



ΔΙΕΘΝΕΣ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ  
ΑΝΑΠΤΥΞΗ ΠΛΑΤΦΟΡΜΑΣ ΔΗΜΟΠΡΑΣΙΩΝ



Των φοιτητών:  
Νικολάου Σαπουντζή 164824  
Γεωργίου Χαραλαμπίδη 164774

Επιβλέπων  
Παναγιώτης Τζέκης

Τίτλος Δ.Ε. Ανάπτυξη Πλατφόρμας Δημοπρασιών  
Κωδικός Δ.Ε. 23220

Όνοματεπώνυμο φοιτητών Νικόλαος Σαπουντζής και Γεώργιος Χαραλαμπίδης

Όνοματεπώνυμο εισηγητή Παναγιώτης Τζέκης

Ημερομηνία ανάληψης Δ.Ε. 13/05/2023

Ημερομηνία περάτωσης Δ.Ε. 13/05/2025

*Βεβαιώνω ότι είμαστε οι συγγραφείς αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχαμε για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχουμε καταγράψει τις όποιες πηγές από τις οποίες κάναμε χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνουμε ότι αυτή η εργασία προετοιμάστηκε από εμάς προσωπικά, ειδικά ως πτυχιακή εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.Π.Α.Ε.*

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

Η έγκριση της 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 παρούσα πτυχιακή εργασία αποτελεί ένα ταξίδι στον κόσμο της τεχνολογίας και του διαδικτύου, στο πλαίσιο μιας αναζήτησης για την ανάπτυξη ενός διαδικτυακού καταστήματος δημοπράτησης προϊόντων. Με βάση τις σύγχρονες τεχνολογίες και τις αρχές του ανοικτού κώδικα, η εργασία αυτή αποσκοπεί στη δημιουργία ενός ολοκληρωμένου συστήματος που συνδυάζει την αποτελεσματική διαχείριση δημοπρασιών με την εύκολη πρόσβαση για τους χρήστες.

Μέσα από αυτήν την εργασία, εξερευνούμε τις πολλαπλές πτυχές της τεχνολογικής εφαρμογής, αναλύοντας τις γλώσσες προγραμματισμού και τα εργαλεία ανάπτυξης, καθώς και τις τεχνικές διαχείρισης βάσεων δεδομένων. Στόχος μας είναι να προσφέρουμε μια ολοκληρωμένη εικόνα των δυνατοτήτων και των διαδικασιών που απαιτούνται για τη δημιουργία και τη λειτουργία ενός δυναμικού και επιτυχημένου διαδικτυακού καταστήματος.

Με την ανάπτυξη αυτής της πτυχιακής εργασίας, ελπίζουμε να προσφέρουμε μια χρήσιμη προσέγγιση στον κόσμο του ηλεκτρονικού εμπορίου και να εμπνεύσουμε μελλοντικές γενιές προγραμματιστών και επιχειρηματιών να εξερευνήσουν τις ατέλειωτες δυνατότητες που προσφέρει ο κόσμος της τεχνολογίας.

## Περίληψη

Η παρούσα πτυχιακή εργασία αναλύει την ανάπτυξη ενός διαδικτυακού καταστήματος δημοπράτησης προϊόντων, εστιάζοντας στην αποτελεσματική διαχείριση δημοπρασιών και την εύκολη πρόσβαση για τους χρήστες. Χρησιμοποιώντας σύγχρονες τεχνολογίες και αρχές ανοικτού κώδικα, εξερευνά τις γλώσσες προγραμματισμού, τα εργαλεία ανάπτυξης και τις τεχνικές διαχείρισης βάσεων δεδομένων που απαιτούνται για την υλοποίηση του καταστήματος. Ο στόχος είναι να παρέχει μια ολοκληρωμένη εικόνα των δυνατοτήτων και των διαδικασιών που απαιτούνται για τη δημιουργία ενός δυναμικού και επιτυχημένου ηλεκτρονικού καταστήματος. Μέσα από την προσέγγισή της, ελπίζει να εμπνεύσει μελλοντικούς προγραμματιστές και επιχειρηματίες να εξερευνήσουν τον κόσμο της τεχνολογίας και του ηλεκτρονικού εμπορίου.

# «SYSTEM FOR ONLINE AUCTIONS AND SALES»

«Nikolaos Sapountzis and Georgios Charalampidis»

## **Abstract**

This thesis explores the development of an online store for product auctions, focusing on effective auction management and user-friendly access. Utilizing modern technologies and open-source principles, it investigates programming languages, development tools, and database management techniques required for the store's implementation. The goal is to provide a comprehensive overview of the capabilities and processes necessary for creating a dynamic and successful e-commerce platform. Through its approach, it aims to inspire future programmers and entrepreneurs to explore the world of technology and e-commerce.

# Περιεχόμενα

Πρόλογος.....	v
Περίληψη.....	vi
Abstract .....	vii
Περιεχόμενα .....	viii
Συνομογραφίες.....	xi
Κεφάλαιο 1ο: Εισαγωγή στην Πτυχιακή Εργασία .....	1
1.1 Εισαγωγή.....	1
1.2 Σκοπός της εργασίας .....	1
1.3 Οργάνωση της εργασίας.....	1
1.4 Επίλογος.....	2
Κεφάλαιο 2ο: Επισκόπηση Αντίστοιχων Προγραμμάτων Δημοπρασιών .....	2
2.1 Εισαγωγή.....	2
2.2 Δημοφιλή Προγράμματα Δημοπρασιών .....	3
2.2.1 eBay.....	3
2.2.2 Catawiki.....	4
2.2.3 AuctionZip.....	6
2.3 Συγκριτική ανάλυση.....	7
2.4 Επίλογος.....	8
Κεφάλαιο 3ο: Εργαλεία και Γλώσσες Προγραμματισμού .....	3
3.1 Εισαγωγή.....	3
3.2 HTML/CSS .....	3
3.2.1 BootStrap.....	5
3.3 JavaScript .....	6
3.3.1 AJAX.....	7
3.3.2 JSON .....	8
3.4 PHP.....	9
3.4.1 Laravel.....	10
3.4.2 Routing (Δρομολόγηση).....	10
3.4.3 Controllers (Ελεγκτές).....	11
3.4.4 Views (Προβολές).....	11
3.4.5 Διαχείριση Εξαρτήσεων (Dependency Management).....	12
3.4.6 Εργαλεία Εντολών (Command-Line Tools).....	12

3.5	Βάση Δεδομένων και Συστήματα Διαχείρισης Βάσεων Δεδομένων .....	12
3.5.1	SQL .....	13
3.5.2	MariaDB.....	14
3.6	Τεχνικές υλοποίησης και συνδυασμός όλων των προαναφερθέντων τεχνολογιών .....	14
3.7	Visual Studio Code.....	16
3.8	Επίλογος.....	16
Κεφάλαιο 4ο:	Τεχνικά Δομικά Στοιχεία Έργου .....	18
4.1	Ρύθμιση του περιβάλλοντος εργασίας.....	18
4.2	Μεταφόρτωση του Laravel framework .....	18
4.3	Ρύθμιση των routes.....	18
4.4	Εγγραφή, είσοδος και προφίλ των χρηστών.....	19
4.5	Μενού.....	20
4.6	Εμφάνιση και εισαγωγή προϊόντων για το διαχειριστή.....	20
4.7	Εισαγωγή δημοπρασίας από το διαχειριστή.....	21
4.8	Αποστολή προϊόντος κατοχυρωμένης δημοπρασίας σε πελάτη από το διαχειριστή .....	21
4.9	Εμφάνιση ενεργής δημοπρασίας .....	21
4.10	Υποβολή προσφοράς για ενεργή δημοπρασία .....	22
4.11	Αυτόματη ενημέρωση ενεργής δημοπρασίας.....	22
4.12	Εμφάνιση δημοπρασίας με βάση το id της (ενεργή ή μη).....	23
4.13	Αναζήτηση δημοπρασίας .....	23
4.14	Αναφορές δημοπρασιών για το διαχειριστή.....	24
4.15	Περιγραφή του σχήματος της βάσης δεδομένων .....	24
4.16	Επίλογος.....	27
Κεφάλαιο 5ο:	Στιγμιότυπα Χρήσης της Εφαρμογής.....	28
	BIBΛΙΟΓΡΑΦΙΑ.....	33
	ΠΑΡΑΡΤΗΜΑ Α : Κώδικας routes/web.php .....	34
	ΠΑΡΑΡΤΗΜΑ Β : Κώδικας αρχείου register.....	35
	ΠΑΡΑΡΤΗΜΑ C : Κώδικας μεθόδου process_register.....	36
	ΠΑΡΑΡΤΗΜΑ D : Κώδικας αρχείου view login .....	37
	ΠΑΡΑΡΤΗΜΑ E : Κώδικας μεθόδου process login.....	38
	ΠΑΡΑΡΤΗΜΑ F : Κώδικας μεθόδου profile .....	40
	ΠΑΡΑΡΤΗΜΑ G : Κώδικας menu εφαρμογής .....	41
	ΠΑΡΑΡΤΗΜΑ Η : Κώδικας διαχειριστικού .....	42
	ΠΑΡΑΡΤΗΜΑ I : Κώδικας εισαγωγής δημοπρασίας.....	45
	ΠΑΡΑΡΤΗΜΑ J : Κώδικας κατοχύρωσης δημοπρασίας σε πελάτη .....	47

ΠΑΡΑΡΤΗΜΑ Κ : Κώδικας ενεργής δημοπρασίας .....	48
ΠΑΡΑΡΤΗΜΑ Λ : Κώδικας ενεργής προσφοράς.....	51
ΠΑΡΑΡΤΗΜΑ Μ : Κώδικας αυτόματης ενημέρωσης δημοπρασίας .....	53
ΠΑΡΑΡΤΗΜΑ Ν : Κώδικας εμφάνισης δημοπρασίας με βάση το id .....	54
ΠΑΡΑΡΤΗΜΑ Ο : Κώδικας αναζήτησης δημοπρασίας.....	57
ΠΑΡΑΡΤΗΜΑ Ρ : Κώδικας αναφορών δημοπρασιών .....	60

## Συντομογραφίες

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



# Κεφάλαιο 1ο: Εισαγωγή στην Πτυχιακή Εργασία

## 1.1 Εισαγωγή

Το αντικείμενο της παρούσας πτυχιακής εργασίας είναι η μελέτη και έπειτα η χρήση των τεχνολογιών Ιστού, ή αλλιώς των Web τεχνολογιών, με τις οποίες θα γίνει υλοποίηση ενός συστήματος διαδικτυακών δημοπρασιών και πωλήσεων προϊόντων για ένα ηλεκτρονικό κατάστημα *eshop*. Η μεγαλύτερη έμφαση έχει δοθεί στις λειτουργίες που αφορούν την ηλεκτρονική διαχείριση και δημοπράτηση προϊόντων τα οποία εμπορεύεται το ηλεκτρονικό κατάστημα. Η απευθείας πώληση προϊόντων δεν προβλέπεται από τη διαχείριση του καταστήματος και η δημοπράτησή τους είναι ο μόνος τρόπος πώλησης.

Η υλοποίηση της εφαρμογής βασίζεται σε σύγχρονες τεχνολογίες ανοικτού κώδικα *open-source*, οι οποίες έχουν χρησιμοποιηθεί εκτενώς από επιχειρήσεις και ιδιώτες για πολλά έτη. Συγκεκριμένα, ο *server* που φιλοξενεί την εφαρμογή-ιστοσελίδα είναι ο *Apache*, ενώ η επικοινωνία με τον *server* γίνεται μέσω της γλώσσας προγραμματισμού *PHP*. Ως βάση δεδομένων χρησιμοποιείται η *MariaDB*. Στο επίπεδο πελάτη χρησιμοποιούνται η *HTML* για την υλοποίηση της δομής, *CSS* για την μορφοποίηση της σελίδας και τέλος *JavaScript* για τις λειτουργικές ανάγκες της σελίδας. Επίσης χρησιμοποιούνται βιβλιοθήκες και *frameworks* για να ενισχύουν την εμπειρία του χρήστη καθώς και την ευκολία ανάπτυξης της σελίδας από τους προγραμματιστές. Τέλος, το περιβάλλον όπου έγινε η συγγραφή του κώδικα είναι το *Visual Studio*.

## 1.2 Σκοπός της εργασίας

Στόχος της παρούσας εργασίας είναι να προτείνει μια συστηματική προσέγγιση για την αποτελεσματική διαχείριση και διενέργεια δημοπρασιών σε ένα ηλεκτρονικό περιβάλλον. Συγκεκριμένα, η εργασία αποσκοπεί στην δημιουργία ενός ευέλικτου συστήματος που θα επιτρέπει τον εύκολο χειρισμό τόσο από την πλευρά του προσωπικού του καταστήματος όσο και από την πλευρά του πελάτη. Η αποτελεσματική διαχείριση και η ευχρηστία της εφαρμογής οδηγούν σε ταχύτερη και πιο αποτελεσματική λήψη αποφάσεων, ενισχύοντας έτσι την εμπειρία των χρηστών.

Κατά τη διαδικασία υλοποίησης της εφαρμογής, έχει δοθεί ιδιαίτερη έμφαση στην εκμάθηση και εξοικείωση με τις γλώσσες προγραμματισμού που απαιτούνται για τη δημιουργία ολοκληρωμένων συστημάτων που προορίζονται για χρήση μέσα από έναν ιστότοπο. Παράλληλα, η χρήση εργαλείων ανοικτού κώδικα συνδυάζει την αποτελεσματική διαχείριση με μικρό κόστος χρησιμοποίησης.

## 1.3 Οργάνωση της εργασίας

Στο δεύτερο κεφάλαιο θα γίνει μια εισαγωγή και σε άλλες πλατφόρμες αντίστοιχων προγραμμάτων δημοπρασιών.

Στο τρίτο κεφάλαιο θα γίνει η θεωρητική περιγραφή των τεχνολογιών και των γλωσσών προγραμματισμού που χρησιμοποιήθηκαν για την επιτυχή περάτωση του συστήματος.

Στο τέταρτο κεφάλαιο θα γίνει εισαγωγή στα τεχνικά δομικά στοιχεία με τα οποία αναπτύχθηκε η εφαρμογή καθώς και εμβάθυνση στον κώδικα.

Τέλος στο πέμπτο κεφάλαιο θα γίνει παράθεση από στιγμιότυπα οθόνης από τη χρήση της εφαρμογής, τα οποία θα μπορούσαν να θεωρηθούν ως ένα σύντομο εγχειρίδιο οδηγιών.

## 1.4 Επίλογος

Στο πρώτο κεφάλαιο της παρούσας πτυχιακής εργασίας, πραγματοποιήσαμε μια εισαγωγή στο θέμα της διαδικασίας δημοπρασιών και πωλήσεων προϊόντων μέσω ηλεκτρονικού καταστήματος. Αναλύσαμε το αντικείμενο της εργασίας μας, εστιάζοντας στη μελέτη των τεχνολογιών Ιστού και την υλοποίηση ενός συστήματος διαδικτυακών δημοπρασιών. Στη συνέχεια, περιγράψαμε τον σκοπό της εργασίας, ο οποίος στοχεύει στην πρόταση αποτελεσματικής διαχείρισης και διενέργειας των δημοπρασιών, καθώς και στη διευκόλυνση του χειρισμού από το προσωπικό του καταστήματος και τους πελάτες. Επίσης, παρουσιάσαμε την οργάνωση της εργασίας, προετοιμάζοντας το έδαφος για την ανάπτυξη των επόμενων κεφαλαίων.

Συνοψίζοντας, η πρώτη ενότητα της εργασίας μας παρέθεσε το θεματικό πλαίσιο και τους στόχους που θέλουμε να επιτύχουμε. Με την αποκτηθείσα γνώση και τις αναλύσεις που πραγματοποιήθηκαν, θα προχωρήσουμε στα επόμενα στάδια της εργασίας.

# Κεφάλαιο 2ο: Επισκόπηση Αντίστοιχων Προγραμμάτων Δημοπρασιών

## 2.1 Εισαγωγή

Στην εποχή της ψηφιακής επανάστασης, οι διαδικτυακές δημοπρασίες έχουν αναδειχθεί ως ένας από τους πιο δημοφιλείς τρόπους αγοραπωλησίας προϊόντων και υπηρεσιών. Οι πλατφόρμες δημοπρασιών προσφέρουν ευκαιρίες τόσο στους πωλητές να προσεγγίσουν ένα ευρύτερο κοινό όσο και στους αγοραστές να βρουν προϊόντα σε ανταγωνιστικές τιμές. Η παρούσα ενότητα στοχεύει να παρουσιάσει και να αναλύσει διάφορα προγράμματα δημοπρασιών που είναι ήδη διαθέσιμα στην αγορά. Μέσα από αυτή την ανάλυση, θα κατανοήσουμε τις δυνατότητες και τις αδυναμίες κάθε πλατφόρμας, προσφέροντας έτσι μια πλήρη εικόνα του τοπίου των διαδικτυακών δημοπρασιών.

Αρχικά, θα εξετάσουμε τα πιο δημοφιλή προγράμματα δημοπρασιών, όπως το eBay, το Catawiki και το AuctionZip. Το eBay, για παράδειγμα, είναι μία από τις παλαιότερες και μεγαλύτερες πλατφόρμες δημοπρασιών παγκοσμίως. Προσφέρει μια τεράστια ποικιλία προϊόντων, από καθημερινά αντικείμενα μέχρι σπάνια συλλεκτικά. Οι χρήστες μπορούν να δημοπρατήσουν τα προϊόντα τους, να υποβάλουν προσφορές και να χρησιμοποιήσουν ένα αυτοματοποιημένο σύστημα προσφορών για να μεγιστοποιήσουν τις πιθανότητες επιτυχίας τους. Παρά την επιτυχία του, το eBay αντιμετωπίζει προκλήσεις όπως η πολυπλοκότητα της χρήσης για νέους χρήστες και τα υψηλά τέλη συναλλαγών.

Στη συνέχεια, θα αναλύσουμε το Catawiki, μια πλατφόρμα που εστιάζει σε εξειδικευμένα αντικείμενα όπως έργα τέχνης, πολυτελή αγαθά και συλλεκτικά. Η μοναδικότητα του Catawiki έγκειται στη διαδικασία έγκρισης από ειδικούς επιμελητές που διασφαλίζουν την αυθεντικότητα και την ποιότητα των αντικειμένων. Αυτή η προσέγγιση ενισχύει την εμπιστοσύνη των χρηστών, αλλά παράλληλα περιορίζει τον αριθμό των αντικειμένων που μπορούν να δημοπρατηθούν, καθώς απαιτείται έγκριση από τους επιμελητές.

Τέλος, θα εξετάσουμε το AuctionZip, μια πλατφόρμα που εξυπηρετεί κυρίως δημοπράτες και σπίτια δημοπρασιών. Το AuctionZip προσφέρει υπηρεσίες ζωντανής μετάδοσης δημοπρασιών, επιτρέποντας στους χρήστες να συμμετέχουν σε πραγματικό χρόνο. Παρά τις εξειδικευμένες

δυνατότητές του, το AuctionZip μπορεί να μην είναι τόσο φιλικό προς τους απλούς χρήστες που δεν είναι εξοικειωμένοι με τον κόσμο των δημοπρασιών.

## 2.2 Δημοφιλή Προγράμματα Δημοπρασιών

Εδώ θα παρουσιαστούν συγκεκριμένα προγράμματα δημοπρασιών. Για κάθε πρόγραμμα, θα παρέχουμε μια περιγραφή των βασικών χαρακτηριστικών και λειτουργιών του, καθώς και πλεονεκτήματα και μειονεκτήματα.

### 2.2.1 eBay

Το eBay είναι μία από τις μεγαλύτερες και πιο διαδεδομένες πλατφόρμες δημοπρασιών παγκοσμίως, επιτρέποντας στους χρήστες να δημοπρατούν και να αγοράζουν προϊόντα από όλο τον κόσμο. Ιδρύθηκε το 1995 και έκτοτε έχει εξελιχθεί σε έναν γίγαντα του ηλεκτρονικού εμπορίου, εξυπηρετώντας εκατομμύρια χρήστες σε διάφορες χώρες. Η πλατφόρμα προσφέρει ένα ευρύ φάσμα προϊόντων, από καθημερινά αντικείμενα μέχρι σπάνια συλλεκτικά κομμάτια, καθιστώντας την ιδανική για μια μεγάλη ποικιλία αγοραστών και πωλητών.

#### Τα βασικά χαρακτηριστικά του είναι:

**Εγγραφή και προφίλ χρήστη:** Το eBay επιτρέπει στους χρήστες να δημιουργούν προσωπικούς λογαριασμούς, οι οποίοι μπορούν να περιλαμβάνουν προφίλ χρήστη με προσωπικές πληροφορίες, ιστορικό αγορών και πωλήσεων, καθώς και ρυθμίσεις ασφαλείας. Οι χρήστες μπορούν επίσης να παρακολουθούν δημοπρασίες που τους ενδιαφέρουν και να λαμβάνουν ειδοποιήσεις για νέες προσφορές και λήξεις δημοπρασιών.

**Δημιουργία και διαχείριση δημοπρασιών:** Οι χρήστες μπορούν να δημιουργούν δημοπρασίες για διάφορα προϊόντα, ανεβάζοντας φωτογραφίες, παρέχοντας λεπτομερείς περιγραφές και καθορίζοντας τιμές εκκίνησης. Η πλατφόρμα προσφέρει εργαλεία για την παρακολούθηση της πορείας των δημοπρασιών, την επεξεργασία των στοιχείων τους και τη διαχείριση των πωλήσεων.

**Σύστημα υποβολής προσφορών:** Το eBay διαθέτει ένα δυναμικό σύστημα υποβολής προσφορών, επιτρέποντας στους χρήστες να υποβάλουν προσφορές σε πραγματικό χρόνο. Επιπλέον, το auto-bidding επιτρέπει στους χρήστες να καθορίσουν τη μέγιστη προσφορά τους, με το σύστημα να υποβάλλει αυτόματα προσφορές εκ μέρους τους μέχρι να φτάσει το όριο.

**Σύστημα πληρωμών:** Το eBay υποστηρίζει πολλές μεθόδους πληρωμής, με πιο διαδεδομένη το PayPal, που προσφέρει ασφάλεια και ευκολία στις συναλλαγές. Επίσης, η πλατφόρμα υποστηρίζει πληρωμές μέσω πιστωτικών και χρεωστικών καρτών, καθώς και άλλες τοπικές μεθόδους πληρωμής ανάλογα με τη χώρα του χρήστη.

**Αξιολογήσεις και σχόλια:** Το σύστημα αξιολογήσεων και σχολίων του eBay επιτρέπει στους χρήστες να αξιολογούν και να σχολιάζουν τις συναλλαγές τους, προωθώντας τη διαφάνεια και την αξιοπιστία. Οι αξιολογήσεις αυτές βοηθούν τους χρήστες να επιλέξουν αξιόπιστους πωλητές και αγοραστές, δημιουργώντας ένα ασφαλές περιβάλλον συναλλαγών.

#### Πλεονεκτήματα:

- **Παγκόσμια εμβέλεια:** Το eBay παρέχει πρόσβαση σε ένα τεράστιο κοινό, επιτρέποντας στους χρήστες να αγοράζουν και να πωλούν προϊόντα διεθνώς.

- **Ποικιλία προϊόντων:** Από καθημερινά είδη μέχρι σπάνια συλλεκτικά, η πλατφόρμα καλύπτει ένα ευρύ φάσμα κατηγοριών προϊόντων.
- **Αξιοπιστία και ασφάλεια:** Το σύστημα αξιολογήσεων και η υποστήριξη μεθόδων πληρωμής όπως το PayPal εξασφαλίζουν ασφαλείς συναλλαγές.
- **Auto-bidding:** Το σύστημα αυτόματων προσφορών διευκολύνει τους χρήστες να παραμένουν ανταγωνιστικοί χωρίς να παρακολουθούν συνεχώς τις δημοπρασίες.

### **Μειονεκτήματα:**

- **Πολυπλοκότητα χρήσης:** Η πλατφόρμα μπορεί να είναι δύσκολη για νέους χρήστες, με πολλές λειτουργίες και ρυθμίσεις που απαιτούν χρόνο για να γίνουν κατανοητές.
- **Υψηλά τέλη συναλλαγών:** Τα τέλη που χρεώνει το eBay μπορεί να είναι υψηλά, ιδιαίτερα για πωλητές με μεγάλο όγκο πωλήσεων, μειώνοντας τα καθαρά κέρδη.
- **Ανταγωνισμός:** Λόγω της μεγάλης βάσης χρηστών, ο ανταγωνισμός στις δημοπρασίες μπορεί να είναι έντονος, καθιστώντας δύσκολη την εξασφάλιση προϊόντων σε χαμηλές τιμές.

Συνολικά, το eBay παραμένει μία από τις κορυφαίες πλατφόρμες δημοπρασιών παγκοσμίως, προσφέροντας πλούσια χαρακτηριστικά και ευκαιρίες για αγοραστές και πωλητές, παρά τις προκλήσεις που μπορεί να αντιμετωπίζουν.

### **2.2.2 Catawiki**

Το Catawiki είναι μια διαδικτυακή πλατφόρμα δημοπρασιών που εστιάζει σε εξειδικευμένα αντικείμενα, όπως συλλεκτικά, έργα τέχνης και πολυτελή αγαθά. Ιδρύθηκε το 2008 και έκτοτε έχει αναπτυχθεί σημαντικά, προσελκύοντας πωλητές και αγοραστές που αναζητούν αντικείμενα υψηλής αξίας και μοναδικότητας. Σε αντίθεση με πιο γενικές πλατφόρμες δημοπρασιών, το Catawiki επικεντρώνεται σε συγκεκριμένες κατηγορίες αντικειμένων, προσφέροντας μια πιο εξειδικευμένη εμπειρία τόσο για πωλητές όσο και για αγοραστές.

