



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ

ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
«ΔΗΜΙΟΥΡΓΙΑ Ε-SHOP ΣΥΝΔΕΔΕΜΕΝΟ
ΜΕ ΠΡΟΓΡΑΜΜΑ ΜΗΧΑΝΟΓΡΑΦΗΣΗΣ»



Του φοιτητή
Παπαρισούλη Βαΐου
Αρ. Μητρώου: 515115

Επιβλέπων
Όνοματεπώνυμο: Τζέκης
Παναγιώτης
Βαθμίδα: Αναπληρωτής
Καθηγητής

21 Ιουνίου 2022

Τίτλος Δ.Ε.: “Δημιουργία e-shop συνδεδεμένο με πρόγραμμα μηχανογράφησης”

Κωδικός Δ.Ε.: 21398

Όνοματεπώνυμο φοιτητή: Παπαρισούλης Βάιος

Όνοματεπώνυμο εισηγητή: Τζεκης Παναγιώτης, Αν. Καθηγητής

Ημερομηνία ανάληψης Π.Ε.: Νοέμβριος 2021

Ημερομηνία περάτωσης Π.Ε. Ιούνιος 2022

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 διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Παπαρισούλη Βαΐου 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’ οποιονδήποτε 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 διπλωματικής 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.

Πρόλογος

Η προκείμενη Πτυχιακή Εργασία αφορά τον σχεδιασμό και την υλοποίηση ενός λειτουργικού ηλεκτρονικού καταστήματος, συνδεδεμένου με πρόγραμμα μηχανογράφησης. Έναυσμα για την ανάληψή της αποτέλεσε η λειτουργική και εμπορική ανάγκη για την ανάπτυξη ενός ηλεκτρονικού καταστήματος με την προαναφερθείσα δυνατότητα από προσωπική επιχείριση του φοιτητή που την εκπονεί.

Με το πέρας αυτής, αποσκοπούμε στην υλοποίηση ενός ιστοτόπου με επαγγελματικές προδιαγραφές και δυνατότητες επέκτασης και περαιτέρω προσαρμογής στις ανάγκες της επιχείρισης. Κίνητρο για το βέλτιστο αποτέλεσμα, πέραν της ικανοποίησης της λειτουργικής ανάγκης της επιχείρισης για την απόκτηση ενός ηλεκτρονικού καταστήματος, αποτελεί και η πρόκληση που συνεπάγεται η εκπόνηση ενός τέτοιου έργου, αλλά και η ηθική ικανοποίηση που θα μας αποφέρει η επιτυχία μας στο έργο γνωρίζοντας ότι δι' ιδίων κόπων καταφέραμε να δημιουργήσουμε κάτι προς όφελός μας.

Περίληψη

Ως ηλεκτρονικό εμπόριο ορίζεται η δυνατότητα του καταναλωτή να αγοράζει αγαθά ή υπηρεσίες μέσω Διαδικτύου. Στην πιο κοινότυπη, συνήθη μορφή του μπορεί να περιγραφεί πιο σωστά μέσω του οικονομικο-εμπορικού μοντέλου Business-to-Consumer ("B2C"), το οποίο σαν φαινόμενο τα τελευταία χρόνια αναπτύσσεται με ταχύτατους ρυθμούς. Τα ηλεκτρονικά καταστήματα (e-shops) αποτελούν μια διαδικτυακή 'πύλη' σε ένα φυσικό κατάστημα, δίνοντας την δυνατότητα στον επισκέπτη - πελάτη μέσω του ηλεκτρονικού του υπολογιστή ή του κινητού του τηλεφώνου να έρθει σε επαφή με τα προσφερόμενα προϊόντα ή υπηρεσίες της επιχείρησης μέσω ενός ψηφιακού περιβάλλοντος διεπαφής, να περιηγηθεί σε αυτά μέσω μιας σχετικής μηχανής αναζήτησης, να επιλέξει όποια προϊόντα τον ικανοποιούν και να αποπληρώσει την συναλλαγή με την επιχείρηση μέσω ενός συστήματος ηλεκτρονικών πληρωμών.

Για την προσέγγιση του ζητήματος του εκσυγχρονισμού μιας επιχείρησης με ήδη υφιστάμενο και εν λειτουργία πρόγραμμα μηχανογράφησης που εξυπηρετεί το φυσικό της κατάστημα, αποσκοπούμε μέσω αυτής της Πτυχιακής Εργασίας στον σχεδιασμό και την υλοποίηση ενός σύγχρονου ηλεκτρονικού καταστήματος με δυνατότητες εξυπηρέτησης όλων των μελλοντικών λειτουργικών και εμπορικών απαιτήσεων της επιχείρησης αυτής, παράλληλα με την εξυπηρέτηση των αντίστοιχων τρεχόντων. Εντός του παρόντος εγγράφου, αρχικά αναλύεται το ζήτημα το οποίο καλείται να επιλύσει η προτεινόμενη προσέγγιση της ΠΕ. Έπειτα, περιγράφεται συνοπτικά το θεωρητικό υπόβαθρο της πρότασης, καθώς επίσης και τα τεχνικά εργαλεία μέσω των οποίων αναπτύσσεται η προσέγγιση. Στη συνέχεια, αναλύεται η τεχνική υποδομή της προσέγγισης, καθώς και η διασύνδεσή της με την ήδη υπάρχουσα τεχνική υποδομή της ανάδοχης επιχείρησης. Επιπλέον, περιγράφεται το παραχθέν "προϊόν" της προσέγγισης, δηλαδή το ηλεκτρονικό κατάστημα όπως (θα) το βιώνει ο τελικός του χρήστης. Τέλος, συνοψίζονται τα όσα πραγματεύτηκε η ΠΕ αυτή, όπως επίσης και τις μελλοντικές προεκτάσεις της προσέγγισης που παρουσιάζεται στα πλαίσια των οφελών της επιχείρησης που (θα) εξυπηρετεί.

«E-shop creation, connected to electronic data processing software»

Paparisoulis Vaios

Abstract

As e-commerce we define the capability of the consumer to purchase goods or services via the Internet. In its most common form it can be described more accurately via the financial - business model "Business-to-Consumer (B2C)", which develops rapidly as a phenomenon the past recent years. E-shops constitute a web portal to a physical store, providing each prospective guest or client the capability to make contact with the provided goods or services of a business through a graphical user interface via their personal computer or mobile system, navigate through them via a search engine, chose whichever ones serve them better, and ultimately complete their purchase by paying for it through a digital payments system.

In approaching the issue of modernizing a business with an already acquired electronic data management software for their physical store, we aim through this Thesis and its subsequent deliverable to design and develop a contemporary, fully operational e-shop with capabilities of expansion in order to fulfil every future market or operational need of the business in hand, while concurrently optimally servicing its already existent needs. Within the document hereby, initially, we analyze the issue the proposed approach aims to solve. After so, the theoretical background of our approach is summarized, along with the technical tools through whom the approach is being developed. Furthermore, we describe the technical infrastructure of our approach, along with its interconnection with the current infrastructure of the concessionaire business. Moreover, the final "product" is described and visualized, i.e. the e-shop as it will be perceived and utilized by the end-user. Ultimately, in the final chapter we summarize everything this Thesis exerts, as well as all the future, potential work that can be done within the context of the benefits of the business the overall e-shop system infrastructure will serve.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον Καθηγητή μου κ. Τζέκη Παναγιώτη, αρχικά για την εμπιστοσύνη που μου έδειξε κατά την ανάθεση της πτυχιακής αυτής εργασίας, αλλά και για την συνεχή του στήριξη και εμπιστοσύνη σε όλο το διάστημα της εκπόνησής της.

Επιπλέον, θα ήθελα να ευχαριστήσω την οικογένειά μου και τους οικείους μου για την υποστήριξή τους καθ'όλη την διάρκεια του ακαδημαϊκού μου ταξιδιού, και το κουράγιο και θάρρος που μου έδωσαν για να συνεχίσω έναντι των δυσκολιών που παρουσιάστηκαν, και να το ολοκληρώσω επιτυχώς.

Περιεχόμενα

Πρόλογος	iii
Περίληψη	iv
Abstract	v
Ευχαριστίες	vi
Περιεχόμενα	vii
Κατάλογος Εικόνων	ix
Συντομογραφίες	x
1 Εισαγωγή	1
1.1 Εισαγωγή	1
1.2 Πεδίο Μελέτης	1
1.3 Διάρθρωση Κεφαλαίων	2
2 Θεωρητικό Μέρος	3
2.1 Εισαγωγή	3
2.2 Front-End	5
2.2.1 Γραφικό Περιβάλλον Διεπαφής	5
2.2.2 HTML5	6
2.2.3 JavaScript	6
2.2.4 CSS	6
2.3 Back-end	7
2.3.1 PHP	7
2.3.2 Laravel	8
2.3.3 Blade	8
2.3.4 Κρυπτογράφηση	9
2.3.5 SSL/TLS	9
2.4 Βάσεις Δεδομένων	10
2.5 Επίλογος	11
3 Σχεδιασμός και Υλοποίηση Ιστοσελίδας - Back-End	12
3.1 Εισαγωγή	12
3.2 Λειτουργικές Απαιτήσεις	12
3.3 Υλοποίηση	13
3.3.1 Κλάσεις και Αντικείμενα	14
3.3.2 Δημιουργία Λογαριασμού Χρήστη	18
3.3.3 Σύνδεση - Αποσύνδεση Λογαριασμού	18
3.3.4 Επαναφορά Κωδικού Πρόσβασης	23
3.3.5 Ενημέρωση Στοιχείων Λογαριασμού	26
3.3.6 Σελίδες Προϊόντων	28
3.3.7 Λίστα “Αγαπημένων”	29
3.3.8 Καλάθι Αγορών	30
3.3.9 Ηλεκτρονική Πληρωμή	32
3.3.10 Προϊόντα & Κατηγορίες	34
3.3.11 Παραγγελίες	44
3.3.12 Διαχείριση Λογαριασμών Χρηστών	46
3.4 Βάση Δεδομένων - Πρόγραμμα Μηχανογράφησης	48
3.4.1 Πρόγραμμα Μηχανογράφησης	51
3.4.2 Βάση Δεδομένων	55
3.5 Επίλογος	61

4	Σχεδιασμός και Υλοποίηση Ιστοσελίδας - Front-End	62
4.1	Εισαγωγή	62
4.2	Γραφικό Περιβάλλον Διεπαφής	62
4.3	Επίλογος	68
5	Σύνοψη	69
5.1	Εισαγωγή	69
5.2	Παραδοτέα Αποτελέσματα	69
5.3	Μελλοντικές Επεκτάσεις	69
5.4	Επίλογος	70
	ΒΙΒΛΙΟΓΡΑΦΙΑ	71
	Α □ ΚΩΔΙΚΑΣ ΣΕ PHP - LARAVEL	73
A□.1Π.A.1:	Class User	73
A□.2Π.A.2:	Class "Product"	74
A□.3Π.A.3:	Class "Category"	75
A□.4Π.A.4:	Class "Cart"	76
A.5 Π.A.5:	Class "Order"	76
A□.6Π.A.6:	Create "User"	77
A□.7Π.A.7:	Sign-in/Sign-out	78
A□.8Π.A.8:	Reset Password	79
A□.9Π.A.9:	Confirm New Password	80
A□.10Π.A.10:	Store New Password	80
A□.11Π.A.11:	Update User Info	81
A□.12Π.A.12:	Product Pages	82
A□.13Π.A.13:	"Favourites"	84
A□.14Π.A.14:	Show "Favourites"	85
A□.15Π.A.15:	Add to Cart	85
A□.16Π.A.16:	Update Cart	86
A□.17Π.A.17:	Show Cart	86
A□.18Π.A.18:	PayPal Integration	87
A□.19Π.A.19:	Add New Product	88
A□.20Π.A.20:	Edit Product	88
A□.21Π.A.21:	Show Product	90
A□.22Π.A.22:	Add New Category	90
A□.23Π.A.23:	Edit Category	91
A□.24Π.A.24:	Show Vategory	92
A□.25Π.A.25:	Add New Order	92
A□.26Π.A.26:	Edit Order	93
A□.27Π.A.27:	Show Order	94
A□.28Π.A.28:	Ban/Unban User	95
A□.29Π.A.29:	Show User Info	96

Κατάλογος Εικόνων

3.1	Διαγραμματική Αναπαράσταση Εγγραφής Χρήστη στο Σύστημα.	19
3.2	Διαγραμματική Αναπαράσταση Σύνδεσης Χρήστη στο Σύστημα.	21
3.3	Διαγραμματική Αναπαράσταση Αποσύνδεσης Χρήστη από το Σύστημα.	22
3.4	Διαγραμματική Αναπαράσταση Διαδικασίας Επαναφοράς Κωδικού Πρόσβασης.	25
3.5	Διαγραμματική Αναπαράσταση Διαδικασίας Ενημέρωσης Στοιχείων Χρήστη.	27
3.6	Διαγραμματική Αναπαράσταση Διαδικασίας Προσθήκης/Αφαίρεσης Προϊόντος από την Λίστα “Αγαπημένων”.	29
3.7	Διαγραμματική Αναπαράσταση Διαδικασίας Προθήκης Προϊόντος στο Καλάθι Αγορών.	31
3.8	Διαγραμματική Αναπαράσταση Διαδικασίας Ολοκλήρωσης Παραγγελίας - ”Checkout”	33
3.9	Διαγραμματική Αναπαράσταση Διαδικασίας Προσθήκης Νέου Προϊόντος στο Σύστημα	35
3.10	Διαγραμματική Αναπαράσταση Διαδικασίας Επεξεργασίας Προϊόντος στο Σύστημα	37
3.11	Διαγραμματική Αναπαράσταση Διαδικασίας Διαγραφής Προϊόντος στο Σύστημα	38
3.12	Διαγραμματική Αναπαράσταση Διαδικασίας Προσθήκης Νέας Κατηγορίας στο Σύστημα	40
3.13	Διαγραμματική Αναπαράσταση Διαδικασίας Τροποποίησης μιας Κατηγορίας στο Σύστημα	42
3.14	Διαγραμματική Αναπαράσταση Διαδικασίας Διαγραφής μιας Κατηγορίας στο Σύστημα	43
3.15	Διαγραμματική Αναπαράσταση Διαδικασίας Ενημέρωσης Κατάστασης μιας Παραγγελίας στο Σύστημα	45
3.16	Διαγραμματική Αναπαράσταση Διαδικασίας Αποκλεισμού/Επαναφοράς Χρήστη από το/στο Σύστημα	47
3.17	Διαγραμματική Αναπαράσταση της συνολικής Τεχνικής Υποδομής της Επιχείρησης	50
3.18	Διαγραμματική Αναπαράσταση Στιγμιότυπου Επικοινωνίας E-shop και Προγράμματος Μηχανογράφησης	50
3.19	Έκδοση τιμολογίου από το φυσικό κατάστημα - ταμειακή μηχανή, και προσθήκη στην Βάση Δεδομένων.	52
3.20	Καταχώρηση στοιχείων νέου πελάτη.	52
3.21	Καταχώρηση Νέου Προϊόντος - Βασικά Στοιχεία.	54
3.22	Καταχώρηση Νέου Προϊόντος - Στοιχεία προβολής στο e-shop	54
3.23	Βάση Δεδομένων - Απόθεμα Προϊόντων.	56
3.24	Πρόγραμμα Μηχανογράφησης - Διαχείριση και Προβολή Αποθέματος Προϊόντων.	56
3.25	Πρόγραμμα Μηχανογράφησης - Πελατολόγιο.	57
3.26	Βάση Δεδομένων - Παραγγελίες Πελατών.	58
3.27	Βάση Δεδομένων - Αγορές Πελατών.	58
3.28	Πρόγραμμα Μηχανογράφησης - Τιμολόγια αγορών προϊόντων.	59
3.29	Βάση Δεδομένων - Προμηθευτές.	60
4.1	Eshop: Αρχική Σελίδα	63
4.2	Eshop: Πλευρικό Μενού “Κατηγορίες” - “Δόμηση - Ξηρά Δόμηση”	64
4.3	Eshop: Πλευρικό Μενού “Κατηγορίες” - “Εργαλεία Κήπου - Αγρού”	64
4.4	Eshop: Σύνδεση Μέλους	65
4.5	Eshop: Εγγραφή Μέλους	66
4.6	Eshop: Σελίδα Προϊόντων	67
4.7	Eshop: Αναθεώρηση Καλαθιού Αγορών - Checkout	67
4.8	Eshop: Ολοκλήρωση και Αποστολή Παραγγελίας	68

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία

Κεφάλαιο 1ο: Εισαγωγή

1.1 Εισαγωγή

Στο πρώτο Κεφάλαιο αυτού του συγγράμματος προλογίζεται εν συντομία το ποιόν της πτυχιακής αυτής διατριβής. Αρχικά, περιγράφεται συνοπτικά η ανάγκη ύπαρξης ενός ηλεκτρονικού καταστήματος για την εξυπηρέτηση των αναγκών μιας σύγχρονης επιχείρησης και ο σχεδιασμός και η υλοποίησή του ως πεδίο μελέτης αυτής της εργασίας. Έπειτα, παρουσιάζονται τα βασικά στοιχεία της προτεινόμενης από εμάς προσέγγισης. Τέλος, αναλύεται η διάρθρωση των Κεφαλαίων, ώστε να παρέχεται μια συγκεντρωτική εικόνα του τι έπεται.

1.2 Πεδίο Μελέτης

Ως ηλεκτρονικό εμπόριο ορίζεται η δυνατότητα του καταναλωτή να αγοράζει αγαθά ή υπηρεσίες μέσω Διαδικτύου. Στην πιο κοινότυπη, συνήθη μορφή του μπορεί να περιγραφεί πιο σωστά μέσω του οικονομικο-εμπορικού μοντέλου Business-to-Consumer ("B2C") [1], το οποίο σαν φαινόμενο τα τελευταία χρόνια αναπτύσσεται με ταχύτατους ρυθμούς. Συγκεκριμένα, σύμφωνα με έρευνες, ενώ το 2019 το ηλεκτρονικό εμπόριο στην χώρα μας παρουσίαζε ανάπτυξη της τάξης του 7% [2], το 2021 ο αντίστοιχος ρυθμός ανάπτυξης ανήλθε στην τάξη του 77%. Αρωγός στην ανάπτυξη αυτή, πέραν των εξελίξεων στην παγκόσμια υγεία, αποτελεί και ανέκαθεν αποτελούσε και η ανάπτυξη του Διαδικτύου [3], και η ευχρηστία αυτού από τον μέσο καταναλωτή - χρήστη. Παράλληλα, αποτέλεσε κομβικό παράγοντα για την ανάπτυξη ηλεκτρονικών καταστημάτων από πλευράς των επιχειρήσεων, στις οποίες δίνεται έτσι η δυνατότητα να διευρύνουν το πελατολόγιό τους και τα κέρδη τους.

Τα προαναφερθέντα στατιστικά παρουσιάζονται επιλεκτικά έναντι απείρως αναλυτικότερων μελετών επί του αντικειμένου. Κοινό γνώμονα όλων αποτελεί η αδιάσειστη αναγκαιότητα ενός ηλεκτρονικού καταστήματος για μια επιχείρηση στην σύγχρονη αγορά, εφόσον αυτή επιθυμεί να παραμείνει ανταγωνιστική έναντι των υπολοίπων, αλλά σύγχρονη των χαρακτηριστικών μιας εποχής.

Αντικείμενο της παρούσας Πτυχιακής Εργασίας αποτελεί ο σχεδιασμός και η υλοποίηση ενός ηλεκτρονικού καταστήματος που θα πληροί πάσα μελλοντική εμπορική και λειτουργική ανάγκη της επιχείρησης, αλλά και θα εξυπηρετεί τις ήδη υφιστάμενες. Συγκεκριμένα, η προκειμένη ΠΕ εστιάζει στην ανάπτυξη ενός e-shop για μια ήδη υφιστάμενη εμπορική επιχείρηση, η οποία έχει ήδη στις υπηρεσίες της για αρκετά χρόνια ένα πρόγραμμα μηχανογράφησης που εξυπηρετεί το φυσικό κατάστημα.

Βαρύνουσα σημασία είναι η ενσωμάτωση της λειτουργικότητας του υφιστάμενου προγράμματος μηχανογράφησης του φυσικού καταστήματος στις λειτουργίες του ηλεκτρονικού, αλλά και η συμπερίληψη των απαιτήσεων του πρώτου κατά τον σχεδιασμό της τεχνικής υποδομής

και τον καθορισμό των λειτουργικών απαιτήσεων του δεύτερου.

1.3 Διάρθρωση Κεφαλαίων

Το παρόν έγγραφο είναι διαρθρωμένο σε πέντε Κεφάλαια. Στο Πρώτο, παρόν, Κεφάλαιο γίνεται η διατύπωση του ζητήματος του οποίου την επίλυση πραγματεύεται η προκείμενη εργασία. Στο Δεύτερο κεφάλαιο παρουσιάζεται αναλυτικά το θεωρητικό υπόβαθρο της νοοτροπίας σχεδιασμού ενός ηλεκτρονικού καταστήματος, καθώς επίσης και μια σύνοψη των δομικών εργαλείων για την ανάπτυξη αυτού. Στο Τρίτο Κεφάλαιο παρουσιάζεται ο σχεδιασμός και η υλοποίηση της συνολικής τεχνικής υποδομής του συστήματος: το back-end του ιστοτόπου του ηλεκτρονικού καταστήματος, η διασύνδεσή του με το πρόγραμμα μηχανογράφησης, οι δυνατότητες του τελευταίου και η Βάση Δεδομένων. Στο Τέταρτο Κεφάλαιο παρουσιάζεται το περιβάλλον γραφικής διεπαφής με το οποίο αλληλεπιδρά ο τελικός χρήστης της πλατφόρμας - πελάτης της επιχείρησης. Τέλος, στο Πέμπτο Κεφάλαιο εμπεριέχεται μια σύνοψη και ανακεφαλαίωση των προαναφερθέντων, αλλά και μία ανάλυση των βραχυπρόθεσμων, μεσοπρόθεσμων και μακροπρόθεσμων επιθυμητών μελλοντικών επεκτάσεων.

Κεφάλαιο 2ο: Θεωρητικό Μέρος

2.1 Εισαγωγή

Τα ηλεκτρονικά καταστήματα (e-shops) αποτελούν μια διαδικτυακή “πύλη” σε ένα φυσικό κατάστημα, δίνοντας την δυνατότητα στον επισκέπτη - πελάτη μέσω του ηλεκτρονικού του υπολογιστή ή του κινητού του τηλεφώνου να έρθει σε επαφή με τα προσφερόμενα προϊόντα ή υπηρεσίες της επιχείρησης μέσω ενός ψηφιακού περιβάλλοντος διεπαφής, να περιηγηθεί σε αυτά μέσω μιας σχετικής μηχανής αναζήτησης, να επιλέξει όποια προϊόντα τον ικανοποιούν και να αποπληρώσει την συναλλαγή με την επιχείρηση μέσω ενός συστήματος ηλεκτρονικών πληρωμών.

Ο σχεδιασμός και η υλοποίηση ενός ηλεκτρονικού καταστήματος πρέπει να πληρεί κάποια τεχνικά κριτήρια και απαιτήσεις, προκειμένου να διασφαλίζεται η αποτελεσματική λειτουργία του - φυσικού και μη - καταστήματος, αλλά και η ορθή και ασφαλής εξυπηρέτηση του πελάτη/χρήστη. Αυτά τα κριτήρια προκύπτουν βάσει του προϊόντος ή της υπηρεσίας που προσφέρεται, του μεγέθους της επιχείρησης, του αγοραστικού κοινού στο οποίο απευθύνεται, καθώς και το εύρος της αγοράς στην οποία προσανατολίζεται [4].

Αρχικά, με βασικό γνώμονα την ποιότητα των προσφερόμενων υπηρεσιών προς τον τελικό χρήστη, ιδιαίτερα σημαντική αποτελεί η παρουσίαση των λεπτομερειών του εκάστοτε παρεχόμενου προϊόντος ή υπηρεσίας από το κατάστημα, προκειμένου ο χρήστης να έχει όλα τα απαραίτητα στοιχεία στην διάθεσή του ώστε να προχωρήσει με σιγουριά στην αγορά αυτών. Ένα σωστά και έξυπνα σχεδιασμένο γραφικό περιβάλλον της ιστοσελίδας του καταστήματος αποβαίνει συχνά επαρκές στην εξυπηρέτηση αυτής της ανάγκης. Κομβική αποτελεί η ισορροπία μεταξύ απλότητας στον τρόπο παροχής της πληροφορίας προς τον τελικό χρήστη και πληρότητας των παρεχόμενων πληροφοριών, ούτως ώστε η περιήγησή του στον ιστότοπο να μην αποτελεί μια κουραστική και χαοτική διαδικασία.

Η ομαλή ολοκλήρωση της αγοράς των επιλεγμένων πραγματοποιείται μέσω της ενότητας, συχνά αποκαλούμενης ως “Καλάθι Αγορών”, όπου ο πελάτης έχει μια συνολική, εποπτική εικόνα των μέχρι τώρα εν δυνάμει αγορών του. Πριν την οριστικοποίηση της αγοράς είναι σημαντικό να δίνεται η δυνατότητα στον χρήστη προσθαφαίρεσης προϊόντων από το “καλάθι”. Επιπλέον, κατόπιν ολοκλήρωσης και αποστολής της παραγγελίας, είναι συνετό να έχει προβλεφθεί κατά τον σχεδιασμό του ηλεκτρονικού καταστήματος η δυνατότητα διόρθωσης ή ολικής ακύρωσης μιας παραγγελίας εντός κάποιου λογικού χρονικού περιθωρίου. Υψίστης σημασίας αποτελεί και η προσφορά ευέλικτων τρόπων πληρωμής (π.χ. μέσω αντικαταβολής, τραπεζικού εμβάσματος, χρήση πιστωτικής/χρεωστικής κάρτας, PayPal - Viva Wallet ή άλλων συναφών ψηφιακών υπηρεσιών, κλπ.), που να εξυπηρετεί τόσο τον πελάτη, όσο και την επιχείρηση σε οποιοδήποτε σενάριο συναλλαγής (π.χ. ακύρωση).

Από πλευράς λειτουργικού οφέλους, ένα ηλεκτρονικό κατάστημα με σωστή υποδομή διαχείρισης πληροφοριών μπορεί να βοηθήσει στην βελτιστοποίηση της οικονομικής οργάνωσης και των παρεχόμενων υπηρεσιών της επιχείρησης, ενώ παράλληλα επωφελείται και ο πελάτης. Μέσω της επιλογής δημιουργίας ατομικού λογαριασμού χρήστη και καταχωρώντας κάποιες προσωπικές πληροφορίες του, δίνεται η δυνατότητα στον πελάτη να αντιμετωπίζεται διαφορετικά από το σύστημα σε σχέση με έναν απλό επισκέπτη, είτε αυτό σημαίνει μια πιο εξατομικευμένη εμπειρία χρήσης του ιστοτόπου σε επαναλαμβανόμενες επισκέψεις, είτε πιο ευνοϊκοί όροι τιμολόγησης των προϊόντων ή εξυπηρέτησής του. Παράλληλα, με την δημιουργία ενός αρχείου παραγγελιών, δίνεται η δυνατότητα στην επιχείρηση να οργανώνει αποδοτικότερα, και βάσει κάποιας προτεραιότητας, της επερχόμενες, ή ακόμα και μελλοντικές, παραγγελίες και διανομές προϊόντων προς τους πελάτες της.

Τέλος, θεμελιώδης ανάγκη αποτελεί ο σχεδιασμός ενός ψηφιακού συστήματος επικοινωνίας στην πλατφόρμα μεταξύ πελάτη και επιχείρησης. Αυτός ο διάυλος επικοινωνίας μπορεί να υλοποιηθεί είτε μονόδρομα, μέσω αυτοματοποιημένων email από την επιχείρηση προς τον πελάτη, συμπληρωματικά με τις παραδοσιακές μορφές επικοινωνίας (τηλέφωνο, φαξ, κλπ.), είτε αμφίδρομα, μέσω μιας “φόρμας επικοινωνίας” η οποία καθιστά την επικοινωνία μεταξύ των δύο πλευρών πιο αυτοματοποιημένη, οργανωμένη και διαχειρίσιμη σε σχέση με την κλασική ανταλλαγή email.

Βάσει των προαναφερθέντων, καθίσταται πλέον σαφής η αναγκαιότητα μιας ορθώς σχεδιασμένης βάσης δεδομένων, ενός κομψού γραφικού περιβάλλοντος, ενός αποδοτικού συστήματος μηχανογράφησης και εύρωστων πρωτοκόλλων ασφαλείας που αφορούν την επικοινωνία, την ψηφιακή ασφάλεια των διακινούμενων πληροφοριών και την προστασία ευαίσθητων δεδομένων.

2.2 Front-End

Με τον όρο front-end περιγράφουμε το τμήμα μιας ιστοσελίδας με το οποίο αλληλεπιδρά ο επισκέπτης/χρήστης. Συχνά στην βιβλιογραφία γίνεται λόγος για το τμήμα μιας εφαρμογής “από πλευράς του client” (“client-side”). Ο όρος “client” δεν σχετίζεται σε αυτή την περίπτωση με το marketing, υπονοώντας τον “πελάτη” μιας επιχείρησης, αλλά τον παραλήπτη πληροφορίας σε ένα πληροφοριακό σύστημα αρχιτεκτονικής “πελάτη - εξυπηρετητή” (“client - server”).

Το front-end περιλαμβάνει όλα τα στοιχεία μιας ιστοσελίδας με τα οποία ο χρήστης έρχεται σε επαφή, όπως για παράδειγμα το κείμενο, τα θέματα, τα στυλ, τις εικόνες, τα γραφήματα και τους πίνακες, πλήκτρα, χρώματα, μενού πλοήγησης, κλπ. Οι πλέον διαδεδομένες γλώσσες προγραμματισμού οι οποίες συνδέονται με την ανάπτυξη του front-end είναι η HTML, η CSS και η JavaScript. Η δομή, ο σχεδιασμός, η συμπεριφορά και το περιεχόμενο μιας ιστοσελίδας, μιας διαδικτυακής εφαρμογής, ή μιας εφαρμογής για κινητά αποτελούν κομμάτι της ανάπτυξης front-end λογισμικού, με βασικό γνώμονα την δυνατότητα απόκρισης, δηλαδή την δυνατότητα μιας εφαρμογής ή ιστοσελίδας να προσαρμόζεται και να λειτουργεί σωστά σε οποιαδήποτε, σταθερή ή κινητή, συσκευή και οθόνη, και τις επιδόσεις λειτουργίας, δηλαδή η ανταπόκρισή της στα αιτήματα του χρήστη να είναι όσο το δυνατόν ταχύτερη.

Στις επερχόμενες παραγράφους αναλύεται περαιτέρω το τι εστί Γραφικό Περιβάλλον Διεπαφής με το οποίο αλληλεπιδρά ένας χρήστης μιας ιστοσελίδας, όπως επίσης περιγράφονται σε ένα βαθμό τα χαρακτηριστικά των γλωσσών προγραμματισμού που προαναφέρθηκαν και σχετίζονται με την ανάπτυξη του front-end.

2.2.1 Γραφικό Περιβάλλον Διεπαφής

Με τον όρο “(Γραφικό) Περιβάλλον Διεπαφής ((Graphical) User Interface - (G)UI)” περιγράφεται ένα σύνολο οπτικών στοιχείων (ψηφιακές οθόνες, παράθυρα, σελίδες, κουμπιά, εικονίδια, κλπ.) τα οποία χρησιμοποιεί κανείς για να αλληλεπιδράσει με μια ηλεκτρονική ή ψηφιακή συσκευή [5]. Με την πάροδο του χρόνου και την εξέλιξη της γενικότερης φιλοσοφίας και διαδικασίας του design, ο όρος UI έφτασε να είναι άρρηκτα συνδεδεμένος με την “Εμπειρία Χρήστη (User Experience - UX)”, και ταυτόχρονα να διαφέρει από αυτόν. Κατά τον σχεδιασμό ή την βελτίωση ενός UI δίνεται βαρύτητα στις αντιδράσεις του χρήστη κατά την αλληλεπίδρασή του με το περιβάλλον διεπαφής. Σκοπός πλέον είναι ο σχεδιασμός και η υλοποίηση, όχι απλώς ενός καλαίσθητου γραφικού περιβάλλοντος, αλλά ενός λειτουργικού, προσβάσιμου περιβάλλοντος διεπαφής το οποίο καλύπτει συγκεκριμένες ανάγκες του εκάστοτε χρήστη του.

Ο σχεδιασμός και το “στήσιμο” των σύγχρονων e-shops τα τελευταία χρόνια ακολουθεί κάποιες νόρμες και τάσεις. Γεωμετρικά σχήματα και συμμετρικά σχέδια σε πιο οργανικές παλέτες ή μοτίβα που τείνουν προς τις αποχρώσεις του προϊόντος αποτελούν τις συνηθέστερες τακτικές για την υλοποίηση των βασικών γραφικών στοιχείων. Εξίσου δημοφιλείς λύσεις αποτελούν τα ολοσέλιδα backgrounds και οι μεγάλες γραμματοσειρές με έντονες χρωματικές αντιθέσεις

σε σχέση με το φόντο. Η ψηφιακή αναπαράσταση των προϊόντων αυτών καθ' αυτών δύναται να υλοποιηθεί πολύ δημιουργικά μέσω δυναμικά μεταβαλλόμενων, αλληλεπιδράσιμων γραφικών απεικονίσεων αυτών. Τέλος, προς εξοικονόμηση - ή καλύτερη αξιοποίηση - χώρου στον ιστότοπο, η υλοποίηση κρυφών ("hidden") ή αναδύομενων ("pop-up") μενού αντικατέστησε τις κλασικές υλοποιήσεις των αντίστοιχων στατικών κάθετων, οριζόντιων και drop down menus.

Τα προαναφερθέντα, τα οποία ως σύνολο αποτελούν το λεγόμενο "front-end" του ιστοτόπου και αποτελεί το μέρος αυτού με το οποίο έρχεται σε επαφή ο τελικός χρήστης, υλοποιούνται προγραμματιστικά μέσω των γλωσσών προγραμματισμού HTML5 (HyperText Markup Language), JS (JavaScript) και CSS3 (Cascade Style Sheets).

2.2.2 HTML5

Η HTML αποτελεί μια γλώσσα σήμανσης για έγγραφα ή αρχεία σχεδιασμένα να προβληθούν σε έναν φυλλομετρητή ο οποίος, όταν τα παραλαμβάνει κατόπιν αιτήματος σε έναν διακομιστή (server), τα μετατρέπει σε πολυμεσικές ιστοσελίδες. Λανσαρίστηκε πρώτη φορά το 1993 [6] και έκτοτε συνεχώς βελτιώνεται. Σήμερα, στην πλέον σύγχρονη έκδοσή της [7, 8], έχει την δυνατότητα να ενσωματώνει στοιχεία τα οποία έχουν υλοποιηθεί σε JS και CSS. Σε συνδυασμό με την PHP ή την ASP (Active Server Pages) - ASP.NET, οι οποίες αποτελούν γλώσσες προγραμματισμού "server-side" για την υλοποίηση και διαχείρισή του - αθέατου προς τον τελικό χρήστη - λεγόμενου "back-end" ενός ιστοτόπου, η HTML απαρτίζει τα βασικά εργαλεία για την ανάπτυξη διαδικτυακού λογισμικού, και κατ' επέκταση, ιστοσελίδων.

2.2.3 JavaScript

Η JavaScript αποτελεί μια διερμηνευμένη γλώσσα προγραμματισμού που αντιμετωπίζει τις συναρτήσεις ως αντικείμενα πρώτης τάξης (δηλ. το ίδιο με τις λοιπές μεταβλητές). Αρχικός της σκοπός ήταν η υλοποίηση φυλλομετρητών και σταδιακά εξελίχθηκε και επέκτεινε την χρησιμότητά της και στην γενικότερη ανάπτυξη εφαρμογών (ιστού). Βασίζεται σε πρωτότυπα, είναι δυναμική και υποστηρίζει αντικειμενοστραφή, προστακτικό και συναρτησιακό προγραμματισμό [9].

2.2.4 CSS

Η CSS είναι μια γλώσσα προγραμματισμού για την περιγραφή των στοιχείων που απαρτίζουν τις ιστοσελίδες, όπως για παράδειγμα, τα χρώματα, τις γραμματοσειρές και την γενικότερη διάταξη αυτών. Επιτρέπει στους προγραμματιστές να προσαρμόσουν την εμφάνιση της ιστοσελίδας σε διάφορα φυσικά μέσα (οθόνες υπολογιστών μεγάλου μεγέθους, οθόνες κινητών, εκτυπωτών, φορητών συσκευών, κλπ.) χωρίς να τροποποιούν την δομή τους. Είναι ανεξάρτητη από την HTML και μπορεί να χρησιμοποιηθεί όπως κάθε άλλη γλώσσα σήμανσης. Η ανεξαρτησία της αυτή διευκολύνει σημαντικά την συντήρηση των ιστοτόπων, την αναπαραγωγή της διάταξης

μιας ιστοσελίδας και σε άλλες και την προσαρμογή μιας ιστοσελίδας σε διάφορα περιβάλλοντα (“διαχωρισμός περιεχομένου από την παρουσίαση”) [10, 11].

2.3 Back-end

Με τον όρο Back-end αναφερόμαστε στο τμήμα μιας ιστοσελίδας το οποίο σχετίζεται με το κομμάτι της υλοποίησης από πλευράς εξυπηρετητή (“server”). Το τμήμα αυτό είναι υπεύθυνο για την αποθήκευση και την οργάνωση των δεδομένων της εκάστοτε ιστοσελίδας ή εφαρμογής, όπως επίσης για την γενικότερη ορθή λειτουργία και συμπεριφορά του front-end.

