|  |  |
| --- | --- |
|  | **ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ**  **ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ**  **ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ**  **ΕΡΓΑΣΤΗΡΙΟ ΔΙΑΔΡΑΣΤΙΚΩΝ ΤΕΧΝΟΛΟΓΙΩΝ** |

**ΜΕΛΈΤΗ ΚΑΙ ΑΝΆΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΉΣ ΕΦΑΡΜΟΓΉΣ ΥΠΟΣΤΉΡΙΞΗΣ ΦΟΙΤΗΤΙΚΉΣ ΜΈΡΙΜΝΑΣ ΠΑΝΕΠΙΣΤΗΜΊΟΥ**

**Δ Ι Π Λ Ω Μ Α Τ Ι Κ Η Ε Ρ Γ Α Σ Ι Α**

**ΜΟΥΡΑΤΙΔΗΣ ΝΙΚΟΛΑΟΣ**

**ΕΠΙΒΛΕΠΩΝ: ΑΒΟΥΡΗΣ ΝΙΚΟΛΑΟΣ**

**ΠΑΤΡΑ - ΜΗΝΑΣ ΕΤΟΣ**

Πανεπιστήμιο Πατρών, Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών.

Νικόλαος Μουρατίδης​

© 2025 – Με την επιφύλαξη παντός δικαιώματος

Το σύνολο της εργασίας αποτελεί πρωτότυπο έργο, παραχθέν από τον Νικόλαο Μουρατίδη και δεν παραβιάζει δικαιώματα τρίτων καθ’ οιονδήποτε τρόπο. Αν η εργασία περιέχει υλικό, το οποίο δεν έχει παραχθεί από τον/την ίδιο/α, αυτό είναι ευδιάκριτο και αναφέρεται ρητώς εντός του κειμένου της εργασίας ως προϊόν εργασίας τρίτου, σημειώνοντας με παρομοίως σαφή τρόπο τα στοιχεία ταυτοποίησής του, ενώ παράλληλα βεβαιώνει πως στην περίπτωση χρήσης αυτούσιων γραφικών αναπαραστάσεων, εικόνων, γραφημάτων κ.λπ., έχει λάβει τη χωρίς περιορισμούς άδεια του κατόχου των πνευματικών δικαιωμάτων για την συμπερίληψη και επακόλουθη δημοσίευση του υλικού αυτού.

**ΠΙΣΤΟΠΟΙΗΣΗ**

Πιστοποιείται ότι η Διπλωματική Εργασία με τίτλο

**ΜΕΛΈΤΗ ΚΑΙ ΑΝΆΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΉΣ ΕΦΑΡΜΟΓΉΣ ΥΠΟΣΤΉΡΙΞΗΣ ΦΟΙΤΗΤΙΚΉΣ ΜΈΡΙΜΝΑΣ ΠΑΝΕΠΙΣΤΗΜΊΟΥ**

του φοιτητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

**ΜΟΥΡΑΤΙΔΗΣ ΝΙΚΟΛΑΟΣ** ΤΟΥ **ΓΕΩΡΓΙΟΥ**

Αριθμός Μητρώου: 1019943

Παρουσιάστηκε δημόσια στο Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών στις

…….../……../………

και εξετάστηκε από την ακόλουθη εξεταστική επιτροπή:

Αβούρης Νικόλαος, Καθηγητής, ΤΗΜΤΥ (επιβλέπων)

Θραμπουλίδης Κλεάνθης, Καθηγητής, ΤΗΜΤΥ (μέλος επιτροπής)

|  |  |  |
| --- | --- | --- |
| Ο/Η Επιβλέπων/ουσα |  | Ο/Η Διευθυντής/τρια του Τομέα |
|  |  |  |
| Αβούρης Νικόλαος |  | Καλύβας Γρηγόριος |
| Καθηγητής |  | Καθηγητής |

**ΠΡΟΛΟΓΟΣ**

Ο πρόλογος είναι προαιρετικός. Εδώ ο συγγραφέας βάζει τυχόν πληροφορίες που δεν έχουν άμεση σχέση με το επιστημονικό περιεχόμενο της Διπλωματικής Εργασίας, όπως ευχαριστίες, κτλ.

Η δομή, η μορφή και η έκταση του προλόγου είναι στη διακριτική ευχέρεια του φοιτητή, εκτός αν προδιαγράψει διαφορετικά ο επιβλέπων.

**ΠΕΡΙΛΗΨΗ**

**ΜΕΛΈΤΗ ΚΑΙ ΑΝΆΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΉΣ ΕΦΑΡΜΟΓΉΣ ΥΠΟΣΤΉΡΙΞΗΣ ΦΟΙΤΗΤΙΚΉΣ ΜΈΡΙΜΝΑΣ ΠΑΝΕΠΙΣΤΗΜΊΟΥ**

|  |  |
| --- | --- |
| **ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΦΟΙΤΗΤΗ:**  **Μουρατίδης Νικόλαος** | **ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΕΠΙΒΛΕΠΟΝΤΟΣ:**  **Νικόλαος Αβούρης** |

Η παρούσα διπλωματική επικεντρώνεται στον σχεδιασμό και την ανάπτυξη μίας διαδικτυακής εφαρμογής που έχει ως στόχο τη βελτιστοποίηση της διαδικασίας διαχείρισης των αιτήσεων και της εισόδου στο εστιατόριο της φοιτιτικής μέριμνας του Πανεπιστημίου Πατρών. Η διαχείριση αυτή περιλαμβάνει την υποβολή και τον έλενχο των αιτήσεων δωρεάν σίτισης, την πώληση κουπονιών καθώς και τον ελένχο κατα την είσοδο στο εστιατόριο.

Στην εργασία αναλύονται λεπτομερώς τα βήματα που ακολουθήθηκαν για τον σχεδιασμό της εφαρμογής, περιλαμβάνοντας τις λειτουργίες που αναπτύχθηκαν, καθώς και τις τεχνολογίες και τις γλώσσες προγραμματισμού που χρησιμοποιήθηκαν.

Λέξεις κλειδιά: Ιστοσελίδα, Διαδικτυακή Εφαρμογή, Διπλωματική Εργασία, Διπλωματική, Διαχείριση Αιτήσεων Σίτησης, Ελένχος Είσοδου

**EXTENSIVE ENGLISH SUMMARY**

**STUDY AND DEVELOPMENT OF A UNIVERSITY STUDENT CARE SUPPORT WEB APPLICATION**

|  |  |
| --- | --- |
| **STUDENT NAME, SURNAME:**  **Nikolaos Mouratidis** | **SUPERVISOR NAME, SURNAME:**  **Nikolaos Avouris** |

The present diploma thesis focuses on the design and development of a web application aimed at optimizing the management process of the applications and the entry of the university restaurant. This management includes submission and checking of the application for free entry pass , purchasing of coupons and entry checking on the restaurant.

The paper provides a detailed analysis of the steps followed for the design of the application, including the functions that were developed, as well as the technologies and programming languages that were used.

Keywords: Website, Web Application, Diploma Thesis, Thesis, Application Management,Entry checking

# Ευχαριστίες

Θα ήθελα καταρχάς να ευχαριστήσω τον κ. Νικόλαο Αβούρη για την επίβλεψη αυτής της διπλωματικής εργασίας. Τον ευχαριστώ θερμά για την ευκαιρία και την εμπιστοσύνη που μου έδειξε στο να υλοποιήσω μια δική μου ιδέα στα πλαίσια της διπλωματικής εργασίας. Επί προσθέτως, ανεκτίμητες ήταν οι συζητήσεις που πραγματοποιόντουσαν ανά τακτά χρονικά διαστήματα οι οποίες συντέλεσαν στην ομαλή ολοκλήρωση αυτής της εργασίας.

Επιπλέον, θα ήθελα να εκφράσω τις ευχαριστίες μου προς τους φίλους και γνωστόυς μου που με στήριξαν καθ΄ όλο αυτό το διάστημα αλλά και για τη συμμετοχή τους στο κομμάτι της αξιολόγησης χρηστών.

Τέλος, θα ήθελα να ευχαριστήσω τους γονείς μου για τη στήριξη και την ηθική συμπαράσταση που μου προσέφεραν όλα αυτά τα χρόνια.

Πίνακας Περιεχομένων

[Ευχαριστίες vii](#__RefHeading___Toc2458_2940574846)

[ΚΕΦΑΛΑΙΟ 1 - ΕΙΣΑΓΩΓΗ 13](#__RefHeading___Toc4485_4248105787)

[1.1 Σκοπός 13](#__RefHeading___Toc4721_4248105787)

[1.2 Τρέχουσά Κατάσταση 13](#__RefHeading___Toc2262_2940574846)

[1.2.1 Χρήστες κουπονίων 13](#__RefHeading___Toc214477_3614309570)

[1.2.2 Χρήστες δωρεάν σίτισης 13](#__RefHeading___Toc214479_3614309570)

[1.2.2.1 Μη Σωστός Έλεγχος Αποδοχής Προσωπικών Δεδομένων 14](#__RefHeading___Toc214481_3614309570)

[1.2.2.2 Περιορισμοί στην Υποβολή Δικαιολογητικών της Ίδιας Κατηγορίας 14](#__RefHeading___Toc214483_3614309570)

[1.2.2.3 Ανεπαρκείς Περιγραφές Δικαιολογητικών 15](#__RefHeading___Toc214485_3614309570)

[1.2.2.4 Υποχρεωτικά έγγραφα 15](#__RefHeading___Toc214487_3614309570)

[1.3 Μεθοδολογία 16](#__RefHeading___Toc2264_2940574846)

[1.4 Διάρθρωση της Διπλωματικής Εργασίας 16](#__RefHeading___Toc2266_2940574846)

[ΚΕΦΑΛΑΙΟ 2 - Σχεδιασμός Πλατφόρμας 18](#__RefHeading___Toc2268_2940574846)

[2.1 Ανάλυση Απαιτήσεων \*\*\* 18](#__RefHeading___Toc2270_2940574846)

[2.2 Καταγραφή μοντέλου δεδομένων \*\*\* 18](#__RefHeading___Toc2272_2940574846)

[2.2.1 ERD 18](#__RefHeading___Toc2274_2940574846)

[2.2.2 Σχεσιακό μοντέλο­ 19](#__RefHeading___Toc2276_2940574846)

[2.3 Χάρτης πλόηγησης και σενάρια χρήσης \*\*\* 23](#__RefHeading___Toc214489_3614309570)

[ΚΕΦΑΛΑΙΟ 3 - Αρχιτεκτονική Πλατφόρμας 24](#__RefHeading___Toc4725_4248105787)

[3.1 Ορισμοί 24](#__RefHeading___Toc3386_1895780320)

[3.2 Τεχνολογίες που χρησιμοποιήθηκαν 25](#__RefHeading___Toc1350_958307911)

[3.2.1 Node Package Manager 25](#__RefHeading___Toc4737_4248105787_Copy_1)

[3.2.2 Λογισμικό Vite 25](#__RefHeading___Toc2280_2940574846)

[3.2.3 Javascript 25](#__RefHeading___Toc2282_2940574846)

[3.2.4 Vue.js --- -+ 26](#__RefHeading___Toc4729_4248105787)

[3.2.5 Vuex 26](#__RefHeading___Toc3388_1895780320)

[3.2.6 Vuetify --- -+ 27](#__RefHeading___Toc4731_4248105787)

[3.2.7 Vue-i18n​ 27](#__RefHeading___Toc214491_3614309570)

[3.2.8 vue-router 28](#__RefHeading___Toc214493_3614309570)

[3.2.9 laravel-echo 28](#__RefHeading___Toc214495_3614309570)

[3.2.10 axios 28](#__RefHeading___Toc214497_3614309570)

[3.2.11 CSS 28](#__RefHeading___Toc4733_4248105787)

[3.2.12 GitHub \*\*\* 29](#__RefHeading___Toc2462_2940574846)

[3.2.13 Apache2 29](#__RefHeading___Toc1249_958307911)

[3.2.14 MySQL --- -+ 29](#__RefHeading___Toc4735_4248105787)

[3.2.15 PHP \*\*\* 30](#__RefHeading___Toc2464_2940574846)

[3.2.16 PhpMyAdmin ---- -+ 30](#__RefHeading___Toc4737_4248105787)

[3.2.17 Composer 31](#__RefHeading___Toc4737_4248105787_Copy_2)

[3.2.18 Laravel 32](#__RefHeading___Toc1251_958307911)

[3.2.19 Laravel/Telescope \*\*\* 32](#__RefHeading___Toc3390_1895780320)

[3.2.20 fakerphp/faker \*\*\* 33](#__RefHeading___Toc2468_2940574846)

[3.2.21 barryvdh/laravel-dompdf \*\*\* 33](#__RefHeading___Toc2470_2940574846)

[3.2.22 tightenco/ziggy \*\*\* 33](#__RefHeading___Toc2472_2940574846)

[3.2.23 propaganistas/laravel-phone \*\*\* 33](#__RefHeading___Toc2474_2940574846)