#### **Τα βασικά χαρακτηριστικά του είναι:**

**Επιμελητές:** Ένα από τα κύρια χαρακτηριστικά του Catawiki είναι η παρουσία ειδικών επιμελητών. Αυτοί οι επιμελητές είναι ειδικοί στους τομείς τους και επιβλέπουν κάθε δημοπρασία, αξιολογώντας τα αντικείμενα που καταχωρούνται. Αυτό το χαρακτηριστικό ενισχύει την εμπιστοσύνη των χρηστών, καθώς διασφαλίζεται ότι τα αντικείμενα είναι αυθεντικά και αξιόλογα. Οι επιμελητές παρέχουν επίσης λεπτομερείς περιγραφές και επαγγελματικές εκτιμήσεις, δίνοντας στους αγοραστές περισσότερες πληροφορίες για τα αντικείμενα.

**Δημιουργία και διαχείριση δημοπρασιών:** Οι πωλητές στο Catawiki μπορούν να ανεβάσουν αντικείμενα για δημοπρασία, παρέχοντας λεπτομερείς περιγραφές και φωτογραφίες. Η πλατφόρμα προσφέρει εργαλεία για την εύκολη διαχείριση των δημοπρασιών, όπως παρακολούθηση προσφορών, επεξεργασία καταχωρίσεων και επικοινωνία με πιθανούς αγοραστές. Οι δημοπρασίες είναι χρονικά περιορισμένες, προσφέροντας στους αγοραστές ένα συγκεκριμένο χρονικό πλαίσιο για να υποβάλουν τις προσφορές τους.

**Διαδικασία έγκρισης:** Πριν τα αντικείμενα καταχωρηθούν στην πλατφόρμα, πρέπει να εγκριθούν από τους επιμελητές. Αυτή η διαδικασία διασφαλίζει την ποιότητα και την αυθεντικότητα των αντικειμένων, μειώνοντας τον κίνδυνο απάτης και προσφέροντας στους αγοραστές την βεβαιότητα ότι τα αντικείμενα που αγοράζουν είναι αξιόπιστα. Ωστόσο, αυτό μπορεί να προκαλέσει καθυστερήσεις στην καταχώριση των αντικειμένων, καθώς η διαδικασία έγκρισης μπορεί να πάρει χρόνο.

**Σύστημα πληρωμών και αποστολών:** Το Catawiki διαθέτει ένα ολοκληρωμένο σύστημα για πληρωμές και ασφαλείς αποστολές. Οι πληρωμές μπορούν να γίνουν μέσω διάφορων μεθόδων, όπως πιστωτικές κάρτες και τραπεζικές μεταφορές, εξασφαλίζοντας την ασφάλεια των συναλλαγών. Η πλατφόρμα συνεργάζεται με αξιόπιστους μεταφορείς για την αποστολή των αντικειμένων, παρέχοντας ασφάλεια και παρακολούθηση των αποστολών.

#### **Πλεονεκτήματα:**

- **Εξειδίκευση:** Η εστίαση σε συλλεκτικά και πολυτελή αντικείμενα προσελκύει αγοραστές και πωλητές που ενδιαφέρονται για εξειδικευμένες αγορές.
- **Εμπιστοσύνη:** Η διαδικασία επιμέλειας από ειδικούς επιμελητές ενισχύει την αξιοπιστία της πλατφόρμας, διασφαλίζοντας την αυθεντικότητα και την ποιότητα των αντικειμένων.
- **Επαγγελματικές εκτιμήσεις:** Οι επιμελητές παρέχουν λεπτομερείς περιγραφές και επαγγελματικές εκτιμήσεις, βοηθώντας τους αγοραστές να λάβουν ενημερωμένες αποφάσεις.
- **Ασφαλείς συναλλαγές:** Το ολοκληρωμένο σύστημα πληρωμών και αποστολών εξασφαλίζει την ασφάλεια και την αξιοπιστία των συναλλαγών.

#### **Μειονεκτήματα:**

- **Περιορισμοί στα αντικείμενα:** Το Catawiki περιορίζεται σε συγκεκριμένες κατηγορίες αντικειμένων, κάτι που μπορεί να αποθαρρύνει πωλητές και αγοραστές που ενδιαφέρονται για πιο γενικά προϊόντα.
- **Χρόνος έγκρισης:** Η διαδικασία έγκρισης από τους επιμελητές μπορεί να πάρει χρόνο, καθυστερώντας την καταχώριση των αντικειμένων για δημοπρασία.
- **Κόστος συμμετοχής:** Αν και η ποιότητα των υπηρεσιών είναι υψηλή, οι πωλητές μπορεί να αντιμετωπίσουν υψηλότερα κόστη λόγω των τελών και των προμηθειών που χρεώνει η πλατφόρμα.

Συνολικά, το Catawiki προσφέρει μια εξειδικευμένη πλατφόρμα δημοπρασιών με έμφαση στην ποιότητα και την αξιοπιστία των αντικειμένων. Παρά τις προκλήσεις που σχετίζονται με την περιορισμένη γκάμα αντικειμένων και τον χρόνο έγκρισης, η πλατφόρμα προσελκύει χρήστες που

αναζητούν μοναδικά και αξιόλογα αντικείμενα, προσφέροντας μια ασφαλή και αξιόπιστη εμπειρία δημοπρασίας.

### 2.2.3 AuctionZip

Το AuctionZip είναι μια διαδικτυακή πλατφόρμα που παρέχει υπηρεσίες δημοπρασιών κυρίως για δημοπράτες και σπίτια δημοπρασιών. Ιδρύθηκε το 2003 και έχει καθιερωθεί ως ένας από τους κορυφαίους ιστότοπους για ζωντανές δημοπρασίες στις Ηνωμένες Πολιτείες και διεθνώς. Η πλατφόρμα αυτή επικεντρώνεται στη σύνδεση δημοπρατών με ένα ευρύ κοινό, προσφέροντας εργαλεία και υπηρεσίες που διευκολύνουν τη διαχείριση και την παρακολούθηση των δημοπρασιών.

#### **Τα βασικά χαρακτηριστικά του είναι:**

**Καταλόγους δημοπρασιών:** Οι χρήστες του AuctionZip μπορούν να αναζητήσουν και να παρακολουθήσουν ζωντανές δημοπρασίες μέσω της πλατφόρμας. Οι κατάλογοι δημοπρασιών περιλαμβάνουν λεπτομερείς περιγραφές των αντικειμένων, φωτογραφίες, ημερομηνίες και ώρες των δημοπρασιών, καθώς και πληροφορίες για τους δημοπράτες. Αυτό επιτρέπει στους χρήστες να σχεδιάζουν τις συμμετοχές τους και να ενημερώνονται για τις επερχόμενες δημοπρασίες.

**Υπηρεσίες ζωντανής μετάδοσης:** Μία από τις πιο σημαντικές λειτουργίες του AuctionZip είναι η δυνατότητα ζωντανής μετάδοσης των δημοπρασιών μέσω διαδικτύου. Αυτό επιτρέπει στους χρήστες να παρακολουθούν και να συμμετέχουν σε δημοπρασίες σε πραγματικό χρόνο από οπουδήποτε στον κόσμο. Οι ζωντανές μεταδόσεις βελτιώνουν την προσβασιμότητα των δημοπρασιών και διευρύνουν το κοινό των δημοπρατών.

**Ειδοποιήσεις:** Οι χρήστες μπορούν να ρυθμίσουν ειδοποιήσεις για συγκεκριμένες δημοπρασίες ή είδη. Αυτό το χαρακτηριστικό επιτρέπει στους χρήστες να παραμένουν ενημερωμένοι για δημοπρασίες που τους ενδιαφέρουν, χωρίς να χρειάζεται να ελέγχουν συνεχώς την πλατφόρμα. Οι ειδοποιήσεις μπορούν να αποστέλλονται μέσω email ή άλλων μέσων, διευκολύνοντας τη συμμετοχή των χρηστών στις δημοπρασίες.

**Πληροφορίες και υποστήριξη:** Το AuctionZip προσφέρει εκτεταμένες πληροφορίες και υποστήριξη για τις δημοπρασίες και τους πωλητές. Η πλατφόρμα παρέχει οδηγίες για τη συμμετοχή στις δημοπρασίες, πληροφορίες για τους κανόνες και τις διαδικασίες, καθώς και υποστήριξη για τεχνικά ζητήματα. Αυτό βοηθά τους χρήστες να κατανοήσουν καλύτερα τη διαδικασία των δημοπρασιών και να συμμετέχουν με ασφάλεια.

#### **Πλεονεκτήματα:**

- **Εξειδίκευση:** Το AuctionZip επικεντρώνεται σε ζωντανές δημοπρασίες, παρέχοντας εξειδικευμένες υπηρεσίες για δημοπράτες και σπίτια δημοπρασιών.
- **Ζωντανή μετάδοση:** Η δυνατότητα ζωντανής μετάδοσης επιτρέπει στους χρήστες να συμμετέχουν σε δημοπρασίες σε πραγματικό χρόνο από οπουδήποτε, αυξάνοντας την προσβασιμότητα και τη συμμετοχή.
- **Εκτεταμένες πληροφορίες:** Η πλατφόρμα παρέχει λεπτομερείς καταλόγους δημοπρασιών και υποστήριξη, διευκολύνοντας την κατανόηση και τη συμμετοχή στις δημοπρασίες.

- **Ειδοποιήσεις:** Οι ειδοποιήσεις κρατούν τους χρήστες ενημερωμένους για τις δημοπρασίες που τους ενδιαφέρουν, χωρίς να χρειάζεται συνεχής παρακολούθηση.

#### Μειονεκτήματα:

- **Φιλικότητα προς το χρήστη:** Η πλατφόρμα μπορεί να μην είναι τόσο φιλική προς το χρήστη για άτομα που δεν είναι επαγγελματίες στον τομέα των δημοπρασιών. Οι νέοι χρήστες μπορεί να βρουν το περιβάλλον και τις διαδικασίες πολύπλοκες.
- **Περιορισμένες δυνατότητες για μη επαγγελματίες:** Οι χρήστες που δεν ενδιαφέρονται για ζωντανές δημοπρασίες μπορεί να βρουν την πλατφόρμα λιγότερο ελκυστική, καθώς η κύρια εστίαση του AuctionZip είναι στις ζωντανές δημοπρασίες.
- **Ανάγκη για καλή σύνδεση στο διαδίκτυο:** Η συμμετοχή σε ζωντανές δημοπρασίες απαιτεί σταθερή και γρήγορη σύνδεση στο διαδίκτυο, κάτι που μπορεί να είναι περιοριστικό για χρήστες με αργές συνδέσεις.

Συνολικά, το AuctionZip είναι μια εξειδικευμένη πλατφόρμα δημοπρασιών που προσφέρει σημαντικά πλεονεκτήματα για δημοπράτες και σπίτια δημοπρασιών, με έμφαση στη ζωντανή μετάδοση και την εκτενή υποστήριξη. Ωστόσο, η πλατφόρμα μπορεί να είναι πιο απαιτητική για νέους χρήστες και λιγότερο ελκυστική για αγοραστές που δεν ενδιαφέρονται για ζωντανές δημοπρασίες.

### 2.3 Συγκριτική ανάλυση

Σε αυτή την ενότητα, θα γίνει μια συγκριτική ανάλυση των προαναφερθέντων προγραμμάτων δημοπρασιών (eBay, Catawiki, AuctionZip) με βάση διάφορα κριτήρια, όπως η ευκολία χρήσης, τα χαρακτηριστικά, η υποστήριξη και ασφάλεια, καθώς και τα τέλη και το κόστος.

#### Ευκολία χρήσης:

Η ευκολία χρήσης είναι κρίσιμος παράγοντας για την επιτυχία μιας πλατφόρμας δημοπρασιών. Το eBay είναι γνωστό για τη φιλικότητα προς το χρήστη, με ένα εύχρηστο περιβάλλον που απευθύνεται τόσο σε αρχάριους όσο και σε έμπειρους χρήστες. Η διαδικασία εγγραφής, καταχώρισης προϊόντων και υποβολής προσφορών είναι απλή και κατανοητή. Αντίθετα, το Catawiki και το AuctionZip μπορεί να απαιτούν περισσότερο χρόνο για να τα μάθει κανείς, ειδικά το AuctionZip που απευθύνεται κυρίως σε επαγγελματίες δημοπράτες. Το Catawiki, με την παρουσία ειδικών επιμελητών, προσφέρει μια πιο δομημένη και επαγγελματική προσέγγιση, η οποία μπορεί να είναι λιγότερο φιλική προς τους νέους χρήστες αλλά εξαιρετικά ωφέλιμη για τους συλλέκτες.

**Χαρακτηριστικά:** Όσον αφορά τα χαρακτηριστικά, το eBay προσφέρει μια ευρεία γκάμα δυνατοτήτων, όπως αυτόματες προσφορές, αξιολογήσεις και σχόλια χρηστών, και πολυάριθμες μεθόδους πληρωμής. Το Catawiki ξεχωρίζει για την επιμέλεια των αντικειμένων από ειδικούς, προσφέροντας έτσι μια μοναδική εμπειρία αγοράς και πώλησης για συλλεκτικά και πολυτελή αντικείμενα. Το AuctionZip προσφέρει την εξειδικευμένη δυνατότητα ζωντανής μετάδοσης δημοπρασιών, κάτι που δεν προσφέρεται από τα άλλα δύο προγράμματα. Ωστόσο, τα χαρακτηριστικά του AuctionZip είναι πιο προσανατολισμένα στους δημοπράτες και στα σπίτια δημοπρασιών, παρά στους μεμονωμένους αγοραστές.

**Υποστήριξη και ασφάλεια:** Στην κατηγορία της υποστήριξης και ασφάλειας, το Catawiki υπερέχει χάρη στην παρουσία των επιμελητών, που διασφαλίζουν την ποιότητα και την αυθεντικότητα των αντικειμένων, καθώς και την υποστήριξη που παρέχουν στους χρήστες. Το eBay

επίσης προσφέρει αξιόπιστη υποστήριξη και μέτρα ασφαλείας, όπως το σύστημα αξιολογήσεων και το PayPal για ασφαλείς πληρωμές. Το AuctionZip παρέχει πληροφορίες και υποστήριξη κυρίως για τους δημοπράτες, με εκτενείς οδηγίες και ζωντανή υποστήριξη κατά τη διάρκεια των δημοπρασιών, διασφαλίζοντας μια ομαλή και ασφαλή διαδικασία.

**Τέλη και κόστος:** Όσον αφορά τα τέλη και το κόστος, το eBay συχνά επικρίνεται για τα υψηλά τέλη συναλλαγών και καταχώρισης, γεγονός που μπορεί να είναι αποτρεπτικό για ορισμένους χρήστες. Το Catawiki χρεώνει επίσης τέλη για τις υπηρεσίες επιμέλειας και τις ασφαλείς πληρωμές, αλλά τα υψηλότερα τέλη του αντικατοπτρίζουν την ποιότητα και την ασφάλεια που προσφέρει η πλατφόρμα. Το AuctionZip μπορεί να έχει χαμηλότερα τέλη για καταχωρίσεις, αλλά οι δημοπράτες πρέπει να καλύψουν τα κόστη ζωντανής μετάδοσης και άλλες υπηρεσίες, κάνοντάς το λιγότερο προσιτό για μικρούς πωλητές.

**Συμπεράσματα:** Σε γενικές γραμμές, το eBay ξεχωρίζει για την ευκολία χρήσης και την ευρεία γκάμα χαρακτηριστικών του, κάνοντάς το ιδανικό για καθημερινούς χρήστες που επιθυμούν να πουλήσουν ή να αγοράσουν προϊόντα. Το Catawiki προσφέρει εξαιρετική υποστήριξη και ασφάλεια μέσω των ειδικών επιμελητών, καθιστώντας το ιδανικό για συλλέκτες και αγοραστές υψηλής αξίας αντικειμένων, παρά το υψηλότερο κόστος του. Το AuctionZip είναι η καλύτερη επιλογή για επαγγελματίες δημοπράτες και σπίτια δημοπρασιών που χρειάζονται ζωντανή μετάδοση και εξειδικευμένες υπηρεσίες, αν και η πολυπλοκότητά του μπορεί να είναι αποτρεπτική για απλούς χρήστες.

Αυτή η συγκριτική ανάλυση υπογραμμίζει ότι η επιλογή της κατάλληλης πλατφόρμας εξαρτάται από τις συγκεκριμένες ανάγκες και τις προτεραιότητες των χρηστών, είτε πρόκειται για ευκολία χρήσης, πλούτο χαρακτηριστικών, επίπεδο υποστήριξης και ασφάλειας, είτε για τα συνολικά κόστη.

## 2.4 Επίλογος

Συνοψίζοντας, η συγκριτική ανάλυση των προγραμμάτων δημοπρασιών eBay, Catawiki και AuctionZip αποδεικνύει ότι κάθε πλατφόρμα διαθέτει μοναδικά πλεονεκτήματα και αδυναμίες που την καθιστούν κατάλληλη για διαφορετικούς τύπους χρηστών. Το eBay προσφέρει μια ευρεία και φιλική προς το χρήστη εμπειρία με εκτεταμένες δυνατότητες, καθιστώντας το ιδανικό για καθημερινούς χρήστες που επιθυμούν να αγοράσουν ή να πουλήσουν προϊόντα χωρίς ιδιαίτερη δυσκολία. Το Catawiki, με την έμφαση στην ποιότητα και την ασφάλεια μέσω των ειδικών επιμελητών, προσελκύει κυρίως συλλέκτες και αγοραστές υψηλής αξίας αντικειμένων, παρά το υψηλότερο κόστος του. Το AuctionZip, με την εξειδίκευσή του στις ζωντανές δημοπρασίες και την υποστήριξη επαγγελματιών δημοπρατών, είναι η καλύτερη επιλογή για δημοπράτες και σπίτια δημοπρασιών που απαιτούν προηγμένες υπηρεσίες ζωντανής μετάδοσης.

Η επιλογή της κατάλληλης πλατφόρμας δημοπρασιών εξαρτάται από τις συγκεκριμένες ανάγκες του χρήστη. Για τον καθημερινό χρήστη που αναζητά ευκολία και ευελιξία, το eBay είναι πιθανότατα η καλύτερη επιλογή. Για τους συλλέκτες και αγοραστές που δίνουν προτεραιότητα στην ποιότητα και την ασφάλεια, το Catawiki προσφέρει μια ανώτερη εμπειρία. Τέλος, για τους επαγγελματίες δημοπράτες, το AuctionZip παρέχει τα απαραίτητα εργαλεία και την υποστήριξη για τη διεξαγωγή επιτυχημένων ζωντανών δημοπρασιών. Αυτή η ανάλυση υπογραμμίζει την ποικιλία και τον πλούτο των επιλογών που είναι διαθέσιμες στην αγορά δημοπρασιών, επιτρέποντας στους χρήστες να επιλέξουν την πλατφόρμα που ταιριάζει καλύτερα στις ανάγκες και τους στόχους τους.





## Κεφάλαιο 3ο: Εργαλεία και Γλώσσες Προγραμματισμού

### 3.1 Εισαγωγή

Στο δεύτερο κεφάλαιο της παρούσας πτυχιακής εργασίας, θα εξετάσουμε διάφορα εργαλεία και γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για την υλοποίηση του διαδικτυακού καταστήματος. Κατά τη διάρκεια αυτής της ενότητας, θα εξετάσουμε τις λεπτομέρειες της χρήσης και της εφαρμογής των πιο κοινών εργαλείων και γλωσσών προγραμματισμού στον κόσμο της ανάπτυξης εφαρμογών ιστού.

Αρχικά, θα εξετάσουμε τη χρήση των βασικών γλωσσών σήμανσης *HTML* και *CSS*, καθώς και τον τρόπο με τον οποίο μπορούν να ενισχυθούν με τη χρήση της βιβλιοθήκης *Bootstrap*. Στη συνέχεια, θα εξετάσουμε τη σημασία και τη χρήση της γλώσσας προγραμματισμού *JavaScript*, συμπεριλαμβανομένων των τεχνικών *AJAX* και *JSON* για ασύγχρονη επικοινωνία με τον διακομιστή.

Στη συνέχεια, θα εστιάσουμε στην πλευρά του διακομιστή, εξετάζοντας τη χρήση της γλώσσας προγραμματισμού *PHP* και του πλαισίου εργασίας *Laravel* για την ανάπτυξη περισσότερο πολύπλοκων εφαρμογών. Θα αναλύσουμε επίσης τη χρήση βάσεων δεδομένων, συμπεριλαμβανομένης της γλώσσας *SQL* και του συστήματος διαχείρισης βάσεων δεδομένων *MariaDB*.

Τέλος, θα δούμε πώς ολοκληρώνονται όλα αυτά τα εργαλεία και οι γλώσσες προγραμματισμού για την ανάπτυξη μιας συνολικής εφαρμογής ηλεκτρονικού καταστήματος, προσφέροντας ένα ολοκληρωμένο πλαίσιο για την κατανόηση της τεχνολογικής πλευράς του έργου μας.

### 3.2 HTML/CSS

Η *HTML* (*HyperText Markup Language*) αποτελεί το θεμέλιο του Παγκόσμιου Ιστού (*World Wide Web*), επιτρέποντας τη δημιουργία και τη μορφοποίηση ιστοσελίδων. Από τη συγκρότησή της τη δεκαετία του 1990 μέχρι τις σύγχρονες εφαρμογές του διαδικτύου, η *HTML* έχει προχωρήσει σε μια σειρά εξελίξεων που έχουν διαμορφώσει το τοπίο του ιστού.

Η *HTML* δημιουργήθηκε από τους Τιμ Μπέρνερς-Λι, Ρομπέρτ Καϊλιου και άλλους στο *CERN* (*European Organization for Nuclear Research*) το 1989. Αρχικά, ήταν απλή και περιορισμένη στις δυνατότητες της, αλλά συνεχώς εξελίσσεται για να αντικατοπτρίζει τις ανάγκες των χρηστών και τις τεχνολογικές εξελίξεις.

#### Βασικές Έννοιες της HTML:

- **Ετικέτες (Tags):** Οι ετικέτες δηλώνουν την αρχή και το τέλος ενός στοιχείου και περιγράφουν πώς πρέπει να εμφανιστεί το περιεχόμενο. Για παράδειγμα, η ετικέτα `<p>` δηλώνει ένα παράγραφο. Μερικά βασικά *HTML* Tags παρουσιάζονται στην εικόνα.

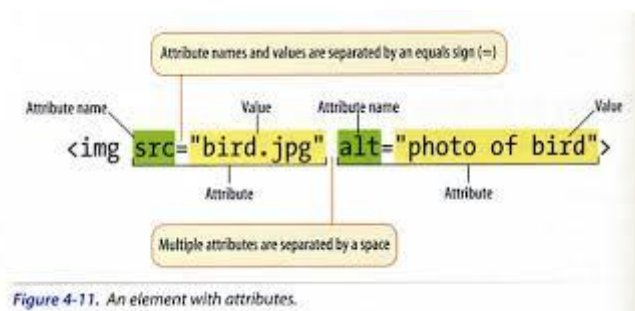
### List of HTML Tags

• <!-->	• <cite>	• <frameset>	• <object>	• <style>
• <!DOCTYPE>	• <code>	• <h1> to <h6>	• <ol>	• <sub>
• <a>	• <col>	• <head>	• <optgroup>	• <summary>
• <abbr>	• <colgroup>	• <header>	• <option>	• <sup>
• <acronym>	• <data>	• <hr>	• <output>	• <svg>
• <address>	• <datalist>	• <html>	• <p>	• <tbody>
• <applet>	• <dd>	• <i>	• <param>	• <td>
• <area>	• <del>	• <iframe>	• <picture>	• <thead>
• <article>	• <details>	• <img>	• <pre>	• <textarea>
• <aside>	• <dfn>	• <input>	• <progress>	• <tfoot>
• <audio>	• <dialog>	• <ins>	• <q>	• <th>
• <b>	• <dir>	• <kbd>	• <rp>	• <thead>
• <base>	• <div>	• <label>	• <rt>	• <tbody>
• <basefont>	• <dl>	• <legend>	• <ruby>	• <time>
• <bdi>	• <dt>	• <li>	• <s>	• <title>
• <bdo>	• <em>	• <link>	• <samp>	• <tr>
• <big>	• <embed>	• <main>	• <script>	• <track>
• <blockquote>	• <fieldset>	• <map>	• <section>	• <tt>
• <body>	• <figcaption>	• <mark>	• <select>	• <u>
•  	• <figure>	• <meta>	• <small>	• <ul>
• <button>	• <font>	• <meter>	• <source>	• <var>
• <canvas>	• <footer>	• <nav>	• <span>	• <video>
• <caption>	• <form>	• <noframes>	• <strike>	• <wbr>
• <center>	• <frame>	• <noscript>	• <strong>	

● Not supported in HTML5

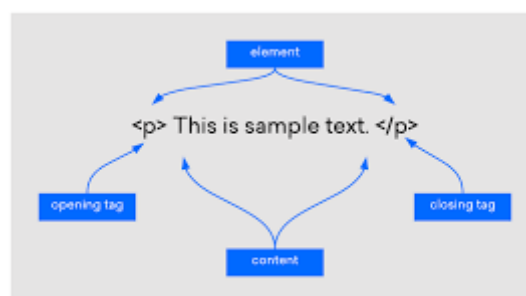
Εικόνα 2.1: HTML Tags

- **Χαρακτηριστικά (Attributes):** Τα *Attributes* παρέχουν περαιτέρω πληροφορίες για τα στοιχεία και ορίζουν τη συμπεριφορά τους. Για παράδειγμα, το "src" καθορίζει τη διεύθυνση της εικόνας που πρέπει να εμφανιστεί.