Το τμήμα αυτό δεν είναι - και δεν πρέπει να είναι - προσβάσιμο και αλληλεπιδράσιμο απ’ ευθείας από τον χρήστη μιας ιστοσελίδας. Τα μέρη και τα χαρακτηριστικά μιας ιστοσελίδας που αναπτύσσονται στο back-end είναι προσβάσιμα στον τελικό χρήστη μόνο μέσω του front-end και των σχετικών εφαρμογών. Το κομμάτι της διασφάλισης της ασφάλειας αυτής αποτελεί ένα εκ των κυρίων μελημάτων της ανάπτυξης back-end λογισμικού. Για τον σκοπό αυτό, η πληροφορία που ανταλλάσσεται μεταξύ front-end - back-end συχνά κρυπτογραφείται.

Οι πλέον δημοφιλείς γλώσσες προγραμματισμού οι οποίες συνδέονται στενά με την ανάπτυξη του back-end ιστοσελίδων είναι η PHP και η Node.js. Η ανάπτυξη και η διαχείριση όλων των τμημάτων του back-end μπορεί συχνά να αποδειχθεί μια πολύπλοκη διαδικασία για να διενεργηθεί καθαρά μέσω μιας γλώσσας προγραμματισμού, ιδιαίτερα όσο αυξάνονται οι απαιτήσεις και το μέγεθος μιας εφαρμογής ή ιστοσελίδας. Για το σκοπό αυτό χρησιμοποιούνται συχνά συγκεκριμένα frameworks. Με τον όρο “framework” περιγράφουμε ένα σαφώς οργανωμένο σύνολο βιβλιοθηκών, εργαλείων, πόρων, και λοιπών τμημάτων λογισμικού τα οποία βοηθούν στην ανάπτυξη εφαρμογών μέσω μιας ήδη υπάρχουσας γλώσσας προγραμματισμού. Παραδείγματος χάριν, το Laravel αποτελεί ένα PHP back-end framework.

Στις επερχόμενες παραγράφους περιγράφονται αναλυτικότερα τα χαρακτηριστικά των προαναφερθέντων γλωσσών προγραμματισμού και συγκεκριμένων πρωτοκόλλων ασφαλείας που συνδέονται με την κρυπτογράφηση πληροφοριών και των τρόπο ανταλλαγής τους.

2.3.1 PHP

Η PHP [12] (“Hypertext PreProcessor”) είναι μια ευρέως διαδεδομένη και χρησιμοποιούμενη γλώσσα προγραμματισμού γενικού σκοπού, που είναι ιδιαίτερα ενδεδειγμένη για ανάπτυξη του back-end διαδικτυακών εφαρμογών και ιστοσελίδων, καθώς μπορεί πολύ εύκολα να ενσωματώσει κώδικα για ένα front-end σχεδιασμένο σε HTML. Σε αντίθεση με άλλες γλώσσες προγραμματισμού που δύνανται να αναπτύξουν back-end υλοποιήσεις με πολλαπλές εντολές, όπως π.χ. η C++ ή η Java, η HTML δύναται να συμπεριλάβει στην ιστοσελίδα τμήματα PHP κώδικα με συγκεκριμένη σημειολογία, και να τα εκτελέσει απ’ ευθείας, χωρίς ανάγκη κλήσης αιτήματος.

Αυτό που διαφοροποιεί την PHP σε σχέση με άλλες γλώσσες προγραμματισμού από πλευράς client, όπως π.χ η JavaScript, είναι ότι ο κώδικας εκτελείται στον server και δημιουργεί HTML κώδικα ο οποίος αποστέλλεται έπειτα στον client. Ο client θα λάβει τα αποτελέσματα του κώδικα που εκτελέστηκε, χωρίς να γνωρίζει τον πηγαίο κώδικα που τα δημιούργησε. Με επαρκή ρύθμιση και σωστή διάρθρωση του HTML κώδικα, καθίσταται πρακτικά αδύνατο να αναγνωριστεί ο τρόπος με τον οποίον παράγονται τα τελικά οπτικά αποτελέσματα μιας ιστοσελίδας, παρέχοντας κατ' αυτόν τον τρόπο υψηλά επίπεδα ασφάλειας.

Εκ των προαναφερθέντων συμπεραίνουμε πως η PHP είναι πλέον μία από τις πλέον δημοφιλείς γλώσσες προγραμματισμού, χρησιμοποιούμενη τόσο για την υλοποίηση έργων ανοικτού κώδικα, όσο και για την ανάπτυξη διαδικτυακού λογισμικού σε επαγγελματικό επίπεδο, με τουλάχιστον το 80% των λειτουργικών ιστοτόπων του κόσμου να έχει αναπτυχθεί σε αυτή [13, 14].

2.3.2 Laravel

Το Laravel [15] είναι ένα σχετικά καινούριο PHP back-end development framework ανοικτού κώδικα. Βασισμένο στη σχεδιαστική λογική MVC [16] (Model - View - Controller), και επαναχρησιμοποιώντας υπάρχοντα τμήματα διαφορετικών λογισμικών και βιβλιοθηκών, βοηθά στην κατασκευή ορθώς και εύληπτα δομημένων ιστοσελίδων.

Το Laravel αρχικά κυκλοφόρησε το 2011 σαν λογισμικό ανοικτού κώδικα υπό άδεια MIT, και παρέχει στους χρήστες τις εξής δυνατότητες:

- Περιβάλλον γραμμής εντολών.
- Πρότυπα Blade για την εύκολη ανάπτυξη προτύπων front-end.
- Υποστήριξη κώδικα PHP.
- Σύστημα μεταφοράς βάσεων δεδομένων (Database Migration System).
- Δυνατότητες επεκτασιμότητας διαδικτυακών εφαρμογών και ιστοσελίδων (scalability).
- Αναλυτική επίσημη τεκμηρίωση και μεγάλη κοινότητα χρηστών.

Λόγω των δυνατοτήτων του και της απλότητάς του, αποτελεί ένα από τα πλέον δημοφιλή frameworks για ανάπτυξη εφαρμογών σε PHP.

2.3.3 Blade

Η Blade¹ [17] είναι μια απλή, γρήγορη γλώσσα προγραμματισμού η οποία δίνει την δυνατότητα στον χρήστη να αναπτύσσει εφαρμογές γρήγορα. Για αυτό το λόγο, η Blade εστιάζει περισσότε-

¹<https://github.com/blade-lang/blade>

ρο στην αλγοριθμική δομή του κώδικα, παρά το συντακτικό αυτού, διαθέτοντας ένα πολύ μικρό μεν, αλλά ισχυρό σετ εντολών.

Το Laravel framework δίνει δυνατότητες ενσωμάτωσης κώδικα Blade στην υποδομή του, επιτρέποντας ουσιαστικά τον γρήγορο σχεδιασμό, ανάπτυξη και εκτέλεση τμημάτων κώδικα στο PHP back-end μιας διαδικτυακής εφαρμογής ή ιστοσελίδας.

2.3.4 Κρυπτογράφηση

Για την διασφάλιση της ορθής λειτουργίας ενός ιστοτόπου ή μιας διαδικτυακής εφαρμογής, υψίστης σημασίας αποτελεί η ασφάλεια των δεδομένων και των πληροφοριών που διακινούνται μέσω αυτής. Προσωπικά δεδομένα χρηστών, διαπιστευτήρια σύνδεσης λογαριασμών, μεταφορά δεδομένων μεταξύ client και server, ηλεκτρονικές τραπεζικές συναλλαγές, επικοινωνία πελάτη και επιχείρησης δια ηλεκτρονικής αλληλογραφίας είναι μόνο μερικά από τα παραδείγματα συχνών περιπτώσεων που ενέχουν ρίσκο απώλειας, παραποίησης ή υποκλοπής πληροφοριών. Για αυτό τον σκοπό, τέτοιες “συναλλαγές” πληροφορίας κωδικοποιούνται μέσω πρωτοκόλλων κρυπτογράφησης, με τα συνηθέστερα να είναι τα SSL, TLS και HTTPS.

2.3.5 SSL/TLS

Ένα πιστοποιητικό SSL (Secure Sockets Layer) είναι ένα ψηφιακό έγγραφο το οποίο συνδέει την ταυτότητα ενός ιστοτόπου με ένα κλειδί κρυπτογράφησης δύο μερών, του δημόσιου κλειδιού και του ιδιωτικού. Μέσω του δημόσιου κλειδιού, ένας φυλλομετρητής εκκινεί μια κρυπτογραφημένη ανταλλαγή μηνυμάτων με έναν εξυπηρετητή (server). Το ιδιωτικό κλειδί φυλάσσεται στον εξυπηρετητή αυτόν και χρησιμοποιείται για την ψηφιακή υπογραφή σελίδων και άλλων ψηφιακών αρχείων [18, 19].

Το 1999 η αυτούσια χρήση του πρωτοκόλλου SSL καταργείται, καθώς αυτό αντικαθίσταται από το βελτιωμένο πρωτόκολλο TLS [20, 21]. Σήμερα, αναφερόμαστε ενιαία σε αυτά τα πρωτόκολλα ως SSL/TSL [22].

Το ενιαίο πρωτόκολλο SSL/TSL ενσωματώνεται σε μία εφαρμογή ή σε έναν ιστότοπο για την προστασία δεδομένων που αποστέλλονται μεταξύ client και server μέσω HTTP (Hyper Text Transfer Protocol). Πρώτο βήμα στην διαδικασία αυτή είναι η αποστολή ενός αναγνωριστικού μηνύματος από τον client στον server. Αφού ο server επιβεβαιώσει την ταυτότητα του client και επικυρώσει της επιμέρους παραμέτρους επικοινωνίας, επιστρέφει στον client το μήνυμα που του έστειλε αρχικά, μαζί με το ψηφιακό πιστοποιητικό του, το οποίο περιλαμβάνει πληροφορίες για το δημόσιο κλειδί του server, την εγκυρότητα του πιστοποιητικού, την ημερομηνία λήξης του, τον ιδιοκτήτη του και την υπεύθυνη Αρχή έκδοσης του πιστοποιητικού αυτού. Μόλις και ο client επικυρώσει και εξουσιοδοτήσει τον server, οι δύο πλευρές δημιουργούν συμμετρικά κλειδιά κρυπτογράφησης - αποκρυπτογράφησης μηνυμάτων και επικύρωσης αυτών.

Το πρωτόκολλο SSL/TSL αποτελείται από τέσσερα υποπρωτόκολλα, συγκεκριμένα τα Handshake, ChangeCipherSpec (CCS), Record, Alert. Το πρώτο χρησιμοποιείται από τον client κατά την επικύρωση του server με χρήση του δημοσίου κλειδιού του. Το δεύτερο χρησιμοποιείται κατά την “διαπραγμάτευση” των επιμέρους παραμέτρων εγκαθίδρυσης των όρων επικοινωνίας μεταξύ server-client, και πιο συγκεκριμένα, κατά την ανταλλαγή κρυπτογραφημάτων και αλγορίθμων κατά την διαδικασία αυτή. Το τρίτο υποπρωτόκολλο είναι υπεύθυνο για την προστασία των δεδομένων κατά την συνεδρία επικοινωνίας του πρώτου υποπρωτοκόλλου μέσω των κωδικών - κλειδιών που δημιουργούνται τότε. Το τελευταίο υποπρωτόκολλο καλείται όταν παρουσιαστεί οποιοδήποτε σφάλμα κατά την λειτουργία των υπολοίπων [23].

2.4 Βάσεις Δεδομένων

Για την αποθήκευση των δεδομένων των χρηστών, του αποθέματος και των παραγγελιών, αποτελεί επιτακτική ανάγκη ο σχεδιασμός και η δημιουργία μιας σχεσιακής βάσης δεδομένων. Μια σχεσιακή βάση δεδομένων αποτελεί μια συλλογή δεδομένων με προκαθορισμένες συσχετίσεις μεταξύ τους, οργανωμένα σε “πίνακες” στηλών και γραμμών. Στις γραμμές του πίνακα αποθηκεύονται οι διάφορες εγγραφές των αντικειμένων πληροφορίας του ομοίου τύπου που απαρτίζουν τον πίνακα. Κάθε στήλη του πίνακα απεικονίζει ένα διαφορετικό χαρακτηριστικό εκ του συνόλου των οποίων περιγράφουν το εκάστοτε αποθηκευμένο αντικείμενο - εγγραφή. Κάθε εγγραφή διαφοροποιείται από τις υπόλοιπες με ένα κοινό, αλλά διαφορετικά διατιμημένο, χαρακτηριστικό. Αυτό το χαρακτηριστικό ονομάζεται “πρωτεύον κλειδί (primary key)” του πίνακα. Δεδομένα που συσχετίζονται μεταξύ τους, αλλά περιγράφονται από διαφορετικούς πίνακες, μπορούν να αλληλοσυσχετιστούν στην ίδια βάση δεδομένων με ένα κοινό χαρακτηριστικό, το λεγόμενο “ξένο κλειδί (foreign key)”. Κατ’ αυτόν τον τρόπο, δίνεται η δυνατότητα στον χρήστη της βάσης δεδομένων να έχει πρόσβαση στα δεδομένα αυτής με διάφορους τρόπους και μορφές αυτών, χωρίς να μεταβάλλει την μορφή της [24].

Αν και μπορεί να παρουσιάσει κάποιες διαφοροποιήσεις ανάλογα με την αρχιτεκτονική του γενικότερου συστήματος, η διαχείριση μιας βάσης δεδομένων από μια ιστοσελίδα ή μια διαδικτυακή εφαρμογή γίνεται κατά βάση μέσω PHP και εμπίπτει στο κομμάτι του back-end development. Συνήθη συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων (Relational DataBase Management System - RDBMS) αποτελούν τα SQL Server, MySQL Server, Microsoft SQL Server, OracleDB, IBM DB2, PostgreSQL, κλπ., και χρησιμοποιούν τις SQL (Structured Query Language) ή MySQL (ή παραλλαγές αυτών) ως γλώσσες προγραμματισμού για αυτό τον σκοπό. Τα προαναφερθέντα συστήματα είναι ευρέως διαδεδομένα και αποτελούν πολύ αποδοτικές λύσεις στο ζήτημα της αποθήκευσης και διαχείρισης μεγάλου όγκου δεδομένων [25].

2.5 Επίλογος

Στο παρόν Κεφάλαιο αναλύθηκε σε ένα βαθμό το επιχειρηματικό υπόβαθρο στο οποίο βασίζεται και αιτιολογείται η δημιουργία ενός e-shop από επιχείριση, και η αρωγή του στον σύγχρονο εμπορικό κόσμο. Επιπλέον, διερευνήθηκαν τα τεχνολογικά εργαλεία λογισμικού τα οποία επιτρέπουν την ανάπτυξη ιστοτόπων, και δη ηλεκτρονικών καταστημάτων. Επ' αυτών βασίζεται και η υλοποίηση του e-shop που πραγματεύεται και η παρούσα πτυχιακή εργασία.

Κεφάλαιο 3ο: Σχεδιασμός και Υλοποίηση Ιστοσελίδας - Back-End

3.1 Εισαγωγή

Το αντικείμενο το οποίο πραγματεύεται η συγκεκριμένη πτυχιακή εργασία είναι ο σχεδιασμός και η υλοποίηση ενός ηλεκτρονικού καταστήματος για υφιστάμενη επιχείρηση. Στο παρόν Κεφάλαιο αναλύεται η δομή του ιστοτόπου του ηλεκτρονικού καταστήματος, οι λειτουργικές απαιτήσεις του σαν λογισμικό, οι μέθοδοι που υλοποιήθηκαν και ο τρόπος με τον οποίον πληρούν τις απαιτήσεις, και οι έλεγχοι που διενεργήθηκαν για την επικύρωση της ορθότητας λειτουργίας της ιστοσελίδας.

3.2 Λειτουργικές Απαιτήσεις

Ο άξονας επί του οποίου βασίζεται η δομή του ιστοτόπου είναι ότι υπάρχουν τρεις κατηγορίες χρηστών που έρχονται σε επαφή με τον ιστότοπο: Μη συνδεδεμένοι πελάτες ("Guest"), συνδεδεμένοι πελάτες ("User") και διαχειριστές της ιστοσελίδας ("Admin"). Βάσει αυτής της λογικής, ορίζονται οι εξής λειτουργικές απαιτήσεις για κάθε κατηγορία χρηστών:

- Guest:

Ένας μη συνδεδεμένος πελάτης πρέπει να:

- δύναται να δημιουργήσει λογαριασμό στο ηλεκτρονικό κατάστημα προκειμένου να συνδεθεί.
- έχει τη δυνατότητα να συνδέεται στον λογαριασμό του στον ιστότοπο.
- δύναται να επαναφέρει τον κωδικό πρόσβασης του λογαριασμού του.
- δύναται να έχει πρόσβαση στην αρχική σελίδα του ηλεκτρονικού καταστήματος.
- δύναται να έχει πρόσβαση στις σελίδες των προϊόντων του καταστήματος.

- User:

Ένας συνδεδεμένος πελάτης πρέπει να:

- Ένας συνδεδεμένος χρήστης πρέπει να δύναται να προσθέτει και να αφαιρεί προϊόντα από την λίστα "Αγαπημένων" του.
- έχει πρόσβαση στη λίστα "Αγαπημένων" του.
- δύναται να προσθέτει και να αφαιρεί προϊόντα από το καλάθι αγορών του.
- έχει πρόσβαση στο καλάθι αγορών του.
- δύναται να ενημερώνει την ποσότητα κάθε προϊόντος στο καλάθι του.
- δύναται να αποστέλλει την τελική παραγγελία του.

- εάν δεν είναι αποκλεισμένος από το σύστημα, να δύναται, αφού αποστείλει την τελική παραγγελία του, να την αποπληρώσει ηλεκτρονικά.
- Admin:
 - Ο διαχειριστής του συστήματος πρέπει να δύναται:
 - να δημιουργήσει, να προσπελάσει, να ενημερώσει και να διαγράψει μια κατηγορία προϊόντων.
 - να καταχωρήσει, να προσπελάσει, να ενημερώσει και να διαγράψει ένα προϊόν σε/από μια κατηγορία.
 - να προσπελάσει όλες τις καταχωρημένες παραγγελίες προϊόντων στο σύστημα.
 - να ενημερώνει την κατάσταση μιας παραγγελίας κατά τους εξής τρόπους:
 - Εάν μια παραγγελία:
 - * είναι σε “αναμονή”, δύναται να “αποσταλεί” ή να “ακυρωθεί”.
 - * “αποσταλεί”, μπορεί να “παραδοθεί” ή να “ακυρωθεί”.
 - * “παραδοθεί”, μπορεί να “ακυρωθεί” εντός δεκατεσσάρων ημερών από την ημέρα που “παραδόθηκε”.
 - * “ακυρωθεί”, τα προϊόντα που δέσμευσε πρέπει να επανέρχονται στο stock της αποθήκης.
 - να προσπελάσει όλους τους υφιστάμενους λογαριασμούς χρηστών.
 - να αποκλείει έναν λογαριασμό χρήστη από το σύστημα, και αντίστοιχα να τον επαναφέρει σε αυτό.

Χωρίς να αναιρούνται τα προαναφερθέντα, προκύπτουν και οι εξής λειτουργικές απαιτήσεις για τους συνδεδεμένους χρήστες (User) και τους διαχειριστές του ιστοτόπου (Admin):

Ένας συνδεδεμένος χρήστης ή/και ένας διαχειριστής πρέπει να δύναται:

- να ενημερώνει τις πληροφορίες του λογαριασμού του και τον κωδικό πρόσβασής του.
- να αποσυνδέεται από τον λογαριασμό του.
- να έχει πρόσβαση στην αρχική σελίδα του ηλεκτρονικού καταστήματος.
- να έχει πρόσβαση στις σελίδες προϊόντων του ηλεκτρονικού καταστήματος.

3.3 Υλοποίηση

Έχοντας υπ’ όψιν τις απαιτήσεις όπως αυτές διατυπώνονται στην προηγούμενη παράγραφο, προχωρούμε στην παρουσίαση της λειτουργικής υλοποίησης των απαιτήσεων αυτών.

Είναι απαραίτητο να επισημάνουμε σε αυτό το σημείο πως η συγκεκριμένη παράγραφος του Κεφαλαίου περιλαμβάνει αποσπάσματα του πηγαίου κώδικα του ηλεκτρονικού καταστήματος σε PHP - Laravel. Στα πλαίσια της πτυχιακής αυτής εργασίας, θεωρήθηκε πρέπον να παρουσιαστούν τα αποσπάσματα αυτά, προκειμένου να δοθεί στον αναγνώστη μια πιο διαυγής εικόνα των μεθόδων που δημιουργήθηκαν για την υλοποίηση των προαναφερθέντων λειτουργικών απαιτήσεων. Παρ' όλα αυτά, το ηλεκτρονικό κατάστημα που περιγράφεται θα αποτελέσει στο εγγύς μέλλον μια λειτουργική πλατφόρμα, με πολύ λιγότερες αλλαγές από ότι προσθήκες στην δομή της. Για αυτό το λόγο, και προκειμένου να διατηρηθεί η ασφάλεια και η ακεραιότητα του έργου, τα αποσπάσματα του πηγαίου κώδικα αυτόνομα ΔΕΝ περιγράφουν πλήρως το back-end του ιστοτόπου, καθώς χρησιμοποιούνται επιπλέον μέθοδοι και συναρτήσεις για την διασύνδεση όλων αυτών.

Επιπλέον, σε τεχνικό επίπεδο, τα αποσπάσματα κώδικα δεν περιγράφουν ακριβώς μεθόδους, αλλά παράγωγες κλάσεις της μητρικής ονόματι "Controller". Στο framework Laravel, ένας "Controller" αποτελεί ένα αντικείμενο το οποίο είναι υπεύθυνο για την λογική διαχείρισης αιτημάτων ενός λογισμικού γραμμένο σε αυτό. Συγκεκριμένα, είναι υπεύθυνος για την ομαδοποίηση συσχετισμένων μεταξύ τους αιτημάτων που αφορούν μια συγκεκριμένη κλάση του λογισμικού, σε μία ενιαία δομή δεδομένων (κλάση). Κατά σύμβαση λοιπόν, και προς αποφυγή σύγχυσης των Controllers με τις υπόλοιπες βασικές δομικές κλάσεις του συστήματος ("User", "Product", "Category", "Cart", "Order" - περιγράφονται στην επόμενη παράγραφο) και των φυσικών οντοτήτων που περιγράφουν, στο παρόν έγγραφο θα αναφερόμαστε σε αυτούς ως "μεθόδους".

3.3.1 Κλάσεις και Αντικείμενα

Το ηλεκτρονικό κατάστημα που υλοποιήθηκε στα πλαίσια αυτής της πτυχιακής εργασίας υπάγεται στην σχεδιαστική λογική του αντικειμενοστραφούς προγραμματισμού. Προτού αναλύσουμε τις συναρτήσεις και τις μεθόδους που υλοποιούν την λειτουργικότητα του ιστοτόπου, είναι απαραίτητο να περιγράψουμε τις πέντε κλάσεις των αντικειμένων τα οποία σχετίζονται με αυτή: τον χρήστη, το προϊόν, την κατηγορία του προϊόντος, το καλάθι αγορών, και την παραγγελία.

Χρήστης (User)

Η κλάση που περιγράφει και υλοποιεί τον εκάστοτε χρήστη του συστήματος (εγγεγραμμένο και μη, συνδεδεμένο και μη, διαχειριστή ή μη) απαρτίζεται από εννέα protected μεταβλητές:

- **name:** (*string*) το καταχωρημένο όνομα του χρήστη
- **email:** (*string*) το καταχωρημένο email του χρήστη
- **password:** (*string*) ο καταχωρημένος κωδικός πρόσβασης του χρήστη.

- Η συγκεκριμένη μεταβλητή είναι κρυφή με σκοπό να σειριοποιηθεί - κρυπτογραφηθεί για λόγους ασφαλείας.
- **address:** (*string*) η καταχωρημένη διεύθυνση κατοικίας του χρήστη
- **is_banned:** (*boolean*) εάν ο χρήστης είναι ή δεν είναι αποκλεισμένος από το σύστημα
- **last_active:** (*datetime*) η ημερομηνία που συνδέθηκε τελευταία φορά ένας χρήστης
- **remember_token:** (*boolean*) εάν το σύστημα θα απομνημονεύσει ή μη τα στοιχεία σύνδεσης του χρήστη.
 - Η συγκεκριμένη μεταβλητή είναι κρυφή με σκοπό να σειριοποιηθεί - κρυπτογραφηθεί για λόγους ασφαλείας.
- **is_admin:** (*boolean*) εάν ο χρήστης είναι ή δεν είναι διαχειριστής του συστήματος.
 - Εάν η συγκεκριμένη μεταβλητή είναι True, ο συγκεκριμένος χρήστης φέρει δικαιώματα διαχειριστή. Ειδιάλλως, είναι απλά χρήστης - πελάτης του ηλεκτρονικού καταστήματος.

Επιπλέον, η κλάση απαρτίζεται από τρεις συναρτήσεις που καθορίζουν τη σχέση του εκάστοτε αντικειμένου της κλάσης User με τα αντικείμενα των υπολοίπων κλάσεων. Αυτές είναι:

- **favourites:** Ένας χρήστης μπορεί να έχει πολλά αγαπημένα προϊόντα (One-To-Many).
- **item orders:** Ένας χρήστης μπορεί να έχει πολλές παραγγελίες (One-To-Many).
- **cart:** Ένας χρήστης μπορεί να έχει ένα καλάθι προϊόντων. (One-To-One).

Τέλος, δημιουργείται και μία συνάρτηση (`getHasPasswordAttribute()`) που ελέγχει εάν ο χρήστης έχει καταχωρημένο κωδικό πρόσβασης στο σύστημα. Στο Παράρτημα Π.Α.1: Class User εμπεριέχεται ο κώδικας σε PHP - Laravel που υλοποιεί την παραπάνω κλάση.

Προϊόν (Product)

Η κλάση που περιγράφει και υλοποιεί το εκάστοτε προϊόν που καταχωρείται στο σύστημα απαρτίζεται από επτά protected μεταβλητές:

- **name:** (*string*) το καταχωρημένο όνομα του προϊόντος
- **description:** (*string*) η περιγραφή του προϊόντος
- **category_id:** (*int*) το αναγνωριστικό της κατηγορίας του προϊόντος
- **stock:** (*int*) η εναπομείνασα ποσότητα του προϊόντος στην αποθήκη (απόθεμα)

- **image:** (*string*) το path του αρχείου της εικόνας του προϊόντος στο σύστημα
- **price:** (*int*) η τιμή του προϊόντος
- **show:** (*int*) μεταβλητή που ανάλογα με την τιμή της ρυθμίζει τον τρόπο εμφάνισης του προϊόντος στις λίστες αγαπημένων και το καλάθι αγορών.

Επιπλέον, η κλάση *apαρτίζεται από τέσσερις συναρτήσεις που καθορίζουν τη σχέση του εκάστοτε αντικειμένου της κλάσης Product με τα αντικείμενα των υπολοίπων κλάσεων. Αυτές είναι:*

- **category:** Ένα αντικείμενο μπορεί να ανήκει σε μία κατηγορία (One-To-One).
- **favourites:** Ένας αντικείμενο μπορεί να είναι σε πολλές λίστες “Αγαπημένων” χρηστών (One-To-Many).
- **orders:** Ένας αντικείμενο μπορεί να είναι σε πολλές παραγγελίες χρηστών (One-To-Many).
- **cart:** Ένας αντικείμενο μπορεί να ανήκει σε καλάθια αγορών πολλών χρηστών (One-To-Many).

Τέλος, δημιουργείται και μία συνάρτηση (*scopeShop(...)*) που καλεί ένα query και επιστρέφει σε ποιες λίστες “Αγαπημένων” και καλάθια αγορών εντοπίζεται το συγκεκριμένο προϊόν. Στο Παράρτημα Π.Α.2: Class “Product” *εμπεριέχεται ο κώδικας σε PHP - Laravel που υλοποιεί την παραπάνω κλάση.*

Κατηγορία Προϊόντος (Category)

Η κλάση που περιγράφει και υλοποιεί την εκάστοτε κατηγορία προϊόντων στο e-shop *απαρτίζεται από τρεις protected μεταβλητές:*

- **name:** (*string*) το καταχωρημένο όνομα της κατηγορίας
- **parent_id:** (*int*) το αναγνωριστικό της γενικότερης κατηγορίας που εμπίπτει η συγκεκριμένη. Κατ’ αυτόν τον τρόπο προβλέπεται η υλοποίηση υπο-κατηγοριών.
- **show:** (*int*) μεταβλητή που ανάλογα με την τιμή της ρυθμίζει τον τρόπο εμφάνισης της κατηγορίας στο σύστημα.

Επιπλέον, η κλάση *απαρτίζεται από τρεις συναρτήσεις που καθορίζουν τη σχέση του εκάστοτε αντικειμένου της κλάσης Category με τα αντικείμενα των υπολοίπων κλάσεων. Αυτές είναι:*

- **parent:** Μια κατηγορία μπορεί να ανήκει μόνο σε μία κατηγορία (One-To-One).

- **children:** Σε μια κατηγορία μπορούν να ανήκουν πολλές υπο-κατηγορίες (One-To-Many).
- **products:** Σε μια κατηγορία μπορούν να υπάγονται πολλά προϊόντα (One-To-Many).

Στο Παράρτημα Π.Α.3: Class "Category" παρουσιάζεται ο κώδικας σε PHP - Laravel που υλοποιεί την παραπάνω κλάση.

Καλάθι αγορών (Cart)

Η κλάση που περιγράφει και υλοποιεί το καλάθι αγορών του εκάστοτε πελάτη του e-shop απαρτίζεται από μία protected μεταβλητή:

- **user_id:** (*int*) Το αναγνωριστικό του χρήστη στον οποίο ανήκει το καλάθι.

Επιπλέον, η κλάση απαρτίζεται από δύο συναρτήσεις που καθορίζουν τη σχέση του εκάστοτε αντικειμένου της κλάσης Cart με τα αντικείμενα των υπολοίπων κλάσεων. Αυτές είναι:

- **user:** Ένα καλάθι αγορών μπορεί να ανήκει μόνο σε έναν πελάτη (One-To-One).
- **products:** Σε ένα καλάθι αγορών μπορούν να υπάρχουν πολλά προϊόντα (One-To-Many).

Στο Παράρτημα Π.Α.4: Class "Cart" παρουσιάζεται ο κώδικας σε PHP - Laravel που υλοποιεί την παραπάνω κλάση.

Παραγγελίες (Order)

Η κλάση που περιγράφει και υλοποιεί τις παραγγελίες του εκάστοτε πελάτη του e-shop απαρτίζεται από πέντε protected μεταβλητές:

- **user_id:** (*int*) Το αναγνωριστικό του χρήστη στον οποίο ανήκει η παραγγελία.
- **status:** (*string*) Η κατάσταση της παραγγελίας στο σύστημα.
- **ship_fee:** (*int*) Τα έξοδα αποστολής της παραγγελίας.
- **shipped_at:** (*datetime*) Η ημερομηνία και ώρα αποστολής της παραγγελίας.
- **delivered_at:** (*datetime*) Η ημερομηνία και ώρα παραλαβής της παραγγελίας.

Επιπλέον, η κλάση απαρτίζεται από δύο συναρτήσεις που καθορίζουν τη σχέση του εκάστοτε αντικειμένου της κλάσης Order με τα αντικείμενα των υπολοίπων κλάσεων. Αυτές είναι:

- **user**: Μία παραγγελία μπορεί να ανήκει μόνο σε έναν πελάτη (One-To-One).
- **products**: Σε μία παραγγελία μπορούν να υπάρχουν πολλά προϊόντα (One-To-Many).

Στο Παράρτημα Π.Α.5: Class "Order" παρουσιάζεται ο κώδικας σε PHP - Laravel που υλοποιεί την παραπάνω κλάση.

3.3.2 Δημιουργία Λογαριασμού Χρήστη

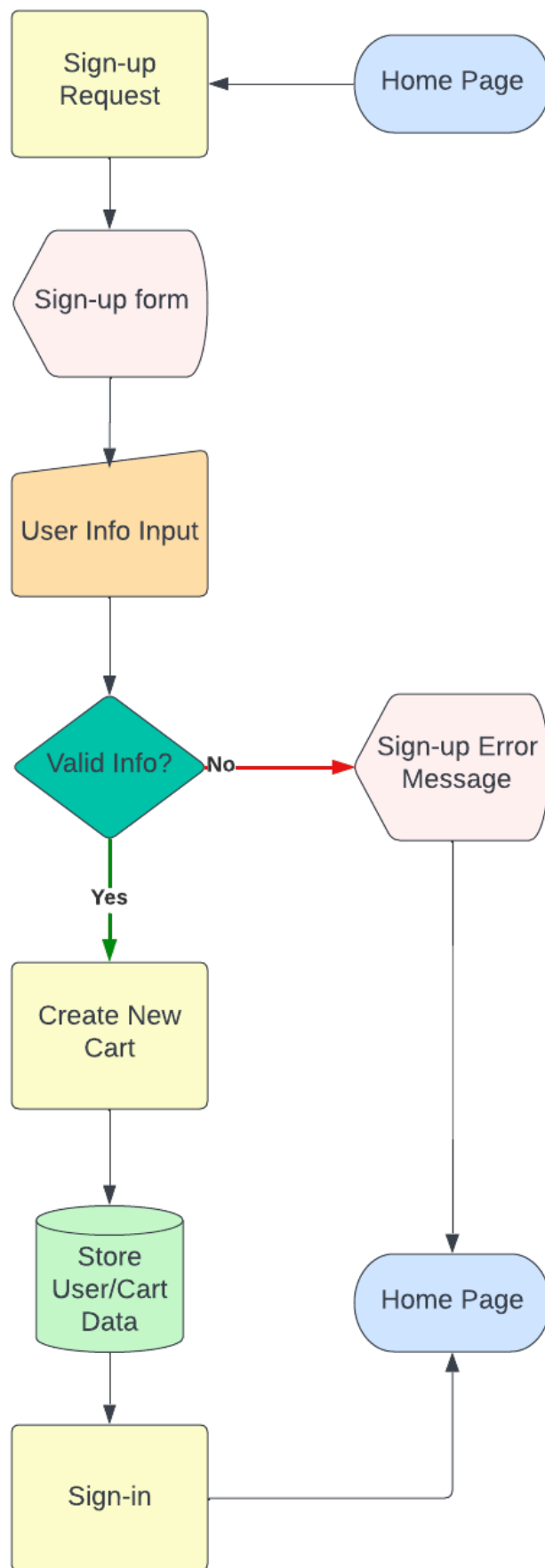
Το εμπορικό ταξίδι του εκάστοτε πελάτη του ηλεκτρονικού καταστήματος που πραγματεύεται αυτή η διπλωματική εργασία ξεκινά με την Δημιουργία Λογαριασμού Χρήστη στον ιστοτόπο. Ας σημειωθεί εδώ πως σε αυτή τη διαδικασία υπόκεινται μόνο οι χρήστες του ιστοτόπου με δικαιώματα User. Οι λογαριασμοί διαχειριστών του συστήματος (Admin) δημιουργούνται με ξεχωριστή διαδικασία, εσωτερικά του συστήματος. Η διαδικασία αυτή δεν περιγράφεται εντός αυτού του εγγράφου για λόγους διατήρησης της ακεραιότητας του ιστοτόπου.

Για την εξυπηρέτηση αυτής της λειτουργικότητας της ιστοσελίδας υλοποιείται μία μέθοδος αποτελούμενη από δύο συναρτήσεις. Η πρώτη αναλαμβάνει την διαμόρφωση του αιτήματος εγγραφής ενός χρήστη στην ιστοσελίδα, μέσω της προβολής μιας φόρμας συμπλήρωσης στοιχείων. Εντός αυτής ο χρήστης συμπληρώνει το όνομά του, την διεύθυνση ηλεκτρονικού ταχυδρομείου του (η οποία είναι δεσμευτική για τον λογαριασμό του), τη διεύθυνσή του, το τηλέφωνό του και τον κωδικό πρόσβασής του. Η δεύτερη συνάρτηση της μεθόδου αναλαμβάνει την δημιουργία και την αποστολή του σχετικού query στην Βάση Δεδομένων, αφού δημιουργήσει ένα object της κλάσης User στις μεταβλητές του οποίου αναθέτει τα στοιχεία που δόθηκαν στο σύστημα μέσω της φόρμας που προαναφέρθηκε. Τέλος, δημιουργείται ένα object της κλάσης Cart που αντιπροσωπεύει το Καλάθι Αγορών του χρήστη. Σε αυτό δεσμεύεται ένα μοναδικό αναγνωριστικό που το συνδέει με τον λογαριασμό του χρήστη. Με την επιτυχή ολοκλήρωση της διαδικασίας, ο χρήστης συνδέεται αυτόματα στο σύστημα και ανακατευθύνεται στην Αρχική Σελίδα του καταστήματος. Στο Παράρτημα Π.Α.6: Create "User" παρουσιάζεται ο κώδικας σε PHP - Laravel που υλοποιεί την παραπάνω μέθοδο.

Στην Εικόνα 3.1 παρουσιάζεται μέσω ενός διαγράμματος ροής η διαδικασία που ακολουθείται στο σύστημα κατά την εγγραφή ενός καινούριου χρήστη σε αυτό.