[3.2.24 laravel/sanctum \*\*\* 34](#__RefHeading___Toc2476_2940574846)

[3.2.25 laravel/ui \*\*\* 34](#__RefHeading___Toc2478_2940574846)

[3.3 Οργάνωση αρχείων framework Laravel 34](#__RefHeading___Toc3392_1895780320)

[3.3.1 Ο Κεντρικός Φάκελος (Root Directory) 34](#__RefHeading___Toc3394_1895780320)

[3.3.1.1 Ο φάκελος App 34](#__RefHeading___Toc3396_1895780320)

[3.3.1.2 The Bootstrap Directory 34](#__RefHeading___Toc3398_1895780320)

[3.3.1.3 The Config Directory 34](#__RefHeading___Toc3400_1895780320)

[3.3.1.4 The Database Directory 35](#__RefHeading___Toc3402_1895780320)

[3.3.1.5 Φάκελος docs 35](#__RefHeading___Toc3404_1895780320)

[3.3.1.6 Φάκελος lang 36](#__RefHeading___Toc3406_1895780320)

[3.3.1.7 The node\_modules Directory 36](#__RefHeading___Toc3408_1895780320)

[3.3.1.8 The Public Directory 36](#__RefHeading___Toc3410_1895780320)

[3.3.1.9 The Resources Directory 36](#__RefHeading___Toc3412_1895780320)

[3.3.1.10 The Routes Directory 36](#__RefHeading___Toc3414_1895780320)

[3.3.1.11 The Storage Directory 37](#__RefHeading___Toc3416_1895780320)

[3.3.1.12 The Tests Directory 37](#__RefHeading___Toc3418_1895780320)

[3.3.1.13 The Vendor Directory 37](#__RefHeading___Toc3420_1895780320)

[3.3.1.14 Μεμονομένα αρχεία 37](#__RefHeading___Toc3422_1895780320)

[3.3.2 The App Directory 38](#__RefHeading___Toc3424_1895780320)

[3.3.2.1 The Broadcasting Directory 39](#__RefHeading___Toc3426_1895780320)

[3.3.2.2 The Console Directory 39](#__RefHeading___Toc3428_1895780320)

[3.3.2.3 The Enum Directory 39](#__RefHeading___Toc3430_1895780320)

[3.3.2.4 The Events Directory 39](#__RefHeading___Toc3432_1895780320)

[3.3.2.5 The Exceptions Directory 39](#__RefHeading___Toc3434_1895780320)

[3.3.2.6 The Http Directory 39](#__RefHeading___Toc3436_1895780320)

[3.3.2.7 The Jobs Directory 39](#__RefHeading___Toc3438_1895780320)

[3.3.2.8 The Listeners Directory 39](#__RefHeading___Toc3440_1895780320)

[3.3.2.9 The Mail Directory 40](#__RefHeading___Toc3442_1895780320)

[3.3.2.10 The Models Directory 40](#__RefHeading___Toc3444_1895780320)

[3.3.2.11 The Notifications Directory 40](#__RefHeading___Toc3446_1895780320)

[3.3.2.12 The Observers Directory 40](#__RefHeading___Toc3448_1895780320)

[3.3.2.13 The Policies Directory 40](#__RefHeading___Toc3450_1895780320)

[3.3.2.14 The Providers Directory 40](#__RefHeading___Toc3452_1895780320)

[3.3.2.15 The Rules Directory 41](#__RefHeading___Toc3454_1895780320)

[3.3.2.16 The Services Directory 41](#__RefHeading___Toc3456_1895780320)

[3.3.2.17 The Traits Directory 41](#__RefHeading___Toc3458_1895780320)

[3.3.3 The resources/js Directory 41](#__RefHeading___Toc3460_1895780320)

[ΚΕΦΑΛΑΙΟ 4 - ΣΧΕΔΙΑΣΜΟΣ & ΑΝΑΠΤΥΞΗ – ΕΦΑΡΜΟΓΗΣ 43](#__RefHeading___Toc4743_4248105787)

[4.1 ΚΑΘΟΡΙΣΜΟΣ ΣΤΟΧΟΥ - 43](#__RefHeading___Toc1255_958307911)

[4.2 ΟΡΙΣΜΟΣ ΠΡΟΔΙΑΓΡΑΦΩΝ -+ 44](#__RefHeading___Toc4747_4248105787)

[4.3 ΧΡΗΣΤΕΣ ΕΦΑΡΜΟΓΗΣ 45](#__RefHeading___Toc1352_958307911)

[4.3.1 Ελεγκτής Αιτήσεων 45](#__RefHeading___Toc1354_958307911)

[4.3.2 Ελεγκτής Αιτήσεων 45](#__RefHeading___Toc1259_958307911)

[4.3.2.1 Διαχείριση Δικαιολογητικών 45](#__RefHeading___Toc1345_958307911)

[4.3.2.2 Αξιολόγηση Αίτησης 45](#__RefHeading___Toc1347_958307911)

[4.3.3 Υπάλληλος Υποδοχής Καρτών 46](#__RefHeading___Toc1356_958307911)

[4.3.3.1 Διαδικασία Ελέγχου: 46](#__RefHeading___Toc1358_958307911)

[4.3.3.2 Δυνατότητες: 46](#__RefHeading___Toc1360_958307911)

[4.3.4 Πωλητής Κουπονιών 46](#__RefHeading___Toc1362_958307911)

[4.3.4.1 Διαδικασία Πώλησης: 46](#__RefHeading___Toc1364_958307911)

[4.3.4.2 Δυνατότητες: 46](#__RefHeading___Toc1366_958307911)

[4.3.5 Ελεγκτής Αιτήσεων Δωρεάν Σίτισης 46](#__RefHeading___Toc1368_958307911)

[4.3.5.1 Διαδικασία Ελέγχου 46](#__RefHeading___Toc1370_958307911)

[4.3.5.2 Δυνατότητες: 47](#__RefHeading___Toc1372_958307911)

[4.3.6 Δικαιούχος Σίτισης 47](#__RefHeading___Toc1261_958307911)

[4.3.6.1 Φοιτητής 47](#__RefHeading___Toc1263_958307911)

[4.3.6.1.1 Δικαιώματα Σίτισης: 47](#__RefHeading___Toc1374_958307911)

[4.3.6.1.2 Δυνατότητες: 47](#__RefHeading___Toc1376_958307911)

[4.3.6.2 Προσωπικό Δ.Ε.Π./Δ.Τ./Δ.Π. 47](#__RefHeading___Toc1265_958307911)

[4.3.6.2.1 Δικαιώματα Σίτισης: 47](#__RefHeading___Toc1378_958307911)

[4.3.6.2.2 Δυνατότητες: 47](#__RefHeading___Toc1380_958307911)

[4.4 Πορεία Αίτησης Δωρεάν Σίτησης 48](#__RefHeading___Toc1267_958307911)

[4.4.1 Υποβολή Αίτησης από τον Φοιτητή 48](#__RefHeading___Toc1269_958307911)

[4.4.1.1 Έλεγχος Στοιχείων: 48](#__RefHeading___Toc1382_958307911)

[4.4.1.2 Κατάθεση Δικαιολογητικών: 48](#__RefHeading___Toc1384_958307911)

[4.4.1.3 Προβολή Κατατεθημένης αίτησης 48](#__RefHeading___Toc1386_958307911)

[4.4.2 Έλεγχος Αίτησης από τον Ελεγκτή 48](#__RefHeading___Toc1271_958307911)

[4.4.2.1 Αποφάσεις για τα Δικαιολογητικά: 48](#__RefHeading___Toc1388_958307911)

[4.4.2.2 Αποφάσεις για την Αίτηση: 48](#__RefHeading___Toc1390_958307911)

[4.5 Επεξεργασία Ελλιπούς Αίτησης από τον Φοιτητή 50](#__RefHeading___Toc1392_958307911)

[4.6 Χρήση της Έγκρισης για Σίτιση 50](#__RefHeading___Toc1275_958307911)

[4.7 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ --- 51](#__RefHeading___Toc4751_4248105787)

[4.7.1 Γέμισμα Βάσης Δεδομένων --- 51](#__RefHeading___Toc4757_4248105787)

[4.8 ΛΕΙΤΟΥΡΓΙΕΣ ΕΦΑΡΜΟΓΗΣ --- 53](#__RefHeading___Toc4759_4248105787)

[4.8.1 Διαχείριση Πρώτης Σύνδεσης Χρήστη 53](#__RefHeading___Toc4761_4248105787)

[4.8.2 Διαχείριση Ελέγχου Εισόδου μέσω Συστήματος Barcode 54](#__RefHeading___Toc1279_958307911)

[4.8.2.1 Διαδικασία Ελέγχου: 54](#__RefHeading___Toc1281_958307911)

[4.8.2.2 Επιπλέον Δυνατότητες για Ελεγκτή: 55](#__RefHeading___Toc1283_958307911)

[4.8.3 Δυνατότητα Εξαγωγής Στατιστικών 56](#__RefHeading___Toc1285_958307911)

[4.8.3.1 Απεικόνιση Στατιστικών: 58](#__RefHeading___Toc1287_958307911)

[4.8.4 Πώληση κουπονιών 58](#__RefHeading___Toc1289_958307911)

[4.8.4.1 Καταχώριση Αγοράς Κουπονιών: 58](#__RefHeading___Toc1291_958307911)

[4.8.4.2 Επιλογές Διαχείρισης Συναλλαγής - Επιβεβαίωση: 58](#__RefHeading___Toc1293_958307911)

[4.8.4.3 Επιλογές Διαχείρισης Συναλλαγής - Ολοκλήρωση: 59](#__RefHeading___Toc1293_958307911_Copy_1)

[4.8.5 Λειτουργίες Κατόχου Κουπονιών 60](#__RefHeading___Toc1295_958307911)

[4.8.5.1 Ιστορικό Συναλλαγών 60](#__RefHeading___Toc1297_958307911)

[4.8.5.2 Μεταφορά Κουπονιών 61](#__RefHeading___Toc1299_958307911)

[4.8.5.3 Ενημέρωση Υπολοίπου 62](#__RefHeading___Toc1301_958307911)

[4.8.6 Λειτουργίες Δικαιούχου Δωρεάν Σίτισης 62](#__RefHeading___Toc1303_958307911)

[4.8.6.1 Παρακολούθηση Τρέχουσας Αίτησης 62](#__RefHeading___Toc1305_958307911)

[4.8.6.2 Ιστορικό Δωρεάν Σίτισης 62](#__RefHeading___Toc1307_958307911)

[4.8.7 Υποβολη Αιτήσεων Δωρεάν Σίτισης 62](#__RefHeading___Toc1309_958307911)

[4.8.7.1 Τρέχουσα Αίτηση 62](#__RefHeading___Toc1311_958307911)

[4.8.7.2 Διαδικασία για Νέα Αίτηση 62](#__RefHeading___Toc1313_958307911)

[4.8.8 Έλεγχος Αιτήσεων Δωρεάν Σίτισης 67](#__RefHeading___Toc1315_958307911)

[4.8.8.1 Διαχείριση Δικαιολογητικών 67](#__RefHeading___Toc1317_958307911)

[ΚΕΦΑΛΑΙΟ 5 - ΑΞΙΟΛΟΓΗΣΗ ΕΦΑΡΜΟΓΗΣ --- 69](#__RefHeading___Toc1321_958307911)

[5.1 ΜΕΘΟΔΟΣ ΑΞΙΟΛΟΓΗΣΗΣ --- 69](#__RefHeading___Toc1323_958307911)

[5.2 ΧΡΗΣΤΕΣ ΑΞΙΟΛΟΓΗΣΗΣ --- 69](#__RefHeading___Toc1325_958307911)

[5.3 ΣΕΝΑΡΙΑ ΧΡΗΣΗΣ --- 69](#__RefHeading___Toc1327_958307911)

[5.4 ΑΠΟΤΕΛΕΣΜΑΤΑ --- 69](#__RefHeading___Toc1329_958307911)

[ΚΕΦΑΛΑΙΟ 6 - ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΕΠΕΚΤΑΣΕΙΣ --- 70](#__RefHeading___Toc1331_958307911)

[6.1 ΣΥΜΠΕΡΑΣΜΑΤΑ --- 70](#__RefHeading___Toc1333_958307911)

[6.2 ΒΕΛΤΙΩΣΕΙΣ & ΕΠΕΚΤΑΣΕΙΣ --- 70](#__RefHeading___Toc1335_958307911)

[ΚΕΦΑΛΑΙΟ 7 - ΒΙΒΛΙΟΓΡΑΦΙΑ 71](#__RefHeading___Toc1337_958307911)

Πίνακας Εικόνων

[*Figure 1.1:Τρέχουσα υλοποίηση - Πάνω μέρος* 14](#Figure!0|sequence)

[Figure 1.2: Τρέχουσα υλοποίηση – Κάτω μέρος 14](#Figure!1|sequence)

[Figure 1.3: Τρέχουσα υλοποίηση - επιπλέον δικαιολογητικά 15](#Figure!2|sequence)

# ΕΙΣΑΓΩΓΗ

## Σκοπός

Σκοπός της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη μιας διαδικτυακής εφαρμογής που θα βελτιώσει τη διαδικασία διαχείρισης και υποστήριξης φοιτητικής μέριμνας στο Πανεπιστήμιο Πατρών. Η εφαρμογή αποσκοπεί στην αντιμετώπιση κενών και δυσλειτουργιών στη διαδικασία αιτήσεων δωρεάν σίτισης και στην αγορά κουπονιών, καθιστώντας τις διαδικασίες αυτές πιο εύχρηστες και αποτελεσματικές.