Εικόνα 2.2: HTML Attributes

- **Στοιχεία (Elements):** Τα στοιχεία αποτελούν τα βασικά κτίρια της *HTML* και περιγράφουν τη δομή της σελίδας. Κάθε στοιχείο αρχίζει με μια ετικέτα "tag" και μπορεί να περιέχει περιεχόμενο και άλλα στοιχεία.



Εικόνα 2.3: HTML Element

Η *HTML* είναι ένα απαραίτητο εργαλείο για τη δημιουργία ιστοσελίδων και έχει διαμορφώσει τον τρόπο με τον οποίο αλληλεπιδρούμε με το διαδίκτυο. Με την κατανόηση της ιστορίας και των βασικών αρχών της *HTML* αλλά και την χρήση *CSS/JavaScript* που θα εξηγήσουμε αργότερα, μπορούμε να δημιουργήσουμε ιστοσελίδες που είναι λειτουργικές, εύχρηστες και αισθητικά ελκυστικές.

Από την άλλη πλευρά, η *CSS (Cascading Style Sheets)* αποτελεί ένα ισχυρό εργαλείο για τη μορφοποίηση ιστοσελίδων, επιτρέποντας το διαχωρισμό του περιεχομένου από την παρουσίαση του. Η ιστορία της *CSS* αποτελεί μια αφήγηση της εξέλιξης της από τις αρχές του διαδικτύου μέχρι τις σύγχρονες προσεγγίσεις στο *web design*.

Η ανάγκη για ένα εργαλείο που θα επέτρεπε τη διαχείριση του σχεδιασμού και της μορφοποίησης των ιστοσελίδων οδήγησε στη δημιουργία της *CSS*. Αρχικά, η *HTML* χρησιμοποιούνταν για τη μορφοποίηση του περιεχομένου, αλλά αυτό οδηγούσε σε μια σύγχυση και δυσκολία στη διαχείριση του κώδικα.

Το 1994, ο Χάκον Βέρουμ πρότεινε την ιδέα της *CSS* στον Τιμ Μπέρνερς-Λι, ο οποίος ήταν επίσης ο δημιουργός της *HTML*. Αυτό άνοιξε το δρόμο για την ανάπτυξη μιας ξεχωριστής γλώσσας για τη μορφοποίηση, ξεχωριστή από τη δομή της σελίδας. Η πρώτη πρόταση για την *CSS* κατατέθηκε το 1996 από τον Χάκον Βέρουμ και τον Μπερτ Μποζιερ, και το 1997 κυκλοφόρησε η πρώτη έκδοση της *CSS*.

### Βασικές Έννοιες της *CSS*:

- **Επιλογείς (Selectors):** Οι επιλογείς καθορίζουν ποια στοιχεία της *HTML* θα επηρεαστούν από το στυλ που ορίζεται στο *CSS* κανόνα. Υπάρχουν πολλών ειδών όπως για παράδειγμα με βάση το *ID*, το *ClassName*, το *TagName* και άλλα ακόμη.
- **Ιδιότητες (Properties):** Οι ιδιότητες καθορίζουν την εμφάνιση και τη συμπεριφορά των επιλεγμένων στοιχείων, όπως το χρώμα, η γραμματοσειρά, η διάταξη κλπ.
- **Τιμές (Values):** Οι τιμές ορίζουν τις παραμέτρους των ιδιοτήτων, όπως το συγκεκριμένο χρώμα ή η γραμματοσειρά που θα χρησιμοποιηθεί.

Η *CSS* έχει αναδειχθεί ως κρίσιμο εργαλείο για τη δημιουργία μοντέρνων και λειτουργικών ιστοσελίδων. Η εξέλιξή της από μια απλή ιδέα σε ένα ισχυρό εργαλείο αντικατοπτρίζει τη σημαντική εξέλιξη του διαδικτύου και τη συνεχή ανάγκη για καλύτερες δυνατότητες σχεδιασμού και παρουσίασης.

### 3.2.1 Bootstrap

Η *Bootstrap* αντιπροσωπεύει μια από τις πιο επιτυχημένες και ευρέως χρησιμοποιούμενες βιβλιοθήκες *CSS* στον κόσμο του *web design*. Με τη δυνατότητά της να παρέχει έτοιμα στυλ και στοιχεία, η *Bootstrap* έχει αναδειχθεί ως καθοριστικό εργαλείο για τη δημιουργία αποδοτικών και αισθητικά εντυπωσιακών ιστοσελίδων.

Η *Bootstrap* δημιουργήθηκε από τους Μαρκ Οτς (Mark Otto) και Τζακ Ντόρσεϊ (Jacob Thornton) στο Twitter το 2011. Αρχικά, αναπτύχθηκε ως εργαλείο εσωτερικής χρήσης για τη διευκόλυνση της ανάπτυξης ιστοσελίδων στην πλατφόρμα του *Twitter*. Ωστόσο, η δυναμική της βιβλιοθήκης γρήγορα έκανε αίσθηση στην κοινότητα του *web design*, με αποτέλεσμα να κυκλοφορήσει ως ανοικτού κώδικα έργο.

### Βασικές Έννοιες της *Bootstrap*:

- **Πλέγμα (Grid):** Η *Bootstrap* παρέχει ένα ευέλικτο και εύκολο στη χρήση πλέγμα, το οποίο επιτρέπει τη διαμόρφωση της διάταξης των στοιχείων σε μια ιστοσελίδα.

- **Στοιχεία UI (User Interface):** Η Bootstrap περιλαμβάνει έτοιμα στοιχεία UI όπως κουμπιά, φόρμες, μενού πλοήγησης κλπ., που βοηθούν στην ταχεία και εύκολη δημιουργία ιστοσελίδων.
- **Υποστήριξη Responsive Design:** Με τη Bootstrap, ο σχεδιασμός μπορεί να προσαρμοστεί αυτόματα σε διάφορες συσκευές και μεγέθη οθονών, προσφέροντας μια ομαλή εμπειρία χρήστη και ευκολία στον προγραμματιστή.
- **Προσαρμογή (Customization):** Η Bootstrap επιτρέπει την εύκολη προσαρμογή του στυλ και των χρωμάτων, επιτρέποντας στους σχεδιαστές να δημιουργήσουν μοναδικές ιστοσελίδες που ανταποκρίνονται στις ανάγκες τους.

Η Bootstrap έχει επαναπροσδιορίσει τον τρόπο με τον οποίο οι ιστοσελίδες αναπτύσσονται και σχεδιάζονται. Η ιστορία της αποτελεί μια αφήγηση της επανάστασης στον τομέα του web design.

### 3.3 JavaScript

Η JavaScript αποτελεί ένα από τα πιο δημοφιλή εργαλεία στον κόσμο του web development, επιτρέποντας τη δημιουργία δυναμικών και αλληλεπιδραστικών ιστοσελίδων. Από τη στιγμή της δημιουργίας της, η JavaScript έχει εξελιχθεί σε μια ισχυρή και ευέλικτη γλώσσα προγραμματισμού που χρησιμοποιείται σε πολλούς τομείς της τεχνολογίας.

#### Βασικές Έννοιες της JavaScript:

- **Μεταβλητές (Variables):** Οι μεταβλητές χρησιμοποιούνται για την αποθήκευση δεδομένων. Μπορούν να περιέχουν αριθμούς, κείμενο, λογικές τιμές και άλλα δεδομένα.

```
var number = 10;  
var text = "Hello, world!";  
var isTrue = true;
```

Εικόνα 2.4: JavaScript Variables

- **Τύποι Δεδομένων (Data Types):** Η JavaScript υποστηρίζει διάφορους τύπους δεδομένων, συμπεριλαμβανομένων των αριθμών, των συμβολοσειρών, των λογικών τιμών και των αντικειμένων.
- **Συναρτήσεις (Functions):** Οι συναρτήσεις είναι τμήματα κώδικα που εκτελούν συγκεκριμένες εργασίες. Μπορούν να κληθούν για να εκτελέσουν ενέργειες και να επιστρέψουν αποτελέσματα.

```
function add(a, b) {  
    return a + b;  
}
```

Εικόνα 2.4: JavaScript Function

- **Συμβάντα (Events):** Τα συμβάντα συμβαίνουν όταν ο χρήστης αλληλεπιδρά με την ιστοσελίδα. Η JavaScript μπορεί να αντιδράσει σε συμβάντα όπως κλικ πάνω σε ένα κουμπί, η υποβολή μιας φόρμας κλπ.
- **Συνθήκες και Επαναλήψεις (Conditions and Loops):** Οι συνθήκες χρησιμοποιούνται για να ελέγξουν αν μια συνθήκη είναι αληθής ή ψευδής και να εκτελέσουν αντίστοιχο κώδικα. Οι επαναλήψεις επιτρέπουν την επανάληψη ενός κώδικα για έναν ορισμένο αριθμό φορών ή μέχρις ότου μια συνθήκη γίνει ψευδής.
- **Πίνακες (Arrays):** Οι πίνακες είναι δομές δεδομένων που χρησιμοποιούνται για την αποθήκευση πολλών τιμών σε ένα μεταβλητή. Οι τιμές στον πίνακα μπορούν να προσπελαστούν με τη χρήση δείκτη.
- **Αντικείμενα (Objects):** Τα αντικείμενα είναι συλλογές από κλειδιά-τιμές, όπου η τιμή μπορεί να είναι οποιοδήποτε τύπος δεδομένων. Χρησιμοποιούνται για την οργάνωση και τη διαχείριση πολύπλοκων δεδομένων.

Αυτά είναι μερικά από τα βασικά στοιχεία της JavaScript. Κάθε ένα από αυτά παίζει έναν σημαντικό ρόλο στη δημιουργία δυναμικών και ευέλικτων ιστοσελίδων.

### 3.3.1 AJAX

Η τεχνολογία AJAX (Asynchronous JavaScript and XML) αποτελεί μια σύγχρονη προσέγγιση για τη δημιουργία δυναμικών και αλληλεπιδραστικών ιστοσελίδων. Αναπτύχθηκε με σκοπό τη βελτίωση της εμπειρίας του χρήστη, προσφέροντας ασύγχρονη επικοινωνία με τον διακομιστή χωρίς την ανάγκη ανανέωσης ολόκληρης της σελίδας.

Το AJAX είναι μια τεχνική που επιτρέπει την ασύγχρονη ανταλλαγή δεδομένων μεταξύ του περιηγητή του χρήστη και του διακομιστή, χωρίς να χρειάζεται να φορτωθεί η ολόκληρη σελίδα. Αυτό επιτρέπει στις ιστοσελίδες να ανταποκρίνονται γρήγορα στις ενέργειες του χρήστη και να ενημερώνονται δυναμικά χωρίς ανανέωση της σελίδας.

#### Χρησιμότητα του AJAX:

- **Δυναμική Φόρτωση Περιεχομένου:** Η τεχνολογία AJAX επιτρέπει τη φόρτωση νέου περιεχομένου σε μια σελίδα χωρίς την ανάγκη να ανανεωθεί ολόκληρη η σελίδα, βελτιώνοντας έτσι την απόκριση της ιστοσελίδας.
- **Δυναμική Ενημέρωση Δεδομένων:** Με τη χρήση του AJAX, είναι δυνατή η ασύγχρονη ανανέωση τμημάτων μιας σελίδας, όπως πίνακες δεδομένων ή γραφήματα, χωρίς να χρειάζεται να ανανεωθεί όλη η σελίδα.
- **Διαδραστικότητα:** Η τεχνολογία AJAX επιτρέπει τη δημιουργία διαδραστικών εφαρμογών, όπως live chat, δυναμικές φόρμες και αναζητήσεις με αυτόματη συμπλήρωση, προσφέροντας μια βελτιωμένη εμπειρία χρήστη.
- **Μείωση Φόρτου Διακομιστή:** Εφόσον το AJAX επιτρέπει την ασύγχρονη ανταλλαγή δεδομένων, οι ιστοσελίδες μπορούν να μειώσουν τον φόρτο στο διακομιστή, καθώς δεν χρειάζεται να φορτωθεί η σελίδα ξανά κάθε φορά που ο χρήστης δρα.

Συμπερασματικά, η τεχνολογία AJAX έχει επαναπροσδιορίσει τον τρόπο που οι ιστοσελίδες αλληλεπιδρούν με τους χρήστες και προσφέρουν δυναμικό και αλληλεπιδραστικό περιεχόμενο. Η χρήση της τεχνολογίας AJAX βελτιώνει την εμπειρία του χρήστη και μειώνει τον φόρτο στον

διακομιστή, κάνοντας τις ιστοσελίδες πιο αποδοτικές και αποκρίνονται στις ανάγκες των σύγχρονων διαδικτυακών εφαρμογών.

### 3.3.2 JSON

Το JSON (JavaScript Object Notation) αποτελεί ένα δημοφιλές μορφότυπο αποθήκευσης και ανταλλαγής δεδομένων που χρησιμοποιείται ευρέως στον κόσμο του διαδικτύου. Αποτελεί μια απλή, ευανάγνωστη και ευέλικτη μορφή δεδομένων που χρησιμοποιείται για τη μεταφορά δεδομένων ανάμεσα σε εφαρμογές.

Το JSON είναι ένα ανοιχτό μορφότυπο ανταλλαγής δεδομένων που βασίζεται στη σύνταξη αντικειμένων της JavaScript. Αποτελείται από ζεύγη κλειδιών-τιμών και δομές δεδομένων, όπως αντικείμενα, πίνακες και συμβολοσειρές, τα οποία μπορούν να είναι ιεραρχικά οργανωμένα.

#### Χρησιμότητα του JSON:

- **Μεταφορά Δεδομένων:** Το JSON χρησιμοποιείται για τη μεταφορά δεδομένων μεταξύ πελατών και διακομιστών σε διαδικτυακές εφαρμογές. Είναι ιδανικό για τη μεταφορά δομημένων δεδομένων, όπως πληροφορίες χρηστών, αντικείμενα προϊόντων κλπ.
- **Αποθήκευση Δεδομένων:** Το JSON χρησιμοποιείται επίσης για την αποθήκευση δεδομένων σε αρχεία και βάσεις δεδομένων. Η δομή του είναι απλή και ευανάγνωστη, κάνοντας το ιδανικό για αποθήκευση δεδομένων σε αρχεία κειμένου ή σε μνήμη.
- **Διασύνδεση Εφαρμογών:** Εφαρμογές που χρησιμοποιούν διαφορετικές γλώσσες προγραμματισμού μπορούν να επικοινωνούν μεταξύ τους χρησιμοποιώντας το JSON ως κοινό μορφότυπο δεδομένων.
- **Διαχείριση Δομημένων Δεδομένων:** Η δομή του JSON είναι ιδανική για τη διαχείριση δομημένων δεδομένων όπως λίστες, αντικείμενα και συμβολοσειρές, κάνοντας το εύκολο στην ανάλυση και την επεξεργασία.

Το JSON αποτελεί έναν απλό αλλά ισχυρό μορφότυπο δεδομένων που χρησιμοποιείται ευρέως στον κόσμο του διαδικτύου. Η απλή σύνταξή του και η ευελιξία του το καθιστούν ιδανικό για τη μεταφορά, αποθήκευση και διαχείριση δομημένων δεδομένων σε διάφορες εφαρμογές και περιβάλλοντα ανάπτυξης.

Ας δούμε ένα πλήρες παράδειγμα χρήσης της τεχνολογίας AJAX για να ανακτήσουμε δεδομένα από ένα αρχείο JSON και να τα εμφανίσουμε δυναμικά σε μια ιστοσελίδα. Έστω ότι έχουμε το αρχείο data.json που περιέχει τα ακόλουθα δεδομένα:

```
{
  "name": "John Doe",
  "age": 30,
  "city": "New York"
}
```

Εικόνα 2.5: Json file

Και ένα HTML αρχείο που περιέχει ένα κουμπί και έναν κενό δοχείο (<div>) όπου θα εμφανιστούν τα δεδομένα:

```

<body>

<button onclick="loadData()">Load Data</button>
<div id="output"></div>

<script>
function loadData() {
  var xhr = new XMLHttpRequest(); // Δημιουργία ενός νέου αντικειμένου XMLHttpRequest
  xhr.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) { // Εάν η ανταπόκριση είναι έτοιμη και ο κωδικός κατά
      var data = JSON.parse(this.responseText); // Μετατροπή των δεδομένων JSON σε αντικείμενο JavaScript
      document.getElementById("output").innerHTML = `
        <p>Name: ${data.name}</p>
        <p>Age: ${data.age}</p>
        <p>City: ${data.city}</p>
      `; // Εμφάνιση των δεδομένων στο δοχείο με id "output"
    }
  };
  xhr.open("GET", "data.json", true); // Ορισμός της μεθόδου GET και του URL του αρχείου JSON
  xhr.send(); // Αποστολή του αιτήματος
}
</script>
</body>

```

Εικόνα 2.6: HTML Ajax Call

Σε αυτό το παράδειγμα, όταν ο χρήστης κάνει κλικ στο κουμπί "Load Data", η συνάρτηση loadData() εκτελείται. Αυτή η συνάρτηση χρησιμοποιεί το αντικείμενο XMLHttpRequest για να ανακτήσει τα δεδομένα από το αρχείο JSON. Όταν τα δεδομένα ληφθούν επιτυχώς, εμφανίζονται δυναμικά μέσα στο δοχείο με id "output" στη σελίδα HTML.

### 3.4 PHP

Η PHP είναι μια γλώσσα προγραμματισμού για scripting στην πλευρά του διακομιστή (server) και κατά κύριο λόγο χρησιμοποιείται για την δημιουργία δυναμικών και διαδραστικών ιστοσελίδων. Η γλώσσα αυτή είναι από τις πλέον διαδεδομένες και επιλέχθηκε προς χρήση στην εφαρμογή μας καθώς μπορεί να χρησιμοποιηθεί δωρεάν, είναι αρκετά εύκολη στη χρήση ενώ ταυτόχρονα μπορεί να χρησιμοποιηθεί παράλληλα με τους πιο πολλούς διακομιστές διαδικτύου (web servers) και είναι συμβατή με όλα τα λειτουργικά συστήματα.

Η PHP δημιουργήθηκε από τον Rasmus Lerdorf το 1994 ως ένα σύντομο πρόγραμμα περιοδικής ενημέρωσης για τον ιστό. Αρχικά, το ακρωνύμιό της σημαίνει "Personal Home Page" (προσωπική σελίδα), αλλά αργότερα άλλαξε σε "Hypertext Preprocessor" για να δείξει την ικανότητά της να επεξεργάζεται δυναμικά το περιεχόμενο των ιστοσελίδων. Η PHP έγινε γρήγορα δημοφιλής λόγω της απλότητάς της και της δυνατότητάς της να ενσωματώνει HTML.

Καθώς ο ιστός εξελισσόταν, η PHP εξελίσσονταν επίσης. Έγινε ανοιχτού κώδικα το 1995, δίνοντας σε προγραμματιστές από όλο τον κόσμο τη δυνατότητα να συνεισφέρουν στην ανάπτυξή της. Με την πάροδο του χρόνου, προστέθηκαν πολλά χαρακτηριστικά και βελτιώσεις, και η PHP έγινε ένα ισχυρό εργαλείο για την ανάπτυξη δυναμικών ιστοσελίδων και διαδικτυακών εφαρμογών.

Η PHP περιλαμβάνει πολλές ενσωματωμένες βιβλιοθήκες που βοηθάνε στην εκτέλεση συνηθισμένων διαδικτυακών εργασιών, ενώ προσφέρει διασύνδεση με διάφορα συστήματα βάσεων δεδομένων. Συνεπώς, η διαδικτυακή εφαρμογή μας χρησιμοποιεί την γλώσσα PHP για τη διασύνδεση με τη βάση δεδομένων και την εμφάνιση των δεδομένων που βρίσκονται σε αυτή, σχετικά με τα αποθηκευμένα τα στοιχεία των χρηστών.

### 3.4.1 Laravel

Στον κόσμο της ανάπτυξης ιστοσελίδων και διαδικτυακών εφαρμογών, το Laravel έχει καθιερωθεί ως ένα από τα πιο δημοφιλή frameworks. Κατασκευασμένο πάνω στη γλώσσα προγραμματισμού PHP, το Laravel προσφέρει ένα πλήρες και ευέλικτο περιβάλλον ανάπτυξης για τη δημιουργία δυναμικών και σύγχρονων εφαρμογών.

Με την αισθητική και την λειτουργικότητά του, το Laravel αναδεικνύεται ως επιλογή που επιδιώκουν πολλοί προγραμματιστές και εταιρείες για την ανάπτυξη των δικτυακών τους παρουσιάσεων. Από απλές ιστοσελίδες έως πολύπλοκα διαδικτυακά εμπορικά συστήματα, το Laravel προσφέρει ένα ευέλικτο περιβάλλον που επιτρέπει την ανάπτυξη γρήγορα και αποτελεσματικά.

Στο επίκεντρο του Laravel βρίσκεται η ιδέα της απλότητας και της ευκολίας χρήσης, χωρίς όμως να θυσιάζεται η ισχύς και η ευελιξία. Με τη χρήση του Laravel, οι προγραμματιστές μπορούν να επικεντρωθούν στη δημιουργία λειτουργικού και ευανάγνωστου κώδικα, ενώ το πλαίσιο αναλαμβάνει τη διαχείριση πολλών κοινών εργασιών και προβλημάτων.

Με πλούσια τεκμηρίωση, ενεργή κοινότητα και συνεχή εξέλιξη, το Laravel παρέχει ένα σταθερό και αξιόπιστο περιβάλλον για τη δημιουργία μοντέρνων διαδικτυακών εφαρμογών. Ανεξαρτήτως της περιπλοκότητας του έργου, το Laravel προσφέρει τα εργαλεία και τις δυνατότητες που απαιτούνται για την επίτευξη των στόχων ανάπτυξης και δημιουργίας.

### 3.4.2 Routing (Δρομολόγηση)

Η δρομολόγηση είναι ένα από τα βασικά στοιχεία κάθε διαδικτυακής εφαρμογής. Αναλαμβάνει τον ρόλο του "οδηγού" που καθοδηγεί τα αιτήματα των χρηστών προς τις κατάλληλες περιοχές της εφαρμογής. Στο πλαίσιο του Laravel, η δρομολόγηση αποτελεί ένα κρίσιμο κομμάτι του πλαισίου εργασίας, προσφέροντας ευελιξία και οργάνωση στη διαχείριση των αιτημάτων.

Στο πλαίσιο του Laravel, η δρομολόγηση αναφέρεται στη διαδικασία καθοδήγησης των HTTP αιτημάτων από τον server προς τις αντίστοιχες λειτουργίες της εφαρμογής. Κάθε αίτημα HTTP που λαμβάνει η εφαρμογή πρέπει να δρομολογηθεί προς το σωστή ελεγκτή (controller) που θα αναλάβει την αντιμετώπισή του.

Το Laravel χρησιμοποιεί ένα αρχείο δρομολόγησης (routes/web.php ή routes/api.php) για τον ορισμό των διαδρομών της εφαρμογής. Αυτό το αρχείο περιέχει μια σειρά από δηλώσεις διαδρομών, καθεμία από τις οποίες αντιστοιχεί σε ένα URL και μια ελεγκτική λειτουργία. Για παράδειγμα:

```
Route::get('/about', 'AboutController@index');
```

Σε αυτό το παράδειγμα, ένα αίτημα GET προς το URL "/about" θα δρομολογηθεί προς τη μέθοδο "index" του ελεγκτή "AboutController". Μέσω αυτών των διαδρομών, το Laravel διαχειρίζεται την κατεύθυνση των αιτημάτων HTTP προς τις κατάλληλες λειτουργίες της εφαρμογής, παρέχοντας έναν καθαρό, οργανωμένο και ευέλικτο τρόπο διαχείρισης των διαδικτυακών δρομολογήσεων.

### 3.4.3 Controllers (Ελεγκτές)

Οι ελεγκτές (Controllers) αποτελούν ένα σημαντικό στοιχείο στο πλαίσιο εργασίας του Laravel. Αναλαμβάνουν το ρόλο της επεξεργασίας των HTTP αιτημάτων και της απόκρισης προς τους χρήστες. Σε αυτήν την παράγραφο, θα εξετάσουμε τη σημασία των ελεγκτών στο Laravel και πώς λειτουργούν στο πλαίσιο της ανάπτυξης εφαρμογών.

Στο πλαίσιο του Laravel, οι ελεγκτές αναλαμβάνουν τον έλεγχο των διαφόρων δραστηριοτήτων που σχετίζονται με τη λειτουργία της εφαρμογής. Αυτές οι δραστηριότητες μπορεί να περιλαμβάνουν την επεξεργασία δεδομένων, την ανάκτηση πληροφοριών από τη βάση δεδομένων, τη διαχείριση αιτημάτων χρηστών και άλλα.

Στο Laravel, οι ελεγκτές ορίζονται χρησιμοποιώντας τη σύμβαση Model-View-Controller (MVC). Κάθε ελεγκτής αντιστοιχεί σε ένα αρχείο και μπορεί να περιέχει μια σειρά από μεθόδους που αντιστοιχούν σε διαφορετικές λειτουργίες της εφαρμογής.

Οι ελεγκτές στο Laravel διευκολύνουν την οργάνωση του κώδικα, καθώς κάθε λειτουργικότητα της εφαρμογής μπορεί να τοποθετηθεί σε έναν ξεχωριστό ελεγκτή. Αυτό επιτρέπει τη διαχείριση του κώδικα με περισσότερη ευκολία και καθιστά την εφαρμογή πιο δομημένη και εύκολη στη συντήρηση. Επιπλέον, οι ελεγκτές στο Laravel μπορούν να επικοινωνούν με τα μοντέλα (Models) για την ανάκτηση ή ενημέρωση δεδομένων από τη βάση δεδομένων και με τις προβολές (Views) για την παρουσίαση των δεδομένων στους χρήστες.

