στοιχείο γιατί λειτουργεί ως περιβάλλον για τη διαχείριση βάσης δεδομένων και συνδέεται μέσω του τερματικού ή αλλιώς γραμμή εντολών που παρέχει ο Visual Studio Code ή μέσω του λειτουργικού συστήματος των windows .

3.1 Φιλοξενία server

Σήμερα έχει αυξηθεί ο αριθμός φιλοξενίας ιστοσελίδων και παγκόσμιου ιστού εφαρμογών. Πρέπει να σημειωθεί πως το project δεν έχει εγκατασταθεί στην ιστοσελίδα που φιλοξενεί την εφαρμογή, πάρα μόνο εμφανίζεται τοπικά. Βάσει μιας άλλης Laravel εφαρμογής που υλοποιήθηκε για να εξυπηρετήσει μια καθηγήτρια αγγλικών στην Βραζιλία, βασίζεται η έννοια φιλοξενίας στο HostGator. Η έννοια αυτή είναι πως για να εγκαταστήσει και όχι για να ανεβάσει, όπως πολλοί ίσως συγχέουν, αφού δε γίνεται uploading, αλλά εγκατάσταση περιβάλλοντος με τα αρχεία της εφαρμογής, εφόσον ανεβούν στον server. Για να συμβεί αυτό σε κάθε φιλοξενία, υπάρχει ένα Cpanel με διάφορα εργαλεία που προσφέρει η συγκεκριμένη ιστοσελίδα-φιλοξενία server, Όπως φαίνεται στην εικόνα 3.1.1.



Εικόνα 3.1 1

Όταν κάποιος πελάτης θέλει να του υλοποιήσει ο προγραμματιστής μια εφαρμογή, θα πρέπει να δοθεί ένας αποκλειστικός κωδικός χρήστη και κωδικό πρόσβασης εφόσον αγοράσει το domain, δηλαδή την φιλοξενία. Όμως, εκτός από τη χρήση κωδικών, υπάρχει και δεύτερος τρόπος πρόσβασης στον server. Αυτό γίνεται με το SSH η Secure Shell μέσω του τερματικού των windows. Πρόκειται για ένα πρωτόκολλο δικτύου επικοινωνίας που επιτρέπει την επικοινωνία και

μεταφορά δεδομένων μεταξύ υπολογιστών. Όλα τα αρχεία θα πρέπει να ανεβούν μέσω του file manager(διαχείριση αρχείων) στον φάκελο public_html για να εμφανιστεί η εφαρμογή .

3.2 Apache server

Ο διακομιστής Apache HTTP, είναι ένα δωρεάν και ανοιχτού κώδικα λογισμικό διακομιστών παγκόσμιου ιστού (web) για πολλαπλές πλατφόρμες, που κυκλοφορεί σύμφωνα με τους όρους της Άδειας χρήσης Apache 2.0. Το Apache αναπτύσσεται και διατηρείται από μια ανοιχτή κοινότητα προγραμματιστών υπό την αιγίδα του Apache Software Foundation. Ο apache server και η mysql server είναι οι δύο βασικές λειτουργίες που απαιτούνται για το MAMP πρόγραμμα που αναφέρθηκε στο αμέσως προηγούμενο κεφάλαιο. Το Apache HTTP Server Project είναι μια συλλογική προσπάθεια ανάπτυξης λογισμικού που στοχεύει στη δημιουργία μιας ισχυρής, ποιοτικής εμπορικά, πλούσιας σε χαρακτηριστικά και ελεύθερα διαθέσιμης υλοποίησης πηγαίου κώδικα ενός διακομιστή HTTP (Web). Το έργο το διαχειρίζεται από κοινού μια ομάδα εθελοντών που βρίσκονται σε όλο τον κόσμο, ανήκουν στο Apache Software Foundation και χρησιμοποιούν το Διαδίκτυο και τον Ιστό για να επικοινωνήσουν, να σχεδιάσουν, να αναπτύξουν και να τελειοποιήσουν τον διακομιστή. Σε αυτό έχουν συνεισφέρει και εκατοντάδες άλλοι χρήστες. Ο Apache, όπως και άλλο λογισμικό διακομιστή, μπορεί να παραβιαστεί και να γίνει αντικείμενο εκμετάλλευσης. Με λίγα λόγια ο apache server είναι υπεύθυνος για την φόρτωση μόνο σε τοπικό επίπεδο της ιστοσελίδας ή της εφαρμογής. Αλλά θα πρέπει να είναι ενεργός και ο mysql server όταν πρόκειται για μια εφαρμογή. Υπάρχουν αρκετά προγράμματα όπως το MAMP και XAMP που υποστηρίζουν αυτή τη λειτουργία.

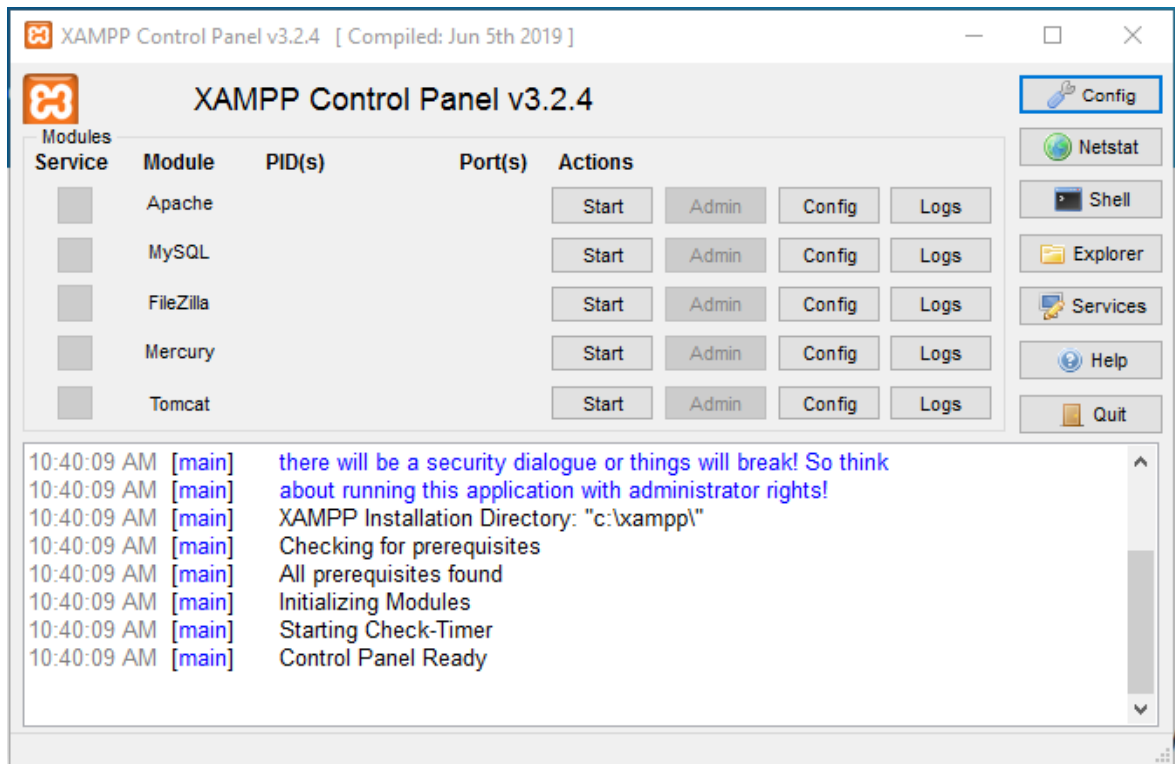
3.3 Phpmyadmin

Το phpMyAdmin είναι ένα δωρεάν εργαλείο λογισμικού γραμμένο σε PHP, που προορίζεται να διαχειριστεί τη MySQL μέσω του Web. Το phpMyAdmin υποστηρίζει ένα ευρύ φάσμα λειτουργιών σε MySQL και MariaDB. Οι λειτουργίες που χρησιμοποιούνται συχνά (διαχείριση βάσεων δεδομένων, πίνακες, στήλες, σχέσεις,

ευρετήρια, χρήστες, δικαιώματα, κ.λπ.) μπορούν να εκτελεστούν μέσω της διεπαφής χρηστών, οι οποίοι εξακολουθούν να έχουν τη δυνατότητα να εκτελέσουν απευθείας οποιαδήποτε πρόταση SQL. Για να διευκολύνεται η χρήση σε ένα ευρύ φάσμα ατόμων, το phpMyAdmin μεταφράζεται σε 72 γλώσσες και υποστηρίζει γλώσσες LTR και RTL.

Το έργο phpMyAdmin είναι μέλος του Software Freedom Conservancy. Το SFC είναι ένας μη κερδοσκοπικός οργανισμός που βοηθά στην προώθηση, βελτίωση, ανάπτυξη και υπεράσπιση έργων Λογισμικού Ανοικτού Κώδικα (FLOSS).

Για να εγκατασταθεί και να συνδεθεί η διαχείριση βάσεων δεδομένων phpMyAdmin, χρειάζεται όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, η δήλωση ονόματος χρήστη, το όνομα της βάσης δεδομένων και ο κωδικός πρόσβασης. Συγκεκριμένα για την Laravel αυτό γίνεται στο αρχείο .env που είναι πολύ σημαντικό αρχείο. Σε άλλη περίπτωση η δήλωση στοιχείων γίνεται στο αρχείο inc.php. Το αρχείο αυτό μπορεί να βρεθεί άμεσα μέσω του προγράμματος MAMP. Μπορεί ακόμα ο χρήστης να κατεβάσει σε πρόγραμμα το phpMyAdmin από τον σύνδεσμο: http://www.phpmyadmin.net/home_page/downloads.php πατώντας το κουμπί config όπως φαίνεται στην εικόνα 3.3.1.