Πηγή έμπνευσης για τη διπλωματική εργασία αποτέλεσε η προσωπική μου εμπειρία με τη γραφειοκρατία κατά την υποβολή αιτήσεων δωρεάν σίτισης. Επιλέγοντας το συγκεκριμένο θέμα, στόχος μου ήταν να επιτύχω ορισμένες βελτιώσεις στις διαδικασίες σίτισης, όπως:

* Η εξάλειψη της φυσικής παρουσίας για την υποβολή της αίτησης και την παραλαβή της κάρτας σίτισης.
* Η εξάλειψη της ανάγκης για εκτυπωμένα έγγραφα κατά την υποβολή αιτήσεων.
* Η αποφυγή της συσσώρευσης έντυπων κουπονιών.

Στην πλατφόρμα αυτή, ο κάθε χρήστης θα μπορεί:

* Να υποβάλλει την αίτησή του μαζί με τα απαραίτητα δικαιολογητικά.
* Να ενημερώνεται για την κατάσταση της αίτησής του και να βλέπει το ιστορικό των κινήσεών του.
* Να παρακολουθεί το υπόλοιπο των κουπονιών του.
* Να αποστέλλει κουπόνια σε άλλους χρήστες της ίδιας κατηγορίας.

Από την άλλη πλευρά, το προσωπικό της φοιτητικής μέριμνας θα έχει τη δυνατότητα:

* Να ελέγχει τις υποβληθείσες αιτήσεις και να αποφασίζει για την έγκριση ή απόρριψή τους.

Το προσωπικό στο εστιατόριο θα μπορεί:

* Να ελέγχει αν ένα άτομο δικαιούται να σιτιστεί.
* Να πουλά κουπόνια.

Για την είσοδο στο εστιατόριο, θα χρησιμοποιείται ο αριθμός της ακαδημαϊκής ταυτότητας, αντικαθιστώντας την ανάγκη για φυσικά κουπόνια.

Η ανάπτυξη της πλατφόρμας θα γίνει με γνώμονα τις παρακάτω αρχές:

* Αξιοπιστία: Εξασφάλιση της ορθής λειτουργίας όλων των διαδικασιών, χωρίς σφάλματα ή καθυστερήσεις.
* Ασφάλεια: Προστασία των δεδομένων των χρηστών από μη εξουσιοδοτημένη πρόσβαση.
* Ευχρηστία: Παροχή ενός φιλικού και απλού στη χρήση περιβάλλοντος για όλους τους χρήστες.
* Συμβατότητα: Εξασφάλιση ομαλής λειτουργίας σε διάφορες συσκευές, όπως κινητά τηλέφωνα, tablets και υπολογιστές.

## Τρέχουσά Κατάσταση

### Χρήστες κουπονίων

Οι χρήστες που δεν δικαιούνται δωρεάν σίτιση οφείλουν να αγοράζουν κουπόνια για να εισέρχονται στο εστιατόριο. Τα κουπόνια έχουν τη μορφή μικρών χαρτιών και χρειάζεται να αγοράζονται σε ομάδες πολλών τεμαχίων. Αυτή η διαδικασία αποτελεί μια βασική αλλά περιοριστική λύση για την πρόσβαση στις υπηρεσίες σίτισης.

### Χρήστες δωρεάν σίτισης

Η υποβολή αιτήσεων για δωρεάν σίτιση απαιτούσε, μέχρι πριν από λίγα χρόνια, τη δια ζώσης υποβολή δικαιολογητικών​. Ωστόσο, η διαδικασία έχει αναβαθμιστεί και πλέον γίνεται ηλεκτρονικά.

*Figure 1:Τρέχουσα υλοποίηση - Πάνω μέρος*

Figure 2: Τρέχουσα υλοποίηση – Κάτω μέρος

Τα Κοινά δικαιολογητικά [1] είναι :

Figure 3: Τρέχουσα υλοποίηση - επιπλέον δικαιολογητικά

* Ταυτοποίηση:
  + Αντίγραφο ακαδημαϊκής ταυτότητας
  + Φωτοαντίγραφο ταυτότητας ή διαβατηρίου (και οι δύο όψεις)

Επιπλέον Δικαιολογητικά (κατά περίπτωση):

* Οικονομική Κατάσταση:
  + Πράξη προσδιορισμού φόρου και φορολογική δήλωση (Ε1) ή βεβαίωση οικογενειακού εισοδήματος από την αρμόδια δημόσια οικονομική υπηρεσία της χώρας τους.
* Φοιτητική Κατάσταση:
  + Υπεύθυνη δήλωση φοιτητή/τριας του Ν.1599/86
  + Βεβαίωση χορήγησης υποτροφίας
  + Βεβαίωση από το Τμήμα Διεθνών Σχέσεων του Πανεπιστημίου Πατρών
  + Πιστοποιητικό ενεργού φοίτησης από το Τμήμα Σπουδών του φοιτητή και αλλών μελών της οικογένιας
* Οικογενειακή Κατάσταση:
  + Πρόσφατο πιστοποιητικό οικογενειακής κατάστασης
  + Ληξιαρχική πράξη θανάτου γονέα (αν υπάρχει)
  + Φωτοαντίγραφο διαζυγίου και/ή συμφωνητικό επιμέλειας τέκνων (αν υπάρχει)
  + Υπεύθυνη δήλωση γονέων του Ν.1599/86

Ο φοιτητής, κατά τη διάρκεια της αίτησης, καλείται να υποβάλει τα απαραίτητα έγγραφα για να εγκριθεί ή όχι η αίτησή του για δωρεάν σίτιση. Κατά την ανάπτυξη και συγγραφή της διπλωματικής, παρατηρήθηκαν αρκετά προβλήματα σχετικά με τη διαδικασία, τα οποία εντοπίστηκαν σε συγκεκριμένα σημεία:

* Η αποδοχή χρήσης προσωπικών δεδομένων.
* Η υποβολή δικαιολογητικών, η οποία φαίνεται να σχεδιάστηκε έχοντας υπόψη την οικογένεια ενός Έλληνα φοιτητή, όπου για τον υπολογισμό του οικογενειακού εισοδήματος υπάρχει μόνο μία πράξη προσδιορισμού φόρου.

#### Μη Σωστός Έλεγχος Αποδοχής Προσωπικών Δεδομένων

Ένα σφάλμα στην υλοποίηση του τρέχοντος συστήματος (sitisi.upatras.gr) είναι η μη υποχρεωτική αποδοχή των όρων χρήσης προσωπικών δεδομένων. Αυτό παρατηρήθηκε όταν μπόρεσα να συνδεθώ απευθείας στη σελίδα υποβολής εγγράφων (https://sitisi.upatras.gr/sitisi/index\_panepistimio.php) και να υποβάλω αίτηση χωρίς να χρειαστεί να αποδεχθώ τους όρους.

#### Περιορισμοί στην Υποβολή Δικαιολογητικών της Ίδιας Κατηγορίας

Το σύστημα επιτρέπει την υποβολή ενός μόνο εγγράφου ανά κατηγορία (ακόμα και στην κατηγορία «αλλο»), γεγονός που δημιουργεί προβλήματα σε περιπτώσεις όπου απαιτούνται περισσότερα του ενός έγγραφα. Ενδεικτικά παραδείγματα περιλαμβάνουν:

* Φορολογικές δηλώσεις από περισσότερα του ενός μέλη της οικογένειας, όπως γονείς, φοιτητές ή αδέρφια εφόσον θεορούντε προστατεύομενα.
* Θάνατοι και των δύο γονέων, όπου απαιτούνται δύο ληξιαρχικές πράξεις θανάτου.
* Πολλαπλές υπεύθυνες δηλώσεις γονέων που βρίσκονται σε διάσταση..
* Η βεβαίωση σπουδών αδερφών.

#### Ανεπαρκείς Περιγραφές Δικαιολογητικών

Το σύστημα περιλαμβάνει προκαθορισμένες περιγραφές δικαιολογητικών για συγκεκριμένες κατηγορίες φοιτητών. Ωστόσο, παρατηρείται ότι:

* Δεν καλύπτονται όλες οι πιθανές περιπτώσεις, όπως:
  + Βεβαιώσεις από το τμήμα διεθνών σχέσεων για φοιτητές Erasmus.
  + Βεβαιώσεις υποτροφιών από το Υπουργείο Παιδείας για αλλοδαπούς φοιτητές.
* Περιγραφές που είναι κατάλληλες μόνο για Έλληνες φοιτητές χρησιμοποιούνται γενικά, όπως η αναφορά σε ελληνικές φορολογικές δηλώσεις (π.χ., ΦΕΦΠ/εκκαθαριστικό), ενώ τα αντίστοιχα για φοιτητές από άλλες χώρες έχουν διαφορετική ονομασία ( βεβαίωση της αρμόδιες δημόσιας οικονομικής υπηρεσίας της χώρας τους) η οποία απουσιάζει. Ενώ μια γενικότερη περιγραφή θα μπορούσε να οικανοποιει ολες τις περιπτώσεις

#### Υποχρεωτικά έγγραφα

Τα κοινά έγγραφα που απαιτούνται ανεξαρτήτως κατηγορίας φοιτητών είναι δύο, η ακαδημαϊκή ταυτότητα και η εθνική ταυτότητα. Ωστόσο, στην υπάρχουσα υλοποίηση του συστήματος, υπάρχουν συνολικά πέντε υποχρεωτικά έγγραφα, τα οποία μπορεί να μην είναι κατάλληλα ή απαραίτητα για όλες τις κατηγορίες φοιτητών. Σε ορισμένες περιπτώσεις, χρειάζετε η υποβολή λιγότερων εκ των 5 δικαιολογιτικών.

Για παράδειγμα, οι φοιτητές από την Κύπρο, αλλοδαποί ή τέκνα Ελλήνων του εξωτερικού, υποβάλλουν μόνο μία βεβαίωση από την οικονομική υπηρεσία της χώρας τους. Αυτή η βεβαίωση αντικαθιστά το ένα εκ των δύο υποχρεωτικών οικονομικών εγραφων ( πχ ΦΕΦΠ), το οποίο θεωρείται υποχρεωτικό στο σύστημα. Παρόλα αυτά, το σύστημα απαιτεί τη συμπλήρωση της θέσης για το Ε1, αναγκάζοντας τους φοιτητές να ανεβάσουν ασχετο δικαιολογιτικό στην θέση του (πχ το πιστοποιητικό οικογενειακής κατάστασης), ακόμα και αν η περιγραφή του συγκεκριμένου εγγράφου εμφανίζεται στις επιλογές για τα μη υποχρεωτικά έγγραφα.

Παρομοίως, οι φοιτητές Erasmus, οι οποίοι δεν χρειάζεται να υποβάλουν έγγραφα σχετικά με την οικονομική τους κατάσταση, αναγκάζονται να ανεβάσουν άλλα έγγραφα στις θέσεις αυτών των δύο υποχρεωτικών δικαιολογητικών. Στην περίπτωση αυτή, απαιτείται η βεβαίωση από το τμήμα διεθνών σχέσεων καθώς και το πιστοποιητικό ενεργούς φοίτησης από το τμήμα σπουδών.

Για τους φοιτητές αυτούς, όπως και για τους αλλοδαπούς υπότροφους, ο αριθμός των απαιτούμενων δικαιολογητικών είναι γενικά μικρότερος. Παρόλα αυτά, το σύστημα συχνά τους αναγκάζει να υποβάλουν κενά ή διπλά έγγραφα για να ολοκληρωθεί η αίτησή τους. Αυτό το πρόβλημα αναδεικνύει την ανάγκη για προσαρμογή των απαιτήσεων ανάλογα με την κατηγορία του φοιτητή, ώστε να διασφαλιστεί η ορθότητα και η ευκολία στη διαδικασία υποβολής.