Οι ελεγκτές αποτελούν ένα κρίσιμο στοιχείο στην ανάπτυξη εφαρμογών στο Laravel. Αναλαμβάνουν τον έλεγχο των διαφόρων λειτουργιών της εφαρμογής και συμβάλλουν στη δομημένη οργάνωση του κώδικα. Μέσω των ελεγκτών, οι προγραμματιστές μπορούν να δημιουργήσουν εφαρμογές που είναι ευέλικτες, εύκολες στη συντήρηση και ανταποκρίνονται αποτελεσματικά στις ανάγκες των χρηστών.

### 3.4.4 Views (Προβολές)

Οι προβολές (Views) αποτελούν ένα ουσιώδες στοιχείο στο πλαίσιο εργασίας του Laravel. Αναλαμβάνουν τον ρόλο της παρουσίασης των δεδομένων στους χρήστες και δημιουργούν τη διεπαφή που αλληλεπιδρά με τον τελικό χρήστη. Σε αυτό το σημείο, θα εξετάσουμε τη σημασία των προβολών στο Laravel και πώς λειτουργούν στο πλαίσιο της ανάπτυξης εφαρμογών.

Στο πλαίσιο του Laravel, οι προβολές αναφέρονται στα αρχεία που περιέχουν το HTML, το CSS και τον JavaScript κώδικα που απαιτείται για τη δημιουργία της διεπαφής του χρήστη. Κάθε προβολή αντιστοιχεί σε μια σελίδα ή ένα τμήμα σελίδας της εφαρμογής και παρουσιάζει δεδομένα στον χρήστη με έναν ευανάγνωστο και ελκυστικό τρόπο.

Στο Laravel, οι προβολές αντιστοιχούν στα αρχεία που βρίσκονται στον κατάλογο `resources/views`. Κάθε προβολή έχει ένα όνομα αρχείου που αντιπροσωπεύει τη σελίδα ή το τμήμα σελίδας που αντιπροσωπεύει. Οι προβολές μπορούν να περιέχουν HTML κώδικα, αλλά επίσης μπορούν να χρησιμοποιούν το Blade, την ενσωματωμένη μηχανή προτύπων του Laravel, για τη δημιουργία δυναμικών προβολών.

Οι προβολές στο Laravel χρησιμοποιούνται συνήθως από τους ελεγκτές για την παρουσίαση δεδομένων στους χρήστες. Μέσω του συστήματος διεύθυνσης του Blade, οι προβολές μπορούν επίσης να ενσωματώνουν δεδομένα από τον ελεγκτή ή άλλες πηγές και να δημιουργήσουν δυναμικό περιεχόμενο.

Κλείνοντας, οι προβολές αποτελούν ένα κρίσιμο στοιχείο στην ανάπτυξη εφαρμογών στο Laravel. Αναλαμβάνουν τον ρόλο της παρουσίασης των δεδομένων στους χρήστες και δημιουργούν τη διεπαφή που αλληλεπιδρά με τον τελικό χρήστη. Μέσω του Blade, της ενσωματωμένης μηχανής προτύπων του Laravel, οι προβολές μπορούν να δημιουργούνται δυναμικά και να παρουσιάζουν πλούσιο περιεχόμενο με ευκολία και ευελιξία.

### 3.4.5 Διαχείριση Εξαρτήσεων (Dependency Management)

Η διαχείριση εξαρτήσεων αποτελεί ένα σημαντικό κομμάτι της διαδικασίας ανάπτυξης λογισμικού. Στο πλαίσιο της εφαρμογής λογισμικού, οι εξαρτήσεις αναφέρονται στις εξωτερικές βιβλιοθήκες, τα πλαίσια (frameworks) ή τα πακέτα λογισμικού που απαιτούνται για τη σωστή λειτουργία της εφαρμογής. Εδώ θα εξετάσουμε τη σημασία της διαχείρισης εξαρτήσεων και τον τρόπο με τον οποίο γίνονται συνήθως στον κόσμο της ανάπτυξης λογισμικού.

Η διαχείριση εξαρτήσεων αναφέρεται στη διαδικασία εύρεσης, εγκατάστασης, ενημέρωσης και διαχείρισης των εξαρτήσεων λογισμικού που απαιτούνται για την ανάπτυξη και τη λειτουργία μιας εφαρμογής. Αυτές οι εξαρτήσεις μπορεί να περιλαμβάνουν βιβλιοθήκες κώδικα, πλαίσια προγραμματισμού, πακέτα λογισμικού ή άλλα εξωτερικά στοιχεία που απαιτούνται για την εκτέλεση της εφαρμογής.

Η διαχείριση εξαρτήσεων είναι κρίσιμη για την επιτυχή ανάπτυξη και λειτουργία της εφαρμογής. Αναλαμβάνει τον ρόλο της διασφάλισης ότι όλα τα απαιτούμενα εξωτερικά στοιχεία είναι διαθέσιμα και εγκατεστημένα σωστά, επιτρέποντας έτσι στην εφαρμογή να λειτουργεί με ομαλότητα και σταθερότητα.

Η διαχείριση συνήθως γίνεται μέσω εργαλείων διαχείρισης πακέτων ή διαχειριστών εξαρτήσεων, όπως το Composer στην περίπτωση του PHP, το npm για τον κώδικα JavaScript, το pip για τη γλώσσα Python κ.ά. Αυτά τα εργαλεία επιτρέπουν στους προγραμματιστές να δηλώνουν τις εξαρτήσεις της εφαρμογής τους σε ένα αρχείο ρυθμίσεων και να εγκαθιστούν αυτές τις εξαρτήσεις με μια απλή εντολή από το πληκτρολόγιο.

### 3.4.6 Εργαλεία Εντολών (Command-Line Tools)

Τα εργαλεία εντολών αποτελούν ένα κρίσιμο στοιχείο στον κόσμο της πληροφορικής και της ανάπτυξης λογισμικού. Αναλαμβάνουν την εκτέλεση διαφόρων καθηκόντων και διεργασιών μέσω της γραμμής εντολών, προσφέροντας έναν αποτελεσματικό και ευέλικτο τρόπο διαχείρισης συστημάτων και εφαρμογών. Θα εξετάσουμε τη σημασία και τη χρήση των εργαλείων εντολών στον κόσμο της τεχνολογίας.

Τα εργαλεία εντολών είναι προγράμματα λογισμικού που εκτελούνται από τη γραμμή εντολών ή το τερματικό παράθυρο ενός υπολογιστή. Αναλαμβάνουν διάφορες λειτουργίες, όπως τη διαχείριση αρχείων, την εκτέλεση εντολών συστήματος, τη διαχείριση πακέτων λογισμικού, τη διαχείριση δικτύου και πολλά άλλα.

## 3.5 Βάση Δεδομένων και Συστήματα Διαχείρισης Βάσεων Δεδομένων

Στο πλαίσιο της ανάπτυξης ιστοσελίδων και εφαρμογών λογισμικού, η χρήση βάσεων δεδομένων είναι ζωτικής σημασίας για την αποθήκευση, την οργάνωση και τη διαχείριση των δεδομένων. Στο παρόν κεφάλαιο, εξετάζουμε τη σημασία των βάσεων δεδομένων και των

συστημάτων διαχείρισης βάσεων δεδομένων (DBMS), καθώς και τις τεχνικές υλοποίησης που χρησιμοποιούνται στην παρούσα εργασία.

### Βαση Δεδομένων

Η βάση δεδομένων αποτελεί ένα οργανωμένο σύνολο δεδομένων που αποθηκεύονται ηλεκτρονικά και είναι προσβάσιμα από πολλαπλούς χρήστες και εφαρμογές. Τα δεδομένα αυτά οργανώνονται σε πίνακες, που αποτελούνται από σειρές και στήλες. Η δομή της βάσης δεδομένων ορίζεται από το σχήμα της, ενώ ο τρόπος πρόσβασης και επεξεργασίας των δεδομένων διέπεται από τις διαφορετικές γλώσσες προγραμματισμού και τα συστήματα διαχείρισης βάσεων δεδομένων.

### Συστήμα Διαχείρισης Βάσεων Δεδομένων

Τα συστήματα διαχείρισης βάσεων δεδομένων (DBMS) αποτελούν λογισμικά εργαλεία που χρησιμοποιούνται για τη διαχείριση, την οργάνωση και την ανάκτηση των δεδομένων σε μια βάση δεδομένων. Τα DBMS παρέχουν ευέλικτες λύσεις για τη δημιουργία, τη διαχείριση και την αναζήτηση των δεδομένων, ενώ παρέχουν επίσης μηχανισμούς για την ασφαλή πρόσβαση και την εξαγωγή δεδομένων.

Για την υλοποίηση της βάσης δεδομένων και των συστημάτων διαχείρισης βάσεων δεδομένων, χρησιμοποιούνται διάφορες τεχνικές ανάπτυξης και προγραμματισμού. Αυτές περιλαμβάνουν τη χρήση γλωσσών προγραμματισμού όπως η SQL (Structured Query Language), όπου θα την δούμε παρακάτω αναλυτικά, την ανάπτυξη κανόνων και περιορισμών στα δεδομένα, καθώς και την εφαρμογή μεθόδων ασφαλείας για την προστασία των πληροφοριών.

Η ορθή υλοποίηση και διαχείριση της βάσης δεδομένων και των συστημάτων διαχείρισης βάσεων δεδομένων αποτελεί κρίσιμο στάδιο για την ανάπτυξη λειτουργικών και αποτελεσματικών ιστοσελίδων και εφαρμογών λογισμικού. Η χρήση σύγχρονων τεχνικών και εργαλείων είναι απαραίτητη για τη δημιουργία αξιόπιστων και αποδοτικών συστημάτων βάσεων δεδομένων που ανταποκρίνονται στις απαιτήσεις της σύγχρονης τεχνολογίας.

### 3.5.1 SQL

Η SQL είναι μια γλώσσα προγραμματισμού που χρησιμοποιείται για τη διαχείριση και τον έλεγχο βάσεων δεδομένων. Αποτελεί το βασικό μέσο επικοινωνίας μεταξύ του χρήστη και της βάσης δεδομένων και επιτρέπει την ανάκτηση, την ενημέρωση, τη διαγραφή και την εισαγωγή δεδομένων.

#### Βασικές Εντολές SQL

- **SELECT:** Η εντολή SELECT χρησιμοποιείται για την ανάκτηση δεδομένων από μια βάση δεδομένων. Η σύνταξή της είναι:

*SELECT column1, column2, ... FROM table\_name;*

- **INSERT INTO:** Η εντολή INSERT INTO χρησιμοποιείται για την εισαγωγή νέων εγγραφών σε έναν πίνακα. Η σύνταξή της είναι:

*INSERT INTO table\_name (column1, column2, ...) VALUES (value1, value2, ...);*

- **UPDATE:** Η εντολή UPDATE χρησιμοποιείται για την τροποποίηση υπάρχοντων εγγραφών σε έναν πίνακα. Η σύνταξή της είναι:

***UPDATE table\_name SET column1 = value1, column2 = value2, ... WHERE condition;***

- **DELETE:** Η εντολή DELETE χρησιμοποιείται για τη διαγραφή υπαρχόντων εγγραφών από έναν πίνακα. Η σύνταξή της είναι:

***DELETE FROM table\_name WHERE condition;***

- **CREATE TABLE:** Η εντολή CREATE TABLE χρησιμοποιείται για τη δημιουργία ενός νέου πίνακα στη βάση δεδομένων. Η σύνταξή της είναι:

***CREATE TABLE table\_name ( column1 datatype, column2 datatype, ... );***

- **DROP TABLE:** Η εντολή DROP TABLE χρησιμοποιείται για τη διαγραφή ενός πίνακα από τη βάση δεδομένων. Η σύνταξή της είναι:

***DROP TABLE table\_name;***

Αυτές είναι μερικές από τις βασικές εντολές της SQL που χρησιμοποιούνται για τη διαχείριση και την αλληλεπίδραση με βάσεις δεδομένων.

### 3.5.2 MariaDB

Η MariaDB είναι ένα ανοιχτού κώδικα σύστημα διαχείρισης βάσεων δεδομένων (ΣΔΒΔ) που παρέχει υψηλή απόδοση, αξιοπιστία και ευελιξία. Αποτελεί μια διακλαδισμένη έκδοση της MySQL και προσφέρει προηγμένες λειτουργίες, ενώ παραμένει συμβατή με την MySQL.

#### Χαρακτηριστικά της MariaDB

- **Ανοιχτός Κώδικας:** Η MariaDB είναι ελεύθερο και ανοιχτού κώδικα λογισμικό, που σημαίνει ότι μπορεί να χρησιμοποιηθεί δωρεάν και να προσαρμοστεί σύμφωνα με τις ανάγκες.
- **Υψηλή Απόδοση:** Η MariaDB προσφέρει υψηλή απόδοση στις επερωτήσεις και την επεξεργασία των δεδομένων.
- **Συμβατότητα με την MySQL:** Είναι σχεδιασμένη να είναι συμβατή με την MySQL, επιτρέποντας την αναβάθμιση από την MySQL στη MariaDB με ελάχιστες ή καθόλου αλλαγές στον κώδικα.
- **Ασφάλεια:** Προσφέρει πολλές λειτουργίες ασφαλείας για την προστασία των δεδομένων, συμπεριλαμβανομένης της κρυπτογράφησης δεδομένων και της διαχείρισης των δικαιωμάτων πρόσβασης.
- **Ευελιξία:** Υποστηρίζει διάφορες μορφές αποθήκευσης δεδομένων, όπως InnoDB, MyISAM και Aria, και παρέχει επιλογές για τη διαμόρφωση της σύνδεσης και της απόδοσης της βάσης δεδομένων.

### 3.6 Τεχνικές υλοποίησης και συνδυασμός όλων των προαναφερθέντων τεχνολογιών

Στο πλαίσιο της παρούσας εργασίας θα γίνει η μελέτη, η ανάλυση απαιτήσεων, ο σχεδιασμός και η έναρξη μέρους της υλοποίησης μίας εφαρμογής Ιστού. Το μοντέλο HTTP που επιλέχθηκε είναι το πλέον διαδεδομένο server-client, όπου η εφαρμογή εκτελείται (runs) από έναν διακομιστή/εξυπηρετητή (server) και προβάλλεται από ένα πρόγραμμα πελάτη (client) μέσω του

περιηγητή Ιστού (browser) του χρήστη. Ο διακομιστής που θα χρησιμοποιηθεί έχει το όνομα Apache, το οποίο διαθέτει ένα σύνολο οδηγιών αναφορικά με την επεξεργασία και διεκπεραίωση των αιτημάτων (requests) των clients από και προς το διακομιστή. Το λειτουργικό σύστημα (Operating System) που θα «στεγάζεται» ο server είναι το Linux, το πλέον δημοφιλές σύστημα για φιλοξενία ιστοσελίδων όπως και δυναμικότερων εφαρμογών.

Η εφαρμογή που θα υλοποιηθεί χαρακτηρίζεται ως δυναμική ιστοσελίδα, με την έννοια ότι τα δεδομένα που προβάλλονται στους χρήστες προέρχονται από τη βάση δεδομένων που φιλοξενείται στο server και αυτή είναι η ουσιώδης διαφορά με τις στατικές ιστοσελίδες, δηλαδή η προέλευση των δεδομένων και η ενημέρωση αυτών μέσα από τη βάση δεδομένων. Οι δυναμικές ιστοσελίδες απαιτούν για την εκτέλεσή τους ένα μεγάλο πλήθος τεχνολογιών, οι οποίες θεωρούνται σήμερα απαραίτητες για την κάλυψη διαφόρων αναγκών, όπως για παράδειγμα η ταχύτητα φόρτωσης της εφαρμογής, η υποστήριξη για κινητές συσκευές και η φιλικότητα αλλά και η συνολική εμπειρία που λαμβάνει ο χρήστης.

Οι δυναμικές εφαρμογές απαιτούν τη χρήση γλώσσας διακομιστή, όπως τις Ruby, PHP, JSP κλπ. Η PHP επιλέχθηκε για την εφαρμογή, λόγω της ταχύτητας υλοποίησης, της ευκολίας που προσφέρει, των δυνατοτήτων που παρέχει και κυρίως λόγω της ευρείας διάδοσής της, αφού η συντριπτική πλειοψηφία των web εφαρμογών έχουν υλοποιηθεί με την εν λόγω γλώσσα. Οι γλώσσες διακομιστή ερμηνεύονται και δρομολογούνται από την τεχνολογία Apache, όπου η τελευταία αναλαμβάνει να εκτελέσει στη μηχανή τα αιτήματα των πελατών.

Η αρχιτεκτονική του έργου που θα δημιουργηθεί ακολουθεί το μοντέλο MVC (Model View Controller) και αποτελεί μια στρατηγική απόφαση για τη φύση αυτού, αφού η εφαρμογή θα ακολουθεί τη δομή του συγκεκριμένου μοντέλου. Το MVC θεωρείται μία κλασική και αποδοτική μέθοδος οργάνωσης του έργου (project) και των μεμονωμένων αρχείων κώδικα και ο πρωταρχικός σκοπός της είναι το όφελος προς τον προγραμματιστή αναφορικά με την περιπλοκότητα και τη μελλοντική ενημέρωση (Continuous Integration) της εφαρμογής. Τα αρχικά MVC προέρχονται από τις λέξεις: Model, View, Controller, που αποτελούν τα 3 κύρια και διακριτά μέρη της εφαρμογής. Το model αφορά την οργάνωση της λογικής του έργου και αποτελεί τον ενδιάμεσο που επιτρέπει τη μεταφορά των δεδομένων από τον controller στο view και αντίστροφα, από τη βάση δεδομένων στον controller. Εδώ είναι ο χώρος όπου γράφονται τα ερωτήματα προς τη βάση δεδομένων. Ο controller έχει άμεση επικοινωνία με το model και αναλαμβάνει να επεξεργαστεί και να εκτελέσει τα δεδομένα που επιστρέφονται από αυτό. Η τελική του εργασία είναι να παραδώσει τα δεδομένα στο view τα οποία θα προβληθούν στον client. Το view είναι ο τελικός κώδικας που βλέπει ο χρήστης και αλληλοεπιδρά μόνο με αυτό. Αυτή η λογική επιτρέπει τη μεγαλύτερη ασφάλεια των εφαρμογών, καθώς ο χρήστης έχει πρόσβαση μόνο στα δεδομένα που αποφασίζεται να εμφανιστούν σε αυτόν.

Τα δεδομένα της εφαρμογής «κατοικούν» σε μια βάση δεδομένων, με τη MySQL να έχει επιλεγεί γι' αυτόν το σκοπό. Η MySQL είναι ένα σχεσιακό σύστημα βάσεων δεδομένων, όπου αυτά είναι οργανωμένα σε γραμμές που βρίσκονται μέσα σε πίνακες, οι οποίοι έχουν δυνατή ή χαλαρή συσχέτιση με άλλους πίνακες. Για παράδειγμα, οι δημοπρασίες που θα προβάλλονται στους χρήστες και τους επισκέπτες της ιστοσελίδας έχουν αποθηκευτεί στον πίνακα auctions που ορίζονται κατ' ελάχιστον από τις στήλες (πεδία/columns) του ονόματος και της περιγραφής. Οι χρήστες και οι κωδικοί αυτών βρίσκονται αποθηκευμένοι στον πίνακα των χρηστών (users), τόσο από πλευρά οργάνωσης των δεδομένων όσο και από πλευρά της ασφάλειας της πληροφορίας. Με λίγα λόγια, οι πιο σημαντικές πληροφορίες της εφαρμογής οργανώνονται σε πίνακες αναφορικά με το είδος και τον

τύπο της πληροφορίας. Τα δεδομένα φορτώνονται, εισάγονται, ενημερώνονται και διαγράφονται στη βάση μέσω της χρήσης και εκτέλεσης ερωτημάτων SQL που γίνονται προς αυτήν.

Η γλώσσα εμφάνισης των δεδομένων της εφαρμογής είναι η γλώσσα σήμανσης (markup) HTML με τα μεμονωμένα στοιχεία της εφαρμογής να οργανώνονται σε κόμβους (nodes) XML. Η HTML περιλαμβάνει ορισμένα στοιχεία αλληλεπίδρασης με το χρήστη, όπως τις φόρμες και εμφάνισης των δεδομένων όπως τους πίνακες. Αυτά τα στοιχεία να μην χρησιμοποιούνται από την πλειοψηφία των εφαρμογών, αλλά η απλή προβολή τους θεωρείται ξεπερασμένη, γι' αυτό και χρησιμοποιούνται επιπλέον βιβλιοθήκες, όπως το Bootstrap, το οποίο και επιλέχθηκε για την υλοποίηση της εφαρμογής. Το Bootstrap είναι ένα εργαλείο (framework) για τη δημιουργία έργων προγραμματισμού στο Web. Απαιτεί πολύ λίγο κώδικα από τον προγραμματιστή, καθώς αναλαμβάνει να κάνει τη δύσκολη σχεδιαστική δουλειά γι' αυτόν. Η κύρια εργασία είναι η βελτιστοποίηση της εμφάνισης των στοιχείων HTML με αποτέλεσμα να κάνει πιο άνετη και φιλική την ιστοσελίδα προς το χρήστη. Τα φύλλα στυλ CSS υπάρχουν πλέον σε όλες τις ιστοσελίδες και ο κύριος σκοπός τους είναι η σχεδίαση και μορφοποίηση των δεδομένων σε κατάλληλες ενότητες και τελικά αποδίδουν μια ξεχωριστή προσωπικότητα σε κάθε ιστότοπο και τον καθιστούν μοναδικό και διαφορετικό σε σχέση με άλλους. Μεγάλο μέρος της βιβλιοθήκης έχει γραφτεί για την υποστήριξη των κινητών συσκευών, οι οποίες αποτελούν σήμερα σημαντικό μέρος της δραστηριότητας στον Ιστό.

Οι σύγχρονες εφαρμογές είναι απαιτητικές από πλευράς φιλικότητας προς το χρήστη και διαχείρισης της πληροφορίας, γι' αυτό και χρησιμοποιείται η γλώσσα client, Javascript. Η εν λόγω γλώσσα επιτρέπει την εκτέλεση σεναρίων στον υπολογιστή του χρήστη. Το κύριο έργο τους είναι η οργάνωση της πληροφορίας και της διαδρομής του χρήστη όσον και η περιήγηση του χρήστη στην ιστοσελίδα. Αναλαμβάνουν επίσης την αλληλεπίδραση με το χρήστη και συνολικά προσφέρουν μια βελτιωμένη εμπειρία χρήσης.

### 3.7 Visual Studio Code

Το Visual Studio Code είναι ένα πρόγραμμα επεξεργασίας κώδικα που αναπτύχθηκε από τη Microsoft για τα Windows, το Linux και το macOS. Χρησιμοποιείται από προγραμματιστές από όλον τον κόσμο για τη γραφή κώδικα σε μια μεγάλη πληθώρα γλωσσών προγραμματισμού. Αυτό το επιτυγχάνει μέσω της επισήμανσης σύνταξης (syntax highlighting), έξυπνης συμπλήρωσης κώδικα (intelligent code completion) και refactoring κώδικα μέσω της μετονομασίας κλάσεων, μεθόδων, ιδιοτήτων και μεταβλητών.

Επιπλέον έχει υποστήριξη για αποσφαλμάτωση κώδικα (debugging), ενώ επιτρέπει στους χρήστες να το προσαρμόσουν αλλά και να το τροποποιήσουν μέσω των επεκτάσεων (extensions) που διαθέτει στις δικές τους ανάγκες. Με λίγα λόγια πρόκειται για ένα δυναμικό με άνετο και φιλικό γραφικό περιβάλλον πρόγραμμα διαθέτοντας πολλές δυνατότητες στους χρήστες τόσο από πλευράς τροποποίησης της εμφάνισης και των λειτουργιών του όσο και από πλευράς επεκτασιμότητας και μάλιστα διατίθεται δωρεάν. Στην περίπτωση της πτυχιακής εργασίας χρησιμοποιήθηκε για τη γραφή κώδικα όλων των απαραίτητων αρχείων της εφαρμογής και συγκεκριμένα για την κωδικοποίηση HTML, CSS, Javascript και PHP.

### 3.8 Επίλογος

Στο κεφάλαιο αυτό αναλύσαμε μια σειρά από εργαλεία και γλώσσες προγραμματισμού που χρησιμοποιήθηκαν στην υλοποίηση της πτυχιακής εργασίας. Αναφέραμε τη σημασία και τη χρήση

τους στο πλαίσιο του συστήματος διαδικτυακών δημοπρασιών και πώλησης προϊόντων που αναπτύχθηκε.

Μέσω της HTML/CSS περιγράψαμε τη δομή και την εμφάνιση της ιστοσελίδας, ενώ με τη JavaScript επιτρέψαμε δυναμική αλληλεπίδραση με τον χρήστη. Η PHP και το framework Laravel χρησιμοποιήθηκαν για την ανάπτυξη της λογικής της εφαρμογής και τη διαχείριση των δεδομένων στο επίπεδο του server. Ακόμη, εξετάσαμε τις δυνατότητες και τα πλεονεκτήματα της MariaDB ως συστήματος διαχείρισης βάσεων δεδομένων.

Με τη χρήση αυτών των εργαλείων και την κατανόηση των γλωσσών προγραμματισμού, κατορθώσαμε να αναπτύξουμε ένα σύστημα που προσφέρει λειτουργικότητα, απόδοση και ασφάλεια. Οι τεχνολογίες αυτές αποτελούν τη βάση ενός σύγχρονου ηλεκτρονικού καταστήματος και αποτελούν το εργαλείο που επέτρεψε την επίτευξη των στόχων της εργασίας.

Συνολικά, η παρουσίαση αυτών των εργαλείων και τεχνολογιών αποτελεί ουσιαστικό μέρος της πτυχιακής εργασίας και αντικατοπτρίζει την προσπάθεια και τη σχολαστικότητα που διακρίνουν την ανάπτυξη του συστήματος.