Εικόνα 3.3.1

Μέσω του κώδικα, συγκεκριμένα και περίπου στο τμήμα κώδικα:

```
$cfg['Servers'][$i]['verbose'] = 'localhost';
```

```
$cfg['Servers'][$i]['host'] = 'localhost';
```

```
$cfg['Servers'][$i]['port'] = '';
```

```
$cfg['Servers'][$i]['socket'] = '';
```

```
$cfg['Servers'][$i]['connect_type'] = 'tcp';
```

```
$cfg['Servers'][$i]['extension'] = 'MySQLi';
```

```
$cfg['Servers'][$i]['auth_type'] = 'config';
```

```
$cfg['Servers'][$i]['user'] = 'root';
```

```
$cfg['Servers'][$i]['password'] = κωδικός; //αντικατάσταση “κωδικός” με την επιλογή του χρήστη.
```

3.4 Ανάπτυξη παγκόσμιου ιστού

Στην αγγλική ορολογία αναφέρεται ως web development και στα ελληνικά μεταφράζεται ως ανάπτυξη παγκόσμιου ιστού. Με πιο απλά λόγια είναι η ανάπτυξη κώδικα ή script όπως η html(Hypertext Markup Language), CSS(Cascade Style Sheets), JavaScript, PHP. Επίσης τα scripts μπορούν να περιλαμβάνουν γλώσσες προγραμματισμού όπως η C, C++, JAVA. Με όλες αυτές τις γλώσσες προγραμματισμού και τα scripts μπορούν να υλοποιηθούν εξαιρετικές και με πιο πολλά γραφικά, καθώς και δυναμικές ιστοσελίδες ή εφαρμογές. Το μόνο που χρειάζεται είναι ένας προγραμματιστής και ένα πολύ καλό πρόγραμμα επεξεργασίας κώδικα όπως το διάσημο πρόγραμμα Visual Studio Code που αναφέρθηκε στον προηγούμενο κεφάλαιο. Επιπροσθέτως, η ανάπτυξη Ιστού είναι η εργασία που απαιτείται για την ανάπτυξη ενός ιστότοπου για το Διαδίκτυο (World Wide Web) ή ενός intranet (ενός ιδιωτικού δικτύου). Η ανάπτυξη Ιστού μπορεί να κυμαίνεται από την ανάπτυξη μιας απλής στατικής σελίδας απλού κειμένου έως πολύπλοκες εφαρμογές Ιστού, ηλεκτρονικές επιχειρήσεις και υπηρεσίες κοινωνικών δικτύων. Είναι μια εξαιρετική τεχνολογία, η οποία μπορεί να χρησιμοποιηθεί με πολλούς τρόπους για διάφορους σκοπούς, όπως για παράδειγμα στις επιχειρήσεις. Υπάρχει μια βασική κατηγοριοποίηση ενός web developer, η οποία είναι front και back end. Το Front-end

development αφορά στην εμφάνιση, δηλαδή πως φαίνεται η ιστοσελίδα στον πελάτη. Το Back-end development αναφέρεται στην πλευρά του server και δείχνει πως λειτουργεί μια ιστοσελίδα, πράγμα το οποίο αναφέρει το προγραμματιστικό μέρος.

Χρησιμοποιείται για τη δημιουργία στατικών ιστότοπων, όπου σκοπός είναι η εμφάνιση της ιστοσελίδας. Ωστόσο, εάν κάποιος θέλει να κάνει τον ιστότοπο δυναμικό, όπως μέσα από την διαχείριση αρχείων και βάσεων δεδομένων, από την προσθήκη φορμών επικοινωνίας, από τον έλεγχο πρόσβασης χρήστη κ.λπ, θα πρέπει να μάθει μια γλώσσα προγραμματισμού back-end, όπως PHP ή Python, και να χρησιμοποιήσει την SQL για επικοινωνία με βάσεις δεδομένων. Ένα από τα μειονεκτήματα που αναφέρθηκαν στο πρώτο κεφάλαιο είναι το υψηλό κόστος που χρειάζεται για την πρόσληψη έναν προγραμματιστή ο οποίος θα υλοποιήσει ένα έργο.

4.Περιγραφή

Στο κεφάλαιο αυτό περιγράφεται ο τρόπος που υλοποιείται το έργο. Αναφέρονται οι βασικές λειτουργίες της ιστοσελίδας όπως είναι τα routes, blades, NodeJS. Controllers, Auth. Η θεματολογία του είναι μια εφαρμογή, όπου θα μπορεί ο ενδιαφερόμενος να επιλέξει μια από τις δραστηριότητες που προσφέρει ένα κτήμα με μια παραδοσιακή γερμανική καφετέρια, στην Κύμη της Ευβοίας, οι οποίες είναι ιπασσία, καλλιτεχνικά και ξενάγηση στα γύρω τοπία. Η κάθε επιλογή αναφέρει ημέρα και ώρα συνάντησης για την εκδήλωση. Αφού κάνει την επιλογή ο χρήστης, τότε θα αποστέλλεται στο email του η κράτηση που έκανε με τα στοιχεία της, το είδος της δραστηριότητας και την τιμή του προϊόντος.

4.1 Routes

Τα routes, ή αλλιώς διαδρομές βρίσκονται στο αρχείο web.php του καταλόγου στο project, και φαίνονται στην εικόνα 4.1.1.

```
use App\Mail\WelcomeEmail;
use Illuminate\Support\Facades\Mail;
use App\Http\Controllers\PostController;
use App\Http\Controllers\ProductController;
use Illuminate\Support\Facades\Route;
use Illuminate\Database\Seeder;

Route::get('shop.index', 'ProductController@getIndex');
Route::get('shop.index/getindex/{id}', 'ProductController@getIndex');

Route::get('post/create', [PostController::class, 'create']);

Route::post('post', [PostController::class, 'store']);
Route::get('/', [
    'uses' => 'ProductController@getIndex',
    'as' => 'product.index'
]);
Route::get('/add-to-cart/{id}', [
    'uses' => 'ProductController@getAddToCart',
    'as' => 'product.addToCart'
]);
Route::get('/reduce/{id}', [
    'uses' => 'ProductController@getReduceByOne',
    'as' => 'product.reduceByOne',
]);
Route::get('/remove/{id}', [
    'uses' => 'ProductController@getRemoveItem',
    'as' => 'product.remove'
]);
Route::get('/shopping-cart', [
    'uses' => 'ProductController@getCart',
    'as' => 'product.shoppingCart'
]);
Route::get('/checkout', [
    'uses' => 'ProductController@getCheckout',
    'as' => 'checkout1',
    'middleware' => 'auth'
]);
Route::post('/checkout', [
    'uses' => 'ProductController@postCheckout',
    'as' => 'checkout',
    'middleware' => 'auth'
]);
Route::get('/password/reset/{token}/{email}', 'Auth\ResetPasswordController@showResetForm')->name('password.reset0');
Route::get('/contact', 'ContactMessageController@create')->name('contact');
Route::post('/contact', 'ContactMessageController@store')->name('contact.store');
```

Εικόνα 4.1.1

Υπάρχουν βιβλιοθήκες, όπως σε κάθε γλώσσα προγραμματισμού, απαραίτητες για να εκτελεστεί σωστά το project. Οι διαδρομές αυτές λαμβάνουν πληροφορίες από μια blade.php σελίδα, όπου γίνεται εμφάνιση δεδομένων προκειμένου να λειτουργήσει σωστά η εφαρμογή. Το Blade παρέχει επίσης βολικές συντομεύσεις για κοινές δομές ελέγχου PHP.

Κάθε διαδρομή ορίζει από ποιά σελίδα θα λάβει πληροφορίες από τον controller και ποιόν controller θα χρησιμοποιήσει. Χρησιμοποιούνται 2 τρόποι σύνταξης της διαδρομής διότι το project βασίστηκε στην έκδοση της Laravel 5 όπως φαίνεται στην γραμμή 22 και 32 της εικόνας 4.1.2

```
22 Route::get('shop.index','ProductController@getIndex');
23 Route::get('shop.index/getindex/{id}','ProductController@getIndex');
24
25 Route::get('post/create', [PostController::class, 'create']);
26
27 Route::post('post', [PostController::class, 'store']);
28 Route::get('/', [
29     'uses' => 'ProductController@getIndex',
30     'as' => 'product.index'
31 ]);
32 Route::get('/add-to-cart/{id}', [
33     'uses' => 'ProductController@getAddToCart',
34     'as' => 'product.addToCart'
35 ]);
36 Route::get('/reduce/{id}', [
37     'uses' => 'ProductController@getReduceByOne',
38     'as' => 'product.reduceByOne',
39 ]);
```

Εικόνα 4.1.2

Η διαδρομή στην γραμμή 25 δείχνει πως λαμβάνει πληροφορίες από τη συνάρτηση create στον controller ProductController. Τέτοια διαδρομή χρησιμοποιούμε για παράδειγμα, όταν θέλουμε να στείλουμε τα στοιχεία που έδωσε ο χρήστης στην σελίδα sign up όπου κάνει εγγραφή, για να μπορέσει να κάνει την κράτηση.