## Μεθοδολογία

Για την επίτευξη των στόχων της διπλωματικής εργασίας, ακολουθήθηκε μια συστηματική προσέγγιση που περιλαμβάνει:

* **Ανάλυση Υφιστάμενων Διαδικασιών**:
  + Μελέτη της παρούσας κατάστασης στις υπηρεσίες φοιτητικής μέριμνας του Πανεπιστημίου Πατρών.
  + Καταγραφή προβλημάτων και αναγκών που παρατηρήθηκαν στις διαδικασίες.
* **Σχεδιασμός Λύσης**:
  + Προσδιορισμός λειτουργικών και μη λειτουργικών απαιτήσεων της εφαρμογής.
  + Σχεδιασμός της αρχιτεκτονικής της πλατφόρμας, συμπεριλαμβανομένων της βάσης δεδομένων και του περιβάλλοντος χρήστη.
* **Ανάπτυξη**:
  + Υλοποίηση της διαδικτυακής εφαρμογής χρησιμοποιώντας τεχνολογίες όπως JavaScript, Vue.js, Laravel, και MySQL.
* **Δοκιμές και Αξιολόγηση**:
  + Εκτέλεση λειτουργικών δοκιμών για την επιβεβαίωση της σωστής λειτουργίας της εφαρμογής.
  + Συλλογή ανατροφοδότησης από πιλοτικούς χρήστες για βελτιώσεις.

Η προσέγγιση αυτή διασφαλίζει ότι το τελικό προϊόν είναι αποτελεσματικό, εύχρηστο και ικανοποιεί τις ανάγκες των χρηστών.

## Διάρθρωση της Διπλωματικής Εργασίας

Η διπλωματική εργασία δομείται ως εξής:

* **Κεφάλαιο 1**: Εισαγωγή. Παρουσιάζονται η τρέχουσα κατάσταση, η μεθοδολογία, και η διάρθρωση της εργασίας.
* **Κεφάλαιο 2**: Σχεδιασμός Πλατφόρμας. Περιγράφονται η ανάλυση απαιτήσεων, το μοντέλο δεδομένων, οι δοσοληψίες, και οι σενάρια χρήσης.
* **Κεφάλαιο 3**: Αρχιτεκτονική Πλατφόρμας. Παρουσιάζεται η αρχιτεκτονική σχεδίαση, οι τεχνολογίες που χρησιμοποιήθηκαν και η οργάνωση των αρχείων.
* **Κεφάλαιο 4**: Περιγραφή Υλοποίησης. Εξηγούνται ο αλγόριθμος δημιουργίας στιγμιοτύπων και η υλοποίηση της γραφικής διεπαφής.
* **Κεφάλαιο 5**: Έλεγχος και Αξιολόγηση. Καταγράφονται οι μέθοδοι ελέγχου, η αξιολόγηση της πλατφόρμας, και τα αποτελέσματα.
* **Κεφάλαιο 6**: Επίλογος. Συμπεράσματα και προτάσεις για μελλοντικές επεκτάσεις.

Η διάρθρωση αυτή ακολουθεί τη λογική ροή της ανάπτυξης και αξιολόγησης της εφαρμογής, παρέχοντας μια ολοκληρωμένη εικόνα του έργου.

# Σχεδιασμός Πλατφόρμας

## Ανάλυση Απαιτήσεων \*\*\*

## Καταγραφή μοντέλου δεδομένων \*\*\*

### ERD

Για την αρχική καταγραφή του μοντέλου δεδομένων χρησιμοποιήθηκε το μοντέλο ERD(Entity-Relationship Diagram) **μέσω του h​​​​ttps://erdplus.com/** ώστε να γίνει πιο κατανοητή η σχέση μεταξύ των οντοτήτων που υπάρχουν στην πλατφόρμα και η αλληλεπίδραση μεταξύ τους.

Το **ERD** είναι ένα εργαλείο σχεδίασης που χρησιμοποιείται για να απεικονίσει τις σχέσεις μεταξύ των οντοτήτων ενός συστήματος βάσης δεδομένων. Στο ERD, οι οντότητες (όπως οι πίνακες δεδομένων) αναπαρίστανται με **ορθογώνια παραλληλόγραμμα** και οι σχέσεις μεταξύ τους με **γραμμές**. Κάθε σχέση μπορεί να έχει ιδιότητες, όπως ο τύπος της σχέσης (1:1, 1:N, N:M). Τα ERD είναι πολύ χρήσιμα κατά τη διάρκεια του σχεδιασμού μιας βάσης δεδομένων, καθώς βοηθούν στην κατανόηση της δομής των δεδομένων και των σχέσεων τους προτού υλοποιηθεί το φυσικό μοντέλο της βάσης δεδομένων.

Αυτές οι έννοιες και τα εργαλεία χρησιμοποιούνται συχνά για να σχεδιάσουν βάσεις δεδομένων που θα είναι **ευέλικτες**, **αξιόπιστες** και **εύκολες στην επεκτασιμότητα**. Μπορείτε να αναφέρετε στην εργασία σας πώς αυτά τα εργαλεία σας βοήθησαν να κατανοήσετε τις ανάγκες και τις απαιτήσεις του συστήματος που αναπτύσσετε, καθώς και την τελική σχεδίαση της βάσης δεδομένων.

### Σχεσιακό μοντέλο­

Στην σύνέχεια μετατράπηκε το ERD σε **Σχεσιακό Μοντέλο Δεδομένων, το οποίο όμως επεκτάθηκε κατά την διάρκεια της ηλοποίησης λαμβάνοντάς υποψήν νέα δεδομένα οπώς η ανάγκη υπάρξης οντότητας για τον ελεκτή ετείσεων και για τον ελεκτή είσοδου. Για την τελική έκδωση του σχεσιακού μοντέλου δεδομένων αξιοποιήθηκε η διεπαφή phpmyadmin μεσώ του apache2.**

Το **Σχεσιακό Μοντέλο Δεδομένων** είναι μια θεωρητική προσέγγιση για την οργάνωση και διαχείριση των δεδομένων σε βάσεις δεδομένων. Βασίζεται στην έννοια των **σχέσεων (tables)**, όπου τα δεδομένα οργανώνονται σε πίνακες και συσχετίζονται μεταξύ τους μέσω **κλειδιών** (keys). Κάθε πίνακας περιέχει γραμμές (ή εγγραφές) και στήλες (ή πεδία). Τα **σχεσιακά δεδομένα** διασφαλίζουν την **κανονικοποίηση** για να ελαχιστοποιήσουν την πλεονασματικότητα και τις ανεπιθύμητες εξαρτήσεις. Η επικοινωνία μεταξύ των πινάκων γίνεται μέσω **ξένων κλειδιών (foreign keys)** που συνδέουν έναν πίνακα με άλλον, εξασφαλίζοντας την ακεραιότητα των δεδομένων.

Παρακάτω θα αναλυθούν τα περιεχόμενα των οντοτήτων της βάσης δεδομένων:

1. **departments**: Αναφέρεται στα Προπτυχιακά Προγράμματα.  
   • **id**: Ο μοναδικός κωδικός του Προγράμματος.  
   • **name**: Το όνομα του Προγράμματος.
2. **addresses**: Αποθηκεύει διευθύνσεις ακαδημαϊκών.  
   • **id**: Ο μοναδικός κωδικός της διεύθυνσης.  
   • **academic\_id**: Σύνδεση με τον ακαδημαϊκό.  
   • **is\_permanent**: Δηλώνει αν η διεύθυνση είναι μόνιμη (1/0).  
   • **location**: Η διεύθυνσης.  
   • **phone**: Τηλέφωνο επικοινωνίας.  
   • **created\_at**: Χρόνος δημιουργίας της εγγραφής.  
   • **updated\_at**: Χρόνος τελευταίας ενημέρωσης.
3. **card\_applicants**: Αναφέρεται στους αιτούντες καρτών φαγητού.  
   • **academic\_id**: Σύνδεση με τον ακαδημαϊκό.  
   • **first\_year**: Το έτος πρώτης εγγραφής.  
   • **department\_id**: Ο κωδικός του τμήματος που ανήκει ο αιτών.  
   • **created\_at**: Χρόνος δημιουργίας της εγγραφής.  
   • **updated\_at**: Χρόνος τελευταίας ενημέρωσης.
4. **card\_applications**: Αναφέρεται στις αιτήσεις καρτών φαγητού.  
   • **id**: Ο μοναδικός κωδικός της αίτησης.  
   • **academic\_id**: Σύνδεση με τον ακαδημαϊκό που αιτείται.  
   • **created\_at**: Χρόνος δημιουργίας της αίτησης.  
   • **updated\_at**: Χρόνος τελευταίας ενημέρωσης.  
   • **expiration\_date**: Ημερομηνία λήξης της αίτησης.
5. **usage\_cards**: Καταγράφει τη χρήση καρτών φαγητού.  
   • **id**: Ο μοναδικός κωδικός χρήσης.  
   • **academic\_id**: Σύνδεση με τον ακαδημαϊκό.  
   • **date**: Η ημερομηνία χρήσης.  
   • **period**: Η περίοδος χρήσης (πρωινό, μεσημεριανό, βραδινό).  
   • **time**: Η ώρα χρήσης.  
   • **entry\_staff\_id**: Ο υπεύθυνος καταγραφής της χρήσης.
6. **coupon\_owners**: Καταγράφει τους κατόχους κουπονιών φαγητού.  
   • **academic\_id**: Σύνδεση με τον ακαδημαϊκό.  
   • **money**: Το ποσό που κατέχει ο χρήστης.  
   • **BREAKFAST**: Αριθμός κουπονιών για πρωινό.  
   • **LUNCH**: Αριθμός κουπονιών για μεσημεριανό.  
   • **DINNER**: Αριθμός κουπονιών για βραδινό.  
   • **created\_at**: Χρόνος δημιουργίας της εγγραφής.  
   • **updated\_at**: Χρόνος τελευταίας ενημέρωσης.
7. **purchase\_coupons**: Καταγράφει τις αγορές κουπονιών.  
   • **id**: Ο μοναδικός κωδικός αγοράς.  
   • **academic\_id**: Σύνδεση με τον ακαδημαϊκό.  
   • **coupon\_staff\_id**: Σύνδεση με τον υπεύθυνο κουπονιών.  
   • **money**: Το συνολικό ποσό αγοράς.  
   • **BREAKFAST**: Κουπόνια πρωινού που αγοράστηκαν.  
   • **LUNCH**: Κουπόνια μεσημεριανού που αγοράστηκαν.  
   • **DINNER**: Κουπόνια βραδινού που αγοράστηκαν.  
   • **created\_at**: Χρόνος δημιουργίας της αγοράς.
8. **transfer\_coupon**: Καταγράφει τις μεταφορές κουπονιών μεταξύ χρηστών.  
   • **id**: Ο μοναδικός κωδικός μεταφοράς.  
   • **sender\_id**: Ο κωδικός του αποστολέα των κουπονιών.  
   • **receiver\_id**: Ο κωδικός του παραλήπτη των κουπονιών.  
   • **BREAKFAST**: Ποσότητα κουπονιών πρωινού που μεταφέρθηκαν.  
   • **LUNCH**: Ποσότητα κουπονιών μεσημεριανού που μεταφέρθηκαν.  
   • **DINNER**: Ποσότητα κουπονιών βραδινού που μεταφέρθηκαν.  
   • **created\_at**: Χρόνος δημιουργίας της εγγραφής.
9. **usage\_coupons**: Καταγράφει τη χρήση κουπονιών από ακαδημαϊκούς.  
   • **id**: Ο μοναδικός κωδικός χρήσης.  
   • **academic\_id**: Σύνδεση με τον ακαδημαϊκό που χρησιμοποίησε κουπόνια.  
   • **entry\_staff\_id**: Ο υπεύθυνος καταγραφής της χρήσης.  
   • **period**: Η περίοδος χρήσης (πρωινό, μεσημεριανό, βραδινό).  
   • **created\_at**: Χρόνος καταγραφής της χρήσης.
10. **card\_application\_update**: Καταγράφει ενημερώσεις σε αιτήσεις καρτών.  
    • **id**: Ο μοναδικός κωδικός ενημέρωσης.  
    • **card\_application\_staff\_id**: Ο κωδικός του υπεύθυνου προσωπικού για την ενημέρωση. Null εαν την ενημερωση την κανει ο αιτούντας  
    • **card\_application\_id**: Η σύνδεση με την αίτηση που ενημερώθηκε.  
    • **created\_at**: Χρόνος δημιουργίας της ενημέρωσης.  
    • **updated\_at**: Χρόνος τελευταίας ενημέρωσης.  
    • **comment**: Σχόλιο σχετικά με την ενημέρωση.  
    • **status**: Η κατάσταση της αίτησης (submitted, accepted, rejected, κλπ).
11. **card\_application\_documents**: Αποθηκεύει έγγραφα που σχετίζονται με αιτήσεις καρτών.  
    • **id**: Ο μοναδικός κωδικός εγγράφου.  
    • **card\_application\_id**: Σύνδεση με την αίτηση που σχετίζεται το έγγραφο.  
    • **status**: Η κατάσταση του εγγράφου (submitted, accepted, rejected).  
    • **file\_name**: Το όνομα του αρχείου εγγράφου.  
    • **description**: Περιγραφή του εγγράφου.  
    • **created\_at**: Χρόνος δημιουργίας της εγγραφής.  
    • **updated\_at**: Χρόνος τελευταίας ενημέρωσης.  
    • **deleted\_at**: Χρόνος διαγραφής (αν υπάρχει).
12. **academics**: Αναφέρεται στα στοιχεία ακαδημαϊκών χρηστών.  
    • **academic\_id**: Ο μοναδικός κωδικός του ακαδημαϊκού.  
    • **name**: Το όνομα του χρήστη.  
    • **email**: Η διεύθυνση email του χρήστη.  
    • **password**: Ο κωδικός πρόσβασης του χρήστη.  
    • **status**: Η κατάσταση του χρήστη (undergraduate, postgraduate, phd, researcher, κλπ).  
    • **is\_active**: Καθορίζει αν ο χρήστης είναι ενεργός (1/0).  
    • **a\_m**: Το μητρώο αριθμού του ακαδημαϊκού.  
    • **created\_at**: Χρόνος δημιουργίας της εγγραφής.  
    • **updated\_at**: Χρόνος τελευταίας ενημέρωσης.
13. **entry\_staff**: Αφορά το προσωπικό που διαχειρίζεται τις εισόδους/χρήσεις κουπονιών.  
    • **id**: Ο μοναδικός κωδικός του υπαλλήλου.  
    • **name**: Το όνομα του υπαλλήλου.  
    • **email**: Η διεύθυνση email του υπαλλήλου.  
    • **password**: Ο κωδικός πρόσβασης του υπαλλήλου.  
    • **status**: Η κατάσταση του υπαλλήλου (staff entry, staff coupon, κλπ).  
    • **created\_at**: Χρόνος δημιουργίας της εγγραφής.  
    • **updated\_at**: Χρόνος τελευταίας ενημέρωσης.
14. **coupon\_staff**: Προσωπικό διαχείρισης κουπονιών.  
    • **id**: Ο μοναδικός κωδικός του υπαλλήλου.  
    • **name**: Το όνομα του υπαλλήλου.  
    • **email**: Η διεύθυνση email του υπαλλήλου.  
    • **password**: Ο κωδικός πρόσβασης του υπαλλήλου.  
    • **status**: Η κατάσταση του υπαλλήλου (staff coupon, staff card application, κλπ).  
    • **created\_at**: Χρόνος δημιουργίας της εγγραφής.  
    • **updated\_at**: Χρόνος τελευταίας ενημέρωσης.
15. **card\_application\_staff**: Καταγράφει το προσωπικό που διαχειρίζεται αιτήσεις καρτών.  
    • **id**: Ο μοναδικός κωδικός του προσωπικού.  
    • **name**: Το όνομα του υπαλλήλου.  
    • **email**: Η διεύθυνση email του υπαλλήλου.  
    • **password**: Ο κωδικός πρόσβασης του υπαλλήλου.  
    • **status**: Η κατάσταση του υπαλλήλου (staff card application, staff coupon, κλπ).  
    • **created\_at**: Χρόνος δημιουργίας της εγγραφής.  
    • **updated\_at**: Χρόνος τελευταίας ενημέρωσης.

