



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ  
ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Δημιουργία εφαρμογής καταχώρησης προϊόντων σε  
ηλεκτρονικό κατάστημα»

A screenshot of a web form for API configuration. At the top left is the NTUA logo. The form has three input fields: "To url του eshop" with the value "http://192.168.2.5", "To Username για το API" with the value "dipaeapi", and "To key του API" with the value "VK31Zxg7orMgn488z4WPptuyilLcalkDWx". At the bottom is a blue button labeled "Είσοδος".

**Φοιτητής**

Κωστόπουλος Κωνσταντίνος 515068

**Επιβλέπων**

Δρ. Κυριάκος Τσιακμάκης

**Ιούνιος 2021**

Δημιουργία εφαρμογής καταχώρησης προϊόντων σε ηλεκτρονικό κατάστημα μέσω διαδικτύου

Κωδικός: 21166

Φοιτητής: Κωστόπουλος Κωνσταντίνος

Εισηγητής: Δρ Κυριάκος Τσιακμάκης

Ημερομηνία ανάληψης Π.Ε. 03-03-2021

Ημερομηνία περάτωσης Π.Ε. 30-05-2021

*Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως πτυχιακή εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.*

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Κωνσταντίνου Κωστόπουλου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.*

Η έγκριση της πτυχιακής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.



## Περίληψη

Η εργασία αυτή αφορά τη δημιουργία Android εφαρμογής γρήγορης καταχώρησης προϊόντος σε ηλεκτρονικό κατάστημα. Ο ιδιοκτήτης θα μπαίνει στην εφαρμογή, θα φωτογραφίζει το προϊόν, θα του βάζει τίτλο και τιμή και θα το ανεβάζει απευθείας στο eshop χωρίς να χάνει χρόνο και να πρέπει να συνδεθεί στο διαχειριστικό του eshop μέσω του υπολογιστή του. Για την υλοποίηση του ηλεκτρονικού καταστήματος επιλέχθηκε το OpenCart.

# « Implementation of mobile application for uploading and registering products in eshop »

## **Abstract**

This work concerns the implementation of an Android application for quick product registration in an online store. The owner will enter the application, photograph the product, give it a title and price and upload it directly to the eshop without wasting time in the eshop management. OpenCart was chosen for the implementation of the online store.

## **Ευχαριστίες**

Θέλω να ευχαριστήσω την οικογένεια μου για την υποστήριξη τους.

# Περιεχόμενα

Περίληψη .....	iv
Abstract .....	v
Ευχαριστίες .....	vi
Περιεχόμενα.....	vii
Κατάλογος Σχημάτων .....	ix
Κεφάλαιο 1ο: Εισαγωγή.....	1
1.1 Εισαγωγή.....	1
1.2 Δομή της εργασίας.....	1
Κεφάλαιο 2ο: Ηλεκτρονικό Κατάστημα και τρόποι διαχείρισης προϊόντος.....	3
2.1 Ηλεκτρονικά Καταστήματα .....	3
2.1.1 OpenCart .....	3
2.1.2 WooCommerce .....	4
2.1.3 Άλλα ηλεκτρονικά καταστήματα .....	5
2.2 Τρόποι διαχείρισης εισαγωγής προϊόντων .....	6
2.2.1 Opencart και τρόποι εισαγωγής προϊόντων.....	6
2.2.2 Opencart API και διαχείριση προϊόντων από το API.....	11
Κεφάλαιο 3ο: Συστήματα διαχείρισης και λογισμικά ανάπτυξης.....	14
3.1 Εισαγωγή.....	14
3.2 Opencart .....	14
3.2.1 Εγκατάσταση opencart.....	16
3.3 Flutter – Android Studio .....	25
Κεφάλαιο 4ο: Τα πρόσθετα στο ηλεκτρονικό κατάστημα .....	28
4.1 Εισαγωγή.....	28
4.2 Ρύθμιση του API στο opencart.....	29
4.3 Τροποποίηση στον κώδικα του opencart API.....	32
4.3.1 Αρχείο login.php .....	33
4.3.2 Αρχείο cart.php .....	34
4.3.3 Αρχείο productadd.php .....	38
4.3.4 Αρχείο session.php.....	40

4.4	Ανάλυση ασφάλειας στο σύστημα .....	40
Κεφάλαιο 5ο:	Η εφαρμογή στο κινητό.....	42
5.1	Εισαγωγή.....	42
5.2	Κώδικας στο Flutter .....	48
Κεφάλαιο 6ο:	Συμπεράσματα και προτάσεις βελτίωσης.....	58
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		59

## Κατάλογος Σχημάτων

Εικόνα 2.1: Μενού για εισαγωγή προϊόντος .....	6
Εικόνα 2.2: OpenCart – Καρτέλα προϊόντων .....	7
Εικόνα 2.3: Προσθήκη προϊόντος .....	7
Εικόνα 2.4: Προϊόν – Γενικά στοιχεία .....	8
Εικόνα 2.5: Προϊόν – Δεδομένα .....	8
Εικόνα 2.6: Προϊόν – Δεδομένα (το υπόλοιπο μέρος).....	9
Εικόνα 2.7: Προϊόν - Εικόνες .....	10
Εικόνα 2.8: Προϊόν – Προσθήκη εικόνων .....	10
Εικόνα 2.9: Opencart REST API - Shopping cart API 1 .....	12
Εικόνα 2.10: Opencart REST API - Shopping cart API 2 .....	12
Εικόνα 2.11: Opencart REST API - Shopping cart API 3 .....	13
Εικόνα 3.1: Λήψη OpenCart.....	16
Εικόνα 3.2: Το πρόγραμμα FileZilla.....	17
Εικόνα 3.3: Το πρόγραμμα phpMyAdmin.....	18
Εικόνα 3.4: Εγκατάσταση του OpenCart – Τα βήματα .....	19
Εικόνα 3.5: Εγκατάσταση OpenCart – Βήμα 4 .....	20
Εικόνα 3.6: Το OpenCart όταν γίνεται η εγκατάσταση (theme Jewellery)[13] .....	21
Εικόνα 3.7: Στοιχεία πρόσβασης στο login - admin .....	22
Εικόνα 3.8: Μετακίνηση storage φακέλου .....	22
Εικόνα 3.9: Το διαχειριστικό OpenCart .....	23
Εικόνα 3.10: Τα προϊόντα του ηλεκτρονικού καταστήματος .....	24
Εικόνα 3.11: Προϊόντα – Προσθήκη ή διαγραφή .....	24
Εικόνα 3.12: Προσθήκη νέου προϊόντος .....	24
Εικόνα 3.13: Flutter.....	26
Εικόνα 4.1: Ρύθμιση γλώσσας .....	28
Εικόνα 4.2: Ενεργοποίηση γλώσσας στο σύστημα .....	29
Εικόνα 4.3: API του Opencart.....	30
Εικόνα 4.4: API Username και Key .....	31
Εικόνα 4.5: Ρύθμιση IP για επιτρεπτές συνδέσεις.....	31
Εικόνα 4.6: API tokens.....	32
Εικόνα 4.7: Διαδικασία σύνδεσης και αποστολής προϊόντος μέσω του API .....	41
Εικόνα 5.1: Πρώτη οθόνη – Σύνδεση με το API .....	42
Εικόνα 5.2: Οθόνη για την προσθήκη προϊόντος .....	43
Εικόνα 5.3: Τιμές στα πεδία.....	44
Εικόνα 5.4: Ρύθμιση για διαστάσεις εικόνας του προϊόντος .....	45
Εικόνα 5.5: Αναζήτηση στο δίσκο του κινητού για επιλογή εικόνας.....	46
Εικόνα 5.6: Επιλογή και προβολή εικόνας.....	46
Εικόνα 5.7: Λάθος τιμής στο πεδίο Τιμή – Δεν είναι αριθμός.....	47

Εικόνα 5.9: Λάθος τιμής στο πεδίο Τιμή – Ο αριθμός είναι εκτός ορίων.....	47
Εικόνα 5.8: Λάθος στο πεδίο Τίτλος.....	47
Εικόνα 5.10: Προσπάθεια εκχώρησης προϊόντος χωρίς εικόνα.....	47
Εικόνα 5.11: Περίπτωση χρήσης του API χωρίς πρόσβαση ή εξουσιοδότηση.....	48

# Κεφάλαιο 1ο: Εισαγωγή

## 1.1 Εισαγωγή

Η εργασία αυτή αφορά τη δημιουργία Android εφαρμογής γρήγορης καταχώρησης προϊόντος σε ηλεκτρονικό κατάστημα. Ο ιδιοκτήτης θα μπαίνει στην εφαρμογή, θα φωτογραφίζει το προϊόν, θα του βάζει τίτλο και τιμή και θα το ανεβάζει απευθείας στο eshop χωρίς να χάνει χρόνο και να πρέπει να συνδεθεί στο διαχειριστικό του eshop μέσω του υπολογιστή του. Για την υλοποίηση του ηλεκτρονικού καταστήματος επιλέχθηκε το OpenCart. [1]

Το OpenCart ανήκει στις ηλεκτρονικές εμπορικές πλατφόρμες ανοιχτού κώδικα γραμμένο στη γλώσσα PHP. Είναι μια πολύ καλή λύση ηλεκτρονικού εμπορίου και δίνει τη δυνατότητα σε χρήστες-επιχειρηματίες να προωθήσουν την επιχείρησή τους στο διαδίκτυο και να συμμετέχουν στο ηλεκτρονικό εμπόριο με μικρό κόστος με εύκολο τρόπο. Το περιβάλλον διεπαφής με τον χρήστη έχει πολλές επιλογές που διευκολύνουν πολύ τις διαδικτυακές συναλλαγές-αγορές. Έχει ένα πολύ καλό διαχειριστικό (admin) και ένας απλός χρήστης μπορεί να εκτελέσει πολλές λειτουργίες και ρυθμίσεις για το ηλεκτρονικό του κατάστημα, όπως προσθήκη-τροποποίηση προϊόντων και κατηγοριών, διαφορετικούς τύπους πληρωμών και αποστολών.

Οι επιμέρους στόχοι της εργασίας είναι:

Υλοποίηση ενός ηλεκτρονικού καταστήματος με opencart

Παραμετροποίηση με τροποποίηση του κώδικα του ηλεκτρονικού καταστήματος για την επικοινωνία με την εφαρμογή android στο κινητό

Υλοποίηση εφαρμογής στο κινητό που να στέλνει τα βασικά στοιχεία στο ηλεκτρονικό κατάστημα για την άμεση προσθήκη του προϊόντος

## 1.2 Δομή της εργασίας

Στο πρώτο κεφάλαιο παρουσιάζεται μια εισαγωγή, οι στόχοι και η δομή της εργασίας.

Στο δεύτερο κεφάλαιο αναφέρεται μια αναφορά παρόμοιων συστημάτων για ηλεκτρονικά καταστήματα και εισαγωγής προϊόντων σε αυτά.

Στο τρίτο κεφάλαιο παρουσιάζονται τα συστήματα διαχείρισης και λογισμικά ανάπτυξης που χρησιμοποιήθηκαν για την υλοποίηση του ηλεκτρονικού καταστήματος και τη εφαρμογής android.

Στο τέταρτο κεφάλαιο παρουσιάζεται το ηλεκτρονικό κατάστημα που υλοποιήθηκε με το OpenCart με όλες τις λεπτομέρειες που αφορούν την τροποποίηση του κώδικα για να λειτουργήσει το API του όπως σχεδιάστηκε η εφαρμογή.

Στο πέμπτο κεφάλαιο παρουσιάζεται η εφαρμογή android που υλοποιήθηκε με Flutter που επικοινωνεί με το ηλεκτρονικό κατάστημα και παρουσιάζεται ο κώδικας του.

Στο τελευταίο κεφάλαιο παρουσιάζονται τα συμπεράσματα της εργασίας και θέματα για βελτίωση.

## **Κεφάλαιο 2ο: Ηλεκτρονικό Κατάστημα και τρόποι διαχείρισης προϊόντος**

### **2.1 Ηλεκτρονικά Καταστήματα**

Σήμερα υπάρχουν πάρα πολλά διαθέσιμοι κώδικες ηλεκτρονικού καταστήματος, άλλοι δωρεάν και άλλοι όχι. Εδώ θα αναφέρουμε τρία σημαντικά δωρεάν διαχειριστικά ηλεκτρονικών καταστημάτων ανοιχτού κώδικα και θα γίνει μια μικρή περιγραφή των βασικών χαρακτηριστικών τους. [2]

#### **2.1.1 OpenCart**

Το OpenCart είναι ένα σύστημα ηλεκτρονικού εμπορίου ανοιχτού κώδικα σε PHP. Παρέχει μια πολύ καλή λύση για το ηλεκτρονικό εμπόριο και επιτρέπει στους επιχειρηματίες να δημιουργήσουν τη δική τους διαδικτυακή επιχείρηση και να έχουν πλήρη διαχείριση των προϊόντων μέσω ιστοσελίδας. Η διεπαφή του χρήστη με την ιστοσελίδα δίνει πολλές επιλογές που διευκολύνουν τις διαδικτυακές αγορές. Το OpenCart είναι μια αυτόνομη πλατφόρμα ηλεκτρονικού εμπορίου και μπορεί να εγκατασταθεί με δύο κυρίως τρόπους. Μπορεί να εγκατασταθεί χειροκίνητα, βήμα βήμα, είτε χρησιμοποιώντας ένα πρόγραμμα εγκατάστασης. Η μη αυτόματη εγκατάσταση προσφέρει μια ποικιλία επιλογών και λειτουργιών προσαρμογής αλλά είναι δύσκολη και θα χρειαστεί η βοήθεια ενός επαγγελματία προγραμματιστή για τις ρυθμίσεις. Για τη δεύτερη μέθοδο μπορεί κάποιος να το κάνει μόνος. Θα πρέπει ο χρήστης να κάνει κλικ στο κουμπί "Εγκατάσταση" και τότε θα ξεκινήσει ο οδηγός εγκατάστασης. Το OpenCart είναι μια πλατφόρμα ανοιχτού κώδικα που είναι δωρεάν για λήψη και κατοχή. Να αναφερθεί ότι λειτουργεί με μοντέλο Core Extensions και απαιτεί πρόσθετα για την εκτέλεση λειτουργιών που κάνουν συγκεκριμένες λειτουργίες. Διαθέτει πάρα πολλά δωρεάν πρόσθετα και με πληρωμή, τα περισσότερα από τα οποία είναι κάτω των 35 €.

#### **Μειονεκτήματα OpenCart**

Ένα από τα πιο σημαντικά μειονεκτήματα είναι ότι δεν είναι ιδανικό για κλιμάκωση με πάρα πολλά προϊόντα. Παρέχει περιορισμένες δυνατότητες και επεκτάσεις που ενδέχεται να μην ικανοποιούν κάποιες επιχειρήσεις. Επίσης παρέχει περιορισμένες επιλογές προσαρμογής και οι επεκτάσεις ενδέχεται να μην είναι ασφαλείς αν δεν επιβλέπονται.

## 2.1.2 WooCommerce

Το WooCommerce[3] είναι ένα πρόσθετο για δημιουργία ηλεκτρονικού καταστήματος ανοιχτού κώδικα για το WordPress.[4] Εμφανίζεται ως εργαλείο για εμπόρους συνήθως λιανικής που επιθυμούν να δημιουργήσουν ένα ηλεκτρονικό κατάστημα με βάση το την ιστοσελίδα τους στο WordPress. Αφού εγκατασταθεί, ο επιχειρηματίας μπορεί να πουλάει τα προϊόντα του ή συνδρομές ή υπηρεσίες και άλλα πολλά.

Η εγκατάσταση του WooCommerce διαρκεί λίγα λεπτά και είναι κατάλληλο για χρήστες με μικρή τεχνική εξειδίκευση. Η διεπαφή του WordPress προσφέρει πολλά καλά themes και αλλά μπορεί να ενσωματώσει σε μεγάλο βαθμό σε οποιοδήποτε θέμα.

Για την εγκατάσταση αρκεί η σύνδεση στην ιστοσελίδα που είναι εγκαταστημένο το WordPress. Πηγαίνοντας στο Plugins -> Προσθήκη νέου προστίθεται το "WooCommerce" και ξεκινάει η εγκατάσταση.

Το WooCommerce είναι ένα δωρεάν πρόσθετο και μπορεί να επιλεγεί ανάλογα με τη φύση της επιχείρησής ή καταστήματος. Συνήθως ο επιχειρηματίας χρειάζεται περίπου 40 έως 200 € για επεκτάσεις ώστε να γίνει λειτουργικό.

.

### Πλεονεκτήματα WooCommerce

Με το σύστημα επέκτασης που έχει μπορεί να χρησιμοποιήσει πολλά στοιχεία του WordPress και επιτρέπει οποιονδήποτε τρόπο πληρωμών θέλετε με χρήση επεκτάσεων.

Υποστηρίζει απεριόριστο αριθμό προϊόντων.

### Μειονεκτήματα WooCommerce

Σε αντίθεση με άλλα συστήματα ηλεκτρονικού καταστήματος που έχουν επεκτάσεις σχεδιασμένες ειδικά για το περιβάλλον τους στο Woocommerce μπορεί μια προσθήκη να καταστρέψει το σύστημα.

Ενώ υπάρχουν πολλά πρόσθετα στο Woocommerce που είναι διαθέσιμα να εισαχθούν προς και από το WordPress επειδή δεν είναι επίσημα μπορεί να δημιουργήσουν πάρα πολλά προβλήματα κυρίως σε θέματα ασφάλειας.

Έτσι θα πρέπει πληρώσει κάποιος κάποιο σημαντικό ποσό για ένα πολύ προσαρμοσμένο θέμα με ασφαλείς πρόσθετες λειτουργίες.

Τέλος, για να δημιουργήσει κάποιος το ηλεκτρονικό κατάστημα Woocommerce πρέπει να εγκαταστήσει πρώτα το WordPress, δηλαδή να προχωρήσει σε δύο κύριες εγκαταστάσεις.