4.2 Controllers

Ο ρόλος των controllers είναι να διαχειρίζονται με ταξινόμηση αιτήματα από τα αρχεία των διαδρομών. Πιο απλά μπορεί να αναφερθεί η χρήση ενός UserController όπου σε μια κλάση φαίνεται πως εισάγει από την πλατφόρμα εγγραφής των χρηστών τα στοιχεία στη βάση δεδομένων όπως φαίνεται στην εικόνα 4.2.1

```

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\User;
6
7 use Illuminate\Http\Request;
8 use App\Http\Requests;
9 use Auth;
10 use Session;
11 use Illuminate\Support\Facades\Mail;
12 use Illuminate\Support\Facades>Password;
13 use Validator;
14 use DB;
15 class UserController extends Controller
16 {
17     public function getSignup() {
18         return view('user.signup');
19     }
20
21     public function postSignup(Request $request) {
22         $this->validate($request, [
23             'email' => 'required|string|email|max:255|unique:users|confirmed|exists:users',
24             'password' => 'required|string|min:6|confirmed|regex:/^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[#?!@$%^&*~]).{6,}$/',
25             'password_confirmation' => 'required|min:6',
26             'phone' => 'required|regex:/[6-9]{10}/',
27             'stathero' => 'numeric|regex:/[2-9]{10}/',
28             'name' => 'required|max:12',
29             'surname' => 'required|max:13',
30             'city' => 'required|max:12',
31             'address' => 'required|max:25',
32             'zip' => 'required|numeric|min:5',
33             'vatnumber' => 'nullable|regex:/(\d{01})[0-8]{8}/'
34         ],
35         [
36             'email.required' => 'An email required',
37             'email.unique' => 'This email is not unique',
38             'password.required' => 'A password required',
39             'password.regex' => 'Password must include: /^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[#?!@$%^&*~]).{6,}$/',
40             'phone.required' => 'A phone required',
41             'phone.required' => 'Phone number must be: [6-9]{10}',
42             'stathero.required' => 'stathero must be: [2-9]{10}',
43             'name.required' => 'A name required',
44             'surname.required' => 'A surname required',
45             'city.required' => 'A city required',
46             'address.required' => 'An address required',
47             'zip.required' => 'A zip required',
48         ]
49     )
50 }

```

Εικόνα 4.2.1

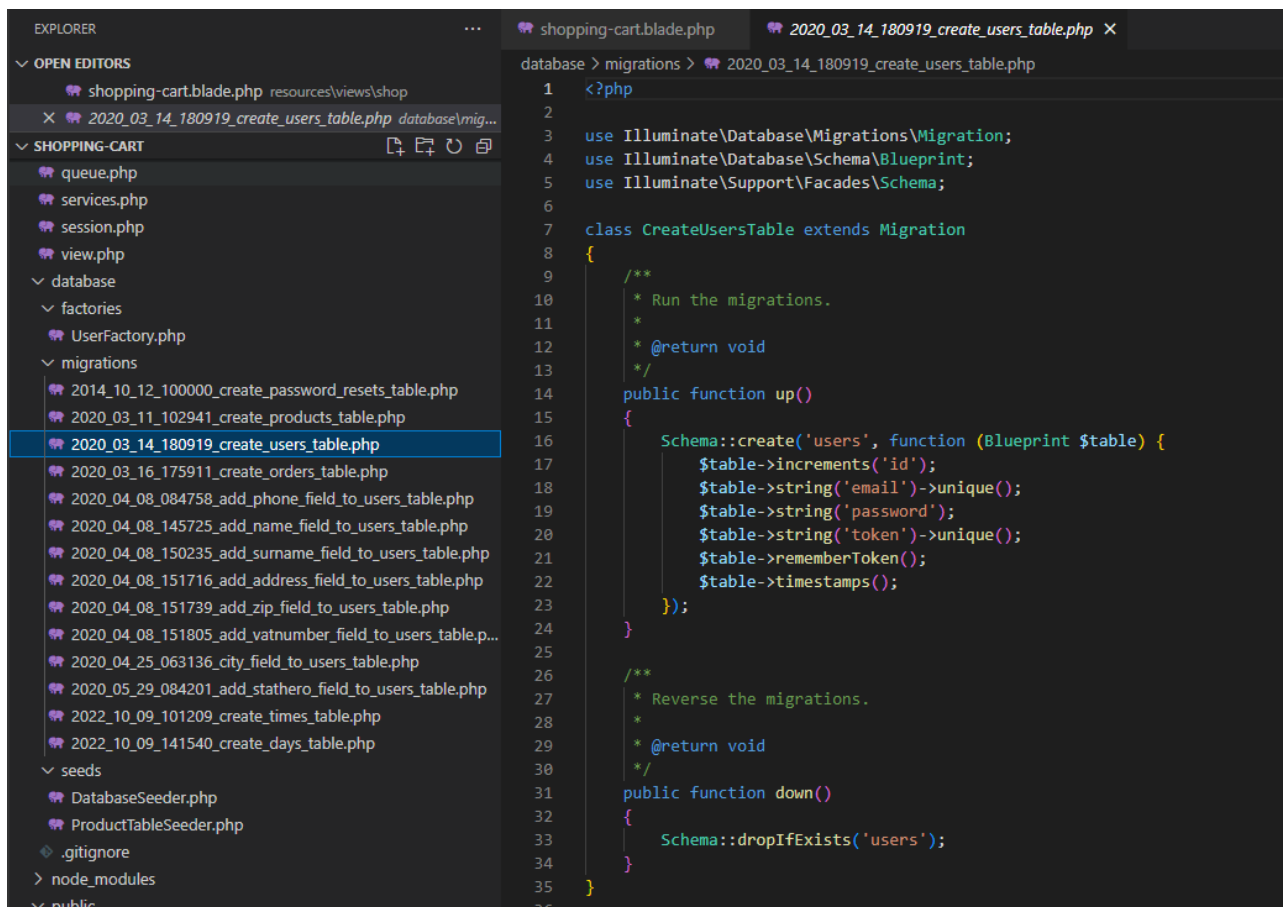
Αξίζει να αναφερθεί ότι στη γραμμή 17 απαιτείται η παρουσία της συνάρτησης "getSignup" προκειμένου να εμφανίζεται και να φορτώνεται η εικόνα. Συγκεκριμένα, αυτή η συνάρτηση λαμβάνει τα πεδία που έχει εισαγάγει ο χρήστης, όπως το email, τον κωδικό πρόσβασης, το τηλέφωνο, την πόλη και τον ταχυδρομικό κώδικα, μέσω της πλατφόρμας εγγραφής στη βάση δεδομένων. Στη συνέχεια του κεφαλαίου, με βάση αυτό το παράδειγμα, συνεχίζεται να περιγράφεται η λειτουργία της βάσης δεδομένων.

Αυτή η διευκρίνιση εξηγεί πως λειτουργεί η συνάρτηση "getSignup" και πως συνδέεται με την εικόνα που φορτώνεται. Μετά από αυτήν την επεξήγηση, συνεχίζεται να

περιγράφεται η λειτουργία της βάσης δεδομένων σύμφωνα με το παράδειγμα που παρουσιάζεται.

4.3 Database: Migrations

Ένα από τα πιο βασικά κομμάτια της περιγραφής της εφαρμογής αυτής είναι το τι γίνεται με τη βάση δεδομένων. Στο συγκεκριμένο παράδειγμα η δημιουργία controllers και βάσεων δεδομένων κλπ. γίνεται μέσω τερματικού. Η εντολή: “php artisan make:migration create_users_table” όπως φαίνεται στην εικόνα 4.3.1



The image shows a screenshot of an IDE with two panes. The left pane is the Explorer, showing a file tree for a project named 'shopping-cart'. The 'migrations' folder is expanded, and the file '2020_03_14_180919_create_users_table.php' is selected. The right pane shows the content of this migration file. The code is as follows:

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateUsersTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      * @return void
12      */
13     public function up()
14     {
15         Schema::create('users', function (Blueprint $table) {
16             $table->increments('id');
17             $table->string('email')->unique();
18             $table->string('password');
19             $table->string('token')->unique();
20             $table->rememberToken();
21             $table->timestamps();
22         });
23     }
24
25     /**
26      * Reverse the migrations.
27      * @return void
28      */
29     public function down()
30     {
31         Schema::dropIfExists('users');
32     }
33 }
34
35 }
```

Εικόνα 4.3.1

δημιουργεί έναν πίνακα με χρήστες(users) όπου τα πεδία είναι το email και το password. Όταν χρειαστεί να γίνει κάποια αλλαγή στη βάση δεδομένων θα πρέπει να εκτελείται η εντολή “php artisan migrate:refresh –seed” ώστε να γίνει η αλλαγή στον server, γιατί αυτή θα γίνει μόνο τοπικά.

4.4 Middleware

Το Middleware παρέχει έναν βολικό μηχανισμό για την επιθεώρηση και το φιλτράρισμα των αιτημάτων HTTP που εισέρχονται στην εφαρμογή. Για παράδειγμα, η Laravel περιλαμβάνει ένα ενδιάμεσο λογισμικό που επαληθεύει ότι ο χρήστης της εφαρμογής έχει πιστοποιηθεί. Εάν ο χρήστης δεν έχει πιστοποιηθεί, τότε μέσω του ενδιάμεσου λογισμικού θα μεταφερθεί στην οθόνη σύνδεσης της εφαρμογής. Αντίθετα, εάν ο χρήστης έχει πιστοποιηθεί, το ενδιάμεσο λογισμικό θα επιτρέψει στο αίτημα να προχωρήσει περαιτέρω στην εφαρμογή.

Πρόσθετο ενδιάμεσο λογισμικό μπορεί να γραφτεί για την εκτέλεση ποικίλων εργασιών εκτός από τον έλεγχο ταυτότητας. Για παράδειγμα, ένα ενδιάμεσο λογισμικό καταγραφής μπορεί να καταγράφει όλα τα εισερχόμενα αιτήματα στην εφαρμογή. Υπάρχουν πολλά ενδιάμεσα προγράμματα που περιλαμβάνονται στο πλαίσιο Laravel, συμπεριλαμβανομένου του ενδιάμεσου λογισμικού για έλεγχο ταυτότητας και προστασίας CSRF(Cross-Site Request Forgery), που είναι ένας τρόπος ασφάλειας από παραβιάσεις κακόβουλων ενεργειών. Όλα αυτά τα ενδιάμεσα προγράμματα βρίσκονται στον κατάλογο `app/Http/Middleware`. Στην εικόνα 4.4.1 φαίνεται η διαφορά στις διαδρομές που αφορούν τον μη εγγεγραμμένο χρήστη. Δηλαδή μπορεί ο χρήστης να κάνει την κράτηση αλλά όχι να προχωρήσει στην πληρωμή, αφού προϋπόθεση για αυτήν είναι η εγγραφή. Αυτό γίνεται στην γραμμή 75 και 97 όπου στην γραμμή 75 αφορά την ελεύθερη πρόσβαση ενώ στην γραμμή 97 πρέπει ο χρήστης να δηλώσει τα στοιχεία του ώστε να γίνει εγγραφή.