## Χάρτης πλόηγησης και σενάρια χρήσης \*\*\*

# Αρχιτεκτονική Πλατφόρμας

Στο κεφάλαιο αυτό παρουσιάζεται η μελέτη που έγινε για την αρχιτεκτονική της πλατφόρμας. Αρχικά περιγράφονται οι ορισμοί και οι τεχνολογίες που χρησιμοποιήθηκαν, στη συνέχεια αναλύεται η αρχιτεκτονική της πλατφόρμας και γίνεται ο διαχωρισμός της στα επιμέρους υποσυστήματα και τέλος περιγράφεται η οργάνωση των αρχείων κώδικα της πλατφόρμας.

## Ορισμοί

* **Artisan:** Laravel's built-in CLI tool for performing common development tasks.
* **Backend:** The server-side of an application. It handles the logic, data storage, and communication with the frontend.
* **Blade:** Laravel's templating engine for rendering dynamic views.
* **Command:** A custom Artisan command, usually defined for automating project-specific tasks.
* **Controller:** A class that handles HTTP requests and orchestrates the retrieval of data and rendering of views or API responses.
* **Eloquent ORM:** Laravel's Object-Relational Mapper, which provides a fluent and expressive interface for interacting with the database.
* **Frontend:** The user interface (UI) of an application. It's what the user directly interacts with, such as the pages, buttons, forms, and visual elements.
* **Middleware:** A mechanism for filtering HTTP requests before they reach the controller or modifying responses before sending them to the client.
* **Migration:** A file that defines the structure of database tables and allows for version control of the database schema.
* **Model:** A class that represents a table in the database and provides methods to interact with its records.
* **Routes:** Definitions that map URLs to specific controllers or closures to handle incoming requests.
* **Seeder:** A class used to populate the database with sample or testing data.
* **Service Container:** A dependency injection system that manages class dependencies and their lifecycle.
* **Service Provider:** A class that binds services and dependencies into the Laravel application.
* **SPA (Single-Page Application):** A web application that loads a single HTML page and dynamically updates the content as the user interacts with it, without requiring full page reloads.
* **View:** A file (usually Blade template) responsible for rendering HTML or other content.

## Τεχνολογίες που χρησιμοποιήθηκαν

### Node Package Manager

Το **Node Package Manager(NPM)****[2]** είναι το μεγαλύτερο οικοσύστημα διαχείρισης πακέτων για JavaScript. Ορισμένα χαρακτηριστικά του NPM είναι:

1. **Διαχείριση εξαρτήσεων front-end**: Εγκατάσταση και συντήρηση βιβλιοθηκών όπως το Vue.js και το Vuetify.
2. **Δημιουργία & μεταγλώττιση κώδικα**: Χρήση εργαλείων όπως το Vite για τη μετατροπή του μοντέρνου JavaScript σε μορφή συμβατή με όλους τους browsers.
3. **Διαχείριση scripts**: Χρήση προσαρμοσμένων εντολών για τη μεταγλώττιση και την εκκίνηση της εφαρμογής.

Το NPM βοήθησε στη γρήγορη εγκατάσταση και ενημέρωση όλων των πακέτων που απαιτούνται για το front-end development.

### Λογισμικό Vite

Το Vite[3] είναι ένας local development server που δημιουργήθηκε από τον Evan You, τον εφευρέτη του Vue.js, ο οποίος χρησιμοποιείται από προεπιλογή σε Vue και React project templates. Υποστηρίζει JavaScript, TypeScript, vue. Εσωτερικά, χρησιμοποιεί το Rollup και το esbuild για την ομαδοποίηση.

Παρακολουθεί τα αρχεία καθώς ενημερώνονται, και όταν το αρχείο αποθηκεύεται, το πρόγραμμα περιήγησης στον ιστό επαναφορτώνει τον κώδικα που τροποποιείται χρησιμοποιώντας μια διαδικασία γνωστή ως Hot Module Replacement (HMR), η οποία λειτουργεί επαναφορτώνοντας μόνο το συγκεκριμένο αρχείο που τροποποιείται χρησιμοποιώντας ES6 modules (ESM) αντί να μεταγλωττίσει εκ νέου ολόκληρο το πρόγραμμα.

Το Vite διαθέτει ενσωματωμένη υποστήριξη για απόδοση από την πλευρά του διακομιστή (SSR). Από προεπιλογή, ακούει στη θύρα TCP 5173. Το Vite μπορεί να ρυθμιστεί ώστε να παραδίδει περιεχόμενο μέσω HTTPS και αιτήσεων μεσολάβησης (συμπεριλαμβανομένου του WebSocket) σε έναν back-end διακομιστή ιστού (όπως ο Apache HTTP server ή ο lighttpd). Το λογισμικό αυτό χρησιμοποιήθηκε κατά τη διάρκεια του development καθώς προσφέρει πολλά πλεονεκτήματα όπως αναφέρθηκε προηγουμένως, αποδίδοντας μια ευχάριστη εμπειρία δίχως να χάνεται χρόνος σε τετριμμένες διαδικασίες που μπορούν να αυτοματοποιηθούν με τη χρήση του λογισμικού αυτού, όπως το HMR.

### Javascript

Η Javascript[4] είναι μια υψηλού επιπέδου, αντικειμενοστραφής, δυναμική γλώσσα προγραμματισμού που χρησιμοποιείται σε διάφορους τομείς, όπως για την ανάπτυξη διαδραστικών ιστοσελίδων και εφαρμογών στον πελάτη (client-side), για ανάπτυξη mobile εφαρμογών, για ανάπτυξη server-side εφαρμογών ή για ανάπτυξη εφαρμογών στον χώρο του Internet of Things (IoT). Είναι μία από τις τρεις βασικές τεχνολογίες του Ιστού (Web), μαζί με το HTML και το CSS.

Οι βασικές ιδέες και ιδιότητες της javascript περιλαμβάνουν:

1. Υψηλή προγραμματιστική ευελιξία: Η javascript επιτρέπει τη δυναμική δημιουργία και αλλαγή των αντικειμένων και των λειτουργιών τους, καθιστώντας την ιδανική για προγραμματιστικές προσεγγίσεις όπως ο αντικειμενοστραφής προγραμματισμός (Object-Oriented Programming) ή ο λειτουργικός προγραμματισμός (Functional Programming).
2. Αλληλεπίδραση με το DOM (Document Object Model): Η javascript επιτρέπει την αλληλεπίδραση με το DOM του προγράμματος περιήγησης (browser), προσφέροντας δυνατότητες όπως η δυναμική τροποποίηση του περιεχομένου και της εμφάνισης των ιστοσελίδων, ο χειρισμός των συμβάντων (events) και η αλλαγή των στοιχείων της ιστοσελίδας κατά τη διάρκεια της εκτέλεσης της εφαρμογής.

Για τα πλαίσια της εργασίας, η javascript χρησιμοποιήθηκε για τη δημιουργία της διεπιφάνειας του χρήστη (frontend) .

### Vue.js --- -+

Το Vue.js [5] είναι ένα ευρέως διαδεδομένο framework για τη δημιουργία διαδικτυακών εφαρμογών.

Έχει σχεδιαστεί για να είναι ελαφρύ και εύκολο στη χρήση, και προσφέρει πολλά επιπλέον χαρακτηριστικά που διευκολύνουν την ανάπτυξη εφαρμογών.

Το Vue.js χρησιμοποιεί μια σύγχρονη αρχιτεκτονική με βάση τα components, τα οποία είναι επαναχρησιμοποιούμενα και επεκτάσιμα στοιχεία, δίνοντας την δυνατότητα στους προγραμματιστές να δημιουργήσουν σύνθετες εφαρμογές. Επιπλέον, το Vue.js προσφέρει ευέλικτα εργαλεία για τη διαχείριση της κατάστασης των components και τη δρομολόγηση, ενώ υποστηρίζει επίσης την ενσωμάτωση του framework σε άλλα project μέσω του Vue CLI. Τέλος, το Vue.js προσφέρει ένα ισχυρό σύστημα αυτόματης ανανέωσης του UI, προσαρμόζοντας αυτόματα τα στοιχεία της διεπαφής όταν τα δεδομένα τους αλλάζουν, χωρίς την ανάγκη για μηχανισμούς χειροκίνητης ενημέρωσης.

### Vuex

Το Vuex[6] είναι μια βιβλιοθήκη διαχείρισης κατάστασης για εφαρμογές Vue.js. Προσφέρει έναν συγκεντρωτικό αποθηκευτικό χώρο (store) για όλα τα δεδομένα της εφαρμογής, διευκολύνοντας την κοινή χρήση και διαχείρισή τους ανάμεσα σε διαφορετικά στοιχεία.

Στην εφαρμογή, το Vuex χρησιμοποιήθηκε για:

* **Διαχείριση Χρηστών**: Αποθήκευση πληροφοριών σχετικά με τους συνδεδεμένους χρήστες, όπως τα δεδομένα σύνδεσης και τα δικαιώματά τους.
* **Διαχείριση Αιτήσεων και Κουπονιών**: Ενιαίος χώρος αποθήκευσης για την παρακολούθηση της κατάστασης αιτήσεων και την ενημέρωση των κουπονιών.
* **Επικοινωνία με το API**: Ενοποίηση δεδομένων που προέρχονται από το backend, παρέχοντας συνεπή πρόσβαση στην κατάσταση της εφαρμογής.

### Vuetify --- -+

Το πακέτο Vuetify[7] είναι ένα δημοφιλές πακέτο από Vue components που στοχεύει στην επιτάχυνση και απλοποίηση της ανάπτυξης UI. Προσφέρει μια συλλογή από προσαρμόσιμα και συνθέσιμα components που οι προγραμματιστές μπορούν να ενσωματώσουν γρήγορα στις Vue εφαρμογές τους. Ορισμένα βασικά χαρακτηριστικά του Vuetify περιλαμβάνουν τα εξής:

1. Ευέλικτο σύστημα πλέγματος (Flexible Grid System): Το Vuetify παρέχει ένα ισχυρό σύστημα πλέγματος με πληθώρα προεπιλεγμένων κλάσεων για τοποθέτηση στοιχείων στο UI με ευελιξία και ακρίβεια.
2. Έτοιμα θεματικά στοιχεία (Material Design Components): Το Vuetify περιλαμβάνει ένα εκτεταμένο σύνολο έτοιμων θεματικών στοιχείων βασισμένων στο Material Design της Google, όπως κουμπιά, φόρμες, πίνακες, κάρτες, μενού, διαλογιστικά παράθυρα και πολλά άλλα, που μπορούν να χρησιμοποιηθούν άμεσα στις εφαρμογές.
3. Ευκολία προσαρμογής (Customization): Το Vuetify προσφέρει πλούσιες δυνατότητες προσαρμογής, όπως προσαρμογή του θέματος, προσαρμογή των χρωμάτων, προσθήκη των δικών σας στυλ και προσαρμογή του UI σύμφωνα με τις ανάγκες και τις απαιτήσεις της εφαρμογής.

