



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πτυχιακή Εργασία

Τίτλος Πτυχιακής Εργασίας	Εφαρμογή Δωρεών με Χρήση SpringBoot για το backend, Angular για το frontend και MySQL για την βάση δεδομένων Donation Application using SpringBoot for the backend, Angular for the frontend and MySQL for the database
Όνοματεπώνυμο Φοιτητή	Αλέξανδρος Παλατσίδης
Πατρώνυμο	Μιχαήλ Παλατσίδης
Αριθμός Μητρώου	Π19128
Επιβλέπων	Ευθύμιος Αλέπης

Ημερομηνία Παράδοσης

Σεπτέμβριος 2024

Copyright ©

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πειραιώς.

Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.

Ευχαριστίες

Η ολοκλήρωση αυτής της πτυχιακής δεν θα ήταν εφικτή χωρίς την καθοδήγηση και βοήθεια ορισμένων ατόμων, τα οποία θα ήθελα να ευχαριστήσω.

Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, Ευθύμιο Αλέπη, ο οποίος ξεχώρισε στα μάτια μου ως καθηγητής κατά την διάρκεια των σπουδών μου και είχε σημαντικό ρόλο στην πτυχιακή μου εργασία με τις άμεσες και σαφείς απαντήσεις του στις απορίες μου.

Επίσης, θα ήθελα να ευχαριστήσω τους συναδέλφους μου και την ακαδημαϊκή κοινότητα που με στήριξαν με την γνώση και την συνεργασία μας καθ' όλη την διάρκεια των σπουδών μου. Η βοήθεια τους ήταν καθοριστική στην ολοκλήρωση των σπουδών μου.

Τέλος ευχαριστώ βαθιά την οικογένεια μου και ιδιαίτερα την μητέρα μου που με υποστήριξαν στις σπουδές μου ψυχολογικά όταν οι καταστάσεις γίνονταν απαιτητικές.

Περίληψη

Η συγκεκριμένη πτυχιακή επικεντρώνεται στην υλοποίηση μιας διαδικτυακής εφαρμογής με σκοπό να βοηθήσει τους ανθρώπους να κάνουν δωρεές αντικειμένων. Αυτό σημαίνει ότι κάποιος θα μπορεί εύκολα να αναρτήσει αντικείμενα που επιθυμεί να κάνει δωρεά αλλά και να επιλέξει αντικείμενα που άλλοι άνθρωποι έχουν αναρτήσει προς δωρεά. Στα πλαίσια της υλοποίησης αυτής της πλατφόρμας χρησιμοποιήθηκαν τεχνολογίες backend (Spring Boot σε Java), frontend (Angular σε TypeScript, Angular Material) και βάση δεδομένων (MySQL), που επιτρέπουν την εύκολη και γρήγορη ανάπτυξη ιστοσελίδων/εφαρμογών με το Client-Server Model. Χρησιμοποιήθηκε η Agile μεθοδολογία για την ευελιξία της και τα αποτελέσματα δείχνουν ότι ο κώδικας είναι λειτουργικός και αποδοτικός.

Επιστημονική Περιοχή: Πληροφορική / Ανάπτυξη Λογισμικού

Λέξεις Κλειδιά: Angular, Spring Boot, Κώδικας, Backend, Frontend

Abstract

This thesis focuses on the implementation of an online application aimed at helping people donate items. This means that individuals will be able to easily post items they wish to donate and select items that others have posted for donation. In the context of implementing this platform, backend technologies (Spring Boot in Java), frontend technologies (Angular in TypeScript, Angular Material), and a database (MySQL) were used, allowing for easy and rapid development of websites/applications using the Client-Server Model. The Agile methodology was employed for its flexibility, and the results show that the code is functional and efficient.

Scientific Area: Information Technology / Software Development

Key Words: Angular, Spring Boot, Code, Backend, Frontend

Πίνακας Περιεχομένων

Copyright ©.....	i
Ευχαριστίες.....	ii
Περίληψη.....	iii
Abstract.....	iii
Πίνακας Περιεχομένων.....	iv
Κατάλογος Εικόνων	vi
Κατάλογος Πινάκων	viii
Κατάλογος Διαγραμμάτων.....	ix
Εισαγωγή.....	1
1. Χρήση Εφαρμογής.....	2
1.1 Εισαγωγή.....	2
1.2 Σελίδες Εφαρμογής	2
1.2.1 Login	2
1.2.2 Sign Up.....	4
1.2.3 Home	7
1.3 Παράθυρα Εφαρμογής.....	8
1.3.1 Edit User.....	8
1.3.2 Picked Items.....	10
1.3.3 Posted Items	12
1.3.4 Add Item	13
1.3.5 Edit Item.....	15
2. Frontend.....	17
2.1 Εισαγωγή.....	17
2.2 Τεχνολογίες που χρησιμοποιήθηκαν	17
2.2.1 Angular.....	17
2.2.2 Τρόπος εκκίνησης του server.....	18
2.2.3 Angular Material	18
2.3 Ασφάλεια	18
2.4 Services(Υπηρεσίες)	21
2.4.1 Εισαγωγή	21
2.4.2 authenticateUser	22
2.4.3 registerUser	22
2.4.4 logout.....	23

2.4.5	getUserDetails.....	23
2.4.6	editUserDetails	23
2.4.7	deleteUser	23
2.4.8	addItem.....	23
2.4.9	getItem	23
2.4.10	pickItem	23
2.4.11	deselectItem	24
2.4.12	deleteItem	24
2.4.13	editItem	24
2.4.14	getPickedItems.....	24
2.4.15	getPostedItems	24
2.5	Μορφοποίηση Εφαρμογής	27
2.5.1	Εισαγωγή	27
2.5.2	Κάρτες.....	27
2.5.3	Κουμπιά	28
2.5.4	Toolbar	28
2.5.5	Paginator(Εργαλείο σελιδοποίησης)	28
2.5.6	Toasts	28
2.5.7	Εικόνες και Icons(Εικονίδια)	29
3.	Backend	30
3.1	Εισαγωγή.....	30
3.2	Ασφάλεια	30
3.2.1	JWT tokens.....	30
3.2.2	Κρυπτογράφηση κωδίκων και συμπίεση εικόνων	31
4.	Βάση Δεδομένων.....	32
4.1	Εισαγωγή.....	32
4.2	Δομή της βάσης δεδομένων	32
	Συμπεράσματα.....	34
	Πίνακας Ορολογίας	35
	Πίνακας Συντημήσεων-Αρτικόμενων-Ακρονημάτων	37
	Βιβλιογραφία.....	38

Κατάλογος Εικόνων

ΕΙΚΟΝΑ 1: Login σελίδα	2
ΕΙΚΟΝΑ 2: Εικονίδιο ματιού στο πεδίο password για να μην εμφανίζεται το password.....	3
ΕΙΚΟΝΑ 3: Εικονίδιο ματιού στο πεδίο password για να εμφανίζεται το password.....	3
ΕΙΚΟΝΑ 4: Login σελίδα με tooltip(Συμβουλή εργαλείου).....	3
ΕΙΚΟΝΑ 5: Login σελίδα με error messages(Μηνύματα σφάλματος) και tooltip	4
ΕΙΚΟΝΑ 6: Sign Up σελίδα	5
ΕΙΚΟΝΑ 7: Παράθυρο επιλογής εικόνας.....	5
ΕΙΚΟΝΑ 8: Sign Up σελίδα με επιλεγμένη εικόνα προφίλ.....	6
ΕΙΚΟΝΑ 9: Sign Up σελίδα με tooltip	6
ΕΙΚΟΝΑ 10: Sign Up σελίδα με error messages και tooltip	7
ΕΙΚΟΝΑ 11: Γραμμή εργαλείων στην σελίδα Home.....	7
ΕΙΚΟΝΑ 12: Γραμμή πλοήγησης μεταξύ αντικειμένων προς δωρεά στην σελίδα Home	7
ΕΙΚΟΝΑ 13: Home σελίδα.....	8
ΕΙΚΟΝΑ 14: Edit User κουμπί με ανθρωπάκι στην σελίδα Home.....	9
ΕΙΚΟΝΑ 15: Edit User κουμπί με εικόνα προφίλ στην σελίδα Home	9
ΕΙΚΟΝΑ 16: Edit User παράθυρο	9
ΕΙΚΟΝΑ 17: Edit User παράθυρο με error message και tooltip	10
ΕΙΚΟΝΑ 18: Picked Item κουμπί με καρδιά στην σελίδα Home	10
ΕΙΚΟΝΑ 19: Picked Items παράθυρο με επιλεγμένα αντικείμενα	11
ΕΙΚΟΝΑ 20: Picked Items παράθυρο χωρίς επιλεγμένα αντικείμενα.....	11
ΕΙΚΟΝΑ 21: Γραμμή πλοήγησης μεταξύ επιλεγμένων αντικειμένων στο παράθυρο Picked Items ..	12
ΕΙΚΟΝΑ 22: Posted Items κουμπί με εικονίδιο φακέλου στην σελίδα Home	12
ΕΙΚΟΝΑ 23: Posted Items παράθυρο	12
ΕΙΚΟΝΑ 24: Γραμμή πλοήγησης μεταξύ επιλεγμένων αντικειμένων στο παράθυρο Posted Items ..	13
ΕΙΚΟΝΑ 25: Add Item κουμπί με εικονίδιο σταυρού στην σελίδα Home.....	13
ΕΙΚΟΝΑ 26: Add Item παράθυρο.....	13
ΕΙΚΟΝΑ 27: Add Item παράθυρο με tooltip	14
ΕΙΚΟΝΑ 28: Add Item με error messages και tooltip	14
ΕΙΚΟΝΑ 29: Edit Item κουμπί στο παράθυρο Posted Items	15
ΕΙΚΟΝΑ 30: Edit Item παράθυρο.....	15
ΕΙΚΟΝΑ 31: Edit Item παράθυρο με tooltip	16
ΕΙΚΟΝΑ 32: Εκτέλεση ng serve -o για την εκκίνηση του frontend	18
ΕΙΚΟΝΑ 33: Παράδειγμα JWT token	19
ΕΙΚΟΝΑ 34: Interceptor στο frontend.....	20
ΕΙΚΟΝΑ 35: Interceptor στο αρχείο app.config.ts.....	21
ΕΙΚΟΝΑ 36: Import του Service αρχείου στο Login component	21
ΕΙΚΟΝΑ 37: Αναφορά του Service στο constructor του Login component	22
ΕΙΚΟΝΑ 38: Πρώτο Μέρος Services στο frontend.....	25

ΕΙΚΟΝΑ 39: Δεύτερο Μέρος Services στο frontend	26
ΕΙΚΟΝΑ 40: Κάρτα	27
ΕΙΚΟΝΑ 41: Toolbar	28
ΕΙΚΟΝΑ 42: Paginator	28
ΕΙΚΟΝΑ 43: Toast	28
ΕΙΚΟΝΑ 44: Εικόνα προφίλ	29
ΕΙΚΟΝΑ 45: Εικόνα αντικειμένου	29
ΕΙΚΟΝΑ 46: Logout icon από το Angular Material	29
ΕΙΚΟΝΑ 47: Κώδικας δημιουργίας JWT token.....	30
ΕΙΚΟΝΑ 48: Κώδικας συμπίεσης και αποσυμπίεσης.....	31
ΕΙΚΟΝΑ 49: Μορφή αποθήκευσης κωδικού στην βάση δεδομένων.....	32

Κατάλογος Πινάκων

ΠΙΝΑΚΑΣ 1 ΟΡΟΛΟΓΙΑΣ.....	35
ΠΙΝΑΚΑΣ 2: ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ.....	37

Κατάλογος Διαγραμμάτων

ΔΙΑΓΡΑΜΜΑ 1: Σχήμα βάσης δεδομένων	33
--	----

Εισαγωγή

Η παρούσα πτυχιακή εργασία εστιάζει στην ανάπτυξη μιας διαδικτυακής εφαρμογής που αποσκοπεί στην εύκολη διαμοίραση αντικειμένων προς δωρεά. Σε αυτή την πλατφόρμα οι χρήστες έχουν την δυνατότητα να αναρτούν πράγματα προς δωρεά αλλά και να αναζητούν και να επιλέγουν πράγματα που έχουν δωρίσει άλλοι. Το αντικείμενο της μελέτης έχει μεγάλη σημασία καθώς ανταποκρίνεται στην όλο και μεγαλύτερη ανάγκη για επαναχρησιμοποίηση αντικειμένων και την μείωση της ρύπανσης του περιβάλλοντος που συμβαίνει από την υπερβολική κατανάλωση. Διεθνώς υπάρχουν αρκετές παρόμοιες στον σκοπό πλατφόρμες όμως αυτή ξεχωρίζει δίνοντας βάση στην απλότητα και στην ευχρηστία για τους χρήστες.

Η βασική ερευνητική στόχευση της εργασίας είναι η υλοποίηση μιας εφαρμογής χρησιμοποιώντας σύγχρονες τεχνολογίες, όπως το Spring Boot για το backend, Angular για το frontend και το MySQL για την βάση δεδομένων. Η μεθοδολογία που χρησιμοποιήθηκε ήταν η Agile(Ευκίνητος) επιτρέποντας την ύπαρξη ευελιξίας κατά την υλοποίηση και την συνεχή διόρθωση και βελτίωση της εφαρμογής, μέσα από επαναληπτικές φάσεις ανάπτυξης (Sprints(Σπριντ)) και δοκιμών.

Οι στόχοι της πτυχιακής είναι η υλοποίηση μιας εύχρηστης και λειτουργικής πλατφόρμας δωρεών που θα διευκολύνει τους χρήστες να κάνουν και να δέχονται δωρεές διασφαλίζοντας πως τα δεδομένα τους αποθηκεύονται και χρησιμοποιούνται με ασφάλεια. Επίσης παρέχοντας ένα απλό σύστημα διεπαφής οι χρήστες ενθαρρύνονται να επαναχρησιμοποιήσουν την εφαρμογή και να προβούν σε επιπλέον δράσεις αλληλεγγύης. Η επιστημονική συνεισφορά της πτυχιακής εργασίας περιλαμβάνει την ανάπτυξη μιας λύσης με σύγχρονες τεχνολογίες που θα ενθαρρύνει τους χρήστες σε κοινωνικές δράσεις. Μέσω της εφαρμογής προωθείται η ευκολότερη συμμετοχή σε ενέργειες που θα επωφελήσουν την κοινωνία συμβάλλοντας στην κυκλική οικονομία και ευαισθητοποιώντας τους ανθρώπους στα περιβαλλοντικά και κοινωνικά θέματα.