3.3.3 Σύνδεση - Αποσύνδεση Λογαριασμού

Εφόσον ένας χρήστης (πελάτης ή διαχειριστής) του ηλεκτρονικού καταστήματος έχει δημιουργήσει ένα λογαριασμό, έχει την δυνατότητα ανά πάσα στιγμή να συνδεθεί (και να αποσυνδεθεί) σε αυτόν. Η "σύνδεση" αυτή λαμβάνει χώρα ως μια επικυρωμένη συνεδρία κατά την περιήγηση του ιστοτόπου. Η συνεδρία ξεκινά με την σύνδεση του χρήστη στο λογαριασμό του, μέσω μιας



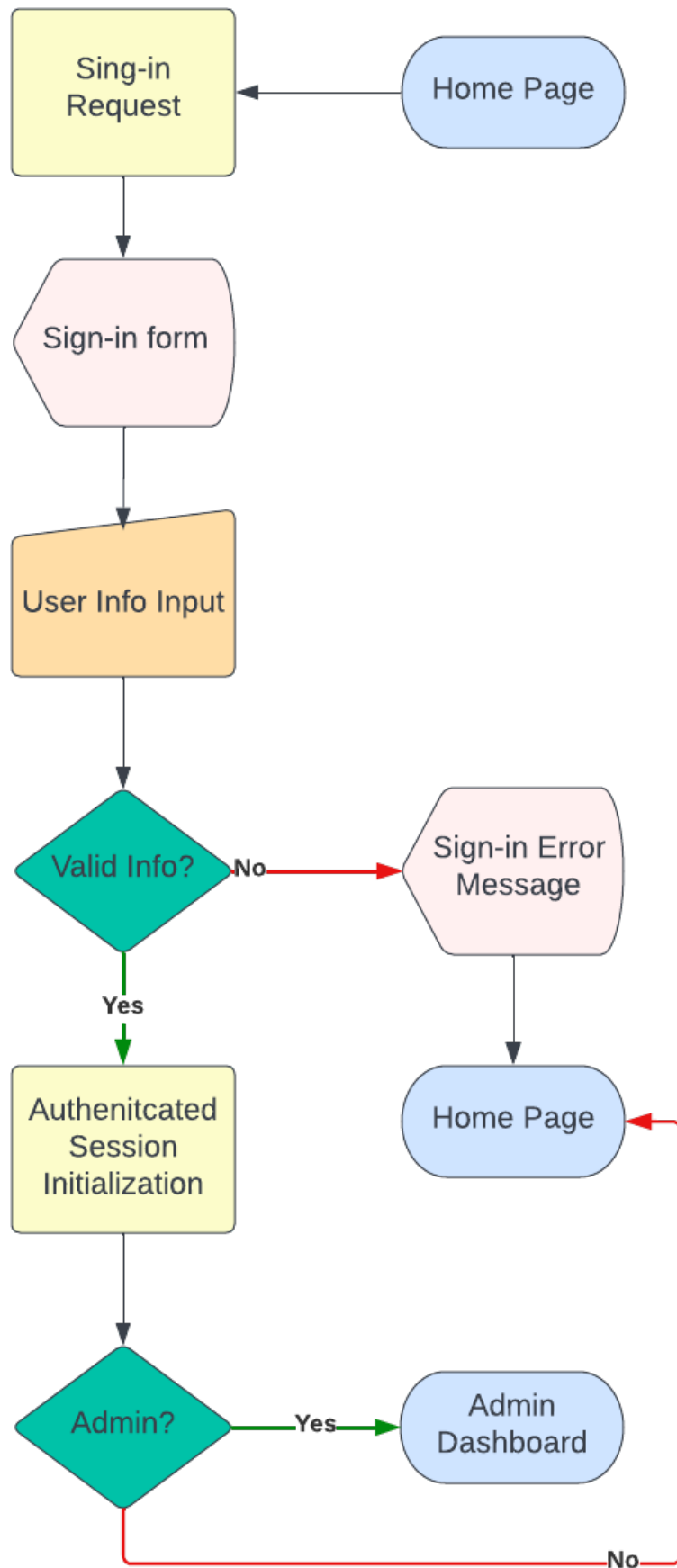
Εικόνα 3.1: Διαγραμματική Αναπαράσταση Εγγραφής Χρήστη στο Σύστημα.

αντίστοιχης φόρμας συμπλήρωσης στοιχείων: email, κωδικός πρόσβασης και την επιλογή το σύστημα να αποστηθίσει τα διαπιστευτήριά του για την επόμενη φορά. Για όλη αυτή τη διαδικασία είναι υπεύθυνη η πρώτη συνάρτηση της μεθόδου που υλοποιήθηκε για το κομμάτι της σύνδεσης και της αποσύνδεσης ενός χρήστη από το λογαριασμό του.

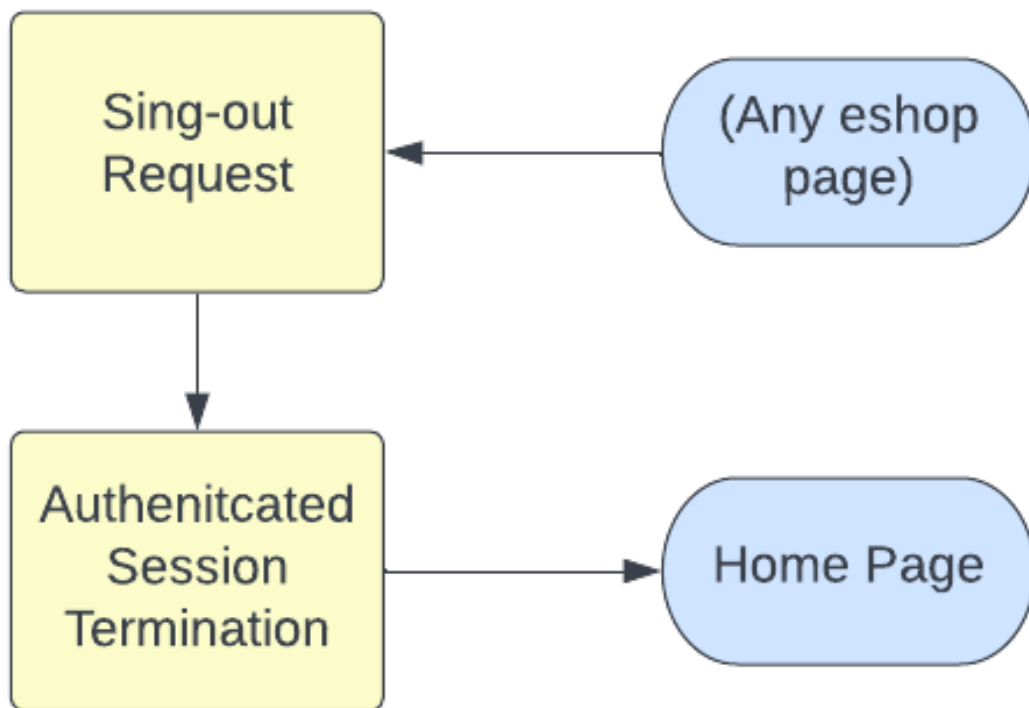
Η δεύτερη συνάρτηση της μεθόδου αναλαμβάνει την αποστολή του αιτήματος στο back-end του συστήματος. Με την επιτυχή επικύρωση των στοιχείων που έδωσε ο χρήστης, ξεκινά την συνεδρία αρχικοποιώντας το καλάθι αγορών. Με την ολοκλήρωση της διαδικασίας, ανακατευθύνει τον χρήστη στην Αρχική Σελίδα του καταστήματος εάν ο χρήστης πρόκειται για απλό πελάτη, ή στο dashboard διαχείρισης της ιστοσελίδας εάν ο χρήστης πρόκειται για διαχειριστή.

Η τρίτη και τελευταία συνάρτηση της μεθόδου σύνδεσης - αποσύνδεσης αναλαμβάνει την διαδικασία αποσύνδεσης ενός λογαριασμό, ή πιο συγκεκριμένα, τον τερματισμό της επικυρωμένης συνεδρίας για την οποία έγινε λόγος στην προηγούμενη παράγραφο. Με το πέρας της διαδικασίας, ο χρήστης ανακατευθύνεται στην Αρχική Σελίδα του καταστήματος.

Στο Παράρτημα Π.Α.7: Sign-in/Sign-out παρουσιάζεται ο κώδικας που υλοποιεί την παραπάνω διαδικασία σε PHP - Laravel. Στην Εικόνα 3.2 παρουσιάζεται μέσω ενός διαγράμματος ροής η διαδικασία που ακολουθείται στο σύστημα κατά την σύνδεση ενός χρήστη σε αυτό. Αντίθετα, στην Εικόνα 3.3 παρουσιάζεται διαγραμματικά η διαδικασία που διενεργείται στο σύστημα κατά την αποσύνδεση ενός χρήστη από αυτό.



Εικόνα 3.2: Διαγραμματική Αναπαράσταση Σύνδεσης Χρήστη στο Σύστημα.



Εικόνα 3.3: Διαγραμματική Αναπαράσταση Αποσύνδεσης Χρήστη από το Σύστημα.

3.3.4 Επαναφορά Κωδικού Πρόσβασης

Η Επαναφορά Κωδικού Πρόσβασης ενός λογαριασμού ενός χρήστη αποτελεί μια διαδικασία τεσσάρων σταδίων: το αίτημα του χρήστη για την επαναφορά του κωδικού πρόσβασής του, την αποστολή ενός συνδέσμου στην διεύθυνση ηλεκτρονικού ταχυδρομείου που έχει δηλώσει στο σύστημα, την καταχώρηση από τον χρήστη του καινούριου κωδικού πρόσβασης που επιθυμεί για τον λογαριασμό του στο σύστημα, και την ενημέρωση του λογαριασμού και την συνεπαγόμενη ενημέρωση των στοιχείων του στη Βάση Δεδομένων από το Σύστημα.

Για το πρώτο στάδιο και το δεύτερο στάδιο υλοποιείται μία ενιαία μέθοδος δύο συναρτήσεων, η κάθε μία εκ των οποίων αναλαμβάνει το αντίστοιχο στάδιο. Η πρώτη δίνει την δυνατότητα στον χρήστη να αποστείλει ένα αίτημα στο σύστημα για επαναφορά του κωδικού πρόσβασής του. Η δεύτερη συνάρτηση ανακτά και επικυρώνει την διεύθυνση ηλεκτρονικού ταχυδρομείου του χρήστη που αιτείται την αλλαγή, και του αποστέλλει αυτόματα έναν σύνδεσμο που τον ανακατευθύνει στο περιβάλλον καταχώρησης νέου κωδικού πρόσβασης. Ο σχετικός κώδικας υλοποίησης σε PHP - Laravel παρουσιάζεται στο Παράρτημα Π.Α.8: Reset Password.

Κατόπιν χρήσης του συνδέσμου που του έχει αποσταλεί μέσω ηλεκτρονικού ταχυδρομείου, ο χρήστης ανακατευθύνεται σε περιβάλλον καταχώρησης νέου κωδικού πρόσβασης. Η καταχώρηση αυτή ακολουθεί την ίδια λογική με την διαδικασία δημιουργίας λογαριασμού. Το σύστημα αιτείται και επικυρώνει το email του χρήστη, ελέγχει τον νέο κωδικό πρόσβασής του εάν πληροί τα κριτήρια που έχουν τεθεί, και κατόπιν επιβεβαίωσης, ενημερώνει την Βάση Δεδομένων με τα νέα στοιχεία και την χρονική στιγμή της ανανέωσής τους, και επιστρέφει μήνυμα επιβεβαίωσης στον χρήστη, πριν τον ανακατευθύνει στην Αρχική Σελίδα.

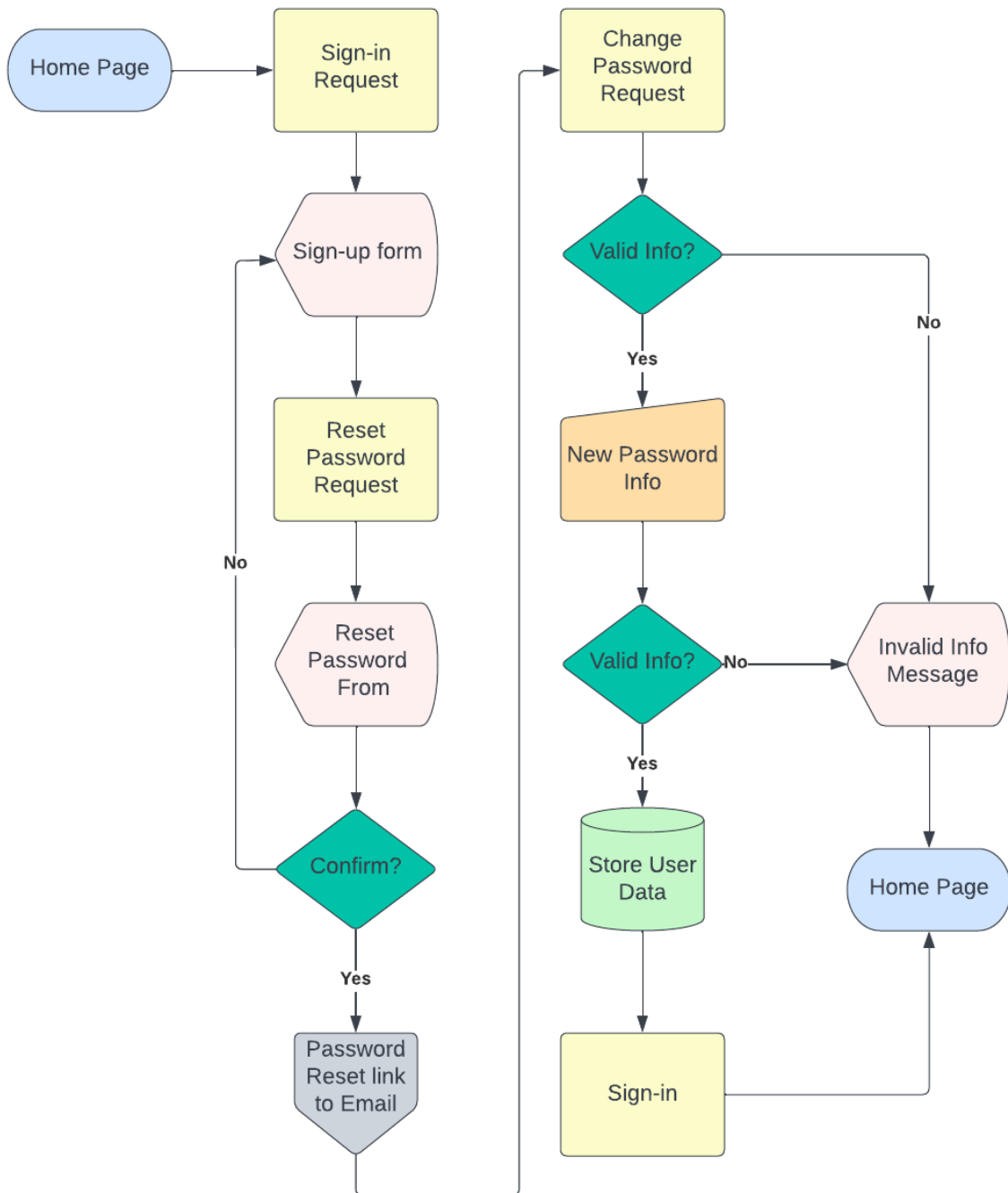
Η παραπάνω διαδικασία λαμβάνει χώρα μέσω μίας μεθόδου δύο συναρτήσεων: η πρώτη αναλαμβάνει την εισαγωγή του νέου κωδικού πρόσβασης, και η δεύτερη την διαδικασία της επικύρωσης και ενημέρωσης στοιχείων. Ο κώδικάς τους σε PHP - Laravel εμπεριέχεται στο Παράρτημα Π.Α.9: Confirm New Password.

Τέλος, το τελευταίο στάδιο λαμβάνει χώρα μεταξύ του αιτήματος του χρήστη για επαναφορά κωδικού και ενημέρωσης των στοιχείων του με τον καινούριο. Για αυτό το σκοπό υλοποιείται μία μέθοδος δύο συναρτήσεων. Η πρώτη απλώς μεταφέρει το αίτημα του χρήστη για επικύρωση του νέου κωδικού πρόσβασης στο back-end του συστήματος. Η δεύτερη είναι υπεύθυνη για όλους τους ενδιάμεσους ελέγχους και τροποποιήσεις στη Βάση Δεδομένων. Αρχικά, επικυρώνονται τα δεδομένα που έχει δώσει ο χρήστης, και συγκεκριμένα, εάν το αίτημα που προέκυψε προήλθε από έναν επικυρωμένο σύνδεσμο ανάκτησης κωδικού πρόσβασης ο οποίος απεστάλη σε μια διεύθυνση ηλεκτρονικού ταχυδρομείου που όντως είχε αιτηθεί αλλαγή κωδικού πρόσβασης, εάν όντως δόθηκε από τον χρήστη η διεύθυνση αυτή, και τέλος εάν ο νέος κωδικός πρόσβασης είναι τουλάχιστον έξι χαρακτήρων. Στην συνέχεια, εάν η διαδικασία που προαναφέρθηκε είναι επιτυχής, ενημερώνει τα στοιχεία του χρήστη στην Βάση Δεδομένων και δρομολογεί την διαδικασία ανακατεύθυνσής του σε μια σελίδα επιβεβαίωσης επιτυχούς διαδικασίας.

Κεφάλαιο 3

Εάν η διαδικασία αποτύχει, ανακατευθύνει τον χρήστη σε μια σελίδα όπου τον ενημερώνει για την αποτυχία και τον προτρέπει να ξαναπροσπαθήσει. Ο κώδικας σε PHP - Laravel της μεθόδου αυτής παρουσιάζεται στο Παράρτημα Π.Α.10: Store New Password.

Στην Εικόνα 3.4 παρουσιάζεται μέσω διαγράμματος ροής η συνολική διαδικασία που περιγράφεται στην παράγραφο αυτή.



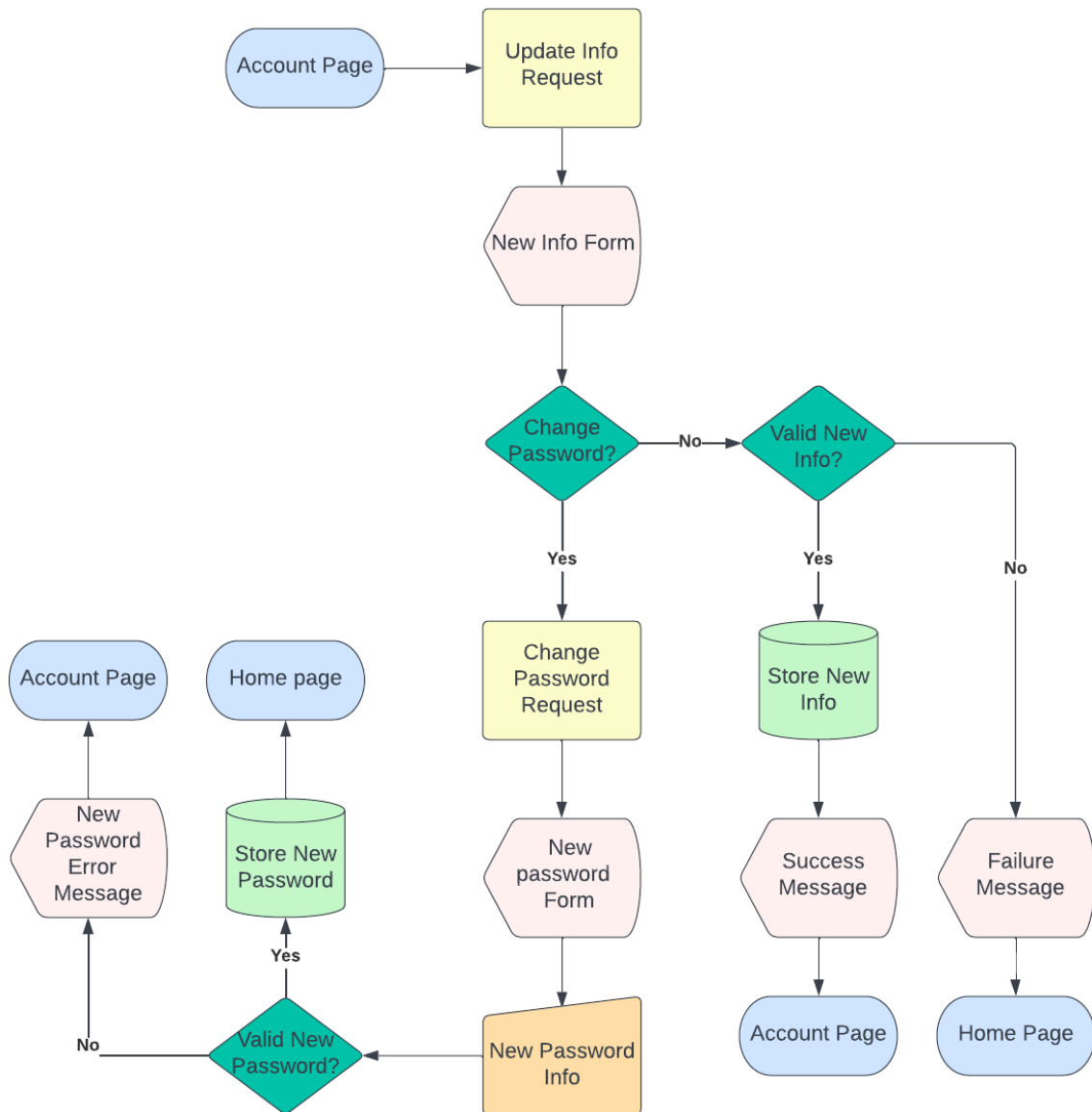
Εικόνα 3.4: Διαγραμματική Αναπαράσταση Διαδικασίας Επαναφοράς Κωδικού Πρόσβασης.

3.3.5 Ενημέρωση Στοιχείων Λογαριασμού

Η Ενημέρωση Στοιχείων Λογαριασμού ενός εγγεγραμμένου χρήστη στο σύστημα απαιτεί αρχικά ο χρήστης να είναι συνδεδεμένος από αυτό. Ο χρήστης έχει τη δυνατότητα να τροποποιήσει οποιαδήποτε πληροφορία για αυτόν στο σύστημα, πλην του λογαριασμού ηλεκτρονικού ταχυδρομείου με τον οποίο έχει εγγραφεί στο σύστημα. Οι υπόλοιπες πληροφορίες δεν θεωρούνται ευαίσθητες για την ακεραιότητα του συστήματος, εκτός φυσικά από την αλλαγή του κωδικού πρόσβασης του χρήστη σε αυτό. Για τον λόγο αυτό, οι πληροφορίες αυτές υπόκεινται σε διαφορετικές διαδικασίες ενημέρωσης - τροποποίησης.

Για την εξυπηρέτηση της συνολικής αυτής λειτουργικότητας υλοποιείται μία ενιαία μέθοδος που απαρτίζεται από τέσσερις συναρτήσεις. Η πρώτη αναλαμβάνει την εξυπηρέτηση του αιτήματος του χρήστη, προβάλλοντάς του μία φόρμα επεξεργασίας των στοιχείων του. Η δεύτερη αναλαμβάνει την ενημέρωση όλων των στοιχείων του, πλην της διεύθυνσης ηλεκτρονικού ταχυδρομείου του και του κωδικού πρόσβασής του. Σκοπός της είναι η εκτέλεση του σχετικού query στην Βάση Δεδομένου με τα νέα δεδομένα προς ενημέρωση των παλιών. Με την επιτυχή ολοκλήρωση της διαδικασίας επιστρέφει στον χρήστη το αντίστοιχο μήνυμα. Η τρίτη συνάρτηση αναλαμβάνει την τροποποίηση του κωδικού πρόσβασης ενός χρήστη στο σύστημα, ανακατευθύνοντάς τον σε μια φόρμα ειδικού σκοπού, όπου καλείται να συμπληρώσει τον παλιό και τον καινούριο του κωδικό πρόσβασης. Με την ολοκλήρωση της καταχώρησης του καινούριου κωδικού, καλείται η τέταρτη συνάρτηση της μεθόδου. Αρχικά, η συνάρτηση αυτή επιβεβαιώνει την ορθότητα του παλιού κωδικού πρόσβασης στο σύστημα, ελέγχει τον καινούριο κωδικό για το αν πληροί τα κριτήρια που έχουν τεθεί, και τέλος εκτελεί το αντίστοιχο query στην Βάση Δεδομένων. Με την επιτυχή ολοκλήρωση της διαδικασίας αποστέλλει στον χρήστη το ανάλογο μήνυμα.

Στο Παράρτημα Π.Α.11: Update User Info παρουσιάζεται ο κώδικας που υλοποιεί την παραπάνω μέθοδο σε PHP - Laravel. Στην Εικόνα 3.5 παρουσιάζεται δι' ενός διαγράμματος ροής η συνολική διαδικασία ενημέρωσης στοιχείων χρήστη, όπως αυτή διενεργείται εντός του συστήματος.



Εικόνα 3.5: Διαγραμματική Αναπαράσταση Διαδικασίας Ενημέρωσης Στοιχείων Χρήστη.

3.3.6 Σελίδες Προϊόντων

Όπως αναφέρθηκε στην αρχή του τρέχοντος Κεφαλαίου, κάθε επισκέπτης του ηλεκτρονικού καταστήματος πρέπει να έχει πρόσβαση στις (ιστο)σελίδες των προϊόντων. Αυτό περιλαμβάνει τόσο τις επιμέρους σελίδες των προϊόντων που προσφέρει το ηλεκτρονικό κατάστημα, όσο και την Αρχική Σελίδα του ιστοτόπου. Ο ιστοτόπος του καταστήματος πρέπει επίσης να παρέχει πρόσβαση στις Κατηγορίες προϊόντων του καταστήματος στους επισκέπτες, όπως και την δυνατότητα αναζήτησης προϊόντων μέσω ενός σχετικού εργαλείου. Η λειτουργικότητα αυτή υλοποιείται μέσω μιας ενιαίας μεθόδου για όλες τις υπηρεσίες αυτές, η οποία απαρτίζεται από έξι κύριες συναρτήσεις.

Η πρώτη συνάρτηση αφορά την προβολή της Αρχικής Σελίδας του ηλεκτρονικού καταστήματος. Σε αυτή προβάλλεται ένας συγκεκριμένος αριθμός προϊόντων με συγκεκριμένα κριτήρια: τα κορυφαία δύο προϊόντα που εντοπίζονται σε λίστες "Αγαπημένων" των πελατών του καταστήματος, τα οκτώ πιο δημοφιλή προϊόντα βάσει αριθμού παραγγελιών, και κάποια από τα προϊόντα που απαρτίζουν τις τρεις πιο "πολυπληθείς" Κατηγορίες. Με την ανάλογη διαχείριση, τοποθετούνται σε διαφορετικές μεριές του γραφικού περιβάλλοντος της Αρχικής Σελίδας του καταστήματος, χωρίς να είναι απολύτως δεσμευτικό το πού σε αυτή τη φάση.

Η δεύτερη συνάρτηση αφορά τη γενική μορφή των υπόλοιπων σελίδων προϊόντων. Τα προϊόντα που προβάλλονται σε αυτές ομαδοποιούνται ανά κατηγορία, και δύνανται να ταξινομηθούν ανά σειρά δημοφιλίας, τιμής (αύξουσας ή φθίνουσας), αλφαβητικής ή ακόμα και βάσει της περιγραφής τους.

Η τρίτη συνάρτηση προβάλλει στον χρήστη όλα τα προϊόντα που εμπεριέχονται σε μια συγκεκριμένη κατηγορία. Η σειρά προβολής τους βασίζεται στο αναγνωριστικό τους. Για την ολοκλήρωση αυτού του τμήματος της λειτουργικότητας, η τέταρτη συνάρτηση που υλοποιήθηκε αφορά την προβολή προϊόντων από διάφορες κατηγορίες βάσει φίλτρων.

Η πέμπτη συνάρτηση είναι υπεύθυνη για το εργαλείο αναζήτησης προϊόντων του καταστήματος. Ουσιαστικά, είναι υπεύθυνη για την εκτέλεση του query προς την Βάση Δεδομένων με τις λέξεις κλειδιά που πληκτρολογεί ο χρήστης. Εφόσον εντοπίσει όλα τα προϊόντα που πληρούν τα κριτήρια αναζήτησης, τα επιστρέφει σε αυτόν οργανωμένα ανά Κατηγορία.

Τέλος, η έκτη συνάρτηση είναι υπεύθυνη για την προβολή της σελίδας του εκάστοτε προϊόντος στον χρήστη. Κάθε μια από αυτές τις σελίδες περιλαμβάνουν τις λεπτομέρειες του εκάστοτε προϊόντος: την Κατηγορία του, το αναγνωριστικό του, την περιγραφή του, την τιμή του, και επιπλέον την δυνατότητα να το προσθέσει ο χρήστης στη λίστα "Αγαπημένων" του, ή/και στο Καλάθι Αγορών του.

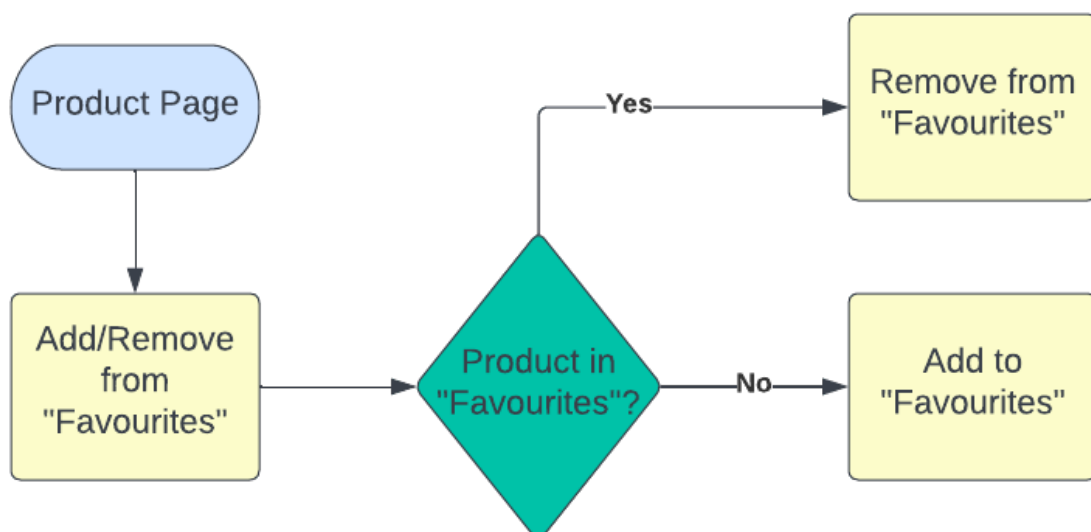
Επιπλέον, υλοποιείται μια private συνάρτηση, η οποία χρησιμοποιείται εσωτερικά της μεθόδου για την ανάκτηση όλων των Κατηγοριών των προϊόντων, και των ιδιοτήτων τους σαν objects.

Ο κώδικας που υλοποιεί την μέθοδο αυτή σε PHP - Laravel εμπεριέχεται στο Παράρτημα Π.Α.12: Product Pages. Η οπτικοποίηση της λειτουργικότητας της μεθόδου γίνεται πιο κατανοητή εκ' του αποτελέσματος στις Εικόνες 4.1 και 4.6 του επόμενου Κεφαλαίου.

3.3.7 Λίστα “Αγαπημένων”

Σαν concept, η Λίστα “Αγαπημένων” ενός πελάτη του ηλεκτρονικού καταστήματος υλοποιείται με απλό τρόπο αποθηκεύοντας για τον κάθε πελάτη σε μια μεταβλητή την προτίμησή του για ένα προϊόν. Συγκεκριμένα, στην καταχώρηση του κάθε προϊόντος στην Βάση Δεδομένων υπάρχει ένα συγκεκριμένο πεδίο το οποίο περιλαμβάνει τα αναγνωριστικά όλων των πελατών οι οποίοι προσέθεσαν το συγκεκριμένο προϊόν στην Λίστα “Αγαπημένων” τους. Υπενθυμίζεται ότι η καταχώρηση αυτή εκτελείται σαν διαδικασία μέσω της συνάρτησης "favorites()" της κλάσης "Product". Η μέθοδος που υλοποιείται για την προσθήκη των προϊόντων στην συγκεκριμένη λίστα - και περιγράφεται σε αυτή την παράγραφο - είναι υπεύθυνη, μέσω της μοναδικής συνάρτησης που περιλαμβάνει, να εκτελέσει αυτή τη διαδικασία και να επιστρέψει στο σύστημα απάντηση επιτυχούς ή μη εκτέλεσης (κωδικοί σφάλματος 200 και 201 αντίστοιχα). Η επαναληπτική κλήση της μεθόδου - συνάρτησης, λειτουργεί αντίστροφα, αφαιρώντας ένα προϊόν από τη Λίστα “Αγαπημένων”.

Στο Παράρτημα Π.Α.13: "Favourites" παρουσιάζεται ο κώδικας της μεθόδου σε PHP - Laravel. Η αλγοριθμική διαδικασία της μεθόδου που περιγράφεται άνωθεν παρουσιάζεται μέσω ενός απλού διαγράμματος ροής στην Εικόνα 3.6



Εικόνα 3.6: Διαγραμματική Αναπαράσταση Διαδικασίας Προσθήκης/Αφαίρεσης Προϊόντος από την Λίστα “Αγαπημένων”.

Επιπλέον, υλοποιείται μια δεύτερη μέθοδος μίας συνάρτησης για την προβολή στον χρήστη

όλων των προϊόντων που έχει προσθέσει στην Λίστα “Αγαπημένων” του. Ο κώδικάς της σε PHP - Laravel περιλαμβάνεται στο Παράρτημα Π.Α.14: Show ”Favourites”.

3.3.8 Καλάθι Αγορών

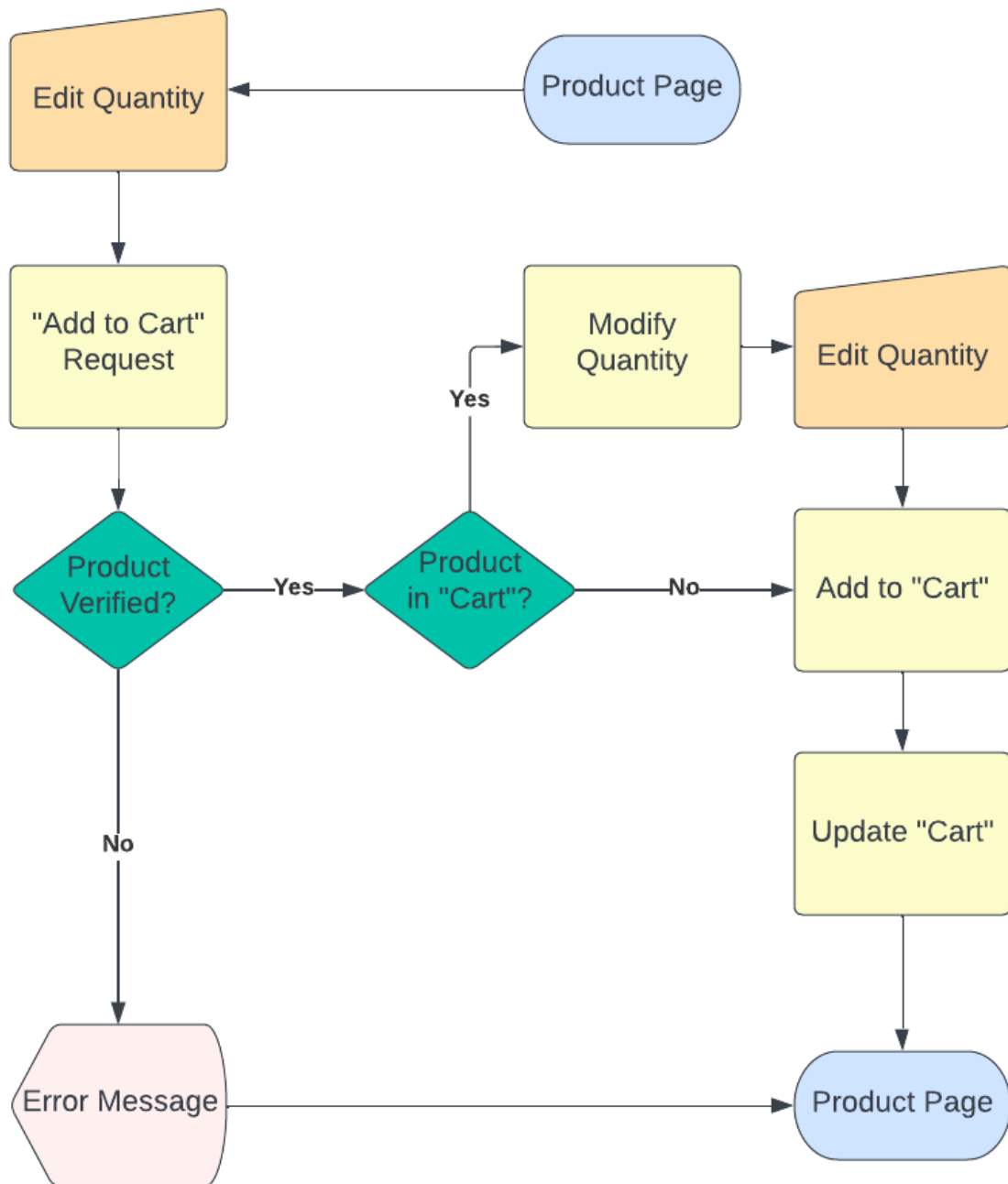
Όλες οι διαδικασίες που αφορούν το Καλάθι Αγορών ενός πελάτη του ηλεκτρονικού καταστήματος εμπίπτουν στα δικαιώματα του ίδιου του πελάτη - χρήστη (“User”). Οι διαδικασίες αυτές αφορούν την προσθήκη ενός προϊόντος στο καλάθι, την ενημέρωση (αλλαγή) της ποσότητάς του σε αυτό και την ολική αφαίρεσή του (διαγραφή) από αυτό. Υπενθυμίζεται πως το Καλάθι Αγορών αποτελεί ξεχωριστό object για κάθε User το οποίο προκύπτει από δικιά του κλάση (“Cart”), όπως αυτή περιγράφεται στην παράγραφο 3.2.1.4 αυτού του Κεφαλαίου.