```
75 Route::group(['middleware' => 'guest'], function () {
76
77     Route::get('/signup', [
78         'uses' => 'UserController@getSignup',
79         'as' => 'user.signup0'
80     ]);
81     Route::post('/signup', [
82         'uses' => 'UserController@postSignup',
83         'as' => 'user.signup'
84     ]);
85
86     Route::get('/signin', [
87         'uses' => 'UserController@getSignup',
88         'as' => 'user.signin0'
89     ]);
90
91     Route::post('/signin', [
92         'uses' => 'UserController@postSignup',
93         'as' => 'user.signin'
94     ]);
95
96 });
97
98 Route::group(['middleware' => 'auth'], function () {
99
100     Route::get('/profile', [
101         'uses' => 'UserController@getProfile',
102         'as' => 'user.profile'
103     ]);
104
105     Route::get('/logout', [
106         'uses' => 'UserController@getLogout',
107         'as' => 'user.logout'
108     ]);
109
110     Route::get('/user/{user}/', [
111         'as' => 'user.editprofile',
112         'uses' => 'UserController@edit'
113     ]);
114
115     //Route::patch('/user/{user}/update', [
116     //    'as' => 'user.update',
117     //    'uses' => 'UserController@update'
118     // ]);
119
120     Route::get('/change-password', [
121         'uses' => 'ChangePasswordController@index',
122         'as' => 'user.changePassword'
123     ]);
124
125     Route::post('/change-password', [
126         'uses' => 'ChangePasswordController@store',
127         'as' => 'change.password'
128     ]);
129
130     Route::patch('users/update-profile', 'UserController@update')->name('users.update-profile');
131     Route::delete('users/delete-profile', 'UserController@destroy')->name('users.delete-profile');
```

Εικόνα 4.4.1

5. Υλοποίηση

Στο κεφάλαιο αυτό θα αναλυθεί η αρχή και το τέλος της δημιουργίας και εκτέλεσης της εφαρμογής που περιεγράφηκε στο προηγούμενο κεφάλαιο. Η εγκατάσταση MAMP, composer, NodeJS, Laravel framework και η δημιουργία του project που είναι απαραίτητη για να λειτουργήσει το έργο αυτό, η διαδρομή που είναι πολύ σημαντική για να “τρέξει” το composer μέσω του MAMP κατά την διάρκεια της δημιουργίας του έργου και τέλος ο ρόλος του NodeJS, καθώς και η εγκατάστασή του.

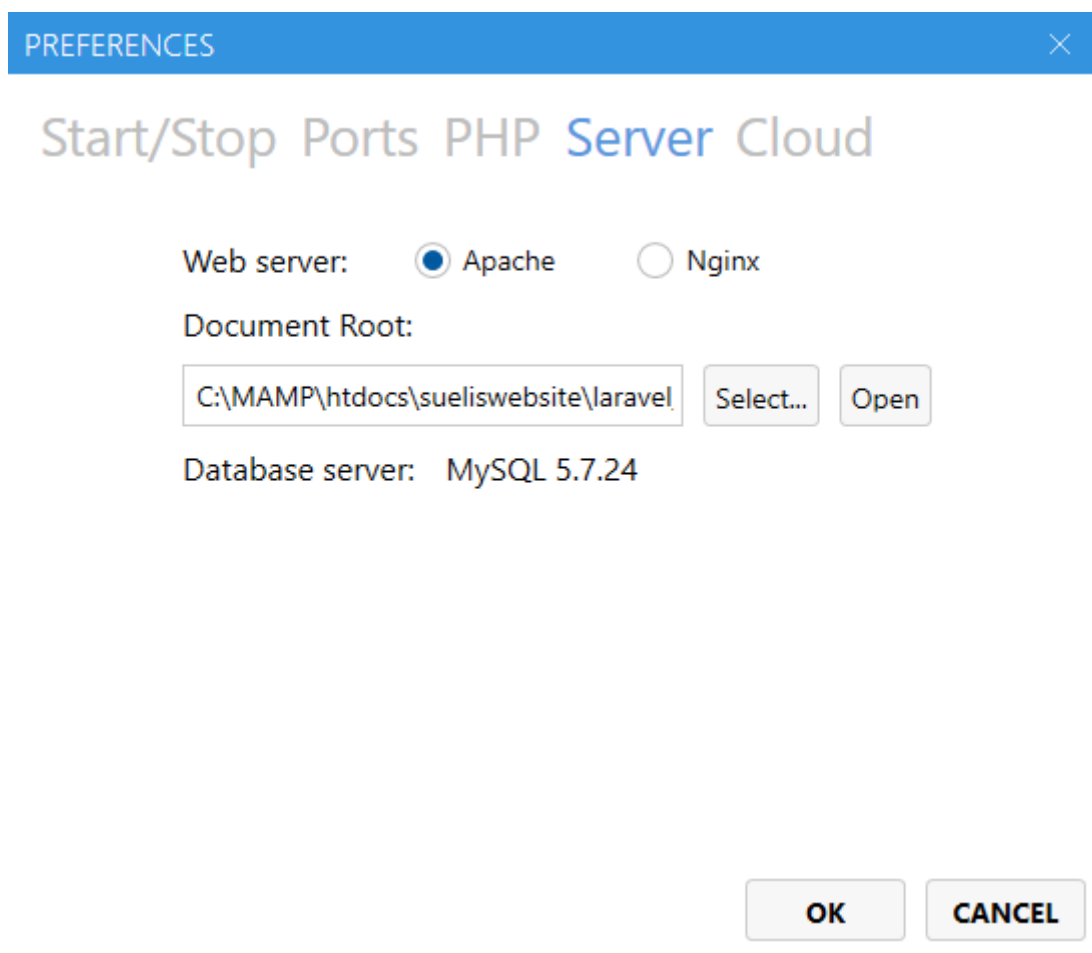
5.1 Δημιουργία Laravel project

Χρησιμοποιώντας το τερματικό και εκτελώντας την εντολή: `composer create-project laravel/laravel example-app` πολύ εύκολα δημιουργείται το project μέσω του composer. Μετά την ολοκλήρωση της εγκατάστασης, μέσω του τερματικού με την εντολή `cd` εντοπίζουμε το όνομα του φακέλου του project που βρίσκεται στον φάκελο του MAMP στην εξής διαδρομή: `C:\MAMP\htdocs\project` και όπου `project` το όνομα του έργου που έχουμε επιλέξει με βάση την εντολή: `composer create-project laravel/laravel example-app` που το όνομα το επιλέγουμε αντί ονόματος `example-app`. Ο συγκεκριμένος τρόπος δημιουργίας στηρίζεται με λογισμικό WindowsOS και είναι ο πιο απλός και κατανοητός για κάποιον στην χρήση της Laravel framework. Συγκεκριμένα ακολούθησα τη σειρά βίντεο για να υλοποιήσω το έργο αυτό και να το προσαρμόσω στις ανάγκες της καφετέριας. Μπορεί να γίνει και με χρήση του Docker, παραδείγματος χάρη ή Homebrew. Κάθε οδηγός εγκατάστασης σε λογισμικό όπως το MacOS, αναφέρεται στην ηλεκτρονική διεύθυνση <https://laravel.com/docs/9.x>, με αναζήτηση του όρου `Laravel installation`.

5.2 Εγκατάσταση MAMP και Composer

Πρώτο βήμα είναι η λήψη MAMP από την ιστοσελίδα <https://www.mamp.info/en/windows/>. Στη συνέχεια άνοιγμα του αρχείου και εγκατάσταση με βάση τον οδηγό εγκατάστασης όπως σε κάθε άλλο πρόγραμμα. Αξίζει να σημειωθεί ξανά πως οι περισσότεροι χρήστες χρησιμοποιούν το XAMP το οποίο κάνει

την ίδια εργασία με το MAMP με τη διαφορά ότι η διαχείριση με γραφικό περιβάλλον είναι απλούστερη σε σχέση με το XAMP. Εφόσον γίνει εγκατάσταση και πλέον είναι σε διαθεσιμότητα το πρόγραμμα, πρέπει να γίνεται επιλογή από τις προτιμήσεις στο μενού(Preferences) για τη διαδρομή του έργου όπως φαίνεται στην εικόνα 5.2.1 Πολλή προσοχή απαιτείται στην προεπιλεγμένη διαδρομή η οποία θα χρησιμοποιηθεί στην εγκατάσταση του composer.



Εικόνα 5.2 1

Στη συνέχεια ο χρήστης κάνει κλικ στο OK και βρίσκετε σε αναμονή μέχρι να ξεκινήσει ο server, που σημαίνει πως πρασινίζουν τα δύο “λαμπάκια”, όπως φαίνεται στην εικόνα 5.2.8 Όσο αφορά την εγκατάσταση composer, η οποία είναι η πιο σημαντική εγκατάσταση από όλες με λογισμικό WindowsOS και η πιο απλή καθώς η λήψη γίνεται από <https://getcomposer.org/>. Οι διαδικασίες φαίνονται στην εικόνα 5.2.1 και 5.2.2.



A Dependency Manager for PHP

Latest: [2.1.12](#) ([changelog](#))



[Getting Started](#)

[Download](#)

[Documentation](#)

[Browse Packages](#)

[Issues](#)

[GitHub](#)

Εικόνα 5.2.2

[Home](#) | [Getting Started](#) | [Download](#) | [Documentation](#) | [Browse Packages](#)

Download Composer

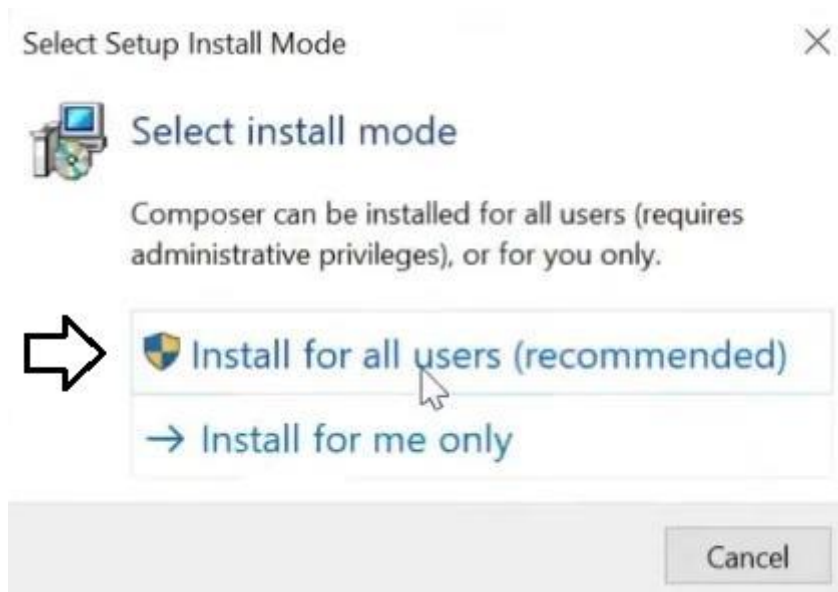
 Latest: v2.1.12

Windows Installer

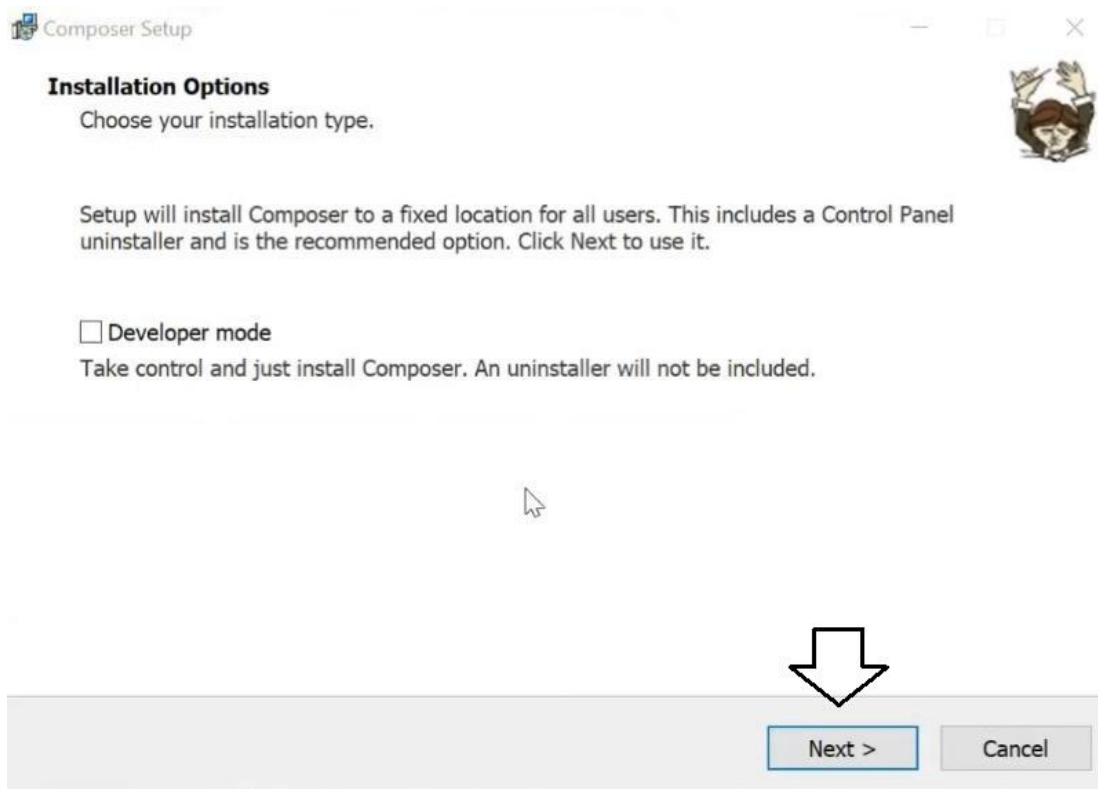
The installer - which requires that you have PHP already installed - will download Composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

Download and run [Composer-Setup.exe](#) it will install the latest composer version whenever it is executed.

Εικόνα 5.2.3

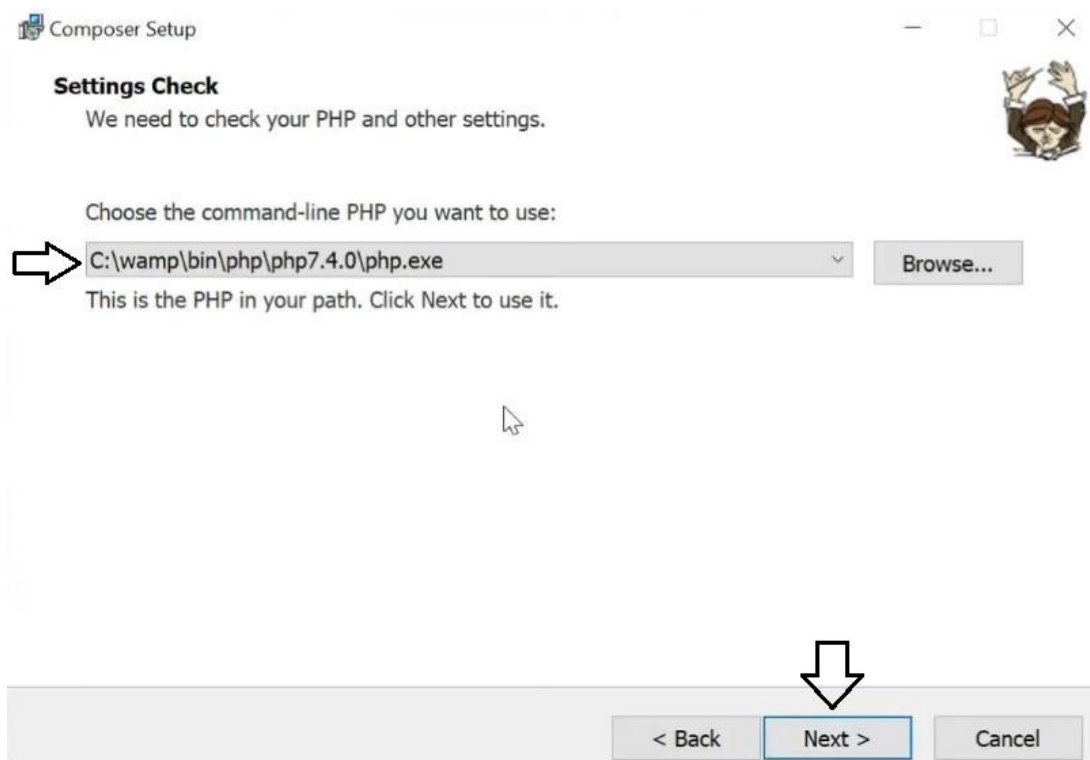


Εικόνα 5.2.4



Εικόνα 5.2.5

Στην εικόνα 5.2.6 φαίνεται το πιο σημαντικό στάδιο, διότι από αυτό καθορίζεται η διαδρομή της PHP όπου βρίσκεται στον φάκελο που έγινε η εγκατάσταση του MAMP, ο οποίος είναι: C:\MAMP\bin\php\php.8.1.0\php.exe και το μέρος της διαδρομής με την έκδοση της php της φωτογραφίας διαφέρει από την τρέχουσα έκδοση που υπάρχει σήμερα.



Εικόνα 5.2.6

Μετά τη λήψη ακολουθεί η εγκατάσταση και για τον σωστό ορισμό διαδρομής ανοίγουμε το Visual Studio Code και με τη χρήση του τερματικού επιλέγοντας από το μενού αν η γλώσσα είναι στην Αγγλική, Terminal, και τέλος `new terminal`. Γράφοντας την εντολή: `composer` θα εμφανίσει η εικόνα 5.2.7 που σημαίνει πως εγκαταστάθηκε σωστά το πρόγραμμα. Διαφορετικά ακολουθήθηκε λάθος διαδρομή ή υπάρχει κάποιο άλλο πρόβλημα. Πρώτα πρέπει να γίνει η εγκατάσταση του MAMP και μετά του Composer για τον λόγο πως το composer πρέπει να “τρέξει” πάνω στο MAMP. Στην εικόνα 5.2.8 φαίνεται πως το MAMP βρίσκεται σε λειτουργία.