Η δομή της παρούσας πτυχιακής εργασίας είναι η εξής:

- Στο Κεφάλαιο 1 περιγράφεται ο τρόπος χρήσης της εφαρμογής και οι λειτουργίες της.
- Στο Κεφάλαιο 2 αναλύεται το frontend κομμάτι της εφαρμογής.
- Στο Κεφάλαιο 3 αναλύεται το backend κομμάτι της εφαρμογής.
- Στο Κεφάλαιο 4 παρουσιάζεται η βάση δεδομένων.
- Στο τέλος αναλύονται τα συμπεράσματα και οι πιθανές επεκτάσεις.

1. Χρήση Εφαρμογής

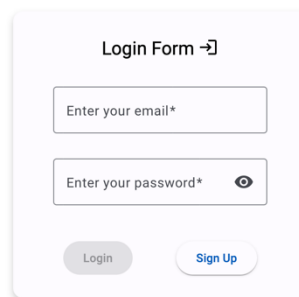
1.1 Εισαγωγή

Η εφαρμογή υλοποιήθηκε με στόχο την απλότητα και την εύκολη χρήση από έναν καινούργιο χρήστη. Για αυτόν τον λόγο οι σελίδες περιέχουν όλα όσα μπορεί να χρειαστεί ο χρήστης για τις δωρεές που θα κάνει ή θα δεχτεί χωρίς να τον μπερδεύουν. Συνοπτικά η εφαρμογή αποτελείται από 3 σελίδες και 5 pop up (αναδυόμενες) σελίδες. Αυτές οι σελίδες είναι το Login(Σύνδεση), Sign Up(Εγγραφή), Home(Σπίτι), Add Item(Προσθήκη αντικειμένου), Posted Items(Αναρτημένα αντικείμενα), Picked Items και Edit User(Επεξεργασία χρήστη) αντίστοιχα. Για την είσοδο στην εφαρμογή ο χρήστης πρέπει να έχει δημιουργήσει λογαριασμό και να συνδεθεί μέσω της σελίδας Login εισάγοντας το email και το password του. Στην περίπτωση που δεν έχει λογαριασμό θα πρέπει να φτιάξει έναν μέσω της σελίδας Sign Up και έπειτα από την δημιουργία του λογαριασμού θα συνδεθεί αυτόματα με τα στοιχεία του καινούργιου λογαριασμού. Στην συνέχεια θα εξηγηθούν σε μεγαλύτερο βάθος αυτές και οι υπόλοιπες λειτουργίες.

1.2 Σελίδες Εφαρμογής

1.2.1 Login

Η σελίδα Login είναι η πρώτη που θα αντικρίσει ένας νέος χρήστης. Σε αυτή την σελίδα θα πρέπει να συμπληρώσει το email και το password του λογαριασμού του στα πεδία που αναφέρουν Enter your email*(Εισάγετε το ηλεκτρονικό σας ταχυδρομείο) και Enter your password*(Εισάγετε τον κωδικό σας) αντιστοίχως, για να συνδεθεί στην εφαρμογή. Τα σύμβολα * που εμφανίζονται σε διάφορες φόρμες στην εφαρμογή δηλώνουν πως τα πεδία είναι υποχρεωτικά.



ΕΙΚΟΝΑ 1: Login σελίδα

Υπάρχουν 3 κουμπιά εκ των οποίων το ένα είναι για την διευκόλυνση του χρήστη στην συμπλήρωση του password(Κωδικός) του. Πατώντας το κουμπί που βρίσκεται μέσα στο πεδίο του

password με το εικονίδιο του ματιού θα μπορέσει να αλλάξει την μορφή των χαρακτήρων που συμπληρώνει από τα σύμβολα * στα κανονικά σύμβολα που πληκτρολογεί.

Επίσης πατώντας αυτό το κουμπί ξανά αλλάζει το εικονίδιο στο μάτι με μια γραμμή συμβολίζοντας το ότι αν το πατήσει δεν θα μπορεί να δει τον κωδικό του, μετατρέποντας την μορφή του πεδίου στην αρχική.



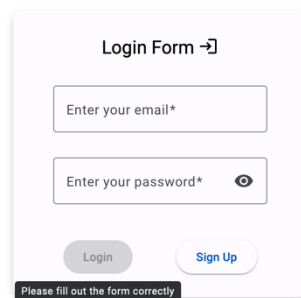
ΕΙΚΟΝΑ 2: Εικονίδιο ματιού στο πεδίο password για να μην εμφανίζεται το password



ΕΙΚΟΝΑ 3: Εικονίδιο ματιού στο πεδίο password για να εμφανίζεται το password

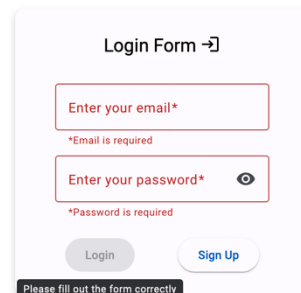
Η αρχική μορφή που εμφανίζεται σε αυτό το πεδίο είναι με τα σύμβολα * που κρύβουν το password του χρήστη και αυτό γίνεται για λόγους ασφάλειας έτσι ώστε να μην φαίνεται τι πληκτρολογεί. Τα άλλα δύο κουμπιά που υπάρχουν είναι το κουμπί Login και το κουμπί Sign Up. Στην περίπτωση του κουμπιού Login ο χρήστης θα μπορέσει να το πατήσει μόνο όταν έχει συμπληρώσει σωστά την φόρμα ενώ το κουμπί Sign Up θα αλλάξει την σελίδα στην Sign Up.

Ένα ενδιαφέρον και πολύ χρήσιμο μέρος αυτής και των άλλων φορμών που υπάρχουν στην εφαρμογή είναι ότι υποδεικνύουν στον χρήστη τα λάθη που έχει κάνει στην συμπλήρωση της φόρμας. Αρχικά πριν πατήσει το κουμπί Login αν πάει το ποντίκι του πάνω από το συγκεκριμένο κουμπί θα του εμφανίσει ένα μικρό μήνυμα κάτω από το κουμπί ενημερώνοντας τον πως πρέπει να συμπληρώσει σωστά την φόρμα.



ΕΙΚΟΝΑ 4: Login σελίδα με tooltip(Συμβουλή εργαλείου)

Στην περίπτωση που θα πατήσει το κουμπί χωρίς να έχει συμπληρωθεί σωστά η φόρμα θα εμφανιστούν κόκκινα γράμματα κάτω από τα πεδία που δεν έχουν συμπληρωθεί σωστά ενημερώνοντας τον χρήστη τι συμπληρώνει λανθασμένα.



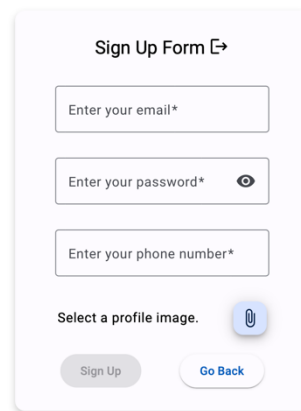
The image shows a login form titled "Login Form →". It contains two input fields: "Enter your email*" and "Enter your password*". Below the email field is the error message "*Email is required". Below the password field is the error message "*Password is required" and a small eye icon. At the bottom of the form are two buttons: "Login" and "Sign Up". A tooltip at the bottom left of the form reads "Please fill out the form correctly".

ΕΙΚΟΝΑ 5: Login σελίδα με error messages(Μηνύματα σφάλματος) και tooltip

Μόλις συμπληρώσει σωστά την φόρμα με τα στοιχεία του λογαριασμού του και πατήσει το κουμπί Login θα έχει πρόσβαση στην εφαρμογή.


1.2.2 Sign Up

Ο χρήστης θα πρέπει να επισκεφθεί την σελίδα Sign Up για να δημιουργήσει λογαριασμό για να έχει πρόσβαση στην εφαρμογή. Έχει πολλά κοινά στοιχεία με την σελίδα Login αλλά ζητά επιπλέον στοιχεία από τον χρήστη. Ένα από τα δύο επιπλέον στοιχεία είναι ένα νούμερο τηλεφώνου που είναι υποχρεωτικό πεδίο και η σημασία του θα φανεί στα παρακάτω κεφάλαια. Το δεύτερο επιπλέον στοιχείο είναι μία εικόνα προφίλ που δεν είναι υποχρεωτικό πεδίο και πάλι η σημασία του θα εξηγηθεί στα επόμενα κεφάλαια. Οπότε για να μπορέσει ο χρήστης να δημιουργήσει έναν λογαριασμό θα πρέπει να εισάγει ένα email(Ηλεκτρονικό ταχυδρομείο), password, τηλέφωνο και αν το επιθυμεί μια εικόνα profile(Προφίλ).




Sign Up Form ↗

Enter your email*

Enter your password* 

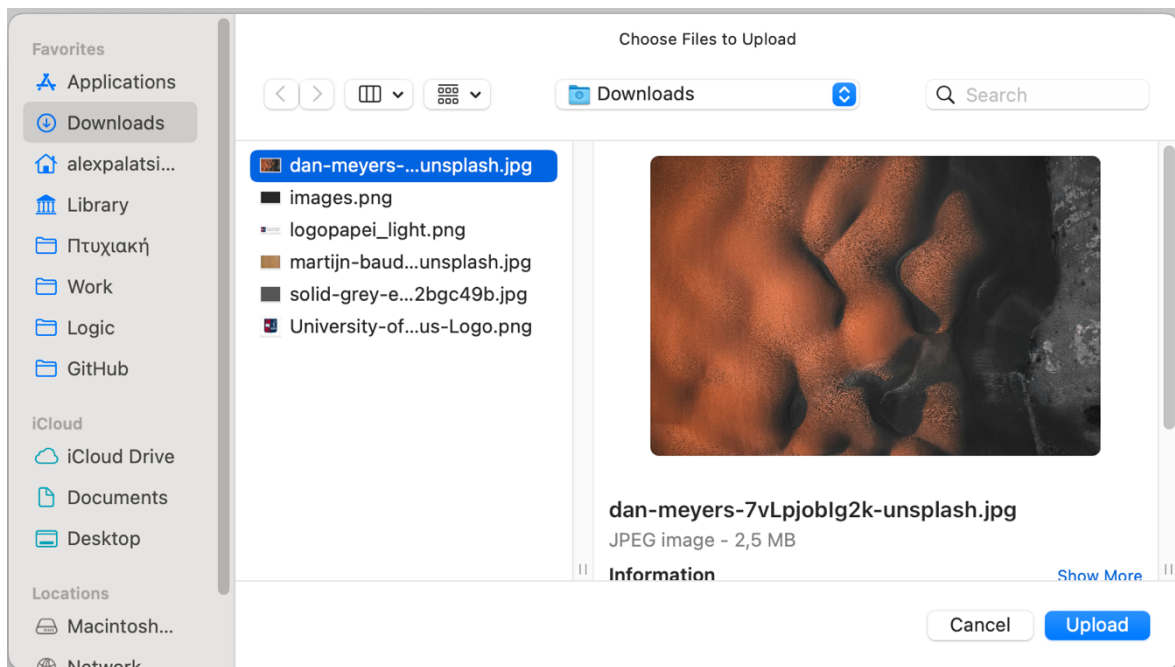
Enter your phone number*

Select a profile image. 

Sign Up Go Back

ΕΙΚΟΝΑ 6: Sign Up σελίδα

Για να εισάγει μια εικόνα προφίλ πρέπει να πατήσει το κουμπί με το εικονίδιο του συνδετήρα και να επιλέξει μια εικόνα από τα αρχεία του από το παράθυρο που θα εμφανιστεί.



ΕΙΚΟΝΑ 7: Παράθυρο επιλογής εικόνας

Μόλις επιλέξει μια εικόνα θα εμφανιστεί δίπλα στον συνδετήρα.

The image shows a 'Sign Up Form' with the following elements:

- Title: Sign Up Form ↗
- Input field: Enter your email*
- Input field: Enter your password* with an eye icon for visibility toggle.
- Input field: Enter your phone number*
- Profile picture selection: A circular image of a globe is selected, with a blue paperclip icon to its right.
- Buttons: 'Sign Up' (grey) and 'Go Back' (blue).

ΕΙΚΟΝΑ 8: Sign Up σελίδα με επιλεγμένη εικόνα προφίλ

Όπως και στην φόρμα Login έτσι και εδώ υπάρχουν το tooltip και τα κόκκινα γράμματα επεξήγησης στην περίπτωση λανθασμένης συμπλήρωσης της φόρμας.

This image is identical to the previous one, but with a tooltip at the bottom:

- Text: Select a profile image. (with a blue paperclip icon)
- Buttons: 'Sign Up' (grey) and 'Go Back' (blue).
- Tooltip: A black box with white text at the bottom that reads 'Please fill out the form correctly'.

ΕΙΚΟΝΑ 9: Sign Up σελίδα με tooltip

The screenshot shows a 'Sign Up Form' with three input fields: 'Enter your email*', 'Enter your password*', and 'Enter your phone number*'. Each field has a red border and a red error message below it: '*Email is required', '*Password is required', and '*Phone is required'. There is a 'Select a profile image.' label with a blue upload icon. At the bottom, there are 'Sign Up' and 'Go Back' buttons. A tooltip at the bottom left says 'Please fill out the form correctly'.

ΕΙΚΟΝΑ 10: Sign Up σελίδα με error messages και tooltip

Τέλος υπάρχουν άλλα δύο κουμπιά, το Sign Up που δημιουργεί τον λογαριασμό εφόσον είναι σωστά συμπληρωμένη η φόρμα και συνδέει τον χρήστη στην εφαρμογή και το κουμπί Go Back(Πήγαينه πίσω) που μεταφέρει τον χρήστη πίσω στην σελίδα Login.