### **2.1.3 Άλλα ηλεκτρονικά καταστήματα**

#### **Magento**

Ένα από τις δημοφιλείς πλατφόρμες ηλεκτρονικού εμπορίου ανοιχτού κώδικα είναι και το Magento[5]. Παρέχει το υψηλότερο επίπεδο ελέγχου και ευελιξίας.

#### **Πλεονεκτήματα Magento**

Παρέχει το δικό του θέμα και διάταξη.

Περιλαμβάνει αποτελεσματικά API [6] που μπορούν να συνδεθούν με οποιαδήποτε λύση τρίτου μέρους.

Έχει πάνω από 1600 πρόσθετα και επεκτάσεις και προσφέρει πολλαπλές επιλογές πληρωμής και αποστολής.

Έχει τακτικές ενημερώσεις με κύριο στόχο την ισχυρή ασφάλεια.

Παρέχει δωρεάν έκδοση επί πληρωμή.

#### **Μειονεκτήματα Magento**

Θα χρειαστεί η βοήθεια ενός προγραμματιστή ή ειδικού.

Είναι περίπλοκο και μεγάλο πακέτο για απλά καταστήματα.

Η διαθέσιμη τεκμηρίωση δεν είναι πολύ καλή.

#### **PrestaShop**

Μια νέα πλατφόρμα ηλεκτρονικού εμπορίου ανοιχτού κώδικα είναι και το PrestaShop[7]. Είναι εύκολο στη χρήση και δεν απαιτεί να γνωρίζει κάποιος προγραμματισμό ιστοσελίδων. Η διεπαφή που παρέχει επιτρέπει τον αποτελεσματικό χειρισμό αναφορών, τη διαχείριση προϊόντων, τη διαχείριση καταστημάτων κλπ.

## Πλεονεκτήματα PrestaShop

Προσφέρει πάνω από 26000 προσθήκες και επεκτάσεις.

Υποστηρίζει όλα τα είδη προϊόντων και βοηθά στη διαχείριση ενός δυναμικού καταλόγου.

Έχει πολλά εργαλεία για μάρκετινγκ και προώθησης με αναλυτικά στοιχεία και αναφορές.

## Μειονεκτήματα PrestaShop

Οι πρόσθετες ενότητες είναι ακριβότερες από τις υπόλοιπες πλατφόρμες.

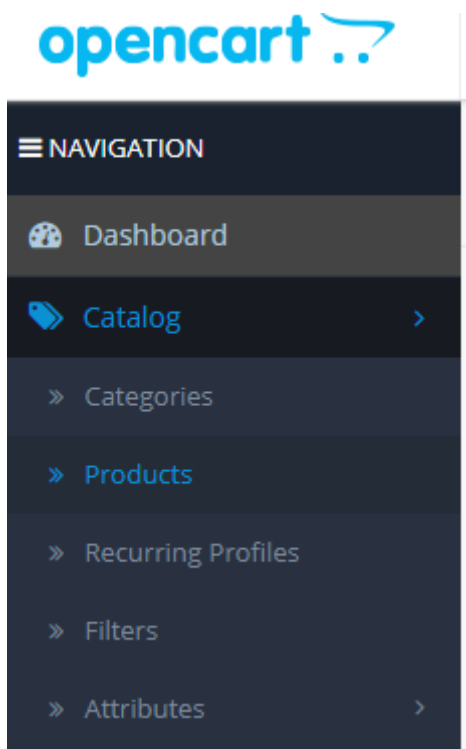
Δεν διαθέτει ενεργή ομάδα υποστήριξης.

Δεν διαθέτει δυνατότητες πολλαπλών καταστημάτων

## 2.2 Τρόποι διαχείρισης εισαγωγής προϊόντων

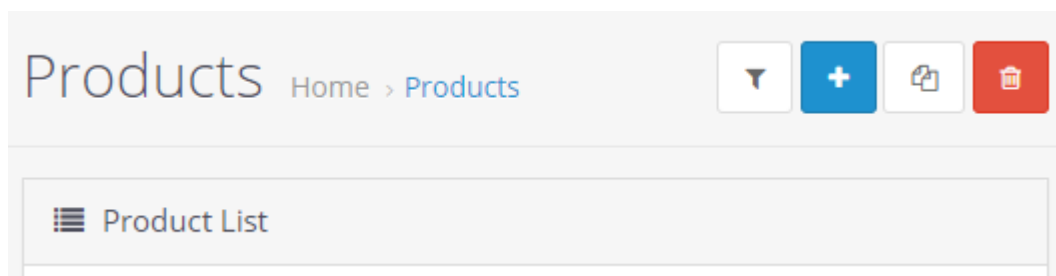
### 2.2.1 Opencart και τρόποι εισαγωγής προϊόντων

Για την εισαγωγή προϊόντος στο opencart ακολουθούμε της εξής διαδικασία.



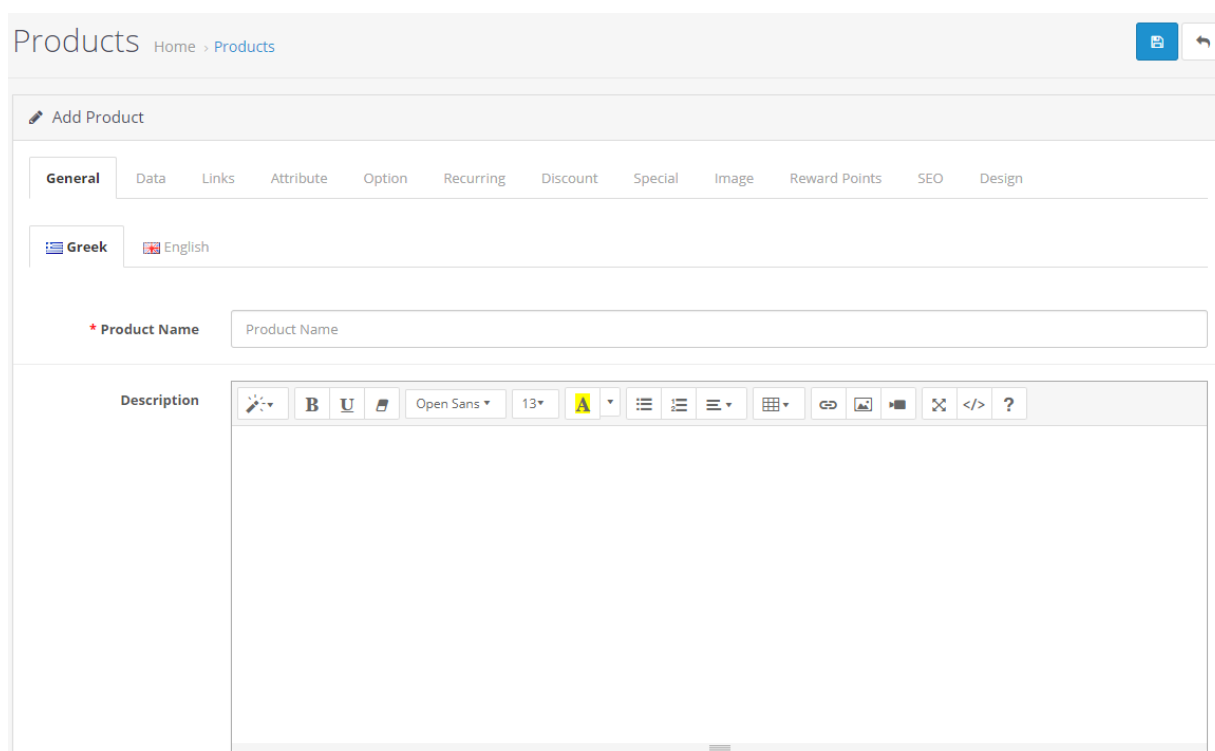
Εικόνα 2.1: Μενού για εισαγωγή προϊόντος

Στο Εικόνα 2.1 παρουσιάζεται το μενού του διαχειριστικού του ηλεκτρονικού καταστήματος όπου ο χρήστης πρέπει να πατήσει την επιλογή products για να δει τα προϊόντα του καταστήματος



Εικόνα 2.2: OpenCart – Καρτέλα προϊόντων

Και θα εμφανιστεί η σελίδα που φαίνεται στην Εικόνα 2.3. Στη συγκεκριμένη σελίδα υπάρχει ένα κουμπί όπου ο χρήστης μπορεί να το πατήσει για να κάνει προσθήκη νέου προϊόντος.




Εικόνα 2.3: Προσθήκη προϊόντος

Στην σελίδα του νέου προϊόντος εμφανίζονται πολλές επιλογές-καρτέλες οι οποίες φαίνονται στο Εικόνα 2.4 όπου στην πρώτη καρτέλα ο χρήστης πρέπει να εισάγει το όνομα του προϊόντος, την περιγραφή του προϊόντος και το Meta tag τίτλο του. Τα παραπάνω θα πρέπει να τα εισάγει στα ελληνικά και στα αγγλικά ή στις γλώσσες στις οποίες θέλει να υποστηρίξει.

<b>* Meta Tag Title</b>	Meta Tag Title
<b>Meta Tag Description</b>	Meta Tag Description
<b>Meta Tag Keywords</b>	Meta Tag Keywords
<b>Product Tags</b> ?	Product Tags




Εικόνα 2.4: Προϊόν – Γενικά στοιχεία

 Add Product

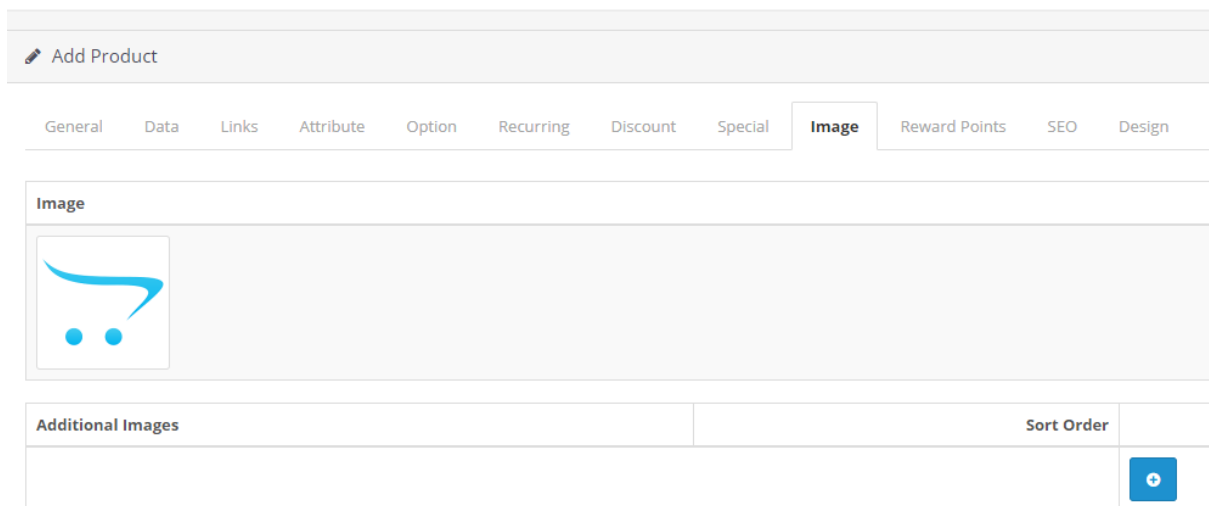
General	<b>Data</b>	Links	Attribute	Option	Recurring	Discount	Special	Image	Reward Points	SEO	Design
<b>* Model</b>	Model										
<b>SKU</b> ?	SKU										
<b>UPC</b> ?	UPC										
<b>EAN</b> ?	EAN										
<b>JAN</b> ?	JAN										
<b>ISBN</b> ?	ISBN										
<b>MPN</b> ?	MPN										
<b>Location</b>	Location										
<b>Price</b>	Price										

Εικόνα 2.5: Προϊόν – Δεδομένα

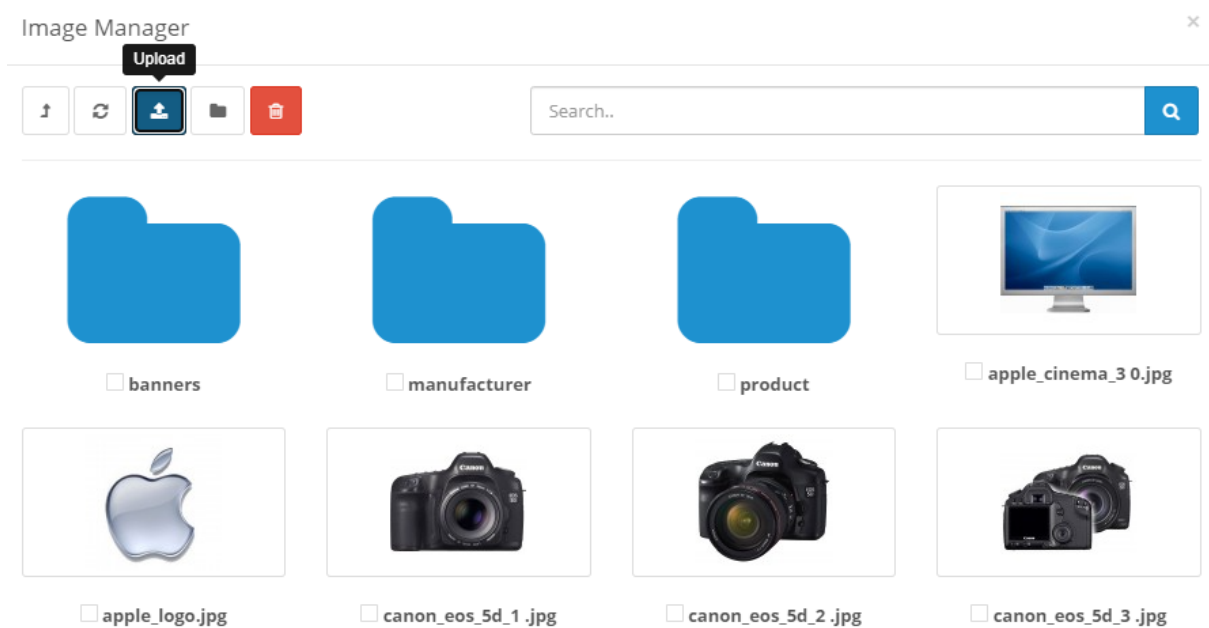
Στη δεύτερη καρτέλα εμφανίζονται τα πεδία για διάφορα χαρακτηριστικά του προϊόντος όπως είναι το όνομα για το μοντέλο ή αυτό που συνήθως λέμε product id. Σε αυτήν εδώ την καρτέλα υπάρχει το πεδίο για την τιμή του προϊόντος, για την φορολογική κλάση που θα χρησιμοποιηθεί, για την ποσότητα των προϊόντων στο κατάστημα, η ημερομηνία όπου θα είναι διαθέσιμο, εάν απαιτεί αποστολή, τις διαστάσεις και το βάρος του προϊόντος και τέλος η επιλογή Status που δηλώνει αν αυτό το προϊόν είναι ενεργό ή όχι για να εμφανιστεί στον ηλεκτρονικό κατάστημα.

Tax Class	--- None ---		
Quantity	1		
Minimum Quantity 	1		
Subtract Stock	Yes		
Out Of Stock Status 	2-3 Days		
Requires Shipping	<input checked="" type="radio"/> Yes <input type="radio"/> No		
Date Available	2021-03-27		
Dimensions (L x W x H)	Length	Width	Height
Length Class	Centimeter		
Weight	Weight		
Weight Class	Kilogram		
Status	Enabled		

Εικόνα 2.6: Προϊόν – Δεδομένα (το υπόλοιπο μέρος)



Εικόνα 2.7: Προϊόν - Εικόνες



Εικόνα 2.8: Προϊόν – Προσθήκη εικόνων

Μία από αυτές τις καρτέλες είναι και η καρτέλα που αφορά την εισαγωγή εικόνων για το προϊόν. Όπως φαίνεται στην Εικόνα 2.7 για ένα νέο προϊόν ή για ένα υπάρχον ο χρήστης θα πρέπει να πατήσει το κουμπί για προσθήκη εικόνων ώστε να του εμφανιστεί ο διαχειριστής εικόνων όπως φαίνεται στην Εικόνα 2.8 όπου θα πρέπει να επιλέξει μια εικόνα από το σκληρό δίσκο του υπολογιστή για να το φορτώσει σαν εικόνα του προϊόντος. Αυτή μπορεί να είναι πρώτη ή δεύτερη για το προϊόν.

Στην Εικόνα 2.3, στην Εικόνα 2.4 και στην Εικόνα 2.5 φαίνονται με κόκκινο αστερίσκο τα υποχρεωτικά πεδία τα οποία πρέπει να εισάγει ο χρήστης κατά την προσθήκη ενός νέου προϊόντος. Η εικόνα δεν

είναι υποχρεωτική κατά την εισαγωγή νέου προϊόντος αλλά αν δεν τοποθετηθεί τότε θα πάρει την προεπιλεγμένη όπως επίσης δεν είναι υποχρεωτική η εισαγωγή της τιμής-κόστους, της ποσότητας και της φορολογικής κλάσης για το κάθε προϊόν.

Επομένως, για την προσθήκη νέου προϊόντος ο χρήστης θα πρέπει να ακολουθήσει την παραπάνω διαδικασία και να εισάγει το όνομα του προϊόντος και το Mega τίτλο στις γλώσσες στις οποίες θέλει να υποστηρίξει. Στη συνέχεια να εισάγει το μοντέλο ή αλλιώς product id -τον κωδικό του προϊόντος και την τιμή και ίσως και την ποσότητα. Στο τέλος πρέπει να εισάγει φωτογραφίες για το προϊόν.

Όλη αυτή η διαδικασία πρέπει να πραγματοποιηθεί από τον υπολογιστή του διαχειριστή, οι εικόνες του κάθε προϊόντος ή καλύτερα του νέου προϊόντος θα πρέπει να βρίσκονται ήδη στον υπολογιστή και ο χρήστης διαχειριστής κατά την εισαγωγή ενός νέου προϊόντος θα πρέπει να τις αναζητεί.