Η βιβλιοθήκη αυτή χρησιμοποιήθηκε για τον εμπλουτισμό του UI με όμορφα στοιχεία και με στόχο την αυτόματη προσαρμογή ανεξαρτήτως συσκευής και μεγέθους οθόνης. Είναι μια έμπιστη βιβλιοθήκη καθώς χρησιμοποιείται από μεγάλους οργανισμούς για την κατασκευή των δικών τους γραφικών διεπαφών.

### Vue-i18n​

Το πακέτο Vue-i18n[8] αποτελεί ένα πακέτο που επιτρέπει τη μετάφραση εφαρμογών αξιοποιώντας i18n[9] εφαρμογών Vue.js, προσφέροντας τη δυνατότητα υποστήριξης πολλαπλών γλωσσών. Παρέχει εργαλεία για τη δημιουργία μεταφράσεων, τη μορφοποίηση ημερομηνιών, αριθμών και άλλων δεδομένων. Χρησιμοποιείται συχνά σε εφαρμογές που απαιτούν δυναμική αλλαγή γλώσσας χωρίς την ανάγκη ανανέωσης της σελίδας. Με την ενσωμάτωσή του, οι εφαρμογές μπορούν να καλύψουν τις ανάγκες ενός διεθνούς κοινού, ενισχύοντας την προσβασιμότητα και την εμπειρία χρήστη.

### vue-router

Το πακέτο vue-router[10] αποτελεί τον επίσημο διαχειριστή δρομολόγησης (router) για το Vue.js. Επιτρέπει τη διαχείριση της πλοήγησης μεταξύ διαφορετικών σελίδων ή προβολών σε μια εφαρμογή. Μέσω της χρήσης του, ορίζονται διαδρομές που συνδέονται με συγκεκριμένα URLs, ενώ παρέχονται δυνατότητες όπως παραμετροποιημένα routes, ανακατευθύνσεις και middleware για προστασία. Είναι βασικό εργαλείο για την ανάπτυξη εφαρμογών μιας σελίδας (SPA), καθώς διευκολύνει την οργάνωση και την πλοήγηση ακόμη και σε σύνθετα περιβάλλοντα.

### laravel-echo

Το πακέτο laravel-echo[11] είναι ένα εργαλείο που ενσωματώνει λειτουργίες σε πραγματικό χρόνο (real-time) σε εφαρμογές Laravel. Διαχειρίζεται γεγονότα που μεταδίδονται μέσω WebSockets ή άλλων τεχνολογιών, επιτρέποντας την ανάπτυξη δυναμικών λειτουργιών όπως ζωντανές ειδοποιήσεις, συνομιλίες και ενημερώσεις δεδομένων σε πραγματικό χρόνο. Αξιοποιεί κανάλια (channels) για τη μεταφορά δεδομένων σε πολλούς χρήστες ταυτόχρονα, ενισχύοντας τη διαδραστικότητα και τη συνολική εμπειρία χρήστη.

### axios

Το πακέτο axios[12] είναι μια βιβλιοθήκη που διευκολύνει την επικοινωνία μεταξύ του frontend και του backend μέσω αιτημάτων HTTP. Χρησιμοποιείται για την αποστολή και λήψη δεδομένων, υποστηρίζοντας λειτουργίες όπως η δημιουργία, η ανάγνωση, η ενημέρωση και η διαγραφή πληροφοριών. Παρέχει δυνατότητες για τη ρύθμιση headers, τη χρήση tokens ασφαλείας και την αποτελεσματική διαχείριση σφαλμάτων. Το axios αποτελεί βασικό στοιχείο για την κατασκευή API-driven εφαρμογών, προσφέροντας ταχύτητα και ευελιξία στην επικοινωνία μεταξύ διαφορετικών συστημάτων.

### CSS

Η CSS (Cascading Style Sheets)[13] είναι μία γλώσσα προγραμματισμού που χρησιμοποιείται για τον σχεδιασμό και τον ενσωματωμένο στυλισμό των ιστοσελίδων. Εδώ είναι μερικές βασικές πληροφορίες για την CSS:

1. Συντακτική Δομή: Η CSS αποτελείται από κανόνες (rules) που περιέχουν έναν επιλογέα (selector) και έναν κανόνα στυλ (style rule). Ο επιλογέας καθορίζει ποια στοιχεία της ιστοσελίδας θα επηρεαστούν από τον κανόνα στυλ, ενώ ο κανόνας στυλ καθορίζει την εμφάνιση των στοιχείων αυτών.
2. Ιεραρχία (Hierarchy): Η CSS ακολουθεί ένα σύστημα ιεραρχίας, όπου οι κανόνες στυλ μπορούν να κληρονομούνται από πάνω προς τα κάτω. Αυτό σημαίνει ότι οι κανόνες στυλ που ορίζονται σε υψηλότερο επίπεδο μπορούν να επηρεάσουν στοιχεία που βρίσκονται μέσα σε αυτά.
3. Επιλογείς (Selectors): Οι επιλογείς είναι τα μέρη του CSS κανόνα που καθορίζουν ποια στοιχεία της ιστοσελίδας θα επηρεαστούν από τον κανόνα στυλ. Μπορούν να είναι στοιχεία HTML (π.χ. <p>, <h1>, <div>), κλάσματα των στοιχείων HTML (π.χ. class ή id) ή άλλοι τύποι επιλογέων.
4. Ιδιότητες (Properties): Οι ιδιότητες καθορίζουν τον τρόπο με τον οποίο θα εμφανιστούν τα στοιχεία στην ιστοσελίδα. Μπορούν να περιλαμβάνουν το χρώμα, το μέγεθος, το περιθώριο, το περίγραμμα, την γραμματοσειρά, την διάταξη και πολλά άλλα.
5. Τιμές (Values): Οι τιμές καθορίζουν την τιμή μιας ιδιότητας. Για παράδειγμα, μπορεί να είναι ένα χρώμα (π.χ. red, #ff0000), ένα μέγεθος (π.χ. 16px), ένα περίγραμμα (π.χ. solid, dashed), κ.λπ.

### GitHub \*\*\*

Το GitHub[14] είναι μια πλατφόρμα που βασίζεται στο Git και επιτρέπει τη διαχείριση πηγαίου κώδικα και τη συνεργασία σε έργα. Εκτός από τη φιλοξενία αποθετηρίων, προσφέρει εργαλεία όπως Pull Requests, Actions για αυτοματοποίηση, και ενσωμάτωση CI/CD.

Στην εφαρμογή, το GitHub χρησιμοποιήθηκε για:

* **Φιλοξενία Κώδικα**: Αποθήκευση του πηγαίου κώδικα της εφαρμογής.
* **Ιστορικό Αλλαγών**: Παρακολούθηση αλλαγών στον κώδικα μέσω commits.

### Apache2

Ο Apache HTTP Server (Apache2)[15] είναι ένας από τους πιο ευρέως χρησιμοποιούμενους web servers στον κόσμο. Αποτελεί θεμέλιο λίθο για τη φιλοξενία και εξυπηρέτηση ιστοσελίδων και web εφαρμογών.Ο Apache2 μπορεί να χρησιμοποιηθεί:

1. Την ασφαλή και αποδοτική διαχείριση των αιτημάτων HTTP/HTTPS.
2. Την εξυπηρέτηση των στατικών αρχείων (CSS, JavaScript) και των δυναμικών σελίδων μέσω του Laravel framework.
3. Τη διαχείριση των virtual hosts για την ανάπτυξη και δοκιμή της εφαρμογής.

### MySQL --- -+

Η MySQL [16] είναι ένα ανοιχτού κώδικα σύστημα διαχείρισης βάσεων δεδομένων (ΣΔΒΔ) που χρησιμοποιεί τη γλώσσα SQL (Structured Query Language) για τη δημιουργία, τη διαχείριση και τον επαναπροσδιορισμό των σχέσεων μεταξύ των δεδομένων σε μια βάση δεδομένων. Ορίζεται ως ένα από τα πιο δημοφιλή ΣΔΒΔ και χρησιμοποιείται ευρέως σε πολλές εφαρμογές, από μικρούς ιστότοπους μέχρι μεγάλα επιχειρησιακά συστήματα.

Ορισμένα χαρακτηριστικά της MySQL είναι τα εξής:

Αποδοτική αποθήκευση και ανάκτηση δεδομένων: Το MySQL προσφέρει γρήγορη αποθήκευση και ανάκτηση των δεδομένων, καθιστώντας το κατάλληλο για εφαρμογές μεγάλης κλίμακας και υψηλής προσβασιμότητας.

Υψηλή ασφάλεια: Η MySQL παρέχει δυνατότητες ασφαλείας όπως κρυπτογράφηση των δεδομένων, αυθεντικοποίηση και εξουσιοδότηση χρηστών, δυνατότητα απομόνωσης των δεδομένων μεταξύ χρηστών και πρόσβαση με βάση ρόλους.

Ευελιξία: Η MySQL υποστηρίζει πολλές πλατφόρμες λειτουργικού συστήματος, συμπεριλαμβανομένων των Windows, macOS και Linux, και είναι συμβατό με πολλές γλώσσες προγραμματισμού, όπως C++, Java, PHP, Python, κ.ά.

Δυνατότητες αναγνώρισης και αποκατάστασης σφαλμάτων: Η MySQL παρέχει πολλά εργαλεία για τον έλεγχο, τον εντοπισμό και την αποκατάσταση σφαλμάτων στη βάση δεδομένων.

Σταθερότητα και αξιοπιστία: Το MySQL έχει αποδείξει την αξιοπιστία του σε πολλές εφαρμογές, με σταθερή λειτουργία και ανθεκτικότητα σε ανώτερα φορτία και αντοχή σε βλάβες.

Εκτεταμένη κοινότητα χρηστών και υποστήριξη: Το MySQL έχει μια μεγάλη και ενεργή κοινότητα χρηστών και παρέχει εκτεταμένη τεκμηρίωση, παραδείγματα και υποστήριξη, καθιστώντας το ευκολότερο για τους προγραμματιστές να το χρησιμοποιήσουν και να επιλύσουν προβλήματα.

### PHP \*\*\*

Η PHP(Hypertext Preprocessor)[17] είναι μια γλώσσα προγραμματισμού που χρησιμοποιείται ευρέως για την ανάπτυξη δυναμικών ιστοσελίδων και backend εφαρμογών.

Στην εφαρμογή, η PHP χρησιμοποιήθηκε για:

* **Backend Υλοποίηση**: Ανάπτυξη του API που εξυπηρετεί την frontend εφαρμογή.
* **Διαχείριση Αιτήσεων**: Επεξεργασία και αποθήκευση δεδομένων σχετικών με τις αιτήσεις και τα κουπόνια.
* **Ενσωμάτωση Βιβλιοθηκών**: Χρήση πρόσθετων πακέτων μέσω Composer για τη διευκόλυνση της ανάπτυξης.

### PhpMyAdmin ---- -+

Το phpMyAdmin [18] είναι μια δωρεάν και ανοικτού κώδικα διαχειριστική διεπαφή για τη διαχείριση των βάσεων δεδομένων MySQL μέσω ιστού. Εδώ είναι μερικές πληροφορίες για το phpMyAdmin:

1. Χαρακτηριστικά διαχείρισης βάσεων δεδομένων: Το phpMyAdmin παρέχει πολλά εργαλεία για τη διαχείριση των βάσεων δεδομένων MySQL, όπως δημιουργία, διαγραφή, τροποποίηση πινάκων, εισαγωγή, εξαγωγή και επεξεργασία δεδομένων.
2. Οπτικοποίηση και ανάλυση δεδομένων: Το phpMyAdmin παρέχει δυνατότητες οπτικοποίησης και ανάλυσης των δεδομένων στις βάσεις δεδομένων, όπως γραφήματα, διαγράμματα και αναφορές.
3. Ασφάλεια: Το phpMyAdmin προσφέρει διάφορα μέτρα ασφαλείας, όπως αυθεντικοποίηση χρηστών, περιορισμό πρόσβασης σε βάσεις δεδομένων μόνο για εξουσιοδοτημένους χρήστες, κρυπτογράφηση και δυνατότητα προσαρμογής των ρυθμίσεων ασφαλείας.
4. Πολυγλωσσική υποστήριξη: Το phpMyAdmin υποστηρίζει πολλές γλώσσες, προσφέροντας δυνατότητα επιλογής γλώσσας κατά την είσοδο στο περιβάλλον.
5. Δυνατότητα εξατομίκευσης: Το phpMyAdmin επιτρέπει την εξατομίκευση του περιβάλλοντος εργασίας, όπως την προσαρμογή του θέματος, την προσθήκη προσαρμοσμένων λειτουργιών ή αλλαγή των ρυθμίσεων προβολής.
6. Εύκολη χρήση: Το phpMyAdmin προσφέρει μια φιλική διεπαφή χρήστη με ένα απλό περιβάλλον εργασίας, καθιστώντας το ευκολότερο για τους χρήστες να διαχειρίζονται τις βάσεις δεδομένων τους.
7. Κοινότητα και υποστήριξη: Το phpMyAdmin έχει μια ισχυρή κοινότητα χρηστών και προγραμματιστών που παρέχουν υποστήριξη, συμβουλές και αναβαθμίσεις. Υπάρχει επίσης εκτενής τεκμηρίωση και πόροι για την εκμάθηση και την ανάπτυξη με το phpMyAdmin.