1.2.3 Home

Το Home είναι η κύρια σελίδα της εφαρμογής. Σε αυτή την σελίδα ο χρήστης έχει την δυνατότητα να χρησιμοποιήσει τις βασικές λειτουργίες της εφαρμογής. Αυτές είναι οι εξής: αποσύνδεση από τον λογαριασμό του, προσθήκη αντικειμένου για δωρεά, εμφάνιση επιλεγμένων αντικειμένων, εμφάνιση αναρτημένων αντικειμένων, επεξεργασία προφίλ και επιλογή αντικειμένων που έχουν αναρτήσει άλλοι χρήστες προς δωρεά.

Στο πάνω μέρος της σελίδας υπάρχει μια γραμμή εργαλείων, η οποία στα αριστερά γράφει ένα μήνυμα καλωσορίσματος ακολουθούμενο από το email του χρήστη. Στην μέση της γραμμής είναι τα κουμπιά του Add Item, Posted Items, Picked Items και Edit User οι λειτουργίες των οποίων θα αναλυθούν στα παρακάτω κεφάλαια.

Welcome pal@com Log Out

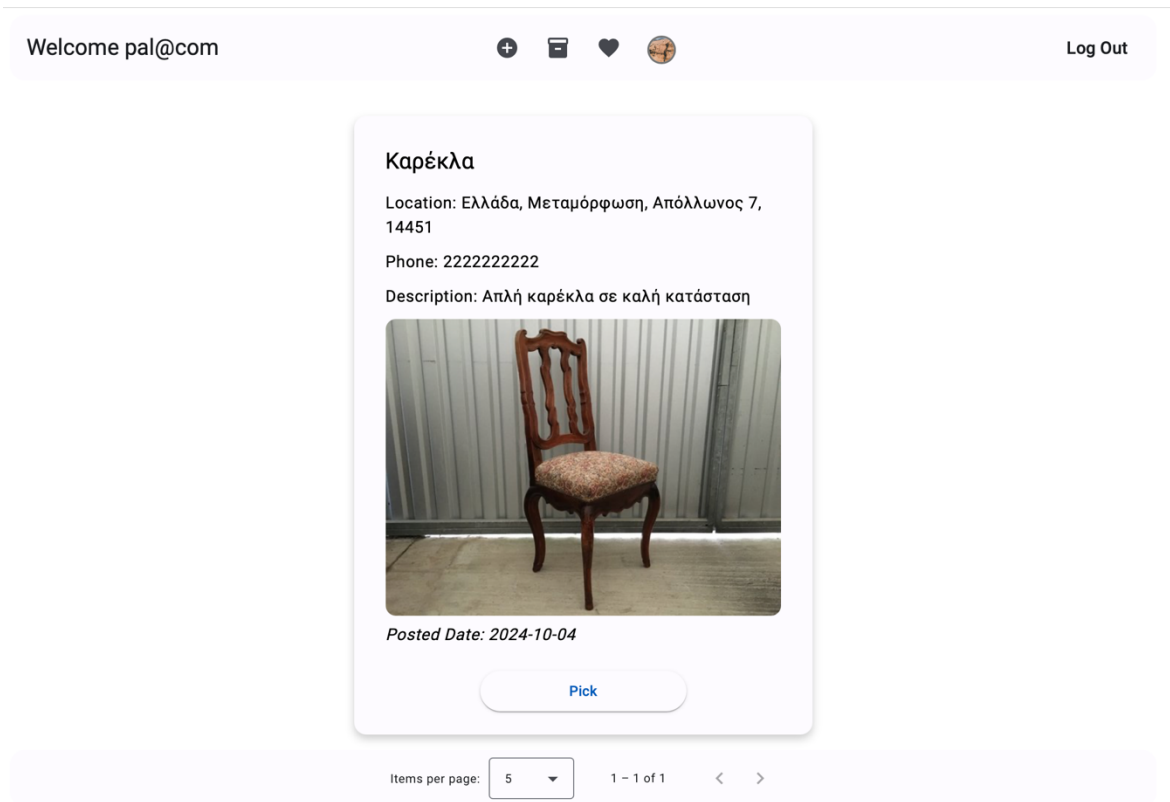
ΕΙΚΟΝΑ 11: Γραμμή εργαλείων στην σελίδα Home

Στο κάτω μέρος της σελίδας είναι το εργαλείο σελιδοποίησης που δίνει την δυνατότητα στον χρήστη να περιηγηθεί στα αντικείμενα που υπάρχουν προς δωρεά. Σε αυτό το σημείο μπορεί να επιλέξει πόσα αντικείμενα να εμφανίζονται την φορά αλλά και να δει πόσα αντικείμενα υπάρχουν. Πατώντας τα κουμπιά < και > πλοηγείτε στην προηγούμενη και επόμενη σελίδα των αντικειμένων.

Items per page: 5 1 - 1 of 1 < >

ΕΙΚΟΝΑ 12: Γραμμή πλοήγησης μεταξύ αντικειμένων προς δωρεά στην σελίδα Home

Στο κύριο μέρος που είναι στην μέση της σελίδας είναι τα προϊόντα προς δωρεά. Το κάθε αντικείμενο βρίσκεται στην δική του κάρτα όπου εμφανίζονται πληροφορίες για το αντικείμενο. Αυτές οι πληροφορίες είναι ο τίτλος του αντικειμένου, η τοποθεσία του, ένα τηλέφωνο επικοινωνίας, ένα σχόλιο, η ημερομηνία ανάρτησης, μια εικόνα του αντικειμένου και ένα κουμπί Pick(Επέλεξε) που δίνει την δυνατότητα στον χρήστη να επιλέξει το αντικείμενο. Εφόσον ένα αντικείμενο επιλεγθεί δεν θα είναι πλέον ορατό στην σελίδα Home αλλά θα υπάρχει στην σελίδα Picked Items.



ΕΙΚΟΝΑ 13: Home σελίδα

1.3 Παράθυρα Εφαρμογής

1.3.1 Edit User

Το Edit User δεν είναι σελίδα όπως οι προηγούμενες που αναφέρθηκαν αλλά ένα παράθυρο στο οποίο ο χρήστης μπορεί να κάνει αλλαγές στον λογαριασμό του. Για να επισκεφθεί αυτό το παράθυρο ο χρήστης θα πρέπει να πατήσει το κουμπί Edit User στην σελίδα Home στην γραμμή εντολών. Το συγκεκριμένα κουμπί είναι είτε η εικόνα προφίλ του είτε ένα ανθρωπάκι αναλόγως με το αν έχει εισάγει εικόνα για το προφίλ του και βρίσκεται στην μέση της γραμμής εργαλείων δεξιά από τα υπόλοιπα κουμπιά.



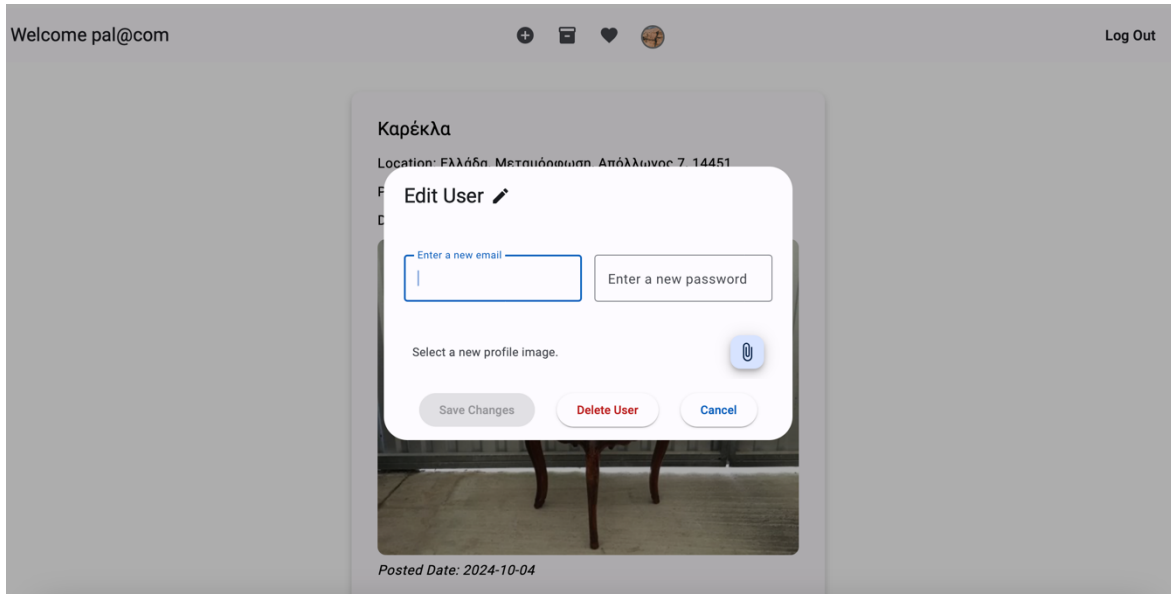
ΕΙΚΟΝΑ 14: Edit User κουμπί με ανθρωπάκι στην σελίδα Home



ΕΙΚΟΝΑ 15: Edit User κουμπί με εικόνα προφίλ στην σελίδα Home

Σε αυτό το παράθυρο ο χρήστης μπορεί να αλλάξει το email, το password ή/και την εικόνα προφίλ του. Επίσης μπορεί να διαγράψει τον λογαριασμό του. Για την επίτευξη αυτών των λειτουργιών υπάρχουν πεδία στα οποία μπορεί να εισάγει τα καινούργια στοιχεία και ένα κουμπί με το εικονίδιο του συνδετήρα το οποίο αν πατήσει θα του εμφανιστεί ένα παράθυρο επιλογής εικόνας προφίλ από τα αρχεία όπως και στην σελίδα Sign Up.

Στο κάτω μέρος του παραθύρου υπάρχουν 3 κουμπιά τα οποία είναι το Save Changes(Αποθήκευση αλλαγών), το Delete User(Διαγραφή χρήστη) και το Cancel(Ακύρωση).



ΕΙΚΟΝΑ 16: Edit User παράθυρο

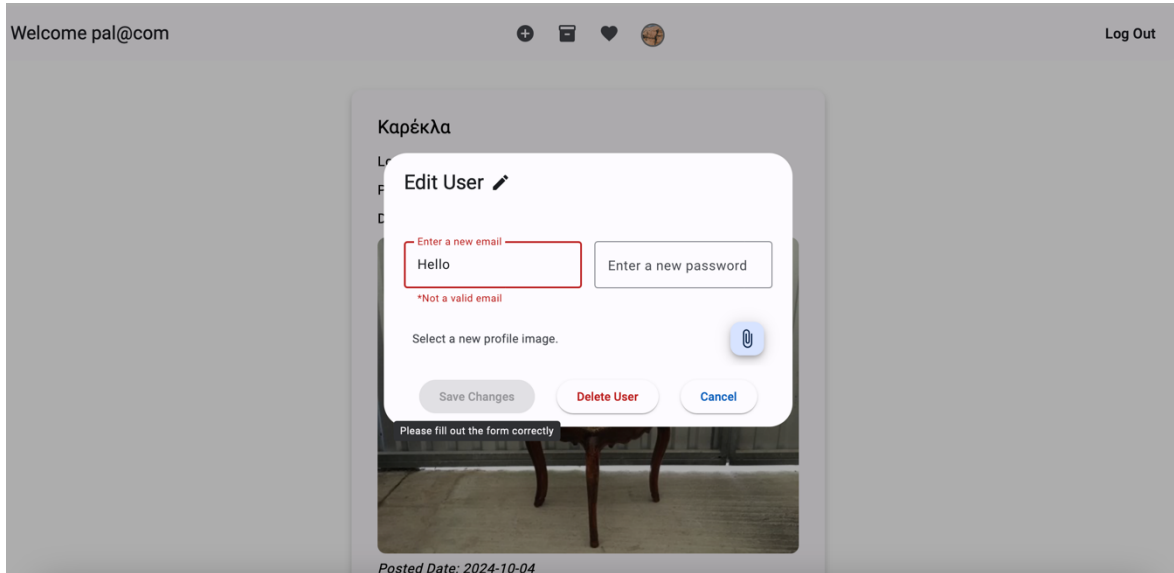
Το Save Changes είναι αρχικά γκριζοαρισμένο και δεν μπορεί να πατηθεί επειδή για να αλλάξει κάτι από τον λογαριασμό του θα πρέπει να συμπληρώσει σωστά κάποιο/κάποια από τα στοιχεία που αναφέρθηκαν προηγουμένως. Ο μόνος περιορισμός είναι το ότι δεν μπορεί να βάλει ένα email με το οποίο ήδη υπάρχει λογαριασμός. Στην περίπτωση που εισάγει ένα email που ήδη υπάρχει θα σταλθεί ο χρήστης πίσω στην σελίδα Home χωρίς να περαστούν οι αλλαγές.

Πατώντας το κουμπί Delete User θα διαγραφεί ο λογαριασμός και θα αλλάξει η σελίδα στην Login. Σημαντικό είναι να αναφέρω πως αυτή η ενέργεια δεν είναι αντιστρέψιμη γι' αυτό τον λόγο ο χρήστης πρέπει να είναι σίγουρος ότι θέλει να διαγράψει τον λογαριασμό του πριν πατήσει το κουμπί Delete User. Επίσης μαζί με τον λογαριασμό θα διαγραφτούν τα αναρτημένα αντικείμενα του χρήστη και τα αντικείμενα που έχει επιλέξει θα μπορούν να επιλεγθούν ξανά από κάποιον άλλο χρήστη γι' αυτό τον λόγο είναι σημαντικό όταν ένα αντικείμενο έχει δωριστεί να διαγράφεται από τον

χρήστη που το έχει αναρτήσει ώστε να μην υπάρχει κακή επικοινωνία στο ότι κάποιο αντικείμενο είναι για δωρεά όταν στην πραγματικότητα δεν είναι.

Το τελευταίο κουμπί που βρίσκεται στην κάτω-δεξιά γωνία του παραθύρου είναι το Cancel και μεταφέρει τον χρήστη στην σελίδα Home. Αυτή η λειτουργία μπορεί επίσης να επιτευχθεί με το πάτημα έξω από το παράθυρο αλλά και πάλι στην ιστοσελίδα οδηγώντας τον χρήστη πάλι στην σελίδα Home.