## Κεφάλαιο 4ο: Τεχνικά Δομικά Στοιχεία Έργου

### 4.1 Ρύθμιση του περιβάλλοντος εργασίας

Ο υπολογιστής με τον οποίο έγινε η ανάπτυξη της εφαρμογής έχει το λειτουργικό Windows 10. Αναφέρθηκε πρωτίτερα ότι η γλώσσα προγραμματισμού PHP απαιτεί την ύπαρξη ενός server που αναλαμβάνει την εκτέλεση των σεναρίων της.

Το δεύτερο απαιτούμενο είναι το σύστημα της βάσης δεδομένων με το οποίο θα συνεργάζεται η εφαρμογή. Οι πλέον βασικές υπηρεσίες που απαιτούνται να υπάρχουν στη μηχανή για τον προγραμματισμό web based εφαρμογής είναι ο web server, μια γλώσσα που εκτελείται σε επίπεδο server και το σύστημα της βάσης δεδομένων. Το εργαλείο XAMPP είναι μια σουίτα προγραμμάτων, ρουτίνων και υπηρεσιών για προγραμματισμό εφαρμογών και περιλαμβάνει την εγκατάσταση εικονικού web server, του Apache και του συστήματος βάσεων δεδομένων MariaDB (MySQL) και τη γλώσσα προγραμματισμού PHP.

Με το πέρας της εγκατάστασης του XAMPP γίνεται εκκίνηση του Apache που ακούει στην πόρτα (port) 80 του μηχανήματος και της εκκίνησης της υπηρεσίας MySQL που ακούει στην πόρτα 3306. Για τη γραφή κώδικα χρησιμοποιήθηκε το πρόγραμμα Visual Studio Code, στο οποίο δεν απαιτήθηκαν ειδικές ρυθμίσεις, παρά μόνο έγινε εισαγωγή του φακέλου του έργου στο πρόγραμμα, έτσι ώστε να επιτρέπεται η εύκολη διαχείριση των αρχείων.

### 4.2 Μεταφόρτωση του Laravel framework

Για την υλοποίηση του συστήματος επιλέχθηκε το framework Laravel, ένα δημοφιλές και εξαιρετικό εργαλείο για γραφή κώδικα PHP. Αποφεύχθηκε η γραφή κώδικα σε core PHP (δηλαδή χωρίς τη χρήση άλλων εργαλείων και βιβλιοθηκών. Μερικοί από τους λόγους της απόφασης αυτής είναι ότι το Laravel επιτρέπει την οργάνωση των αρχείων και φακέλων που περιλαμβάνεται ο κώδικας, κάτι ιδιαίτερα σημαντικό σε μεγάλα έργα. Η τεχνική MVC που περιγράφηκε σε προηγούμενο κεφάλαιο είναι ένα χαρακτηριστικό με σημαντική αξία για τον ίδιο τον προγραμματιστή. Επιπρόσθετα, το Laravel, επιτρέπει την ταχεία και άνετη γραφή κώδικα, με πολλές εργασίες και ρουτίνες να έχουν ήδη υλοποιηθεί από αυτό.

Τρέχοντας την εντολή: `composer create-project --prefer-dist laravel/Laravel` στο Windows Powershell, το Laravel (<https://laravel.com/docs/8.x>) εγκαταστάθηκε στο τοπικό μηχάνημα. Το Laravel, όπως και τα περισσότερα PHP Framework διαθέτουν το αρχείο `.env` που βρίσκεται στον αρχικό φάκελο του έργου, στο οποίο ο προγραμματιστής μπορεί να ορίσει ευαίσθητα δεδομένα, τα οποία είναι προσβάσιμα μόνο από το ίδιο το framework. Ορίστηκαν τα στοιχεία σύνδεσης με τη βάση δεδομένων. Μέσα στο αρχείο μπορούν να αποθηκεύονται άλλα στοιχεία, τα οποία είναι καθολικές μεταβλητές, όπως για παράδειγμα η σύνδεση σε mail server ή σε άλλες υπηρεσίες και APIs (Application Programming Interfaces).

### 4.3 Ρύθμιση των routes

Το επόμενο βήμα ήταν η ρύθμιση των routes, δηλαδή των δρομολογίων κάθε πόρου, δηλαδή ποιο controller ή ποιο view θα αναλάβει να επεξεργαστεί τη διαδρομή πόρου που αιτείται ο χρήστης. Σε κάθε route δόθηκε μια εσωτερική ονομασία για εύκολη χρήση μέσα από την εφαρμογή.

Στο ΠΑΡΑΡΤΗΜΑ Α βλέπουμε το περιεχόμενο του αρχείου `routes/web.php`.

## 4.4 Εγγραφή, είσοδος και προφίλ των χρηστών

Ο controller που αναλαμβάνει την εγγραφή των χρηστών είναι ο AuthController. Όταν ο χρήστης επισκεφτεί την αρχική σελίδα του συστήματος, μπορεί να πατήσει στο σύνδεσμο «Σύνδεση» και εκεί θα βρει ένα κουμπί «Εγγραφή» και θα τον οδηγήσει στη σελίδα join, την οποία διαχειρίζεται η μέθοδος register στον εν λόγω controller. Θα γίνει έλεγχος εάν ο χρήστης είναι ήδη συνδεδεμένος και τότε το σύστημα θα τον ανακατευθύνει στη σελίδα του προφίλ. Ειδιάλλως θα εμφανιστεί το view auth.register (που βρίσκεται στο φάκελο auth). Τα views βρίσκονται στο φάκελο resources. Το περιεχόμενο των PHP μεταβλητών εισάγεται σε διπλά άγκιστρα {{ }} και η εγγενής μηχανή Blade του Laravel θα αναλάβει να εμφανίσει ή και να εκτελέσει το περιεχόμενό τους.

Τα αρχεία CSS και Javascript του κάθε view γίνονται συμπερίληψη των αντίστοιχων αρχείων με την Blade εντολή @include. Τα αρχεία CSS ορίζονται στο view inc.css, ενώ τα αρχεία Javascript βρίσκονται στο αρχείο inc.js.

Στο ΠΑΡΑΡΤΗΜΑ Β βλέπουμε το περιεχόμενο του αρχείου register.

Σε αυτό το σημείο είναι σημαντικό να τονίσουμε τη σημασία του CSRF (Cross Site Request Forgery), το οποίο εισάγεται σε όλες τις φόρμες που στέλνουν αιτήματα στο διακομιστή. Είναι μία μοναδική συμβολοσειρά που ορίζεται από το Laravel και κατά την επεξεργασία αιτήματος (που έπεται της αποστολής της φόρμας) ελέγχεται η ορθότητα και η προέλευση της φόρμας με σκοπό να αποτρέψει επιθέσεις αυτού του είδους από hackers.

Επομένως πρόκειται για ένα χαρακτηριστικό ασφαλείας που προστατεύει σημαντικά την εφαρμογή. Στο view βλέπουμε για παράδειγμα το: \_\_('Όνομα Χρήστη'), το οποίο θα εμφανίσει το λεκτικό Όνομα χρήστη στα Ελληνικά. Στην περίπτωση που προστεθεί νέα γλώσσα στην εφαρμογή, πχ. Αγγλικά, τότε θα εμφανιστεί αυτόματα το λεκτικό Username, ένα χαρακτηριστικό ιδιαίτερα χρήσιμο με το Laravel να μεταφράζει για εμάς τα λεκτικά στα views. Επίσης βλέπουμε και το config('app.name'), το οποίο θα εμφανίσει το λεκτικό Auction House, το οποίο έχει οριστεί στο αρχείο .env. Το αίτημα εγγραφής θα διαχειριστεί από τη μέθοδο process\_register στον controller. Θα γίνει και πάλι έλεγχος εάν ο χρήστης είναι ήδη συνδεδεμένος.

Το επόμενο βήμα είναι ο ορισμός της εγκυρότητας του username και του κωδικού. Το πρώτο πεδίο είναι υποχρεωτικό και πρέπει να είναι μοναδικό στον πίνακα των χρηστών. Το πεδίο του κωδικού είναι κι αυτό υποχρεωτικό με ελάχιστο μήκος 6 χαρακτήρων. Εάν αποτύχει ο έλεγχος της εγκυρότητας, η εφαρμογή θα εμφανίσει στο χρήστη τα λάθη που βρέθηκαν και βρίσκονται στη μεταβλητή \$errors στο view. Σε περίπτωση που ο χρήστης περάσει τον έλεγχο, τότε θα εισαχθεί μια εγγραφή στον πίνακα των χρηστών. Ο ρόλος του χρήστη θα είναι customer και ο κωδικός του θα κατακερματιστεί και θα αποθηκευτεί έτσι στη βάση δίνοντας στην εφαρμογή ένα αξιόλογο επίπεδο ασφαλείας.

Έτσι όταν κάποιος κακόβουλος χρήστης αποκτήσει πρόσβαση στη βάση, δε θα μπορεί να αξιοποιήσει τους κωδικούς των χρηστών. Το αίτημα ολοκληρώνεται με την εμφάνιση του μηνύματος επιτυχίας και ανακατευθύνει το χρήστη στη σελίδα εισόδου.

Στο ΠΑΡΑΡΤΗΜΑ C βλέπουμε το σώμα της μεθόδου process\_register.

Η σελίδα της εισόδου (route: login) θα εμφανίσει το περιεχόμενο του view auth.login όπως αναφέρεται στη μέθοδο login του controller. Επίσης στο ΠΑΡΑΡΤΗΜΑ D βλέπουμε το αρχείο view login.

Όταν ο χρήστης υποβάλλει τη φόρμα πατώντας στο κουμπί «Είσοδος», η μέθοδος `process_login` θα αναλάβει να διαχειριστεί το αίτημα του client. Θα γίνει και πάλι έλεγχος εάν ο χρήστης είναι ήδη συνδεδεμένος και σε αυτήν την περίπτωση θα γίνει ανακατεύθυνσή του στη σελίδα του προφίλ. Αλλιώς θα γίνει έλεγχος της ορθότητας των στοιχείων της φόρμας δηλαδή του `username` και του κωδικού. Εάν ανιχνευθούν λάθη σχετικά με τη μη αποστολή των στοιχείων εισόδου ή τη ταυτοποίηση του χρήστη με τα αντίστοιχα στοιχεία στη βάση δεδομένων, θα τερματίσει η εκτέλεση της μεθόδου και θα εμφανιστούν τα αντίστοιχα λάθη στο χρήστη στη μεταβλητή `$errors`. Όταν ταυτοποιηθεί επιτυχώς ο χρήστης, εάν αυτός είναι διαχειριστής θα μεταβεί αυτομάτως στη σελίδα της διαχείρισης των προϊόντων, ειδάλως εάν ο χρήστης είναι πελάτης θα μεταβεί στη σελίδα του προφίλ.

Στο ΠΑΡΑΡΤΗΜΑ Ε βλέπουμε την συνάρτηση `process_login`.

Η μέθοδος `profile` έχει διπλή δουλειά, τόσο στο να εμφανίσει το περιεχόμενο του view όσο και να επεξεργαστεί το αίτημα του πελάτη. Εδώ θα γίνει έλεγχος εάν δεν ανιχνευθεί `cookie` συνδεδεμένου χρήστη, οπότε και θα γίνει ανακατεύθυνσή του στη σελίδα εισόδου. Σε άλλη περίπτωση θα αναζητηθεί ο συνδεδεμένος χρήστης (με βάση το `id` του) και θα εμφανιστούν τα στοιχεία του χρήστη στα αντίστοιχα πεδία της φόρμας του view, όπως το ονοματεπώνυμο, η διεύθυνσή του κλπ. Όταν ο χρήστης υποβάλλει τη φόρμα, θα αποθηκευτούν στον πίνακα των χρηστών οι τιμές των αντίστοιχων πεδίων που αυτός έχει συμπληρώσει και ένα μήνυμα επιτυχίας (`cookie` το οποίο διαγράφεται στην επόμενη φόρτωση της σελίδας) θα ενημερώσει αναλόγως το χρήστη.

Στο ΠΑΡΑΡΤΗΜΑ F βλέπουμε την μέθοδο `profile`.

## 4.5 Μενού

Στο ΠΑΡΑΡΤΗΜΑ G θα δούμε τον κώδικα του Menu της εφαρμογής.

Αρχικά θα λάβουμε το τελευταίο μέρος του URL (μετά το τελευταίο slash) έτσι ώστε να εντοπίσουμε τη σελίδα στην οποία βρίσκεται ο χρήστης. Αυτό το χρειαζόμαστε για να μπορέσουμε να δείξουμε το ενεργό μενού (με γκρι `background color`) στην τρέχουσα σελίδα του χρήστη. Αυτό το κάνουμε ελέγχοντας σε κάθε μενού εάν το εκάστοτε τελευταίο τμήμα το URL είναι το ίδιο με το τελευταίο τμήμα του URL της τρέχουσα σελίδας. Σε κάθε σύνδεσμο μενού δίνουμε ως ιδιότητα `href` το `{{ route('X') }}`, όπου X το όνομα του αντίστοιχου route που ορίζεται στο `routes/web.php`. Ανάλογα με το εάν ο χρήστης είναι συνδεδεμένος ή όχι θα δείξουμε τους αντίστοιχους συνδέσμους.

Για παράδειγμα εάν ισχύει η πρώτη περίπτωση, θα εμφανίσουμε το σύνδεσμο Έξοδος, ενώ αλλιώς θα δείξουμε το σύνδεσμο Σύνδεση. Παρομοίως, εάν ο χρήστης είναι διαχειριστής, θα του εμφανίζονται οι σύνδεσμοι Προϊόντα και Αναφορές, ενώ αν ο χρήστης είναι πελάτης, θα έχει το σύνδεσμο Προφίλ

## 4.6 Εμφάνιση και εισαγωγή προϊόντων για το διαχειριστή

Στη σελίδα `products` ο διαχειριστής μπορεί να δει συγκεντρωτικά όλα τα προϊόντα του καταστήματος και το έργο της εμφάνισης της συγκεκριμένης σελίδας θα το αναλάβει η μέθοδος `products` στον controller `MyController` στο URL με ονομασία `products`. Ο εν λόγω controller είναι ο τελευταίος που θα δούμε στην εφαρμογή, καθώς έχει επιφορτιστεί με το έργο της λειτουργίας ολόκληρης της εφαρμογής (με εξαίρεση τον έλεγχο των χρηστών στο `AuthController`) τόσο για το διαχειριστή όσο και για τον πελάτη.

Ο διαχειριστής μπορεί να βάλει τα προϊόντα σε δημοπρασία, να εισάγει νέα και να σημειώσει ως απεσταλμένο ένα προϊόν στο χρήστη που κατοχυρώθηκε. Τα προϊόντα βρίσκονται στη μεταβλητή

`$products` και ένας βρόγχος `foreach` (εντός ενός HTML table) θα εμφανίσει πληροφορίες αυτών, την εικόνα τους, το εάν έχουν μπει και πότε σε δημοπρασία όπως και το εάν έχει σταλεί το προϊόν στο χρήστη που κατοχυρώθηκε. Επειδή τα προϊόντα του καταστήματος είναι αρκετά και είναι βέβαιο ότι θα αυξηθούν στο μέλλον, αυτά εμφανίζονται σε σελίδες με κάθε σελίδα να περιλαμβάνει 50 προϊόντα. Η τρέχουσα σελίδα ανιχνεύεται στο URL της σελίδας (με την παράμετρο `currentPage`). Η πρώτη σελίδα είτε δεν έχει αυτή την παράμετρο, είτε αυτή είναι 1. Ο συνολικός αριθμός των σελίδων εξάγεται από τη μεταβλητή `$totalPages` που συγκρατεί το ακέραιο πηλίκο της διαίρεσης Συνολικός αριθμός προϊόντων / μέγεθος σελίδας. Οι σύνδεσμοι πλοήγησης των προϊόντων θα εμφανιστούν στο κάτω μέρος της σελίδας. Συγκεκριμένα, θα εμφανιστούν σύνδεσμοι για την προηγούμενη και επόμενη σελίδα, όπως και κάθε μεμονωμένη σελίδα.

Στο ΠΑΡΑΡΤΗΜΑ Η θα δούμε αναλυτικά τον κώδικα.

Συνεχίζοντας, η εφαρμογή θα εμφανίσει για όλα τα προϊόντα (τα οποία δεν έχουν μπει ακόμα σε δημοπρασία) την επιλογή της δυνατότητας να μπει το προϊόν σε δημοπρασία τη σημερινή ημέρα στο επόμενο κοντινότερο τέταρτο της ώρας (ακριβώς, και τέταρτο, και μισή, και παρά τέταρτο), έτσι ώστε να είναι εύχρηστη και γρήγορη η διαδικασία για το διαχειριστή. Στο ΠΑΡΑΡΤΗΜΑ Η θα δούμε επίσης πως θα γίνει αυτό.

### 4.7 Εισαγωγή δημοπρασίας από το διαχειριστή

Ο διαχειριστής μπορεί να επιλέξει την ημερομηνία και ώρα της δημοπρασίας, να εισάγει την τιμή έναρξης του προϊόντος και θα πατήσει στο κουμπί δημοπρασία. Αυτό το αίτημα γίνεται μέσω Ajax (με Javascript κώδικα), χωρίς να υπάρχει η ανάγκη να επαναφορτώσει ολόκληρη η ιστοσελίδα, παρά μόνο αποστέλλονται στο server μόνο οι πληροφορίες που είναι απαραίτητες για το εκάστοτε αίτημα. Επομένως στο αίτημα της δημοπρασίας θα σταλεί το `id` του προϊόντος, η ημερομηνία εκκίνησης της δημοπρασίας και η τιμή έναρξης. Ο κώδικας υπεύθυνος για την αποστολή και την ολοκλήρωση της επεξεργασίας του αιτήματος είναι στο ΠΑΡΑΡΤΗΜΑ Ι.

### 4.8 Αποστολή προϊόντος κατοχυρωμένης δημοπρασίας σε πελάτη από το διαχειριστή

Όταν ένα προϊόν δημοπρασίας κατοχυρώνεται σε πελάτη, ο χρήστης μπορεί να το δει αυτό στη σελίδα όπου βρίσκονται όλα τα προϊόντα. Συγκεκριμένα, θα δει ένα κουμπί Αποστολή σε X, όπου X είναι είτε το ονοματεπώνυμο του πελάτη εφόσον έχει συμπληρωθεί ή το `id` του πελάτη. Όταν ο χρήστης πατήσει αυτό το κουμπί, ένα Ajax αίτημα θα σταλεί μαζί το `id` της δημοπρασίας στον εξυπηρετητή στο URL: `send/auction/user`. Ο κώδικας υπεύθυνος για την αποστολή του αιτήματος είναι στο ΠΑΡΑΡΤΗΜΑ J.

### 4.9 Εμφάνιση ενεργής δημοπρασίας

Στο URL `auction` εμφανίζεται η ενεργή δημοπρασία (εφόσον υπάρχει ενεργή δημοπρασία τη συγκεκριμένη χρονική στιγμή). Υπεύθυνη για αυτή τη λειτουργία είναι η μέθοδος `auction`, η οποία θα κάνει κλήση στη μέθοδο `getActiveAuction` με όρισμα την επιστροφή τιμής της μεθόδου `getNextCandidateAuctions`. Η τελευταία μέθοδος θα κάνει ερώτημα στον πίνακα `auctions` και θα αναζητήσει τις μη πραγματοποιημένες πιο κοντινές χρονικά (μελλοντικά) δημοπρασίες. Η μέθοδος `getActiveAuction` θα διατρέξει όλες τις δημοπρασίες που βρέθηκαν από το προηγούμενο ερώτημα επιλογής και με τη μέθοδο `between` της βιβλιοθήκης Carbon θα διαπιστωθεί εάν η ημερομηνία έναρξης και η ημερομηνία λήξης της εκάστοτε δημοπρασίας συμπίπτει με την τρέχουσα ημέρα και ώρα. Σε αυτήν την περίπτωση έχουμε ενεργή δημοπρασία και αυτή επιστρέφεται από τη μέθοδο.

Επιπλέον με την ενεργή δημοπρασία, αναζητείται και η επόμενη δημοπρασία με τη μέθοδο `getNextAuction`, η οποία θα κάνει ερώτημα στον πίνακα `auctions` αναζητώντας την επόμενη μη πραγματοποιημένη (εξαιρώντας την ενεργή δημοπρασία εάν υπάρχει) πιο κοντινή μελλοντική χρονικά δημοπρασία.

Το `view` που θα εμφανίσει τα δεδομένα της ενεργής δημοπρασίας είναι το `active-auction`. Συγκεκριμένα, θα εμφανιστεί το μοναδικό αναγνωριστικό της δημοπρασίας, η εικόνα, το όνομα, ο κατασκευαστής και η κατηγορία του προϊόντος. Επιπρόσθετα, θα εμφανιστεί ο συνολικός αριθμός των προσφορών και η προσφορά του συνδεδεμένου χρήστη (εάν υπάρχει). Θα εμφανιστεί η μέγιστη προσφορά εάν υπάρχει και εάν αυτή είναι μεγαλύτερη της τιμής έναρξης, ενώ σε άλλη περίπτωση θα εμφανιστεί η τιμή έναρξης. Εάν η δημοπρασία δεν είναι άγωνα και δεν έχει κατοχυρωθεί ακόμη τότε θα εμφανιστεί ένα πεδίο όπου ο χρήστης μπορεί να βάλει την προσφορά του και το κουμπί «Υποβολή», που όταν ο χρήστης το πατήσει, θα υποβάλλει την προσφορά του. Στην περίπτωση που η δημοπρασία είναι άγωνα, θα εμφανιστεί το λεκτικό Άγωνα δημοπρασία, ενώ εάν είναι κατοχυρωμένη θα εμφανιστεί το λεκτικό Η δημοπρασία κατακυρώθηκε στο χρήστη `#(ID ΤΟΥ ΧΡΗΣΤΗ)`.

Επιπλέον για την ενεργή δημοπρασία εμφανίζεται το λεκτικό Έναρξη και δεξιά η ημερομηνία που εκκίνησε και δίπλα το λεκτικό Λήξη και δεξιά του η ημερομηνία που αυτή θα λήξει. Η αντίστροφη μέτρηση θα εμφανιστεί επίσης (ξεκινώντας από την ημερομηνία έναρξης), με ενημέρωση κάθε δευτερόλεπτο. Αυτό επιτυγχάνεται με τη χρήση της βιβλιοθήκης `Javascript The Final Countdown` (<https://hilios.github.io/jquery.countdown/>), η οποία δεχόμενη την ημερομηνία λήξης της δημοπρασίας θα μετρήσει αντίστροφα προς αυτή ξεκινώντας από την τρέχουσα ώρα.

Στο ΠΑΡΑΡΤΗΜΑ Κ θα δούμε αναλυτικά το τμήμα του κώδικα.

#### 4.10 Υποβολή προσφοράς για ενεργή δημοπρασία

Όταν ο χρήστης πατήσει στο κουμπί Υποβολή θα γίνει ασύγχρονο αίτημα Ajax στο URL: `bid/auction`. Αρχικά θα γίνει έλεγχος εάν ο χρήστης έχει δώσει κενή τιμή, οπότε θα εμφανιστεί μήνυμα λάθους. Ύστερα θα γίνει το αίτημα προς το `server` στέλνοντας το `id` της δημοπρασίας και την προσφορά όπως φαίνεται στο ΠΑΡΑΡΤΗΜΑ Λ.

#### 4.11 Αυτόματη ενημέρωση ενεργής δημοπρασίας

Οι ενεργές δημοπρασίες ενημερώνονται κάθε 10 δευτερόλεπτα στέλνοντας αίτημα Ajax στο URL: `update/auction` με παράμετρο το αναγνωριστικό `id` της δημοπρασίας. Η μέθοδος `updateAuction` θα εκτελεστεί από την πλευρά του `server` και θα καλέσει τη `checkAuctionStatus`, η οποία έχει το εξής έργο: Να αναζητήσει την τελευταία προσφορά της δημοπρασίας και θα ελέγξει πόσα δευτερόλεπτα έχουν περάσει από την τωρινή ώρα. Εάν έχουν παρέλθει 5 λεπτά τότε θα συγκρίνει τη μέγιστη προσφορά της δημοπρασίας με την εναρκτήρια τιμή. Εάν είναι μεγαλύτερη η πρώτη τιμή, τότε αυτό σημαίνει ότι βρέθηκε νικητής και κατοχυρώνεται στο `id` του χρήστη εισάγοντας μια χρονοσφραγίδα στη στήλη `won`. Θα επιστραφεί αυτό το μήνυμα και το `status` του αιτήματος θα είναι `finish` με αποτέλεσμα όταν λάβει η `Javascript` τέτοιο `status`, θα κάνει ανακατεύθυνση του χρήστη στο URL `auction/{ID ΔΗΜΟΠΡΑΣΙΑΣ}`, ένας κόμβος που θα συζητήσουμε σε επόμενη ενότητα.

Επιστρέφουμε στην πλευρά του `server` και εάν δεν έχει βρεθεί νικητής στο πέρας του πεντάλεπτου τότε η δημοπρασία γίνεται άγωνα εισάγοντας τη τρέχουσα χρονοσφραγίδα `fruitless` και θα ενημερώσει ανάλογα το χρήστη. Το `status` του αιτήματος θα είναι κι εδώ `finish`. Στην περίπτωση που έχουν παρέλθει 10 λεπτά από την τελευταία προσφορά, η δημοπρασία γίνεται άγωνα και θα γίνει η ίδια διαδικασία που περιγράφηκε στην προηγούμενη παράγραφο. Σε κάθε άλλη περίπτωση, το `status`

είναι success και θα γίνει κλήση στη μέθοδο updating στη Javascript, ενημερώνοντας το χρήστη ότι μόλις έγινε η ανανέωση της δημοπρασίας.

Στο ΠΑΡΑΡΤΗΜΑ Μ βλέπουμε το σώμα της updateAuction.