Ας πάρουμε υπόψη ότι σε ένα ηλεκτρονικό κατάστημα όπου υπάρχουν πάρα πολλά προϊόντα ο ιδιοκτήτης θα πρέπει να τα τοποθετήσει σε αυτό. Έστω ότι τα προϊόντα αυτά είναι τέτοιου είδους όπου δεν υπάρχουν αντίστοιχες φωτογραφίες στο διαδίκτυο ή είναι προϊόντα που θα πρέπει να τα φωτογραφήσει. Συνήθως τέτοια προϊόντα είναι συνήθως είδη ενδυμασίας, υπόδησης ή χειροποίητα προϊόντα.

Ο ιδιοκτήτης θα πρέπει να εισάγει από στο κατάστημα πολλά προϊόντα και αυτό θα πρέπει να το κάνει γρήγορα χωρίς να ακολουθεί τη διαδικασία φωτογράφιση, μεταφορά στο σκληρό δίσκο και αργότερα δημιουργία προϊόντος και ανέβασμα φωτογραφίας από τον υπολογιστή.

Η υπάρχουσα εφαρμογή - σύστημα έρχεται να μειώσει τον χρόνο εισαγωγής ενός νέου προϊόντος στο ηλεκτρονικό κατάστημα και να απλοποιήσει την όλη διαδικασία για τον ιδιοκτήτη. Μπορεί με το κινητό του να φωτογραφίζει το προϊόν, να του βάλει απλά ένα τίτλο, μία τιμή και με την φωτογραφία που έχει ήδη τραβήξει να την αποστέλλει κατευθείαν στο ηλεκτρονικό κατάστημα.

### **2.2.2 Opencart API και διαχείριση προϊόντων από το API**

Υπάρχουν πρόσθετα στο Opencart που μπορούν να υλοποιήσουν διαχείριση προϊόντος μέσω του API δεν είναι δωρεάν.

Το πρώτο από αυτό είναι το

Opencart REST API - Shopping cart API (Εικόνα 2.9) [8]



Εικόνα 2.9: Opencart REST API - Shopping cart API 1

Το πρόσθετο αυτό κοστίζει 99\$, έχει πάρα πολλές επιλογές διαχείρισης προϊόντων, κατηγοριών αλλά δεν διαθέτει εισαγωγή προϊόντος μέσω API. Επίσης, δεν παρέχει εφαρμογή στο κινητό.

Το δεύτερο πρόσθετο είναι το Opencart Rest API. (Εικόνα 2.10) [9]

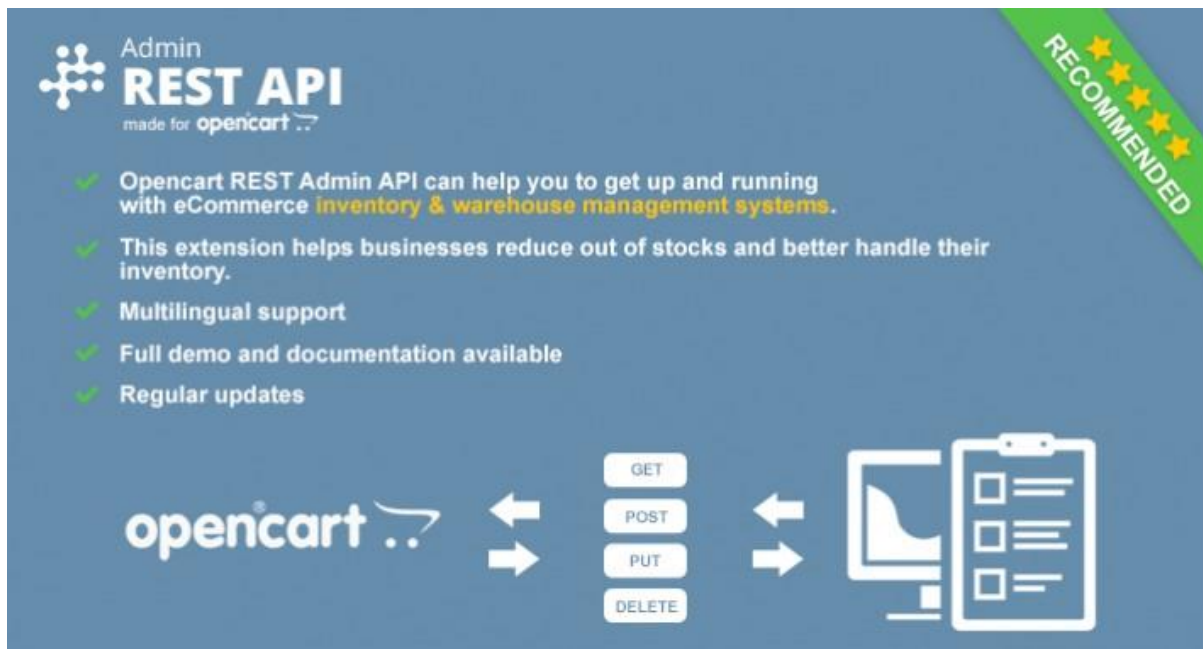


Εικόνα 2.10: Opencart REST API - Shopping cart API 2

Το πρόσθετο αυτό κοστίζει 60\$, έχει πάρα πολλές επιλογές διαχείρισης προϊόντων, κατηγοριών αλλά δεν διαθέτει εισαγωγή προϊόντος μέσω API. Επίσης, δεν παρέχει εφαρμογή στο κινητό.

Υπάρχει ένα πρόσθετο (Εικόνα 2.11) [10]

## Opencart REST API - Shopping cart API



Εικόνα 2.11: Opencart REST API - Shopping cart API 3

το οποίο κοστίζει 129\$, έχει πάρα πολλές επιλογές διαχείρισης προϊόντων, κατηγοριών κτλ μέσω API.

Μερικές από αυτές είναι:

- List / create / edit / delete categories
- List / create / edit / delete products
- List / create / edit / delete orders
- List / create / edit / delete customers
- List / create / delete manufacturers
- upload images

Δεν παρέχει εφαρμογή στο κινητό. Δεν είναι σύστημα που ο χρήστης να μπορεί να προσθέσει ένα προϊόν γρήγορα και εύκολα. Αλλά αυτό όπως και τα υπόλοιπα παρέχουν το σύνολο των μεθόδων για να καλέσει ο προγραμματιστής – χρήστης για να υλοποιήσει αυτό που θέλει.

## **Κεφάλαιο 3ο: Συστήματα διαχείρισης και λογισμικά ανάπτυξης**

### **3.1 Εισαγωγή**

Το ηλεκτρονικό εμπόριο, επίσης γνωστό ως ηλεκτρονικό εμπόριο ή διαδικτυακό εμπόριο, αναφέρεται στην αγορά και πώληση αγαθών ή υπηρεσιών μέσω του Διαδικτύου και στη μεταφορά χρημάτων και δεδομένων για την εκτέλεση αυτών των συναλλαγών. Το ηλεκτρονικό εμπόριο χρησιμοποιείται συχνά για αναφορά στην πώληση φυσικών προϊόντων στο διαδίκτυο, αλλά μπορεί επίσης να περιγράψει οποιοδήποτε είδος εμπορικής συναλλαγής που διευκολύνεται μέσω του Διαδικτύου.

Ενώ το ηλεκτρονικό εμπόριο αναφέρεται σε όλες τις πτυχές της λειτουργίας μιας διαδικτυακής επιχείρησης, το ηλεκτρονικό εμπόριο αναφέρεται συγκεκριμένα στη συναλλαγή αγαθών και υπηρεσιών.

Το ηλεκτρονικό εμπόριο, που αναφέρεται επίσης ως ηλεκτρονικό εμπόριο ή διαδικτυακό εμπόριο, αφορά τη μεταφορά χρημάτων και δεδομένων για την εκτέλεση συναλλαγών για ένα ηλεκτρονικό κατάστημα. Είναι οι συνδυασμένες καινοτόμες τεχνολογίες πληροφοριών και επικοινωνιών, που χρησιμοποιούν τα ηλεκτρονικά καταστήματα για την υποστήριξη και την ολοκλήρωση λειτουργικών και στρατηγικών εργασιών για τις πωλήσεις τους, χωρίς τη χρήση εγγράφων. Το ηλεκτρονικό εμπόριο θεωρείται ευρέως η πώληση φυσικών και εικονικών προϊόντων στο διαδίκτυο, αλλά μπορεί επίσης να περιγράψει κάθε είδους εμπορική συναλλαγή μέσω ηλεκτρονικών μέτρων.

Ένα ηλεκτρονικό κατάστημα ή ηλεκτρονικό κατάστημα είναι το «μέρος» όπου μια εταιρεία μπορεί να πουλήσει τα προϊόντα ή τις υπηρεσίες της χρησιμοποιώντας ψηφιακά δίκτυα. Είναι ένα διαδικτυακό κατάστημα όπου οι πελάτες μπορούν να «επισκέπτονται» από οπουδήποτε και ανά πάσα στιγμή χρησιμοποιώντας ένα πρόγραμμα περιήγησης Web σε κινητό τηλέφωνο ή φορητό υπολογιστή.

### **3.2 Opencart**

Σε μια εποχή πλατφόρμων ηλεκτρονικού εμπορίου που βασίζονται σε συνδρομές και δημιουργίας νέων ιστότοπων, μερικές φορές είναι ωραίο να έχουμε μια δωρεάν λύση. Μια επιλογή ανοιχτού κώδικα είναι το OpenCart. Είναι ανοιχτό για δωρεάν λήψη, μπορεί κάποιος να βρει τη δική του φιλοξενία-server και να κάνει αναβάθμιση χωρίς καμία πληρωμή. Ο λόγος που αρέσει τόσο πολύ είναι επειδή πολλές πλατφόρμες ανοιχτού κώδικα προσπαθούν να προσφέρουν ετήσιες αναβαθμίσεις.

Από την άλλη πλευρά, υπάρχουν έτοιμες λύσεις όπως το Shopify όπου υπάρχει μηνιαία χρέωση χωρίς πραγματική ιδιοκτησία του ιστότοπού. Λοιπόν, το OpenCart απέχει από το τέλειο, αλλά έχει ένα ωραίο δημιουργό διαδικτυακών καταστημάτων και αν κάποιος θέλει πραγματικά να διατηρήσει το κόστος χαμηλό μπορεί να ενισχύσει τις επιλογές προσαρμογής του.

Το OpenCart είναι εντελώς δωρεάν για λήψη δεν απαιτούνται δεξιότητες ανάπτυξης. Λειτουργεί με παρόμοιο τρόπο με το WordPress, όπου λαμβάνεται το βασικό προϊόν OpenCart και στη συνέχεια, αναπτύσσεται συνήθως με τη βοήθεια επεκτάσεων. Πολλοί άνθρωποι απολαμβάνουν αυτό το μοντέλο, επειδή μπορούν να διαλέξουν και να επιλέξουν ποιες επεκτάσεις χρειάζονται. Ωστόσο, κάποιος μπορεί να καταλήξει να δαπανήσει ένα αξιοπρεπές χρηματικό ποσό ανάλογα με τον αριθμό των επεκτάσεων που χρειάζεται.

Τα κύρια χαρακτηριστικά του είναι:

Δωρεάν λήψη

Εύκολο στη χρήση UI (ιδανικό για εξερεύνηση)

Εξαιρετικά προσαρμόσιμο περιβάλλον χάρη στην κατασκευή ανοιχτού κώδικα

Πολλές ενσωματώσεις με κορυφαία εργαλεία

Χωρίς συνεχές κόστος ανάπτυξης (εάν έχετε τις σωστές δεξιότητες)

Πολύγλωσσο με υποστήριξη πολλαπλών νομισμάτων

Πολλά θέματα

Marketplace γεμάτο με προσθήκες, πύλες και άλλα εργαλεία

Εξαιρετική κοινότητα και υποστήριξη πελατών

Απεριόριστα προϊόντα, κατηγορίες και κατασκευαστές

Αξιολογήσεις προϊόντων και συστήματα αναθεώρησης

Το OpenCart προσφέρει πολλές επεκτάσεις. Επειδή πολλοί προγραμματιστές εργάζονται για τη δημιουργία αυτών των επεκτάσεων, υπάρχει ένα ευρύ φάσμα πρόσθετων για ενσωμάτωση στο κατάστημά. Η αγορά είναι γεμάτη με χιλιάδες θέματα και επεκτάσεις για πράγματα όπως ροές προϊόντων, αναφορές, τρόπους αποστολής και πύλες πληρωμής. Το μεγαλύτερο μέρος της αγοράς είναι ότι η πλειοψηφία των επεκτάσεων γίνεται για το ηλεκτρονικό εμπόριο, σε αντίθεση με μια πλατφόρμα όπως το WordPress όπου υπάρχουν επίσης προσθήκες για κανονικούς ιστότοπους.

Συνδυάστε τις επεκτάσεις με την ισχυρή πρόσβαση κώδικα και έχετε ένα συμπαγές πακέτο για την προσαρμογή του ιστότοπού σας. Εν ολίγοις, οι προγραμματιστές θα βρουν το OpenCart απίστευτο να συνεργαστεί, βλέποντας πώς μπορούν να κατασκευάσουν οποιοδήποτε τύπο ιστότοπου θέλουν χωρίς πολλούς περιορισμούς. Εκτός από αυτό, οι επιχειρηματίες μπορούν να αναβαθμίσουν τις επιχειρήσεις τους γρηγορότερα, σε αντίθεση με τη διατήρηση μιας πλατφόρμας με μόνο ορισμένα θέματα, πρόσθετα και επιλογές κωδικοποίησης.

Ο πίνακας ελέγχου είναι αρκετά απλός για να τον κατανοήσουν οι περισσότεροι χρήστες. Σίγουρα βοηθά να έχει κάποιος γνώσεις ανάπτυξης κατά τη δημιουργία ενός ιστότοπου με το OpenCart, αλλά τουλάχιστον οποιοσδήποτε μπορεί να κινηθεί γύρω από το ταμπλό και να μάθει για τις δυνατότητές του χωρίς πολύ εκπαίδευση.

Τέλος, το καλύτερο μέρος του OpenCart είναι η τιμή. Όπως αναφέρθηκε προηγουμένως, είναι μια εντελώς δωρεάν πλατφόρμα ηλεκτρονικού εμπορίου ανοιχτού κώδικα, με κουμπί λήψης απευθείας στον ιστότοπο. Δεν ζητάει από αυτόν που το εγκαθιστά να υποβάλει τη διεύθυνση email ή οτιδήποτε άλλο. Επιπλέον, το OpenCart δεν απαιτεί να δαπανάει ο ιδιοκτήτης μηνιαίες ή ετήσιες χρεώσεις για αναβαθμίσεις ή συντήρηση. Οι πληρωμένες επεκτάσεις είναι κάτι διαφορετικό οι οποίες πολλές φορές είναι προτιμότερες για λόγους ασφάλειας.

### 3.2.1 Εγκατάσταση opencart

Για να το εγκαταστήσει κάποιος πρέπει πρώτα να το κατεβάσει από τη σελίδα λήψης του OpenCart και να πατήσει το κουμπί Free Download.

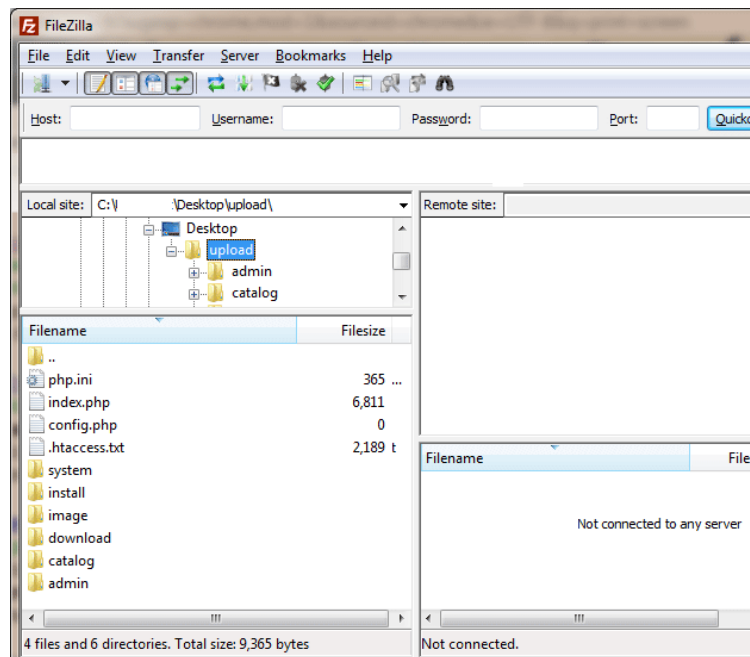
Εικόνα 3.1: Λήψη OpenCart

Αφού κατέβει το αρχείο zip αποσυμπιέζουμε. Μπορούμε να κάνουμε:

Εγκατάσταση στον τοπικό υπολογιστή

Εγκατάσταση σε server με μεταφορά αρχείων με FTP ή SSH.

Μέσω FTP με το FileZilla [11] συνδεόμαστε στον server και μεταφέρουμε τα αρχεία.



Εικόνα 3.2: Το πρόγραμμα FileZilla

Στη συνέχεια δημιουργούμε τη βάση δεδομένων για το Opencart.

Στη βάση το OpenCart θα αποθηκεύσει όλο το περιεχόμενο, τα ονόματα χρήστη, τους κωδικούς πρόσβασης και ότι άλλο χρειάζεται το ηλεκτρονικό κατάστημα. Αποτελεί τον πυρήνα του διαδικτυακού σας καταστήματος και το OpenCart είναι το εργαλείο που έχει πρόσβαση και εμφανίζει τα αποτελέσματα.