Όπως έχει αναφερθεί αρκετές φορές και στις προηγούμενες φόρμες υπάρχουν έλεγχοι κατά την συμπλήρωση των στοιχείων προς αλλαγή με το tooltip κάτω από το κουμπί Save Changes και τα κοκκινισμένα γράμματα κάτω από τα πεδία σε περίπτωση λανθασμένης συμπλήρωσης της φόρμας.



ΕΙΚΟΝΑ 17: Edit User παράθυρο με error message και tooltip

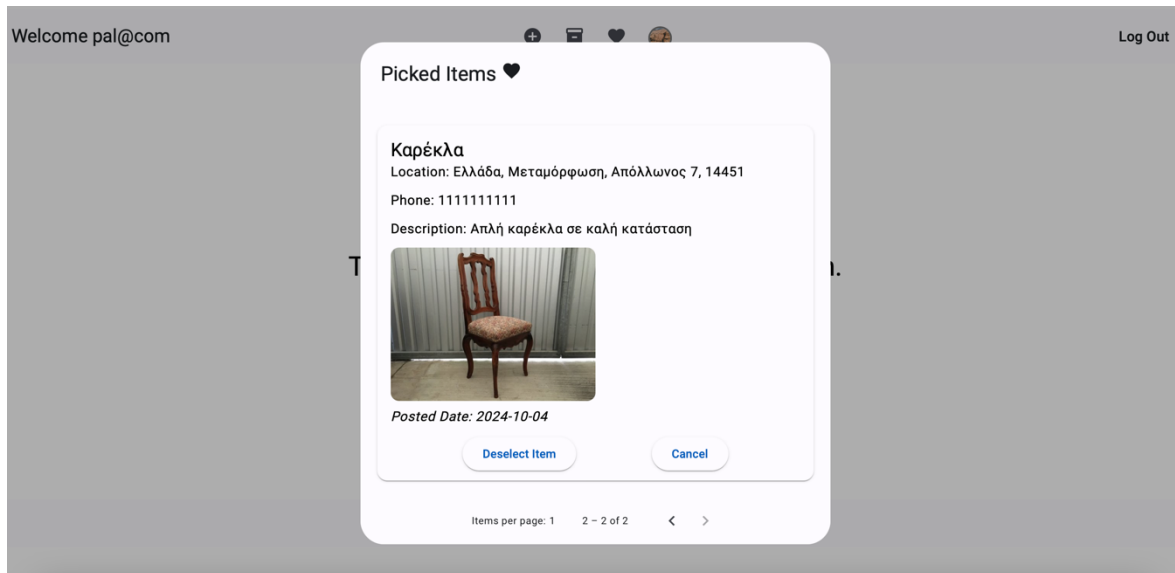
1.3.2 Picked Items

Το παράθυρο Picked Items είναι ο τρόπος με τον οποίο ο χρήστης μπορεί να δει τα πράγματα που έχει επιλέξει να δεχτεί σαν δωρεά. Μόλις πατήσει το κουμπί Picked Items στην σελίδα Home στην γραμμή εργαλείων θα ανοίξει το παράθυρο. Το συγκεκριμένο κουμπί είναι με το εικονίδιο της καρδιάς και βρίσκεται τρίτο από τα αριστερά στην μέση της γραμμής εργαλείων.



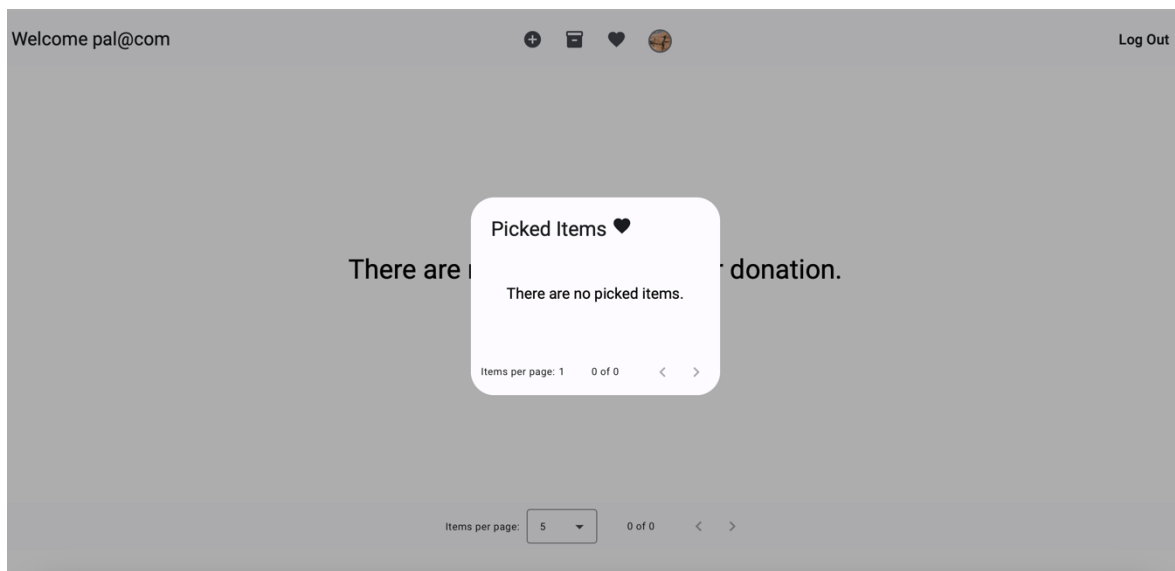
ΕΙΚΟΝΑ 18: Picked Item κουμπί με καρδιά στην σελίδα Home

Στην μέση του παραθύρου αναγράφονται πληροφορίες για τα αντικείμενα που έχει επιλέξει και περιλαμβάνονται σε μία κάρτα για κάθε αντικείμενο. Αυτές είναι ένας τίτλος για το αντικείμενο, η τοποθεσία του, ένα τηλέφωνο επικοινωνίας, ένα σχόλιο, μία εικόνα του αντικειμένου και η ημερομηνία ανάρτησης. Κάτω από αυτά τα στοιχεία υπάρχουν τα κουμπιά Deselect Item (Αποεπιλογή αντικειμένου) και το Cancel. Το Deselect Item αν πατηθεί θα βγάλει το αντικείμενο από τα επιλεγμένα και θα μπορεί να επιλεγθεί από τους άλλους χρήστες ενώ το κουμπί Cancel αν πατηθεί θα πλοηγηθεί ο χρήστης στην σελίδα Home. Η λειτουργία του κουμπιού Cancel μπορεί να επιτευχθεί με το πάτημα έξω από το παράθυρο.



ΕΙΚΟΝΑ 19: Picked Items παράθυρο με επιλεγμένα αντικείμενα

Στην περίπτωση που δεν έχει επιλέξει αντικείμενα να δεχτεί ως δωρεά θα του εμφανιστεί το μήνυμα There are no picked items(Δεν υπάρχουν επιλεγμένα αντικείμενα) αντί για τα αντικείμενα.



ΕΙΚΟΝΑ 20: Picked Items παράθυρο χωρίς επιλεγμένα αντικείμενα

Στο κάτω μέρος του παραθύρου υπάρχει το εργαλείο με το οποίο ο χρήστης μπορεί να πλοηγηθεί μεταξύ των επιλεγμένων αντικειμένων του. Πατώντας τα κουμπιά < και > αλλάζει στο προηγούμενο και στο επόμενο επιλεγμένο αντικείμενο αντίστοιχα. Επίσης μπορεί να δει πόσα επιλεγμένα αντικείμενα έχει.

Items per page: 1 2 – 2 of 2 < >

ΕΙΚΟΝΑ 21: Γραμμή πλοήγησης μεταξύ επιλεγμένων αντικειμένων στο παράθυρο Picked Items

1.3.3 Posted Items

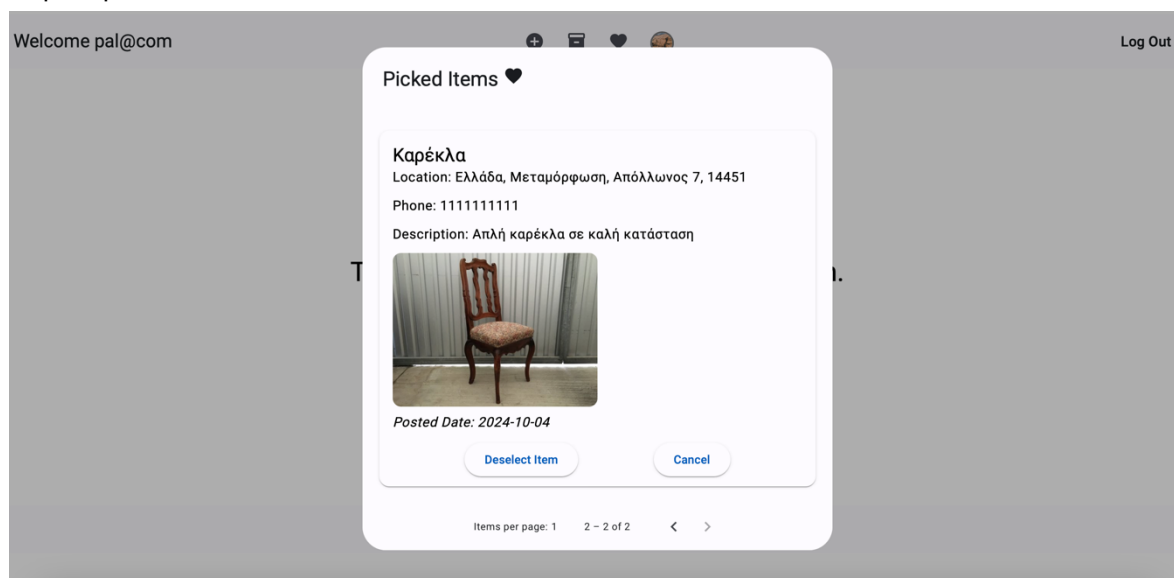
Στο παράθυρο Posted Items ο χρήστης βλέπει τα αντικείμενα που έχει αναρτήσει. Για να πλοηγηθεί σε αυτό το παράθυρο θα πρέπει να πατήσει το κουμπί Posted Items στην σελίδα Home στην γραμμή εργαλείων. Το κουμπί είναι με το εικονίδιο του φακέλου και βρίσκεται δεύτερο στην μέση της γραμμής εργαλείων στην σελίδα Home.



ΕΙΚΟΝΑ 22: Posted Items κουμπί με εικονίδιο φακέλου στην σελίδα Home

Στην μέση του παραθύρου είναι τα στοιχεία του αντικειμένου που έχει αναρτήσει μέσα σε μια κάρτα. Αυτά τα στοιχεία είναι ο τίτλος του εντικειμένου, η τοποθεσία του, αν έχει επιλεγθεί, ένα σχόλιο, μία εικόνα του αντικειμένου και την ημερομηνία ανάρτησης.

Κάτω από αυτά είναι τα κουμπιά Edit Item(Επεξεργασία αντικειμένου), Delete Item(Διαγραφή αντικειμένου) και Cancel. Το Edit Item είναι για την αλλαγή των στοιχείων του αντικειμένου, το Delete είναι για την διαγραφή του αντικειμένου και το Cancel είναι για την επιστροφή στην σελίδα Home. Για να επιστρέψει στην σελίδα Home μπορεί επίσης να πατήσει έξω από το παράθυρο.



ΕΙΚΟΝΑ 23: Posted Items παράθυρο

Στο κάτω μέρος του παραθύρου είναι το εργαλείο με το οποίο μπορεί να περιηγηθεί μεταξύ των αναρτημένων αντικειμένων του. Πατώντας το < και το > κουμπί μπορεί να πάει στο προηγούμενο και στο επόμενο αντικείμενο αντίστοιχα. Επίσης μπορεί να δει πόσα αντικείμενα έχει αναρτήσει.

Items per page: 1 5 – 5 of 5 < >

ΕΙΚΟΝΑ 24: Γραμμή πλοήγησης μεταξύ επιλεγμένων αντικειμένων στο παράθυρο Posted Items

1.3.4 Add Item

Αυτό το παράθυρο είναι για την ανάρτηση αντικειμένων προς δωρεά. Για να επισκεφθεί αυτό το παράθυρο πρέπει να πατήσει το κουμπί Add Item στην σελίδα Home στην γραμμή εργαλείων. Το συγκεκριμένο κουμπί είναι με το εικονίδιο του γεμάτου κύκλου με έναν σταυρό μέσα του.



ΕΙΚΟΝΑ 25: Add Item κουμπί με εικονίδιο σταυρού στην σελίδα Home

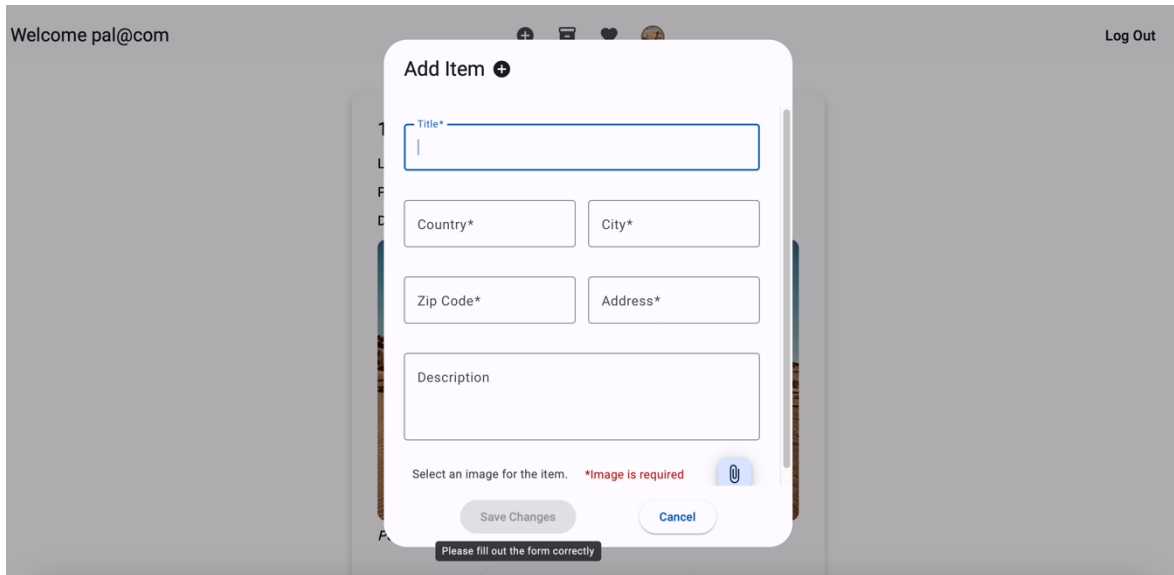
Για να αναρτήσει ο χρήστης ένα αντικείμενο πρέπει να συμπληρώσει τα πεδία Title(Τίτλος), Country(Χώρα), City(Πόλη), Zip Code(Ταχυδρομικός κώδικας), Address(Διεύθυνση) και Εικόνα. Υπάρχει και το πεδίο Description(Σχόλιο) όμως δεν είναι υποχρεωτικό. Μόλις συμπληρώσει τα στοιχεία σωστά θα πρέπει να πατήσει το κουμπί Save Changes που βρίσκεται κάτω από τα πεδία για να αναρτηθεί το αντικείμενο. Για να εισάγει εικόνα για το αντικείμενο θα πρέπει να πατήσει το κουμπί με το εικονίδιο του συνδετήρα και να διαλέξει εικόνα από τα αρχεία του από το παράθυρο που θα εμφανιστεί.