Για την εξυπηρέτηση του συνόλου των διαδικασιών αυτών, υλοποιούνται τρεις μέθοδοι. Η πρώτη αφορά την προσθήκη ενός προϊόντος στο καλάθι και αποτελείται από μία συνάρτηση. Η συνάρτηση αυτή αναλαμβάνει την εκτέλεση του σχετικού αιτήματος προς την Βάση Δεδομένων. Αρχικά, επικυρώνει την ύπαρξη του αντικειμένου και της αιτούμενης ποσότητάς του στην Βάση, και έπειτα το προσθέτει στο Καλάθι Αγορών του πελάτη. Στο Παράρτημα Π.Α.15: Add to Cart ο κώδικας σε PHP - Laravel που υλοποιεί την παραπάνω συνάρτηση - μέθοδο.

Η δεύτερη μέθοδος αφορά την επεξεργασία των περιεχομένων ενός Καλαθιού Αγορών. Αποτελείται από δύο συναρτήσεις. Η πρώτη εκ των δύο αναλαμβάνει την ενημέρωση της ποσότητας ενός προϊόντος στο Καλάθι. Αφού επικυρώσει την ύπαρξη του προϊόντος στο απόθεμα του καταστήματος (Βάση Δεδομένων), αυξάνει ή μειώνει την ποσότητα του προϊόντος, ανάλογα με την επιθυμία του πελάτη. Η δεύτερη συνάρτηση είναι υπεύθυνη για την διαγραφή ενός προϊόντος από το Καλάθι Αγορών. Δεδομένου ότι το προϊόν δεν έχει ακόμα αγοραστεί, χωρίς να εμπλέκει το απόθεμά του στο κατάστημα, απλώς το αφαιρεί από το Καλάθι. Στο Παράρτημα Π.Α.16: Update Cart ο κώδικας σε PHP - Laravel που υλοποιεί την συγκεκριμένη συνάρτηση - μέθοδο.

Η τελευταία μέθοδος που συνδέεται με την υλοποίηση της λειτουργικότητας του Καλαθιού Αγορών αφορά την προσπέλαση και προβολή αυτού στο χρήστη. Αποτελείται από μία συνάρτηση η οποία, αφενός, απαριθμεί τα προϊόντα που εμπεριέχονται σε αυτό, και, αφετέρου, υπολογίζει και προβάλλει την συνολική τους αξία, προ μεταφορικών εξόδων. Ο πηγαίος κώδικας που την υλοποιεί βρίσκεται στο Παράρτημα Π.Α.17: Show Cart.

Η γενική διαδικασία προσθήκης ενός προϊόντος στο Καλάθι Αγορών οπτικοποιείται στην Εικόνα 3.7.



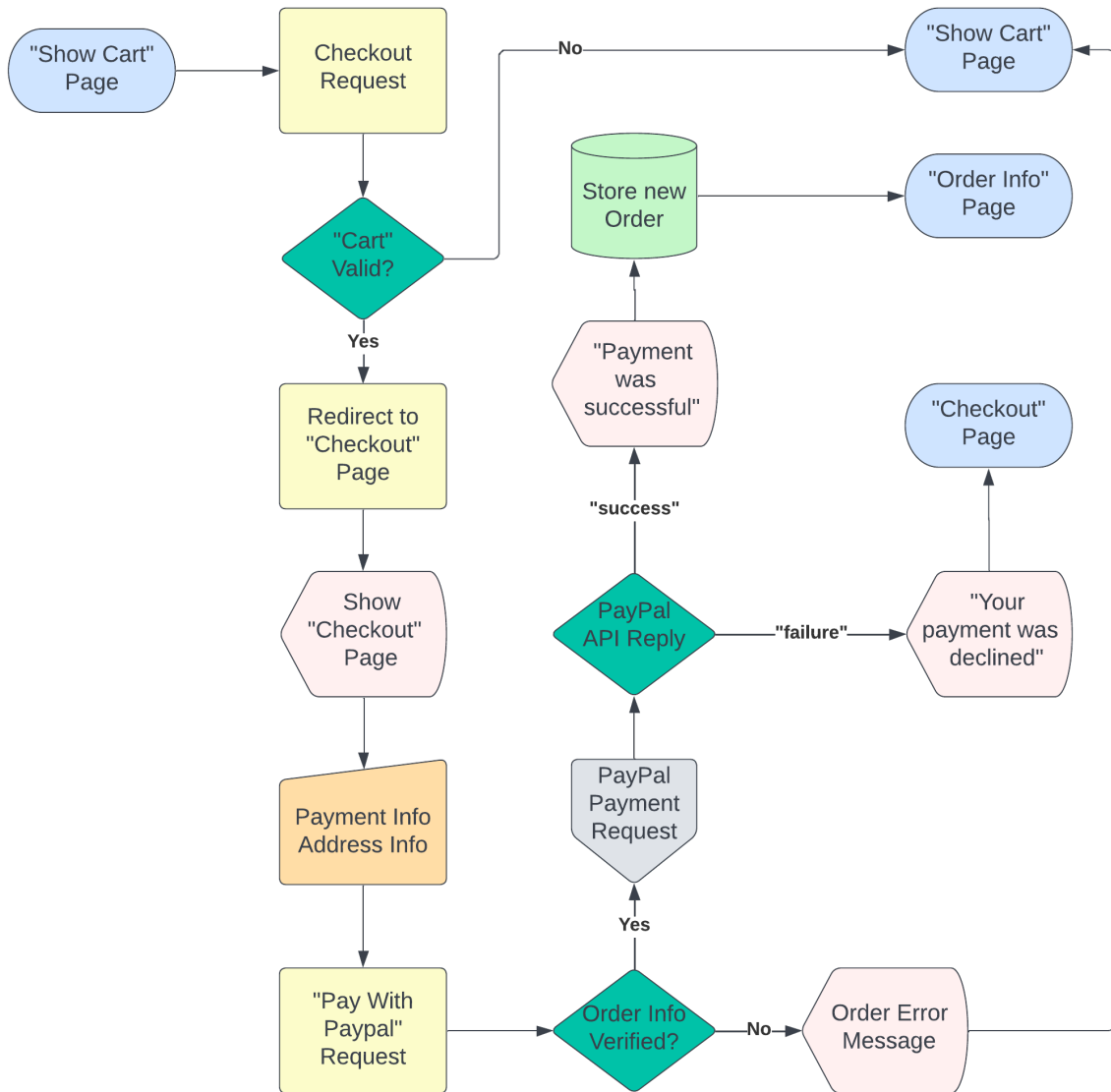
Εικόνα 3.7: Διαγραμματική Αναπαράσταση Διαδικασίας Προθήκης Προϊόντος στο Καλάθι Αγορών.

3.3.9 Ηλεκτρονική Πληρωμή

Στα πλαίσια της ενσωμάτωσης ηλεκτρονικών πληρωμών, σε πρώτη φάση υλοποιήθηκε ένα σύστημα πληρωμών μέσω PayPal. Την συγκεκριμένη μεθοδολογία απαρτίζουν τρεις συναρτήσεις:

- **handlePayment:** Συνάρτηση υπεύθυνη για τον χειρισμό των στοιχείων που απαρτίζουν μια παραγγελία. Αποδίδει τις κατάλληλες τιμές στα κατάλληλα πεδία, υπολογίζει το συνολικό ποσό που καλείται να πληρώσει ο χρήστης, καλεί το API της PayPal, και επιστρέφει μέσω σχετικής συνάρτησης το αποτέλεσμα ως JSON array.
- **paymentCancel:** Διαχειρίζεται το σενάριο ακύρωσης μιας πληρωμής. Η κλήση της εξαρτάται από την απάντηση του API της PayPal στο σχετικό αίτημα. Επιστρέφει ένα μήνυμα αποτυχίας στον χρήστη.
- **paymentSuccess:** Διαχειρίζεται το σενάριο επιτυχούς ολοκλήρωσης μιας πληρωμής. Η κλήση της εξαρτάται από την απάντηση του API της PayPal στο σχετικό αίτημα. Επιστρέφει ένα μήνυμα επιτυχίας στον χρήστη, εάν δεν παρουσιαστεί κάποιο σφάλμα κατά την λήψη της απάντησης στο αίτημα προς το API.

Ο κώδικας σε PHP - Laravel που υλοποιεί την παραπάνω λειτουργία συμπεριλαμβάνεται στο Παράρτημα Π.Α.18: PayPal Integration. Στα πεδία των μεταβλητών έχουν αποδοθεί ενδεικτικές τιμές (placeholder values). Επιπλέον, στην Εικόνα 3.8 οπτικοποιείται μέσω ενός διαγράμματος ροής η συνολική διαδικασία που ακολουθεί ο χρήστης στο σύστημα για την αποπληρωμή και αποστολή κατ' οίκον της παραγγελίας του.



Εικόνα 3.8: Διαγραμματική Αναπαράσταση Διαδικασίας Ολοκλήρωσης Παραγγελίας - "Checkout"

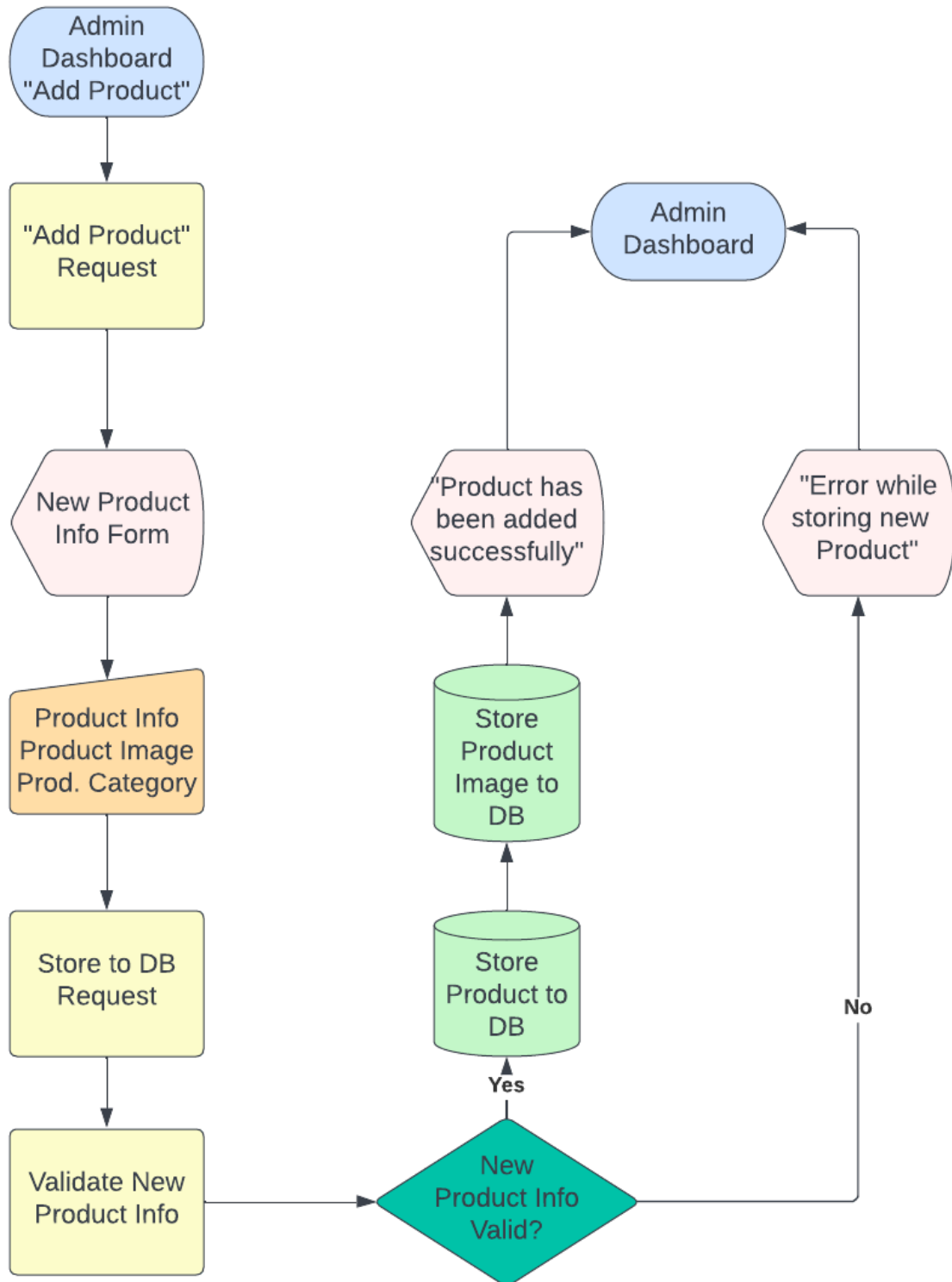
3.3.10 Προϊόντα & Κατηγορίες

Η διαχείριση των προϊόντων και των κατηγοριών στις οποίες υπάγονται είναι αποκλειστικό δικαίωμα των εγγεγραμμένων στο σύστημα διαχειριστών. Ο εκάστοτε διαχειριστής δύναται να δημιουργήσει, να τροποποιήσει και να διαγράψει προϊόντα και κατηγορίες.

Σε αυτή την παράγραφο αναλύονται οι μέθοδοι που υλοποιούν τα προαναφερθέντα και για τις δύο περιπτώσεις.

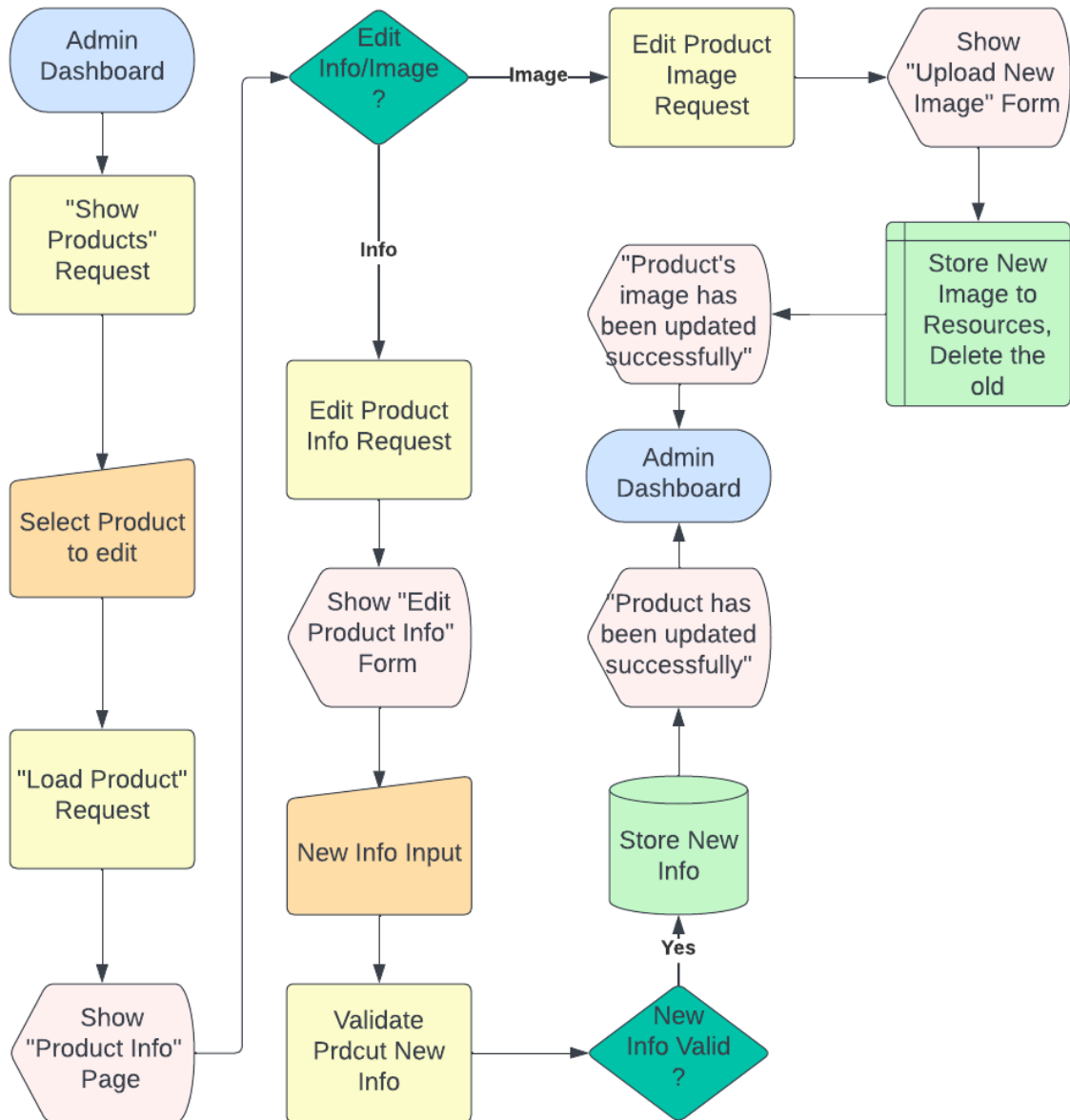
Διαχείριση Προϊόντων

Η δημιουργία μιας καταχώρησης προϊόντος στο σύστημα υλοποιείται μέσω μίας μεθόδου δύο συναρτήσεων. Η πρώτη εκ των δύο συναρτήσεων είναι υπεύθυνη για την υλοποίηση της φόρμας μέσω της οποίας ο χρήστης παρέχει τις τιμές που ανατίθενται στις μεταβλητές που περιγράφουν το εκάστοτε object Product. Τα κύρια στοιχεία είναι το αναγνωριστικό του προϊόντος, το όνομά του και η φωτογραφία του στο σύστημα. Η δεύτερη εκ των δύο συναρτήσεων αναλαμβάνει την εξυπηρέτηση του αιτήματος για καταχώρηση του προϊόντος στο σύστημα. Συγκεκριμένα, επικυρώνει τα δεδομένα της φόρμας και μέσω αυτών δημιουργεί το αντίστοιχο object Product. Με την επιτυχή καταχώρησή του στο σύστημα, επιστρέφει ανάλογο μήνυμα επιτυχούς εκτέλεσης. Στο Παράρτημα Π.Α.19: Add New Product παρουσιάζεται ο κώδικας που υλοποιεί την παραπάνω μέθοδο σε PHP - Laravel. Στην Εικόνα 3.9 παρουσιάζεται μέσω ενός διαγράμματος ροής η ολική διαδικασία που εκτελείται στο σύστημα μέσω της αλληλεπίδρασης αυτού και του διαχειριστή για την προσθήκη ενός νέου προϊόντος.

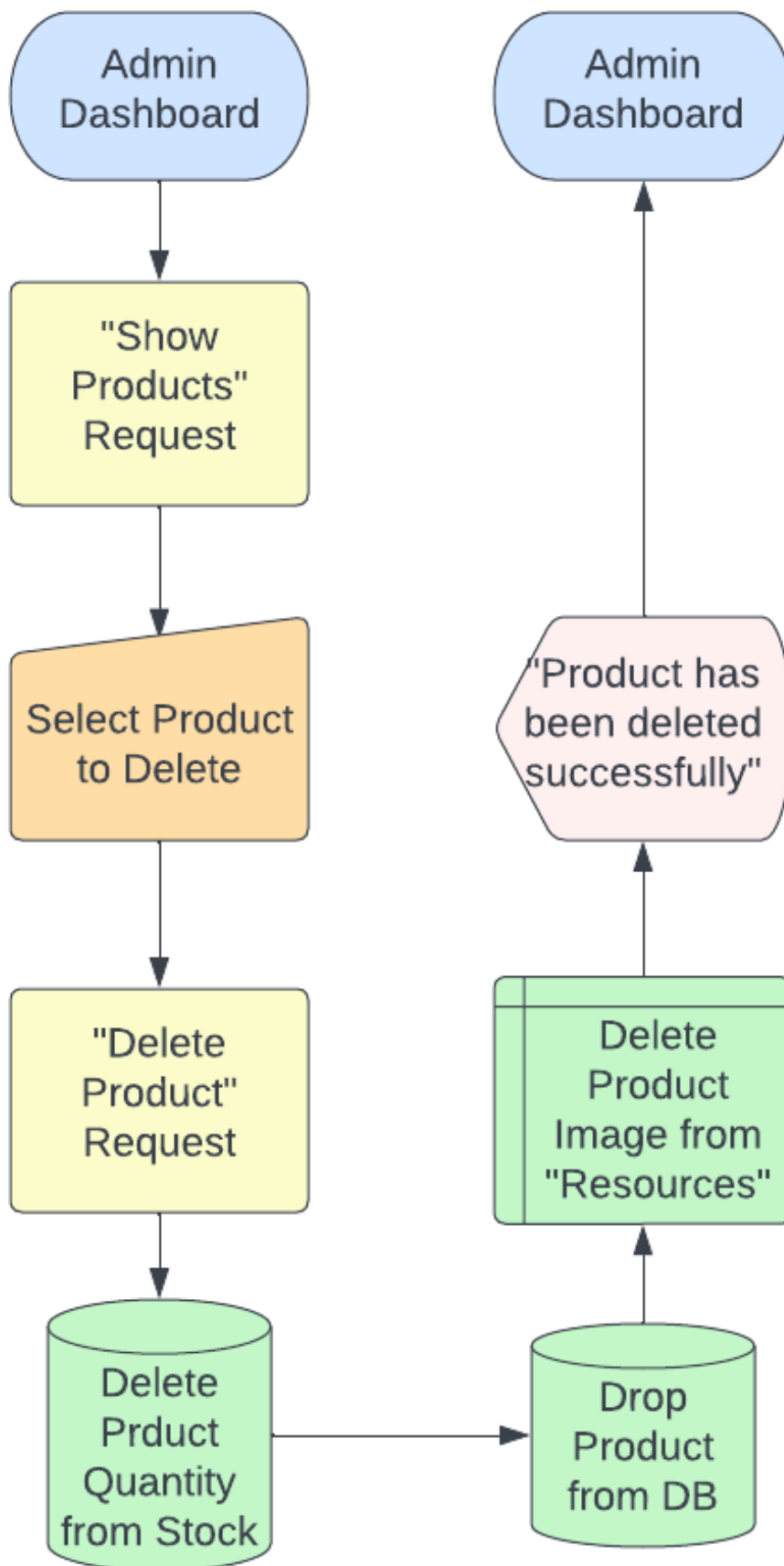


Εικόνα 3.9: Διαγραμματική Αναπαράσταση Διαδικασίας Προσθήκης Νέου Προϊόντος στο Σύστημα

Για την επεξεργασία μιας καταχώρησης ενός προϊόντος στο ηλεκτρονικό κατάστημα, υλοποιήθηκε μία μέθοδος τεσσάρων επιμέρους συναρτήσεων. Η πρώτη αναλαμβάνει την διαχείριση της φόρμας της επεξεργασίας του προϊόντος: φορτώνει το αιτηθέν προϊόν σαν object από το σύστημα μέσω του αναγνωριστικού και του ονόματός του, την κατηγορία του, και δέχεται τις νέες πληροφορίες από τον χειριστή του συστήματος. Η δεύτερη αναλαμβάνει την εκτέλεση του αιτήματος επεξεργασίας του συγκεκριμένου προϊόντος, ανανεώνει τις πληροφορίες στην Βάση Δεδομένων, και επιστρέφει μήνυμα επιτυχίας στον χρήστη στην ανάλογη περίπτωση. Η τρίτη είναι υπεύθυνη για την ανανέωση της φωτογραφίας που απεικονίζει το προϊόν: δέχεται το καινούριο αρχείο εικόνας, εκτελεί έναν έλεγχο για την επικύρωση των διαστάσεών του, διαγράφει την παλιά εικόνα, καταχωρεί την καινούρια στο σύστημα και την αναθέτει στο προϊόν, και με την επιτυχή ολοκλήρωση της διαδικασίας επιστρέφει το ανάλογο μήνυμα στο χρήστη. Τέλος, υλοποιείται και μία συνάρτηση που αφορά τη διαγραφή ενός προϊόντος από το σύστημα, η οποία αρχικά διαγράφει την εικόνα του από τους πόρους του συστήματος, και έπειτα καταστρέφει το object. Στο Παράρτημα Π.Α.20: Edit Product παρουσιάζεται ο κώδικας που υλοποιεί την παραπάνω μέθοδο σε PHP - Laravel. Στην Εικόνα 3.10 παρουσιάζεται μέσω ενός διαγράμματος ροής η ολική διαδικασία που εκτελείται στο σύστημα μέσω της αλληλεπίδρασης αυτού και του χειριστή για την επεξεργασία των πληροφοριών ενός υφιστάμενου προϊόντος. Αντίστοιχα, στην Εικόνα 3.11 παρουσιάζεται η διαδικασία διαγραφής ενός προϊόντος από το σύστημα, κατ' εντολή του χειριστή.



Εικόνα 3.10: Διαγραμματική Αναπαράσταση Διαδικασίας Επεξεργασίας Προϊόντος στο Σύστημα



Εικόνα 3.11: Διαγραμματική Αναπαράσταση Διαδικασίας Διαγραφής Προϊόντος στο Σύστημα

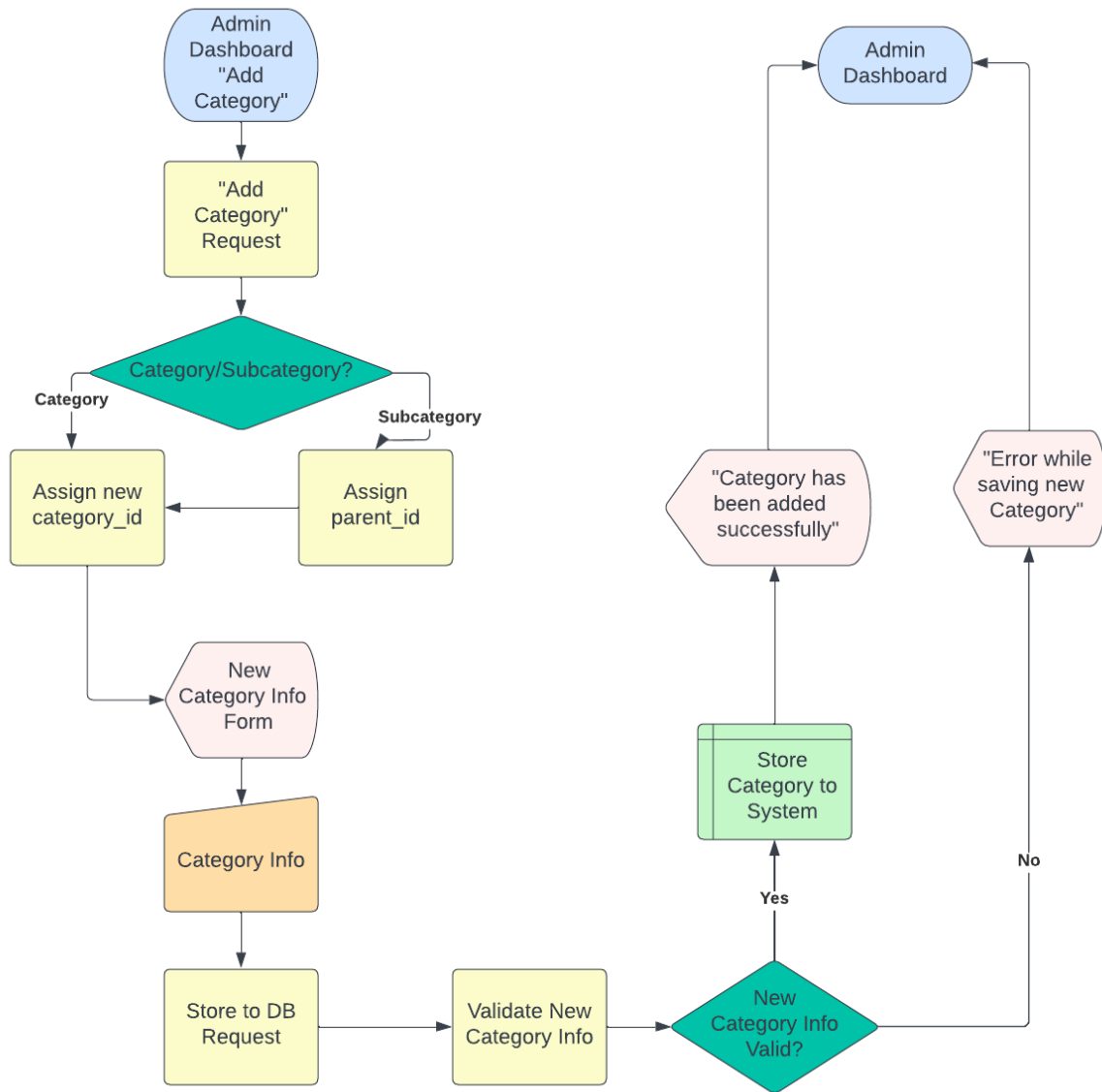
Τέλος, υλοποιήθηκε μια μέθοδος προβολής όλων των προϊόντων του καταστήματος στον χρήστη για λόγους χρηστικότητας και άμεσου εκμετάλλευσης της πληροφορίας που παρέχει η Βάση Δεδομένων. Η μέθοδος απαρτίζεται από δύο συναρτήσεις: μία που προβάλλει όλα τα καταχωρημένα προϊόντα, και μία δεύτερη που αναλαμβάνει την αναζήτηση συγκεκριμένου προϊόντος, ή ενός εύρους αυτών.

Η πρώτη εκ των δύο προβάλλει όλα τα προϊόντα του καταστήματος ανά Κατηγορία, ενώ επιπλέον παρουσιάζει τις συνολικές καταγραφές του εκάστοτε προϊόντος σε Καλάθια Αγορών και σε λίστες “Αγαπημένων”. Η δεύτερη αναλαμβάνει την εκτέλεση του αιτήματος για αναζήτηση προϊόντος/προϊόντων στην Βάση Δεδομένων. Ανακτά τα προϊόντα που πληρούν τα κριτήρια του αιτήματος του χρήστη, τα οργανώνει ανά κατηγορίες, και τέλος, και σε αυτή την περίπτωση, αναγράφει σε πόσες λίστες “Αγαπημένων” και Καλάθια αγορών περιλαμβάνεται το καθένα από αυτά. Επιστρέφει στον χρήστη τα αποτελέσματα της αναζήτησής του μέσω γραφικού περιβάλλοντος. Στο Παράρτημα Π.Α.21: Show Product παρουσιάζεται ο κώδικας που υλοποιεί την παραπάνω μέθοδο σε PHP - Laravel.

Διαχείριση Κατηγοριών

Για την υλοποίηση της λειτουργικότητας της δημιουργίας μιας Κατηγορίας προϊόντος χρησιμοποιούνται δύο συναρτήσεις.

Η πρώτη δημιουργεί την Κατηγορία ως αντικείμενο της κλάσης Category και το επιστρέφει. Εάν η κατηγορία υπάγεται σε άλλη, η συνάρτηση αναθέτει στο αντικείμενο το αναγνωριστικό της “μητρικής” κατηγορίας. Ειδιάλλως, αναθέτει ένα καινούριο. Η διαδικασία διενεργείται από τον διαχειριστή από την πλευρά του front-end μέσω σχετικής φόρμας. Η δεύτερη συνάρτηση είναι υπεύθυνη για την καταχώρηση της καινούριας κατηγορίας στην Βάση Δεδομένων. Ο κώδικας της μεθόδου σε PHP - Laravel παρουσιάζεται στο Παράρτημα Π.Α.22: Add New Category. Στην Εικόνα 3.12 παρουσιάζεται η διαδικασία που διενεργείται κατ’ εντολή του διαχειριστή του συστήματος για την προσθήκη μιας νέας κατηγορίας στο σύστημα, μέσω ενός διαγράμματος ροής.



Εικόνα 3.12: Διαγραμματική Αναπαράσταση Διαδικασίας Προσθήκης Νέας Κατηγορίας στο Σύστημα

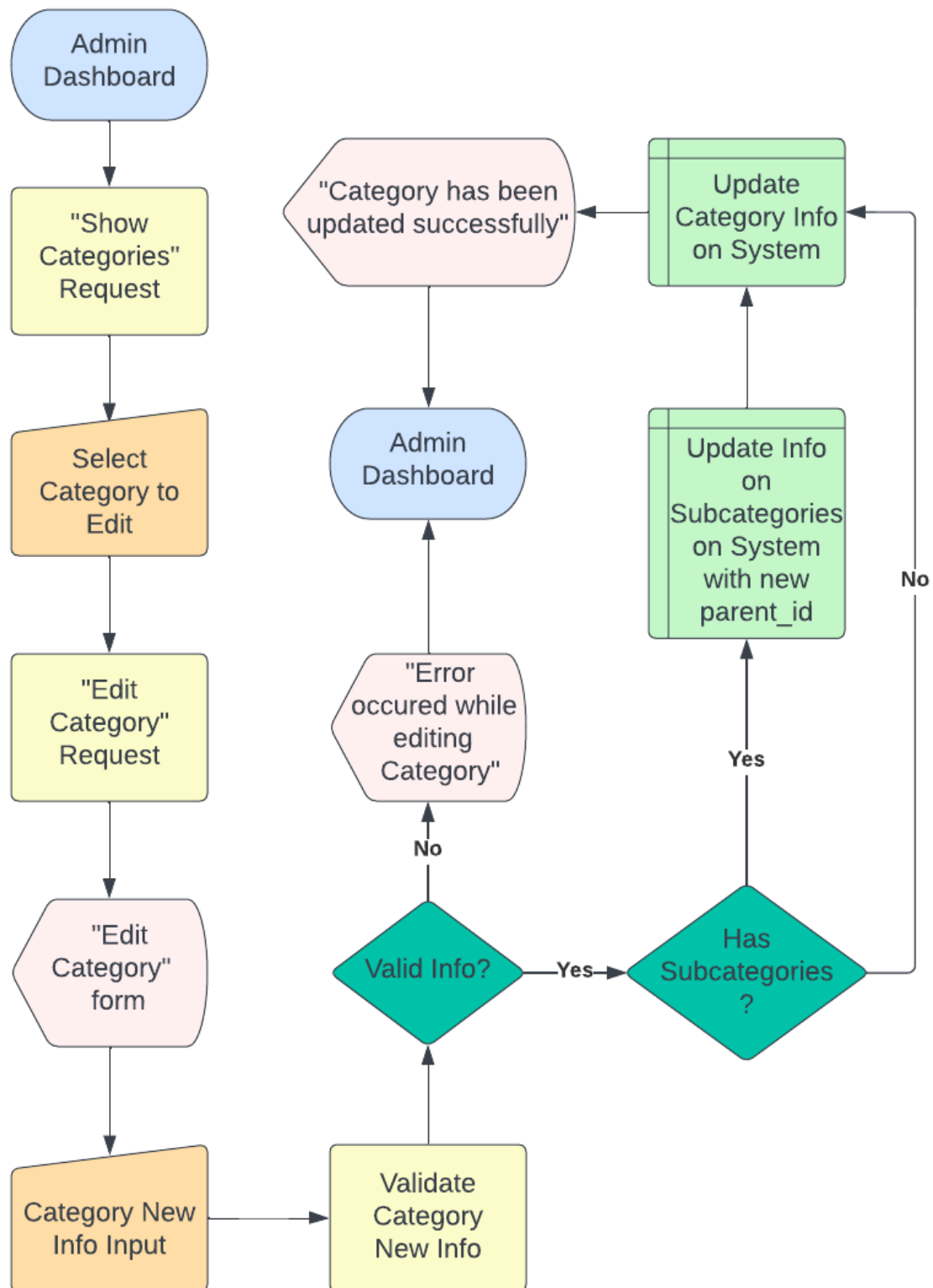
Η τροποποίηση, η ενημέρωση και η διαγραφή μιας κατηγορίας διενεργούνται μέσω αιτημάτων από μία μέθοδο με τρεις διαφορετικές συναρτήσεις.

Η πρώτη, με δοθέν όρισμα το αναγνωριστικό της κατηγορίας, αναλαμβάνει την τροποποίηση μιας κατηγορίας. Η ενέργεια εκτελείται από τον διαχειριστή από πλευράς front-end μέσω σχετικής φόρμας.

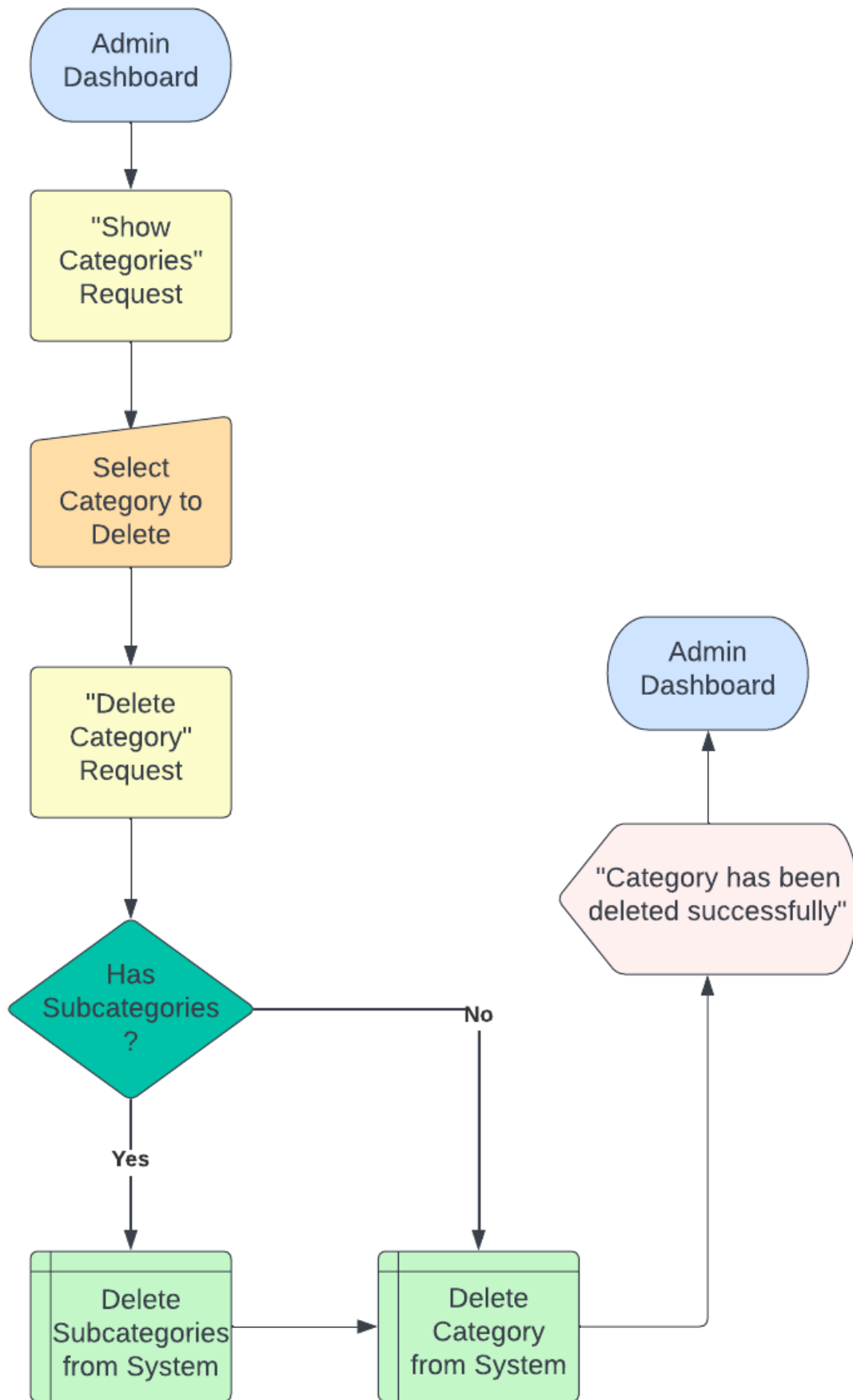
Με την αποστολή του παραπάνω αιτήματος του διαχειριστή στο back-end, τίθεται σε λειτουργία η δεύτερη συνάρτηση, η οποία είναι υπεύθυνη για την ενημέρωση των τιμών των μεταβλητών που χαρακτηρίζουν την προκείμενη κατηγορία υπό επεξεργασία και την καταχώρηση των πληροφοριών αυτών στη Βάση Δεδομένων.