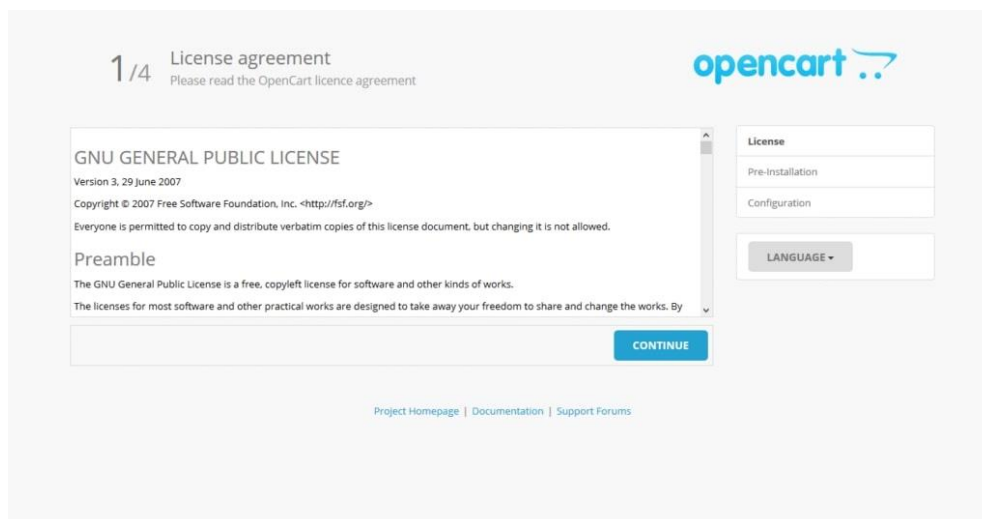
Μέσω του phpMyAdmin [12] μπορεί κάποιος να συνδεθεί με τη βάση, όπως φαίνεται στην παρακάτω εικόνα.

<input type="checkbox"/>	oc_order_option	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	0	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_order_product	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	0	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_order_recurring	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	0	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_order_recurring_transaction	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	0	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_order_shipment	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	0	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_order_status	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	28	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_order_total	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	0	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_order_voucher	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	0	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_product	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	55	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_product_attribute	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	10	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_product_description	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	98	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_product_discount	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	3	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_product_filter	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	0	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_product_image	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	61	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_product_option	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	11	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_product_option_value	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	13	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_product_recurring	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	0	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_product_related	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	4	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_product_reward	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	17	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_product_special	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	3	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_product_to_category	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	97	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_product_to_download	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	0	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_product_to_layout	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	0	MyISAM	utf8_general_ci
<input type="checkbox"/>	oc_product_to_store	★	Περιήγηση	Δομή	Αναζήτηση	Προσθήκη	Αδειασμα	Διαγραφή	52	MyISAM	utf8_general_ci

Εικόνα 3.3: Το πρόγραμμα phpMyAdmin

Αφού ολοκληρωθεί η εγκατάσταση της βάσης συνεχίσουμε στην εγκατάσταση του OpenCart.

Μεταβαίνουμε στη διεύθυνση URL που εντοπίζει τον κατάλογο όπου ανέβηκαν τα αρχεία. Στη συνέχεια ο χρήστης συνεχίζει την εγκατάσταση μέσω των βημάτων που φαίνεται στην Εικόνα 3.4:



2/4 Pre-Installation  
Check your server is set-up correctly

opencart

1. Please configure your PHP settings to match requirements listed below.

PHP Settings	Current Settings	Required Settings	Status
PHP Version	7.1.1	5.4+	✔
Register Globals	Off	Off	✔
Magic Quotes GPC	Off	Off	✔
File Uploads	On	On	✔
Session Auto Start	Off	Off	✔

2. Please make sure the PHP extensions listed below are installed.

Extension Settings	Current Settings	Required Settings	Status
Database	On	On	✔
GD	On	On	✔
cURL	On	On	✔
OpenSSL	On	On	✔

License

Pre-Installation

Configuration

LANGUAGE ▾

3/4 Configuration  
Enter your database and administration details

opencart

1. Please enter your database connection details.

DB Driver: MySQL

\* Hostname: localhost

\* Username: root

Password:

\* Database:

\* Port: 3306

Prefix: oc\_

License

Pre-Installation

Configuration

LANGUAGE ▾

2. Please enter a username and password for the administration.

\* Username: admin

\* Password:

\* E-Mail:

BACK CONTINUE

[Project Homepage](#) | [Documentation](#) | [Support Forums](#)

Εικόνα 3.4: Εγκατάσταση του OpenCart – Τα βήματα

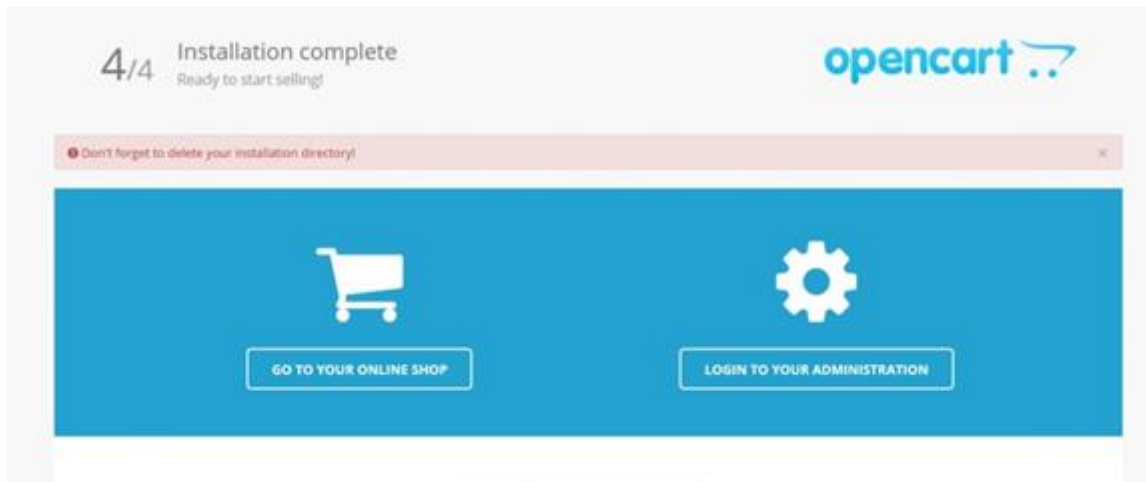
Το OpenCart εκτελεί έναν έλεγχο για να δει εάν το περιβάλλον είναι συμβατό για να εγκατασταθεί.

Στη συνέχεια θα προκύψει μια προειδοποίηση με δύο αρχεία που ονομάζονται config.php.

Αυτό που πρέπει να γίνει είναι μετονομασία των αρχείων config-dist.php και config.php.

Στη συνέχεια εισάγουμε το όνομα του κεντρικού υπολογιστή, το όνομα χρήστη, τον κωδικό πρόσβασης, το όνομα της βάσης δεδομένων στα πεδία που θα ζητηθούν.

Αν προκύψει η παρακάτω οθόνη τότε έχουμε εγκαταστήσει με επιτυχία το OpenCart.



Εικόνα 3.5: Εγκατάσταση OpenCart – Βήμα 4

Φαίνεται στην κόκκινη γραμμή ένα μήνυμα που δηλώνει ότι πρέπει να καταργήσουμε τον κατάλογο εγκατάστασης.





Σε αυτήν σελίδα της εγκατάστασης μπορούμε να δούμε τα δύο κουμπιά που μας μεταφέρουν στο ηλεκτρονικό κατάστημα ή στη σελίδα διαχείρισης.

Εάν πατήσουμε στο πρώτο θα μεταβούμε στο ηλεκτρονικό κατάστημα και στην εικόνα 3.6 παρουσιάζεται ένα ηλεκτρονικό κατάστημα με κοσμήματα που έχει τοποθετηθεί ειδικό theme.

Με το παρουσιαστικό αυτό τονίζεται η ανάγκη της εφαρμογής που κατασκευάστηκε για την εξυπηρέτηση γρήγορης προσθήκης προϊόντων.

## Latest



 <p>Infinite Necklace \$241.99 <a href="#">Add To Cart</a></p>	 <p>Angel City Ring \$122.00 <a href="#">Add To Cart</a></p>	 <p>High Five Ring \$122.00 <a href="#">Add To Cart</a></p>	 <p>Eva Clip Earrings \$1,202.00 <a href="#">Add To Cart</a></p>
---	---	---	---


## Categories

- Clip Earrings (15) +
- Necklace (5) +
- Pin (1)
- Ring (16) +
- Watch (0)
- Pierced Earrings (5)

## Information


- About Us
- Delivery Information
- Privacy Policy
- Terms & Conditions
- Contact Us
- Site Map

## Specials

 <p>Curtain Call Necklace <del>\$422.00</del> \$98.00</p>
--







## About Jewellery Store

Lorem ipsum presta shop amet Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Lorem ipsum dolor sit amet, consectetur adipiscing elit



Latest  
Step out in style  
[Shop Now](#)

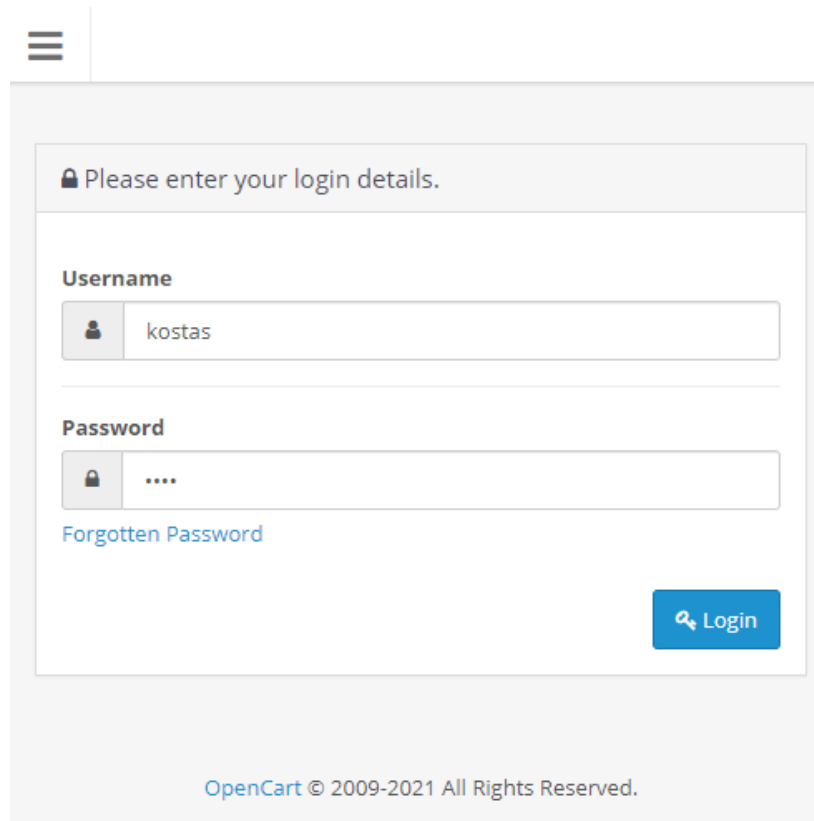
## Featured

 <p>Mojito Ring \$602.00 <a href="#">Add To Cart</a></p>	 <p>Infinite Ring \$123.20 <a href="#">Add To Cart</a></p>	 <p>Le Louvre Necklace <del>\$422.00</del> \$110.00 <a href="#">Add To Cart</a></p>
		

Εικόνα 3.6: Το OpenCart όταν γίνεται η εγκατάσταση (theme Jewellery)[13]

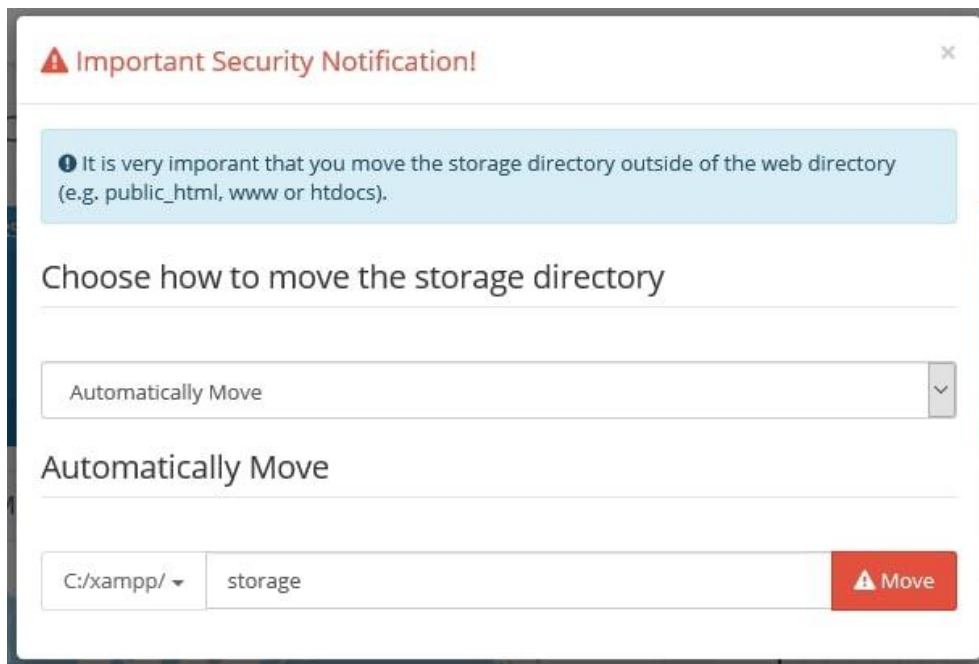
Για να μπει κάποιος στη σελίδα διαχείρισης μπορεί να μεταβεί στην

<http://localhost/admin/>



Εικόνα 3.7: Στοιχεία πρόσβασης στο login - admin

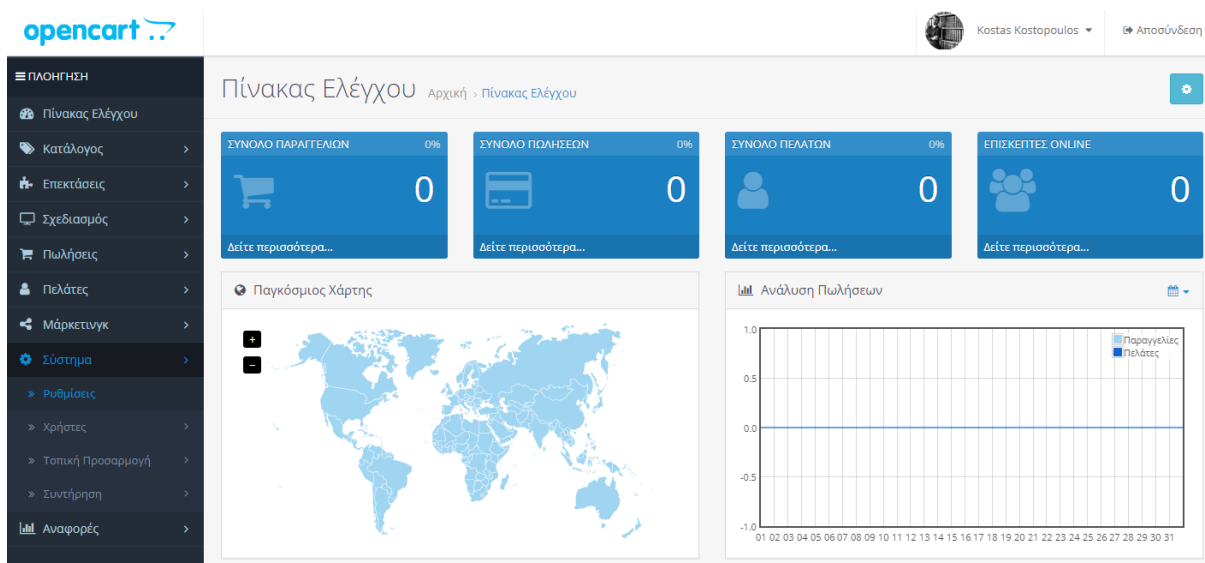
Αφού δώσουμε σωστά τα στοιχεία τότε θα παρουσιαστεί η παρακάτω εικόνα.



Εικόνα 3.8: Μετακίνηση storage φακέλου

Θα πρέπει να γίνει μετακίνηση του φακέλου storage σε μονοπάτι εκτός του φακέλου εγκατάσταση για λόγους ασφάλειας και μπορούμε να το κάνουμε με αυτόματο ή όχι τρόπο.

Η σελίδα διαχείρισης παρουσιάζεται στην Εικόνα 3.9.



Εικόνα 3.9: Το διαχειριστικό OpenCart

Στην αριστερή πλευρά βρίσκεται το μενού με τα κύρια μέρη του διαδικτυακού καταστήματος:

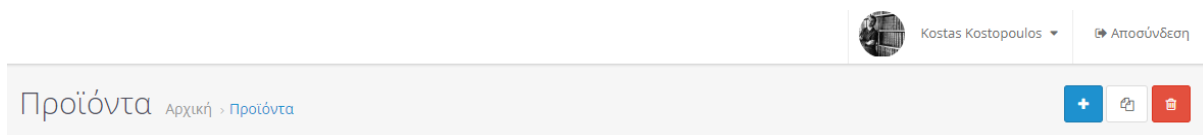
- Πίνακας ελέγχου
- Κατάλογος – Διαχείριση δημιουργία κατηγοριών προϊόντων .
- Επεκτάσεις - Για την εγκατάσταση επεκτάσεων που παρέχουν περισσότερη λειτουργικότητα στο κατάστημα.
- Σχεδιασμός – Για ρύθμιση διατάξεων σελίδας γλωσσών κτλ.
- Πωλήσεις – Διαχείριση πωλήσεων, επιστροφών κτλ.
- Πελάτες - Ότι αφορά τους πελάτες.
- Μάρκετινγκ - Εργαλεία μάρκετινγκ, κουπόνια και άλλα.
- Σύστημα - Ρυθμίσεις, διαχείριση χρηστών κτλ.
- Αναφορές – Για την κατάσταση του καταστήματος.

Πηγαίνοντας στην ενότητα Κατάλογος-> Προϊόντα βλέπουμε τη λίστα των προϊόντων, όπως φαίνεται στην Εικόνα 3.10 και μπορούμε να προσθέσουμε ένα νέο.