ΕΙΚΟΝΑ 26: Add Item παράθυρο

Όπως και σε άλλες φόρμες στην εφαρμογή υπάρχουν έλεγχοι για την σωστή συμπλήρωση της φόρμας. Δηλαδή δεν θα μπορέσει να πατήσει το κουμπί Save Changes όταν είναι λανθασμένα συμπληρωμένη και αν πάει το ποντίκι του πάνω από το κουμπί θα του εμφανιστεί το tooltip

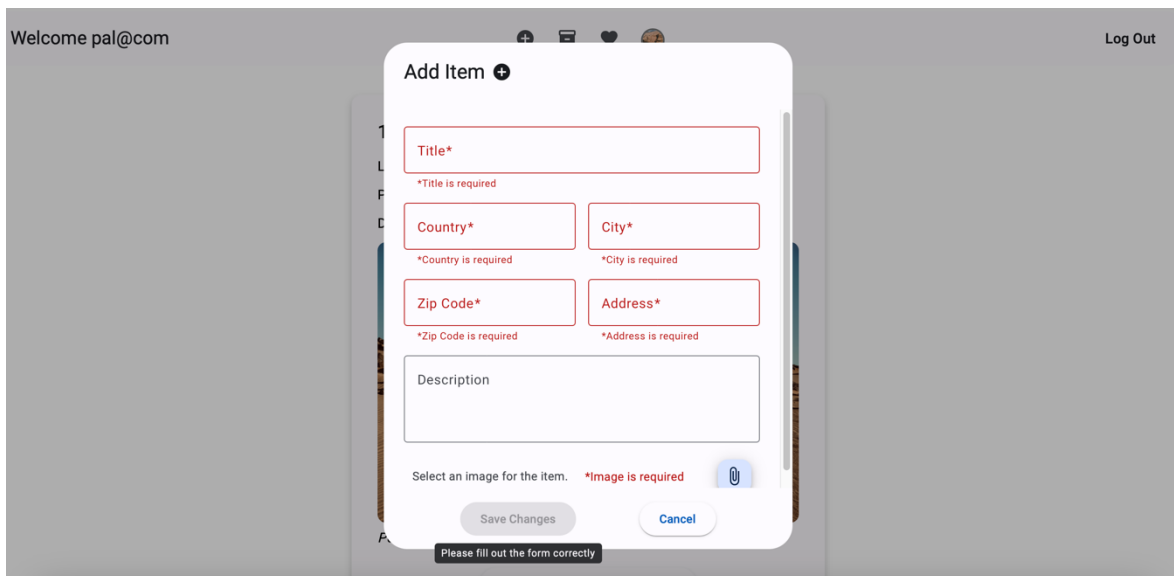
Προπτυχιακή Διατριβή
ενημερώνοντας τον να συμπληρώσει σωστά την φόρμα.

Αλέξανδρος Παλασιδης



ΕΙΚΟΝΑ 27: Add Item παράθυρο με tooltip

Επίσης αν πατήσει το κουμπί Save Changes ή αν εισάγει σε κάποιο πεδίο λανθασμένα θα εμφανιστούν κοκκινισμένα γράμματα κάτω από τα πεδία υποδικνύοντας στον χρήστη τι συμπληρώνει λανθασμένα.

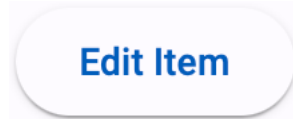


ΕΙΚΟΝΑ 28: Add Item με error messages και tooltip

Τέλος στην κάτω δεξιά γωνία του παραθύρου υπάρχει το κουμπί Cancel που μεταφέρει τον χρήστη πίσω στην σελίδα Home. Αυτή η λειτουργία μπορεί να επιτευχθεί με το πάτημα έξω από το παράθυρο.

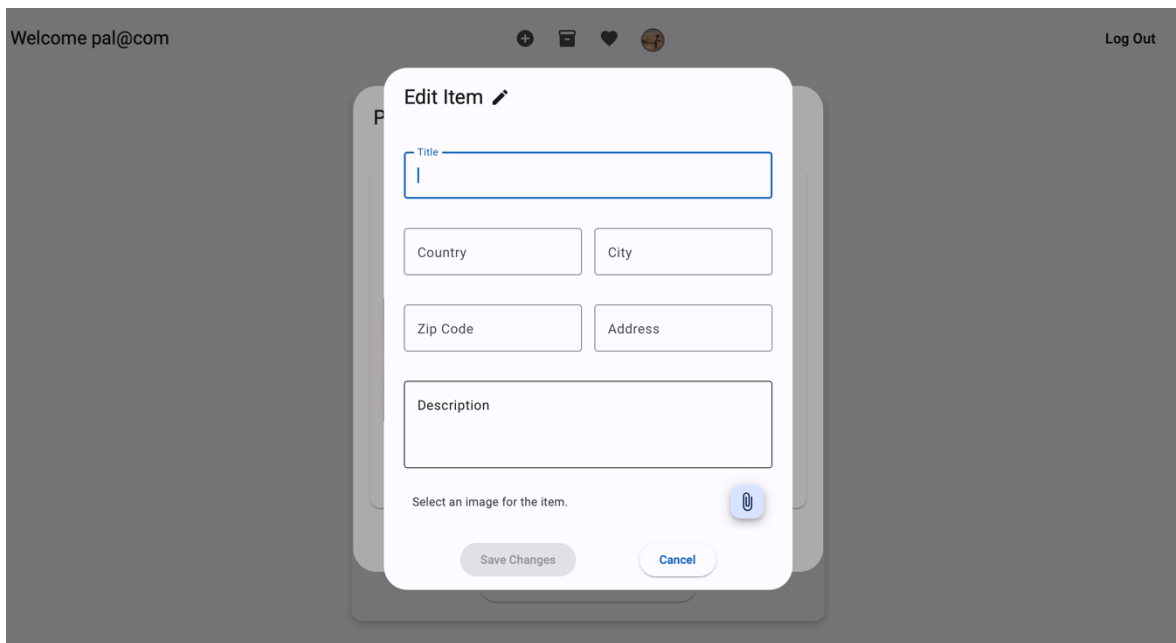
1.3.5 Edit Item

Η ιδιαιτερότητα αυτού του παραθύρου είναι ότι ανοίγει από άλλο παράθυρο και συγκεκριμένα από το παράθυρο Posted Items πατώντας το κουμπί Edit Item στην κάτω αριστερή γωνία του παραθύρου Posted Items.



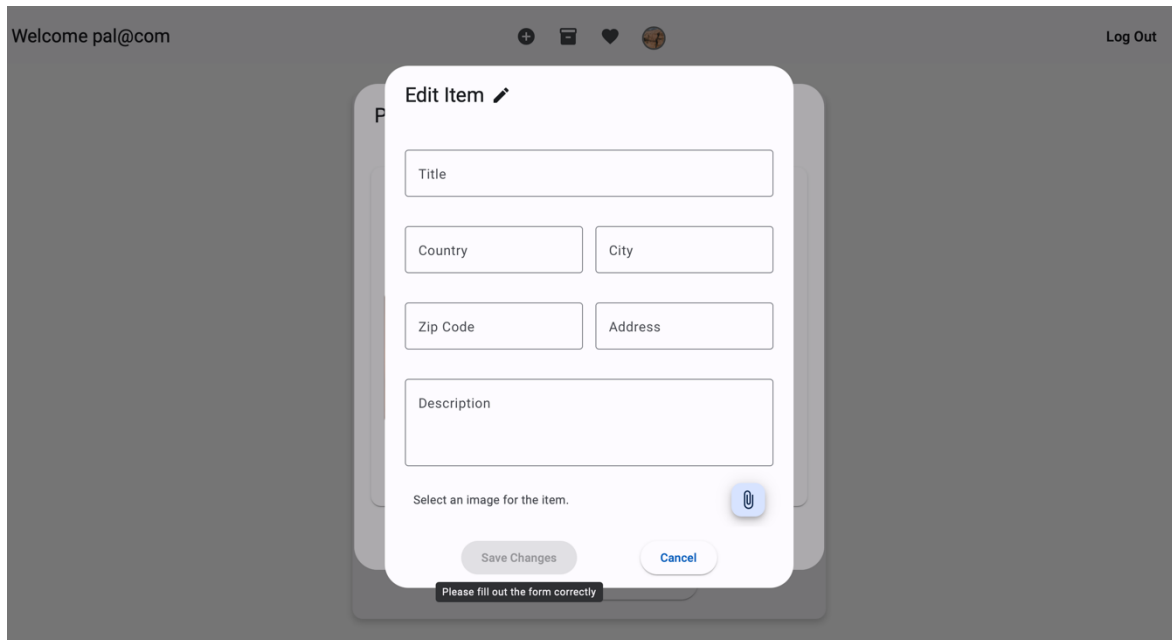
ΕΙΚΟΝΑ 29: Edit Item κουμπί στο παράθυρο Posted Items

Αυτό το παράθυρο είναι πανομοιότυπο με το παράθυρο Add Item με την μόνη διαφορά στους ελέγχους που γίνονται. Κανένα πεδίο δεν είναι υποχρεωτικό όμως για να μπορέσει να πατήσει το κουμπί Save Changes θα πρέπει να έχει συμπληρώσει τουλάχιστον ένα πεδίο της φόρμας.



ΕΙΚΟΝΑ 30: Edit Item παράθυρο

Διαφορετικά αν πάει το ποντίκι του πάνω από το κουμπί θα είναι απενεργοποιημένο και θα του εμφανίσει ένα tool tip κάτω από το κουμπί ενημερώνοντας τον χρήστη να συμπληρώσει σωστά την φόρμα.



ΕΙΚΟΝΑ 31: Edit Item παράθυρο με tooltip

Μόλις συμπληρώσει σωστά την φόρμα θα μπορέσει να πατήσει το κουμπί Save Changes και να αποθηκεύσει τις αλλαγές. Έπειτα θα μεταφερθεί στην σελίδα Home.

2. Frontend

2.1 Εισαγωγή

Η παρούσα εφαρμογή βασίζεται στην αρχιτεκτονική Client-Server(Πελάτης-διακοσμητής) γι' αυτό τον λόγο υπάρχει το frontend κομμάτι αλλά και το backend κομμάτι. Το frontend κομμάτι λειτουργεί στον υπολογιστή του χρήστη (Client(Πελάτη)) ενώ το backend λειτουργεί σε κάποιος απομακρυσμένος server. Στην περίπτωση της συγκεκριμένης εφαρμογής και τα δύο μέρη λειτουργούν στον ίδιο υπολογιστή στο τοπικό δίκτυο του υπολογιστή. Τίποτα όμως δεν περιορίζει την εφαρμογή από το να λειτουργεί σε ένα ευρύτερο δίκτυο αν υπήρχε ο κατάλληλος server(Διακοσμητής) που θα υποστήριζε την λειτουργικότητα του frontend και του backend.

Το frontend είναι το κομμάτι που βλέπει ο χρήστης και βασίζεται στο backend για τις πληροφορίες που εμφανίζει και που επεξεργάζονται κατά την εκτέλεση των ενεργειών που κάνει ο χρήστης στο frontend. Η συνεργασία μεταξύ του frontend και του backend γίνεται με την χρήση των http requests(αιτήματα) τα οποία δέχεται το backend σαν ένα πακέτο πληροφοριών και στέλνει πίσω σαν απάντηση. Παρακάτω θα εξηγηθεί σε μεγαλύτερο βάθος η χρήση των requests.

2.2 Τεχνολογίες που χρησιμοποιήθηκαν

2.2.1 Angular

Το Angular είναι το framework της Google που χρησιμοποιήθηκε για την ανάπτυξη του frontend κομματιού μιας ιστοσελίδας. Σε αυτή την πτυχιακή χρησιμοποιήθηκε η γλώσσα προγραμματισμού TypeScript για την οργάνωση που προσδίδει καθώς και για τον περιορισμό των λαθών κατά την υλοποίηση της εφαρμογής έχοντας επιπλέον λειτουργικότητα σε σχέση με την γλώσσα προγραμματισμού JavaScript στην οποία και βασίζεται. Μία σημαντική προσθήκη του TypeScript είναι το Type Safety(ασφάλεια τύπου) δηλαδή ασφάλεια τύπων και σημαίνει πως οι μεταβλητές έχουν έναν συγκεκριμένο τύπο κατά την αρχικοποίηση τους έτσι ώστε να μην υπάρχουν λογικά λάθη κατά την υλοποίηση της εφαρμογής. Κατά την χρήση του framework(Πλαίσιο) για την ανάπτυξη μιας εφαρμογής κάποιος θα μπορέσει να παρατηρήσει εύκολα πως είναι ιδιαίτερο στον τρόπο με τον οποίο πρέπει να χρησιμοποιηθεί. Αυτό μπορεί αρχικά να παρουσιάσει το framework ως δύσχρηστο όμως προσδίδει στην απλότητα του έχοντας έναν μοναδικό τρόπο χρήσης.