Στην περίπτωση αιτήματος διαγραφής της κατηγορίας υπό εξέταση, αναλαμβάνει δράση η τρίτη συνάρτηση, που αποστέλλει ένα σχετικό αίτημα διαγραφής στη Βάση Δεδομένων.

Ο κώδικας της μεθόδου σε PHP - Laravel παρουσιάζεται στο Παράρτημα Π.Α.23: Edit Category. Οι διαδικασίες που εκτελούνται στο σύστημα για την τροποποίηση/ενημέρωση ή διαγραφή μιας υφιστάμενης Κατηγορίας παρουσιάζονται μέσω διαγραμμάτων ροής στις Εικόνες 3.13 και 3.14 αντίστοιχα.



Εικόνα 3.13: Διαγραμματική Αναπαράσταση Διαδικασίας Τροποποίησης μιας Κατηγορίας στο Σύστημα



Εικόνα 3.14: Διαγραμματική Αναπαράσταση Διαδικασίας Διαγραφής μιας Κατηγορίας στο Σύστημα

Τέλος, υλοποιούνται δύο συναρτήσεις που αποσκοπούν, κατόπιν αιτήματος, στην προβολή όλων των Κατηγοριών στον διαχειριστή, ή στην προβολή μόνο κάποιας συγκεκριμένης.

Ο κώδικας της μεθόδου σε PHP - Laravel παρουσιάζεται στο Παράρτημα Π.Α.24: Show Vategory.

3.3.11 Παραγγελίες

Η διαχείριση των παραγγελιών εμπίπτει εν μέρη και στα δικαιώματα εγγεγραμμένων και συνδεδεμένων στο σύστημα Χρηστών ("User"), αλλά και στα δικαιώματα των καταχωρημένων στο σύστημα χρηστών ως Διαχειριστές ("Admin"): ένας "User" δύναται αποκλειστικά να δημιουργήσει μια παραγγελία, ενώ η τροποποίησή της και η προβολή όλων των υπαρχουσών παραγγελιών εμπίπτει αποκλειστικά στα δικαιώματα του "Admin".

Η δημιουργία μιας παραγγελίας προϋποθέτει αρχικά την ύπαρξη προϊόντων στο Καλάθι Αγορών. Η μέθοδος που υλοποιήθηκε για τον σκοπό αυτό έχει ουσιαστικό στόχο την καταχώρηση μιας παραγγελίας στο σύστημα, και απαρτίζεται από δύο συναρτήσεις.

Η πρώτη αφορά την φόρμα επιβεβαίωσης Αποστολής Παραγγελίας στο σύστημα. Αρχικά, ελέγχεται η εγκυρότητα του Καλαθιού Αγορών και των προϊόντων που περιέχει. Έπειτα, υπολογίζεται το τελικό κόστος μεταφορικών εξόδων της αποστολής το οποίο καλείται να καλύψει ο πελάτης, όπως επίσης και το συνολικό αντίτιμο των προϊόντων τα οποία επιθυμεί να αγοράσει. Η συνάρτηση επιστρέφει στον πελάτη την αντίστοιχη φόρμα επιβεβαίωσης, η οποία εμπεριέχει σχετικό πλήκτρο "Αποστολής" - οριστικοποίησης της παραγγελίας.

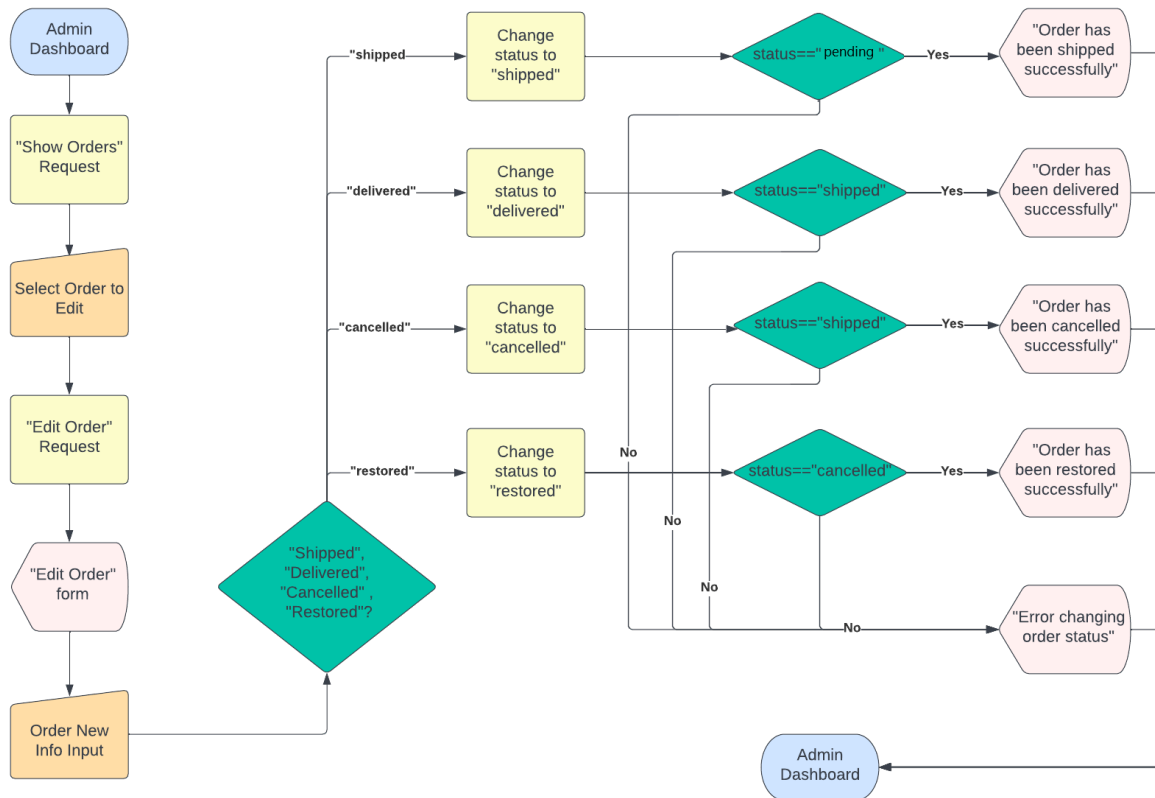
Η δεύτερη συνάρτηση αφορά την καταχώρηση της απεσταλμένης παραγγελίας στο σύστημα. Ουσιαστικά, διαχειρίζεται την εκτέλεση του αντίστοιχου αιτήματος προς την Βάση Δεδομένων. Αναθέτει ένα νέο αναγνωριστικό στην εγγραφή της παραγγελίας, συνδέει την καταχώρηση αυτή με το αναγνωριστικό του πελάτη ο οποίος την αιτήθηκε και συμπεριλαμβάνει στις πληροφορίες αυτές τα μεταφορικά έξοδα. Τέλος, επιβεβαιώνοντας ξανά την εγκυρότητα του τελικού Καλαθιού Αγορών, επιστρέφει μήνυμα επιτυχούς ολοκλήρωσης της διαδικασίας στον χρήστη.

Στο Παράρτημα Π.Α.25: Add New Order παρουσιάζεται ο κώδικας της μεθόδου σε PHP - Laravel.

Η επεξεργασία μιας παραγγελίας αποτελεί καθήκον του διαχειριστή της ιστοσελίδας. Η διαδικασία αυτή περιλαμβάνει την επιβεβαίωση αποστολής μιας παραγγελίας στον πελάτη ("shipped"), την επιβεβαίωση παράδοσης μιας απεσταλμένης παραγγελίας ("delivered"), την ακύρωση μιας παραγγελίας σε περίπτωση που παρουσιαστεί κάποιο ζήτημα κατά την διαδικασία αποστολής της ("cancelled") και, τέλος, την επαναφορά της στο σύστημα κατόπιν ακύρωσής της εάν ο πελάτης το επιθυμεί ("restored").

Για την διαδικασία που περιγράφεται παραπάνω υλοποιείται η αντίστοιχη μέθοδος, η οποία απαρτίζεται από τέσσερις συναρτήσεις που φέρουν σε πέρας τις ανάγκες των προαναφερθέντων

περιπτώσεων επεξεργασίας μιας παραγγελίας. Σκοπός των συναρτήσεων αυτών είναι η διεκπεραίωση των αντίστοιχων αιτημάτων προς την Βάση Δεδομένων. Στο Παράρτημα Π.Α.26: Edit Order παρουσιάζεται ο κώδικας της μεθόδου σε PHP - Laravel. Μέσω του διαγράμματος ροής της Εικόνας 3.15 οπτικοποιείται η συνολική διαδικασία, όπως αυτή διενεργείται από τον διαχειριστή, για την ενημέρωση της κατάστασης μιας παραγγελίας στο σύστημα.



Εικόνα 3.15: Διαγραμματική Αναπαράσταση Διαδικασίας Ενημέρωσης Κατάστασης μιας Παραγγελίας στο Σύστημα

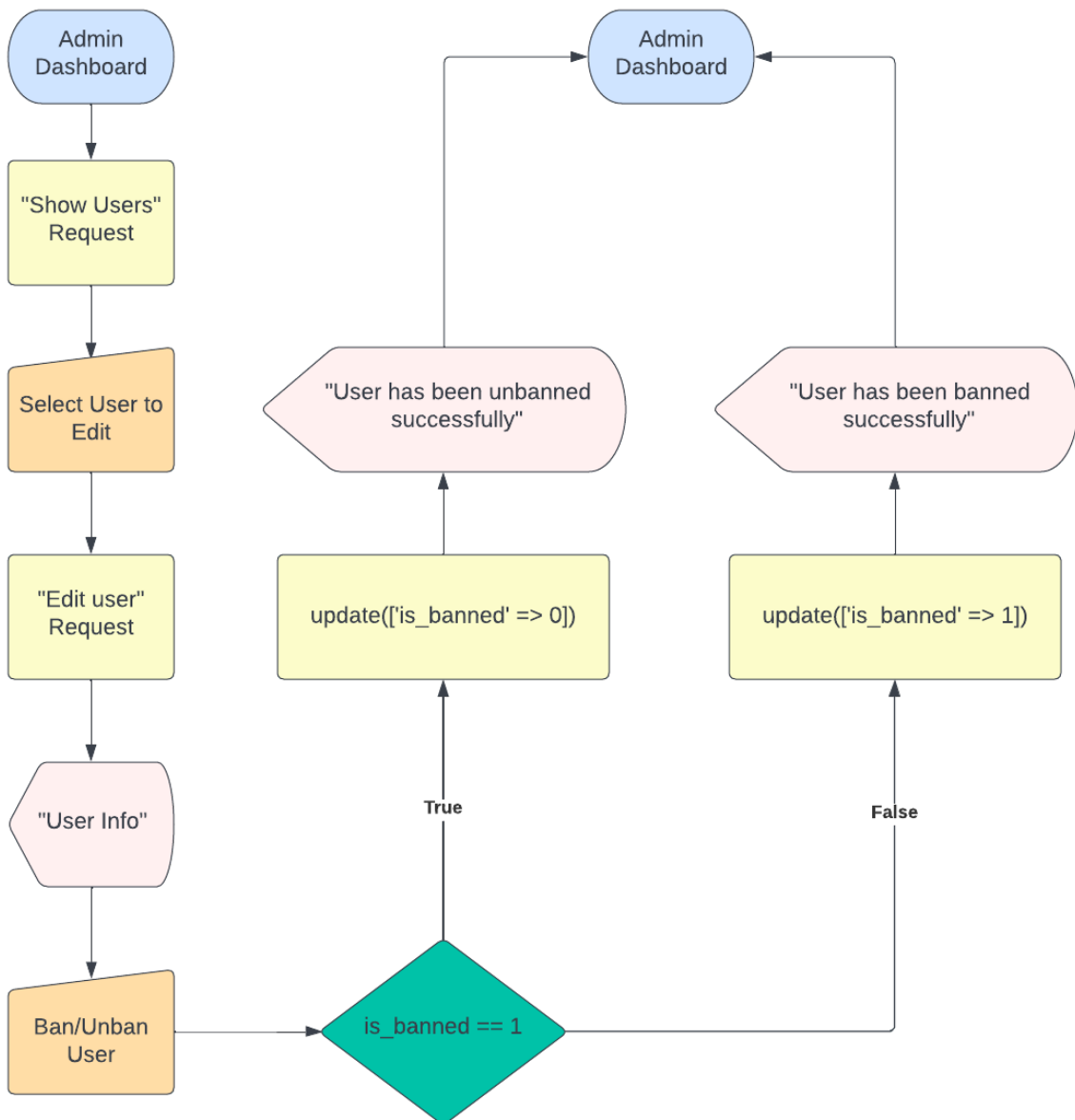
Τέλος, για τον σκοπό της προβολής μίας ή όλων των παραγγελιών υλοποιείται μία ακόμα μέθοδος δύο αντίστοιχων συναρτήσεων, η οποία εμπίπτει, φυσικά, αποκλειστικά στα δικαιώματα του διαχειριστή της ιστοσελίδας. Η πρώτη συνάρτηση αφορά αποστολή ενός αιτήματος στην Βάση Δεδομένων για ανάκτηση και προβολή όλων των παραγγελιών του συστήματος, μαζί με την κατάστασή τους ("pending", "shipped", "delivered", "cancelled", "restored") καθώς και την αξία τους. Η δεύτερη αφορά την ανάκτηση από τη Βάση Δεδομένων συγκεκριμένης παραγγελίας. Επιστρέφει τα προϊόντα που συμπεριλαμβάνονται σε αυτή, τον πελάτη που την απέστειλε, και την συνολική της αξία.

Στο Παράρτημα Π.Α.27: Show Order παρουσιάζεται ο κώδικας σε PHP - Laravel που περιγράφει την μέθοδο που περιγράφηκε άνωθεν.

3.3.12 Διαχείριση Λογαριασμών Χρηστών

Η διαχείριση των λογαριασμών των Χρηστών του συστήματος αποτελεί ευθύνη και εμπίπτει αποκλειστικά στα δικαιώματα του Διαχειριστή της ιστοσελίδας. Στον διαχειριστή δίνεται η δυνατότητα αποκλεισμού ενός χρήστη από το σύστημα (ή/και άρσης του αποκλεισμού αυτού), όπως επίσης και η δυνατότητα προσπέλασης και προβολής όλων των χρηστών του συστήματος.

Για την πρώτη περίπτωση, δημιουργείται μια μέθοδος αποτελούμενη από μία συνάρτηση αποκλεισμού ή άρσης του αποκλεισμού ενός χρήστη από το σύστημα. Η συνάρτηση αυτή λειτουργεί με πολύ απλό τρόπο: κατόπιν κλήσης της, εναλλάσσει την τιμή της μεταβλητής "is_banned" της κλάσης "User" από 0 σε 1 για αποκλεισμό, και αντίστροφα για άρση του αποκλεισμού ("toggling"). Ανάλογα την περίπτωση, και το αποτέλεσμα που συνεπάγεται, επιστρέφει στον χρήστη-διαχειριστή το αντίστοιχο μήνυμα επιτυχίας. Ο κώδικας σε PHP - Laravel που υλοποιεί τη μέθοδο αυτή παρουσιάζεται στο Παράρτημα Π.Α.28: Ban/Unban User. Η σχετική διαδικασία αναπαρίσταται γραφικά μέσω ενός διαγράμματος ροής στην Εικόνα 3.16.



Εικόνα 3.16: Διαγραμματική Αναπαράσταση Διαδικασίας Αποκλεισμού/Επαναφοράς Χρήστη από το/στο Σύστημα

Στην περίπτωση που παρουσιάζεται ανάγκη χειροκίνητης - επιβεβλημένης - αλλαγής στοιχείων λογαριασμού ενός χρήστη, πέραν τις επαναφοράς του κωδικού του ή των δυνατοτήτων που δίνονται στον ίδιο για επεξεργασία των στοιχείων του (πρβλ. παρ. 3.2.4 και 3.2.5 αντίστοιχα), ή της επιβεβλημένης διαγραφής ενός λογαριασμού, ο διαχειριστής δύναται να το κάνει μέσω απευθείας αίτηματος στην Βάση Δεδομένων (MySQL query). Για λόγους ασφαλείας, αφενός δεν υλοποιούνται μέθοδοι σε PHP - Laravel, αφετέρου δεν παρουσιάζεται εντός του παρόντος εγγράφου της πτυχιακής εργασίας το query σε MySQL, προς αποφυγή αποκάλυψης της δομής της Βάσης Δεδομένων.

Όσον αφορά την προσπέλαση και προβολή των λογαριασμών των χρηστών από τον διαχειριστή του ιστοτόπου, υλοποιείται μία μέθοδος δύο συναρτήσεων. Η πρώτη, διεκπεραιώνοντας το αντίστοιχο αίτημα προς τη Βάση Δεδομένων, επιστρέφει στον διαχειριστή όλους τους λογαριασμούς των εγγεγραμμένων στο σύστημα χρηστών (πλην των λογαριασμών με την ιδιότητα του διαχειριστή ("Admin")). Η δεύτερη επιστρέφει τα στοιχεία συγκεκριμένου εγγεγραμμένου χρήστη του συστήματος, μαζί με τις παραγγελίες που έχει κάνει, όπως επίσης και το κάθε προϊόν που έχει αγοράσει μαζί με την ποσότητά του.

Ο κώδικας που υλοποιεί την συγκεκριμένη μέθοδο σε PHP - Laravel παρουσιάζεται στο Παράρτημα Π.Α.29: Show User Info:

3.4 Βάση Δεδομένων - Πρόγραμμα Μηχανογράφησης

Η ειδοποιός διαφορά του μοντέλου του ηλεκτρονικού καταστήματος που πραγματεύεται αυτή η πτυχιακή εργασία είναι η διασύνδεσή του με ένα πρόγραμμα μηχανογράφησης. Ενώ στην προηγούμενη παράγραφο περιγράφηκε αναλυτικά η λειτουργική υποδομή του ηλεκτρονικού καταστήματος και η υλοποίησή της, σε αυτή θα περιγραφεί ο τρόπος διασύνδεσης του ηλεκτρονικού καταστήματος με το εμπορικό, ήδη υφιστάμενο πρόγραμμα μηχανογράφησης το οποίο λειτουργεί και εξυπηρετεί το φυσικό κατάστημα πίσω από το ηλεκτρονικό εδώ και αρκετά χρόνια.

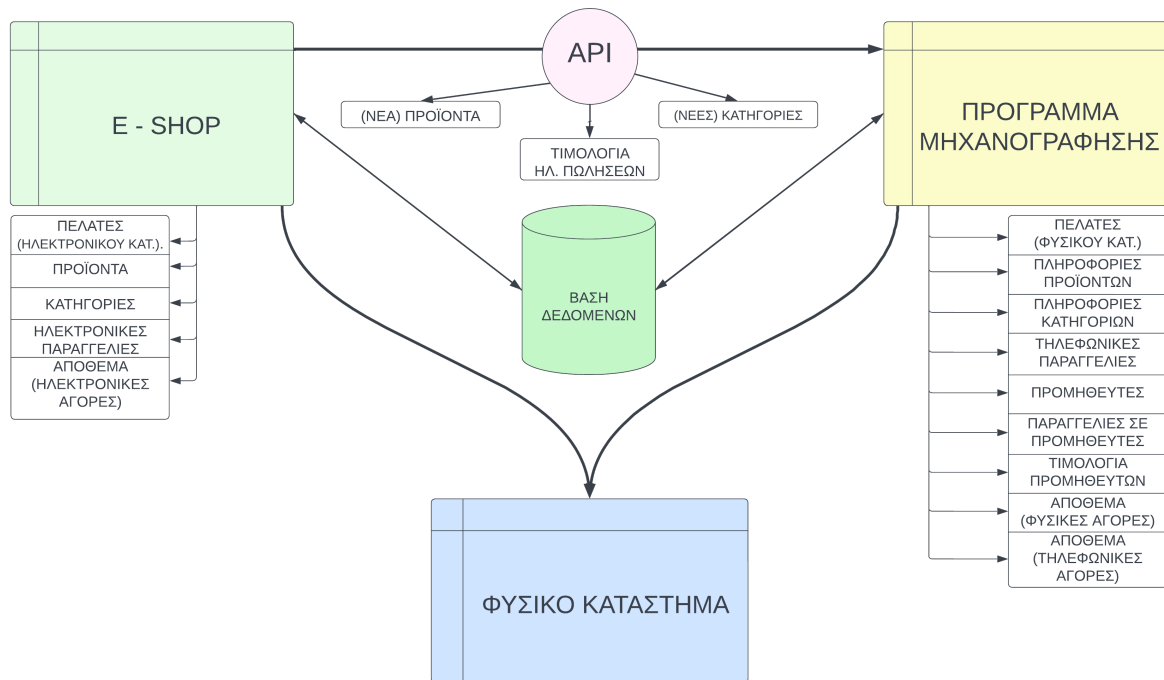
Δομικά, ο κοινός σύνδεσμος μεταξύ του ηλεκτρονικού καταστήματος και του προγράμματος μηχανογράφησης αποτελεί η Βάση Δεδομένων. Συνεπώς, μοιράζονται εν μέρη κάποια κομμάτια της ενιαίας λειτουργικότητας του συστήματος, αλλά εξακολουθούν να έχουν κάποια υπό αποκλειστική ευθύνη τους. Πιο συγκεκριμένα, το πρόγραμμα μηχανογράφησης έχει την δυνατότητα καταγραφής και διαχείρισης του αποθέματος της επιχείρησης, την αποθήκευση και διαχείριση τιμολογίων, την αποστολή παραγγελιών σε επιλεγμένους προμηθευτές και την καταγραφή και διαχείριση των προμηθευτών αυτών. Δεδομένου ότι το ηλεκτρονικό κατάστημα έχει την δυνατότητα πρόσβασης και επεξεργασίας του αποθέματος της επιχείρησης και την αποθήκευση παραγγελιών, μεριμνήσαμε για δύο πράγματα: αφενός την διατήρηση της μορφής και της δομής της Βάσης Δεδομένων υπό μοντέλο το οποίο πληροί τις απαιτήσεις του προγράμματος μηχανογράφησης και εξυπηρετεί τις ανάγκες της επιχείρησης, και αφετέρου, την υλοποίηση

ενός διαύλου επικοινωνίας μεταξύ του back-end του ηλεκτρονικού καταστήματος και του προγράμματος μηχανογράφησης.

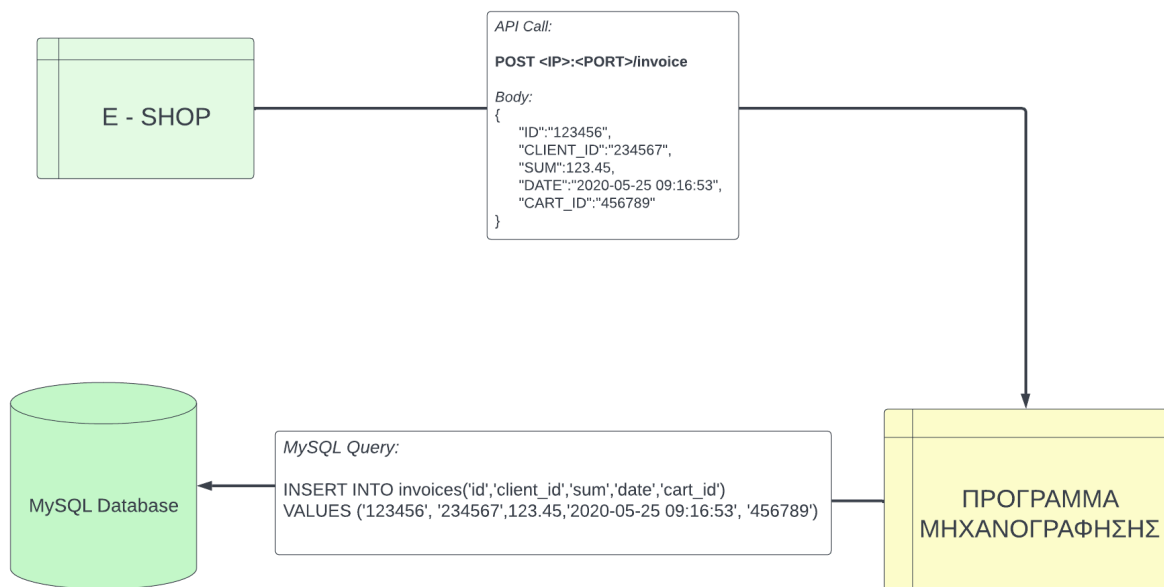
Ο διάυλος επικοινωνίας για τον οποίο γίνεται λόγος αποτελεί ένα API με συγκεκριμένα endpoints στο ηλεκτρονικό κατάστημα μέσω των οποίων μεταφέρεται η απαραίτητη πληροφορία μέσω αιτημάτων POST στο πρόγραμμα μηχανογράφησης. Το API αυτό υλοποιήθηκε ομοίως σε γλώσσα προγραμματισμού PHP - Laravel. Τα αιτήματα που περιλαμβάνει αφορούν κυρίως την αποστολή των δεδομένων μιας παραγγελίας (αναγνωριστικό πελάτη, προϊόντα και ποσότητες αυτών, ποσό πληρωμής, ποσό μεταφορικών εξόδων, ημερομηνία εκτέλεσης της συναλλαγής, κλπ.) υπό μορφή ενός JSON array. Η επικοινωνία αυτή κρυπτογραφείται μέσω πρωτοκόλλου SSL - TLS. Για λόγους ασφαλείας και διατήρησης της ακεραιότητας του συστήματος που αναπτύχθηκε, αποφασίστηκε ο μεθοδολογία κρυπτογράφησης, η μορφή των αιτημάτων, τα endpoints και οι IPv4 διευθύνσεις και τα σχετικά ports στα οποία “ακούει” και “μιλά” η κάθε πλευρά να μην περιγραφούν αναλυτικότερα, όπως επίσης και να μην παρουσιαστεί ο κώδικας που τα υλοποιεί.

Αν και στην συνέχεια αυτής της παραγράφου θα αναλυθούν εκτενώς οι δυνατότητες του προγράμματος μηχανογράφησης και της Βάσης Δεδομένων, μπορούμε σε αυτό το σημείο να καταλήξουμε πως, αν και το πρόγραμμα μηχανογράφησης έχει εμφανώς τις ίδιες (ή και περισσότερες) δυνατότητες από το e-shop, βασικός του σκοπός είναι η εξυπηρέτηση της λειτουργικότητας του φυσικού καταστήματος. Εν αντιθέσει, το e-shop αποσκοπεί περισσότερο στην αύξηση του πελατολογίου εφαρμόζοντας τεχνικές marketing και προωθώντας την επιχείριση σε μερίδια του αγοραστικού κοινού απρόσιτα από την φυσική της τοποθεσία. Σε κάθε περίπτωση, η αποτελεσματική, υβριδική λειτουργία των δύο εργαλείων είναι κομβική για την εμπορική ανάπτυξη και την λειτουργικότητα της επιχείρισης. Στην Εικόνα 3.17 δίνεται σχηματικά ένα σχέδιο της γενικής τεχνικής υποδομής του συστήματος διαχείρισης προϊόντων του καταστήματος, καθώς και της διασύνδεσης των οντοτήτων που το αποτελούν.

Στην Εικόνα 3.18 αναπαρίσταται διαγραμματικά ο τρόπος επικοινωνίας του e-shop με το Πρόγραμμα Μηχανογράφησης μέσω του API που σχεδιάστηκε. Συγκεκριμένα, παρουσιάζουμε ενδεικτικά την διαδικασία αποθήκευσης ενός τιμολογίου που προέκυψε από μια ηλεκτρονική αγορά. Οι πληροφορίες του τιμολογίου αποστέλλονται σε μορφή δεδομένων JSON array μέσω ενός POST API request στο αντίστοιχο endpoint (<IP>:<PORT>/invoice). Το API, από πλευράς Προγράμματος Μηχανογράφησης, με το που το λάβει, δρομολογεί ένα MySQL query στην MySQL Βάση Δεδομένων του συστήματος, όπου τοποθετεί τις τιμές που έλαβε από το e-shop στις αντίστοιχες στήλες του πίνακα ”invoices” της ΒΔ. Τα πεδία και οι τιμές πληροφοριών του τιμολογίου είναι καθαρά ενδεικτικά, και διαφέρουν, όπως θα δούμε στις επόμενες παραγράφους. Συμπεραίνουμε παρ’ όλα αυτά, πως σε αυτή τη φάση το Πρόγραμμα Μηχανογράφησης παίζει το ρόλο του “μεσάζοντα” σε αυτού του τύπου τις συναλλαγές, με σκοπό να διατηρεί τα δεδομένα σε μορφή λειτουργική για αυτό.



Εικόνα 3.17: Διαγραμματική Αναπαράσταση της συνολικής Τεχνικής Υποδομής της Επιχείρησης



Εικόνα 3.18: Διαγραμματική Αναπαράσταση Στιγμιότυπου Επικοινωνίας E-shop και Προγράμματος Μηχανογράφησης

3.4.1 Πρόγραμμα Μηχανογράφησης

Πέραν της προαναφερθείσας λειτουργικότητας που περιγράφηκε, το πρόγραμμα μηχανογράφησης δίνει την δυνατότητα διαχείρισης του αποθέματος της αποθήκης, του πελατολογίου, των προμηθευτών της επιχείρησης και των τιμολογίων που προκύπτουν από το εύρος των συναλλαγών τους. Με τον όρο διαχείριση υπονοείται η προβολή αυτών, και η χειροκίνητη τροποποίηση των τιμών των παραπάνω δεδομένων: αλλαγή στις τιμές των προϊόντων, αλλαγή στο απόθεμά τους, χειροκίνητη επιβεβλημένη αλλαγή στα στοιχεία του εκάστοτε πελάτη (πλην διεύθυνσης ηλεκτρονικού ταχυδρομείου και κωδικού πρόσβασης του λογαριασμού του), καταχώρηση εγγράφων που αφορούν τους προμηθευτές και τα στοιχεία τους, καταχώρηση επιπλέον δεδομένων στα είδη προϊόντων που παρέχονται από το κατάστημα, προβολή των τιμολογίων, κλπ. Τα προαναφερθέντα είναι μόνο μερικές από τις δυνατότητες που προσφέρει το πρόγραμμα μηχανογράφησης αυτό. Παρ' όλα αυτά, επειδή είναι ήδη σε εμπορική κυκλοφορία και η ανάπτυξή του δεν εμπίπτει στα πλαίσια αυτής της πτυχιακής εργασίας, θεωρήθηκε συνετό να μην αναπτύξουμε περαιτέρω επ' αυτού. Ωστόσο, στην συνέχεια παρουσιάζονται κάποια στιγμιότυπα αυτού από την “ζωντανή” λειτουργία του στο κατάστημα, προκειμένου ο αναγνώστης του παρόντος εγγράφου να έχει μια πιο καθαρή εικόνα για αυτό.

Στην Εικόνα 3.19 παρουσιάζεται στιγμιότυπο από το γραφικό περιβάλλον του προγράμματος μηχανογράφησης κατά την εκτέλεση της διαδικασίας έκδοσης τιμολογίου - απόδειξης αγοράς από το φυσικό κατάστημα της επιχείρησης (ταμειακή μηχανή). Το αποτέλεσμα της διαδικασίας αυτής προστίθεται στην Βάση Δεδομένων με το πέρας της.

Στην Εικόνα 3.20 παρουσιάζεται στιγμιότυπο από το πρόγραμμα μηχανογράφησης από την “χειροκίνητη” καταχώρηση νέου πελάτη στο κατάστημα. Αυτή η περίπτωση απαντάται στην περίπτωση που ο πελάτης συναλλάσσεται πρώτη φορά είτε τηλεφωνικώς είτε δια ζώσης με το φυσικό κατάστημα. Το αποτέλεσμα της διαδικασίας αυτής καταχωρείται στη Βάση Δεδομένων, και είναι παρόμοιο με την Δημιουργία Λογαριασμού Πελάτη στο ηλεκτρονικό κατάστημα. Κατ' αυτόν τον τρόπο, το σύστημα δύναται να καταχωρεί την ίδια πληροφορία από όποια πηγή και αν προέρχεται στην ίδια πληροφοριακή υποδομή.

Παρόμοια διαδικασία ακολουθείται και για την καταχώρηση στοιχείων νέου προμηθευτή, φυσικά με διαφορετικές επιμέρους μεταβλητές.

Κεφάλαιο 3

Εικόνα 3.19: Έκδοση τιμολογίου από το φυσικό κατάστημα - ταμειακή μηχανή, και προσθήκη στην Βάση Δεδομένων.

Εικόνα 3.20: Καταχώρηση στοιχείων νέου πελάτη.

Στις Εικόνες 3.21 και 3.22 παρουσιάζεται ένα στιγμιότυπο από το πρόγραμμα μηχανογράφησης κατά την εισαγωγή ενός νέου προϊόντος στο σύστημα. Στην πρώτη προβάλλεται η καρτέλα καταχώρησης των βασικών πληροφοριών που αφορούν το προϊόν, ενώ στην δεύτερη συμπληρώνονται στοιχεία που αφορούν την καταχώρηση και προβολή του προϊόντος στο ηλεκτρονικό κατάστημα. Τα δεδομένα που παράγονται από το πέρας της διαδικασίας προστίθενται στον ανάλογο directory του εξυπηρετητή που φιλοξενεί το ηλεκτρονικό κατάστημα, προκειμένου το backend του e-shop να έχει πρόσβαση σε αυτά. Παρόμοια είναι και η αντίστοιχη διαδικασία που ακολουθείται από πλευράς e-shop για την καταχώρηση νέων προϊόντων. Και πάλι σε αυτή την περίπτωση, το μοντέλο με το οποίο αναπτύσσεται η λειτουργικότητα δίνει την δυνατότητα στην επιχείριση να εκμεταλλεύεται και να διαχειρίζεται πληροφορία από διαφορετικές πηγές στην ίδια πληροφοριακή υποδομή.

Κεφάλαιο 3

Βασικά Στοιχεία

Κωδικός: 02630 A/A: -1 Ενεργό:

Περιγραφή: BarCode:

Μον. Μέτρησης: + ΤΕΜΑΧΙΑ ΦΠΑ: 24 %

Τιμή Χονδρικής: 0,0000 Τύπος: Εμπόρευμα

Τιμή Λιανικής: 0,0000 Τελ. Τιμή Αγοράς: 0,0000 Κέρδος: 0,00%

Ομάδα: + Κατηγορία: + Υποκατηγορία: +

Υπόλοιπο Αποθήκης: - Στάκ: - Ενεργοποίηση Στάκ

Κωδ Κατασκευαστή:

ΑΑΔΕ Ηλεκτρονικά Βιβλία

ΑΑΔΕ Χαρακτηρισμός Εσόδου: Πωλήσεις αγαθών και υπηρεσιών Χονδρικές - Εμπη... ΑΑΔΕ Χαρακτηρισμός Εξόδου: Αγορές εμπορευμάτων χρήσης (καθαρό ποσό)/Λογ...

ΑΑΔΕ Κατηγορία Εξέτασης ΦΠΑ: Χωρίς ΦΠΑ - άρθρο 39α του Κώδικα ΦΠΑ

Παρατηρήσεις:

Ακύρωση Αλλαγών Αποθήκευση

Εικόνα 3.21: Καταχώρηση Νέου Προϊόντος - Βασικά Στοιχεία.