```

C:\WINDOWS\system32\cmd.exe
C:\Users\JTP>composer

Composer version 2.0.8 2020-12-03 17:20:38

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display this help message
  -q, --quiet                Do not output any message
  -V, --version              Display this application version
  --ansi                     Force ANSI output
  --no-ansi                  Disable ANSI output
  -n, --no-interaction      Do not ask any interactive question
  --profile                  Display timing and memory usage information
  --no-plugins               Whether to disable plugins.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache                 Prevent use of the cache
  -v|vv|vvv, --verbose      Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and
                             3 for debug

Available commands:
  about      Shows the short information about Composer.
  archive    Creates an archive of this composer package.

```

Εικόνα 5.2.7



Εικόνα 5.2.8

5.3 Σχετικά με το NodeJS

Το Node.js είναι ένα νέα τεχνολογία, μια σειρά εντολών που εκτελούνται μέσω του τερματικού από το πρόγραμμα visual studio code. Οι εντολές αυτές μπορούν να ρυθμίσουν κάποιες από τις πολλές δυνατότητες, όπως να παράγουν δυναμικό περιεχόμενο ιστοσελίδας προτού σταλεί η σελίδα στο πρόγραμμα περιήγησης(browser) ιστού του χρήστη. Παραδείγματος χάρη, κάθε φορά που γράφει μια πρόταση ο προγραμματιστής για να φανεί στην ιστοσελίδα, να εμφανίζεται η πρόταση αυτή πάντα στα κεφαλαία χωρίς να χρειάζεται να το ορίσει από το πληκτρολόγιο ο ίδιος. Στην ουσία γίνεται επικοινωνία μεταξύ του προγραμματιστή και του κεντρικό υπολογιστή ή αλλιώς εξυπηρετητής. Στην ουσία είναι ένας υπολογιστής που εκτελεί εντολές από άλλους υπολογιστές και είναι υπεύθυνος για τη λειτουργία του δικτύου. Επίσης πολύ σημαντικό είναι πως ο διαχειριστής των εντολών της NodeJS επιτρέπει να χρησιμοποιήσει πολλές άλλες λειτουργίες. Μπορεί να γίνει λήψη από τη διεύθυνση <https://nodejs.org/en/download/>. Η εγκατάσταση είναι απλή και αρκεί ο χρήστης να ακολουθήσει τις οδηγίες εγκατάστασης. Δεν χρειάζεται κάποια ιδιαίτερη προσοχή ή αλλαγή στο σημείο αυτό. Εφόσον γίνει η λήψη και η εγκατάσταση με την εντολή “node -v” θα εμφανιστεί η εικόνα 5.3.1 που σημαίνει πως πρόκειται για την τρέχουσα έκδοση και ταυτόχρονα πως η εγκατάσταση στέφθηκε με επιτυχία.



```
Administrator: Command Prompt
C:\>node -v
v14.15.3
C:\>npm -v
6.14.9
C:\>
```

Εικόνα 5.3.1

Η NodeJS μπορεί να βοηθήσει στη χρήση sass αρχείων της Cascade Style Sheet. Πιο συγκεκριμένα διευκολύνει στη σύνταξη και χρήση της Cascade Style Sheet, διαμορφώνοντας έτσι το στυλ της ιστοσελίδας που εμφανίζεται στον χρήστη.

5.4 Παραμετροποίηση αρχείου .env

Το αρχείο .env είναι σημαντικό αρχείο, όπως αναφέρθηκε και πιο πάνω, και ένας από τους βασικότερους λόγους είναι η σύνδεσή του με τη βάση δεδομένων που δημιουργείται

με την εντολή “php artisan migrate”. Ο κωδικός και το όνομα χρήστη δεν έχουν οριστεί. Αν ο χρήστης θελήσει να συνδεθεί σε μια βάση δεδομένων, τότε θα χρειαστεί να εισαχθεί το όνομα της βάσης δεδομένων, το όνομα χρήστη και ο κωδικός στα πεδία: DB_DATABASE, DB_USERNAME, DB_PASSWORD της εικόνας 5.4.1. Επίσης πολύ σημαντικό είναι όταν θέλουμε να στείλουμε ένα email στον πελάτη που έκανε την κράτηση, να ορίσουμε το email μας ώστε να φανεί το όνομα του αποστολέα στον χρήστη στον οποίο θα σταλθεί το email.

```
.env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:2WLuKXcwDiXTI3II1X30Pwq7XqIQwEebKlZ70ydGqfU=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8
9 DB_CONNECTION=mysql
10 DB_HOST=localhost
11 DB_PORT=3306
12 DB_DATABASE=laravel
13 DB_USERNAME=root
14 DB_PASSWORD=root
15
16 BROADCAST_DRIVER=log
17 CACHE_DRIVER=file
18 QUEUE_CONNECTION=sync
19 SESSION_DRIVER=file
20 SESSION_LIFETIME=120
21
22 REDIS_HOST=localhost
23 REDIS_PASSWORD=null
24 REDIS_PORT=6379
25
26 MAIL_MAILER=smtp
27 MAIL_HOST=smtp.gmail.com
28 MAIL_PORT=25
29 MAIL_USERNAME=konstantinosiakovou@gmail.com
30 MAIL_PASSWORD=ffjtzndzdepcpfvo
31 MAIL_ENCRYPTION=tls
32 MAIL_FROM_ADDRESS=konstantinosiakovou@gmail.com;
33 MAIL_FROM_NAME="${APP_NAME}"
```

Εικόνα 5.4.1

Όταν το project ανέβει σε διακομιστή, παραδείγματος χάρη στο HostGator.com, η γραμμή 1 θα πρέπει να περιλαμβάνει το όνομα της εφαρμογής σε εισαγωγικά “ ” και η γραμμή 5 την διεύθυνση <http://localhost> και όπου localhost η διεύθυνση του domain που

έχει αγοραστεί από τον χρήστη. Επίσης σημαντικό είναι όταν βρίσκεται η εφαρμογή σε κάποιον διακομιστή να περιέχει το αρχείο `.htaccess` και `index.php` όπου με τις κατάλληλες παραμέτρους μπορεί να φορτώσει η εφαρμογή και να γίνει η προβολή της όπως φαίνεται στην εικόνα 5.4.1 και 5.4.2.

```
1 |Options +Indexes
2 RewriteCond %{HTTP_HOST} ^parceirodeconversacao\.com\.br$ [OR]
3 RewriteCond %{HTTP_HOST} ^www\.parceirodeconversacao\.com\.br$
4 RewriteRule ^public_html/public$ "http://www\.parceirodeconversacao\.com\.br/" [R=301,L]
5
6 # php -- BEGIN cPanel-generated handler, do not edit
7 # Defina o pacote "ea-php81" como a linguagem padrão de programação "PHP".
8 <IfModule mime_module>
9     AddHandler application/x-httpd-ea-php81 .php .php8 .phtml
10 </IfModule>
11 # php -- END cPanel-generated handler, do not edit
12
```

Εικόνα 5.4.2