2.2.2 Τρόπος εκκίνησης του server

Για την εκκίνηση του frontend απαιτείται η εκτέλεση της εντολής `ng serve -o` σε ένα terminal στον αρχικό φάκελο frontend της εφαρμογής. Η προσθήκη του `-o` θα ανοίξει την ιστοσελίδα στον default browser (Πρόγραμμα περιήγησης) του υπολογιστή. Μόλις γίνει η εκκίνηση του frontend χωρίς σφάλματα θα εμφανιστεί στο terminal το url στο οποίο λειτουργεί η ιστοσελίδα και στην συγκεκριμένη περίπτωση είναι στο `http://localhost:4200/`.

```

alexpalatsidis@Alexs-MacBook-Air frontend % ng serve -o
Browser bundles
Initial chunk files | Names | Raw size
main.js | main | 106.04 kB
polyfills.js | polyfills | 88.10 kB
styles.css | styles | 70.88 kB
scripts.js | scripts | 29.14 kB
| Initial total | 294.15 kB

Server bundles
Initial chunk files | Names | Raw size
main.server.mjs | main.server | 1.63 MB
chunk-SC3FBPMK.mjs | - | 1.19 MB
chunk-SSQZXQI2.mjs | - | 650.57 kB
polyfills.server.mjs | polyfills.server | 570.81 kB
chunk-QDHZVCWX.mjs | - | 2.57 kB
render-utils.server.mjs | render-utils.server | 462 bytes

Lazy chunk files | Names | Raw size
chunk-XD2MYPR2.mjs | xhr2 | 40.04 kB
chunk-PVW4SIYE.mjs | browser | 1.72 kB

Application bundle generation complete. [2.928 seconds]

Watch mode enabled. Watching for file changes..
NOTE: Raw file sizes do not reflect development server per-request transformations.
  → Local: http://localhost:4200/
  → press h + enter to show help
NG02801: Angular detected that `HttpClient` is not configured to use `fetch` APIs. It's strongly recommended to enable `fetch` for applications that use Server-Side Rendering for better performance and compatibility. To enable `fetch`, add the `withFetch()` to the `provideHttpClient()` call at the root of the application.

```

EIKONA 32: Εκτέλεση `ng serve -o` για την εκκίνηση του frontend

2.2.3 Angular Material

Το Angular Material είναι μια βιβλιοθήκη με έτοιμα UI components (Συστατικά) για εφαρμογές Angular, βασισμένη στις αρχές του Material Design (Σχεδιασμός υλικού) της Google. Παρέχει στοιχεία όπως κουμπιά, φόρμες, διαλόγους, ειδοποιήσεις, μενού κ.α. για την εύκολη δημιουργία μοντέρνων και συνεπών διεπαφών χρήστη. Το frontend μέρος της εφαρμογής βασίζεται σε αυτή την βιβλιοθήκη με το μεγαλύτερο μέρος του να αποτελείται από κουμπιά, κάρτες, toolbar (Γραμμή εργαλείων), εργαλεία σελιδοποίησης, διαλόγους/παράθυρα, toasts (Μηνύματα ειδοποίησης), form fields (Πεδία φόρμας) κλπ. Στο Κεφάλαιο της μορφοποίησης αναλύεται περεταίρω το τι χρησιμοποιήθηκε.

2.3 Ασφάλεια

Από την αρχή της υλοποίησης της εργασίας μέχρι και το τέλος της η ασφάλεια των δεδομένων του χρήστη ήταν πάντα προτεραιότητα. Ένας τρόπος με τον οποίο επιτεύχθηκε αυτό ήταν με την χρήση των JWT tokens τα οποία πολύ απλά είναι κλειδιά που αυθεντικοποιούν την ταυτότητα του χρήστη. Αποθηκεύονται στο local storage του web browser κατά το login ή το sign up και περιλαμβάνονται στα http requests που γίνονται στο backend. Για παράδειγμα ο χρήστης δεν μπορεί να επισκεφθεί την σελίδα Home της εφαρμογής χωρίς να είναι συνδεδεμένος πληκτρολογώντας το url της αφού απαιτείται να έχει συνδεθεί και να έχει δεχτεί ένα κλειδί JWT. Στην περίπτωση που προσπαθήσει να επισκεφθεί την σελίδα χωρίς την κατάλληλη εξουσιοδότηση θα επιστρέψει αμέσως πίσω στην σελίδα Login.


```
1 import { HttpInterceptorFn } from '@angular/common/http';
2
3 export const authInterceptor: HttpInterceptorFn = (req, next) => {
4   let token: string | null = null;
5
6   if (typeof window !== 'undefined') {
7     try {
8       token = localStorage.getItem('token');
9     } catch (error) {
10      console.error('Error accessing localStorage:', error);
11    }
12  }
13
14  if (token) {
15    const authReq = req.clone({
16      setHeaders: {
17        Authorization: 'Bearer ' + JSON.parse(token).token,
18      },
19    });
20
21    return next(authReq);
22  }
23
24  return next(req);
25 };
```

ΕΙΚΟΝΑ 34: Interceptor στο frontend

Τέλος για να λειτουργεί το interceptor στην εφαρμογή προστέθηκε στο αρχείο `app.config.ts` στα providers το `provideHttpClient (withInterceptors([authInterceptor]))`.

```
1 import { ApplicationConfig, provideZoneChangeDetection } from '@angular/core';
2 import { provideRouter } from '@angular/router';
3
4 import { routes } from './app.routes';
5 import { provideClientHydration } from '@angular/platform-browser';
6 import { provideAnimationsAsync } from '@angular/platform-browser/animations/async';
7 import { provideHttpClient, withInterceptors } from '@angular/common/http';
8 import { authInterceptor } from './auth.interceptor';
9 import { IMAGE_CONFIG } from '@angular/common';
10
11 export const appConfig: ApplicationConfig = {
12   providers: [
13     provideZoneChangeDetection({ eventCoalescing: true }),
14     provideRouter(routes),
15     provideClientHydration(),
16     provideAnimationsAsync(),
17     provideHttpClient(withInterceptors([authInterceptor])),
18     {
19       provide: IMAGE_CONFIG,
20       useValue: {
21         disableImageSizeWarning: true,
22         disableImageLazyLoadWarning: true
23       }
24     },
25   ],
26 };
```

EIKONA 35: Interceptor στο αρχείο app.config.ts

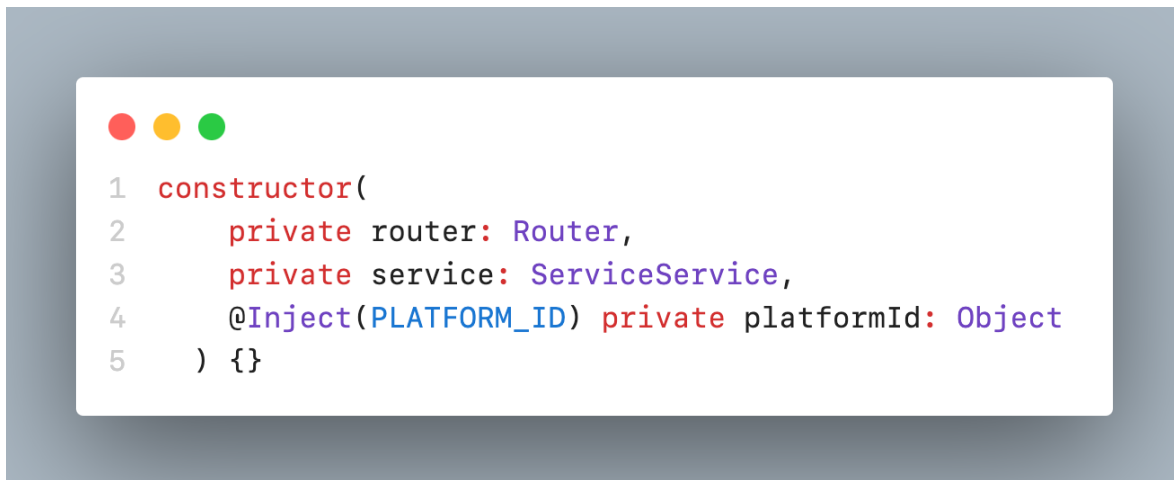
2.4 Services(Υπηρεσίες)

2.4.1 Εισαγωγή

Τα services είναι στο αρχείο service.service.ts και χρησιμοποιούνται για την εκτέλεση των αιτημάτων http. Για να χρησιμοποιηθούν σε κάποιο component (Σελίδα ή παράθυρο) της εφαρμογής πρέπει να γίνει imported το αρχείο του στο component και να οριστεί ως παράμετρος στο constructor του component όπως φαίνεται παρακάτω.

```
1 import { ServiceService } from '../services/service.service';
```

EIKONA 36: Import του Service αρχείου στο Login component



EΙΚΟΝΑ 37: Αναφορά του Service στο constructor του Login component

Τα services είναι τα εξής και φαίνονται στις παρακάτω δύο εικόνες:

- authenticateUser
- registerUser
- logout(Αποσύνδεση)
- getUserDetails
- editUserDetails
- deleteUser
- addItem
- getPicked
- getPosted.

Κάθε ένα από αυτά στέλνει και διαφορετικό αίτημα στο backend με διαφορετικό φορτίο. Όλα εκτός από το authenticateUser, το registerUser και το logout χρειάζονται JWT token.

2.4.2 authenticateUser

Το authenticateUser διαγράφει από το local storage οποιοδήποτε κλειδί μπορεί να παρέμεινε. Έχει σαν παραμέτρους το username και το password του χρήστη και στέλνει ένα post αίτημα στο `http://localhost:8080/api/authenticate` έχοντας ως φορτίο τις παραμέτρους. Αν τα στοιχεία του χρήστη είναι έγκυρα το backend θα στείλει πίσω στον χρήστη το JWT token που θα χρειαστεί για να χρησιμοποιήσει την εφαρμογή. Το service χρησιμοποιείται για το log in του χρήστη στην Login σελίδα.

2.4.3 registerUser

Το registerUser πάλι διαγράφει από το local storage οποιοδήποτε κλειδί μπορεί να παρέμεινε. Έχει για παράμετρο ένα αντικείμενο FormData(Δεδομένα φόρμας) το οποίο μέσα του διαθέτει το email, το password, το phone και την εικόνα προφίλ και στέλνει ένα post αίτημα στο `http://localhost:8080/api/register` έχοντας ως φορτίο το αντικείμενο FormData. Αν το email δεν υπάρχει στην βάση δεδομένων το backend θα στείλει πίσω στον χρήστη το JWT token που θα χρειαστεί για να χρησιμοποιήσει την εφαρμογή. Αν υπάρχει το email στην βάση δεδομένων βγάζει το αντίστοιχο error message. Το service χρησιμοποιείται για την εγγραφή του χρήστη στην εφαρμογή στην Sign Up σελίδα.

2.4.4 logout

Το logout απλά σβήνει από το local storage το JWT token. Χρησιμοποιείται για την αποσύνδεση του χρήστη.

2.4.5 getUserDetails

Το getUserDetails δεν έχει παραμέτρους και στέλνει ένα get αίτημα στο `http://localhost:8080/info/user-details` έχοντας ως φορτίο το JWT token. Αν το JWT token είναι έγκυρο δέχεται από το backend τα στοιχεία του χρήστη στον οποίο αντιστοιχεί το JWT token. Το service χρησιμοποιείται για το auto login και στο Home page κατά την σύνδεση αλλά και το refresh της σελίδας ώστε να μην χάνεται η σύνδεση του χρήστη στην εφαρμογή.

2.4.6 editUserDetails

Το editUserDetails έχει για παράμετρο ένα αντικείμενο FormData μέσα στο οποίο είναι τα στοιχεία του χρήστη προς αλλαγή και στέλνει ένα put αίτημα στο `http://localhost:8080/info/edit-user-details` έχοντας ως φορτίο το αντικείμενο FormData και το JWT token. Το service χρησιμοποιείται για την αλλαγή των στοιχείων του χρήστη στο Edit User παράθυρο.

2.4.7 deleteUser

Το deleteUser έχει για παράμετρο το id του χρήστη προς διαγραφή. Στέλνει ένα delete αίτημα στο `http://localhost:8080/info/delete-user/{id}` έχοντας ως φορτίο το id του χρήστη και το JWT token. Το service χρησιμοποιείται για την διαγραφή του χρήστη που είναι συνδεδεμένος στο παράθυρο Edit User.

2.4.8 addItem

Το addItem έχει για παράμετρο ένα αντικείμενο FormData μέσα στο οποίο είναι ο τίτλος του αντικειμένου, η τοποθεσία του, ένα σχόλιο και μια εικόνα και στέλνει ένα post αίτημα στο `http://localhost:8080/info/add-item` έχοντας ως φορτίο το αντικείμενο FormData και το JWT token. Το service χρησιμοποιείται για την ανάρτηση ενός αντικειμένου από το παράθυρο Add Item.

2.4.9 getItems

Το getItems έχει για παράμετρο το owner_id που είναι το id του χρήστη που ανάρτησε το αντικείμενο και στέλνει ένα get αίτημα στο `http://localhost:8080/info/get-item/{owner_id}` έχοντας ως φορτίο το owner_id και το JWT token. Δέχεται από το backend τα αντικείμενα των άλλων χρηστών. Το service χρησιμοποιείται για την εμφάνιση των αντικειμένων των άλλων χρηστών στην Home σελίδα.

2.4.10 pickItem

Το pickItem έχει ως παράμετρο ένα αντικείμενο FormData μέσα στο οποίο είναι το item_id του αντικειμένου που επιλέχθηκε, το accepting_user_id που είναι ο χρήστης που επιλέγει το αντικείμενο και το giving_user_id που είναι ο χρήστης που κάνει δωρεά το αντικείμενο και στέλνει ένα put αίτημα στο `http://localhost:8080/info/pick-item` έχοντας ως φορτίο το αντικείμενο FormData και το JWT token. Το service χρησιμοποιείται για την επιλογή του αντικειμένου στην Home σελίδα.

2.4.11 deselectItem

Το `deselectItem` έχει ως παράμετρο το `item_id` και στέλνει ένα `put` αίτημα στο `http://localhost:8080/info/deselect-item/{item_id}` έχοντας ως φορτίο το `item_id` και το `JWT token`. Το `service` χρησιμοποιείται για την ακύρωση της επιλογής του αντικειμένου στο παράθυρο `Picked Items`.