Βασικά Στοιχεία

Ενεργό για ε-shop: Μην:

Χώρα Προέλευσης:

Διαστάσεις

Πλάτος: Βάρος:

Υψος: Βάθος:

Τιμή Προσφοράς: Προσφορά:

Ελάχιστη Ποσότητα: 0

Περιγραφή Αγγλικά:

Περιγραφή 2: assort:

ΕΛ Πεδία

Κεμενο 14: Κεμενο 15: Κεμενο 16:

Κεμενο 18: Κεμενο 17: Κεμενο 19: Κεμενο 20:

Ακύρωση Αλλαγών Αποθήκευση

Εικόνα 3.22: Καταχώρηση Νέου Προϊόντος - Στοιχεία προβολής στο e-shop

3.4.2 Βάση Δεδομένων

Όπως έγινε λόγος και στην προηγούμενη παράγραφο, η Βάση Δεδομένων αποτελεί τον ουσιαστικό δομικό σύνδεσμο της υποδομής του ηλεκτρονικού καταστήματος με το πρόγραμμα μηχανογράφησης. Προγραμματιστικά, επιλέξαμε να προσαρμόσουμε την λειτουργία του back-end του ηλεκτρονικού καταστήματος στις λειτουργικές απαιτήσεις που αφορούν τη ΒΔ του προγράμματος μηχανογράφησης, εφόσον το δεύτερο προϋπήρχε του πρώτου. Με αυτή την σύμβαση υπόψιν, σε αυτή τη φάση της ανάπτυξης του ιστοτόπου, αποφασίστηκε το ηλεκτρονικό κατάστημα να μην καλύπτει όλο τις ανάγκες δεδομένων του προγράμματος μηχανογράφησης, αλλά μόνο το απαραίτητο εύρος αυτών. Με άλλα λόγια, οι καταχωρήσεις στην Βάση Δεδομένων από τη λειτουργία του ηλεκτρονικού καταστήματος θα αφορούν μόνο τις απολύτως απαραίτητες πληροφορίες που αφορούν ένα προϊόν, μία κατηγορία, ή έναν πελάτη. Επίσης το ηλεκτρονικό κατάστημα δεν θα είναι υπεύθυνο για την καταγραφή των πληροφοριών των προμηθευτών της επιχείρησης. Οι υπόλοιπες πληροφορίες για κάθε μία από τις προαναφερθείσες περιπτώσεις θα εισάγονται - εφόσον είναι απαραίτητες για την λειτουργία της επιχείρησης πλέον - από το πρόγραμμα μηχανογράφησης. Στη συνέχεια θα παρουσιαστούν στιγμιότυπα από την Βάση Δεδομένων όπως αυτή οπτικοποιείται μέσω του προγράμματος μηχανογράφησης, για την βέλτιστη παρουσίαση της δομής και της λειτουργικότητάς της.

Στις Εικόνες 3.23 και 3.24 παρουσιάζονται στιγμιότυπα της Βάσης Δεδομένων που αφορούν τα προϊόντα του καταστήματος και το απόθεμά τους, μέσω της προβολής της Βάσης Δεδομένων και της σχετικής προβολής της από το πρόγραμμα μηχανογράφησης, από όπου μπορούν και να γίνουν αλλαγές σε αυτή.

Κεφάλαιο 3

A/A Είδους	A/A Βασικού Είδους (Rel Code)	Κωδικός	BarCode	Κωδικός Είδους Με χαρακτηριστικά	FPA_ID	ΤΙΜΗ_X	ΤΙΜΗ_L	ΥΠΟΛΟΙΠΟ_ΑΡΟΤΗΚΗΣ	PERIGRAΦΗ	FS6	FS10	Τιμή Προσφ...	εν... es...	Ελ. Ποσότητα Παρ.
5986	0	XAA-211			0	4,4355	5,50	5	ΡΑΚΟΡ ΚΩΝΙΚΟ ΘΥΛΙΚ...	16-10-14		0	0	
4781	0	ZAM-032			0	5,2419	6,50	8,20	ΚΑΛΩΔΙΟ ΜΠΟΥΖΙ D5	01-11-13		0	0	
5198	0	TZA-003			0	0,5645	0,70	46,10	ΛΑΣΤΙΧΟ ΔΙΑΦ ΑΛΦΑ...	15-10-14		0	0	
342	0	BEA-130	520000003420		0	2,82	3,50	20	ΣΤΥΛΙΑΡΙ ΤΣΕΚΟΥΡΙ Ν...	24-04-14		0	0	
2871	0	ΜΗΤΡ-034	5200000028713		0	5,64	7	0	ΡΑΟΥΛΑ ΡΟΥΛΕΜΑΝ Σ...	30-08-14		0	0	
3616	0	Π-01			0	22,5806	28	0	ΠΟΛΥΚΑΡΒΟΝΙΚΟ ΔΙΑ...	21-09-12		0	0	
2430	0	ΛΑΜ-003	5200000024302		0	3,87	4,80	9	ΤΡΙΒΙΔΙ ΜΕ ΠΛΑΣΤΙΚΗ...	17-09-14		0	0	
3897	0	ΠΑΝ-033			0	4,4355	5,50	0	ΣΤΕΓΝΩΤΙΚΟ 800 Γρ	26-09-14		0	0	
8932	0	B-1231	8033598001434		0	16,1290	20	4	ΚΛΕΙΔΑΡΙΑ ΜΠΙΣΛΙΑΣ F...					
8871	0	B-1183	5200000088717		0	14,9194	18,50	9	ΚΛΕΙΔΑΡΙΑ ΑΣΦΑΛΕΙΑ...					
7764	0	B-366	3253561654787		0	4,0323	5	3	ΚΑΤΣΑΒΙΔΙ WERA ΣΙΣΙ...					
8481	0	B-872	3253561113000		0	2,0161	2,50	10	LAMES STANLEY ΧΑΡΤ...					
4327	0	ΠΑΠΑΓ-090	4011890071403		0	32,2581	40	1	ΔΙΣΚΟΣ ΔΙΑΜΑΝΤΕ RH...	04-04-14		0	0	
5336	0	Φ-8	5205205119020		0	8,6290	10,70	7	ΧΗΜΙΚΟ ΒΙΣΜΑ ΠΟΛΙΕ...	27-08-14		0	0	
9533	0	ΒΛ-135	5200000095333		0	12,9032	16	3	ΔΙΝΑΤΣΑ ΛΕΥΚΗ 150γ...					
7382	0	B-101	CRES7006		0	11,2903	14	5	ΨΑΛΙΔΙ ΚΛΑΔΟΥ CRES...					
7401	0	B-119	CRES7308		0	10,4839	13	5	ΨΑΛΙΔΙ ΚΛΑΔΟΥ CRES...					
9542	0	ΒΛ-142	5200000095425		0	9,6774	12	0	ΚΑΝΤΑΡΑΚΙ ΗΛΕΚΤΡΟ...					
9536	0	ΒΛ-136	5200000095364		0	8,0645	10	1	ΒΑΡΙΟΠΟΥΛΑ FIBER G...					
9545	0	ΒΛ-145	5200000095456		0	8,8710	11	4	ΜΥΣΤΡΙ ΣΩΦΑΤ.ΛΟΥΤΙ...					
9544	0	ΒΛ-144	5200000095449		0	8,8710	11	3	ΜΥΣΤΡΙ ΣΩΦΑΤ.ΛΟΥΤΙ...					
695	0	ΒΟΓ-008	09100308002		0	5,6452	7	1	ΣΦΗΝΑ 1,5kg	15-02-14		0	0	
9541	0	ΒΛ-141	5200000095418		0	4,8387	6	1	ΚΑΝΤΑΡΑΚΙ ΗΛΕΚΤΡΟ...					
9580	0	ΒΛ-147	5200000095807		0	6,4516	8	1	ΠΟΤΗΡΟΤΡΥΠΑΝΟ 76...					
9114	0	B-1364	5200000091144		0	20,1613	25	3	ΤΡΟΧΟΣ ΚΑΡΤΟΥΣΙΟΥ Σ...					
9539	0	ΒΛ-139	5200000095395		0	2,4194	3	6	ΓΑΝΘΙΑ ΜΑΥΡΑ LATEX...					
9538	0	ΒΛ-138	5200000095388		0	1,2097	1,50	8	ΓΑΝΘΙΑ LATEX CRESM...					
8500	0	B-890	5200000085006		0	1,2097	1,50	11	ΓΑΝΘΙΑ LATEX CRESM...					
8499	0	B-889	5200000084993		0	1,2097	1,50	10	ΓΑΝΘΙΑ LATEX CRESM...					
9543	0	ΒΛ-143	01100205002		0	1,2097	1,50	3	ΚΑΤΣΑΒΙΔΙ ΤΕΤΡΑΓΩΝ...					
9582	0	ΒΛ-149	5200000095821		0	0,8871	1,10	100	ΧΑΛΚΑΣ ΛΟΥΚΕΤΟ ΣΤΡ...					

Εικόνα 3.23: Βάση Δεδομένων - Απόθεμα Προϊόντων.

Επιλ...	Αρ...	Κωδικός	BarCode	Περιγραφή	Μον. Μετρ.	Τιμή Χ.	Τιμή Λ.	Τελ. Τιμή	Υπόλοιπο	Ενεργό	Κατηγορία
<input type="checkbox"/>	1	Γ-113	5035048167830	ΚΡΟΥΣΤΙΚΟ ΔΡΑΠΑΝΟ ΑΥΤΟΡΥΘΜΙΖΟΜΕΝΟ...	ΤΕΜΑΧΙΑ	75,8065	94,0001	54,5000	1	<input checked="" type="checkbox"/>	[Η τιμή επεξεργασίας είναι κενή]
<input type="checkbox"/>	1	Γ-562	6942629208105	ΤΡΥΠΑΝΙΑ ΚΟΒΑΛΤΙΟΥ TACTIX ΣΕΤ 15 ΤΕΜ (...)	ΤΕΜΑΧΙΑ	17,3387	21,5000	12,5400	1	<input checked="" type="checkbox"/>	ΕΡΓΑΛΕΙΑ ΚΗΠΟΥ ΑΡΓΟΥ
<input type="checkbox"/>	1	ΕΑΒ-007	5200000012972	ΦΙΣ ΠΟΜΟΛΟ Ν 713 ΚΑΡΝΙΚ ΜΑΤ-ΝΙΚΕΛ Ρ...	ΤΕΜΑΧΙΑ	35,8871	44,5000	26,6000	2	<input checked="" type="checkbox"/>	ΔΙΑΦΟΡΑ
<input type="checkbox"/>	1	B-1470	5200000092578	ΦΙΣ ΠΟΜΟΛΟ Ν 810 ΚΑΡΝΙΚ ΜΑΤ-ΟΡΟ ΡΟΖ...	ΤΕΜΑΧΙΑ	35,8871	44,5000	26,7000	4	<input checked="" type="checkbox"/>	ΔΙΑΦΟΡΑ
<input type="checkbox"/>	1	ΒΑΖ-016	5200000002072	BODY 20.80 HARTER DELIX 1 LITR	ΔΟΧΕΙΟ ...	10,0000	12,4200	0,0000	0	<input checked="" type="checkbox"/>	[Η τιμή επεξεργασίας είναι κενή]
<input type="checkbox"/>	1	ΒΑΖ-020	5200000002102	BODY 999 1 KG ΑΡΜΟΚΟΛΑ ΠΙΝΕΛΟΥ	ΔΟΧΕΙΟ ...	5,1260	6,3600	0,0000	0	<input checked="" type="checkbox"/>	[Η τιμή επεξεργασίας είναι κενή]
<input type="checkbox"/>	1	ΒΑΖ-013	5200000002041	BODYSOFT 1 ΚΙΛ ΔΙΑ/ΚΟΣ	ΔΟΧΕΙΟ ...	3,6970	4,6000	0,0000	5	<input checked="" type="checkbox"/>	[Η τιμή επεξεργασίας είναι κενή]
<input type="checkbox"/>	1	ΒΑΖ-009		BODYSOFT 2 Kg	ΔΟΧΕΙΟ ...	6,7230	8,3500	0,0000	0	<input checked="" type="checkbox"/>	[Η τιμή επεξεργασίας είναι κενή]
<input type="checkbox"/>	1	ΧΟΥ-012	5200000061154	D-34 HDROSTOP 5 Kg	ΤΕΜΑΧΙΑ	13,7097	17,0000	10,0500	6	<input checked="" type="checkbox"/>	ΔΟΜΗΣΗ - ΞΗΡΑ ΔΟΜΗΣΗ
<input type="checkbox"/>	1	ΧΟΥ-046		D-34 HYDROSTOP 1 ΚΙΛ	ΤΕΜΑΧΙΑ	2,6890	3,3400	0,0000	0	<input checked="" type="checkbox"/>	[Η τιμή επεξεργασίας είναι κενή]
<input type="checkbox"/>	1	ΧΟΥ-029		D-34 HYDROSTOP 20 ΚΙΛ	ΤΕΜΑΧΙΑ	46,7700	58,0000	34,8400	1	<input checked="" type="checkbox"/>	ΔΟΜΗΣΗ - ΞΗΡΑ ΔΟΜΗΣΗ
<input type="checkbox"/>	1	ΧΟΥ-033		D-85 ANT/ΕΞ/ΚΟ-ANT/ΤΡΙΒΙΚΟ 5/1	ΤΕΜΑΧΙΑ	14,0650	17,4700	0,0000	2	<input checked="" type="checkbox"/>	[Η τιμή επεξεργασίας είναι κενή]
<input type="checkbox"/>	1	ΧΟΥ-045		DUROLIT 1 LIT	ΤΕΜΑΧΙΑ	2,8226	3,5000	0,0000	4	<input checked="" type="checkbox"/>	ΔΟΜΗΣΗ - ΞΗΡΑ ΔΟΜΗΣΗ
<input type="checkbox"/>	1	ΧΟΥ-025		DUROLIT 5 LIT	ΤΕΜΑΧΙΑ	11,2903	14,0000	8,3100	3	<input checked="" type="checkbox"/>	ΔΟΜΗΣΗ - ΞΗΡΑ ΔΟΜΗΣΗ
<input type="checkbox"/>	1	ΧΟΥ-041	5204277074756	DUROSTIC WAX ΑΥΤΟ SHAMPO 750ml	ΤΕΜΑΧΙΑ	3,6290	4,5000	2,6800	24	<input checked="" type="checkbox"/>	ΔΙΑΦΟΡΑ
<input type="checkbox"/>	1	ΧΟΥ-042	5204277055304	DYROFLASH ΚΑΘ.ΓΥΑΛ.ΤΑΜΠΛΟ300ml	ΤΕΜΑΧΙΑ	3,0645	3,8000	2,2363	3	<input checked="" type="checkbox"/>	ΔΟΜΗΣΗ - ΞΗΡΑ ΔΟΜΗΣΗ
<input type="checkbox"/>	1	ΒΑΖ-007		EXPRESS RAPID 4:1 4 LITR	ΔΟΧΕΙΟ ...	37,8150	46,9800	0,0000	0	<input checked="" type="checkbox"/>	[Η τιμή επεξεργασίας είναι κενή]
<input type="checkbox"/>	1	K-025		FC 5 * EU	ΤΕΜΑΧΙΑ	201,6100	250,0000	0,0000	1	<input checked="" type="checkbox"/>	ΕΡΓΑΛΕΙΑ ΚΗΠΟΥ ΑΡΓΟΥ
<input type="checkbox"/>	1	NTA-039		FLOETROUR 1 LIT	ΤΕΜΑΧΙΑ	9,6640	12,0100	0,0000	5	<input checked="" type="checkbox"/>	[Η τιμή επεξεργασίας είναι κενή]
<input type="checkbox"/>	1	Π-38	4044996041865	GROSS ΜΑΠΗΝΤΙΚΗ ΠΡΟΕΚΤΑΣΗ ΡΥΘΜΙΖΟ...	ΤΕΜΑΧΙΑ	2,9839	3,7000	1,9000	5	<input checked="" type="checkbox"/>	ΕΡΓΑΛΕΙΑ ΚΗΠΟΥ ΑΡΓΟΥ
<input type="checkbox"/>	1	ΧΗΜ-026		NELF HARDE MATTVERNIS SAT 1/1	ΔΟΧΕΙΟ ...	10,5700	13,1300	0,0000	7	<input checked="" type="checkbox"/>	[Η τιμή επεξεργασίας είναι κενή]
<input type="checkbox"/>	1	ΧΟΥ-056		PERLAFIX ΚΟΡΦΙΔΩΝ ΚΕΡ.ΑΝ.25/1	ΔΟΧΕΙΟ ...	12,1000	15,0000	8,8780	14	<input checked="" type="checkbox"/>	ΔΟΜΗΣΗ - ΞΗΡΑ ΔΟΜΗΣΗ
<input type="checkbox"/>	1	Π-854		RARTITION (ΛΑΙΜΟΣ)	ΤΕΜΑΧΙΑ	3,8710	4,8000	2,8000	0	<input checked="" type="checkbox"/>	ΔΙΑΦΟΡΑ
<input type="checkbox"/>	1	ΚΕ-25		RM110 10 LIT ΧΗΜΙΚΟ	ΤΕΜΑΧΙΑ	70,1613	87,0000	0,0000	0	<input checked="" type="checkbox"/>	ΕΡΓΑΛΕΙΑ ΚΗΠΟΥ ΑΡΓΟΥ
<input type="checkbox"/>	1	02629	5200000081459	TestΠαραγραφή ελληνικά	ΤΕΜΑΧΙΑ	0,0000	0,0000	0,0000	20	<input checked="" type="checkbox"/>	ΑΓΡΟΤΙΚΑ
<input type="checkbox"/>	1	ΟΡΕΛ-236		ΑΓΓΙΣΤΡΟ	ΤΕΜΑΧΙΑ	1,3700	1,7100	0,0000	0	<input checked="" type="checkbox"/>	[Η τιμή επεξεργασίας είναι κενή]
<input type="checkbox"/>	1	ΟΡΕΛ-107		ΑΓΓΙΣΤΡΟ 2100	ΤΕΜΑΧΙΑ	1,3400	1,6700	0,0000	1	<input checked="" type="checkbox"/>	[Η τιμή επεξεργασίας είναι κενή]
<input type="checkbox"/>	1	ΟΡΕΛ-144		ΑΓΓΙΣΤΡΟ 335-51-55	ΤΕΜΑΧΙΑ	2,8226	3,5000	0,0000	5	<input checked="" type="checkbox"/>	ΕΡΓΑΛΕΙΑ ΚΗΠΟΥ ΑΡΓΟΥ
<input type="checkbox"/>	1	Π-463		ΑΓΓΙΣΤΡΟ D TYPE	ΤΕΜΑΧΙΑ	5,0800	6,3000	0,0000	1	<input checked="" type="checkbox"/>	ΕΡΓΑΛΕΙΑ ΚΗΠΟΥ ΑΡΓΟΥ
<input type="checkbox"/>	1	Π-493		ΑΓΓΙΣΤΡΟ ΒΟΛΑΝ 455-140	ΤΕΜΑΧΙΑ	2,3400	2,9000	0,0000	2	<input checked="" type="checkbox"/>	ΕΡΓΑΛΕΙΑ ΚΗΠΟΥ ΑΡΓΟΥ
<input type="checkbox"/>	1	Π-507		ΑΓΓΙΣΤΡΟ ΒΟΛΑΝ 61	ΤΕΜΑΧΙΑ	6,3700	7,9000	0,0000	2	<input checked="" type="checkbox"/>	ΕΡΓΑΛΕΙΑ ΚΗΠΟΥ ΑΡΓΟΥ
<input type="checkbox"/>	1	Π-786		ΑΓΓΙΣΤΡΟ ΕΚΦΙΝΗΣ 153R	ΤΕΜΑΧΙΑ	2,8226	3,5000	1,9300	1	<input checked="" type="checkbox"/>	ΕΡΓΑΛΕΙΑ ΚΗΠΟΥ ΑΡΓΟΥ
<input type="checkbox"/>	1	ΟΡΕΛ-252		ΑΓΓΙΣΤΡΟ ΡΕΦΟΥΛΑΤΟΡΟΥ	[Η τιμή εν...	3,6300	4,5000	2,3100	2	<input checked="" type="checkbox"/>	ΕΡΓΑΛΕΙΑ ΚΗΠΟΥ ΑΡΓΟΥ
<input type="checkbox"/>	1	ΟΡΕΛ-199		ΑΓΓΙΣΤΡΟ ΡΕΦΟΥΛΑΤΟΡΟΥ 394	ΤΕΜΑΧΙΑ	3,2258	4,0000	0,0000	3	<input checked="" type="checkbox"/>	ΕΡΓΑΛΕΙΑ ΚΗΠΟΥ ΑΡΓΟΥ
<input type="checkbox"/>	1	ΟΡΕΛ-081		ΑΓΚΙΣΤΡΟ ΓΙΑ 65	ΤΕΜΑΧΙΑ	4,2300	5,2500	0,0000	5	<input checked="" type="checkbox"/>	[Η τιμή επεξεργασίας είναι κενή]
<input type="checkbox"/>	1	ΟΡΕΛ-045		ΑΓΚΙΣΤΡΟ ΓΙΑ ΕΦΚΟ 8510	ΤΕΜΑΧΙΑ	1,0170	1,2600	0,0000	5	<input checked="" type="checkbox"/>	[Η τιμή επεξεργασίας είναι κενή]
<input type="checkbox"/>	1	ΛΑΜΠ-014		ΑΓΚΙΣΤΡΟ ΔΙΠΛΟ ΜΠΑΝΟΥ Ν2188 ΝΙΚΕΛ	ΤΕΜΑΧΙΑ	3,6300	4,5000	0,0000	5	<input checked="" type="checkbox"/>	ΜΟΝΩΣΗ - ΣΤΕΓΑΝΩΣΗ
<input type="checkbox"/>	1	ΧΟΥ-109		ΑΔΙΑΒΡ/ΤΙΚΟ D-29 SPOT GREE 1L	ΔΟΧΕΙΟ ...	12,9032	16,0000	9,7200	5	<input checked="" type="checkbox"/>	[Η τιμή επεξεργασίας είναι κενή]

Εικόνα 3.24: Πρόγραμμα Μηχανογράφησης - Διαχείριση και Προβολή Αποθέματος Προϊόντων.

3 Σχεδιασμός και Υλοποίηση Ιστοσελίδας - Back-End

Κωδικός	Επωνυμία	Επάγγελμα	ΑΔΜ	Ημ/νια Τελ. ...	Πελάτης	Όριο Τιμ...	Τρέχων Υπ...	Κύρια Διανοη	Τηλέφωνο	Πόλη	ΤΚ
30-04-30	ΤΣΙΑΡΑΣ ΔΗ...	ΑΓΡΟΤΗΣ-ΣΑ	049326428	09/12/2021	✓	157,50 €	501,65 €	ΑΓΙΑ	2494023663	ΑΓΙΑ	
30-04-31	ΚΑΤΣΙΟΝΗΣ...	Ν.Π.Ι.Δ	91327746		✓	0,00 €	0,00 €	7ο ΧΙΛ ΛΑΡΙΣΗΣ ΣΥΚΟΥΡΙ...		ΛΑΡΙΣΙΑ	
30-04-32	ΚΑΡΑΝΗΣ Θ...	ΑΓΡΟΤΗΣ-ΣΑ	039470302	05/07/2021	✓	0,00 €	35,70 €	ΜΕΓΑΛΟΒΥΣΣΟ	22643	ΑΓΙΑ	40003
30-04-33	ΓΟΥΡΓΙΩΤΗ ...	ΑΓΡΟΤΗΣ-ΣΑ	111270776	05/09/2020	✓	0,00 €	0,00 €	ΜΑΚΕΔΟΝΙΑΣ 3		ΑΓΙΑ	
30-04-34	ΚΟΥΥΝΤΑΡΑ...			06/05/2021	✓	0,00 €	0,00 €	ΜΕΛΗΒΟΙΑ	6945404079	ΜΕΛΗΒΟΙΑ	
30-04-35	ΧΡΙΣΤΟΠΟΥΛ...	ΕΝΟΙΚΙΑΖΟΜΕΝΑ ΔΩΜΑΤ.	103122257		✓	0,00 €	0,00 €	ΑΓΙΟΚΑΜΠΟΣ		ΑΓΙΑ	
30-04-36	ΤΣΙΟΥΒΑΡΑΣ...	ΑΓΡΟΤΗΣ-ΣΑ	059529368	04/12/2021	✓	0,00 €	0,00 €	ΑΕΤΟΛΟΦΟΣ	6977741154	ΑΓΙΑ	
30-04-37	ΚΑΡΑΝΤΣΟΣ ...	Θ.Ν.Π.ΜΗ Κ.ΧΑΡΑΚ.			✓	190,64 €	190,64 €	ΚΑΛΑΜΑΚΙ	6937987918	ΑΓΙΑ	
30-04-38	ΚΑΤΣΙΜΠΑΛ...	ΑΓΡΟΤΗΣ-ΣΑ	039469849	14/09/2018	✓	0,00 €	0,00 €	ΑΓΙΑ	69384026098ΑΓ	ΑΓΙΑ	
30-01-94	ΓΚΕΣΟΣ ΘΕΟ...	ΤΑΒΕΡΝΑ	045093754		✓	0,00 €	0,00 €	ΑΓΙΑ	2494024666	ΑΓΙΑ	
30-01-95	ΑΒΡΑΔΕ ΕΛΕ...	ΦΡΥΓΙΟ ΣΕΝΩΝ ΓΛΩΣΣΩΝ	111272849		✓	0,00 €	0,00 €	ΑΓΙΟΚΑΜΠΟΣ		ΑΓΙΑ	
30-01-96	ΣΟΛΩΜΟΣ Ν...	ΑΓΡΟΤΗΣ-ΣΑ			✓	257,78 €	257,78 €	.NABPA	6974891217	ΑΓΙΑ	...
30-01-97	ΘΕΟΔΩΡΟΥΑ...	ΑΓΡΟΤΗΣ-ΣΑ	071727136	27/11/2021	✓	0,00 €	0,00 €	ΜΕΓΑΛΟΒΥΣΣΟ	6942865360	ΑΓΙΑ	
30-01-98	ΠΑΠΑΚΩΣΤΑ...	ΑΓΡΟΤΗΣ-ΣΑ	050719123	16/12/2019	✓	0,00 €	0,00 €	ΑΓΙΑ		ΑΓΙΑ	
30-01-99	ΤΖΑΒΕΛΑΣ Ν...	ΓΕΩΡΓΙΚΗΝΟΤΡΟΦΟΣ	037505869	29/12/2020	✓	0,00 €	0,00 €	ΠΟΤΑΜΙΑ	6977746271	ΑΓΙΑ	
30-1555...	ΤΟΥΜΠΑ ΕΛΕ...	ΤΑΒΕΡΝΑ	068027250	16/04/2018	✓	0,00 €	0,00 €	Κ. ΝΕΡΟ		ΑΓΙΑ ΛΑΡΙΣΙΑΣ	
30-02-01	ΜΑΝΤΖΑΝΑΣ...	ΑΓΡΟΤΗΣ-ΣΑ	111268338	23/12/2021	✓	0,00 €	3,70 €	ΝΕΡΟΥΜΥΛΟΙ	6946046803	ΑΓΙΑ	
30-00-24	ΓΩΓΟΣ ΠΕΤΡ...	ΕΜΠΟΡΙΟ ΠΟΔΗΛΑΤΩΝ	078336214		✓	0,00 €	0,00 €	ΓΕΩΡΓΙΑΔΟΥ 67		ΛΑΡΙΣΙΑ	
30-00-25	ΤΣΙΝΤΣΙΡΑΚ...	ΑΓΡΟΤΗΣ-ΣΑ	071851922	16/10/2021	✓	0,00 €	0,00 €	ΜΕΛΙΒΟΙΑ	6973034032	ΑΓΙΑ	
30-00-26	ΕΥΘΥΜΙΟΥ Σ...	ΑΡΤΟΠΟΣΟ			✓	0,00 €	0,00 €	Κ. ΝΕΡΟ		ΛΑΡΙΣΙΑ	
30-00-27	ΔΗΜΗΤΡΟΥΑ...	ΙΔΙΩΤΗΣ	008046334	14/05/2021	✓	0,00 €	0,00 €	ΑΓΙΑ	6945649453ΤΑΚ	ΑΓΙΑ	
30-00-28	ΖΕΚΟΣ ΚΩΝ/...				✓	12,40 €	0,00 €	ΑΝΑΒΡΑ		ΑΓΙΑ	
30-00-52	ΑΣΠΡΙΤΣ ΑΛΒ...	ΨΥΓΓΕΙΑ-ΣΥΣΚΕΥΣΤΗΡΙΑ		05/01/2022	✓	0,00 €	20,00 €	ΑΜΥΓΔΑΛΗ	6944618523	ΑΓΙΑ	
30-00-53	ΗΡΑΚΛΗΣ ΚΩ...			23/09/2021	✓	0,00 €	0,00 €	ΜΕΤΑΧΩΡΙ	6979771942	ΑΓΙΑ	
30-00-54	ΜΠΑΡΟΥΝΙΑ...	ΑΓΡΟΤΗΣ-ΣΑ	074024621	22/12/2021	✓	0,00 €	0,00 €	ΝΕΡΟΥΜΥΛΟΙ		ΑΓΙΑ	
30-01-52	ΚΑΡΑΚΩΣΤΟ...	Θ.Ν.Π.ΜΗ Κ.ΧΑΡΑΚ.		05/11/2021	✓	0,00 €	0,00 €	ΑΓΙΑ		ΑΓΙΑ	
30-1019	ΚΑΛΟΠΗΡΟΣ ...	ΑΓΡΟΤΗΣ-ΣΑ	111273871	22/12/2017	✓	0,00 €	0,00 €	ΑΓΙΑ		ΑΓΙΑ	40003
30-1042	ΜΑΝΤΖΑΝΑ ...	ΑΓΡΟΤΗΣ-ΣΑ	111257533	11/03/2021	✓	0,00 €	0,00 €	ΜΕΤΑΞΟΧΩΡΙ		ΑΓΙΑ	
30-1077	ΠΑΠΑΛΕΞΑ ...	ΑΓΡΟΤΗΣ-ΣΑ	052933100	21/09/2020	✓	0,00 €	0,00 €	ΑΡΓΥΡΟΠΟΥΛΙ	6937492274	ΛΑΡΙΣΙΑ	
30-1094	ΑΡΧΟΝΤΗΣ Ε...	ΛΟΓΙΣΤΗΣ	044020531	17/07/2021	✓	0,00 €	0,00 €	25ης ΜΑΡΤΙΟΥ 18	2494023671	ΑΓΙΑ	
30-1121	ΓΟΥΡΓΙΩΤΗΣ...	ΔΑΣΟΕΡΓΑΤΗΣ	111298773	26/06/2015	✓	0,00 €	0,00 €	ΣΚΛΗΘΡΟ	6934727789	ΑΓΙΑ	
30-1143	ΠΑΠΑΤΡΙΑΝΤ...	ΙΔΙΩΤΗΣ		06/08/2015	✓	0,00 €	0,00 €	ΑΜΥΓΔΑΛΗ	6972867392	ΑΓΙΑ	
30-1151	ΤΣΕΚΜΕΤΣΗΣ...	ΙΔΙΩΤΗΣ		18/08/2015	✓	0,00 €	0,00 €	ΚΑΡΥΤΣΑ-ΑΓΙΑ	6938055725	ΑΓΙΑ	
30-1075	ΚΑΝΑΚΗΣ Β. ...	ΠΟΛ.ΜΗΧΑΝΙΚΟΣ	076736540	22/04/2015	✓	0,00 €	0,00 €	ΣΠΑΘΑΡΗ 18	6972868853	ΛΑΡΙΣΙΑ	
30-1100	ΚΟΜΝΗΤΗΣ Ε...	ΙΔΙΩΤΗΣ		18/05/2015	✓	0,00 €	0,00 €	ΠΡΙΝΙΑ		ΑΓΙΑ	
30-1022	ΜΠΕΛΙΜΕΖΗ...	ΤΖΑΚΙΑ	074249281	23/11/2020	✓	0,00 €	0,00 €	ΘΕΩΔ/ΛΟΥ 90		ΛΑΡΙΣΙΑ	
30-1040	ΚΟΥΥΝΤΑΡΑ...	ΛΟΓΙΣΤΗΣ	044020936	07/09/2021	✓	0,00 €	0,00 €	ΜΕΛΙΒΟΙΑ		ΑΓΙΑ	
30-1115	ΑΝΤΩΝΟΥΑΗ...	ΕΝΟΙΚΙΑΖΟΜΕΝΑ ΔΩΜΑΤ.	074342520	12/06/2015	✓	0,00 €	0,00 €	ΠΑΛΙΟΥΡΓΙΑ	2410530228	ΑΓΙΑ	

Εικόνα 3.25: Πρόγραμμα Μηχανογράφησης - Πελατολόγιο.

Στη Εικόνα 3.25 προβάλλεται μέσω του προγράμματος μηχανογράφησης ένα στιγμιότυπο της Βάσης Δεδομένων στο κομμάτι που αφορά το πελατολόγιο της επιχείρησης. Μέσω του περιβάλλοντος αυτού δύναται να γίνουν και χειροκίνητες επιβεβλημένες αλλαγές στα στοιχεία τους.

Στην Εικόνα 3.26 παρουσιάζεται ένα στιγμιότυπο από τη Βάση Δεδομένων όπου περιέχονται οι παραγγελίες των πελατών του καταστήματος, είτε ηλεκτρονικά, είτε μέσω του φυσικού καταστήματος. Αντίστοιχα, στην Εικόνα 3.27 παρουσιάζονται αντίστοιχα τα παραστατικά των συναλλαγών με τους πελάτες για τις αντίστοιχες πωλήσεις.

Κεφάλαιο 3

Ελ. Ποσότητα Παρ.	ενεργή Προσφορά	Παράτ... Ελλην...	ΚΑΤΗΓΟΡΙΑ_ID	Αγγλική Περιγραφή	Παρατηρήσεις Αγγλικά	FS13	EIDOS_VATHOS	EIDOS_W	EIDOS_H	EIDOS_BAROS	Κωδικός Προμηθευτή	ΚΑΤΗΓΟΡΙΑ	ΥΠΟΚΑΤΗΓΟΡΙΑ	ΥποΥποκατηγορία
0	0	ΧΑΛΟΥΓΗΣ 3...					3,0110	0	0	0	3101207	ΜΟΝΩΣΗ - Σ...		
0	0	ΣΑΜΑΡΑΣ 2,...					19,53	0	0	0		ΕΡΓΑΛΕΙΑ Κ...		
0	0	ΠΡΙΝ 2015 0...					34,6880	0	0	0				
0	0	ΠΑΤΣΙΟΥΡΑΣ...	187				0	0	0	0		ΕΡΓΑΛΕΙΑ Κ...		Στυλάρι Τασκουριού
0	0	ΠΑΠΑΠΑΝΝ...					48,8310	0	0	0,30		ΕΡΓΑΛΕΙΑ Κ...		
0	0	ΟΠΑΛ 19,96...					51,8950	0	0	0				
0	0	Ε 25-5-17ΠΑ...					63,4040	0	0	0,05	322	ΕΡΓΑΛΕΙΑ Κ...		
0	0	ΒΙΤΕΣ 2,98€...					26,7840	0	0	0	101001	ΔΟΜΗΣΗ - Ε...		
0	0	9-9-21 ΣΚΟΡ...								0,30	03764-02-010	ΔΙΑΦΟΡΑ		
0	0	9-9-21 ΣΚΟΡ...								0,40	92125/91125	ΔΙΑΦΟΡΑ		
0	0	9-9-21 ΠΑΠΑ...								0,20	99997	ΕΡΓΑΛΕΙΑ ΧΕ...	Καταβόδια-Δυνα...	
0	0	9-9-21 ΠΑΠΑ...								0,10	1-11-300	ΕΡΓΑΛΕΙΑ ΧΕ...	Διάτρηση-Κοπή	
0	0	9-9-21 ΠΑΠΑ...					0	0	0	0	A-T1371	ΕΡΓΑΛΕΙΑ ΧΕ...	Διάτρηση-Κοπή	
0	0	9-9-21 ΠΑΠΑ...					22,42	0	0	0	A-A204	ΔΟΜΗΣΗ - Ε...	Σιλκάνες-Αφροί ...	
0	0	9-9-21 ΒΛΑΧ...								1	08526	ΑΓΡΟΤΙΚΑ		
0	0	9-9-21 ΒΛΑΧ...								0	04990			
0	0	9-9-21 ΒΛΑΧ...								0	07593			
0	0	9-9-21 ΒΛΑΧ...								04009				
0	0	9-9-21 ΒΛΑΧ...								07874		ΕΡΓΑΛΕΙΑ ΧΕ...	Σφυριά-Ματσόλε...	
0	0	9-9-21 ΒΛΑΧ...								08416				
0	0	9-9-21 ΒΛΑΧ...								08415				
0	0	9-9-21 ΒΛΑΧ...					8,5630	0	0	0	05540			
0	0	9-9-21 ΒΛΑΧ...								04008				
0	0	9-9-21 ΒΛΑΧ...								01547				
0	0	9-9-21 ΒΛΑΧ...								1,60	07782	ΕΡΓΑΛΕΙΑ Κ...		
0	0	9-9-21 ΒΛΑΧ...								07758				
0	0	9-9-21 ΒΛΑΧ...								07672				
0	0	9-9-21 ΒΛΑΧ...								0,10	07671			
0	0	9-9-21 ΒΛΑΧ...								0,10	07670			
0	0	9-9-21 ΒΛΑΧ...								01839				
0	0	9-9-21 ΒΛΑΧ...								02570		ΔΙΑΦΟΡΑ		

Εικόνα 3.26: Βάση Δεδομένων - Παραγγελίες Πελατών.