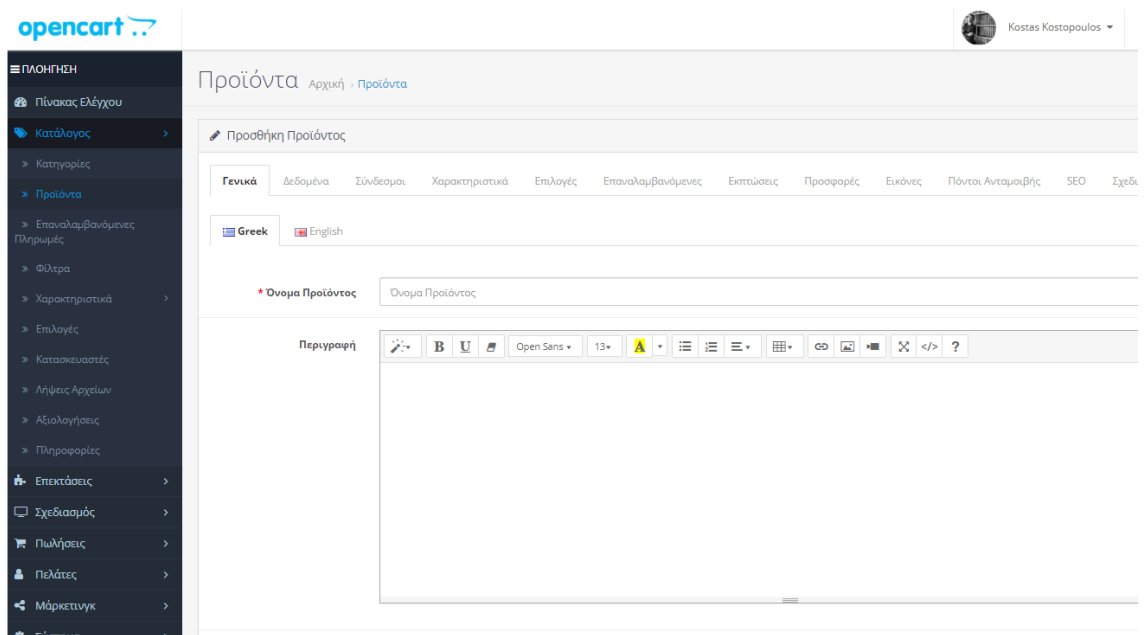
<input type="checkbox"/>		Μπλούζα Μακό 0535	Μπλούζα Μακό 0535	14,00€	<span style="background-color: red; color: white; padding: 2px;">1</span>	Ενεργοποιημένο	
<input type="checkbox"/>		Μπλούζα Μακό 0555	Μπλούζα Μακό 0555	24,00€	<span style="background-color: green; color: white; padding: 2px;">13</span>	Ενεργοποιημένο	
<input type="checkbox"/>		Μπλούζα Μακό 0840	Μπλούζα Μακό 0840	26,00€	<span style="background-color: green; color: white; padding: 2px;">17</span>	Ενεργοποιημένο	
<input type="checkbox"/>		Μπλούζα Μακό 0844	Μπλούζα Μακό 0844	26,00€	<span style="background-color: green; color: white; padding: 2px;">17</span>	Ενεργοποιημένο	
<input type="checkbox"/>		Μπλούζα Μακό 1235	Μπλούζα Μακό 1235	22,00€	<span style="background-color: red; color: white; padding: 2px;">1</span>	Απενεργοποιημένο	

Εικόνα 3.10: Τα προϊόντα του ηλεκτρονικού καταστήματος

Αν πατήσουμε το μπλε κουμπί στην Εικόνα 3.11 θα μεταβούμε στη σελίδα Προσθήκης νέου προϊόντος (Εικόνα 3.12) ή να διαγράψουμε κάποιο πατώντας τον κάδο.



Εικόνα 3.11: Προϊόντα – Προσθήκη ή διαγραφή



Εικόνα 3.12: Προσθήκη νέου προϊόντος

### 3.3 Flutter – Android Studio

Το Flutter[14-15] είναι ένα κιτ ανάπτυξης λογισμικού ανοιχτού κώδικα που δημιουργήθηκε από την Google. Χρησιμοποιείται για την ανάπτυξη εφαρμογών για Android, iOS, Linux, Mac, Windows και τον Ιστό από έναν βασικό κώδικα.

Το Flutter θεωρείται ένα από τα πιο χρήσιμα, δωρεάν, ευέλικτα και εκφραστικά περιβάλλον εργασίας χρήστη. Η Google κυκλοφόρησε το Flutter τον Μάιο του 2017. Είναι ένα γρήγορο και ασφαλές πλαίσιο ανάπτυξης για smartphone και άλλες συσκευές. Το Flutter παρέχει παραγωγικές υπηρεσίες για τη δημιουργία γρήγορων εφαρμογών. Έχει αποδειχθεί ένα ανοιχτό μοντέλο ανάπτυξης.

Είναι μια ιδανική επιλογή για την ανάπτυξη εφαρμογών για Android και iOS. Η γλώσσα Dart χρησιμοποιείται για την ενσωμάτωση πολλών χαρακτηριστικών αυτών των συσκευών. Οι προγραμματιστές μπορούν να χρησιμοποιήσουν τη βάση ενός κώδικα με τον κατάλληλο τρόπο για τη συλλογή πολλών πλατφορμών.

Οι προγραμματιστές μπορούν να χρησιμοποιούν ταυτόχρονα την τεχνολογία Flutter για εφαρμογές iOS και Android. Εξοικονομεί την προσπάθεια και το χρόνο που απαιτείται για την ανάπτυξη μίας εφαρμογής για πολλές πλατφόρμες ξεχωριστά. Οι C ++ και Dart είναι οι κύριες γλώσσες αυτής της εφαρμογής και της τεχνολογίας. Η τεχνολογία Flutter χρησιμοποιείται κυρίως για την ανάπτυξη εφαρμογών Windows, Mac, iOS και Android.

Η πρώτη έκδοση του Flutter ήταν γνωστή ως κωδικός όνομα "Sky" και έτρεξε στο λειτουργικό σύστημα Android. Αποκαλύφθηκε στη σύνοδο κορυφής προγραμματιστών του Dart το 2015. Η Google ανακοίνωσε το Flutter Release Preview 2, η οποία είναι η τελευταία μεγάλη κυκλοφορία πριν από το Flutter 1.0. Στις 4 Δεκεμβρίου 2018, το Flutter 1.0 κυκλοφόρησε στην εκδήλωση Flutter Live, υποδηλώνοντας την πρώτη "σταθερή" έκδοση του Framework. Στις 11 Δεκεμβρίου 2019, το Flutter 1.12 κυκλοφόρησε στο Flutter Interactive event.

Στις 6 Μαΐου 2020, κυκλοφόρησε το Dart SDK στην έκδοση 2.8 και το Flutter στην έκδοση 1.17.0. Στις 3 Μαρτίου 2021, η Google κυκλοφόρησε το Flutter 2 κατά τη διάρκεια μιας διαδικτυακής εκδήλωσης Flutter Engage. Αυτή η σημαντική ενημέρωση έφερε επίσημη υποστήριξη για εφαρμογές που βασίζονται στον Ιστό, καθώς και υποστήριξη για εφαρμογές επιτραπέζιων υπολογιστών για Windows, MacOS και Linux.



Εικόνα 3.13: Flutter

Το Flutter είναι ένα πλαίσιο ανοιχτού κώδικα για τη δημιουργία υψηλής ποιότητας, υψηλής απόδοσης εφαρμογών για κινητά σε λειτουργικά συστήματα για κινητά - Android και iOS. Παρέχει ένα απλό, ισχυρό, αποδοτικό και εύκολο στην κατανόηση SDK για δημιουργία εφαρμογών για κινητά στη γλώσσα της Google, Dart.

Το Flutter είναι ένα δωρεάν UI πλαίσιο ανοιχτού κώδικα για κινητά που δημιουργήθηκε από την Google και κυκλοφόρησε τον Μάιο του 2017. Επιτρέπει να δημιουργήσουμε μια εγγενή εφαρμογή για κινητά με μία μόνο βάση κώδικα. Αυτό σημαίνει ότι μπορούμε να χρησιμοποιήσουμε μία γλώσσα προγραμματισμού και μία βάση κώδικα για να δημιουργήσουμε δύο διαφορετικές εφαρμογές (για iOS και Android).

Το Flutter αποτελείται από δύο σημαντικά μέρη:

Ένα SDK (Κιτ ανάπτυξης λογισμικού): Μια συλλογή εργαλείων που θα μας βοηθήσουν να αναπτύξουμε τις εφαρμογές σας. Αυτό περιλαμβάνει εργαλεία για τη μεταγλώττιση του κώδικα σε εγγενή κώδικα μηχανής (κωδικός για iOS και Android).

Ένα Πλαίσιο (Βιβλιοθήκη UI που βασίζεται σε widget): Μια συλλογή από επαναχρησιμοποιήσιμα στοιχεία διεπαφής χρήστη (κουμπιά, εισόδους κειμένου, ρυθμιστικά κ.ο.κ.) που μπορούμε να εξατομικεύσουμε για τις δικές μας ανάγκες.

Ένα από τα καλύτερα χαρακτηριστικά του Flutter είναι η γρήγορη και συχνή απόκρισή του στην αλλαγή κάτι. Αυτές οι αλλαγές μπορεί να σχετίζονται με αλλαγές στον κώδικα σε υλικό, εξομοιωτές και προσομοιωτές. Αυτή η αλλαγή μπορεί να συμβεί σε δευτερόλεπτα και δεν χρειάζεται να την επανεκκινήσετε. Αυτό είναι επιτυχές όχι μόνο για τη διόρθωση σφαλμάτων, αλλά συμβάλλει επίσης πολύ στη δημιουργία διεπαφών χωρίς καμία παύση.

Δεν υπάρχει καμία ανάγκη για συγκεκριμένες γνώσεις ή τεχνικές για τον προγραμματισμό εγγράφων, και αυτή η δυνατότητα διευκολύνει τη χρήση. Πολλοί προγραμματιστές, καθώς και μη προγραμματιστές, έχουν σχεδιάσει πολλές εφαρμογές flutter. Αυτό δείχνει ότι είναι τόσο απλό να χειριστεί.

Το Flutter παρέχει μια πλατφόρμα εύχρηστων λειτουργιών που την καθιστούν πιο αποτελεσματική και εγγενή μεταξύ των χρηστών. Εάν κάποιος χρησιμοποιεί ήδη Swift, React Native ή Java, μπορεί εύκολα να διαφοροποιήσει το Flutter από αυτά με μεγάλη ακρίβεια. Χωρίς δέσμη κωδικών, κανόνων ή αυστηρών κανονισμών, το Flutter είναι πολύ εύκολο στο χειρισμό. Αυτή είναι μια πραγματική εφαρμογή που ο χρήστης θα απολαύσει και θα αγαπήσει κατά τη χρήση του.

Το Flutter είναι γεμάτο από θησαυρούς και θα σας καταπλήξει σε κάθε βήμα αν πλησιάσετε. Η εφαρμογή το αποδεικνύει καλύτερα κάθε φορά που παραμένει ασφαλές και έχει καλά σημεία μαζί του. Το Flutter παρέχει έναν καλύτερο τρόπο για την ασφαλή συναρμολόγηση εγγράφων. Για παράδειγμα, εάν αντιμετωπίζετε κάποιο πρόβλημα με τον κωδικό widget, είναι πάντα διαθέσιμο να απαντήσετε με καλά καθορισμένο τρόπο.

Για να αναπτύξουμε το Flutter, θα χρησιμοποιήσουμε μια γλώσσα προγραμματισμού που ονομάζεται Dart. Η γλώσσα δημιουργήθηκε από την Google τον Οκτώβριο του 2011, αλλά έχει βελτιωθεί πολύ τα τελευταία χρόνια.

Το Dart επικεντρώνεται στην ανάπτυξη front-end και μπορούμε να τη χρησιμοποιήσουμε για να δημιουργήσουμε εφαρμογές για κινητά και web.

Για να δημιουργήσουμε εφαρμογές για Android θα πρέπει να εγκαταστήσουμε το Android Studio και να κάνουμε τις ανάλογες ρυθμίσεις.

Μπορείτε να ακολουθήσετε τον παρακάτω οδηγό για εγκατάσταση και ρύθμιση flutter για Windows.

[16]

Για να δημιουργήσουμε εφαρμογές για iPhone-iOS θα πρέπει να εγκαταστήσουμε το XCode σε Mac και να κάνουμε τις ανάλογες ρυθμίσεις.

## Κεφάλαιο 4ο: Τα πρόσθετα στο ηλεκτρονικό κατάστημα

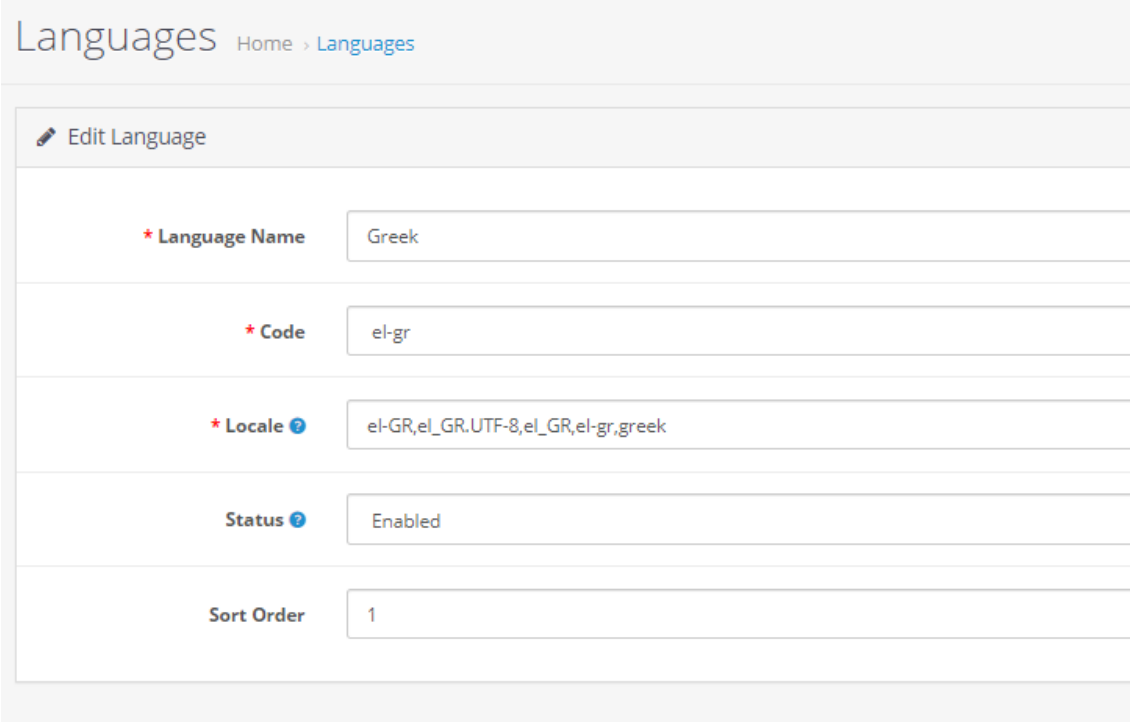
### 4.1 Εισαγωγή

Η εργασία αφορά τη δημιουργία Android εφαρμογής γρήγορης καταχώρησης προϊόντος σε ένα ηλεκτρονικό κατάστημα που είναι υλοποιημένο με το OpenCart.

Στο κεφάλαιο θα αναλυθεί η ρύθμιση του ηλεκτρονικού καταστήματος και η παραμετροποίηση ή τροποποίηση τους κώδικα του OpenCart για την επίτευξη του κώδικα.

Αρχικά θα κατεβάζουμε το αρχείο για την υποστήριξη της Ελληνικής Γλώσσας και θα ακολουθήσουμε τις οδηγίες που περιέχονται στο [17].

Η ρύθμιση της γλώσσας φαίνεται στις εικόνες 4.1 και 4.2.



Languages <a href="#">Home</a> > <a href="#">Languages</a>	
<a href="#">✎ Edit Language</a>	
* Language Name	<input type="text" value="Greek"/>
* Code	<input type="text" value="el-gr"/>
* Locale <a href="#">?</a>	<input type="text" value="el-GR,el_GR.UTF-8,el_GR,el-gr,greek"/>
Status <a href="#">?</a>	<input type="text" value="Enabled"/>
Sort Order	<input type="text" value="1"/>

Εικόνα 4.1: Ρύθμιση γλώσσας

Settings Home > Stores > Settings

Edit Setting

General Store **Local** Option Image Mail Server

Country Greece

Region / State Epirus

Language Greek

Administration Language English

Currency Euro

Auto Update Currency  Yes  No

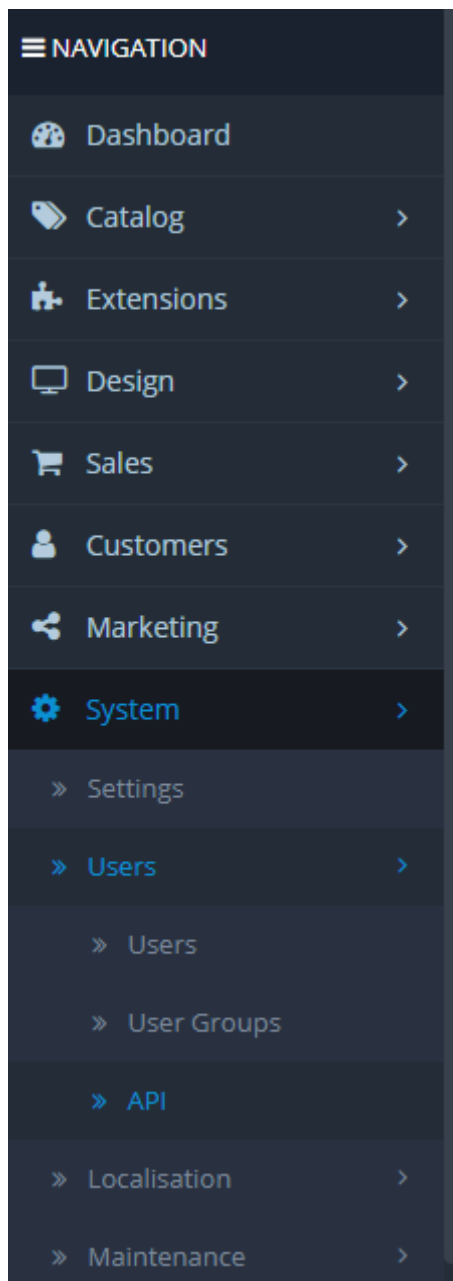
Length Class Centimeter

Weight Class Kilogram

Εικόνα 4.2: Ενεργοποίηση γλώσσας στο σύστημα

## 4.2 Ρύθμιση του API στο opencart

Στη συνέχεια θα ρυθμίσουμε το API OpenCart για να μπορούμε να χειριστούμε από το κινητό-εφαρμογή κάποιες από τις λειτουργίες του.