### 4.12 Εμφάνιση δημοπρασίας με βάση το id της (ενεργή ή μη)

Στο URL: auction/{id} (όπου id, το αριθμητικό id της δημοπρασίας) βλέπουμε την εκάστοτε δημοπρασία. Η υπεύθυνη μέθοδος στο server για αυτόν τον πόρο είναι η singleAuction. Αρχικά θα επαληθεύσει εάν το id είναι έγκυρο, δηλαδή αριθμητικό. Σε αυτή την περίπτωση, ο server θα ανακατευθύνει το χρήστη στην προηγούμενη σελίδα που αυτός βρισκόταν. Θα γίνει αναζήτηση στον πίνακα auctions με βάση το id κι επίσης θα ληφθεί η μέγιστη προσφορά εάν υπάρχει όπως και η μέγιστη προσφορά του συνδεδεμένου χρήστη. Ταυτόχρονα, θα αναζητηθεί η επόμενη κοντινή μελλοντική χρονικά δημοπρασία με τη μέθοδο getActiveAuction, που περιγράψαμε προηγουμένως.

Στο ΠΑΡΑΡΤΗΜΑ Ν βλέπουμε το σώμα της μεθόδου singleAuction.

Συνεχίζουμε στο view (έχει το όνομα single-auction.blade), εάν δε βρεθεί η δημοπρασία θα εμφανιστεί το λεκτικό Δε βρέθηκε η δημοπρασία με ID: {ID}. Σε άλλη περίπτωση θα εμφανιστεί το μοναδικό αναγνωριστικό της δημοπρασίας, η εικόνα, ο κατασκευαστής και η κατηγορία του προϊόντος. Εάν η δημοπρασία είναι ενεργή, τότε θα εμφανιστεί ένα μήνυμα στην κορυφή της σελίδας που θα έχει ένα σύνδεσμο που οδηγεί στη σελίδα της ενεργής δημοπρασίας (view: active-auction.blade), καθώς ο χρήστης μπορεί να δει εκεί την αντίστροφη μέτρηση.

Επιπλέον θα εμφανιστούν ο αριθμός των συνολικών προσφορών και η προσφορά του χρήστη εάν υπάρχει. Η τρέχουσα τιμή θα εμφανιστεί: όταν υπάρχει προσφορά μεγαλύτερη της τιμής έναρξης εμφανίζεται τότε η πρώτη, αλλιώς η δεύτερη. Εάν η δημοπρασία δεν είναι άγνη ούτε κατοχυρωμένη, ο χρήστης μπορεί να υποβάλει προσφορά και τη λειτουργία την περιγράψαμε προηγουμένως στην ενεργή δημοπρασία. Σε άλλη περίπτωση εμφανίζονται τα αντίστοιχα λεκτικά, τα οποία περιγράψαμε σε προηγούμενη ενότητα. Τέλος εμφανίζονται οι ημερομηνίες έναρξης και λήξης της δημοπρασίας καθώς και ο σύνδεσμος της επόμενης δημοπρασίας εάν υπάρχει.

Στο ΠΑΡΑΡΤΗΜΑ Ν βλέπουμε τον κώδικα.

### 4.13 Αναζήτηση δημοπρασίας

Η αναζήτηση γίνεται στο URL search. Παρέχονται 3 πεδία αναζήτησης για να ανιχνευθεί η δημοπρασία, τα οποία μπορούν να χρησιμοποιηθούν είτε κατά μόνας είτε πολλά από αυτά:

- Κατηγορία
- Κατασκευαστής
- Προϊόν ή κατασκευαστής

Τα 2 πρώτα πεδία είναι dropdown menus, προτρέποντας το χρήστη να επιλέξει από τις διαθέσιμες επιλογές. Εμφανίζονται προς επιλογή όλες οι κατηγορίες και όλοι οι κατασκευαστές. Το τελευταίο πεδίο επιτρέπει το χρήστη να αναζητήσει λέξεις ή μέρος αυτών που αφορούν το όνομα ή τον κατασκευαστή των προϊόντων. Όταν ο χρήστη πατήσει στο κουμπί Αναζήτηση θα γίνει ένα POST αίτημα στο server στο URL search που θα αναληφθεί από τη μέθοδο searchAuction. Εάν το 3ο πεδίο είναι συμπληρωμένο, τότε θα γίνει ερώτημα επιλογής στον πίνακα των προϊόντων. Συγκεκριμένα θα γίνουν ερωτήματα LIKE τόσο στη στήλη name όσο και στη στήλη manufacturer. Εφόσον ο χρήστης έχει επιλέξει (είτε μόνο αυτή είτε μαζί με το 3ο πεδίο) τότε θα προστεθεί η συνθήκη για το πεδίο

category\_id που επέλεξε ο χρήστης. Παρόμοια θα γίνει και το πεδίο manufacturer. Για κάθε προϊόν που βρέθηκε, θα ληφθούν οι αντίστοιχες δημοπρασίες, η μέγιστη προσφορά και η κατηγορία στην οποία ανήκει. Στη συνέχεια, θα γίνει επανάληψη όλων των προϊόντων έτσι ώστε να ευρεθούν μόνο τα δημοπρατημένα και μόνο αυτά θα εμφανιστούν στο χρήστη. Τέλος θα εμφανιστούν οι όροι που αναζήτησε ο χρήστης στα αντίστοιχα πεδία.

Στο ΠΑΡΑΡΤΗΜΑ Ο θα δούμε τον κώδικα.

#### 4.14 Αναφορές δημοπρασιών για το διαχειριστή

Στο URL: reports θα εκτελεστεί η μέθοδος auctionReports στην οποία έχει πρόσβαση μόνο ο διαχειριστής. Θα αναζητηθούν οι κατοχυρωμένες δημοπρασίες και θα δοθούν ως όρισμα στη μέθοδο getAverageAuctionTimeInSeconds. Θα διατρέξει όλες τις δημοπρασίες (άγονες και μη) και με τη βοήθεια της βιβλιοθήκης Carbon θα αναζητηθεί ο χρόνος που έχει διανυθεί από την ώρα έναρξης. Έτσι θα προσθέσουμε όλα τα δευτερόλεπτα για όλες τις δημοπρασίες και θα τα διαιρέσουμε διά του συνολικού αριθμού τους.

Στο ΠΑΡΑΡΤΗΜΑ Ρ ακολουθεί το σώμα της getAverageAuctionTimeInSeconds.

#### 4.15 Περιγραφή του σχήματος της βάσης δεδομένων

Η βάση δεδομένων ονομάστηκε auction και αποτελείται από 5 πίνακες. Το σχήμα όπως και μερικές εγγραφές στη βάση έγιναν με τη βοήθεια του εργαλείου Navicat, το οποίο είναι ένα MySQL client που βοηθά σημαντικά στο σχεδιασμό και διαχείριση βάσεων δεδομένων. Η σχεδίαση της βάσης έγινε χρησιμοποιώντας σωστές πρακτικές που υπαγορεύει η γλώσσα SQL και θεωρείται ότι το σχήμα βρίσκεται σε τρίτη κανονική μορφή (3NF). Παρακάτω βλέπουμε το μοντέλο οντοτήτων-συσχετίσεων (E-R diagram):



### Πίνακας users:

Ο πίνακας των χρηστών είναι ο πιο σημαντικός πίνακας στις περισσότερες web εφαρμογές καθώς περιέχει τα δεδομένα των χρηστών, τα οποία είναι ευαίσθητα και η εφαρμογή που τα χρησιμοποιεί πρέπει να εγγυάται την ασφάλειά τους και να είναι προσβάσιμα και διαχειρίσιμα μόνο από τους ίδιους

τους χρήστες που τους αφορούν. Επιπλέον, το όνομα χρήστη και ο κωδικός τους αποτελούν το μοναδικό «διαβατήριο» για την είσοδο στην εφαρμογή, γι' αυτό και απαιτούν ιδιαίτερη προσοχή όλες οι λειτουργίες των χρηστών.

Ο πίνακας των χρηστών έχει τη στήλη id που είναι πρωτεύον κλειδί με αύξουσα αρίθμηση, δηλαδή με κάθε νέα εγγραφή η SQL δίνει αυτόματα ένα νέο μοναδικό αριθμό που χαρακτηρίζει κάθε εγγραφή χρήστη. Το ίδιο ισχύει και για τους υπόλοιπους πίνακες της εφαρμογής που έχουν τη στήλη id. Αυτός ο αριθμός είναι η ταυτότητα του κάθε χρήστη και το id του χρησιμοποιείται από το σύστημα για τη δημιουργία και ανάγνωση του cookie (συνεδρίας). Τα πεδία username, password, role, fullname, address, tel, card\_number, card\_holder, card\_expiry, unique\_id είναι τύπου συμβολοσειράς με μέγιστο μήκος 255 χαρακτήρων.

Το πεδίο username είναι ψευδώνυμο του χρήστη και εδώ αποθηκεύουμε το email του, το οποίο χρησιμοποιείται μαζί με το password (κωδικός) για την ταυτοποίηση του χρήστη κατά την είσοδο της εφαρμογής. Ο κωδικός των χρηστών δεν αποθηκεύεται αυτούσιος στη βάση για λόγους ασφαλείας, αλλά αποθηκεύεται ως μία κατακερματισμένη (hash) συμβολοσειρά που έχει παραχθεί με ασύμμετρη κρυπτογράφηση με τον αλγόριθμο bcrypt. Το πεδίο role αφορά τον τύπο του χρήστη που σχετίζεται με την εκάστοτε εγγραφή. Κάθε νέα εγγραφή χρήστη στο σύστημα έχει το ρόλο customer, ενώ ο διαχειριστής με ρόλο admin, είχε εισαχθεί χειροκίνητα στη βάση. Στο πεδίο address αποθηκεύουμε τη διεύθυνση του πελάτη, στο πεδίο tel αποθηκεύουμε το τηλέφωνο που εισάγει ο πελάτης στο προφίλ του, τα πεδία card\_number, card\_holder, card\_expiry αφορούν τον αριθμό κάρτας, το όνομα στην κάρτα και το μήνα και έτος λήξης της κάρτας αντίστοιχα. Το πεδίο unique\_id δε χρησιμοποιείται, αλλά για την ταυτοποίηση του χρήστη χρησιμοποιείται το id του χρήστη στον ίδιο πίνακα. Στον εν λόγω πίνακα υπάρχουν τα πεδία created\_at και updated\_at. Και τα 2 είναι χρονοσφραγίδες (timestamp), με το created\_at να υποδηλώνει την ημερομηνία και ώρα που η εγγραφή εισήχθη στο σύστημα και το updated\_at να σχετίζεται με την ημερομηνία και ώρα που η εγγραφή ενημερώθηκε στο σύστημα. Αυτά τα πεδία δεν συμπληρώνονται με αυτόματο τρόπο από τη βάση (όπως γίνεται με το πεδίο id), αλλά συμπληρώνονται μέσα από την εφαρμογή.

### **Πίνακας products:**

Ο πίνακας products περιλαμβάνει τα προϊόντα. Το πεδίο id είναι το μοναδικό αναγνωριστικό χαρακτηριστικό του προϊόντος. Το πεδίο name είναι η ονομασία του προϊόντος (συμβολοσειρά 255 χαρακτήρων), το πεδίο price (δεκαδικός 10 ψηφίων και 2 δεκαδικά ψηφία) είναι η τιμή που αυτό έχει, το πεδίο category\_id (ακέραιος 11 ψηφίων) είναι το id της κατηγορίας (νοητό ξένο κλειδί στον πίνακα categories), το πεδίο manufacturer (συμβολοσειρά 255 χαρακτήρων) είναι το όνομα του κατασκευαστή. Εδώ συναντάμε και πάλι τα πεδία created\_at και updated\_at και έχουν περιγραφεί στον πίνακα users.

Στον εν λόγω πίνακα έχουν μπει ενδεικτικά 1000 εγγραφές χειροκίνητα στη βάση με τη βοήθεια του web εργαλείου <https://mockaroo.com>, το οποίο παράγει τυχαίες τιμές, οι οποίες χρησιμοποιήθηκαν στην τιμή, το id της κατηγορίας και τον κατασκευαστή του προϊόντος. Τα προϊόντα δεν ονοματίστηκαν με τυχαίο τρόπο, παρά μόνο έλαβαν το όνομα Product X, όπου X το id του εκάστοτε προϊόντος. Αυτό έγινε για να φαίνεται πιο ολοκληρωμένο το σύστημα αναφορικά με τα προϊόντα που διατίθενται από το ηλεκτρονικό κατάστημα.

### **Πίνακας categories:**

Ο πίνακας categories περιλαμβάνει τις κατηγορίες στις οποίες ανήκουν τα προϊόντα και περιγράφεται από το πεδίο id (μοναδικό αναγνωριστικό) και το πεδίο name, δηλαδή την ονομασία της κατηγορίας. Εδώ ακολουθήθηκε η ίδια διαδικασία συμπλήρωσης ενδεικτικών κατηγοριών. Εισήχθησαν 1000 ενδεικτικές κατηγορίες με ονομασία Category X, όπου το X το id της εκάστοτε κατηγορίας.

### Πίνακας auctions:

Ο πίνακας αυτός αποθηκεύει τις δημοπρασίες που δημιουργεί ο διαχειριστής του καταστήματος. Το πεδίο `id` περιγράφηκε στους προηγούμενες πίνακες, δηλαδή είναι ο αύξων αριθμός της δημοπρασίας. Το πεδίο `product_id` (ακέραιος) είναι το `id` του προϊόντος και είναι νοητό ξένο κλειδί στον πίνακα των προϊόντων. Το πεδίο `unique_id` (συμβολοσειρά) είναι το αναγνωριστικό κάθε δημοπρασίας και είναι μια μοναδική συμβολοσειρά που ανατίθεται από την εφαρμογή κατά τη δημιουργία της δημοπρασίας. Το πεδίο `opening_price` (δεκαδικός 10 ψηφίων με 2 δεκαδικά) είναι η αρχική τιμή του προϊόντος κατά την εκκίνηση της δημοπρασίας. Το πεδίο `start_datetime` (datetime, ημερομηνία) είναι η ημερομηνία και ώρα που θα εκκινήσει η δημοπρασία, ενώ το πεδίο `end_datetime` είναι παρόμοιο με το `start_datetime` και αφορά την ημερομηνία και ώρα που θα λήξει η δημοπρασία. Το πεδίο `initial_duration_seconds` (ακέραιος) είναι η προγραμματιζόμενη διάρκεια σε δευτερόλεπτα της δημοπρασίας. Το πεδίο `auto_fruitless_seconds` (ακέραιος) είναι ο χρόνος σε δευτερόλεπτα στο πέρας του οποίου η δημοπρασία θα θεωρείται άγονη και ξεκινά να μετράει από την τιμή του `start_datetime`. Το πεδίο `fruitless` (datetime) είναι η ημερομηνία και ώρα στην οποία η δημοπρασία έγινε άγονη, ενώ το πεδίο `won` είναι η ημερομηνία και ώρα στην οποία η δημοπρασία κατοχυρώθηκε σε κάποιο χρήστη. Το πεδίο `winner_user_id` (ακέραιος) είναι το `id` του χρήστη στον οποίο έγινε η κατοχύρωση και είναι νοητό ξένο κλειδί στον πίνακα των χρηστών. Το πεδίο `sent_to_user` (timestamp) είναι χρονοσφραγίδα και σηματοδοτεί το πότε το προϊόν που κατοχυρώθηκε στο χρήστη έχει σταλεί σε αυτόν (όταν το πεδίο είναι συμπληρωμένο). Εδώ συναντάμε και πάλι τα πεδία `created_at` και `updated_at` και έχουν περιγραφεί στον πίνακα `users`.

### Πίνακας auction\_bids:

Είναι ο πίνακας των προσφορών που ορίζουν οι χρήστες για τις δημοπρασίες. Το πεδίο `id` έχει περιγραφεί νωρίτερα. Το πεδίο `auction_id` είναι νοητό ξένο κλειδί στον πίνακα των δημοπρασιών. Το πεδίο `price` (δεκαδικός) είναι η προσφορά του χρήστη και το πεδίο `user_id` είναι το `id` του χρήστη που έδωσε την προσφορά επί της δημοπρασίας (νοητό ξένο κλειδί στον πίνακα `users`). Εδώ συναντάμε και πάλι τα πεδία `created_at` και `updated_at` και έχουν περιγραφεί στον πίνακα `users`.

## 4.16 Επίλογος

Σε αυτό το κεφάλαιο κάλυψαμε εκτενώς την λειτουργία του κώδικα καθώς είδαμε και κομμάτια του. Στο επόμενο και τελευταίο κεφάλαιο θα δούμε κάποια στιγμιότυπα από την εφαρμογή.

## Κεφάλαιο 5ο: Στιγμιότυπα Χρήσης της Εφαρμογής

Σε αυτήν την ενότητα αναλύονται τα συμπεράσματα της διπλωματικής εργασίας και παρουσιάζονται προτάσεις βελτίωσής της.





*Είσοδος στο σύστημα*

*Εγγραφή στο σύστημα*





#	Εικόνα	Όνομα	Τιμή	Κατηγορία	Κατασκευαστής	Ενέργειες
1		Product 1	20.52	CATEGORY 974	Chillingworth	Δημοπρασία 09-07-2021 00:15:00
2		Product 2	46.01	CATEGORY 907	Marcone	Δημοπρασία 10-07-2021 00:15:00 Εστάλη 12-07-2021 23:54:20
3		Product 3	60.67	CATEGORY 755	Gowdridge	Δημοπρασία 13-07-2021 22:00:00 Αποσπαστή σε χρήση #1
4		Product 4	79.01	CATEGORY 663	Pail	Δημοπρασία 26-07-2021 21:00:00

*Διαχειριστής – Σελίδα προϊόντων*



## Κεφάλαιο 5

#	Εικόνα	Όνομα	Τιμή	Κατηγορία	Κατασκευαστής	Ενέργειες
	<input type="button" value="Choose File"/> No file chosen	<input type="text" value="Όνομα"/>	<input type="text" value="Τιμή"/>	<input type="text" value="CATEGORY 1"/>	<input type="text" value="Κατασκευαστής"/>	<input type="button" value="Αποθήκευση"/>
1		Product 1	20.52	CATEGORY 974	Chillingworth	<input type="button" value="Δημοπρασία 09-07-2021 00:15:00"/>
2		Product 2	46.01	CATEGORY 907	Marcone	<input type="button" value="Δημοπρασία 10-07-2021 00:15:00"/> <input type="button" value="Εστώλη 12-07-2021 23:54:20"/>
3		Product 3	60.67	CATEGORY 755	Gowdridge	<input type="button" value="Δημοπρασία 13-07-2021 22:00:00"/> <input type="button" value="Αποστολή σε χρήση #1"/>
4		Product 4	79.01	CATEGORY 663	Pail	<input type="button" value="Δημοπρασία 26-07-2021 21:00:00"/>

*Διαχειριστής – Εισαγωγή προϊόντος (πρώτη γραμμή του πίνακα)*

9		Product 9	59.96	CATEGORY 495	Kid	<input type="text" value="04-08-2021 11:15:00"/> <b>August 2021</b> Su Mo Tu We Th Fr Sa 25 26 27 28 29 30 31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4
10		Product 10	80.42	CATEGORY 142	Dixson	
9		Product 9	59.96	CATEGORY 495	Kid	<input type="text" value="04-08-2021 11:15:00"/> <b>4 August 2021</b> 15:00 15:05 15:10 15:15 15:20 15:25 15:30 15:35 15:40 15:45 15:50 15:55
9		Product 9	59.96	CATEGORY 495	Kid	<input type="text" value="04-08-2021 15:15:00"/> <input type="text" value="60"/> <input type="button" value="Δημοπρασία"/>

*Διαχειριστής – Επιλογή ημερομηνίας, ώρας και τιμή προϊόντος για προσθήκη σε δημοπρασία*

3		Product 3	60.67	CATEGORY 755	Gowdridge	<p>Δημοπρασία 13-07-2021 22:00:00</p> <p>Αποστολή σε χρήστη #1</p>
3		Product 3	60.67	CATEGORY 755	Gowdridge	<p>Δημοπρασία 13-07-2021 22:00:00</p> <p>Εστάλη 05-08-2021 00:16:43</p>

*Διαχειριστής – Αποστολή προϊόντος σε χρήστη*

Auction House Αρχική **Δημοπρασίες** Αναζήτηση Προϊόντα Αναφορές Έξοδος

Ενεργή δημοπρασία (κί0ρ#07Wζ)

**Product 9**  
Κατασκευαστής: Kid Κατηγορία: CATEGORY 495

Εναρξη: 05-08-2021 00:15:00 Λήξη: 05-08-2021 01:15:00

00 ώρες 56 λεπτά 37 δευτερόλεπτα

Προσφορές: 0 Τρέχουσα τιμή: € 59.96  
Η προσφορά μου: €

**Υποβολή**

*Ενεργή δημοπρασία*

Auction House Αρχική **Δημοπρασίες** Αναζήτηση Προφίλ ΚΡΗΣΤΗΣ 1 Έξοδος

Ενεργή δημοπρασία (κί0ρ#07Wζ)

**Product 9**  
Κατασκευαστής: Kid Κατηγορία: CATEGORY 495

Εναρξη: 05-08-2021 00:15:00 Λήξη: 05-08-2021 01:15:00

00 ώρες 55 λεπτά 00 δευτερόλεπτα

Προσφορές: 0 Τρέχουσα τιμή: € 60.00  
Η προσφορά μου: €60.00

**Υποβολή**

*Ο χρήστης έχει υποβάλει προσφορά στη δημοπρασία*

Auction House Αρχική Δημοπρασίες **Αναζήτηση** Προφίλ ΚΡΗΣΤΗΣ 1 Έξοδος

Αναζήτηση δημοπρασιών





Επιλέξτε κατηγορία  Επιλέξτε κατασκευαστή  Αναζητήστε προϊόν ή κατασκευαστή  **Αναζήτηση**

## Κεφάλαιο 5

Auction House Αρχική Δημοπρασίες Αναζήτηση Προφίλ ΧΡΗΣΤΗΣ 1 Έξοδος

Αναζήτηση δημοπρασιών


Επιλέξτε κατηγορία  Επιλέξτε κατασκευαστή  PRODUCT 1

#	Εικόνα	Όνομα	Τιμή	Κατηγορία	Κατασκευαστής	Δημοπρασία
1		Product 1	€ 20.52	CATEGORY 974	Chillingworth	09-07-2021 00:15:00 <input type="button" value="Άγωνα"/>
2		Product 15	€ 87.67	CATEGORY 109	Sircomb	12-07-2021 14:35:00 <input type="button" value="Άγωνα"/>
3		Product 17	€ 22.26	CATEGORY 512	Terbrug	12-07-2021 00:04:00 <input type="button" value="Άγωνα"/>
4		Product 101	€ 77.11	CATEGORY 31	Teesdale	12-07-2021 11:30:00 <input type="button" value="Άγωνα"/>

### Αναζήτηση δημοπρασίας (με όρο PRODUCT 1)

Auction House Αρχική Δημοπρασίες Αναζήτηση Προϊόντα Αναφορές Έξοδος

Δημοπρασία (SM2dNNM5M5)



**Product 14**

Κατασκευαστής: Tumily | Κατηγορία: CATEGORY 389

Έναρξη: 05-08-2021 01:30:00 | Λήξη: 05-08-2021 02:30:00

Προσφορές: 0 | Τρέχουσα τιμή: € 67.24

Η προσφορά μου: €





### Μελλοντική δημοπρασία

Auction House Αρχική Δημοπρασίες Αναζήτηση Προϊόντα Αναφορές Έξοδος

Άγωνα δημοπρασίας: 15

Μέσος χρόνος δημοπρασίας: 1,703.35 δευτερόλεπτα (από 17)

Προϊόντα άγωνων δημοπρασιών


#	Εικόνα	Όνομα	Εναρκτήρια τιμή	Μέγιστη προσφορά	Δημοπρασία
1		Product 1	€ 20.52		09-07-2021 00:15:00
2		Product 5	€ 86.41		10-07-2021 01:30:00
3		Product 51	€ 34.35	€ 6.00	10-07-2021 13:24:00
4		Product 6	€ 7.82	€ 7.50	11-07-2021 13:30:00

Διαχειριστής – Αναφορές δημοπρασιών

Auction House Αρχική Δημοπρασίες Αναζήτηση Προϊόντα Αναφορές

Δημοπρασία (κι0pf07WJZ)

**Product 9**  
 Κατασκευαστής: Kid Κατηγορία: CATEGORY 495 Έναρξη: 05-08-2021 00:15:00 Λήξη: 05-08-2021 01:15:00



Προσφορές: 1 Τρέχουσα τιμή: €  
 Η προσφορά μου: € 60,00

Η δημοπρασία κατακυρώθηκε στο χρήστη #2

Κατοχυρωμένη δημοπρασία

Auction House Αρχική **Δημοπρασίες** Αναζήτηση Προϊόντα Αναφορές Έξοδος

Δεν υπάρχει ενεργή δημοπρασία. Επόμενη δημοπρασία: 05-08-2021 01:30:00

Επόμενη δημοπρασία

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] “W3Schools Online Web Tutorials.” *W3Schools Online Web Tutorials*, <https://www.w3schools.com/>. Accessed 13 May 2024.
- [2] *Mozilla Developer Network (MDN) - HTML/CSS*. <https://developer.mozilla.org/en-US/docs/Web>. Accessed 13 May 2024.
- [3] “The Modern JavaScript Tutorial.” *The Modern JavaScript Tutorial*, <https://javascript.info/>. Accessed 13 May 2024.
- [4] “PHP: PHP Manual - Manual.” *PHP: Hypertext Preprocessor*, <https://www.php.net/manual/en/>. Accessed 13 May 2024.
- [5] “Installation - Laravel 11.x - The PHP Framework For Web Artisans.” *Laravel - The PHP Framework For Web Artisans*, <https://laravel.com/docs>. Accessed 13 May 2024.
- [6] “MySQL :: MySQL Documentation.” *MySQL :: Developer Zone*, <https://dev.mysql.com/doc/>. Accessed 13 May 2024.
- [7] “MariaDB Server Documentation - MariaDB Knowledge Base.” *MariaDB KnowledgeBase*, <https://mariadb.com/kb/en/documentation/>. Accessed 13 May 2024.
- [8] Microsoft. “Documentation for Visual Studio Code.” *Visual Studio Code - Code Editing. Redefined*, Microsoft, 3 Nov. 2021, <https://code.visualstudio.com/docs>.