Α/Α	Α/Α	Ημ/νια	Παραστατικό	Αριθμηση	Σερά	ΑΦΜ	Επωνυμία	Εξόφληση	Κατάσταση	Αρ. Ακυρωτικού	Εκ
1	17,40 €	7/1/2022	Απόδειξη Λισακή Πώλησης		Χαρόγρα...	111289948	ΤΑΚΑΣ ΣΠΥΡΙΔΩΝ	ΠΙΣΤΩΣΗ 20 η...	Κερύραφ...		
2	0,00 €	7/1/2022	Ταμειακή		Χαρόγρα...			Μετρητοίς	Κερύραφ...		
3	12,60 €	7/1/2022	Απόδειξη Λισακή Πώλησης		Χαρόγρα...		ΣΑΛΟΝΙΚΑΣ ΜΑΡΙΟΥ ΧΡΗΣΤΟΣ	ΠΙΣΤΩΣΗ 20 η...	Κερύραφ...		
4	169,00 €	7/1/2022	Τιμολόγιο πώλησης - Δελτίο απ...	21	Μηχανογ...	011407524	ΑΔΑΜ ΓΕΩΡΓΙΟΣ	Μετρητοίς	Ολοκληρω...		
5	0,00 €	7/1/2022	Ταμειακή		Χαρόγρα...			Μετρητοίς	Κερύραφ...		
6	7,91 €	7/1/2022	Τιμολόγιο πώλησης - Δελτίο απ...	20	Μηχανογ...	094525129	ΣΤΕΡΓΙΑΚΟΥΛΗΣ ΑΕ	ΠΙΣΤΩΣΗ 20 η...	Ολοκληρω...		
7	0,00 €	7/1/2022	Ταμειακή		Χαρόγρα...			Μετρητοίς	Κερύραφ...		
8	0,00 €	7/1/2022	Ταμειακή		Χαρόγρα...			Μετρητοίς	Κερύραφ...		
9	5,27 €	7/1/2022	Απόδειξη Λισακή Πώλησης		Χαρόγρα...	038480499	ΟΙΚΟΝΟΜΟΥ ΣΤΑΜ.ΝΙΚΟΛΑΟΣ	ΠΙΣΤΩΣΗ 20 η...	Κερύραφ...		
10	40,94 €	7/1/2022	Τιμολόγιο πώλησης - Δελτίο απ...	19	Μηχανογ...	038480499	ΟΙΚΟΝΟΜΟΥ ΣΤΑΜ.ΝΙΚΟΛΑΟΣ	Μετρητοίς	Ολοκληρω...		
11	0,00 €	7/1/2022	Ταμειακή		Χαρόγρα...			Μετρητοίς	Κερύραφ...		
12	0,00 €	7/1/2022	Ταμειακή		Χαρόγρα...			Μετρητοίς	Κερύραφ...		
13	0,00 €	7/1/2022	Ταμειακή		Χαρόγρα...			Μετρητοίς	Κερύραφ...		
14	0,00 €	7/1/2022	Ταμειακή		Χαρόγρα...			Μετρητοίς	Κερύραφ...		
15	0,00 €	7/1/2022	Ταμειακή		Χαρόγρα...			Μετρητοίς	Κερύραφ...		
16	0,00 €	7/1/2022	Ταμειακή		Χαρόγρα...			Μετρητοίς	Κερύραφ...		
17	0,00 €	7/1/2022	Ταμειακή		Χαρόγρα...			Μετρητοίς	Κερύραφ...		
18	0,00 €	7/1/2022	Ταμειακή		Χαρόγρα...			Μετρητοίς	Κερύραφ...		
19	0,00 €	7/1/2022	Ταμειακή		Χαρόγρα...			Μετρητοίς	Κερύραφ...		
20	0,00 €	7/1/2022	Ταμειακή		Χαρόγρα...			Μετρητοίς	Κερύραφ...		
21	8,40 €	7/1/2022	Απόδειξη Λισακή Πώλησης		Χαρόγρα...		ΜΟΣΚΟΣ ΒΑΣΙΛΕΙΟΣ (ΤΙΜΟΛΕΩΝ)	ΠΙΣΤΩΣΗ 20 η...	Κερύραφ...		
22	0,00 €	7/1/2022	Ταμειακή		Χαρόγρα...			Μετρητοίς	Κερύραφ...		
23	1,00 €	7/1/2022	Απόδειξη Λισακή Πώλησης		Χαρόγρα...	044020997	ΚΑΣΙΔΑΣ ΛΕΑΝ. ΑΘΑΝΑΣΙΟΣ	ΠΙΣΤΩΣΗ 20 η...	Κερύραφ...		
24	10,66 €	7/1/2022	Τιμολόγιο πώλησης - Δελτίο απ...	18	Μηχανογ...	082799070	ΜΑΚΡΟΔΗΜΟΣ Θ -ΧΑΣΙΩΤΗΣ Δ Ο.Ε.	ΠΙΣΤΩΣΗ 20 η...	Ολοκληρω...		
25	0,00 €	7/1/2022	Ταμειακή		Χαρόγρα...			Μετρητοίς	Κερύραφ...		
26	0,00 €	7/1/2022	Ταμειακή		Χαρόγρα...			Μετρητοίς	Κερύραφ...		
27	0,00 €	7/1/2022	Ταμειακή		Χαρόγρα...			Μετρητοίς	Κερύραφ...		
28	330,00 €	7/1/2022	Τιμολόγιο πώλησης - Δελτίο απ...	17	Μηχανογ...	136550023	ΨΥΡΡΑ ΑΙΚΑΤΕΡΙΝΗ	Μετρητοίς	Ολοκληρω...		
29	38,30 €	7/1/2022	Τιμολόγιο πώλησης - Δελτίο απ...	16	Μηχανογ...	996775040	ΔΡΟΣΟΣ Ε. ΚΑΤΑΣ/ΚΗ Μ.Ι.Κ.Ε., ΑΣΤΕΡ...	ΠΙΣΤΩΣΗ 20 η...	Ολοκληρω...		
30	0,00 €	7/1/2022	Ταμειακή		Χαρόγρα...			Μετρητοίς	Κερύραφ...		
31	19,20 €	7/1/2022	Απόδειξη Λισακή Πώλησης		Χαρόγρα...	998019451	ΔΑΣΑΡΧΕΙΟΝ ΑΠΑΣ	ΠΙΣΤΩΣΗ 20 η...	Κερύραφ...		
32	20,42 €	7/1/2022	Τιμολόγιο πώλησης - Δελτίο απ...	15	Μηχανογ...	081160240	ΠΕΤΡΟΥ ΝΑΤΣ Α.Ε.	ΠΙΣΤΩΣΗ 20 η...	Ολοκληρω...		

Εικόνα 3.27: Βάση Δεδομένων - Αγορές Πελατών.

3 Σχεδιασμός και Υλοποίηση Ιστοσελίδας - Back-End

Τέλος, στην Εικόνα 3.28 παρουσιάζεται ένα στιγμιότυπο από το πρόγραμμα μηχανογράφησης όπου προβάλλονται τα τιμολόγια των αγορών προϊόντων από τους προμηθευτές της επιχείρησης. Αντίστοιχα, στην Εικόνα 3.29 προβάλλεται ένα στιγμιότυπο το τμήμα της Βάσης Δεδομένων όπου είναι καταχωρημένα τα στοιχεία των προμηθευτών της επιχείρησης. Σε αυτό το σημείο της λειτουργικότητας υπενθυμίζεται πως το back-end του ηλεκτρονικού καταστήματος δεν έχει πρόσβαση όλες οι διαδικασίες δημιουργίας καταχώρησης, επεξεργασίας και προβολής γίνονται μέσω του προγράμματος μηχανογράφησης.

Αριθμηση Παρ/κου	Παραστατικό	Ημ/νια Παρ/κου	Επωνυμία	Χρέωση	Πίστωση	Αξία Παρ/κου	Παρατηρήσεις	Αρχείο
24	Δελτίο Αποστολής σε Προμηθευτή	29/11/2021	ΧΥΤΑΣ ΦΙΛΗΠΠΑΣ ELECTROCOUP HELLAS	0,00 €	0,00 €	0,00 €	ΣΕΡΒΙΣ ΕΓΓ...	
3091	Ποτωτικό Τιμολόγιο Έκπτωσης	29/11/2021	ΦΑΣΟΠΛΑΣΤ ΓΡΗΓ ΧΡ ΦΑΣΟΗΣ ΑΒΕΕ	0,00 €	-38,64 €	38,64 €		
25	Δελτίο Αποστολής σε Προμηθευτή	29/11/2021	ΠΑΠΑΔΟΠΟΥΛΟΣ ΠΑΝΤΕΛΗΣ ΑΒΕΕ	0,00 €	0,00 €	0,00 €	ΔΙΑ ΣΥΝΕΡΓ...	
19150	Δελτίο Αποστολής Τιμολόγιο	30/11/2021	ΜΠΟΥΚΛΑΣ Ι ΒΙΟΜ/ΚΑ ΕΡΓΛΕΙΑ ΜΗΧ/ΤΑ ΑΕ	0,00 €	2,36 €	-2,36 €		
7409	Δελτίο Αποστολής Τιμολόγιο	30/11/2021	ΒΛΑΧΟΣ ΠΑΝ. ΔΗΜΗΤΡΙΟΣ	0,00 €	272,42 €	-272,42 €		
7393	Δελτίο Αποστολής Τιμολόγιο	30/11/2021	ΒΛΑΧΟΣ ΠΑΝ. ΔΗΜΗΤΡΙΟΣ	0,00 €	544,82 €	-544,82 €		
19135	Δελτίο Αποστολής Τιμολόγιο	30/11/2021	ΜΠΟΥΚΛΑΣ Ι ΒΙΟΜ/ΚΑ ΕΡΓΛΕΙΑ ΜΗΧ/ΤΑ ΑΕ	0,00 €	307,07 €	-307,07 €		
1370	Δελτίο Αποστολής Τιμολόγιο	1/12/2021	ΜΠΕΜΠΟΣ ΑΕ	0,00 €	66,48 €	-66,48 €		
69155	Δελτίο Αποστολής Τιμολόγιο	2/12/2021	ΠΑΠΑΓΙΑΝΝΗ ΑΦΟΙ ΑΕ	0,00 €	17,85 €	-17,85 €		
69184	Δελτίο Αποστολής Τιμολόγιο	2/12/2021	ΠΑΠΑΓΙΑΝΝΗ ΑΦΟΙ ΑΕ	0,00 €	283,61 €	-283,61 €		
7446	Δελτίο Αποστολής Τιμολόγιο	2/12/2021	ΒΛΑΧΟΣ ΠΑΝ. ΔΗΜΗΤΡΙΟΣ	0,00 €	176,12 €	-176,12 €		
27269	Δελτίο Αποστολής Τιμολόγιο	2/12/2021	ΧΟΥΛΗΣ ΝΙΚ ΝΤΟΥΣΤΙΚ ΑΒΕ	0,00 €	91,39 €	-91,39 €		
35927	Δελτίο Αποστολής Τιμολόγιο	2/12/2021	ΠΑΠΑΔΟΠΟΥΛΟΣ ΠΑΝΤΕΛΗΣ ΑΒΕΕ	0,00 €	108,20 €	-108,20 €		
36101	Δελτίο Αποστολής Τιμολόγιο	3/12/2021	ΠΑΠΑΔΟΠΟΥΛΟΣ ΠΑΝΤΕΛΗΣ ΑΒΕΕ	0,00 €	268,67 €	-268,67 €		
155.696	Δελτίο Αποστολής Τιμολόγιο	3/12/2021	ΜΕΝΤΟΡ ΕΛΛΑΣ ΙΚΕ	0,00 €	284,57 €	-284,57 €		
48960	Δελτίο Αποστολής Τιμολόγιο	3/12/2021	ΠΑΠΑΔΗΜΗΤΡΙΟΥ ΘΩΜΑΣ	0,00 €	131,29 €	-131,29 €		
144355	Δελτίο Αποστολής Τιμολόγιο	6/12/2021	ΜΠΑΖΑΚΑΣ Β ΓΕΩΡΓΙΟΣ ΜΟΝΟΠΡΟΣΩΠΗ ΕΠΕ	0,00 €	431,89 €	-431,89 €		
9071	Δελτίο Αποστολής Τιμολόγιο	6/12/2021	ΚΑΝΑΚΗΣ ΘΕΜΙΣΤΟΚΛΗΣ	0,00 €	38,19 €	-38,19 €		
6774	Δελτίο Αποστολής Τιμολόγιο	6/12/2021	ΚΟΥΤΑΗΣ Δ. Γ. Α.Β.Ε.Ε.	0,00 €	148,81 €	-148,81 €		
90346	Δελτίο Αποστολής Τιμολόγιο	7/12/2021	ΣΠΥΡΑΤΟΣ ΓΕΩΔ.ΣΠΥΡΙΔΩΝ & ΣΙΑ ΕΕ	0,00 €	250,11 €	-250,11 €		
3213	Δελτίο Αποστολής Τιμολόγιο	7/12/2021	ΚΕΡΑΜΙΔΑΣ Δ&Γ ΟΕ	0,00 €	77,88 €	-77,88 €		
5314	Ποτωτικό Τιμολόγιο Έκπτωσης	7/12/2021	ΠΑΠΑΓΙΑΝΝΗ ΑΦΟΙ ΑΕ	0,00 €	-9,04 €	9,04 €		
1484	Δελτίο Αποστολής Τιμολόγιο	7/12/2021	ΧΥΤΑΣ ΦΙΛΗΠΠΑΣ ELECTROCOUP HELLAS	0,00 €	100,32 €	-100,32 €		
36410	Δελτίο Αποστολής Τιμολόγιο	8/12/2021	ΠΑΠΑΔΟΠΟΥΛΟΣ ΠΑΝΤΕΛΗΣ ΑΒΕΕ	0,00 €	131,34 €	-131,34 €		
3183	Ποτωτικό Τιμολόγιο Έκπτωσης	8/12/2021	ΦΑΣΟΠΛΑΣΤ ΓΡΗΓ ΧΡ ΦΑΣΟΗΣ ΑΒΕΕ	0,00 €	-14,55 €	14,55 €		
432	Δελτίο Αποστολής Τιμολόγιο	8/12/2021	ΠΑΠΑΔΟΠΟΥΛΟΣ ΠΑΝΤΕΛΗΣ ΑΒΕΕ	0,00 €	18,68 €	-18,68 €		
431	Δελτίο Αποστολής Τιμολόγιο	8/12/2021	ΠΑΠΑΔΟΠΟΥΛΟΣ ΠΑΝΤΕΛΗΣ ΑΒΕΕ	0,00 €	121,99 €	-121,99 €		
13 0045113	Ποτωτικό Τιμολόγιο Έκπτωσης	9/12/2021	ΚΕΡΑΡ ΑΕΕ	0,00 €	-53,80 €	53,80 €		
36481	Δελτίο Αποστολής Τιμολόγιο	9/12/2021	ΠΑΠΑΔΟΠΟΥΛΟΣ ΠΑΝΤΕΛΗΣ ΑΒΕΕ	0,00 €	41,90 €	-41,90 €		
920	Δελτίο Αποστολής Τιμολόγιο	9/12/2021	ΜΕΡΜΥΓΚΑΣ Α ΑΡΓΥΡΗΣ	0,00 €	92,18 €	-92,18 €		
27940	Δελτίο Αποστολής Τιμολόγιο	9/12/2021	ΧΡΥΣΑΦΟΠΟΥΛΟΣ Π & ΣΙΑ ΟΕ	0,00 €	410,24 €	-410,24 €		
36632	Δελτίο Αποστολής Τιμολόγιο	10/12/2021	ΠΑΠΑΔΟΠΟΥΛΟΣ ΠΑΝΤΕΛΗΣ ΑΒΕΕ	0,00 €	40,41 €	-40,41 €		
9973	Δελτίο Αποστολής Τιμολόγιο	13/12/2021	ΣΥΡΜΑΤΕΧΝΙΚΗ ΒΟΛΟΥ ΜΟΝ/ΓΗ ΙΚΕ	0,00 €	27,32 €	-27,32 €		
6607	Δελτίο Αποστολής Τιμολόγιο	13/12/2021	ΓΑΛΛΑΝΗΣ ΦΩΤΗΣ Κ ΣΙΑ ΟΕ	0,00 €	77,38 €	-77,38 €		

Εικόνα 3.28: Πρόγραμμα Μηχανογράφησης - Τιμολόγια αγορών προϊόντων.

Κεφάλαιο 3

Κωδικός Προμηθευτή	ΚΑΤΗΓΟΡΙΑ	ΥΡΟΚΑΤΗΓΟΡΙΑ	ΥποΥποκατηγορία	Χαρ1 Περ.	Χαρ2 Περ	Origin
3101207	ΜΟΝΩΣΗ - Σ...					2--
	ΕΡΓΑΛΕΙΑ Κ...					4--
						--
	ΕΡΓΑΛΕΙΑ Κ...		Στυλιάρι Τσεκουριού			4-1067-187
	ΕΡΓΑΛΕΙΑ Κ...					4--
						--
322	ΕΡΓΑΛΕΙΑ Κ...					4--
101001	ΔΟΜΗΣΗ - Ξ...					1--
03764-02-010	ΔΙΑΦΟΡΑ					1648--
92125//91125	ΔΙΑΦΟΡΑ					1648--
99997	ΕΡΓΑΛΕΙΑ ΧΕ...	Κατσαβίδια-Δυνα...				1646-970-
1-11-300	ΕΡΓΑΛΕΙΑ ΧΕ...	Διάτρηση-Κοπή				1646-968-
A-T1371	ΕΡΓΑΛΕΙΑ ΧΕ...	Διάτρηση-Κοπή				1646-968-
A-A204	ΔΟΜΗΣΗ - Ξ...	Σιλικόνες-Αφροί ...				1-946-
08526	ΑΓΡΟΤΙΚΑ					3--
04990						--
07593						--
04009						--
07874	ΕΡΓΑΛΕΙΑ ΧΕ...	Σφυρία-Ματσόλε...				1646-980-
08416						--
08415						--
05540						--
04008						--
01547						--
07782	ΕΡΓΑΛΕΙΑ Κ...					4--
07758						--
07672						--
07671						--
07670						--
01839						--
02570	ΔΙΑΦΟΡΑ					1648--

Εικόνα 3.29: Βάση Δεδομένων - Προμηθευτές.

3.5 Επίλογος

Με το πέρας αυτού του Κεφαλαίου, ο αναγνώστης έχει την πληρέστερη δυνατή εικόνα για την αλγοριθμική λογική, την τεχνική υποδομή, και την δόμηση του συστήματος που αφορά το ολιστικό ηλεκτρονικό κατάστημα συνδεδεμένο με ένα ήδη υφιστάμενο σύστημα μηχανογράφησης.

Κεφάλαιο 4ο: Σχεδιασμός και Υλοποίηση Ιστοσελίδας - Front-End

4.1 Εισαγωγή

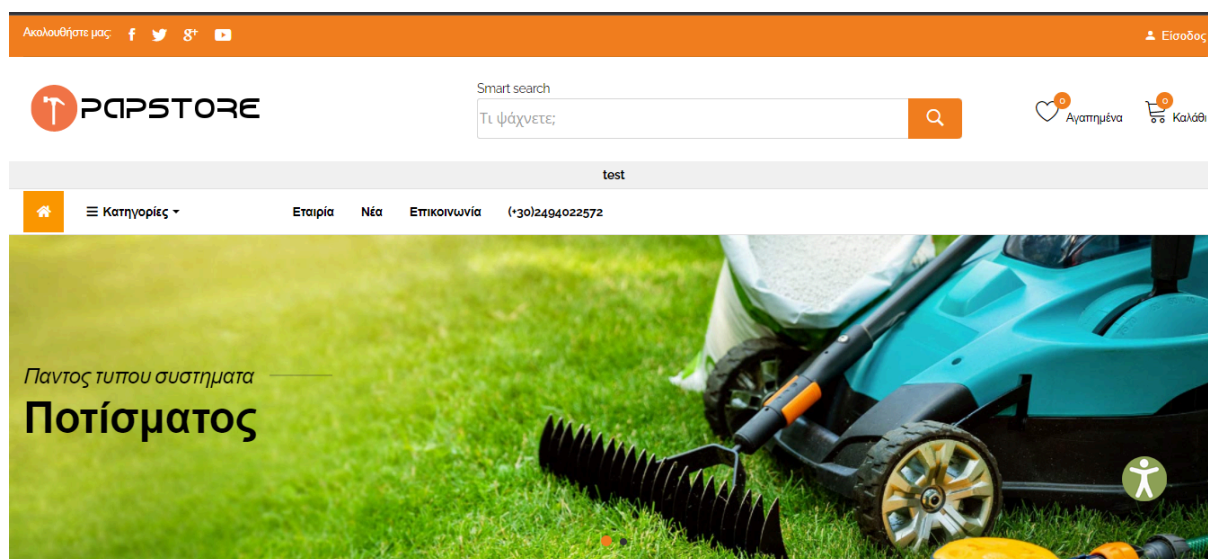
Με το πέρας της ανάλυσης όλης της βασικής τεχνικής υποδομής, όχι μόνο του back-end του ηλεκτρονικού καταστήματος, αλλά του γενικότερου υβριδικού συστήματος λειτουργίας, μπορούμε πλέον να παρουσιάσουμε και το τελικό αποτέλεσμα της εφαρμογής στο οποίο έχει πρόσβαση ο “τελικός χρήστης”: ο εκάστοτε επισκέπτης του ιστοτόπου του καταστήματος και υποψήφιος πελάτης της επιχείρισης.

Το γραφικό περιβάλλον του ηλεκτρονικού καταστήματος - ή αλλιώς “front-end” του ιστοτόπου - κατασκευάστηκε εξ’ ολοκλήρου με χρήση των γλωσσών προγραμματισμού HTML5, CSS, JavaScript και Ajax. Ο κώδικάς του στη συνέχεια ενσωματώνεται σε αρχεία Blade και εκτελείται από το back-end μέσω PHP - Laravel. Κατ’ αυτόν τον τρόπο, ο τελικός χρήστης, αν και έχει μια εικόνα εάν το επιθυμεί για την προγραμματιστική δομή του γραφικού περιβάλλοντος του ιστοτόπου, δεν έχει πλήρη επίγνωση του τρόπου με τον οποίο εκτελούνται ορισμένες διαδικασίες (π.χ. προβολή παραθύρων διαλόγου, προβολή φορμών συμπλήρωσης δεδομένων, κλπ.).

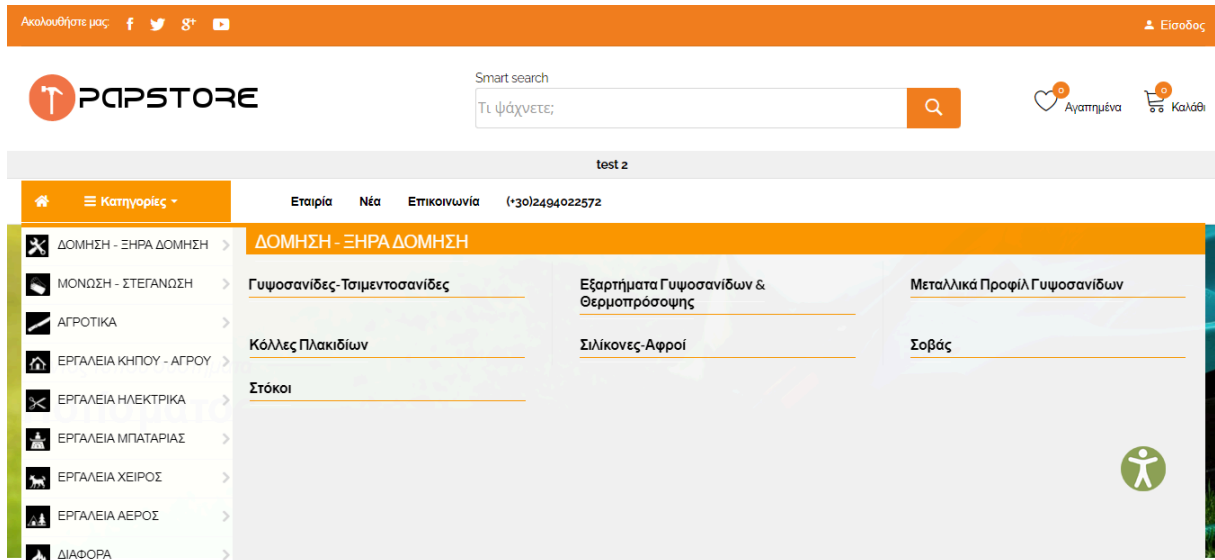
Με αυτά υπόψιν, επιλέγουμε να μην παρουσιάσουμε εντός αυτής της πτυχιακής εργασίας τον κώδικα με τον οποίο υλοποιείται το front-end, καθαρά για λόγους οικονομίας του εγγράφου. Φυσικά, κατόπιν αιτήματος παντός ενδιαφερόμενου, ο κώδικας μπορεί να διατεθεί για αξιολόγηση. Ακολουθεί η παρουσίαση των σημαντικότερων, κατ’ εμάς, στιγμιότυπων από το γραφικό περιβάλλον του ηλεκτρονικού καταστήματος.

4.2 Γραφικό Περιβάλλον Διεπαφής

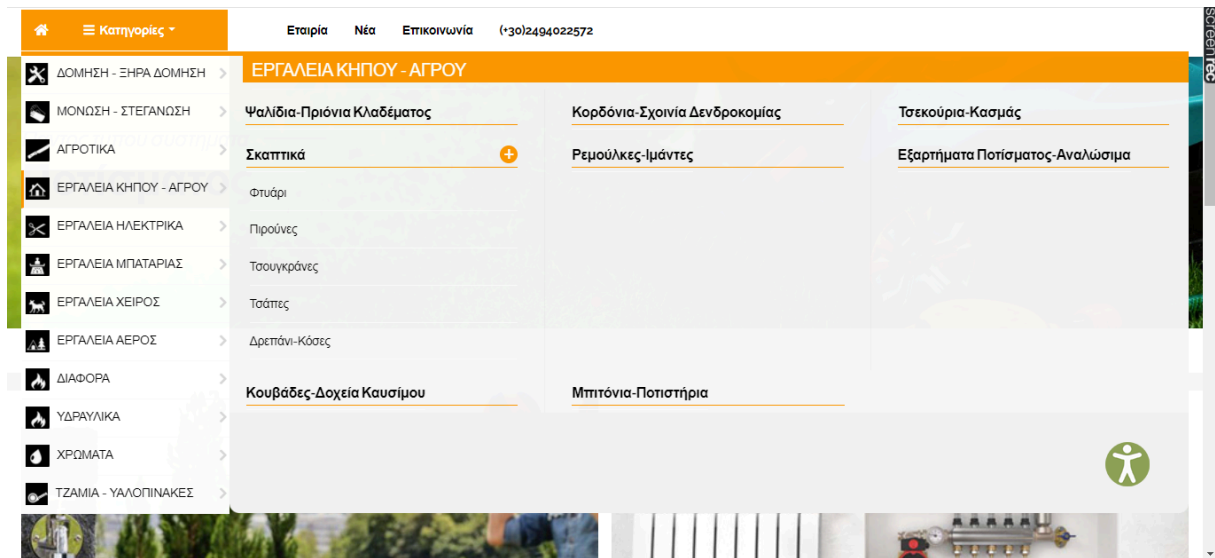
Αρχικά, στην Εικόνα 4.1 παρουσιάζεται ένα στιγμιότυπο της Αρχικής Σελίδας του ηλεκτρονικού Καταστήματος. Από εκεί, ο χρήστης μπορεί να συνδεθεί (ή να εγγραφεί, εάν δεν έχει ήδη λογαριασμό στο σύστημα), ή/και να περιηγηθεί στα προϊόντα του καταστήματος μέσω του πλευρικού μενού “Κατηγορίες”. Στιγμιότυπα του μενού αυτού παρουσιάζονται στις Εικόνες 4.2 και 4.3. Επιπλέον, επιλέγοντας τα μενού “Εταιρία”, “Νέα” και “Επικοινωνία” δύναται ανακατευθυνθεί σε σχετικές ιστοσελίδες όπου περιέχονται λίγα λόγια για την επιχείρησή μας, τα τελευταία νέα της που αφορούν κυρίως προσφορές, τυχούσες τιμολογιακές αλλαγές ή νέα προϊόντα, ή μια φόρμα επικοινωνίας με το κατάστημα, αντίστοιχα.



Εικόνα 4.1: Eshop: Αρχική Σελίδα

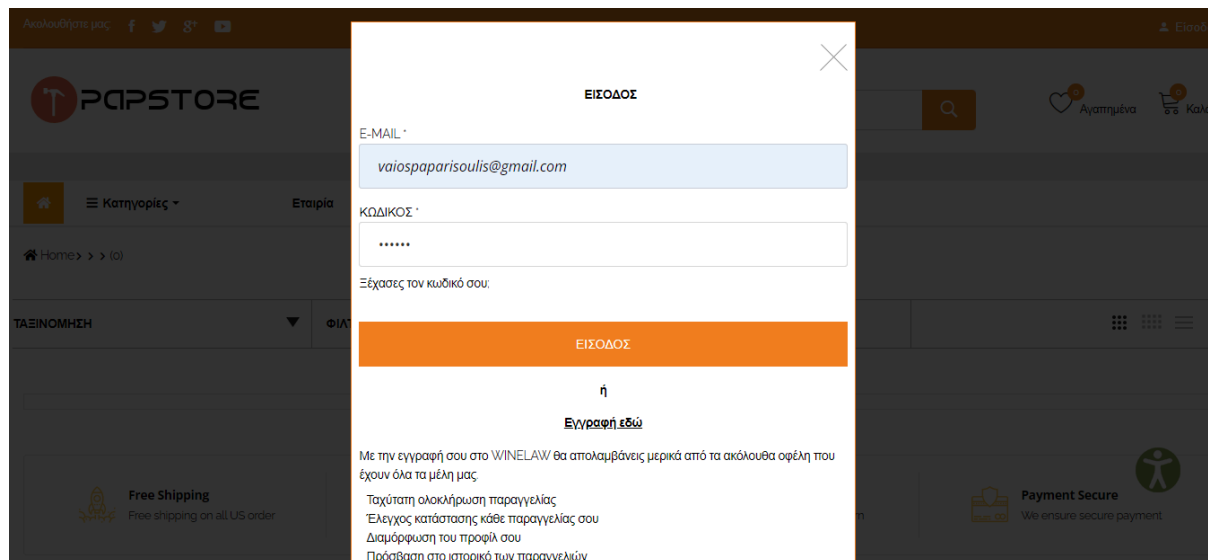


Εικόνα 4.2: Eshop: Πλευρικό Μενού “Κατηγορίες” - “Δόμηση - Ξηρά Δόμηση”



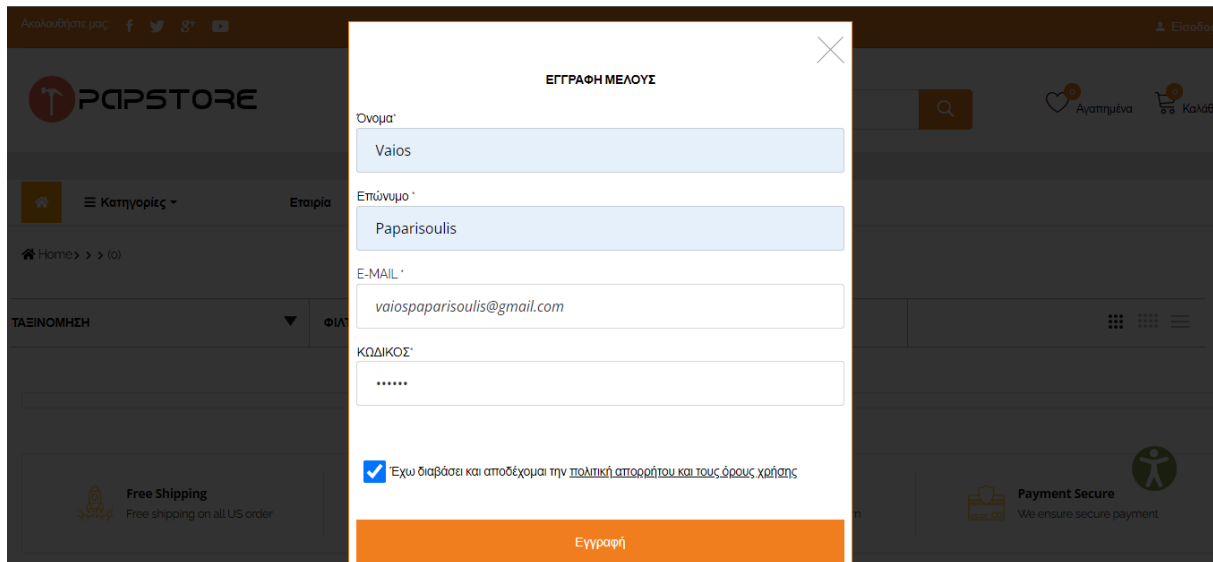
Εικόνα 4.3: Eshop: Πλευρικό Μενού “Κατηγορίες” - “Εργαλεία Κήπου - Αγρού”

Όσον αφορά την δημιουργία ή τη σύνδεση σε λογαριασμό, ο χρήστης μπορεί να εκκινήσει τη σχετική διαδικασία μέσω του πλήκτρου που βρίσκεται άνω δεξιά στην Αρχική Σελίδα του Καταστήματος (πρβλ. Εικόνα 4.1). Με το πάτημα του πλήκτρου, εμφανίζεται η φόρμα συμπλήρωσης στοιχείων σύνδεσης ονόματι “ΕΙΣΟΔΟΣ” (πρβλ. Εικόνα 4.4).



Εικόνα 4.4: Eshop: Σύνδεση Μέλους

Σε περίπτωση που ο χρήστης δεν έχει ακόμα λογαριασμό, με το πάτημα του συνδέσμου “Εγγραφή εδώ” ανακατευθύνεται σε ένα δεύτερο παράθυρο συμπλήρωσης στοιχείων εγγραφής μέλους, ονόματι “ΕΓΓΡΑΦΗ ΜΕΛΟΥΣ” (πρβλ. Εικόνα 4.5). Εκεί, συμπληρώνοντας τα στοιχεία του και αποδεχόμενος τους όρους χρήσης και της πολιτικής απορρήτου, ολοκληρώνει την εγγραφή του στο ηλεκτρονικό μας κατάστημα.

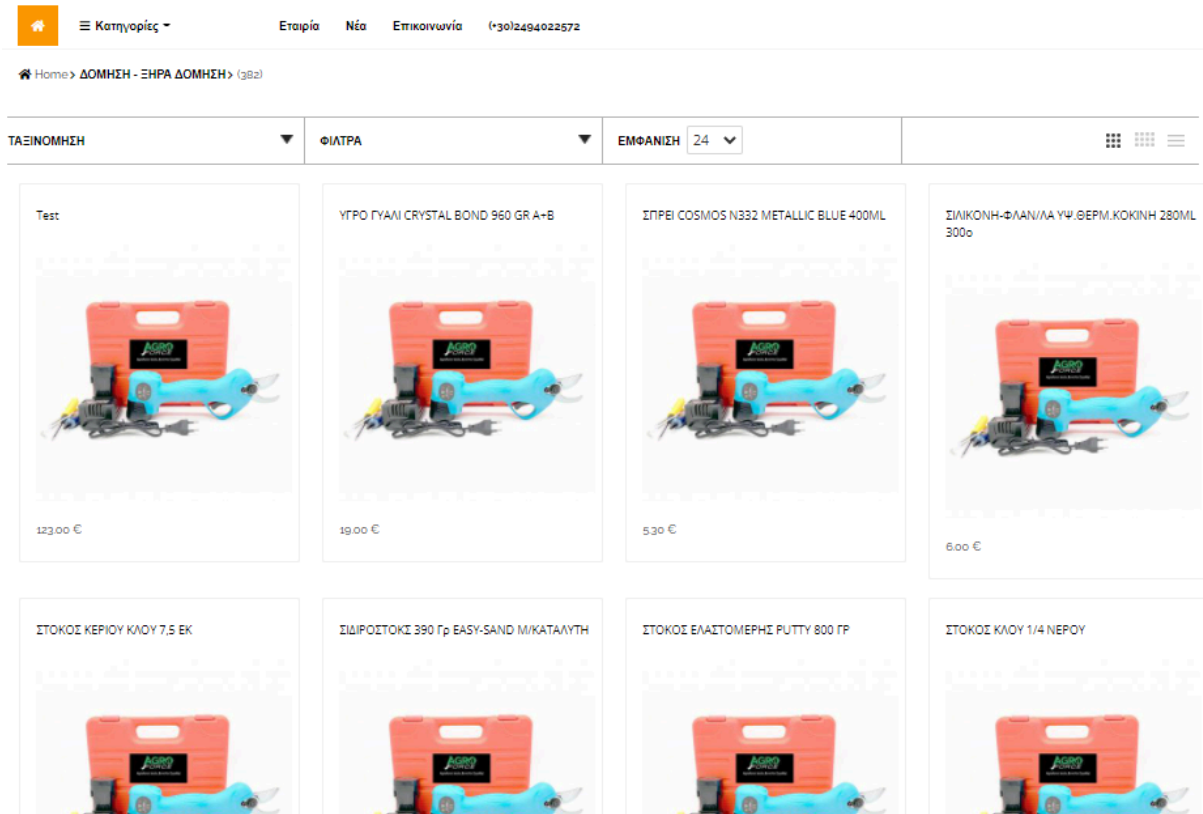


Εικόνα 4.5: Eshop: Εγγραφή Μέλους

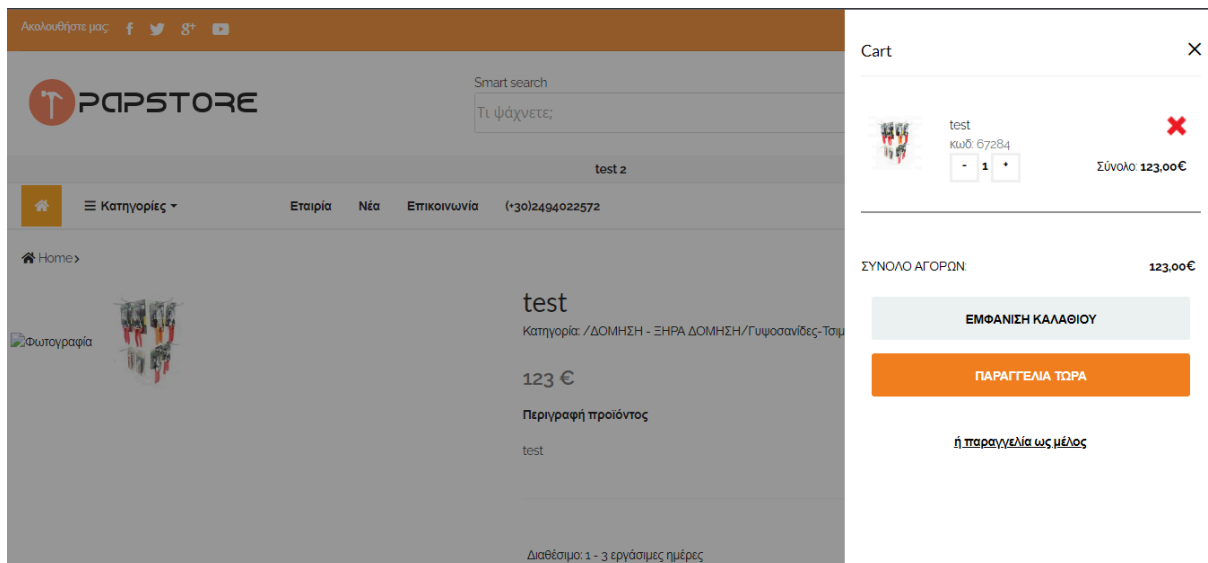
Με την επιλογή μιας Κατηγορίας (η υποκατηγορίας) εκ των διαθέσιμων, ο χρήστης ανακατευθύνεται στην αντίστοιχη Σελίδα Προϊόντων του καταστήματος (πρβλ. Εικόνα 4.6), από όπου μπορεί έπειτα να περιηγηθεί στα μεμονωμένα προϊόντα και να τα προσθέσει είτε στη “Λίστα Αγαπημένων” του, είτε στο Καλάθι Αγορών του. Υπενθυμίζεται ότι και οι δύο αυτές επιλογές είναι πάντα διαθέσιμες στον χρήστη, και μπορεί να τις αναθεωρήσει επιλέγοντάς τες από τα αντίστοιχα πλήκτρα στην σελίδα (πρβλ. παράδειγμα στην Εικόνα 4.1 άνω δεξιά).