Εικόνα 4.3: API του OpenCart

Πηγαίνουμε στο Σύστημα - > Χρήστες και επιλέγουμε το API

APIs Home > APIs

Edit API

General IP Addresses Session

\* API Username

\* API Key

Status

Εικόνα 4.4: API Username και Key

Δημιουργούμε το API Username και Key

Στη συνέχεια ρυθμίζουμε τις διευθύνσεις IP που μπορεί να επιτρέπει συνδέσεις.

Edit API

General IP Addresses Session

Below you can create a list of IP's allowed to access the API. Your current IP is ::1

IP	
<input type="text" value="any"/>	<input type="button" value="−"/>
	<input type="button" value="+"/>

Εικόνα 4.5: Ρύθμιση IP για επιτρεπτές συνδέσεις

APIS Home > APIS

Edit API

General IP Addresses **Session**

Token	IP	Date Added	Date Modified
3895e31c1048ee7694de57af70	any	26/03/2021 17:53:44	26/03/2021 18:07:41

Εικόνα 4.6: API tokens

Στη τελευταία καρτέλα βλέπουμε τα tokens που χρησιμοποιούνται από διάφορες εφαρμογές που χρησιμοποιούν το API του ηλεκτρονικού καταστήματος.

### 4.3 Τροποποίηση στον κώδικα του opencart API

Θα παρουσιάσουμε τις προσθήκες και τις αλλαγές στο κώδικα για το κάθε χρήστη που θέλει να δημιουργήσει ένα παρόμοιο σύστημα.

Αρχικά πρέπει να κάνουμε τη ρύθμιση που παρουσιάσαμε στην προηγούμενη παράγραφο 4.2

Οι αλλαγές στο opencart γίνονται στα εξής αρχεία:

catalog\controller\api\login.php

catalog\controller\api\cart.php

catalog\model\catalog\productadd.php

catalog\controller\startup\session.php

Κρίθηκε πολύ σημαντικό για κάποιον που θέλει να κάνει κάτι παρόμοιο να ακολουθήσει ακριβώς την ίδια διαδικασία και να τροποποιήσει τα παραπάνω αρχεία με τον κώδικα που δίνεται παρακάτω.

Δημιουργήθηκαν αρχεία και άλλα τροποποιήθηκαν σε τέτοιο βαθμό ώστε να μας εξυπηρετήσει.

Να τονισθεί ότι από μόνο του το opencart δεν υποστηρίζει προσθήκη και αποστολή φωτογραφίας μέσω του API από τρίτες εφαρμογές.

Με την υλοποίηση του παρακάτω κώδικα αυτός ο στόχος επιτεύχθηκε.

### 4.3.1 Αpxείο login.php

```
class ControllerApiLogin extends Controller {
    public function index() {
        $this->load->language('api/login');

        $json = array();

        $this->load->model('account/api');

        // Login with API Key
        if(isset($this->request->post['username'])) {
            $api_info = $this->model_account_api->login($this->request-
>post['username'], $this->request->post['key']);
        } else {
            $api_info = $this->model_account_api->login('Default', $this->request-
>post['key']);
        }

        if ($api_info) {
            // Check if IP is allowed
            $ip_data = array();

            $results = $this->model_account_api->getApiIps($api_info['api_id']);

            foreach ($results as $result) {
                $ip_data[] = trim($result['ip']);
            }

            if (!$json) {
                $json['success'] = $this->language->get('text_success');

                $session = new Session($this->config->get('session_engine'), $this-
>registry);

                $session->start();

                $this->model_account_api->addApiSession($api_info['api_id'], $session-
>getId(), 'any');

                $session->data['api_id'] = $api_info['api_id'];

                // Create Token
                $json['api_token'] = $session->getId();
            } else {
                $json['error']['key'] = $this->language->get('error_key');
            }
        }
    }
}
```

```

        $this->response->addHeader('Content-Type: application/json');
        $this->response->setOutput(json_encode($json));
    }
}

```

### 4.3.2 Αρχείο cart.php

```

class ControllerApiCart extends Controller {

    public function add() {
        $this->load->language('api/cart');

        $json = array();

        if (!isset($this->session->data['api_id'])) {
            //if(0){
                $json['error'] = $this->language->get('error_permission');
            } else {

                $title = 'τίτλος κενός';
                if (isset($this->request->post['title'])) {
                    $title = $this->request->post['title'];
                    if (empty($title)) {
                        $json['error'] = 'Ο τίτλος είναι κενός';
                        $this->response->addHeader('Content-Type: application/json');
                        $this->response->setOutput(json_encode($json));
                        return;
                    }
                }
            }
        } else {
            $json['error'] = 'Δεν έχει οριστεί το πεδίο Τίτλος';
            $this->response->addHeader('Content-Type: application/json');
            $this->response->setOutput(json_encode($json));
            return;
        }

        $description = $title;
        if (isset($this->request->post['description'])) {
            if (strlen($this->request->post['description'])>0)
                $description = $this->request->post['description'];
        }

        $product_id_clone = 0;
        if (isset($this->request->post['product_id_clone'])) {
            if (is_numeric($this->request->post['product_id_clone']))
                {

```

```

$product_id_clone = $this->request->post['product_id_clone'];
}
else
{
    $json['error'] = 'Το id του προϊόντος που έβαλες δεν είναι αριθμός';
    $this->response->addHeader('Content-Type: application/json');
    $this->response->setOutput(json_encode($json));
    return;
}
}
$activate = 0;
if (isset($this->request->post['activate'])) {
    if (is_numeric($this->request->post['activate']))
    {
        $activate = $this->request->post['activate'];
    }
    else
    {
        $activate=0;
    }
}
}

$quantity = 1;
if (isset($this->request->post['quantity'])) {
    if (is_numeric($this->request->post['quantity']))
    {
        $quantity = $this->request->post['quantity'];

        if($quantity<=0 || $quantity>100)
        {
            $json['error'] = 'Η τιμή που έβαλες δεν είναι μέσα στα όρια (τιμή>0
και τιμή<100)';
            $this->response->addHeader('Content-Type: application/json');
            $this->response->setOutput(json_encode($json));
            return;
        }
    }
    else
    {
        $quantity=1;
    }
}
}

$price = 10;
if (isset($this->request->post['price'])) {
    if (is_numeric($this->request->post['price']))
    {
        $price = $this->request->post['price'];
    }
}
}

```

```

    }
    else
    {
        $json['error'] = 'Η τιμή που έβαλες δεν είναι αριθμός';
        $this->response->addHeader('Content-Type: application/json');
        $this->response->setOutput(json_encode($json));
        return;
    }

    if($price<=0 || $price>5000)
    {
        $json['error'] = 'Η τιμή που έβαλες δεν είναι μέσα στα όρια (τιμή>0 και
τιμή<5000)';
        $this->response->addHeader('Content-Type: application/json');
        $this->response->setOutput(json_encode($json));
        return;
    }
}
else{
    $json['error'] = 'Δεν έχεις βάλει τιμή';
    $this->response->addHeader('Content-Type: application/json');
    $this->response->setOutput(json_encode($json));
    return;
}

// Image
$this->response->addHeader('Content-Type: application/json');

    //$folder = 'demo/';
    $folder = '';
    $target_file_dir = "catalog/".$folder;
    $target_file = $target_file_dir. basename( $_FILES["image"]["name"]);
    $imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));
    $uploadOk = 1;

    $check = getimagesize($_FILES["image"]["tmp_name"]);
    if($check !== false) {
        //echo "File is an image - " . $check["mime"] . ".";
        $uploadOk = 1;
    } else {
        $this->response->setOutput("File is not an image.");
        return;
        $uploadOk = 0;
    }

    $titlename = str_replace(' ', '_', $title);
    $target_file = $target_file_dir.$titlename."_".substr(time(),6).".$im
ageFileType;

```

```

// Check if file already exists
if (file_exists($target_file)) {
    $this->response->setOutput("Sorry, file name already exists.");
    return;
    $uploadOk = 0;
}

// Check file size
if ($_FILES["image"]["size"] > 500000) {
    $this->response->setOutput("Sorry, your file is too large.");
    return;
    $uploadOk = 0;
}

// Allow certain file formats
if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType
!= "jpeg"
&& $imageFileType != "gif" ) {
    $this->response-
>setOutput("Sorry, only JPG, JPEG, PNG & GIF files are allowed.");
    return;
    $uploadOk = 0;
}

// Check if $uploadOk is set to 0 by an error
if ($uploadOk == 0) {
    $this->response->setOutput("Sorry, your file was not uploaded.");
    return;
    // if everything is ok, try to upload file
} else {
    if (move_uploaded_file($_FILES["image"]["tmp_name"], "image/".$target_file)) {
    } else {
        $this->response-
>setOutput("Sorry, there was an error uploading your file.");
        return;;
    }
}

try {
    $this->load->model('catalog/productadd');

    //C:\xampp\htdocs\dipaeapi\admin\model\catalog\product.php
    $productid = $this->model_catalog_productadd-
>addProductApi($product_id_clone, $title,$description,$price,$target_file,$activate, $quantity);

    $this->response->addHeader('Content-Type: application/json');
    $this->response->setOutput(json_encode($productid));
}

```

```

        return;
    } catch (Exception $e) {
        $this->response->addHeader('Content-Type: application/json');
        $this->response->setOutput(json_encode($e));
        return;
    }

    $json['success'] = 'Όλα πήγαν όλα';
}

$this->response->addHeader('Content-Type: application/json');
$this->response->setOutput(json_encode($json));
}

```

### 4.3.3 Αρχείο productadd.php

Σσςς

```

class ModelCatalogProductadd extends Model {

    public function addProductApi($product_id_clone, $title,$description,$price,$target_file, $activate, $quantity) {

        $querylastid = $this->db->query("SELECT product_id FROM " . DB_PREFIX . "product ORDER BY product_id DESC LIMIT 1");
        if($product_id_clone == 0)
        {
            $product_id = $querylastid->row['product_id'];
        }
        else
        {
            $product_id = $product_id_clone;
        }

        $query = $this->db->query("SELECT DISTINCT * FROM " . DB_PREFIX . "product p WHERE p.product_id = '" . (int)$product_id . "'");

        if ($query->num_rows) {
            $data = $query->row;

            $data['sku'] = '';
            $data['upc'] = '';
            $data['viewed'] = '0';
            $data['keyword'] = '';
            $data['status'] = '0';
        }
    }
}

```

```

$data['product_attribute'] = $this->getProductAttributes($product_id);
$data['product_description'] = $this->getProductDescriptions($product_id);

for ($i=1; $i <= count($data['product_description']); $i++) {
    try {
        $data['product_description'][$i]['description'] = $description;
        $data['product_description'][$i]['name'] = $title;
        $data['product_description'][$i]['meta_title'] = $title;
    } catch (Exception $e) {

    }
}

$data['product_discount'] = $this->getProductDiscounts($product_id);
$data['product_filter'] = $this->getProductFilters($product_id);
$data['product_image'] = $this->getProductImages($product_id);

$data['product_option'] = $this->getProductOptions($product_id);
$data['product_related'] = $this->getProductRelated($product_id);
$data['product_reward'] = $this->getProductRewards($product_id);
$data['product_special'] = $this->getProductSpecials($product_id);
$data['product_category'] = $this->getProductCategories($product_id);
$data['product_download'] = $this->getProductDownloads($product_id);
$data['product_layout'] = $this->getProductLayouts($product_id);
$data['product_store'] = $this->getProductStores($product_id);
$data['product_recurrings'] = $this->getRecurrings($product_id);

$data['quantity'] = $quantity;
$data['model'] = $querylastid;
$data['price'] = $price;
$data['status'] = $activate;

//return
$this->addProduct($data);

$querylastid = $this->db->query("SELECT product_id FROM " . DB_PREFIX . "product ORDER BY product_id DESC LIMIT 1");
$product_id = $querylastid->row['product_id'];
$this->db->query("UPDATE " . DB_PREFIX . "product SET image = '" . $this->db->escape($target_file) . "' WHERE product_id = '" . (int) $product_id."'");

}

return $product_id;
}
}

```

### 4.3.4 Αρχείο session.php

```
class ControllerStartupSession extends Controller {
    public function index() {
        if (isset($this->request->get['api_token']) && isset($this->request->get['route']) && substr($this->request->get['route'], 0, 4) == 'api/') {
            $this->db->query("DELETE FROM `" . DB_PREFIX . "api_session` WHERE TIMESTAMPADD(HOUR, 1, date_modified) < NOW()");
            $api_query = $this->db->query("SELECT DISTINCT * FROM `" . DB_PREFIX . "api` `a` LEFT JOIN `" . DB_PREFIX . "api_session` `as` ON (a.api_id = as.api_id) LEFT JOIN `" . DB_PREFIX . "api_ip` `ai` ON (a.api_id = ai.api_id) WHERE a.status = '1' AND `as`.`session_id` = '" . $this->db->escape($this->request->get['api_token']) . "'");
            if ($api_query->num_rows) {
                $this->session->start($this->request->get['api_token']);

                // keep the session alive
                $this->db->query("UPDATE `" . DB_PREFIX . "api_session` SET `date_modified` = NOW() WHERE `api_session_id` = '" . (int)$api_query->row['api_session_id'] . "'");
            }
        } else {
            if (isset($_COOKIE[$this->config->get('session_name')])) {
                $session_id = $_COOKIE[$this->config->get('session_name')];
            } else {
                $session_id = '';
            }

            $this->session->start($session_id);

            setcookie($this->config->get('session_name'), $this->session->getId(), ini_get('session.cookie_lifetime'), ini_get('session.cookie_path'), ini_get('session.cookie_domain'));
        }
    }
}
```

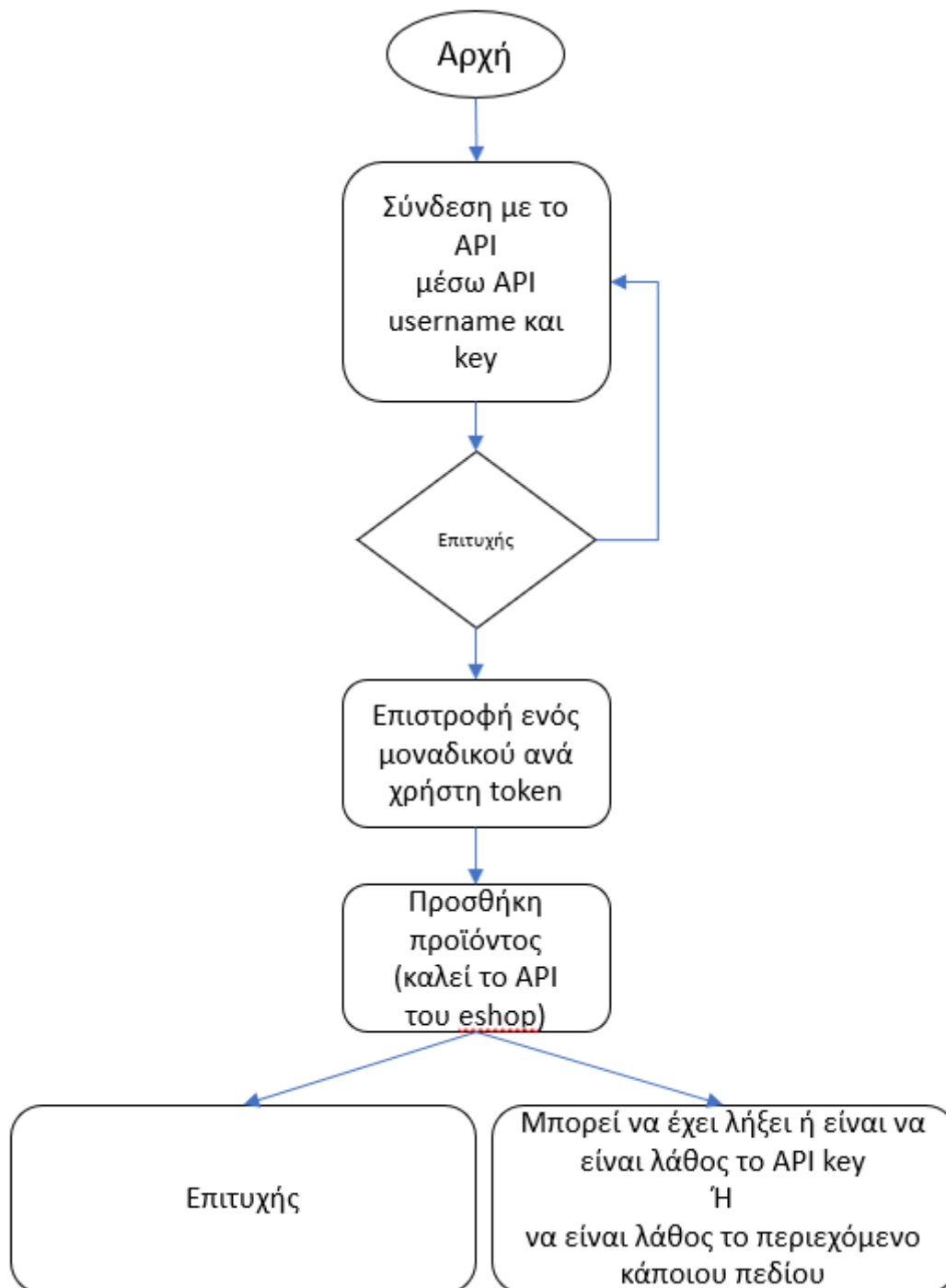
## 4.4 Ανάλυση ασφάλειας στο σύστημα

Αρχικά μετά την εγκατάσταση πρέπει να διαγραφεί ο φάκελος εγκατάστασης και πρέπει να μετακινηθεί ο φάκελος storage εκτός του φακέλου εγκατάστασης. Θα πρέπει να μετακινηθεί εκτός για λόγους ασφάλειας.