```

1 |<?php
2 | define('LARAVEL_START', microtime(true));
3 | /*
4 | -----
5 | | Register The Auto Loader
6 | | -----
7 | |
8 | | Composer provides a convenient, automatically generated class loader for
9 | | our application. We just need to utilize it! We'll simply require it
10 | | into the script here so that we don't have to worry about manual
11 | | loading any of our classes later on. It feels great to relax.
12 | |
13 | */
14 | require __DIR__.'/../laravel/vendor/autoload.php';
15 | /*
16 | | -----
17 | | Turn On The Lights
18 | | -----
19 | |
20 | | We need to illuminate PHP development, so let us turn on the lights.
21 | | This bootstraps the framework and gets it ready for use, then it
22 | | will load up this application so that we can run it and send
23 | | the responses back to the browser and delight our users.
24 | |
25 | */
26 | $app = require_once __DIR__.'/../laravel/bootstrap/app.php';
27 | $kernel = $app->make(Illuminate\Contracts\Http\Kernel::class);
28 | $response = $kernel->handle(
29 |     $request = Illuminate\Http\Request::capture()
30 | );
31 | $response->send();
32 | $kernel->terminate($request, $response);

```

Εικόνα 5.4.3

Τα 2 αυτά αρχεία .htaccess και index.php είναι παράδειγμα από μία δική μου εφαρμογή, που έχω δημιουργήσει για τις ανάγκες μιας μικρής εταιρείας διδασκαλίας της Αγγλικής Γλώσσας στη Βραζιλία και το domain, δηλαδή η ηλεκτρονική διεύθυνση: parceirodeconversacao.com.br.

6.Αποτέλεσμα

Από όλη τη διαδικασία υλοποίησης της εφαρμογής το κεφάλαιο αυτό καταλήγει στις θετικές αλλά και στις αρνητικές επιπτώσεις που προκύπτουν από την εφαρμογή και την υλοποίησή της.

Οι επιπτώσεις της εφαρμογής και τα συμπεράσματα που απορρέουν από την υλοποίησή της, δηλαδή πως μπορεί η Laravel να εξυπηρετήσει μια καφετέρια. Υποστηρίζοντας τις βασικές λειτουργίες του μικρού ηλεκτρονικού καταστήματος, όπως είναι η εγγραφή πελάτη, η διαχείριση προσωπικών στοιχείων και η λειτουργία της αγοράς μέσω stripe. Η εφαρμογή χρησιμοποιεί μια βάση δεδομένων μέσω του phpMyAdmin και συνδέεται με αυτήν για να καταγράφει τα προϊόντα και τους χρήστες. Αφού ολοκληρωθεί μια αγορά, ο πελάτης επιβεβαιώνεται μέσω ηλεκτρονικού ταχυδρομείου και ενημερώνεται για την ολοκλήρωση της εγγραφής του.

6.1 Εγγραφή χρήστη

Όπως αναφέρθηκε από την αρχή των κεφαλαίων η εγγραφή χρήστη απαιτείται για την κράτηση και αγορά ενός προϊόντος, συγκεκριμένα μιας από τις τρεις δραστηριότητες ιππασία, καλλιτεχνία ή ξενάγηση στα χωριά της Κύμης. Με βάση τη συνάρτηση postSignup στον UserController εφόσον ο χρήστης πατήσει το κουμπί υποβολή, στοιχεία όπως: name, surname κλπ. καταγράφονται στη βάση δεδομένων που έχει δημιουργηθεί. Τονίζεται πως η συνάρτηση είναι υπεύθυνη για οτιδήποτε επιθυμεί ο προγραμματιστής με βάση τις απαιτήσεις του πελάτη. Τα στοιχεία που έχει εισαγάγει ο χρήστης έχουν ήδη οριστεί από την επεξεργασία βάσης δεδομένων και επίσης έχουν οριστεί κανόνες που πρέπει να τηρηθούν. Αν ο χρήστης για παράδειγμα δεν έχει δώσει σωστά κάποιο στοιχείο, όπως για παράδειγμα το email του, τότε εμφανίζεται μήνυμα σφάλματος. Αυτό φαίνεται στην εικόνα 6.1.1. Μετά τη εγγραφή του ο χρήστης θα μπορεί να συνδεθεί με το email του και τον κωδικό πρόσβασης. Στην εικόνα 6.1.2 φαίνεται το περιβάλλον

Sign Up

Name:

Surname:

Password:
 Show Password

Your password must be more than 8 characters long,
should contain at-least 1 Uppercase, 1 Lowercase, 1 Numeric
and 1 special character.

Password Confirmation:
 Show Password

E-Mail Address:

Confirm Email

Phone Number:

Σταθερό
τηλέφωνο(προαιρετικό):

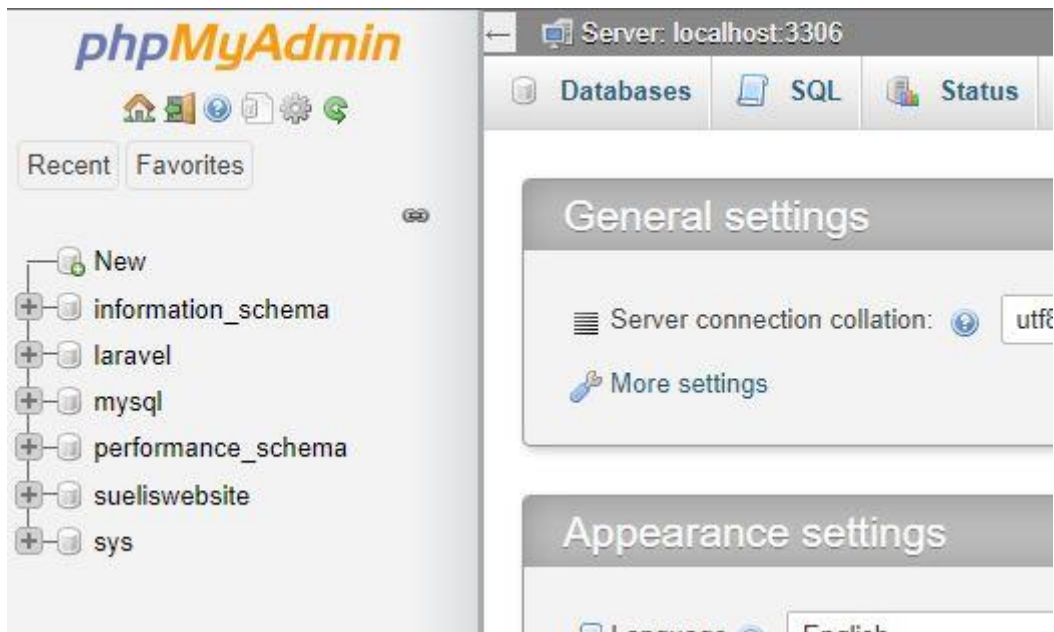
Present Address:

City:

Zip:

Vatnumber:
Optional

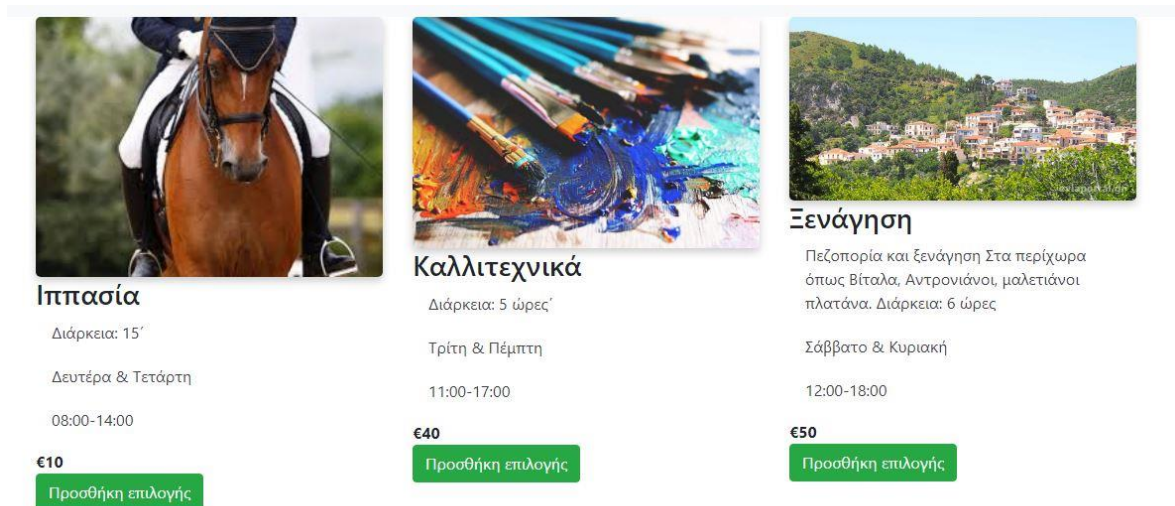
Εικόνα 6.1.1



Εικόνα 6.1 2

6.2 Αγορά προϊόντος μέσω stripe

Εφόσον ο χρήστης επιλέξει το προϊόν που θέλει όπως δείχνει η εικόνα 6.2.1,



Εικόνα 6.2 1

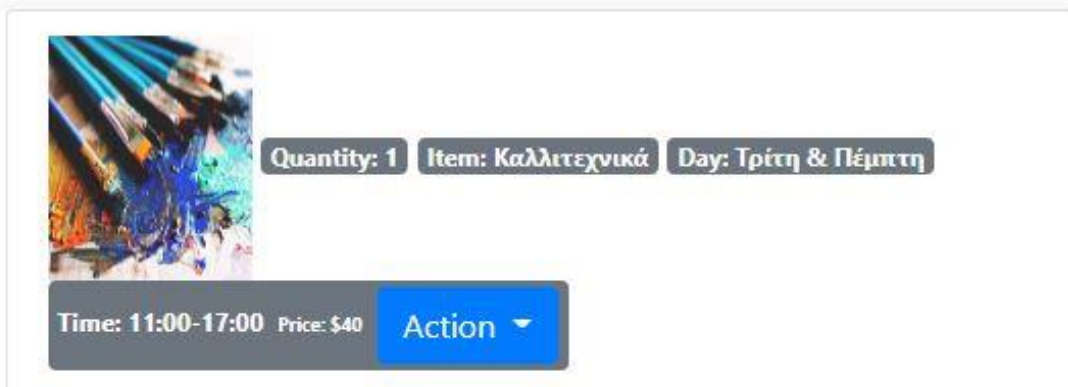
Και αφού έχει κάνει την εγγραφή, θα οδηγηθεί στο καλάθι αγορών και στη συνέχεια στην πληρωμή. Σε αυτό το σημείο επέλεξα την γνωστή μέθοδο Stripe. Είναι μία από τις δημοφιλείς πύλες πληρωμών στο διαδίκτυο που δέχονται πληρωμές με πιστωτική ή

χρεωστική κάρτα. Αφού δημιουργηθεί το migration table της εικόνας 6.2.2