## ΠΑΡΑΡΤΗΜΑ Α : Κώδικας routes/web.php

```
Route::get('/', function () { return view('home'); })->name('home');

Route::get('/login', 'App\Http\Controllers\Auth\AuthController@login')->name('login');

Route::post('/login', 'App\Http\Controllers\Auth\AuthController@process_login')->name('process-login');

Route::get('/join', 'App\Http\Controllers\Auth\AuthController@register')->name('register');

Route::post('/process-join', 'App\Http\Controllers\Auth\AuthController@process_register')->name('process-register');

Route::get('/logout', 'App\Http\Controllers\Auth\AuthController@logout')->name('logout');

Route::group(['middleware' => 'auth'], function() {

    Route::get('/debug', 'App\Http\Controllers\MyController@debug')->name('debug');

    Route::get('/mock', 'App\Http\Controllers\MyController@mock')->name('mock');

    Route::get('/profile', 'App\Http\Controllers\Auth\AuthController@profile')->name('profile');

    Route::post('/profile', 'App\Http\Controllers\Auth\AuthController@profile')->name('profile');

    Route::get('/products', 'App\Http\Controllers\MyController@products')->name('products');

    Route::post('/add/auction', 'App\Http\Controllers\MyController@addAuction')->name('add-auction');

    Route::post('/add/product', 'App\Http\Controllers\MyController@addProduct')->name('add-product');

Route::get('/auction', 'App\Http\Controllers\MyController@auction')->name('auction');

    Route::get('/auction/{id}', 'App\Http\Controllers\MyController@singleAuction')->name('auction-single');
```

```

Route::post('/update/auction', 'App\Http\Controllers\MyController@updateAuction')->name('update-auction');

Route::post('/bid/auction', 'App\Http\Controllers\MyController@bidAuction')->name('bid-auction');

Route::get('/search', 'App\Http\Controllers\MyController@searchAuction')->name('search-auction');

Route::post('/search', 'App\Http\Controllers\MyController@searchAuction')->name('search-auction');

Route::get('/reports', 'App\Http\Controllers\MyController@auctionReports')->name('auction-reports');

Route::post('/send/auction/user', 'App\Http\Controllers\MyController@sendAuctionToUser')->name('send-auction-user');

});

```

## ΠΑΡΑΡΤΗΜΑ Β : Κώδικας αρχείου register

```

<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
<head>
    @include('inc/css')
    <meta charset="utf-8">
    <meta name="csrf-token" content="{{ csrf_token() }}">
    <title>{{ config('app.name') }} | Εγγραφή</title>
</head>
<body>

    @include('inc/menu')

    <div class="alert alert-primary text-center" role="alert">
        Εγγραφή στο {{ config('app.name') }}
    </div>

    <form class="auth-form mt-5" method="post" action="{{ route('process-register') }}">

        {{ csrf_field() }}

        @if ( $errors->any() )
            @foreach ( $errors->all() as $error)

```

```

        <div class="alert alert-danger text-
center" role="alert">{{ $error }}</div>
    @endforeach
@endif

@if ( session('success_msg') )
    <div class="alert alert-success text-
center" role="alert">{!! session('success_msg') !!}</div>
@endif

<div class="form-group">
    <label>{{ __( 'Όνομα Χρήστη' ) }}</label>
    <input type="email" name="username" class="form-control" required />
</div>

<div class="form-group">
    <label>{{ __( 'Κωδικός' ) }}</label>
    <input type="password" name="password" class="form-control" required />
</div>
<div class="text-center">
    <button type="submit" class="btn btn-primary submit-
btn">{{ __( 'Εγγραφή' ) }}</button>
</div>

<div class="text-center bottom-frm-txt">Έχετε ήδη λογαριασμό;
    <a href="{{ route('login') }}">Είσοδος</a>
</div>
</form>

</body>
</html>

```

## ΠΑΡΑΡΤΗΜΑ C : Κώδικας μεθόδου process\_register

```

public function process_register(Request $request)
{
    if ( Auth::check() ){
        return redirect()->route('profile');
    }

    $customMessages = [
        'unique' => 'Το όνομα χρήστη υπάρχει ήδη. Επιλέξτε κάποιο άλλο.'
    ];

    $request->validate([
        'username' => 'required|unique:users',
        'password' => 'required|min:6'
    ], $customMessages);
}

```

```

$username = trim($request->input('username'));

$user = User::create([
    'username' => $username,
    'role' => 'customer',
    'password' => Hash::make($request->input('password')),
]);

if ( !empty($user) ){
    $request->session()-
>flash('success_msg', 'Ο λογαριασμός δημιουργήθηκε με επιτυχία. Μπορείτε να πρ
αγματοποιήσετε είσοδο. ');
    return redirect()->route('login');
}
return redirect()->route('auth.register');
}

```

## ΠΑΡΑΡΤΗΜΑ D : Κώδικας αρχείου view login

```

<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
<head>
    @include('inc/css')
    <meta charset="utf-8">
    <meta name="csrf-token" content="{{ csrf_token() }}">
    <title>{{ config('app.name') }} | Σύνδεση</title>
</head>
<body>

    @include('inc/menu')

    <div class="alert alert-primary text-center" role="alert">
        Είσοδος στο {{ config('app.name') }}
    </div>

    <form class="auth-form mt-5" method="post" action="{{ route('process-
login') }}">

        {{ csrf_field() }}

        @if ( $errors->any() )
            @foreach ($errors->all() as $error)
                <div class="alert alert-danger text-
center" role="alert">{{ $error }}</div>
            @endforeach
        @endif

```

```

@if ( session('error_msg') )
    <div class="alert alert-danger text-
center" role="alert">{!! session('error_msg') !!}</div>
@endif

@if ( session('success_msg') )
    <div class="alert alert-success text-
center" role="alert">{!! session('success_msg') !!}</div>
@endif

<div class="form-group">
    <label>{{ __( 'Όνομα Χρήστη' ) }}</label>
    <input type="email" name="username" class="form-
control" value="" required />
</div>

<div class="form-group">
    <label>{{ __( 'Κωδικός' ) }}</label>
    <input type="password" name="password" class="form-
control" value="" required />
</div>

<div class="text-center">
    <button type="submit" class="btn btn-primary submit-
btn">{{ __( 'Είσοδος' ) }}</button>
</div>

<div class=" bottom-frm-txt">

    <div class="text-center">Δεν έχετε λογαριασμό;
    <a href="{{ route('register') }}">Εγγραφή</a>
    </div>

</form>

</body>
</html>

```

## ΠΑΡΑΡΤΗΜΑ Ε : Κώδικας μεθόδου process login

```

public function process_login(Request $request)
{
    if ( Auth::check() ){
        return redirect()->route('profile');
    }
    $rules = [
        'username' => 'required',
        'password' => 'required'
    ];
}

```

```

];
$validator = Validator::make($request->except(['_token']), $rules);
if ( $validator->fails() ){
    return Redirect::to('login')->withErrors($validator);
}

$username = $request->input('username');
$password = $request->input('password');

$credentials = [
    'username' => $username,
    'password' => $password,
];
#dd($credentials);
if ( Auth::attempt($credentials) ){
    if ( Auth::user()->role == 'admin' ){
        return redirect()->route('products');
    } else{
        return redirect()->route('profile');
    }
} else{
    $request->session()-
>flash('error_msg', 'Λάθος στοιχεία εισόδου. ');
    return redirect()->back();
}
}
}

```

Η επεξεργασία του προφίλ δίνεται στο χρήστη από τη μέθοδο profile και το περιεχόμενο της σελίδας θα το βρούμε στο view auth.profile και φαίνεται παρακάτω:

```

<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <meta name="csrf-token" content="{{ csrf_token() }}" />
        <title>{{ config('app.name') }} | Προφίλ</title>
        @include('inc/css')
    </head>
    <body>
        @include('inc/menu')

        <div class="alert alert-primary text-center" role="alert">Προφίλ</div>

        @if ( session('error_msg') )
            <div class="alert alert-danger text-
center" role="alert">{!! session('error_msg') !!}</div>
        @endif

```

```

        @if ( session('success_msg') )
            <div class="alert alert-success text-
center" role="alert">{!! session('success_msg') !!}</div>
        @endif

        <form id="profile-form" method="POST" action="profile">
            {!! csrf_field() !!}

            <div class="text-right mb-3">
                <button type="submit" class="btn btn-success text-
right">Αποθήκευση</button>
            </div>

            <div class="alert alert-secondary text-
center" role="alert">Στοιχεία επικοινωνίας</div>

            <input class="form-control mt-
3" placeholder="Όνοματεπώνυμο" name="fullname" type="text" value="{{ $user['fu
llname'] }}" />
            <input class="form-control mt-
3" placeholder="Διεύθυνση" type="text" name="address" value="{{ $user['address
'] }}" />
            <input class="form-control mt-
3" placeholder="Τηλέφωνο" type="text" name="tel" value="{{ $user['tel'] }}" />

            <div class="alert alert-secondary text-center mt-
5" role="alert">Στοιχεία πληρωμής</div>

            <input class="form-control mt-
3" placeholder="Αριθμός κάρτας" type="number" name="card_number" value="{{ $us
er['card_number'] }}" />
            <input class="form-control mt-
3" placeholder="Όνομα κατόχου κάρτας" type="text" name="card_holder" value="{{
$user['card_holder'] }}" />
            <input class="form-control mt-
3" placeholder="Ημερομηνία λήξης (ΜΜ ΕΕ)" type="text" name="card_expiry" value
="{{ $user['card_expiry'] }}" />

        </form>

    </body>
</html>

```

## ΠΑΡΑΡΤΗΜΑ F : Κώδικας μεθόδου profile

```

public function profile(Request $request)
{
    if ( !Auth::check() ){

```

```

        $request->session()-
>flash('error_msg', 'Πραγματοποιήστε είσοδο πρώτα. ');
        return redirect()->route('login');
    }
    $user = User::where('id', Auth::user()->id)->first();

    $method = $request->method();
    if ( $method == 'POST' ){
        $user->fullname = $request->input('fullname');
        $user->address = $request->input('address');
        $user->tel = $request->input('tel');

        $user->card_number = $request->input('card_number');
        $user->card_holder = $request->input('card_holder');
        $user->card_expiry = $request->input('card_expiry');
        $user->save();
        $request->session()->flash('success_msg', 'Επιτυχής αποθήκευση. ');
    }
    return view('auth.profile', [
        'user' => $user,
    ]);
}

```

## ΠΑΡΑΡΤΗΜΑ G : Κώδικας menu εφαρμογής

```

@php
    $request = collect(request()->segments()->last());
@endphp
<nav class="navbar navbar-expand-lg navbar-light">
    <a class="navbar-brand" href="#">{{ config('app.name') }}</a>
    <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
        <div class="navbar-nav">
            <a class="nav-item nav-
link {{ $request == 'home' || $request == '' ? 'active' : '' }}" href="{{ rou
te('home') }}">Αρχική</a>
            <a class="nav-item nav-
link {{ $request == 'auction' ? 'active' : '' }}" href="{{ route('auction') }}"
">Δημοπρασίες</a>
            <a class="nav-item nav-
link {{ $request == 'search' ? 'active' : '' }}" href="{{ route('search-
auction') }}">Αναζήτηση</a>
            @if ( !Auth::check() )
                <a class="nav-item nav-link sign-
txt {{ $request == 'login' ? 'active' : '' }}" href="login">Σύνδεση</a>
            @else
                @if ( Auth::user()->role == 'admin' )

```

```

        <a class="nav-item nav-
link {{ $request == 'products' ? 'active' : '' }}" href="{{ route('products')
}}">Προϊόντα</a>
        <a class="nav-item nav-
link {{ $request == 'reports' ? 'active' : '' }}" href="{{ route('auction-
reports') }}">Αναφορές</a>
        @else
        <a class="nav-item nav-
link {{ $request == 'profile' ? 'active' : '' }}" href="{{ route('profile') }}"
">Προφίλ</a>
        @endif
        <span class="user-fullname">{{ Auth::user()->fullname }}</span>
        <a class="nav-item nav-link sign-txt" href="logout">Έξοδος</a>
    @endif
</div>
</div>
</nav>

```

## ΠΑΡΑΡΤΗΜΑ Η : Κώδικας διαχειριστικού

```

<?php if ( $totalPages > 0): ?>
    <nav aria-label="Page navigation">
        <ul class="pagination flex-wrap text-center">
            <li class="page-item"><a class="page-
link" href="<?=$currentPage > 1 ? '?currentPage='.( $currentPage-
1) : '#'?>">Προηγούμενη</a></li>
            <?php for($page = 1; $page <= $totalPages; $page++): ?>
                <li class="page-
item <?=$currentPage == $page ? 'active' : ''?>">
                    <a class="page-
link" href="?currentPage=<?=$page?>"><?=$page?></a>
                </li>
            <?php endfor; ?>
            <li class="page-item"><a class="page-
link" href="<?=$currentPage < $totalPages ? '?currentPage='.( $currentPage+1) :
'#'?>">Επόμενη</a></li>
        </ul>
    </nav>
<?php endif; ?>

$date = date('d-m-Y');
$time = time();
$last_time = ($time - ($time % (15 * 60)));
$in15mins = $last_time + (15 * 60);
$datetime = $date.' '.date('H:i:s', $in15mins);

```

Ο διαχειριστής μπορεί να ορίσει την ημερομηνία και ώρα της δημοπρασίας. Αυτό γίνεται με τη βοήθεια της βιβλιοθήκης του Bootstrap, τη datetimerpicker

(<https://www.malot.fr/bootstrap-datetimepicker/>), η οποία επιτρέπει στο χρήστη να επιλέξει με ένα φιλικό τρόπο την ημερομηνία και ώρα της δημοπρασίας. Βλέπουμε τον κώδικα που κάνουμε χρήση της βιβλιοθήκης:

```
$("#input.datetime-txt").datetimepicker({
    format: 'dd-mm-yyyy hh:ii:ss',
});
```

Από προεπιλογή, η ημερομηνία είναι η σημερινή και η ώρα είναι το επόμενο τέταρτο ή η τρέχουσα ώρα εφόσον είναι ακριβώς. Παρακάτω βλέπουμε το σώμα της μεθόδου products:

```
public function products(Request $request)
{
    if ( Auth::user()->role != 'admin' ){
        return redirect()->route('login');
    }

    $skip = 0;
    if ( $request->has('currentPage') ){
        $currentPage = (int) $request->currentPage;
        if ( $currentPage > 1 ){
            $skip = (($currentPage - 1) * self::$productsPageSize);
        }
    }
    $products = Product::
        with('auction')
        ->with('auction.user')
        ->with('category')
        ->skip($skip)
        ->take(self::$productsPageSize)
        ->get()
        ->toArray();
    #dump($request->currentPage);
    return view('products',
    [
        'page_size' => self::$productsPageSize,
        'products_count' => Product::all()->count(),
        'products' => $products,
        'categories' => Category::all(),
    ]);
}
```

Αξίζει να σημειωθεί ότι οι σχέσεις μεταξύ των πινάκων στη βάση δεδομένων (νοητά ξένα κλειδιά) μεταφράζονται σε αντικείμενα/μοντέλα μέσω του εργαλείου Eloquent που διαθέτει το Laravel. Για παράδειγμα η κατηγορία (πίνακας categories) στην οποία ανήκει ένα προϊόν (πίνακας products) θα ληφθεί με την εντολή with('category') και σε αυτή την περίπτωση θα γίνει ερώτηση στη βάση για join μεταξύ των 2 αυτών πινάκων. Τα μοντέλα που αφορούν τους πίνακες της βάσης δεδομένων ορίζονται στο φάκελο app με το κάθε μοντέλο να έχει ένα μοναδικό όνομα αρχείου και κλάσης. Η συσχέτιση μεταξύ των πινάκων ορίζεται σε αντίστοιχη

μέθοδο του μοντέλου, όπως για παράδειγμα η σχέση κατηγορίας και προϊόντος ορίζεται από τη σχέση `category` στο μοντέλο `Product`:

```
public function category()
{
    return $this->belongsTo('App\Category', 'category_id');
}
```

Το συγκεκριμένο είδος σχέσης είναι το `belongsTo` με την έννοια ότι το προϊόν ανήκει σε μια κατηγορία και εδώ οφείλουμε να ορίσουμε το πεδίο στον πίνακα `products` που σηματοδοτεί τη σχέση μεταξύ τους. Στο προηγούμενο αρχείο βλέπουμε και τη σχέση μεταξύ της δημοπρασίας και του προϊόντος. Εδώ η σχέση είναι τύπου `hasOne`, με την έννοια ότι ένα προϊόν έχει μόνο μια συσχετιζόμενη δημοπρασία:

```
public function auction()
{
    return $this->hasOne('App\Auction', 'product_id');
}
```

Τέλος, υπάρχει η δυνατότητα να ορίζουμε σχέσεις ανάμεσα στα μοντέλα με μεγαλύτερο βάθος εξάρτησης όπως στην περίπτωση `with('auction.user')`. Ένας χρήστης δεν έχει σχέση με ένα προϊόν, αλλά έχει σχέση με μια δημοπρασία και έτσι λέμε ότι έχει έμμεση σχέση με το προϊόν και αυτό συμβολίζεται με την τελεία ανάμεσα στην άμεση σχέση και την έμμεση σχέση. Αυτή η σχέση βοηθά το διαχειριστή να γνωρίζει το χρήστη στον οποίο κατοχυρώθηκε η δημοπρασία. Στον HTML πίνακα των προϊόντων στο αντίστοιχο view, υπάρχει μια περιοχή, η οποία δε φαίνεται αρχικά παρά μόνο όταν ο χρήστης πατήσει στο κουμπί εισαγωγή προϊόντος:

```
<tr id="new-product-row">
    <td></td>
    <td>
        <input name="image" type="file" />
    </td>
    <td>
        <input class="form-
control" type="text" placeholder="Όνομα" name="name" required />
    </td>
    <td>
        <input class="form-
control" type="number" min="0.01" step="0.01" max="999999" placeholder="Τιμή"
name="price" required />
    </td>
    <td>
        <select class="form-
control" name="category_id" required>
            @foreach($categories as $cat)
                <option value="{{ $cat['id'] }}">{{
{ $cat['name'] }}</option>
            @endforeach
        </select>
```

```

        </td>
        <td>
            <input class="form-
control" type="text" placeholder="Κατασκευαστής" name="manufacturer" required
/>
        </td>
        <td class="text-center">
            <button type="submit" class="btn btn-
success save-product-btn">Αποθήκευση</button>
        </td>
    </tr>

```

Όταν ο διαχειριστής κάνει κλικ σε αυτό το κουμπί, εμφανίζονται οι επιλογές για τη συμπλήρωση των πληροφοριών του προϊόντος προς εισαγωγή. Εκεί μπορεί να συμπληρώσει το όνομα, την τιμή και τον κατασκευαστή του προϊόντος. Επίσης μπορεί να επιλέξει την κατηγορία στην οποία ανήκει. Οι κατηγορίες βρίσκονται στη μεταβλητή `$categories`, και μέσω ενός βρόγχου `foreach` συμπληρώνεται δυναμικά το dropdown menu. Τέλος μπορεί να ανεβάσει μια εικόνα για το νέο προϊόν. Όταν ο χρήστης πατήσει στο κουμπί αποθήκευση θα σταλεί αίτημα στο URL `add/product` και η μέθοδος `addProduct` θα αναλάβει την επεξεργασία του αιτήματος. Αυτή θα λάβει τις τιμές που έχει εισάγει ο χρήστης και θα τις θέσει στις αντίστοιχες στήλες στον πίνακα των προϊόντων. Εάν ο χρήστης έχει επιλέξει εικόνα για το προϊόν, θα εκτελεστεί η μέθοδος `saveProductImage` και θα την αποθηκεύσει στο φάκελο `public/img` με όνομα το `id` του νέου προϊόντος που μόλις αποθηκεύτηκε. Παρακάτω βλέπουμε το σώμα της μεθόδου:

```

public function addProduct(Request $request)
{
    if ( Auth::user()->role != 'admin' ){
        return redirect()->route('login');
    }
    $product = new Product();
    $product->save();
    $product->name = $request->input('name');
    $product->price = (float) $request->input('price');
    $product->category_id = $request->input('category_id');
    $product->manufacturer = $request->input('manufacturer');
    $product->save();
    if ( $request->hasfile('image') ){
        $this->saveProductImage($request, $product->id);
    }
    return redirect()->back()-
>with('success_msg', 'Επιτυχής προσθήκη προϊόντος. ');
}

```

## ΠΑΡΑΡΤΗΜΑ Ι : Κώδικας εισαγωγής δημοπρασίας

```

$(document).on('click', 'button.add-product-to-auction-btn', function(){
    var btn = $(this);
    var parent = $(this).closest('tr');

```

```

var productId = $(this).attr('data-product-id');
var startDateTime = parent.find('input.datetime-txt').val();
var openingPrice = parent.find('input[type="number"]').val();
console.log(productId, startDateTime, openingPrice);
btn.attr('disabled', 'disabled');
//return;
$.ajax({
  data: {'product_id': productId, 'start_datetime': startDateTime, '
opening_price': openingPrice},
  type: 'POST',
  url: 'add/auction',
  dataType: 'json',
  success: function(data){
    console.log(data);
    if ( data.status == 'success' && data.message == true ){
      doToast('<a href="auction">Επιτυχής προσθήκη σε δημοπρασία</a>
', 'success');
      btn.closest('td.action-cell').html('<div class="alert alert-
info">Δημοπρασία<br/>'+startDateTime+'</div>');
    } else{
      doToast(data.message, 'error');
    }
  },
  error: function(error){
    console.error(error);
    btn.removeAttr('disabled');
    doToast(error, 'error');
  }, complete: function(data){
    btn.removeAttr('disabled');
  }
});
});
});

```

Το URL του αιτήματος είναι το add/auction και η μέθοδος που θα το επεξεργαστεί είναι η addAuction. Εκεί θα γίνουν οι απαραίτητοι έλεγχοι και εάν περάσουν, η δημοπρασία θα μπει στο σύστημα. Συγκεκριμένα, θα γίνουν έλεγχοι σχετικά με το αν δόθηκε έγκυρη τιμή έναρξης, έγκυρο προϊόν (με βάση το id του) και έγκυρη ημερομηνία. Στην περίπτωση που δοθεί άκυρη ημερομηνία, όπως για παράδειγμα αυτή είναι παρελθοντική, τότε θα επιστραφεί απάντηση με μήνυμα λάθους σε μορφή JSON από το server. Ο Javascript κώδικας πιο πάνω «καταλαβαίνει» τη μορφή JSON και εφόσον εντοπίσει λαθεμένη απάντηση (το status έχει τεθεί στο διακομιστή ως error), τότε ο client θα εμφανίσει ένα φιλικό μήνυμα λάθους με την εντολή doToast, το σώμα της οποίας βλέπουμε παρακάτω:

```

function doToast(message, type){
  $(' .toast-body').removeClass('alert-danger alert-success');
  $(' .toast-body').addClass(type == 'success' ? 'alert-success' : 'alert-
danger');
  $(' .toast-body').html(message);
  $(' .toast').toast('show');
}

```

```
}
```

Εδώ να σημειώσουμε ότι στο view products κάνουμε include το παρακάτω HTML που βρίσκεται στο view inc/toast:

```
<div class="toast" data-delay="10000" role="alert" aria-live="assertive" aria-atomic="true">
  <div class="toast-header">
    <strong class="mr-auto">{{ config('app.name') }}</strong>
    <button type="button" class="ml-2 mb-1 close" data-dismiss="toast" aria-label="Close">
      <span aria-hidden="true">&times;</span>
    </button>
  </div>
  <div class="toast-body"></div>
</div>
```

Πρόκειται για ένα container το οποίο είναι κρυμμένο αρχικά και συμπληρώνεται αναλόγως με το αντίστοιχο μήνυμα λάθους και CSS κλάσης και τελικά εμφανίζεται στο χρήστη για 10 δευτερόλεπτα πριν κρυφτεί και πάλι. Αυτό το container είναι ένα Bootstrap Toast (<https://getbootstrap.com/docs/4.3/components/toasts/>) και επιτρέπει την εμφάνιση σύντομων ενημερώσεων προς το χρήστη. Εάν η ενημέρωση είναι θετική, δηλαδή εμφανιστεί ένα μήνυμα επιτυχίας προς το χρήστη, τότε θα λάβει την κλάση alert-success (πράσινο χρώμα), αλλιώς σε αρνητικό μήνυμα ή μήνυμα λάθους, θα λάβει την κλάση alert-danger (κόκκινο χρώμα).