Η πρώτη διασφάλιση σύνδεσης κάποιου τρίτου με το API του eshop διασφαλίζεται με τα API username και key που εξάγονται μέσω του διαχειριστικού.

Κατά την επιτυχημένη σύνδεση επιστρέφεται ένα μοναδικό token και με αυτό η εφαρμογή μπορεί να καλεί όποιο request επιθυμεί και επιτρέπει το API.

Η διαδικασία φαίνεται στην παρακάτω εικόνα διάγραμμα.

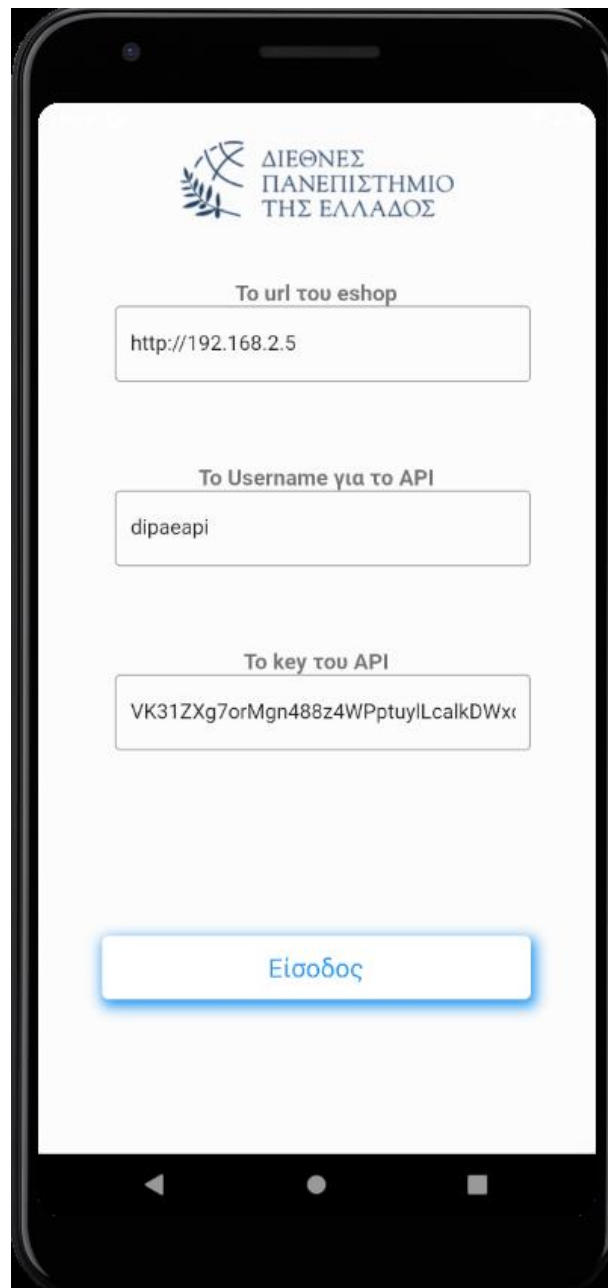


Εικόνα 4.7: Διαδικασία σύνδεσης και αποστολής προϊόντος μέσω του API

## Κεφάλαιο 5ο: Η εφαρμογή στο κινητό

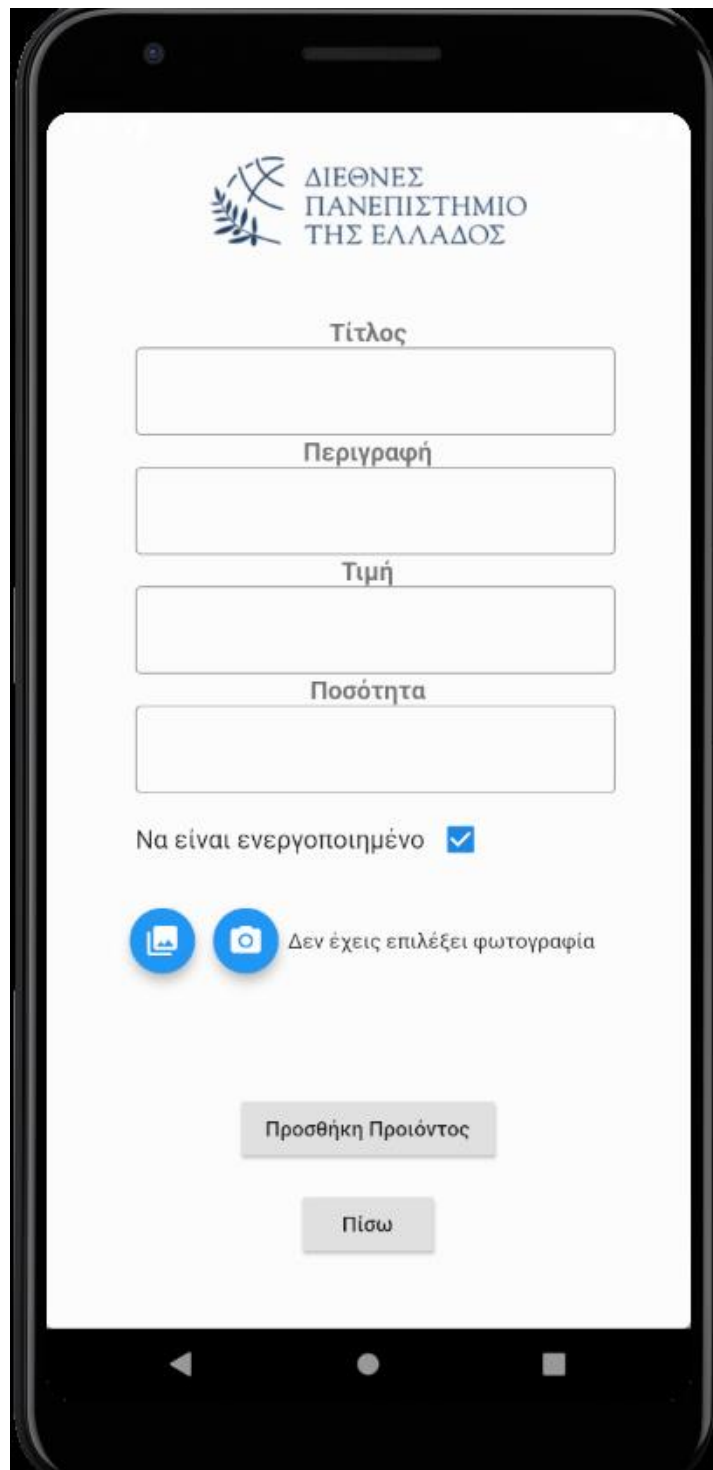
### 5.1 Εισαγωγή

Στο παρόν κεφάλαιο θα αναλυθεί η Android εφαρμογή γρήγορης καταχώρησης προϊόντος στο ηλεκτρονικό κατάστημα. Ο ιδιοκτήτης θα μπαίνει στην εφαρμογή, θα φωτογραφίζει το προϊόν, θα του βάζει τίτλο και τιμή και θα το ανεβάζει απευθείας στο eshop χωρίς να χάνει χρόνο και να πρέπει να συνδεθεί στο διαχειριστικό του eshop μέσω του υπολογιστή του. Για την υλοποίηση της εφαρμογής επιλέχθηκε το Flutter.



Εικόνα 5.1: Πρώτη οθόνη – Σύνδεση με το API

Στην Εικόνα 5.1 παρουσιάζεται η πρώτη οθόνη όταν ο χρήστης ανοίξει την εφαρμογή. Θα πρέπει να βάλει το σωστό URL του ηλεκτρονικού καταστήματος και να βάλει τα στοιχεία του που περιγράψαμε στην παράγραφο 4.2.



ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΟΣ



Τίτλος

Περιγραφή

Τιμή

Ποσότητα

Να είναι ενεργοποιημένο

  Δεν έχεις επιλέξει φωτογραφία

Προσθήκη Προϊόντος

Πίσω

Εικόνα 5.2: Οθόνη για την προσθήκη προϊόντος

Στην Εικόνα 5.2 παρουσιάζεται η δεύτερη οθόνη όταν ο χρήστης κάνει επιτυχημένη σύνδεση. Εδώ μπορεί να βάλει το προϊόν του. Πρέπει να βάλει

- Τίτλο
- Περιγραφή
- Τιμή
- Ποσότητα
- Μια φωτογραφία

όπως φαίνεται στην Εικόνα 5.3.

ΔΙΕΘΝΕΣ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΤΗΣ ΕΛΛΑΔΟΣ


Τίτλος  
Μπλούζα

Περιγραφή  
Μπλούζα Φούτερ Μπλέ

Τιμή  
20

Ποσότητα  
12

Να είναι ενεργοποιημένο

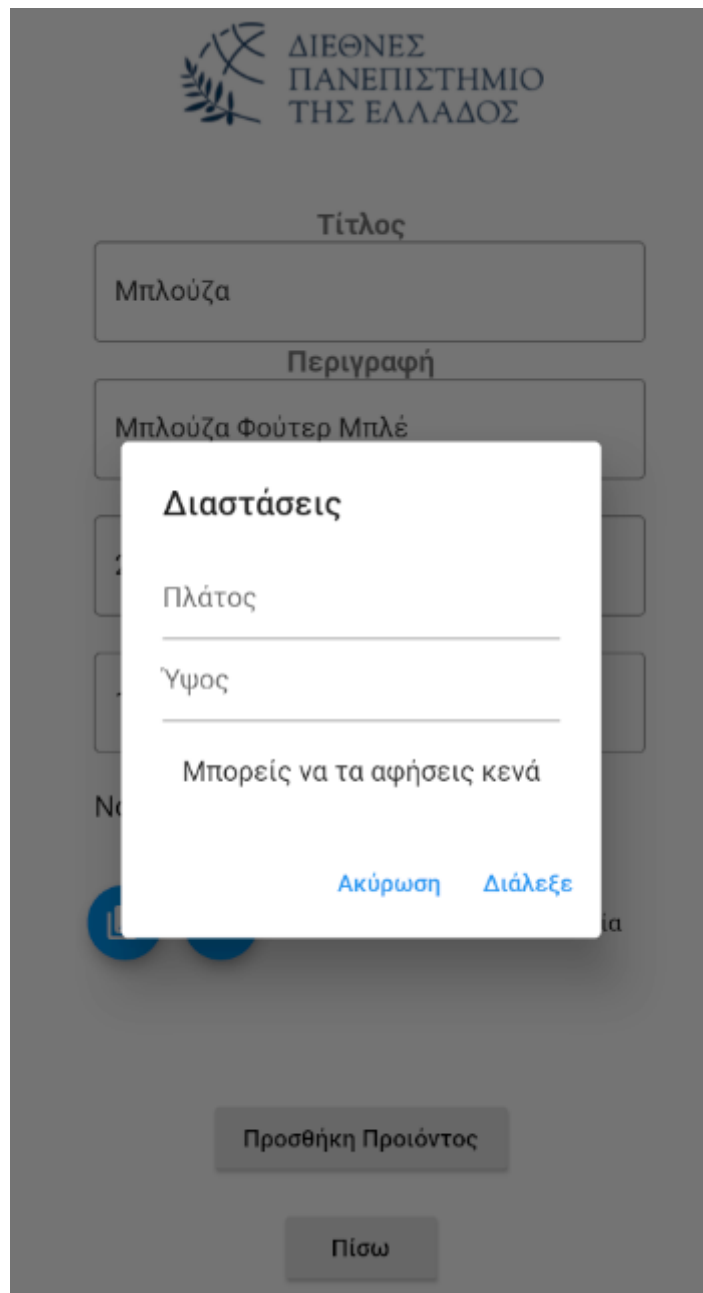
  Δεν έχεις επιλέξει φωτογραφία

Προσθήκη Προϊόντος

Πίσω

Εικόνα 5.3: Τιμές στα πεδία

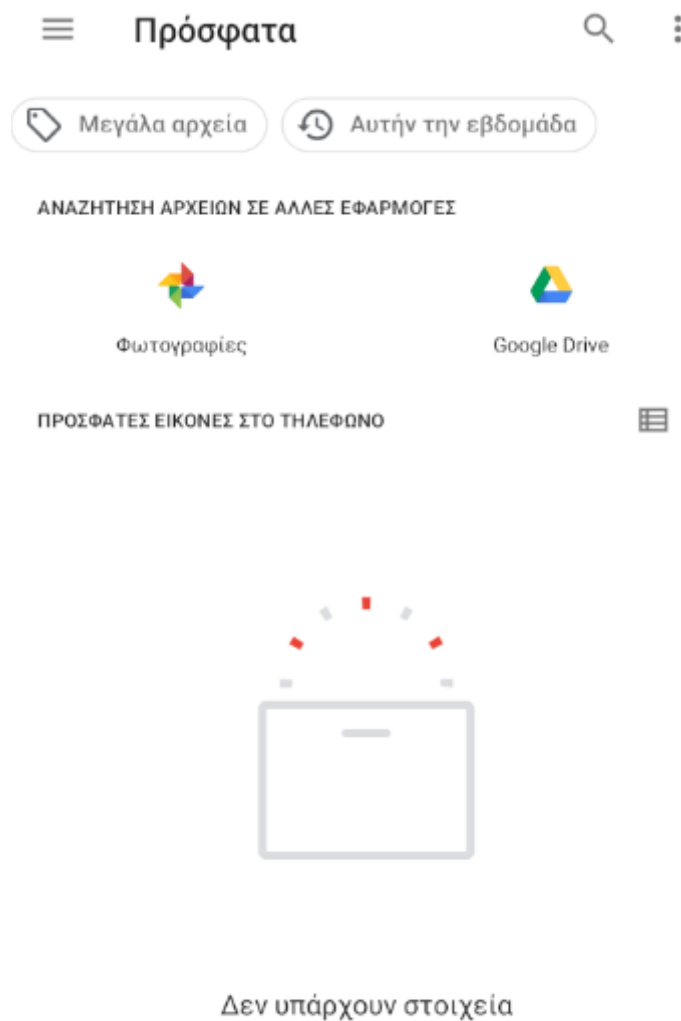
Μπορεί να επιλέξει αν το προϊόν που θα προσθέσει θα είναι αυτόματα ενεργοποιημένο.



Εικόνα 5.4: Ρύθμιση για διαστάσεις εικόνας του προϊόντος

Στην Εικόνα 5.4 παρουσιάζεται η οθόνη όταν ο χρήστης πατήσει την προσθήκη εικόνας. Του ζητάει προτεινόμενες διαστάσεις εικόνας, ενώ μπορεί να τα αφήσει κενά.

Μπορεί να προσθέσει εικόνα από τη βιβλιοθήκη του κινητού ή από τη κάμερα.



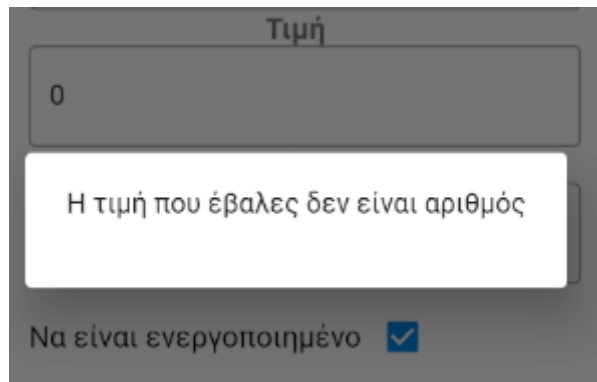
Εικόνα 5.5: Αναζήτηση στο δίσκο του κινητού για επιλογή εικόνας

Στην Εικόνα 5.5 παρουσιάζεται η οθόνη όπου ο χρήστης επιλέγει να φορτώσει μια εικόνα από το δίσκο του κινητού του.

Στην Εικόνα 5.6 παρουσιάζεται η οθόνη όπου χρήστης έχει ανεβάσει μια εικόνα επιτυχημένα.

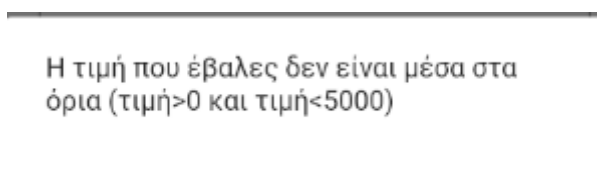


Εικόνα 5.6: Επιλογή και προβολή εικόνας

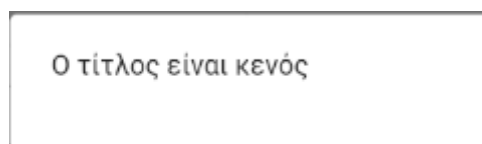


Εικόνα 5.7: Λάθος τιμής στο πεδίο Τιμή – Δεν είναι αριθμός

Αν χρήστης δεν βάλεις σωστά την τιμή τότε εμφανίζεται στην οθόνη κατάλληλο μήνυμα όπως φαίνεται στις Εικόνες 5.7, 5.8 και 5.9.

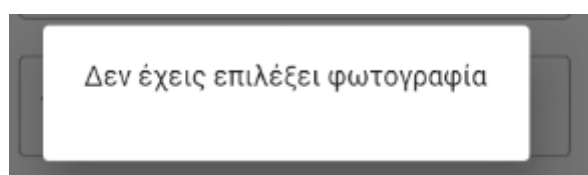


Εικόνα 5.8: Λάθος τιμής στο πεδίο Τιμή – Ο αριθμός είναι εκτός ορίων



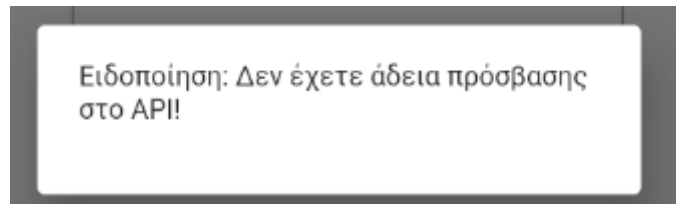
Εικόνα 5.9: Λάθος στο πεδίο Τίτλος

Σε περίπτωση που προσπαθήσει να ανεβάσει το προϊόν χωρίς να έχει προσθέσει φωτογραφία τότε του βγαίνει το κατάλληλο μήνυμα και δεν τον αφήνει, όπως φαίνεται στην εικόνα 5.10.