Συνολικά, το phpMyAdmin είναι ένα ισχυρό εργαλείο διαχείρισης βάσεων δεδομένων MySQL που παρέχει πολλές δυνατότητες και ευκολίες.

### Composer

Ο **Composer****[19]** είναι το βασικό εργαλείο διαχείρισης πακέτων για PHP. Ορισμένα χαρακτηριστικά του Composer είναι:

1. **Διαχείριση εξαρτήσεων**: Εύκολη εγκατάσταση και ενημέρωση βιβλιοθηκών που χρησιμοποιούνται στο Laravel.
2. **Αυτόματη φόρτωση κώδικα**: Η λειτουργία autoloading του Composer επιτάχυνε την ανάπτυξη κώδικα χωρίς να χρειάζεται μη αυτόματη εισαγωγή αρχείων.
3. **Ενημέρωση των βιβλιοθηκών**: Διευκόλυνε τη διατήρηση της εφαρμογής σύγχρονης και ασφαλούς.

### Laravel

Το Laravel[20] είναι ένα σύγχρονο PHP framework που σχεδιάστηκε για να διευκολύνει την ανάπτυξη web εφαρμογών, παρέχοντας ένα ευέλικτο και ισχυρό οικοσύστημα εργαλείων. Το Laravel χρησιμοποιήθηκε για:

1. **Διαχείριση των routes**: Αντιστοίχιση URLs σε λειτουργίες του backend.
2. **MVC αρχιτεκτονική**: Διάσπαση της λογικής σε διάφορες ομάδες για καθαρότερο κώδικα. Οπως για πάραδειγμα είναι οι Controllers ,Models, Policies, Requests,
3. **Βάσεις δεδομένων**: Χρήση των Eloquent ORM και migrations για τη διαχείριση των δεδομένων και τη δημιουργία πινάκωνμε αφιριμένο τρόπο ώστε να είναι ευκολή η αλλαγή σε απο τήν μια βάση δεδομένων στην αλλή.
4. **Αυθεντικοποίηση και Εξουσιοδότηση (authorize)** : Υλοποίηση συστημάτων ασφαλούς ταυτοποίησης χρηστών.
5. **Εργαλεία ανάπτυξης**: Χρήση ενσωματωμένων libraries, όπως το Blade για δυναμικές σελίδες και το Artisan CLI για αυτοματοποιημένες ενέργειες.
6. Laravel includes Eloquent, an object-relational mapper (ORM) that makes it enjoyable to interact with your database. When using Eloquent, each database table has a corresponding "Model" that is used to interact with that table. In addition to retrieving records from the database table, Eloquent models allow you to insert, update, and delete records from the table as well.

### Laravel/Telescope \*\*\*

Το Laravel Telescope[21] είναι ένα εργαλείο παρακολούθησης και εντοπισμού σφαλμάτων (debugging) που παρέχει πληροφορίες σχετικά με αιτήματα, εξαιρέσεις, και λειτουργίες του backend.

Στην εφαρμογή, το Telescope χρησιμοποιήθηκε για:

1. **Παρακολούθηση Αιτημάτων**: Καταγραφή και επισκόπηση εισερχόμενων αιτημάτων στο API.
2. **Εντοπισμό Σφαλμάτων**: Καταγραφή εξαιρέσεων και σφαλμάτων που προκύπτουν κατά την εκτέλεση του κώδικα.
3. **Ανάλυση Απόδοσης**: Παρακολούθηση queries και αναφορών απόδοσης της εφαρμογής.
4. Έλενχος των email που αποστέλντονται.

### fakerphp/faker \*\*\*

Το FakerPHP[22] είναι μια βιβλιοθήκη που επιτρέπει τη δημιουργία ψευδών δεδομένων (mock data) για δοκιμές.Το FakerPHP είναι μια βιβλιοθήκη που επιτρέπει τη δημιουργία ψευδών δεδομένων (mock data) για δοκιμές.

Στην εφαρμογή, χρησιμοποιήθηκε για:

1. **Δημιουργία Δοκιμαστικών Χρηστών**: Εισαγωγή ψευδών δεδομένων χρηστών για δοκιμές.
2. **Γενίκευση Δεδομένων Αιτήσεων**: Εισαγωγή δεδομένων αιτήσεων για τον έλεγχο λειτουργιών.

### barryvdh/laravel-dompdf \*\*\*

Το Dompdf[23] είναι μια βιβλιοθήκη που παρέχει δυνατότητα δημιουργίας PDF από HTML.

Στην εφαρμογή, χρησιμοποιήθηκε για:

1. **Έκδοση Αναφορών**: Δημιουργία PDF για τις αιτήσεις σίτισης και τις λίστες κουπονιών.
2. **Εξαγωγή Δεδομένων**: Παροχή εγγράφων που μπορούν να αποθηκευτούν ή να εκτυπωθούν.

### tightenco/ziggy \*\*\*

Το Ziggy[24] επιτρέπει την πρόσβαση σε Laravel routes από την εφαρμογή Vue.js.

Στην εφαρμογή, χρησιμοποιήθηκε για την α**πλοποίηση Routing**:

Ανάκτηση και χρήση Laravel routes στο frontend χωρίς σκληροκωδικοποιημένες διευθύνσεις URL.

### propaganistas/laravel-phone \*\*\*

Το Laravel Phone[25] είναι ένα πακέτο για την επικύρωση αριθμών τηλεφώνου.

Στην εφαρμογή, χρησιμοποιήθηκε για:

1. **Επικύρωση Δεδομένων Χρηστών**: Έλεγχος της εγκυρότητας των αριθμών τηλεφώνου κατά την εγγραφή.
2. Μετατρόπή δεδομένων γιά αποθήκευση στην βάση δεδομένων αλλά και την ανάκτηση τους

### laravel/sanctum \*\*\*

Το Laravel Sanctum[26] προσφέρει απλοποιημένη διαχείριση API tokens για ασφαλή επικοινωνία.

Στην εφαρμογή, χρησιμοποιήθηκε για:

1. **Ασφάλεια API**: Προστασία endpoints της εφαρμογής.
2. **Έλεγχος Πρόσβασης**: Διαχείριση δικαιωμάτων χρηστών μέσω token-based αυθεντικοποίηση.

### laravel/ui \*\*\*

Το Laravel UI[27] παρέχει προκατασκευασμένο frontend scaffolding για εφαρμογές Laravel.

Στην εφαρμογή, αρκικά αξιοπηθηκε χάρη στα π**ροκαθορισμένα Components**: Δημιουργία views για διαχύριση συνδέσεων. Και στην συνέχεια λογω της μετάβασης σε spa αξιοπηθηκε μόνο για την διαχύρηη είσοδου εξόδου των χρήστών

## Οργάνωση αρχείων framework Laravel

Για την υλοποίηση του εξυπηρετητή απαραίτητη ήταν η οργάνωση της λογικής του σε φακέλους και αρχεία ώστε ο κώδικας να είναι πιο κατανοητός και εύκολα επεκτάσιμος. Η οργάνωση αρχείων και φακέλων ήταν πλήρως καθοδιγούμενη απο την οργάνωση του framework Laravel[20] .

### Ο Κεντρικός Φάκελος (Root Directory)

#### Ο φάκελος App

Φάκελος αυτός περιέχει τον βασικό κώδικα της εφαρμογής σχεδόν όλες οι κλάσεις της εφαρμογής βρίσκονται σε αυτόν τον κατάλογο. Θα αναληθεί τι περιέχει σε επόμενη ενότητα.

#### The Bootstrap Directory

Ο φάκελος bootstrap περιέχει το αρχείο app.php, το οποίο εκκινεί (bootstrap) το framework. Επίσης, φιλοξενεί έναν κατάλογο cache που περιέχει αρχεία που δημιουργούνται από το framework για τη βελτιστοποίηση της απόδοσης, όπως τα αρχεία cache διαδρομών (routes) και υπηρεσιών (services).

#### The Config Directory

Αυτός ο φάκελος περιέχει όλα τα αρχεία για τις ρυθμίσεις της εφαρμογής.

#### The Database Directory

Αυτός ο φάκελος περιέχει όλα τα αρχεία που είναι σχετικά με την βάση δεδομένων. Απαρτίζεται από τις παρακάτω φακέλους :

* Factories Ο οποίος χρησιμοποιείται για να δημιουργεί ψεύτικα δεδομένα για χρήση στις δοκιμαστικές λειτουργίες της εφαρμογής. Συνήθως για κάθε μοντέλο υπάρχει μία κλάση ως γεννήτρια δεδομένων στην οποία έχουν οριστεί οι κατάλληλες παραμέτρους για την κάθε στήλη του πίνακα.
* Migrations. Ένας εξαιρετικά σημαντικός φάκελος ο οποίος περιέχει τα αρχεία διμιουργείας ή επεξεργάσιας ή διαγραφής πινάκων. Τα αρχεία αυτά ειναι php και η Laravel μεταφράζει τις εντολές στην επιλεγμένη βάση δεδομένων Χαρακτηριστικό της Laravel είναι ότι η ίδια διατύπωση αξιοποιείται για να μεταφερθεί σε πολλών ειδών διαφορετικές βάσεις δεδομένων
* Schema. Αυτός ο φάκελος δημιουργείται εκ των υστέρων όταν εκτελεστεί η εντολή artisan database:dump. Αυτό που γίνεται εκτέλοντας την εντόλή είναι να δημιουργεί για την κύρια βάση δεδομένων εκτός εάν οριστεί διαφορετικά ένα αρχείο scema για ταχύτερη δημιουργία της βάσης δεδομένων. Ο παραγόμενος κώδικας σε αυτή την περίπτωση δηλαδή είναι άμεσα συνδεδεμένος με το είδος της βάσης δεδομένων που έχει επιλεγεί. Κιαι οταν διμιουργείτε η βάση θα εκτελήτε αυτός ο κώδικας αντί για το μεγάλο-συνήθως- πλήθος migrations οπου είχαν διμιουργηθεί. Ταυτόχρονα γνωρίζει μεσω του πινακα migration ποια migrations εχουν εκτελεστεί στην θέση αυτου του schema ωστε να τα προσπεράσει και να εκτελέσει μόνο τιχων νεα.
* seeders. Οι αρχικοποιητές είναι αρχεία οπου ορίζοντε η συσχετίσεις των δεδομενων στην βάση. Αυτό είναι εξερετικά χρήσιμο οταν χρειάζετε να διμιοργηθουν ψετικά δεδομενα για τον έλενχο της εφαρμόγη. Καλόντας τα factorys και οριζοντας συσχετισεις μεταξύ τους μπορουν να διμιουργηθουν χιλίαδες εγραφές με καταλήλα ορισμενα τύχεα δεδομενα.

#### Φάκελος docs

Αυτός ο φάκελος δεν αποτελεί χαρακτηριστικό του framework.. επιλέχθηκε όμως για να δημιουργηθεί προκειμένου να υπάρχει μία καλύτερη οργάνωση , και να διασπαστεί σε πολλά αρχεία το αρχικό readme. Σκοπός αυτόν στον αρχείων είναι να εξηγούνε σε όποιον επιθυμεί να χρησιμοποιήσει την εφαρμογή σαν εγκαταστάτης τι υπάρχει και τι πρέπει να κάνει προκειμένου εφαρμογή να τεθεί σε λειτουργία καθώς και τι δεν έχει ακόμα υλοποιηθεί.

#### Φάκελος lang

Εδώ μέσα είναι αποθηκευμένα μορφή αρχείων php είτε σε json η μεταφράσεις όσων πληροφοριών, θέλουμε να μεταφράζονται στο χρήστη. Για την ανάπτυξη της εφαρμογής υπάρχει ελληνική και αγγλική έκδοση.

#### The node\_modules Directory

The node\_modules directory contains the npm dependencies.