## ΠΑΡΑΡΤΗΜΑ J : Κώδικας κατοχύρωσης δημοπρασίας σε πελάτη

```
$(document).on('click', 'button.send-product-to-user-btn', function(){
  var btn = $(this);
  var auctionId = $(this).attr('data-auction-id');
  $.ajax({
    data: {'id': auctionId},
    type: 'POST',
    url: 'send/auction/user',
    dataType: 'json',
    success: function(data){
      console.log(data);
      if ( data.status == 'success' && data.message != '' ){
        doToast(data.message, 'success');
        btn.remove();
      } else{
        doToast(data.message, 'error');
      }
    },
    error: function(error){
      console.error(error);
    }
  });
});
```

```

    }
  });
})

```

## ΠΑΡΑΡΤΗΜΑ Κ : Κώδικας ενεργής δημοπρασίας

```

var end = $('#countdown').attr('data-end');
$('#countdown').countdown(end, function(event) {
  $(this).html(event.strftime('%H ώρες %M λεπτά %S δεπτερέλεπα'));
}).on('finish.countdown', function() {
  window.location.reload();
});

```

Τέλος, η ημερομηνία της επόμενης δημοπρασίας θα εμφανιστεί εάν υπάρχει με σύνδεσμο που οδηγεί προς αυτήν. Παρακάτω βλέπουμε το περιεχόμενο του εν λόγω view:

```

<div id="auctions-
container" class="{{ empty($active_auction) ? '' : 'active' }}">

    @include('inc/toast')

    @if ( !empty($active_auction) )
        <div class="alert alert-info text-
center" role="alert">Ενεργή δημοπρασία <br/> ({{ $active_auction['unique_id']
}})</div>

        <input type="hidden" id="auction-
id" value="{{ $active_auction['id'] }}" />

        <div class="row">
            <div class="col-sm-3">
                
            </div>

            <div class="col-sm-4">
                <b>{{ $active_auction['product']['name'] }}</b>
                <div class="row">
                    <div class="col mt-
3">Κατασκευαστής: {{ $active_auction['product']['manufacturer'] }}</div>
                    <div class="col mt-
3">Κατηγορία: {{ $active_auction['product']['category']['name'] }}</div>
                </div>

                <div class="row mt-5 alert alert-success text-center">
                    <div class="col-sm-5">
                        Προσφορές: {{ $active_auction['bid_count'] }}
                    <br/>

```

```

        Η προσφορά μου:
        &euro;<span id="my-
bid">{{ empty($active_auction['my_bid'][0]['price']) ? '' : $active_auction['m
y_bid'][0]['price'] }}</span>
    </div>
    <div class="col-sm-7">
        Τρέχουσα τιμή:
        &euro;<span id="current-price" data-opening-
price="{{ $active_auction['opening_price'] }}">
            {{ !empty($active_auction['max_bid'][0]['price
']) && $active_auction['max_bid'][0]['price'] > $active_auction['opening_price
'] ? $active_auction['max_bid'][0]['price'] : $active_auction['opening_price']
}}
        </span>
    </div>
</div>

    @if ( empty($active_auction['fruitless']) && empty($ac
tive_auction['winner_user_id']) )
        <div class="row mt-3">
            <input type="number" id="bid" class="form-
control mt-2" min="0.01" step="0.01" value="" max="999999" />
        </div>

        <div class="row mt-3">
            <div class="col">
                <button type="button" class="btn btn-
success bid-btn">Υποβολή</button>
            </div>
        </div>
    @else
        <div class="row mt-3">
            @if ( !empty($active_auction['fruitless']) )
                <div class="alert alert-danger text-
center" role="alert">Άγονη δημοπρασία</div>
            @elseif ( !empty($active_auction['winner_user_
id']) )
                <div class="alert alert-success text-
center" role="alert">Η δημοπρασία κατακυρώθηκε στο χρήστη #{{ $active_auction[
'winner_user_id'] }}</div>
            @endif
        </div>
    @endif
</div>

    <div class="col-sm-2">
        Έναρξη: <br/>
        {{ date('d-m-
Y H:i:s', strtotime($active_auction['start_datetime'])) }}

```

```

        </div>

        <div class="col-sm-3">
            Λήξη: <br/>
            {{ date('d-m-
Y H:i:s', strtotime($active_auction['end_datetime'])) }}
            @if ( $countdown === true && empty($active_auction['fr
uitless']) )
                <div id="countdown" class="alert alert-
warning text-center mt-3" data-
end="{{ { $active_auction['end_datetime'] } }}"></div>
            @endif
        </div>
    </div>

    @else
        <div class="row">
            <div class="alert alert-warning text-center" role="alert">
                Δεν υπάρχει ενεργή δημοπρασία.
                @if ( !empty($next_auction) )
                    @php
                        $nextAuctionShown = true;
                    @endphp
                    Επόμενη δημοπρασία:
                    <a href="{{ { route('auction-
single', $next_auction['id']) } }}">
                        {{ date('d-m-
Y H:i:s', strtotime($next_auction['start_datetime'])) }}
                    </a>
                @endif
            </div>
        </div>
    @endif

    @if ( $nextAuctionShown === false )
        <div class="row next-auction-wrap">
            <div class="alert alert-secondary text-
center" role="alert">
                @if ( !empty($next_auction) )
                    Επόμενη δημοπρασία:
                    <a href="{{ { route('auction-
single', $next_auction['id']) } }}">
                        {{ date('d-m-
Y H:i:s', strtotime($next_auction['start_datetime'])) }}
                    </a>
                @endif
            </div>
        </div>
    @endif

```

```
</div>
```

## ΠΑΡΑΡΤΗΜΑ Λ : Κώδικας ενεργής προσφοράς

```
function bid(){
    var url = 'bid/auction';
    if ( baseName(window.location.href) != 'auction' ){
        url = '../bid/auction';
    }
    var price = $('#bid').val();
    if ( price === '' ){
        doToast('Παρακαλώ εισάγετε μία τιμή', 'error');
        return;
    }
    $.ajax({
        data: {'id': $('#auction-id').val(), 'price': price},
        type: 'POST',
        url: url,
        dataType: 'json',
        success: function(data){
            console.log(data);
            if ( data.status == 'success' && data.message != '' ){
                updating(data);
            } else{
                doToast(data.message, 'error');
            }
        },
        error: function(error){
            console.error(error);
        }
    });
}
```

Τη σκντάλη θα την πάρει η μέθοδος `bidAuction` στο `MyController`, η οποία θα ελέγξει για έγκυρο `id` δημοπρασίας και έγκυρη προσφορά. Με τη βοήθεια της μεθόδου `isBidForAuctionValid` θα γίνει έλεγχος εάν έχει παρέλθει η ημερομηνία λήξης της δημοπρασίας και φυσικά δεν είναι ούτε άγονη ούτε κατοχυρωμένη. Εάν επιστραφεί `false`, τότε ο χρήστης δεν μπορεί να βάλει προσφορά καθώς πρόκειται για παρελθοντική δημοπρασία και αντίστοιχο μήνυμα λάθους θα εμφανιστεί. Αλλιώς εάν η δημοπρασία είναι έγκυρη, θα προστεθεί η προσφορά στον πίνακα `bids` και θα επιστραφεί ως μήνυμα το αποτέλεσμα της μεθόδου `getMaxBidByAuctionId`, η οποία θα αναζητήσει τη μεγαλύτερη προσφορά για τη δημοπρασία και θα συμπεριληφθεί η μέγιστη προσφορά του χρήστη. Παρακάτω βλέπουμε το σώμα της μεθόδου `bidAuction`:

```
public function bidAuction(Request $request)
{
    $auctionId = (int) $request->input('id');
```

```

    $price = $request->input('price');
    if ( is_numeric($auctionId) && is_numeric($price) ){
        if ( $this->isBidForAuctionValid($auctionId) === false ){
            return response()-
>json(['status' => 'error', 'message' => 'Η δημοπρασία είναι παρελθοντική.']);
        }
        $bid = new Bid();
        $bid->user_id = Auth::user()->id;
        $bid->price = (float) $price;
        $bid->auction_id = $auctionId;
        if ( $bid->save() ){
            return response()-
>json(['status' => 'success', 'message' => $this-
>getMaxBidByAuctionId($auctionId)]);
        }
    }
}
}

```

Επιστρέφουμε στον κώδικα Javascript, με το πέρας του αιτήματος θα γίνει κλήση στη μέθοδο updating και θα ελέγξει εάν τη μέγιστη προσφορά είναι μεγαλύτερη από την τιμή έναρξης, οπότε θα εμφανιστεί αυτή. Επίσης θα εμφανιστεί η μέγιστη προσφορά του χρήστη. Η τρέχουσα τιμή του προϊόντος θα λάβει την κλάση updating (ορισμένη στο custom.css) και για 2 δευτερόλεπτα θα αναβοσβήνει ένα γκρι χρώμα παρασκηνίου για να δώσει την αίσθηση στο χρήστη ότι το σύστημα «πήρε» την προσφορά του. Ύστερα από 2 δευτερόλεπτα η συγκεκριμένη κλάση θα αφαιρεθεί. Παρακάτω βλέπουμε το σώμα της άνω μεθόδου:

```

function updating(data){
    var opening_price = parseFloat($('#current-price').attr('data-opening-
price'));
    // MAX BID
    if ( data.message.max_bid !== undefined && data.message.max_bid[0].price
!==' ' && $.isNumeric(data.message.max_bid[0].price) && parseFloat(data.messa
ge.max_bid[0].price) > opening_price ){
        $('#current-price').text(data.message.max_bid[0].price);
    } else{
        $('#current-price').text(opening_price);
    }
    // MY BID
    if ( data.message.my_bid[0]['price'] !== undefined && data.message.my_bi
d[0]['price'] !== ' ' && $.isNumeric(data.message.my_bid[0]['price']) ){
        $('#my-bid').text(data.message.my_bid[0]['price']);
    }
    $('#current-price').addClass('updating');
    setTimeout(function(){ $('#current-
price').removeClass('updating');}, 2000);
}

```

## ΠΑΡΑΡΤΗΜΑ Μ : Κώδικας αυτόματης ενημέρωσης δημοπρασίας

```
function updateAuction(){
  var auctionId = $('#auction-id').val();
  $.ajax({
    data: {'id': auctionId},
    type: 'POST',
    url: 'update/auction',
    dataType: 'json',
    success: function(data){
      console.log(data);
      if ( data.status == 'finish' ){
        //window.location.reload();
        window.location.href = 'auction/'+auctionId;
      }
      else if ( data.status == 'success' && data.message != '' ){
        updating(data);
      }
    },
    error: function(error){
      console.error(error);
    }
  });
}
```

Ακολουθώντας στο server, βλέπουμε το σώμα της μεθόδου checkAuctionStatus, δηλαδή το ουσιαστικό περιεχόμενο της μεθόδου updateAuction:

```
private function checkAuctionStatus($auctionId)
{
  $now = Carbon::now();
  $auction = Auction::where('id', $auctionId)
->with('latest_bid')
->with('max_bid')
->first();
  if ( !empty($auction) ){
    if ( !empty($auction['latest_bid'][0]['created_at']) ){
      $totalDurationSeconds = $now->diffInSeconds(Carbon::createFromFormat('Y-m-d H:i:s', $auction['latest_bid'][0]['created_at'], 'Europe/Athens'));
      if ( $totalDurationSeconds >= self::$noBidAuctionInSeconds ){
        #WE HAVE A WINNER
        if ( $auction['max_bid'][0]['price'] > $auction['opening_price'] ){
          $auction['winner_user_id'] = $auction['max_bid'][0]['user_id'];
          $auction['won'] = $now->format('Y-m-d H:i:s');
```

```

        if ( $auction->save() ){
            return 'Η δημοπρασία κατακυρώθηκε στο χρήστη #' . $a
uction['max_bid'][0]['user_id'];
        }
    }
    #FRUITLESS
    else{
        $auction['fruitless'] = $now->format('Y-m-d H:i:s');
        if ( $auction->save() ){
            return 'Η δημοπρασία είναι άγονη.';
        }
    }
}
}
$totalDurationSeconds = $now-
>diffInSeconds(Carbon::createFromFormat('Y-m-
d H:i:s', $auction['start_datetime'], 'Europe/Athens'));
#FRUITLESS
if ( $totalDurationSeconds >= self::$completelyFruitelessAuctionIn
Seconds ){
    $auction['fruitless'] = $now->format('Y-m-d H:i:s');
    if ( $auction->save() ){
        return 'Η δημοπρασία είναι άγονη.';
    }
}
}
return null;
}
}

```

## ΠΑΡΑΡΤΗΜΑ Ν : Κώδικας εμφάνισης δημοπρασίας με βάση το id

```

public function singleAuction(Request $request)
{
    if ( empty($request->id) || !is_numeric($request->id) ){
        return redirect()->back();
    }
    $auctionId = (int) $request->id;
    $auction = Auction::where('id', $request->id)
->with('product')
->with('product.category')
->with(['max_bid' => function($query){ $query->take(1); }])
->with('my_bid')
->withCount('bid')
->first();
    $active_auction = $this->getActiveAuction($this-
>getNextCandidateAuctions(), $auctionId);
}

```

```

#dump($active_auction);
return view('single-auction',
[
    'not_found' => empty($auction) ? $auctionId : null,
    'is_active' => empty($active_auction) ? false : true,
    'countdown' => false,
    'auction' => $auction,
]);
}
@if ( !empty($not_found) )
    <div class="alert alert-danger text-
center" role="alert">Δε βρέθηκε η δημοπρασία με ID: {{ $not_found }} </div>
@else

    <div id="auctions-container" class="active">

        @include('inc/toast')

        @if ( !empty($auction) )
            <div class="alert alert-info text-
center" role="alert">Δημοπρασία <br/> ({{ $auction['unique_id'] }})</div>

            <input type="hidden" id="auction-
id" value="{{ $auction['id'] }}" />

            @if ( $is_active === true )
                <div class="alert alert-warning text-center" role="alert">
                    <a href="{{ route('auction') }}">
                        Η δημοπρασία είναι ενεργή
                    </a>
                </div>
            @endif

            <div class="row">
                <div class="col-sm-3">
                    
                </div>

                <div class="col-sm-4">
                    <b>{{ $auction['product']['name'] }}</b>
                    <div class="row">
                        <div class="col mt-
3">Κατασκευαστής: {{ $auction['product']['manufacturer'] }}</div>
                        <div class="col mt-
3">Κατηγορία: {{ $auction['product']['category']['name'] }}</div>
                    </div>

                    <div class="row mt-5 alert alert-success text-center">

```

```

        <div class="col-sm-5">
            Προσφορές: {{ $auction['bid_count'] }}
            <br/>
            Η προσφορά μου:
            &euro;<span id="my-
bid">{{ empty($auction['my_bid'][0]['price']) ? '' : $auction['my_bid'][0]['pr
ice'] }}</span>
        </div>
        <div class="col-sm-7">
            Τρέχουσα τιμή:
            &euro;<span id="current-price" data-opening-
price="{{ $auction['opening_price'] }}">
                {{ !empty($auction['max_bid'][0]['price']) &&
$auction['max_bid'][0]['price'] > $auction['opening_price'] ? $auction['max_bi
d'][0]['price'] : $auction['opening_price'] }}
            </span>
        </div>
    </div>

    @if ( empty($auction['fruitless']) && empty($auction['
winner_user_id'] ) )
        <div class="row mt-3">
            <input type="number" id="bid" class="form-
control mt-2" min="0.01" step="0.01" value="" max="999999" />
        </div>

        <div class="row mt-3">
            <div class="col">
                <button type="button" class="btn btn-
success bid-btn">Υποβολή</button>
            </div>
        </div>
    @else
        <div class="row mt-3">
            @if ( !empty($auction['fruitless']) )
                <div class="alert alert-danger text-
center" role="alert">Άγωνα δημοπρασία</div>
            @elseif ( !empty($auction['winner_user_id']) )
                <div class="alert alert-success text-
center" role="alert">Η δημοπρασία κατακυρώθηκε στο χρήστη #{{ $auction['winner
_user_id'] }}</div>
            @endif
        </div>
    @endif
</div>

<div class="col-sm-2">
    Έναρξη: <br/>

```

```

        {{ date('d-m-
Y H:i:s', strtotime($auction['start_datetime'])) }}
    </div>

    <div class="col-sm-3">
        Λήξη: <br/>
        {{ date('d-m-
Y H:i:s', strtotime($auction['end_datetime'])) }}
        @if ( $countdown === true && empty($auction['fruitless
']) )
            <div id="countdown" class="alert alert-
warning text-center mt-3" data-end="{{ $auction['end_datetime'] }}"></div>
        @endif
    </div>
</div>
@endif
</div>
@endif

```

## ΠΑΡΑΡΤΗΜΑ Ο : Κώδικας αναζήτησης δημοπρασίας

```

public function searchAuction(Request $request)
{
    $searchManufacturer = $searchCategory = $searchQuery = '';
    $auctions = [];
    $products = Product::all();
    $manufacturers = [];
    if ( !empty($products) ){
        foreach($products as $p){
            $manufacturers[] = $p['manufacturer'];
        }
    }
    $manufacturers = array_unique($manufacturers);
    $method = $request->method();
    if ( $method == 'POST' ){
        $searchQuery = trim($request->input('query'));
        $query = Product::query();
        if ( !empty($searchQuery) ){
            $query->where(function ( $query ) use($searchQuery) {
                $query
                ->orWhere('name', 'LIKE', '%'.$searchQuery.'%')
                ->orWhere('name', 'LIKE', $searchQuery.'%')
                ->orWhere('manufacturer', 'LIKE', '%'.$searchQuery.'%')
                ->orWhere('manufacturer', 'LIKE', $searchQuery.'%');
            });
        }
    }
}

```

```

    if ( !empty($request->input('category_id')) ){
        $searchCategory = $request->input('category_id');
        $query->where('category_id', $searchCategory);
    }
    if ( !empty($request->input('manufacturer')) ){
        $searchManufacturer = $request->input('manufacturer');
        $query->where('manufacturer', $searchManufacturer);
    }
    $query
    ->with('auction')
    ->with('auction.product')
    ->with('auction.max_bid')
    ->with('auction.product.category');
    $products = $query->get()->toArray();
    if ( !empty($products) ){
        foreach($products as $p){
            if ( !empty($p['auction']) ){
                $auctions[] = $p['auction'];
            }
        }
    }
}
#dump($auctions);
return view('search-auction',
[
    'method' => $method,
    'search_query' => $searchQuery,
    'search_category' => $searchCategory,
    'search_manufacturer' => $searchManufacturer,
    'auctions' => $auctions,
    'manufacturers' => $manufacturers,
    'categories' => Category::all(),
]);
}

```

Στο view θα εμφανιστούν τα προϊόντα σε πίνακα. Οι πληροφορίες που προβάλλονται είναι η τιμή, το όνομα, ο κατασκευαστής, η κατηγορία του προϊόντος, η ημερομηνία έναρξης της δημοπρασίας και το εάν ήταν άγωνα ή κατοχυρωμένη (και σε ποιο id χρήστη κατοχυρώθηκε. Ακολουθεί ο κώδικας του view search-auction.blade:

```

<form method="POST" action="">
    {!! csrf_field() !!}
    <div class="row">
        <div class="col-sm-2 offset-sm-2">
            <select class="form-control" name="category_id">
                <option value="">Επιλέξτε κατηγορία</option>
                @foreach($categories as $cat)
                    <option value="{{ $cat['id'] }}" {{ $cat['id'] ==
                    $search_category ? 'selected="selected"' : '' }}>{{ $cat['name'] }}</option>

```

```

        @endforeach
    </select>
</div>

<div class="col-sm-2">
    <select class="form-control" name="manufacturer">
        <option value="">Επιλέξτε κατασκευαστή</option>
        @foreach($manufacturers as $man)
            <option value="{{ $man }}" {{ $man == $search_manu
facturer ? 'selected="selected"' : '' }}>{{ $man }}</option>
        @endforeach
    </select>
</div>

<div class="col-sm-3">
    <input type="text" class="form-
control" name="query" placeholder="Αναζητήστε προϊόν ή κατασκευαστή" value="{{
empty($query) ? '' : $query }}" />
</div>

<div class="col-sm-3">
    <button type="submit" class="btn btn-
success">Αναζήτηση</button>
</div>
</div>

@if ( $method == 'POST' )

    @if ( !empty($auctions) )
        <div class="row justify-content-center mt-5">
            <div class="col-auto">
                <table class="products-table table table-
responsive table-striped table-bordered table-hover">
                    <thead class="thead-dark">
                        <tr>
                            <th>#</th>
                            <th>Εικόνα</th>
                            <th>Όνομα</th>
                            <th>Τιμή</th>
                            <th>Κατηγορία</th>
                            <th>Κατασκευαστής</th>
                            <th>Δημοπρασία</th>
                        </tr>
                    </thead>
                    <tbody>
                        @foreach($auctions as $k=>$auction)
                            <tr>
                                <td>{{ $k + 1 }}</td>
                                <td class="text-center">

```

```

                                
                                </td>
                                <td>{{ $auction['product']['name']
}}</td>
                                <td>&euro; {{ $auction['product']['
'price'] }}</td>
                                <td>{{ $auction['product']['catego
ry']['name'] }}</td>
                                <td>{{ $auction['product']['manufa
cturer'] }}</td>
                                <td>
                                    {{ date('d-m-
Y H:i:s', strtotime($auction['start_datetime'])) }}
                                    @if ( !empty($auction['winner_
user_id']) )
                                        <div class="alert alert-
success text-center mt-3" role="alert">
                                            Κατοχύρωση &euro; {{ $auct
ion['max_bid'][0]['price'] }} στο χρήστη #{{ $auction['winner_user_id'] }}
                                        </div>
                                    @else
                                        <div class="alert alert-
danger text-center mt-3" role="alert">Άγωνα</div>
                                    @endif
                                </td>
                                </tr>
                                @endforeach
                            </tbody>
                        </table>
                    </div>
                </div>

                @else
                    <div class="alert alert-danger text-center mt-
5" role="alert">Δε βρέθηκαν δημοπρασίες</div>
                @endif

            @endif
        </form>

```

## ΠΑΡΑΡΤΗΜΑ Ρ : Κώδικας αναφορών δημοπρασιών

```

private function getAverageAuctionTimeInSeconds($allAuctions)
{
    $auctionTimes = [];
    if ( !empty($allAuctions) ){
        foreach($allAuctions as $auction){

```

```

        $startDateCarbon = Carbon::createFromFormat('Y-m-
d H:i:s', $auction['start_datetime'], 'Europe/Athens');
        if ( !empty($auction['won']) ){
            $endDateCarbon = Carbon::createFromFormat('Y-m-
d H:i:s', $auction['won'], 'Europe/Athens');
            $auctionTimes[] = $endDateCarbon-
>diffInSeconds($startDateCarbon);
        } else if ( empty($auction['fruitless']) ){
            $auctionTimes[] = $auction['auto_fruitless_seconds'];
        } else{
            $endDateCarbon = Carbon::createFromFormat('Y-m-
d H:i:s', $auction['fruitless'], 'Europe/Athens');
            $auctionTimes[] = $endDateCarbon-
>diffInSeconds($startDateCarbon);
        }
    }
}
#dump($auctionTimes);
$auctionTimes = array_filter($auctionTimes);
if ( count($auctionTimes) ) {
    $averageAuctionTime = 'Μέσος χρόνος δημοπρασίας: '.number_format(a
rray_sum($auctionTimes) / count($auctionTimes), 2).' δεπτερόλεπτα (από '.count
($allAuctions).')';
}
return $averageAuctionTime;
}
}

```

```

public function auctionReports()
{
    if ( Auth::user()->role != 'admin' ){
        return redirect()->route('login');
    }
    $allAuctions = Auction::all();
    $fruitlessAuctions = Auction::whereNull('winner_user_id')
->with('product')
->with('max_bid')
->get()
->toArray();
    $fruitlessAuctionsCount = 'Άγονες δημοπρασίες: '.count($fruitlessAucti
ons);
    return view('auction-reports',
    [
        'averageAuctionTime' => $this-
>getAverageAuctionTimeInSeconds($allAuctions),
        'fruitlessAuctionsCount' => $fruitlessAuctionsCount,
        'fruitlessAuctions' => $fruitlessAuctions,
    ]);
}
}

```

Στο view (auction-reports.blade) θα εμφανιστεί ο αριθμός των συνολικών άγονων δημοπρασιών, ο μέσος χρόνος των δημοπρασιών (μαζί με το σύνολο αυτών και σε έναν πίνακα θα εμφανιστούν οι πληροφορίες όλων των άγονων δημοπρασιών (τιμή, εικόνα, όνομα, εναρκτήρια τιμή, μέγιστη προσφορά και την ημερομηνία που εκκίνησε η εκάστοτε δημοπρασία. Το σώμα του view θα φανεί παρακάτω:

```
@if ( !empty($fruitlessAuctionsCount) )
    <div class="alert alert-danger text-center" role="alert">{{ $fruitlessAuctionsCount }}</div>
@endif

@if ( !empty($averageAuctionTime) )
    <div class="alert alert-info text-center mt-3" role="alert">{{ $averageAuctionTime }}</div>
@endif

@if ( !empty($fruitlessAuctions) )

    <div class="alert alert-warning text-center" role="alert">Προϊόντα άγονων δημοπρασιών</div>

    <div class="row justify-content-center mt-3">
        <div class="col-auto">
            <table class="products-table table table-responsive table-striped table-bordered table-hover">
                <thead class="thead-dark">
                    <tr>
                        <th>#</th>
                        <th>Εικόνα</th>
                        <th>Όνομα</th>
                        <th>Εναρκτήρια τιμή</th>
                        <th>Μέγιστη προσφορά</th>
                        <th>Δημοπρασία</th>
                    </tr>
                </thead>
                <tbody>
                    @foreach($fruitlessAuctions as $k=>$auction)
                        <tr>
                            <td>{{ $k + 1 }}</td>
                            <td class="text-center">
                                
                            </td>
                            <td>{{ $auction['product']['name'] }}</td>
                            <td>&euro; {{ $auction['opening_price'] }}</td>
                        </tr>
                    @endforeach
                </tbody>
            </table>
        </div>
    </div>

```

```
                <td>
                    {!! empty($auction['max_bid'][0]) ? ''
: '&euro; ' . $auction['max_bid'][0]['price'] !!}
                </td>
                <td>
                    {{ date('d-m-
Y H:i:s', strtotime($auction['start_datetime'])) }}
                </td>
            </tr>
        @endforeach
    </tbody>
</table>
</div>
</div>
@endif
```