```
1 public function up()  
2 {  
3     Schema::create('payments', function (Blueprint $table) {  
4         $table->bigIncrements('id');  
5         $table->string('payment_id');  
6         $table->string('payer_email');  
7         $table->float('amount', 10, 2);  
8         $table->string('currency');  
9         $table->string('payment_status');  
10        $table->timestamps();  
11    });  
12 }
```

Εικόνα 6.2.2

η κατάλληλη δομή κώδικα στο checkout.blade.php και η σωστή διαδρομή στο web.php, καθώς επίσης και στο Productcontroller, τότε τα δεδομένα της κάρτας στέλνονται στη βάση δεδομένων του λογαριασμού που έχει δημιουργήσει ο χρήστης στο stripe.com και καταχωρείται το id της πληρωμής, το κόστος του προϊόντος και το ποσό που παρακρατεί ένα ποσό η stripe για την υπηρεσία που προσφέρει. Η μορφή που παρουσιάζει το καλάθι φαίνεται στην εικόνα 6.2.3.



Total Cost: €40

[Checkout](#)

Εικόνα 6.2.3

Να σημειωθεί ότι ο χρήστης έχει τη δυνατότητα να σβήσει από το καλάθι προϊόν που δε θέλει. Η πλατφόρμα δήλωσης κάρτας για την πληρωμή φαίνεται στην εικόνα 6.2.4.

The image shows a payment form with the following sections:

- Email:** A text input field.
- Card information:** A section containing:
 - Card number: 1234 1234 1234 1234
 - Card logos: VISA, Mastercard, American Express, Discover
 - Expiration date: MM / YY
 - CVC: A three-digit security code field
- Name on card:** A text input field.
- Country or region:** A dropdown menu currently showing "United States".
- ZIP:** A text input field.
- Pay \$65.00:** A dark blue button with white text.

Εικόνα 6.2.4

6.3 Επιβεβαίωση αγοράς

Η παράγραφος αυτή αναφέρεται στην ειδοποίηση του πελάτη μέσω ηλεκτρονικού ταχυδρομείου. Συγκεκριμένα, εφόσον ο πελάτης έχει κάνει την αγορά και έχει πληρώσει, λαμβάνει μήνυμα επιβεβαίωσης με τα στοιχεία του προϊόντος που αγόρασε. Στην εικόνα 6.3.1 φαίνεται η συνάρτηση `payondelivery()` στο `postController` που δημιουργήθηκε για τον σκοπό αυτό.


```

103 public function payondelivery(){
104     Mail::send('shop.order_confirmation2', [
105         'user' => Auth()->user(),
106         'products' => $cart->items,
107         'totalPrice' => $cart->totalPrice,
108     ], function($message) use ($user) {
109         $message->to($user->email);
110         $message->from("konstantinosiakovou@gmail.com");
111         $message->subject("Your order confirmation");
112         $message->bcc("konstantinosiakovou@gmail.com");
113     }
114 );
115 return redirect()->route('product.index')->with('success', 'Successfully purchased products! | You w
116 }
117 }
118

```

Εικόνα 6.3 1

Τα βασικά στοιχεία όπως το όνομα του προϊόντος και η τιμή περιλαμβάνονται μέσα από τον κώδικα. Αναλυτικότερα στη συνάρτηση function, χρησιμοποιούνται 2 μεταβλητές, το \$message και τον \$user. Θεωρούνται και οι δύο πολύ σημαντικές διότι με τη μεταβλητή \$user καθορίζεται σε ποίο email και από ποιόν χρήστη θα σταλθεί το μήνυμα σύμφωνα με τα στοιχεία που καταχωρήθηκαν στη βάση δεδομένων κατά την διάρκεια εγγραφής. Η μεταβλητή \$message δηλώνει το μήνυμα στον κώδικα που στέλνεται. Αξίζει να ξανασημειωθεί πως για να μπορεί να γίνει η κράτηση και ενός από τα τρία προϊόντα θα πρέπει να προηγηθεί εγγραφή. Γενικά όμως για λόγους εμπορικούς σε επαγγελματικό και όχι ερασιτεχνικό e-shop ο χρήστης δεν είναι απαραίτητο να κάνει εγγραφή.

Η ειδοποίηση μέσω ηλεκτρονικού ταχυδρομείου είναι βασική λειτουργία για όλα τα e-shop, είναι δεοντολογικά σωστό να ενημερώσει την επιχείρηση για τα στοιχεία της αγοράς του.

6.4 Ενημέρωση στοιχείων στον λογαριασμό.

Αυτή είναι μία από τις βασικές λειτουργίες ενός ηλεκτρονικού καταστήματος. Αν ο χρήστης που έχει δημιουργήσει λογαριασμό και έχει κατοχυρώσει τα στοιχεία και έχει αλλάξει διεύθυνση κατοικίας, θα πρέπει να δηλώσει τα νέα του στοιχεία. Η διεύθυνση

και ο αριθμός τηλεφώνου είναι ένα τα πλέον απαραίτητα. Υπάρχουν τρεις συναρτήσεις για να υλοποιηθεί η δυνατότητα επεξεργασίας χρήστη. Οι συναρτήσεις αυτές είναι η `getProfile`, `edit` και `update` που φαίνονται στις εικόνες 6.4.1, 6.4.2, 6.4.3.

```
93     public function getProfile(){
94         ..... $orders = Auth::user()->orders;
95         ..... $orders->transform(function($order, $key){
96             ..... $order->cart = unserialize($order->cart);
97             ..... return $order;
98         ..... });
99         ..... return view('user.profile', ['orders'=> $orders]);
100     }
```

Εικόνα 6.4 1

```
107     public function edit(User $user)
108     {
109         ..... $user = Auth()->user();
110         ..... return view('user.editprofile', compact('user'));
111     }
```

Εικόνα 6.4 2

```

113     public function update()
114     {
115         $this->validate(request(), [
116             'email' => 'email|unique:users|confirmed',
117             'phone' => 'numeric',
118             'stathero' => 'numeric',
119             'name' => 'max:12',
120             'surname' => 'max:13',
121             'city' => 'max:12',
122             'address' => 'max:25',
123             'zip' => 'numeric|min:5',
124             'vatnumber' => 'nullable|regex:/[0-9]{10}/',
125         ],
126         [
127             'email.required' => 'An email required',
128             'email.unique' => 'This email is not unique',
129             'password.required' => 'A password required',
130             'phone.required' => 'A phone required',
131             'name.required' => 'A name required',
132             'surname.required' => 'A surname required',
133             'city.required' => 'A city required',
134             'address.required' => 'An address required',
135             'zip.required' => 'A zip required',
136         ],
137         );
138     }
139
140     $user = auth()->user();
141     $user->email = request('email');
142     $user->phone = request('phone');
143     $user->stathero = request('stathero');
144     $user->name = request('name');

```

Εικόνα 6.4 3

Να σημειωθεί πως χρησιμοποιούνται οι κανόνες εγγραφής που χρησιμοποιούνται και στη συνάρτηση postSignup στον ίδιο controller, userController.

Οι κανόνες εγγραφής είναι μια απλή φράση απαραίτητη για την περίπτωση που κάποιος χρήστης κατά την υποβολή δηλώσει από παραδρομής λάθος στοιχεία παραδείγματος χάρη [konstantinos\\$iakovou@gmail.com](mailto:konstantinos$iakovou@gmail.com) η konstantinosiakovougr. Τότε η εφαρμογή του εμφανίζει μήνυμα σφάλματος.

Σε καμιά περίπτωση δεν γίνεται αν ο πελάτης έχει ήδη αγοράσει ένα προϊόν και έχει καταχωρηθεί η αγορά του, να αλλάξουν τα στοιχεία άμεσα. Η αλλαγή θα φανεί κατά την επόμενη του αγορά.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Malatesta, F. (July 27, 2015). *Learning Laravel's Eloquent: Develop amazing data-based applications with Eloquent, the Laravel framework ORM*. Packt Publishing.

Martin, M. (July 15, 2019). *PHP: The Complete Guide for Beginners, Intermediate and Advanced Detailed Approach To Master PHP Programming*. Independently published.

Rod Johnson, J. H. (July 8, 2005). *Professional Java Development with the Spring Framework 1st Edition*. Wrox.

Sinha, S. (September 6, 2019). *Beginning Laravel: Build Websites with Laravel 5.8 2nd Edition, Kindle Edition*. Apress.

Staff, G. (2021, December 1). *www.guru.com/blog*. Ανάκτηση από [www.guru.com](https://www.guru.com/blog/the-pros-and-cons-of-laravel/#:~:text=The%20framework%27s%20strong%20points%20include,for%20updates%2C%20and%20slow%20speeds):
<https://www.guru.com/blog/the-pros-and-cons-of-laravel/#:~:text=The%20framework%27s%20strong%20points%20include,for%20updates%2C%20and%20slow%20speeds>

Stauffer, M. (April 20, 2019). *Laravel: Up & Running: A Framework for Building Modern PHP Apps 2nd Edition*. O'Reilly Media.