Εικόνα 5.10: Προσπάθεια εκχώρησης προϊόντος χωρίς εικόνα

Επίσης, αν κάποιος χρήστης προσπαθήσει να μπει στο σύστημα μέσω του API χωρίς να έχει εξουσιοδότησή ή αν έχει βάλει ο χρήστης λάθος url ή στοιχεία τότε βγαίνει το μήνυμα που φαίνεται στην Εικόνα 5.11.



Εικόνα 5.11: Περίπτωση χρήσης του API χωρίς πρόσβαση ή εξουσιοδότηση

## 5.2 Κώδικας στο Flutter

Αρχείο: Αρχική Οθόνη – Σύνδεση με το API του eshop

Προσθήκη Πακέτων

```
import 'dart:io';  
import 'dart:async';  
import 'package:flutter/material.dart';  
import 'package:flutter/cupertino.dart';  
import 'package:flutter/services.dart';  
import 'package:http/http.dart' as http;  
import 'dart:convert' as convert;  
import 'package:dipaeapieshopapp/views/lang.dart';  
import 'package:dipaeapieshopapp/views/addproduct.dart';  
import 'package:shared_preferences/shared_preferences.dart';
```

Αρχικοποίηση της κλάσης για την πρώτη οθόνη

```
class HomePage extends StatefulWidget {  
  static const id = '/homepage';  
  HomePage({Key key, this.title}) : super(key: key);  
  
  final String title;  
  
  @override  
  _HomePageState createState() => _HomePageState();  
}
```

Καθορισμός των πεδίων

```

@override
void initState() {
  keyword1.text = 'http://192.168.2.5';
  keyword2.text = 'dipaeapiapi';
  keyword3.text =
    'VK31ZXg7orMgn488z4WPptuy1LcalkDWxdjrdunZ9zayNo3eyMjDiIjukfKaSMI8gZ6TvxNhj9GFv5g4C8
wbH1PBeuu7xL8eByw0LysTv2fLNIn1XowBAiNKd985i0DfpCJETGI8yocXHi9W4xdbmmnkpIdUqb8q4ppK1rm8riVG2
dFFT2kLGVNptxJTW38osc2WB4MM9XdIyNGXUQnykuDslkMrKwf3TYqTnfrKraoe9Pe3DgmQKkBNftJKNHlt';
  run();
}

```

Δυνατότητα επιλογής Ελληνικής ή Αγγλικής γλώσσας

```

var langid = 'en';
void run() async {
  SharedPreferences prefs = await SharedPreferences.getInstance();
  SharedPreferences.setMockInitialValues({});

  //Only at the beginning
  await prefs.setString('langid', 'el');
  langid = 'el';

  try {
    langid = await prefs.getString('langid');
  } catch (e) {
    await prefs.setString('langid', 'en');
  }

  setState(() {
    langid = langid;
  });
}

```

Ο τρόπος με τον οποίο η εφαρμογή καλεί το API για σύνδεση

```

Widget _submitButton() {
  return InkWell(
    onTap: () async {

      var url = keyword1.text + '/dipaeapi/index.php?route=api/login';
      var response = await client.post(url, headers: {}, body: {
        'username': keyword2.text,
        'key': keyword3
          .text,
      });

      if (response.statusCode == 200) {

```

```

var jsonResponse = convert.jsonDecode(response.body);
var api_token = jsonResponse['api_token'];
print('api_token: $api_token.');
```

```

print(response.body);
Navigator.push(
  context,
  MaterialPageRoute(
    builder: (context) =>
      AddProductPage(url: keyword1.text, apiToken: api_token)));
} else {
  print('Request failed with status: ${response.statusCode}.');
}
},
```

Με ποιο τρόπο εμφανίζονται τα πεδία στην Οθόνη

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: SingleChildScrollView(
      child: Container(
        padding: EdgeInsets.symmetric(horizontal: 50),
        height: MediaQuery.of(context).size.height,
        decoration: BoxDecoration(
          borderRadius: BorderRadius.all(Radius.circular(5)),
          boxShadow: <BoxShadow>[
            BoxShadow(
              color: Colors.grey.shade200,
              offset: Offset(2, 4),
              blurRadius: 5,
              spreadRadius: 2)
          ],
          gradient: LinearGradient(
            begin: Alignment.topCenter,
            end: Alignment.bottomCenter,
            colors: [
              Color.fromRGBO(252, 252, 252, 1),
              Color.fromRGBO(252, 252, 252, 1),
            ]
          )),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.center,
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          _label(),
          //_title(),
          Text(
            Lang.vals[langid]['url'],
```

```

        style: TextStyle(
            fontSize: 17,
            color: Colors.black54,
            fontWeight: FontWeight.bold),
    ),
    Expanded(
        child: Padding(
            padding: EdgeInsets.only(left: 10),
            child: TextField(
                style: TextStyle(color: Colors.black87),
                cursorColor: Colors.black54,
                controller: keyword1,
                decoration: InputDecoration(border: OutlineInputBorder()),
            ),
        ),
    ),
    Text(
        Lang.vals[langid]['usernameapi'],
        style: TextStyle(
            fontSize: 17,
            color: Colors.black54,
            fontWeight: FontWeight.bold),
    ),
    Expanded(
        child: Padding(
            padding: EdgeInsets.only(left: 10),
            child: TextField(
                style: TextStyle(color: Colors.black87),
                cursorColor: Colors.black54,
                controller: keyword2,
                decoration: InputDecoration(border: OutlineInputBorder()),
            ),
        ),
    ),
    Text(
        Lang.vals[langid]['key'],
        style: TextStyle(
            fontSize: 17,
            color: Colors.black54,
            fontWeight: FontWeight.bold),
    ),
    Expanded(
        child: Padding(
            padding: EdgeInsets.only(left: 10),
            child: TextField(
                style: TextStyle(color: Colors.black87),
                cursorColor: Colors.black54,
                controller: keyword3,

```

```

        decoration: InputDecoration(border: OutlineInputBorder()),
      ),
    ),
  ),
  SizedBox(
    height: 80,
  ),
  _submitButton(),
  SizedBox(
    height: 80,
  ),
  // _backButton(),
  SizedBox(
    height: 20,
  ),
  SizedBox(
    height: 20,
  ),
],
),
),
),
);
}
}

```

Στην συνέχεια παρουσιάζεται το αρχείο για την προσθήκη προϊόντος

Για αρχή προστίθενται οι βιβλιοθήκες

```

import 'dart:io';
import 'dart:async';
import 'package:flutter/foundation.dart';
import 'package:flutter/material.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/services.dart';
import 'package:http/http.dart' as http;
import 'dart:convert' as convert;
import 'package:image_picker/image_picker.dart';
import 'package:dipaeapieshopapp/views/lang.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:dipaeapieshopapp/views/home.dart';

```

Προσθήκη των πεδίων και των μεταβλητών που απαιτούνται

```

class _AddProductPageState extends State<AddProductPage> {

```

```

final MethodChannel platform =
    MethodChannel('crossingthestreams.io/resourceResolver');

PickedFile _imageFile;
dynamic _pickImageError;
String _retrieveDataError;

bool valueEnable = true;
bool _isAddProductButtonDisabled;

final ImagePicker _picker = ImagePicker();
final TextEditingController maxWidthController = TextEditingController();
final TextEditingController maxHeightController = TextEditingController();

TextEditingController keyword1 = TextEditingController();
TextEditingController keyword2 = TextEditingController();
TextEditingController keyword3 = TextEditingController();
TextEditingController keyword4 = TextEditingController();
var client = http.Client();

@override
void initState() {
    keyword1.text = 'Μπλούζα';
    keyword2.text = 'Μπλούζα Φούτερ Μπλέ';
    keyword3.text = '20';
    keyword4.text = '12';
    _isAddProductButtonDisabled = false;

    run();
}

```

## Επιλογή γλώσσας

```

var langid = 'en';
void run() async {
    SharedPreferences prefs = await SharedPreferences.getInstance();
    SharedPreferences.setMockInitialValues({});

    //Only at the beginning
    await prefs.setString('langid', 'el');
    langid = 'el';

    try {
        langid = await prefs.getString('langid');
    } catch (e) {
        await prefs.setString('langid', 'en');
    }
}

```

```

setState(() {
  langid = langid;
});
}

```

### Επιλογή και διαμόρφωση εικόνας

```

void _onImageButtonPressed(ImageSource source, {BuildContext context}) async {
  await _displayPickImageDialog(context,
    (double maxWidth, double maxHeight) async {
    try {
      final pickedFile = await _picker.getImage(
        source: source,
        maxWidth: maxWidth,
        maxHeight: maxHeight,
      );
      setState(() {
        _imageFile = pickedFile;
      });
    } catch (e) {
      setState(() {
        _pickImageError = e;
      });
    }
  });
}

Future<void> _displayPickImageDialog(
  BuildContext context, OnPickImageCallback onPick) async {
  return showDialog(
    context: context,
    builder: (context) {
      return AlertDialog(
        title: Text('Διαστάσεις'),
        content: Column(
          mainAxisAlignment: MainAxisAlignment.min,
          children: <Widget>[
            TextField(
              controller: maxWidthController,
              keyboardType: TextInputType.numberWithOptions(decimal: true),
              decoration: InputDecoration(hintText: "Πλάτος"),
            ),
            TextField(
              controller: maxHeightController,
              keyboardType: TextInputType.numberWithOptions(decimal: true),
              decoration: InputDecoration(hintText: "Υψος"),
            ),
            SizedBox(

```

```

        height: 20,
      ),
      Text('Μπορείς να τα αφήσεις κενά'),
    ],
  ),
  actions: <Widget>[
    TextButton(
      child: const Text('Ακύρωση'),
      onPressed: () {
        Navigator.of(context).pop();
      },
    ),
    TextButton(
      child: const Text('Διάλεξε'),
      onPressed: () {
        double width = maxWidthController.text.isNotEmpty
          ? double.parse(maxWidthController.text)
          : null;
        double height = maxHeightController.text.isNotEmpty
          ? double.parse(maxHeightController.text)
          : null;
        onPick(width, height);
        Navigator.of(context).pop();
      },
    ),
  ],
);
});
}

```

## Προεπισκόπηση εικόνας

```

Widget _previewImage() {
  final Text retrieveError = _getRetrieveErrorWidget();
  if (retrieveError != null) {
    return retrieveError;
  }
  if (_imageFile != null) {
    if (kIsWeb) {
      return Image.network(_imageFile.path);
    } else {
      return Semantics(
        child: Image.file(File(_imageFile.path)),
        label: 'image_picker_example_picked_image');
    }
  }
  } else if (_pickImageError != null) {
    return Text(
      'Pick image error: $_pickImageError',
      textAlign: TextAlign.center,
    );
  }
}

```

```

    );
  } else {
    return Text(
      Lang.vals[langid]['msg_emptyimage'],
      textAlign: TextAlign.center,
    );
  }
}

```

Ο τρόπος με τον οποίο η εφαρμογή καλεί το API για προσθήκη νέου προϊόντος

```

void _addProduct() async {
  print("AddProduct");

  var url = widget.url +
    '/dipaeapi/index.php?route=api/cart/add&api_token=' +
    widget.apiToken;

  var request = http.MultipartRequest('POST', Uri.parse(url));
  request.fields['title'] = keyword1.text;
  request.fields['description'] = keyword2.text;
  request.fields['price'] = keyword3.text;
  request.fields['quantity'] = keyword4.text;
  if (this.valueEnable) {
    request.fields['activate'] = '1';
  } else {
    request.fields['activate'] = '0';
  }

  if (_imageFile == null) {
    showAlertDialog(context, Lang.vals[langid]['msg_emptyimage']);
  } else {
    _isAddProductButtonDisabled = true;
    request.files
      .add(await http.MultipartFile.fromPath('image', _imageFile.path));
    var responseA = await request.send();
    final response = await http.Response.fromStream(responseA);

    if (response.statusCode == 200) {
      _isAddProductButtonDisabled = false;
      print(response.body);
      if (response.body.contains('error')) {
        var jsonResponse = convert.jsonDecode(response.body);
        print(jsonResponse);
      }
    }
  }
}

```

```

        if (jsonResponse.length > 0) {
            showAlertDialog(context, jsonResponse['error']);
        } else {
            showAlertDialog(context, jsonResponse['error']);
        }
    } else {
        showAlertDialog(context, Lang.vals[langid]['productadded']);
    }
} else {
    _isAddProductButtonDisabled = false;
    //print('${response.statusCode}');
    var jsonResponse = convert.jsonDecode(response.body);
    print('${response.body}');
    if (jsonResponse['error'] != '')
        showAlertDialog(context, jsonResponse['error']);
}
}

setState(() {
    keyword3.text = '0';
    _isAddProductButtonDisabled = _isAddProductButtonDisabled;
});
}

Widget _submitButton() {
    return new RaisedButton(
        child: new Text(_isAddProductButtonDisabled
            ? "Hold on..."
            : Lang.vals[langid]['addproduct']),
        onPressed: _isAddProductButtonDisabled ? null : _addProduct,
    );
}
}

```

## Κεφάλαιο 6ο: Συμπεράσματα και προτάσεις βελτίωσης

Στην εργασία αυτή παρουσιάστηκε η υλοποίηση μιας Android εφαρμογής γρήγορης καταχώρησης προϊόντος σε ηλεκτρονικό κατάστημα. Ο επιχειρηματίας-ιδιοκτήτης θα μπαίνει στην εφαρμογή, θα φωτογραφίζει το προϊόν, θα του βάζει τίτλο και τιμή και θα το ανεβάζει απευθείας στο eshop χωρίς να χάνει χρόνο και να πρέπει να συνδεθεί στο διαχειριστικό του eshop μέσω του υπολογιστή του. Για την υλοποίηση του ηλεκτρονικού καταστήματος επιλέχθηκε το OpenCart.

Για την ασφάλεια έχουν ληφθεί υπόψη δύο βασικές παράμετροι. Στην πρώτη διασφαλίζεται η σύνδεση κάποιου τρίτου με το API του eshop και αυτό γίνεται μέσω των API username και key που εξάγονται μέσω του διαχειριστικού.

Επίσης, ο χρήστης (η εφαρμογή στο κινητό του) μετά την επιτυχημένη σύνδεση δέχεται ένα μοναδικό token από το API και με αυτό η εφαρμογή μπορεί να καλεί όποιο request επιθυμεί αφού επιτρέπει το API.

Μια βελτίωση που θα μπορούσε να υλοποιηθεί είναι να μπορεί ο χρήστης στην εφαρμογή να βλέπει όλα τα προϊόντα και να διαλέγει ποιο θέλει να κάνει κλώνο για να επιταχύνει την διαδικασία εισαγωγή τίτλου κτλ.

Επίσης μια πολύ σημαντική βελτίωση είναι να εξαχθεί η εφαρμογή και σε iOS (iPhone κτλ.), αφού με τον ίδιο κώδικα το Flutter παράγει εφαρμογή σε αυτό, κάτι που δεν δοκιμάστηκε αφού αδυνατούμε να βρούμε εύκολα πρόσβαση σε iPhone.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] <https://www.opencart.com/> [15 Μαΐ 2021]
- [2] [https://en.wikipedia.org/wiki/Comparison\\_of\\_shopping\\_cart\\_software](https://en.wikipedia.org/wiki/Comparison_of_shopping_cart_software) [15 Μαΐ 2021]
- [3] <https://barn2.com/woocommmerce-stats/#woocommmerce-usage-stats> [15 Φεβ 2021]
- [4] <https://wordpress.com> [15 Μαΐ 2021]
- [5] <https://www.toptal.com/magento/magento-website-development> [15 Μαΐ 2021]
- [6] <https://en.wikipedia.org/wiki/API> [15 Μαΐ 2021]
- [7] <https://www.prestashop.com> [15 Μαΐ 2021]
- [8] [https://www.opencart.com/index.php?route=marketplace/extension/info&extension\\_id=23800](https://www.opencart.com/index.php?route=marketplace/extension/info&extension_id=23800), 10-04-2021
- [9] [https://www.opencart.com/index.php?route=marketplace/extension/info&extension\\_id=23394&filter\\_search=rest%20API](https://www.opencart.com/index.php?route=marketplace/extension/info&extension_id=23394&filter_search=rest%20API), 10-04-2021
- [10] [https://www.opencart.com/index.php?route=marketplace/extension/info&extension\\_id=23801&filter\\_search=rest%20API](https://www.opencart.com/index.php?route=marketplace/extension/info&extension_id=23801&filter_search=rest%20API), 10-04-2021
- [11] [https://gps.uml.edu/online/tutorial/ftp/6.4.14/filezilla\\_ftps\\_tutorial.pdf](https://gps.uml.edu/online/tutorial/ftp/6.4.14/filezilla_ftps_tutorial.pdf)
- [12] <https://www.siteground.com/tutorials/phpmyadmin/>
- [13] [http://preview.themeforest.net/item/jewellery-opencart-responsive-template/full\\_screen\\_preview/7000576](http://preview.themeforest.net/item/jewellery-opencart-responsive-template/full_screen_preview/7000576)
- [14] S. Boukhary and E. Colmenares, "A Clean Approach to Flutter Development through the Flutter Clean Architecture Package," *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2019, pp. 1115-1120, doi: 10.1109/CSCI49370.2019.00211.
- [15] [https://en.wikipedia.org/wiki/Flutter\\_\(software\)](https://en.wikipedia.org/wiki/Flutter_(software)) [15 Μαΐ 2021]
- [16] <https://flutter.dev/docs/get-started/install/windows> [15 Μαΐ 2021]
- [17] [https://www.opencart.com/index.php?route=marketplace%2Fextension%2Finfo&extension\\_id=33334&filter\\_member=People%27s%20Code](https://www.opencart.com/index.php?route=marketplace%2Fextension%2Finfo&extension_id=33334&filter_member=People%27s%20Code) [15 Μαΐ 2021]