2.4.12 deleteItem

Το `deleteItem` έχει ως παράμετρο το `item_id` και στέλνει ένα `delete` αίτημα στο `http://localhost:8080/info/delete-item/{item_id}` έχοντας ως φορτίο το `item_id` και το `JWT token`. Το `service` χρησιμοποιείται για την διαγραφή του αντικειμένου που αναρτήθηκε από τον χρήστη στο παράθυρο `Posted Items`.

2.4.13 editItem

Το `editItem` έχει ως παράμετρο ένα αντικείμενο `FormData` μέσα στο οποίο είναι τα στοιχεία του αντικειμένου προς αλλαγή και στέλνει ένα `put` αίτημα στο `http://localhost:8080/info/edit-item` έχοντας ως φορτίο το αντικείμενο `FormData` και το `JWT token`. Το `service` χρησιμοποιείται για την αλλαγή των στοιχείων ενός αντικειμένου στο παράθυρο `Edit Item`.

2.4.14 getPickedItems

Το `getPickedItems` έχει ως παράμετρο το `id` του χρήστη που είναι συνδεδεμένος και στέλνει ένα `get` αίτημα στο `http://localhost:8080/info/picked-items/{id}` έχοντας ως φορτίο το `id` και το `JWT token`. Δέχεται από το `backend` τα επιλεγμένα αντικείμενα του χρήστη. Το `service` χρησιμοποιείται για την εμφάνιση των επιλεγμένων αντικειμένων του χρήστη στο παράθυρο `Picked Items`.

2.4.15 getPostedItems

Το `getPostedItems` έχει ως παράμετρο το `id` του χρήστη που είναι συνδεδεμένος και στέλνει ένα `get` αίτημα στο `http://localhost:8080/info/posted-items/{id}` έχοντας ως φορτίο το `id` του χρήστη και το `JWT token`. Δέχεται από το `backend` τα αναρτημένα αντικείμενα του χρήστη. Το `service` χρησιμοποιείται για την εμφάνιση των αναρτημένων αντικειμένων από τον χρήστη στο παράθυρο `Posted Items`.

```
1 import {
2   HttpClient,
3   HttpResponse,
4   HttpHeaders,
5 } from '@angular/common/http';
6 import { Injectable } from '@angular/core';
7 import { catchError, Observable, throwError } from 'rxjs';
8
9 @Injectable({
10   providedIn: 'root',
11 })
12 export class ServiceService {
13   constructor(private http: HttpClient) {}
14
15   authenticateUser(email: String, password: String): Observable<any> {
16     localStorage.removeItem('token');
17     return this.http.post('http://localhost:8080/api/authenticate', {
18       email: email,
19       password: password,
20     });
21   }
22
23   registerUser(data: FormData): Observable<any> {
24     localStorage.removeItem('token');
25
26     return this.http
27       .post('http://localhost:8080/api/register', data)
28       .pipe(catchError(this.handleError));
29   }
30
31   private handleError(error: HttpResponse) {
32     if (
33       error.status === 500 &&
34       error.error.message.includes('Duplicate entry')
35     ) {
36       return throwError(() => new Error('This email already exists.'));
37     } else {
38       return throwError(() => new Error('An unknown error occurred.'));
39     }
40   }
41
42   logout(): void {
43     localStorage.removeItem('token');
44   }
45 }
```

ΕΙΚΟΝΑ 38: Πρώτο Μέρος Services στο frontend

```
1  getUserDetails(): Observable<any> {
2    return this.http.get('http://localhost:8080/info/user-details');
3  }
4
5  editUserDetails(data: FormData): Observable<any> {
6    return this.http.put('http://localhost:8080/info/edit-user-details', data);
7  }
8
9  deleteUser(id: number): Observable<any> {
10   return this.http.delete('http://localhost:8080/info/delete-user/' + id);
11 }
12
13 addItem(data: FormData): Observable<any> {
14   return this.http.post('http://localhost:8080/info/add-item', data);
15 }
16
17 getItems(owner_id: string): Observable<any> {
18   return this.http.get('http://localhost:8080/info/get-items/' + owner_id);
19 }
20
21 pickItem(itemData: FormData): Observable<any> {
22   return this.http.put('http://localhost:8080/info/pick-item', itemData);
23 }
24
25 deselectItem(item_id: number): Observable<any> {
26   return this.http.put(
27     'http://localhost:8080/info/deselect-item/' + item_id,
28     {}
29   );
30 }
31
32 deleteItem(item_id: number): Observable<any> {
33   return this.http.delete(
34     'http://localhost:8080/info/delete-item/' + item_id
35   );
36 }
37
38 editItem(itemData: FormData): Observable<any> {
39   return this.http.put('http://localhost:8080/info/edit-item', itemData);
40 }
41
42 getPickedItems(id: number): Observable<any> {
43   return this.http.get('http://localhost:8080/info/picked-items/' + id);
44 }
45
46 getPostedItems(id: number): Observable<any> {
47   return this.http.get('http://localhost:8080/info/posted-items/' + id);
48 }
```

ΕΙΚΟΝΑ 39: Δεύτερο Μέρος Services στο frontend

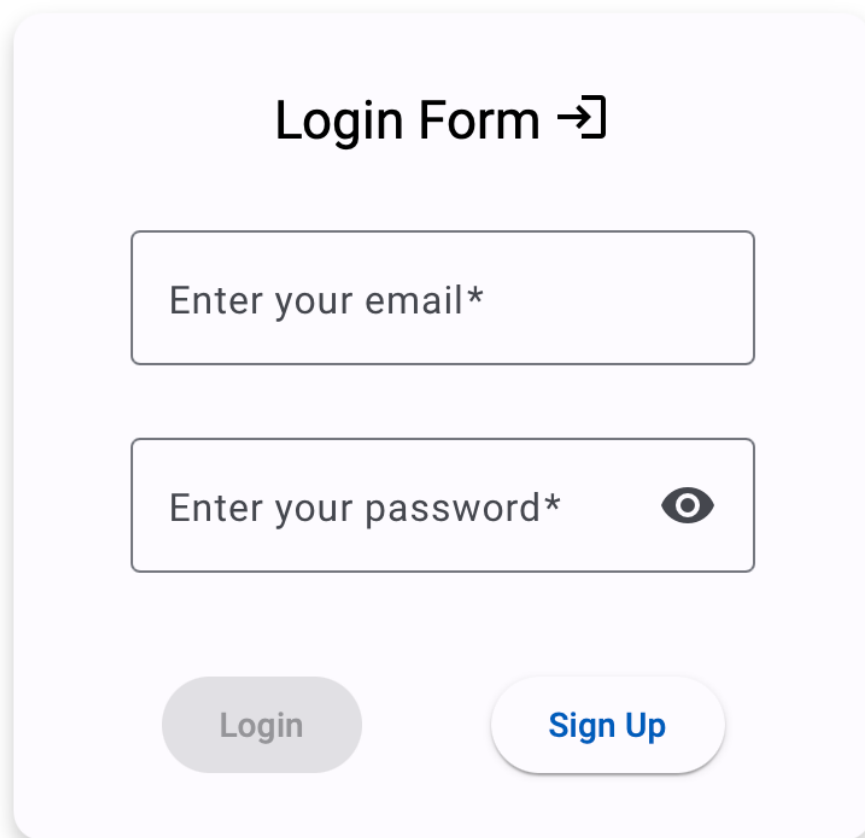
2.5 Μορφοποίηση Εφαρμογής

2.5.1 Εισαγωγή

Όπως είχε αναφερθεί και στην αρχή του κεφαλαίου η μορφοποίηση του frontend έγινε από το Angular Material. Παρακάτω αναλύονται κάποια από τα πιο σημαντικά components που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής.

2.5.2 Κάρτες

Οι κάρτες χρησιμοποιούνται στα περισσότερα σημεία της εφαρμογής για να οργανώσουν την πληροφορία που εμφανίζεται. Για την χρήση μιας κάρτας πρέπει να γίνει το import της στο component που θα χρησιμοποιηθεί και μετά να οριστεί στα imports του @Component ως module(Μονάδα).



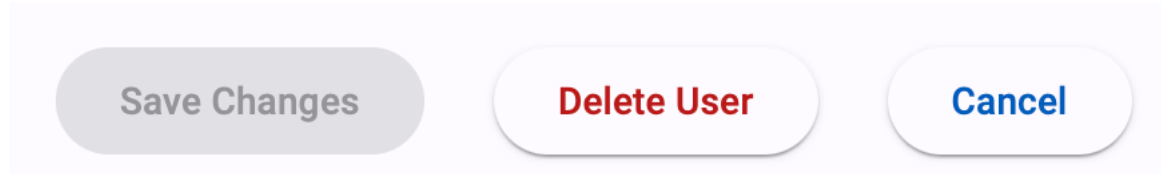
The image shows a login form card with the following elements:

- Title: "Login Form →" with a right-pointing arrow icon.
- Input field 1: "Enter your email*" with a light gray border.
- Input field 2: "Enter your password*" with a light gray border and an eye icon on the right side.
- Buttons: "Login" (light gray rounded button) and "Sign Up" (blue rounded button).

ΕΙΚΟΝΑ 40: Κάρτα

2.5.3 Κουμπιά

Τα κουμπιά στην εφαρμογή έχουν την ίδια μορφοποίηση με το στυλ περιγράμματος σε μπλε χρώμα με την μόνη διαφορά ότι σε κάποια σημεία τα γράμματα του κουμπιού εμφανίζονται με κόκκινο χρώμα προειδοποιώντας για κάποια ενέργεια μη αντιστρέψιμη. Επίσης τα κουμπιά που δεν μπορούν να πατηθούν λόγω σφάλματος στην συμπλήρωση της φόρμας είναι γκριζαρισμένα. Για την χρήση του καθώς και των υπόλοιπων components πρέπει να γίνουν τα imports όπως και στις κάρτες.



2.5.4 Toolbar

Το toolbar είναι από τα πρώτα πράγματα που θα παρατηρήσει ο χρήστης όταν συνδεθεί. Είναι η μπάρα που βρίσκεται στο πάνω μέρος της σελίδας Home και περιέχει όλα τα κουμπιά. Η μπάρα οργανώνει τα πιο σημαντικά κουμπιά της σελίδας σε ένα σημείο.



EIKONA 41: Toolbar

2.5.5 Paginator(Εργαλείο σελιδοποίησης)

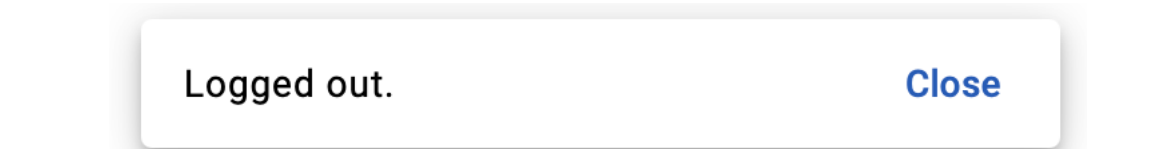
Το paginator είναι ο τρόπος με τον οποίο ο χρήστης μπορεί να δει τα αντικείμενα της εφαρμογής χωρίς να χαθεί σε μια τεράστια λίστα αν είναι αρκετά τα αντικείμενα. Σε άλλες περιπτώσεις μπορεί να διαλέξει πόσα αντικείμενα θα εμφανίζονται και σε άλλες εμφανίζονται ένα ένα και έτσι ώστε να εστιάσει σε ένα αντικείμενο την φορά. Διαθέτει κουμπιά και είναι στο κάτω μέρος πάντα είτε είναι σε σελίδα είτε σε παράθυρο.



EIKONA 42: Paginator

2.5.6 Toasts

Τα toasts είναι τα μικρά λευκά pop up μηνύματα που εμφανίζονται στο πάνω και μέσο μέρος της εφαρμογής για να ενημερώσει τον χρήστη για κάποια ενέργεια που επιτεύχθηκε.



EIKONA 43: Toast

2.5.7 Εικόνες και Icons(Εικονίδια)

Οι εικόνες εμφανίζονται με ένα γκρι περίβλημα και στρογγυλεμένες. Οι εικόνες προφίλ εμφανίζονται σε ένα κυκλικό σχήμα ενώ οι εικόνες των αντικειμένων σε ένα πιο μακρόστενο σχήμα ώστε η εφαρμογή να φαίνεται σύγχρονη και πρακτική.



ΕΙΚΟΝΑ 44: Εικόνα προφίλ



ΕΙΚΟΝΑ 45: Εικόνα αντικειμένου

Τα υπόλοιπα εικονίδια που εμφανίζονται σε όλη την εφαρμογή είναι από το Angular Material, όπως το παρακάτω που έχει την ονομασία logout στην σελίδα Sign Up.



ΕΙΚΟΝΑ 46: Logout icon από το Angular Material

3. Backend

3.1 Εισαγωγή

Για το backend κομμάτι της εφαρμογής χρησιμοποιήθηκε η τεχνολογία Spring Boot η οποία βασίζεται στο Spring Framework και στοχεύει στην μείωση της πολυπλοκότητας που απαιτείται για την ανάπτυξη μιας εφαρμογής.

3.2 Ασφάλεια

3.2.1 JWT tokens

Όπως τονίστηκε και στο κεφάλαιο του frontend η ασφάλεια των δεδομένων των χρηστών ήταν προτεραιότητα από την αρχή της υλοποίησης της εφαρμογής. Το μεγαλύτερο όμως μέρος της ασφάλειας της εφαρμογής είναι στο backend. Σε αυτό το κομμάτι δημιουργούνται τα JWT tokens από τα στοιχεία του χρήστη και στέλνονται στο frontend.