Με την ολοκλήρωση της περιήγησής του, ο χρήστης δύναται να ολοκληρώσει την παραγγελία του επιλέγοντας στο Καλάθι Αγορών του (πρβλ. Εικόνα 4.7) την επιλογή “ΠΑΡΑΓΓΕΛΙΑ ΤΩΡΑ” ή “παραγγελια ως μέλος” (προεπιλογή εάν είναι συνδεδεμένος στο σύστημα). Με το πάτημα του πλήκτρου ανακατευθύνεται στην σελίδα Πληρωμών, όπου συμπληρώνει τα στοιχεία αποστολής της παραγγελίας (ήδη συμπληρωμένα εάν είναι συνδεδεμένος στο σύστημα) και επιλέγοντας ένα τρόπο πληρωμής (πρβλ. Εικόνα 4.8).

4 Σχεδιασμός και Υλοποίηση Ιστοσελίδας - Front-End



Εικόνα 4.6: Eshop: Σελίδα Προϊόντων



Εικόνα 4.7: Eshop: Αναθεώρηση Καλαθιού Αγορών - Checkout

Κατηγορίες ▾

Εταιρία
Νέα
Επικοινωνία
(+30)2494022572

ΠΑΡΑΓΓΕΛΙΑ

ΣΤΟΙΧΕΙΑ ΠΕΛΑΤΗ / ΔΙΕΥΘΥΝΣΗ ΑΠΟΣΤΟΛΗΣ

Όνομα
Επώνυμο
ΕΛΛΑΔΑ ▾
ΑΘΗΝΑ ▾
Διεύθυνση
Περιοχή
Ταχ. Κώδικας
Τηλέφωνο
Email

Αν επιθυμείτε ημολόγιο επιλέξτε εδώ

Αποστολή σε διαφορετική διεύθυνση

ΤΡΟΠΟΣ ΑΠΟΣΤΟΛΗΣ

ACS COURIER

Παράδοση στο χώρο μου

Παραμονή στην έδρα της μεταφορικής

ΤΡΟΠΟΣ ΠΛΗΡΩΜΗΣ

1 ΑΝΤΙΚΑΤΑΒΟΛΗ
Πληρωμή κατά την παράδοση (Χρέωση 2€)

2 ΠΙΣΤΩΤΙΚΗ/ΧΡΕΩΣΤΙΚΗ ΚΑΡΤΑ
Πληρωμή μέσω vinarpayments

3 Paypal
Πληρωμή μέσω Paypal

4 ΤΡΑΠΕΖΙΚΗ ΚΑΤΑΘΕΣΗ
Πληρωμή με τραπεζικό έμβασμα

* Συνεργαζόμενες courier: ACS, SPEEDEX, ΓΕΝΙΚΗ

* Για παράδοση αυθημερόν επικοινωνήστε μαζί μας (εντός Ν.Θεσσαλονίκης)

* Πρακτικά από το κατάστημα γίνεται μόνο στο κατάστημα της ΧΧΧΧΧΧΧΧ στη Λάρισα

Εικόνα 4.8: Eshop: Ολοκλήρωση και Αποστολή Παραγγελίας

4.3 Επίλογος

Με το πέρας του παρόντος Κεφαλαίου, στον αναγνώστη έχουν παρουσιαστεί τα κυρίως μέρη του Γραφικού Περιβάλλοντος Διεπαφής υπό μορφή του τυπικότερου σεναρίου χρήσης του ηλεκτρονικού καταστήματος. Παράλληλα, παρουσιάστηκαν τα αντίστοιχα στιγμιότυπα από τον ιστότοπο προκειμένου να έχει ο αναγνώστης μια πιο συνολική εικόνα του έργου το οποίο παρήχθη.

Κεφάλαιο 5ο: Σύνοψη

5.1 Εισαγωγή

Στο παρόν Κεφάλαιο, και τελευταίο αυτής της ΠΕ, συνοψίζεται όλη η προσπάθεια που έγινε στα πλαίσια αυτής της εργασίας. Πιο συγκεκριμένα, συνοψίζονται τα τελικά παραχθέντα αποτελέσματα του έργου και η τελική τεχνική υποδομή που προέκυψε, σύμφωνα με τους αρχικούς στόχους που είχαν τεθεί. Τέλος, παρουσιάζονται επιθυμητές μελλοντικές προεκτάσεις για το έργο, οι οποίες, ως ανοικτά ζητήματα, κρίθηκε πως δεν αφορούν άμεσα το ποιόν αυτής της ΠΕ, αλλά είναι στα άμεσα ενδιαφέροντα της εμπορικής αξιοποίησης του ηλεκτρονικού καταστήματος που υλοποιήθηκε.

5.2 Παραδοτέα Αποτελέσματα

Η παρούσα ΠΕ πραγματεύεται τον σχεδιασμό και την υλοποίηση ενός λειτουργικού ιστοτόπου ηλεκτρονικού καταστήματος επιχείρισης, με απώτερο σκοπό να αξιοποιηθεί εμπορικά από την ίδια επιχείριση στο εγγύς μέλλον. Ως επιπλέον χαρακτηριστικό τίθεται η λειτουργική απαίτηση στον σχεδιασμό του τελικού συνολικού συστήματος η δυνατότητα ενδοεπικοινωνίας με ήδη υφιστάμενο πρόγραμμα μηχανογράφησης το οποίο διατίθεται στο εμπόριο ως λογισμικό.

Με το πέρας της ΠΕ, παραδίδεται λειτουργικός ιστότοπος ηλεκτρονικού καταστήματος, με επαρκείς δυνατότητες κάλυψης των αναγκών της φυσικής επιχείρισης την οποία εξυπηρετεί: προσφέρει άμεσα την δυνατότητα σε απλούς επισκέπτες και πελάτες του καταστήματος να περιηγηθούν στα προσφερόμενα προϊόντα, να εκτελέσουν ηλεκτρονικά τις αγορές τους και να τις αποπληρώσουν, είτε είναι εγγεγραμμένα μέλη στο ηλεκτρονικό κατάστημα, είτε όχι. Επιπλέον, δίνεται η δυνατότητα στον διαχειριστή του καταστήματος να επιτηρεί κάθε περιεχόμενο στον ιστότοπο στοιχείο (προϊόντα, κατηγορίες, λογαριασμούς πελατών, παραγγελίες, κλπ), και να επιβάλλεται τυχόν σφαλμάτων τα οποία προκύπτουν. Όλα αυτά, χωρίς να χάνεται η λειτουργικότητα του ήδη υφιστάμενου προγράμματος αυτού: διακαναλική (είτε μέσω e-shop είτε μέσω προγράμματος μηχανογράφησης) καταχώρηση προϊόντων, επιτήρηση και διαχείριση του αποθέματος, επιτήρηση και διαχείριση του πελατολογίου, διαχείριση των προμηθευτών της επιχείρισης, διαχείριση των τιμολογίων της επιχείρισης, κλπ. Τέλος, το συνολικό σύστημα βασίζεται σε μια τεχνική υποδομή με δυνατότητες επέκτασης στο άμεσο μέλλον.

5.3 Μελλοντικές Επεκτάσεις

Ως αρχικές, βραχυπρόθεσμες επεκτάσεις του τελικού λογισμικού συστήματος που υλοποιήθηκε, θα μπορούσαν να αποτελούν κάποια από τα ανοικτά ζητήματα της υλοποίησης. Ένα εξ'αυτών αποτελεί η προσθήκη περισσότερων τρόπων ηλεκτρονικής πληρωμής στο κατάστημα. Ενώ ήδη έχει προβλεφθεί και υλοποιηθεί η πληρωμή μέσω PayPal επιθυμούμε πριν την έναρξη κιόλας

της εμπορικής λειτουργίας του ιστοτόπου να έχει ενσωματωθεί ακόμη η δυνατότητα πληρωμής μέσω VivaWallet και Ελληνικών Τραπεζών.

Επιπλέον, στα μεσοπρόθεσμα σχέδιά μας περιλαμβάνεται η δυνατότητα ενσωμάτωσης δυνατοτήτων λογισμικού Διαχείρισης Πελατειακών Σχέσεων (Customer Relationship Managemet Software - CRM) προκειμένου να εκμεταλλευθούμε την πληροφορία που μας δίνει η Βάση Δεδομένων και το Πρόγραμμα Μηχανογράφησης σχετικά με τα καταναλωτικά μοτίβα και τα στοιχεία των πελατών μας. Επιπλέον, θα θέλαμε να εκμεταλλευθούμε εν γένει όλη αυτή την πληροφορία που συγκεντρώνουμε από το ολικό σύστημα, και να την διοχετεύσουμε στον διαχειριστή του συστήματος - ιδιοκτήτη της επιχείρισης υπό μορφή οπτικοποιημένων δεδομένων, γραφημάτων, στατιστικών και μετρικών που σχετίζονται με τις εμπορικές επιδόσεις της επιχείρισης.

Τέλος, στα μακροπρόθεσμα σχέδιά μας είναι η πλήρης απορρόφηση του Προγράμματος Μηχανογράφησης στο δικό μας λογισμικό, είτε επεκτείνοντας το back-end της ιστοσελίδας προκειμένου να περιλαμβάνει όποια δυνατότητα του Προγράμματος Μηχανογράφησης εξυπηρετεί την επιχείριση, είτε δημιουργώντας μια δικιά μας εκδοχή του προγράμματος. Στόχος μας είναι η πλήρης οικονομική και λειτουργική ανεξαρτητοποίηση της επιχείρισης από το πρόγραμμα αυτό.

5.4 Επίλογος

Στο παρόν και τελευταίο Κεφάλαιο, ο αναγνώστης δύναται να ανακεφαλαιώσει εφ' όλης της ΠΕ αυτής, του τι ερευνήθηκε σε αυτή και του τι παραδόθηκε τελικά, όπως επίσης και να πληροφορηθεί για τις άμεσες και μακροχρόνιες ενδεχόμενες επεκτάσεις και βλέψεις για το παραδοτέο έργο.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] S. Tsur, S. Abiteboul, R. Agrawal, U. Dayal, J. Klein, and G. Weikum, “Are web services the next revolution in e-commerce? (panel),” in *Proceedings of the 27th International Conference on Very Large Data Bases, VLDB '01*, (San Francisco, CA, USA), p. 614–617, Morgan Kaufmann Publishers Inc., 2001.
- [2] E. Europe, “Ecommerce europe - greece,” 2020.
- [3] A. Drigas and P. Leliopoulos, “Business to consumer (b2c) e-commerce decade evolution,” *International Journal of Knowledge Society Research*, vol. 4, pp. 1–10, 10 2013.
- [4] J. Rowley, “Retailing and shopping on the internet,” *International Journal of Retail and Distribution Management*, vol. 24, 4 1996.
- [5] T. Zegkinoglou, “Ποιες είναι οι διαφορές μεταξύ ux design και ui design;,” 2021.
- [6] WHATWG, “Html living standard.” <https://html.spec.whatwg.org/multipage/>.
- [7] W. W. W. Consortium, “Html5 reference.” <https://dev.w3.org/html5/html-author/>.
- [8] I. Hickson and D. Hyatt, “Html5,” *W3C Working Draft WD-html5-20110525*, p. 53, 2011.
- [9] M. D. Network, “Javascript.” <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. Accessed: 2010-12-01.
- [10] M. D. Network, “Css developer guide.” <https://developer.mozilla.org/en-US/docs/Learn/CSS>. Accessed: 2010-12-01.
- [11] W. W. W. Consortium, “HTML & CSS what is css?.” <https://www.w3.org/standards/webdesign/htmlcss#whatcss>. Accessed: 2010-12-01.
- [12] S. S. Bakken, Z. Suraski, and E. Schmid, *PHP Manual: Volume 1*. iUniverse, Incorporated, 2000.
- [13] M. Hills, P. Klint, and J. J. Vinju, “Enabling php software engineering research in rascal,” *Science of Computer Programming*, vol. 134, pp. 37–46, 2017. 6th issue of Experimental Software and Toolkits (EST-6).
- [14] P. Kyriakakis and A. Chatzigeorgiou, “Maintenance patterns of large-scale php web applications,” pp. 381–390, 12 2014.
- [15] Laravel, “Laravel - the php framework for web artisans,” 2022.
- [16] J. Deacon, “Model-view-controller (mvc) architecture,” *Online*[[Citado em: 10 de março de 2006.] <http://www.jdl.co.uk/briefings/MVC.pdf>, 2009.

- [17] Blade, “The blade language documentation,” 2022.
- [18] A. Freier, P. Karlton, and P. Kocher, “The secure sockets layer (ssl) protocol version 3.0,” RFC 6101, Internet Engineering Task Force (IETF), 8 2011.
- [19] D. Wagner and B. Schneier, “Analysis of the ssl 3.0 protocol,” in *Proceedings of the 2nd Conference on Proceedings of the Second USENIX Workshop on Electronic Commerce - Volume 2*, WOEC’96, (USA), p. 4, USENIX Association, 1996.
- [20] T. Dierks and E. Rescorla, “The transport layer security (tls) protocol version 1.2,” RFC, vol. 5246, pp. 1–104, 2008.
- [21] E. Rescorla, “The transport layer security (tls) protocol version 1.3,” RFC, vol. 8446, pp. 1–160, 2018.
- [22] SSL.com, “What is ssl?,” 2022.
- [23] M. L. Das and N. Samdaria, “On the security of ssl/tls-enabled applications,” *Applied Computing and Informatics*, vol. 10, no. 1, pp. 68–81, 2014.
- [24] E. F. Codd, “A relational model of data for large shared data banks,” *Commun. ACM*, vol. 13, no. 6, pp. 377–387, 1970.
- [25] K. Islam, K. Ahsan, S. Bari, M. Saeed, and S. Asim, “Huge and real-time database systems: A comparative study and review for sql server 2016, oracle 12c & mysql 5.7 for personal computer,” *Journal of Basic and Applied Sciences*, vol. 13, pp. 481–490, 09 2017.
- [26] S. Shadlou, K. N. J., and A. Hajmoosaei, “Online payment via paypal api case study event registration management system (erms),” *International Journal of Web Portals*, vol. 3(2), pp. 30–37, 2011.
- [27] Microsoft Corporation, “Windows server 2016.”

ΠΑΡΑΡΤΗΜΑ Α: ΚΩΔΙΚΑΣ ΣΕ PHP - LARAVEL

A□.1 Π.Α.1: Class User

```
class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;
    protected $appends = ['has_password'];
    protected $fillable = [
        'name',
        'email',
        'password',
        'address',
        'phone',
        'is_banned',
        'last_active',
    ];
    protected $hidden = [
        'password',
        'remember_token',
    ];
    protected $casts = [
        'is_banned' => 'boolean',
        'is_admin' => 'boolean',
        'last_active' => 'datetime',
    ];
    public function favourites():
    \Illuminate\Database\Eloquent\Relations\BelongsToMany
    {
        return $this->belongsToMany(Product::class , 'favourites')
            ->as('favourites');
    }

    public function orders():
    \Illuminate\Database\Eloquent\Relations\HasMany
    {
        return $this->hasMany(Order::class);
    }

    public function cart():
```

```

\Illuminate\Database\Eloquent\Relations\HasOne
{
    return $this->hasOne(Cart::class);
}

public function getHasPasswordAttribute(): bool
{
    return !empty($this->attributes['password']);
}
}

```

A□.2 II.A.2: Class "Product"

```

class Product extends Model
{
    use HasFactory;
    protected $fillable=[
        'name',
        'description',
        'category_id',
        'stock',
        'image',
        'price',
        'show',
    ];
    public function category():
    \Illuminate\Database\Eloquent\Relations\BelongsTo
    {
        return $this->belongsTo(Category::class);
    }
    public function favourites():
    \Illuminate\Database\Eloquent\Relations\BelongsToMany
    {
        return $this->belongsToMany(User::class , 'favourites')
            ->as('favourites');
    }
    public function orders():
    \Illuminate\Database\Eloquent\Relations\BelongsToMany
    {
        return $this->belongsToMany(Order::class)

```

```

        ->withPivot('quantity' , 'each_price');
    }
    public function carts():
    \Illuminate\Database\Eloquent\Relations\BelongsToMany
    {
        return $this->belongsToMany(Cart::class)->withPivot('quantity');
    }
    public function scopeShop($query){
        $query->where('show' , 1)->with('favourites' , 'carts');
    }
}

```

A□.3 II.A.3: Class "Category"

```

class Category extends Model
{
    use HasFactory;

    protected $fillable=[
        'name',
        'parent_id',
        'show',
    ];
    public function parent():
    \Illuminate\Database\Eloquent\Relations\BelongsTo
    {
        return $this->belongsTo(self::class , 'parent_id');
    }
    public function children():
    \Illuminate\Database\Eloquent\Relations\HasMany
    {
        return $this->hasMany(self::class , 'parent_id');
    }
    public function products():
    \Illuminate\Database\Eloquent\Relations\HasMany
    {
        return $this->hasMany(Product::class);
    }
}

```

A□.4 II.A.4: Class "Cart"

```
class Cart extends Model
{
    use HasFactory;

    protected $fillable=[
        'user_id',
    ];

    public function user():
    \Illuminate\Database\Eloquent\Relations\BelongsTo
    {
        return $this->belongsTo(User::class);
    }

    public function products():
    \Illuminate\Database\Eloquent\Relations\BelongsToMany
    {
        return $this->belongsToMany(Product::class)->withPivot('quantity');
    }
}
```

A.5 II.A.5: Class "Order"

```
class Order extends Model
{
    use HasFactory;

    protected $fillable=[
        'user_id',
        'status',
        'ship_fee',
        'shipped_at',
        'delivered_at',
    ];

    protected $casts =[
        'shipped_at' => 'datetime',
        'delivered_at' => 'datetime',
    ];
}
```

```

];

public function user():
\Illuminate\Database\Eloquent\Relations\BelongsTo
{
    return $this->belongsTo(User::class);
}

public function products():
\Illuminate\Database\Eloquent\Relations\BelongsToMany
{
    return $this->belongsToMany(Product::class)
        ->withPivot('quantity' , 'each_price');
}
}

```

A□.6 II.A.6: Create "User"

```

class RegisteredUserController extends Controller
{

    public function create():
\Illuminate\View\View
{
    return view('auth.register');
}

    public function store(Request $request):
\Illuminate\Http\RedirectResponse
{
    $request->validate([
        'name' => ['required', 'string', 'max:255'],
        'email' => ['required', 'string',
            'email:rfc,dns', 'max:255', 'unique:users'],
        'address' => ['required' , 'string'],
        'phone' => ['required' , 'regex:/^(01)[012][0-9]{8}$/',
        'password' => ['required', 'confirmed',
            Rules\Password::min(6)],
    ]);
}

```

```

$user = User::create([
    'name' => $request->name,
    'email' => $request->email,
    'address' => $request->address,
    'phone' => $request->phone,
    'password' => Hash::make($request->password),
]);

$cart = Cart::create(['user_id' => $user->id]);
session()->put('cartId' , $cart->id);

Auth::login($user);

return redirect(RouteServiceProvider::HOME);
}
}

```

A□.7 II.A.7: Sign-in/Sign-out

```

class AuthenticatedSessionController extends Controller
{

    public function create()
    {
        return view('auth.login');
    }

    public function store(LoginRequest $request)
    {
        $request->authenticate();

        $request->session()->regenerate();

        session()->put('cartId',
            \auth()->user()->cart()->first()->id);

        return ($request->user()->is_admin)?
            redirect(route('dashboard'))
            : redirect(route('home'));
    }
}

```

```

}

public function destroy(Request $request)
{
    Auth::guard('web')->logout();

    $request->session()->invalidate();

    $request->session()->regenerateToken();

    return redirect('/');
}
}

```

A□.8 II.A.8: Reset Password

```

class PasswordResetLinkController extends Controller
{

    public function create()
    {
        return view('auth.forgot-password');
    }

    public function store(Request $request)
    {
        $request->validate([
            'email' => ['required', 'email'],
        ]);

        $status = Password::sendResetLink(
            $request->only('email')
        );

        return $status == Password::RESET_LINK_SENT
            ? back()->with('status', __($status))
            : back()->withInput($request->only('email'))
                ->withErrors(['email' => __($status)]);
    }
}
}

```

A□.9 II.A.9: Confirm New Password

```
class ConfirmablePasswordController extends Controller
{

  public function show()
  {
    return view('auth.confirm-password');
  }

  public function store(Request $request)
  {
    if (! Auth::guard('web')->validate([
      'email' => $request->user()->email,
      'password' => $request->password,
    ])) {
      throw ValidationException::withMessages([
        'password' => __('auth.password'),
      ]);
    }

    $request->session()
      ->put('auth.password_confirmed_at', time());

    return redirect()
      ->intended(RouteServiceProvider::HOME);
  }
}
```

A□.10 II.A.10: Store New Password

```
class NewPasswordController extends Controller
{

  public function create(Request $request)
  {
    return view('auth.reset-password', ['request' => $request]);
  }

  public function store(Request $request)
```

```

{
    $request->validate([
        'token' => ['required'],
        'email' => ['required', 'email'],
        'password' => ['required', 'confirmed',
            Rules\Password::min(6)],
    ]);

    $status = Password::reset(
        $request->only('email', 'password',
            'password_confirmation', 'token'),
        function ($user) use ($request) {
            $user->forceFill([
                'password' => Hash::make($request->password),
                'remember_token' => Str::random(60),
            ]->save());

            event(new PasswordReset($user));
        }
    );

    return $status == Password::PASSWORD_RESET
        ? redirect()->route('login')
            ->with('status', __($status))
        : back()->withInput($request->only('email'))
            ->withErrors(['email' => __($status)]);
}
}

```

A□.11 II.A.11: Update User Info

```

class UpdateUserController extends Controller
{

    public function edit():
        \Illuminate\View\View
    {
        return view('auth.update');
    }
}

```

```

public function update(UpdateDataRequest $request):
\Illuminate\Http\RedirectResponse
{
    auth()->user()->update($request->validated());
    return redirect(route('user.update'))
        ->with('success' , 'Updated successfully');
}

public function editPassword():
\Illuminate\View\View
{
    return view('auth.update-password');
}

public function updatePassword(UpdatePasswordRequest $request):
\Illuminate\Http\RedirectResponse
{
    auth()->user()->update([
        'password' => Hash::make($request->validated()['password']),
    ]);
    return redirect(route('user.update'))
        ->with('success' , 'Password updated successfully');
}
}

```

A□.12 II.A.12: Product Pages

```

class ShowController extends Controller
{

    public function home():
\Illuminate\View\View
{
    $products = Product::shop()
        ->withCount('favourites' , 'orders')
        ->get();
    $sliderProducts = $products
        ->sortBy('favourites_count')
        ->take(2);
    $trendyProducts = $products

```

```

        ->sortBy('orders_count')
            ->take(8);
$categories = Category::has('products')
            ->withCount('products')
                ->take(3);
return view('user.home', compact('sliderProducts', 'categories',
                                'products' , 'trendyProducts'));
}

public function index(string $sort = null):
\Illuminate\View\View
{
    $products = ($sort)
    ? Product::shop()->orderBy($sort , 'desc')->paginate(9)
    : Product::shop()->paginate(9);
    $categories = $this->getCategories();
    return view('user.shop' , compact('products' , 'categories'));
}

public function category(int $id):
\Illuminate\View\View
{
    $products = Product::shop()
        ->where('category_id' , $id)->paginate(9);
    $categories = $this->getCategories();
    return view('user.shop' , compact('products' , 'categories'));
}

public function filter(Request $request):
\Illuminate\View\View
{
    $request->validate(['categories'
        => ['required'] , 'categories.*'
        => ['integer' , 'exists:categories,id'] ]);
    $categories = $this->getCategories();
    $categoriesIDs = array_merge(
        Category::whereIn('parent_id', $request
            ->input('categories'))
            ->pluck('id')->all()
        , $request->input('categories'));
}

```

```

    $products = Product::shop()
        ->whereIn('category_id' , $categoriesIDs)
        ->paginate(9);
    return view('user.shop' , compact('products' , 'categories'));
}

public function search(Request $request):
    \Illuminate\View\View
{
    $request->validate(['product' => ['required' , 'string'] ]);
    $search = $request->input('product');
    $categories = $this->getCategories();
    $products = Product::shop()
        ->where('name' , 'like' , '%' . $search . '%')
        ->paginate(9);
    return view('user.shop', compact('products',
                                    'search', 'categories'));
}

public function show(Product $product):
    \Illuminate\View\View
{
    $product->load('favourites');
    $products = Product::with('favourites', 'carts')
        ->where('show', 1)
        ->where('category_id', $product->category_id)
        ->where('id' , '!=' , $product->id)->get();
    return view('user.details', compact('product', 'products'));
}

private function getCategories(){
    return Category::withCount('products')->get();
}
}

```

A□.13 II.A.13: "Favourites"

```

class CreateController extends Controller
{
    public function store(Product $product):

```

```

\Illuminate\Http\JsonResponse
{
    $result = $product->favourites()->toggle(auth()->id());
    return (count($result['attached']) > 0)
        ? response()->json(status: 201)
        : response()->json(status: 200);
}
}

```

A□.14 II.A.14: Show "Favourites"

```

class ShowController extends Controller
{
    public function index(): \Illuminate\View\View
    {
        $products = auth()->user()->favourites()->paginate(6);
        return view('user.favourites' , compact('products'));
    }
}

```

A□.15 II.A.15: Add to Cart

```

class CreateController extends Controller
{
    public function store(StoreCartRequest $request)
        :\Illuminate\Http\RedirectResponse
    {
        $product = Product::findOrFail($request
            ->validated()['product_id']);
        $product->carts()->attach(session()->get('cartId'));
        return redirect(route('cart'));
    }
}

```

A□.16 II.A.16: Update Cart

```
class EditController extends Controller
{

    public function update(UpdateCartRequest $request):
    \Illuminate\Http\RedirectResponse
    {
        $validated = $request->validated();
        $product = Product::findOrFail($validated['product_id']);
        $product->carts()->updateExistingPivot(session()
            ->get('cartId') , ['quantity' => $validated['quantity']]);
        return redirect(route('cart'));
    }

    public function destroy(UpdateCartRequest $request):
    \Illuminate\Http\RedirectResponse
    {
        $product = Product::findOrFail($request
            ->validated()['product_id']);
        $product->carts()->detach(session()->get('cartId'));
        return redirect(route('cart'));
    }
}
```

A□.17 II.A.17: Show Cart

```
class ShowController extends Controller
{

    public function index(CartTotalCalcService $service):
    \Illuminate\View\View
    {
        $products = auth()->user()->cart()->first()->products()->get();
        $total = $service->calc($products);
        return view('user.cart' , compact('products' , 'total'));
    }
}
```

A□.18 Π.A.18: PayPal Integration

```
class PayPalPaymentController extends Controller
{
    public function handlePayment()
    {
        $product = [];
        $product['items'] = [
            [
                'name' => 'Vaios Papisoulis',
                'price' => 112,
                'desc' => 'Running shoes',
                'qty' => 2
            ]
        ];

        $product['invoice_id'] = 1;
        $product['invoice_description'] =
            "Order #{$product['invoice_id']} bill";
        $product['return_url'] = route('success.payment');
        $product['cancel_url'] = route('cancel.payment');
        $product['total'] = 224;

        $paypalModule = new ExpressCheckout;

        $res = $paypalModule->setExpressCheckout($product);
        $res = $paypalModule->setExpressCheckout($product, true);

        return redirect($res['paypal_link']);
    }

    public function paymentCancel()
    {
        dd('Your payment has been declined.');
```

```

        $paypalModule->getExpressCheckoutDetails($request->token);

        if (in_array(strtoupper($response['ACK']),
            ['SUCCESS', 'SUCCESSWITHWARNING'])) {
            dd('Payment was successfull. ');
        }

        dd('Error occured!');
    }
}

```

A□.19 II.A.19: Add New Product

```

class CreateController extends Controller
{
    use ImageHandler;

    public function create():
        \Illuminate\View\View
    {
        $categories = Category::select('id' , 'name')->get();
        return view('admin.products.create' , ['categories' => $categories]);
    }

    public function store(StoreProductRequest $request):
        \Illuminate\Http\RedirectResponse
    {
        $data = $request->validated();
        $data['image'] = $this->saveImage($data['image'] , 'products');
        Product::create($data);
        return redirect(route('dashboard.products'))
            ->with('success' , 'Product has been added successfully');
    }
}

```

A□.20 II.A.20: Edit Product

```

class EditController extends Controller
{

```

```

use ImageHandler;

public function edit(Product $product):
\Illuminate\View\View
{
    $product->load('category:id,name');
    $categories = Category::where('id', '!=', $product->category_id)
        ->get();
    return view('admin.products.edit', compact('product',
        'categories'));
}

public function update(UpdateProductRequest $request , Product $product):
\Illuminate\Http\RedirectResponse
{
    $product->update($request->validated());
    return redirect(route('dashboard.products'))
        ->with('success' , 'Product has been updated successfully');
}

public function updateImage(Request $request,Product $product):
\Illuminate\Http\RedirectResponse
{
    $request->validate(['image' => ['required' , 'image' , 'max:1024']]);
    $this->deleteImage($product->image);
    $image = $this->saveImage($request->file('image') , 'products');
    $product->update(['image' => $image]);
    return redirect(route('dashboard.products'))
        ->with('success',
            'Product\'s image has been updated successfully');
}

public function destroy(Product $product):
\Illuminate\Http\RedirectResponse
{
    $this->deleteImage($product->image);
    $product->delete();
    return redirect(route('dashboard.products'))
        ->with('success' , 'Product has been deleted successfully');
}

```

```
}
```

A□.21 II.A.21: Show Product

```
class ShowController extends Controller
{

    public function index():
    \Illuminate\View\View
    {
        $products = Product::with('category')
            ->withCount(['favourites' , 'orders']->get());
        return view('admin.products.index' , ['products' => $products]);
    }

    public function search(Request $request):
    \Illuminate\View\View
    {
        $validated = $request
            ->validate(['search' => ['string' , 'required']]);
        $products = Product::with('category')
            ->withCount(['favourites' , 'orders'])
            ->where('name' , 'like' , '%' . $validated['search'] . '%')
            ->get();
        return view('admin.products.index' ,
            ['products' => $products , 'searchFor'
                => $validated['search']]);
    }
}
}
```

A□.22 II.A.22: Add New Category

```
class CreateController extends Controller
{

    public function create():
    \Illuminate\View\View
    {
```

```

        $categories = Category::whereNull('parent_id')->get();
        return view('admin.categories.create' , compact('categories'));
    }

    public function store(StoreCategoryRequest $request):
    \Illuminate\Http\RedirectResponse
    {
        Category::create($request->validated());
        return redirect(route('dashboard.categories')) \
            ->with('success' , 'Category has been added successfully');
    }
}

```

A□.23 II.A.23: Edit Category

```

class EditController extends Controller
{

    public function edit(int $id):
    \Illuminate\View\View
    {
        $category = Category::with('parent')->findOrFail($id);
        $categories = Category::whereNotIn('id' ,
            [$id , $category->parent_id ?? $id])
            ->whereNull('parent_id')->get();
        return view('admin.categories.edit' ,
            compact('category' , 'categories'));
    }

    public function update(StoreCategoryRequest $request ,Category $category):
    \Illuminate\Http\RedirectResponse
    {
        $data = $request->validated();
        $data['parent_id'] = $request['parent_id'] ?? null;
        $category->update($data);
        return redirect(route('dashboard.categories'))
            ->with('success' , 'Category has been updated successfully');
    }
}

```

```

public function destroy(Category $category):
\Illuminate\Http\RedirectResponse
{
    $category->delete();
    return redirect(route('dashboard.categories'))
        ->with('success' , 'Category has been deleted successfully');
}
}

```

A□.24 II.A.24: Show Vategory

```

class ShowController extends Controller
{

    public function index():
\Illuminate\View\View
    {
        $categories = Category::with('children')
            ->whereNull('parent_id')->get();
        return view('admin.categories.index' , compact('categories'));
    }

    public function show(Category $category):
\Illuminate\View\View
    {
        $category->load(['products' , 'children']);
        return view('admin.categories.show' , compact('category'));
    }
}

```

A□.25 II.A.25: Add New Order

```

class CreateController extends Controller
{

    public function create(CartTotalCalcService $totalService,
                        OrderShipCalcService $shipService):
\Illuminate\View\View

```

```

    {
        $products = Cart::findOrFail(session()
            ->get('cartId'))->products()->get();
        $ship = $shipService->calc();
        $total = $totalService->calc($products);
        return view('user.checkout',
            compact('products' , 'total' , 'ship'));
    }

    public function store(OrderShipCalcService $shipService,
        OrderProductsAttachService $attachService):
        \Illuminate\Http\RedirectResponse
    {
        $ship = $shipService->calc();
        $order = Order::create(['user_id' => auth()->id(),
            'ship_fee' => $ship ,]);
        $attachService->attach($order);
        Cart::findOrFail(session()
            ->get('cartId'))->products()->detach();
        return redirect(route('checkout'))
            ->with('success',
                'Order has been placed successfully and its number is '.
                    $order->id);
    }
}

```

A□.26 II.A.26: Edit Order

```

class EditController extends Controller
{
    use StockHandler;

    public function ship(Order $order):
        \Illuminate\Http\RedirectResponse
    {
        $order->update(['status' => 'shipped',
            'shipped_at' => now()]);
        $this->shipped($order);
        return redirect(route('dashboard.orders'))
    }
}

```

```

        ->with('success' , 'Order has been shipped successfully');
    }

    public function deliver(Order $order):
    \Illuminate\Http\RedirectResponse
    {
        $order->update(['status' => 'delivered',
            'delivered_at' => now()]);
        return redirect(route('dashboard.orders'))
            ->with('success' , 'Order has been delivered successfully');
    }

    public function cancel(Order $order):
    \Illuminate\Http\RedirectResponse
    {
        $order->update(['status' => 'cancelled']);
        return redirect(route('dashboard.orders'))
            ->with('success' , 'Order has been cancelled successfully');
    }

    public function restore(Order $order):
    \Illuminate\Http\RedirectResponse
    {
        $order->update(['status' => 'restored']);
        $this->restore($order);
        return redirect(route('dashboard.orders'))
            ->with('success' , 'Order has been restored successfully');
    }
}

```

A□.27 II.A.27: Show Order

```

class ShowController extends Controller
{
    use TotalPriceCalculator;

    public function index(string $status = null):
    \Illuminate\View\View
    {

```

```

        $orders = ($status &&
                    in_array($status ,
                            ['pending', 'shipped' ,
                            'delivered', 'cancelled', 'restored']))
        ? Order::with('user')
          ->with('products')->where('status', $status)->paginate(10)
        : Order::with('user')
          ->with('products')->paginate(10);
    $this->calculateEveryTotalPrice($orders);
    return view('admin.orders.index' , compact('orders'));
}

public function show(Order $order):
    \Illuminate\View\View
{
    $order->load(['products' , 'user']);
    $this->calculateOneTotalPrice($order);
    return view('admin.orders.show' , compact('order'));
}
}

```

A□.28 II.A.28: Ban/Unban User

```

class EditController extends Controller
{

    public function ban(User $user):
        \Illuminate\Http\RedirectResponse
    {
        if($user->is_banned){
            $message = 'unbanned';
            $user->update(['is_banned' => 0]);
        }else{
            $message = 'banned';
            $user->update(['is_banned' => 1]);
        }
        return redirect(route('dashboard.users'))
            ->with('success' , "User has been $message successfully");
    }
}

```

```
}
```

A□.29 II.A.29: Show User Info

```
class ShowController extends Controller
{
  use TotalPriceCalculator;

  public function index():
  \Illuminate\View\View
  {
    $users = User::where('is_admin', 0)->get();
    return view('admin.users.index', compact('users'));
  }

  public function show(User $user):
  \Illuminate\View\View
  {
    $user->load('orders');
    $this->calculateEveryTotalPrice($user->orders);
    return view('admin.users.show',
      compact('user'));
  }
}
```