Κατά την δημιουργία του token δίνονται τα στοιχεία του χρήστη, η ημερομηνία δημιουργίας, η ημερομηνία λήξης του token και το μυστικό κλειδί για να υπογραφεί το token.

```
1 public String generateToken(  
2     Map<String, Object> extraClaims,  
3     UserDetails userDetails) {  
4     return Jwts  
5         .builder()  
6         .setClaims(extraClaims)  
7         .setSubject(userDetails.getUsername())  
8         .setIssuedAt(new Date(System.currentTimeMillis()))  
9         .setExpiration(new Date(System.currentTimeMillis() + 10000 * 60 * 24))  
10        .signWith(getSignInKey(), SignatureAlgorithm.HS256)  
11        .compact();  
12    }
```

EIKONA 47: Κώδικας δημιουργίας JWT token

3.2.2 Κρυπτογράφηση κωδικών και συμπίεση εικόνων

Με σκοπό την ασφάλεια των δεδομένων των χρηστών και την αποθήκευση των εικόνων με αποδοτικό τρόπο γίνεται μείωση του μεγέθους του αρχείου πριν από την αποθήκευσή τους στην βάση δεδομένων. Χρησιμοποιώντας τις βιβλιοθήκες Deflater(Αποσυμπίεστης) και Inflater η συμπίεση και η αποσυμπίεση των εικόνων.

```
1 public static byte[] compressBytes(byte[] data) {
2     Deflater deflater = new Deflater();
3     deflater.setInput(data);
4     deflater.finish();
5
6     ByteArrayOutputStream outputStream = new ByteArrayOutputStream(data.length);
7     byte[] buffer = new byte[1024];
8     while (!deflater.finished()) {
9         int count = deflater.deflate(buffer);
10        outputStream.write(buffer, 0, count);
11    }
12    try {
13        outputStream.close();
14    } catch (IOException e) {
15    }
16
17    return outputStream.toByteArray();
18 }
19
20 public static byte[] decompressBytes(byte[] data) {
21     Inflater inflater = new Inflater();
22     inflater.setInput(data);
23     ByteArrayOutputStream outputStream = new ByteArrayOutputStream(data.length);
24     byte[] buffer = new byte[1024];
25     try {
26         while (!inflater.finished()) {
27             int count = inflater.inflate(buffer);
28             outputStream.write(buffer, 0, count);
29         }
30         outputStream.close();
31     } catch (IOException ioe) {
32     } catch (DataFormatException e) {
33     }
34     return outputStream.toByteArray();
35 }
```

ΕΙΚΟΝΑ 48: Κώδικας συμπίεσης και αποσυμπίεσης

Συνεχίζοντας με την ασφάλεια στην εργασία γίνεται η κρυπτογράφηση των κωδικών πριν αποθηκευτούν στην βάση δεδομένων χρησιμοποιώντας την βιβλιοθήκη PasswordEncoder.

4. Βάση Δεδομένων

4.1 Εισαγωγή

Η τεχνολογία που χρησιμοποιήθηκε για την βάση δεδομένων είναι το MySQL από την εταιρεία Oracle. Ο κωδικός της βάσης δεδομένων είναι db3UBVuM6pJMGkk και λειτουργεί στο localhost:3306.

4.2 Δομή της βάσης δεδομένων

Η βάση δεδομένων αποτελείται από τους πίνακες item, transaction και user. Με αυτό τον τρόπο δεν επαναλαμβάνεται κάποιο κομμάτι πληροφορίας και εκτελούνται ταχύτατα τα queries(Ερωτήματα).

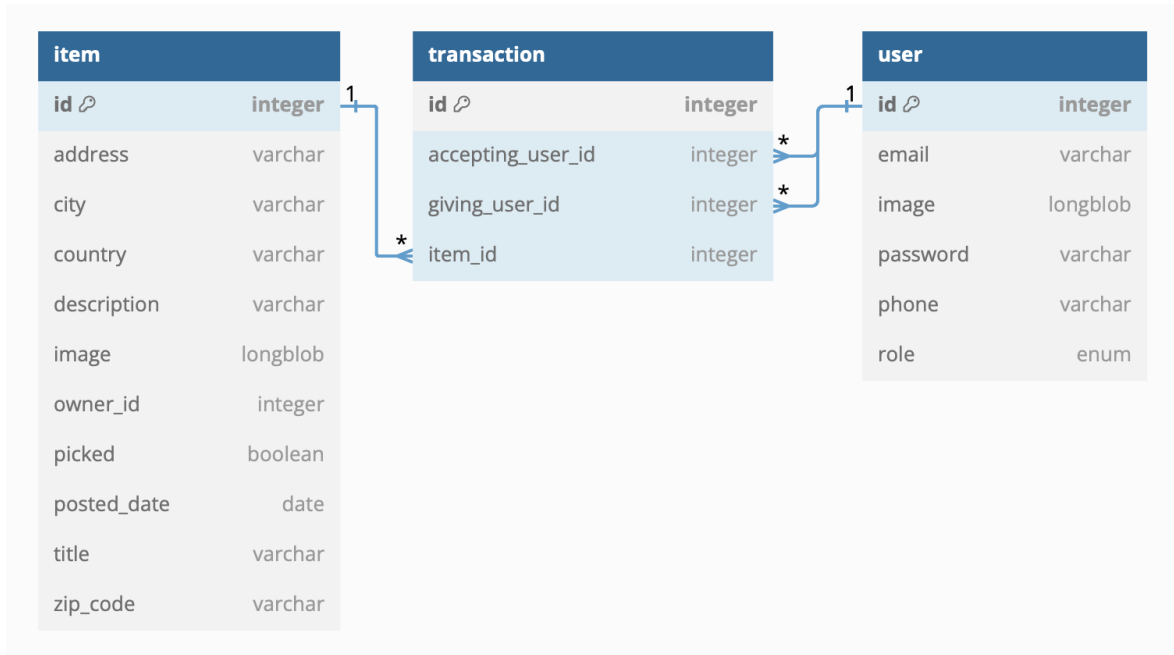
Ο πίνακας item διατηρεί τις πληροφορίες όλων των αναρτημένων αντικειμένων και έχει σχέση one to many(Ένα προς πολλά) με τον πίνακα transaction(Συναλλαγή). Για την αποθήκευση των εικόνων γίνεται η χρήση του τύπου blob που ουσιαστικά διατηρεί την πληροφορία της εικόνας σε μια λίστα δυαδικών αριθμών.

Ο πίνακας user(Χρήστης) διατηρεί τις πληροφορίες όλων των χρηστών και έχει σχέση one to many με τον πίνακα transaction. Η αποθήκευση του κωδικού γίνεται με κρυπτογραφημένο τρόπο ώστε να είναι μόνο γνωστός στον κάτοχο του λογαριασμού για λόγους ασφαλείας.

id	email	image	password	phone	role
2802	pal@com		\$2a\$10\$kenE4aYk06dDiV4LfnokqutNON/QTSg...	1111111111	USER

ΕΙΚΟΝΑ 49: Μορφή αποθήκευσης κωδικού στην βάση δεδομένων

Ο πίνακας transaction διατηρεί τις πληροφορίες των συναλλαγών μεταξύ των χρηστών και έχει σχέσεις many to one(πολλά προς ένα) με τους πίνακες item και user. Λειτουργεί σαν γέφυρα μεταξύ των πινάκων user και item καθώς η πληροφορία που διαθέτει εκτός από το πρωτεύον κλειδί είναι ξένα κλειδιά που αντιστοιχούν στους άλλους πίνακες.



ΔΙΑΓΡΑΜΜΑ 1: Σχήμα βάσης δεδομένων

Συμπεράσματα

Η παρούσα πτυχιακή είχε ως σκοπό την ανάπτυξη μιας διαδικτυακής εφαρμογής που χρησιμοποιεί Spring Boot ως τεχνολογία για το backend, μια τεχνολογία frontend και μια βάση δεδομένων. Εδώ χρησιμοποιήθηκε το Angular για το frontend και το MySQL για την βάση δεδομένων. Μέσω της μελέτης και της ανάπτυξης της εφαρμογής, οι στόχοι επικεντρώθηκαν στην επίλυση συγκεκριμένων προβλημάτων όπως η ασφαλής μετάδοση δεδομένων μέσω των JWT tokens και η απλότητα στην διεπαφή με τον χρήστη. Η χρήση των JWT tokens και του Angular Material για την μορφοποίηση και την λειτουργικότητα του frontend ήταν πολύ σημαντικά κομμάτια της πτυχιακής.

Μερικά από τα προβλήματα που αντιμετωπίστηκαν μπορούν να θεωρηθούν επεκτάσεις της εργασίας είναι η ανάγκη για συνεχή ανανέωση των JWT tokens και η ανάγκη για έναν καλύτερο τρόπο παρουσίασης της τοποθεσίας των αντικειμένων όπως είναι ένας χάρτης. Ωστόσο ο στόχος επιτεύχθηκε έχοντας αναπτύξει μια πλήρως λειτουργική εφαρμογή που μπορεί να επεκταθεί περαιτέρω. Τα δεδομένα των χρηστών διαχειρίζονται με ασφάλεια με την χρήση των JWT tokens και η διεπαφή με τον χρήστη είναι απλούστατη και εύχρηστη.

Η συνεισφορά αυτής της εργασίας είναι στον χώρο της ασφάλειας και του UI design(Σχεδιασμός).

Οι κατευθύνσεις για μελλοντική έρευνα θα μπορούσαν να περιλαμβάνουν έναν καλύτερο τρόπο επικοινωνίας μεταξύ των χρηστών όπως είναι κάποιο chat system και κατηγοριοποιήσεις αντικειμένων με βάση κάποια χαρακτηριστικά.

Πίνακας Ορολογίας

ΠΙΝΑΚΑΣ 1 ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος όρος	Ελληνικός Όρος
Agile	Ευκίνητος
Sprint	Σπριντ
Pop Up	Αναδυόμενο
Login	Σύνδεση
Sign Up	Εγγραφή
Home	Σπίτι
Add Item	Προθήκη αντικειμένου
Posted Items	Αναρτημένα αντικείμενα
Edit User	Επεξεργασία χρήστη
Enter your email	Εισάγετε το ηλεκτρονικό σας ταχυδρομείο
Enter your password	Εισάγετε τον κωδικό σας
Password	Κωδικός
Tooltip	Συμβουλή εργαλείου
Error messages	Μηνύματα σφάλματος
Email	Ηλεκτρονικό ταχυδρομείο
Profile	Προφίλ
Go Back	Πήγαινε πίσω
Pick	Επέλεξε
Save Changes	Αποθήκευση αλλαγών
Delete User	Διαγραφή χρήστη
Cancel	Ακύρωση
Deselect Item	Αποεπιλογή αντικειμένου
The are no picked items	Δεν υπάρχουν επιλεγμένα αντικείμενα
Edit Item	Επεξεργασία αντικειμένου
Delete Item	Διαγραφή αντικειμένου
Title	Τίτλος
Country	Χώρα
City	Πόλη
Zip Code	Ταχυδρομικός Κώδικας
Address	Διεύθυνση
Description	Σχόλιο
Client-Server	Πελάτης-Διακοσμητής
Server	Διακοσμητής
Requests	Αιτήματα
Browser	Πρόγραμμα περιήγησης
Type Safety	Ασφάλεια τύπου

Framework	Πλαίσιο
Components	Συστατικά
Material Design	Σχεδιασμός υλικού
Toolbar	Γραμμή εργαλείων
Toasts	Μήνυμα ειδοποίησης
Form fields	Πεδίο φόρμας
Tokens	Κέρματα
Auto login	Αυτόματη σύνδεση
Interceptor	Παρεμβαίνων
Register	Εγγραφή
FormData	Δεδομένα φόρμας
Logout	Αποσύνδεση
Services	Υπηρεσίες
Module	Μονάδα
Paginator	Εργαλείο σελιδοποίησης
Icons	Εικονίδια
Deflater	Αποσυμπιεστής
Inflater	Συμπιεστής
Queries	Ερωτήματα
One to many	Ένα προς πολλά
Transaction	Συναλλαγή
User	Χρήστης
Item	Αντικείμενο
Many to one	Πολλά προς ένα
Design	Σχεδιασμός

Πίνακας Συντμήσεων-Αρτικόλεξων-Ακρονυμίων

ΠΙΝΑΚΑΣ 2: ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

MySQL	My Structured Query Language
Http	Hyper Text Protocol
Url	Uniform Resource Locator
UI	User Interface
JWT	JSON Web Token
API	Application Programming Interface
BLOB	Binary Large Object

Βιβλιογραφία

- Alepis, E., Virvou, M., & Drakoulis, S. (2014). *Human smartphone interaction: Exploring smartphone senses*. Ανάκτηση από <https://ieeexplore.ieee.org/document/6878825>
- Amigoscode-Spring Boot 3 + Spring Security 6 - JWT Authentication and Authorisation [NEW] [2023]. (2023). Ανάκτηση από YouTube: <https://www.youtube.com/watch?v=KxqJblhzfl>
- Angular Js-#41: How to install Angular Material in Angular 14 Application. (2022). Ανάκτηση από YouTube: <https://www.youtube.com/watch?v=jw7-hyCtjfQ>
- Angular Material*. (χ.χ.). Ανάκτηση από <https://material.angular.io/components/icon/overview>
- Codevolution-Angular Material Tutorial - 1 - Introduction. (2019). Ανάκτηση από YouTube: <https://www.youtube.com/watch?v=bV8emCBmFHk&list=PLC3y8-rFHvwilEuCqFGTL5Gt5U6delrsU>
- Coding Tutorials-Angular HTTP Interceptors. (2023). Ανάκτηση από YouTube: <https://www.youtube.com/watch?v=U2jC4JGdD8c>
- Java Guides-[NEW] ReactJS + Spring Boot 3 CRUD Full Stack App - 1 - Project Overview and Architecture. (2023). Ανάκτηση από YouTube: <https://www.youtube.com/watch?v=dP4goCkKxIw&list=PLGRDMO4rOGcNLnW1L2vgsExTBg-VPoZhr>
- Politou, E., Alepis, E., & Patsakis, C. (2017). *A survey on mobile affective computing*. Ανάκτηση από <https://www.sciencedirect.com/science/article/abs/pii/S1574013717300382?via%3Dihub>
- Programming with Mosh-Angular Tutorial for Beginners: Learn Angular & TypeScript. (2017). Ανάκτηση από YouTube: <https://www.youtube.com/watch?v=k5E2AVpwsko>
- Programming with Mosh-TypeScript Tutorial for Beginners. (2017). Ανάκτηση από YouTube: <https://www.youtube.com/watch?v=d56mG7DezGs>
- Ramos, K., Papadimitriou, S., Virvou, M., & Alepis, E. (2019). *A Web-Based Application for Innovative Hospital Appointment Scheduling Using Neural Network*. Ανάκτηση από A Web-Based Application for Innovative Hospital Appointment Scheduling Using Neural Network
-