#### The Public Directory

Ο φάκελος αυτός περιέχει το αρχείο index.php το οποίο είναι το σημείο εισόδου για όλα τα αιτήματα που έρχονται στο backend και κάνει την αυτόματη φόρτωση autoloading τον κατάλληλο με αρχαίων για να εξυπηρετηθεί το αίτημα. Σε αυτό το φάκελο αποθηκεύονται επίσης τα παραγόμενα αρχεία javascript css μετά την επεξεργασία τους από την vite.

#### The Resources Directory

Εδώ είναι ο κύριος φάκελος διαχείρισης του fronted, εδώ πέρα αποθηκεύονται αρχεία όπως έχουνε γραφτεί χωρίς να έχουνε περάσει από κάποιο compiler. Συνήθως αποθηκεύονται αρχεία javascript typescript css sass καθώς και τα views. Για την ανάπτυξη της εφαρμογής δεν είχαμε καθόλου αρχεία css και sass. Για την δομή των αρχείων javascript θα μιλήσουμε αργότερα. Στα views στην τελική έκδοση υπάρχει μονάχα ένας φάκελος mails όπου περιέχονται μέσα αρχεία για το πώς θα φαίνεται η πληροφορία που θέλουμε να σταλεί στο ηλεκτρονικό ταχυδρομείο των χρηστών.

#### The Routes Directory

Ο φάκελος Routers περιέχει όλους τους ορισμούς route για την εφαρμογή. Εξαρχής υπάρχουν κάποια αρχεία στα οποία είναι δημιουργηθεί για την οργάνωση αυτών.

* Web.php: Eδώ πέρα περιέχονται οι ορισμοί των σελίδων που είναι πιθανό ο χρήστης να το πληκτρολογήσει απευθείας στον φυλομετρητή του. Σε αυτά εφαρμόζεται ο RouteServiceProvider το web middleware. Το οποίο κρυπτογραφη τα cookies. Παρέχει crf protection σε ενδεχόμενα αιτήματα προκειμένου να επαληθεύεται ότι έχει γίνει από την εφαρμογή μας το αίτημα. Για την εφαρμογή υπάρχει μόνο η προβολή των εγράφων και η μεταφορά στο fronted.
* Api.php Εδώ ορίζονται κυρίως οι αλληλεπιδράσεις για την διαχείριση των αιτημάτων που έρχοντε από τον φυλομετριτή του χρήστη.
* Channels.php Εδώ στο οποίο εδώ πέρα ορίζονται όλα τα event broadcast channels που χρειάζεται η εφαρμογή.
* Console.php Το αρχείο χρησιμοποιείται για τον ορισμό όλων των κονσολών (console commands) που βασίζονται σε closures. Κάθε closure συνδέεται με με ένα στοιγμιότοιπο εντολής (command instance), παρέχοντας έναν απλό τρόπο αλληλεπίδρασης με τις μεθόδους εισόδου/εξόδου (IO methods) κάθε εντολής. Παρόλο που αυτό το αρχείο δεν ορίζει HTTP routes, λειτουργεί ως σημείο εισόδου (route) για εντολές που βασίζονται στην κονσόλα και επιτρέπουν την επικοινωνία με την εφαρμογή σας μέσω του τερματικού. Δεν αξιοποιήθηκε στο πλέσιο της εφαρμογής

#### The Storage Directory

The storage directory contains the logs, compiled Blade templates, file based sessions, file caches, and other files generated by the framework. This directory is segregated into app, framework, and logs directories. The app directory will be used to store any files generated by your application. The framework directory is used to store framework generated files and caches. Finally, the logs directory contains your application's log files.

The storage/app/public directory can be used to store user-generated files, that should be publicly accessible. That will happen with a symbolic link at public/storage which points to this directory.

#### The Tests Directory

The tests directory contains automated tests. Example PHPUnit unit tests and feature tests are provided out of the box. Each test class should be suffixed with the word Test. We can run tests using the phpunit or php vendor/bin/phpunit commands. Also if you would like a more detailed representation of your test results, we can run your tests using the php artisan test Artisan command. Here can defined test to check the response of a controller or from one route. The result of a validation. And generaly we can test everything that we implement.

#### The Vendor Directory

The vendor directory contains the Composer dependencies.

#### Μεμονομένα αρχεία

Το αρχείο .env είναι κρίσιμο για τη ρύθμιση του περιβάλλοντος της εφαρμογής. Περιλαμβάνει παραμέτρους που προσαρμόζονται ανάλογα με το περιβάλλον λειτουργίας (π.χ., development, testing, production). Οι σημαντικότερες ρυθμίσεις περιλαμβάνουν:

* Στοιχεία Βάσης Δεδομένων: Πληροφορίες όπως το όνομα της βάσης δεδομένων, ο χρήστης, και ο κωδικός πρόσβασης.
* Ρυθμίσεις SMTP: Διαμόρφωση για την αποστολή emails μέσω εξωτερικών παρόχων.
* Λειτουργία Εφαρμογής: Καθορίζει αν η εφαρμογή λειτουργεί σε κατάσταση ανάπτυξης (development) ή παραγωγής (production).

Η οργάνωση αυτή διασφαλίζει ότι η εφαρμογή είναι εύκολη στη συντήρηση, ενώ ταυτόχρονα υποστηρίζει την ασφαλή και αποδοτική λειτουργία της. Το αρχείο .env έχει 3 παραλαγές .env .env.example .env.testing . Τα .env , .env.testing περίεχει ευεστιτές πληροφόριες και για αυτο δεν μείραζετε. Αντί για αυτο δίνετε το αρχείο .env.example το οποιο έχει οριστει ως ενα αρχικό προτυπο το οποιο ο εγκαταστάτης της εφαρμογης θα αντιγράψει και να προσαρμοσεί στα δικά του δεδομένα.

Τo αρχείο composer.json πέρίέχει τα πακετα που εχουν οριστεί οτι χρείαζοντε για να λυτουργήσει η εφαρμογη και τα οποια εγκαθηστα ο composer όταν εκτελείτε. Ενώ το composer.lock περιέχει πληρόφοριες για το ποιες εκδωσεις των πακετων ειχαν χρήσιμοποιήθει . Αυτο το αρχείο θα χρισιμοποιηθεί εάν εκτελεστει η εντολη composer install για να εγκαταστησει τις ιδιες εκδωσεις .

Τα αρχεία package.json και package-lock.json ειναι τα αντιτοιχα του composer για το npm.

Vite.confing.js Αυτό το αρχείο ρυθμίζει το Vite, ένα σύγχρονο εργαλείο επεξεργασίας και δέσμης (bundling) για frontend assets, όπως JavaScript, CSS, και Vue components, που χρησιμοποιείται στο Laravel.

* Ορίζει σημεία εισόδου για τα assets της εφαρμογής.
* Ενσωματώνει plugins, όπως Vue ή React.
* Παρέχει δυνατότητες προσαρμογής για τον development server και την παραγωγή.

artisan είναι η γραμμή εντολών (CLI) του Laravel, που παρέχει μια σειρά εντολών για την υποστήριξη της ανάπτυξης.

.gitignore Το αρχείο .gitignore καθορίζει ποια αρχεία και φάκελοι πρέπει να εξαιρούνται από την έκδοση στο Git.

### The App Directory

The majority of your application is housed in the app directory. By default, this directory is namespaced under App and is autoloaded by Composer using the PSR-4 autoloading standard.

The app directory contains a variety of additional directories such as Console, Http, and Providers. Think of the Console and Http directories as providing an API into the core of your application. The HTTP protocol and CLI are both mechanisms to interact with the application, but do not actually contain application logic. In other words, they are two ways of issuing commands to your application. The Console directory contains all of your Artisan commands, while the Http directory contains your controllers, middleware, and requests.

A variety of other directories will be generated inside the app directory as you use the make Artisan commands to generate classes. So, for example, the app/Mails directory hadn’t exist until you i execute the make:Mail Artisan command to generate a job class.

#### The Broadcasting Directory

The Broadcasting directory contains all of the broadcast channel classes for the application. These classes are generated using the make:channel artisan command.

#### The Console Directory

The Console directory contains all of the custom Artisan commands for the application.This directory also houses your console kernel, which is where the custom Artisan commands are registered and can scheduled tasks defined.

#### The Enum Directory

The Enum directory contains all the custom Enum Classes for The application.

#### The Events Directory

This directory does not exist by default, but has been created by the event:generate and make:event Artisan commands. The Events directory houses event classes. Events may be used to alert other parts of your application that a given action has occurred, providing a great deal of flexibility and decoupling.

#### The Exceptions Directory

The Exceptions directory contains the application's exception handler and is also a good place to place any exceptions thrown by the application.

#### The Http Directory

The Http directory contains all the controllers, middleware, and form requests. Almost all of the logic to handle requests entering the application has been placed in this directory.

#### ~~The Jobs Directory~~

~~This directory does not exist by default, but will be created for you if you execute the make:job Artisan command. The Jobs directory houses the queueable jobs for your application. Jobs may be queued by your application or run synchronously within the current request lifecycle. Jobs that run synchronously during the current request are sometimes referred to as "commands" since they are an implementation of the command pattern.~~

#### ~~The Listeners Directory~~

~~This directory does not exist by default, but will be created for you if you execute the event:generate or make:listener Artisan commands. The Listeners directory contains the classes that handle your events. Event listeners receive an event instance and perform logic in response to the event being fired. For example, a UserRegistered event might be handled by a SendWelcomeEmail listener.~~

#### The Mail Directory

This directory does not exist by default, but has been created when executed the make:mail Artisan command. The Mail directory contains all of your classes that represent emails sent by your application. Mail objects allows to encapsulate all of the logic of building an email in a single, simple class that may be sent using the Mail::send method.

#### The Models Directory

The Models directory contains all of the Eloquent model classes. The Eloquent ORM included with Laravel provides a beautiful, simple ActiveRecord implementation for working with the database. Each database table has a corresponding "Model" which is used to interact with that table. Models allow to query for data in the tables, as well as insert new records into the table.

#### ~~The Notifications Directory~~

~~This directory does not exist by default, but will be created for you if you execute the make:notification Artisan command. The Notifications directory contains all of the "transactional" notifications that are sent by your application, such as simple notifications about events that happen within your application. Laravel's notification feature abstracts sending notifications over a variety of drivers such as email, Slack, SMS, or stored in a database.~~

#### The Observers Directory

This directory does not exist by default, but has been created when the artisan command make:observer first time executed. Here can defined one class per model that will listen to events of the model and will executed the correspond code when that event emitted. That events are “created”,”updated”,”deleted”, “restored”,”forceDeleted”.

#### The Policies Directory

This directory does not exist by default, but has been created when the artisan command make:policy first time executed. The Policies directory contains the authorization policy classes for your application. Policies are used to determine if a user can perform a given action against a resource.

#### The Providers Directory

The Providers directory contains all of the service providers for your application. Service providers bootstrap your application by binding services in the service container, registering events, or performing any other tasks to prepare your application for incoming requests.

In a fresh Laravel application, this directory will already contain several providers. You are free to add your own providers to this directory as needed.

#### The Rules Directory

The rules directory does not exist by default, but has been created when the artisan command t make:rule executed for first Time. The Rules directory contains the custom validation rule objects for your application. Rules are used to encapsulate complicated validation logic in a simple object.

#### The Services Directory

The Services Directory was also a custom directory. This directory contain logic that was indipended from the controller that was called and to be more easy to update.

#### The Traits Directory

The Traits directory is used to store reusable pieces of functionality that can be shared across different classes using PHP traits.

### The resources/js Directory

The js directory is where has been stored all the frontend logic. The directory has been organise by the following logic

* File main.js :The entry point of the application, where the Vue instance is initialized.
* Folder components: Contains reusable Vue.js components that form the building blocks of the user interface.
* Folder config: Stores application-wide configuration settings.
* Folder enums: Defines enumerations for managing consistent values across the application.
* Folder errors : Custom error classes for specific scenarios.
* Folder guards: Implements navigation guards to control access to routes in the application.
* Folder locales:Stores translations and localization data for multilingual support.
* Folder models: Represents the application’s data structures and business logic on the frontend.
* Folder pages: Contains Vue components representing full pages in the application.Has also subfolders with the same name that contain components that are used only on the specific pages.
* Folder plugins: Manages third-party or custom Vue plugins for the application. Plugin such as axios, echo, i118n, vuetify, ziggy.
* Folder router : Defines the application’s routes and and the vue router instance.
* Folder services: Contains service classes for managing data to expose on the frontend .
* Folder store: Manages the application’s global state using Vuex.
* Folder utilities: Contains helper functions or utility classes used across the application.

# ΣΧΕΔΙΑΣΜΟΣ & ΑΝΑΠΤΥΞΗ – ΕΦΑΡΜΟΓΗΣ

## ΟΡΙΣΜΟΣ ΠΡΟΔΙΑΓΡΑΦΩΝ -+

Για τον σχεδιασμό της εφαρμογής, λήφθηκαν υπ’ όψει οι παρακάτω προδιαγραφές:

1. Σχεδιασμός: Η ιστοσελίδα πρέπει να έχει ένα ελκυστικό σχεδιασμό, ώστε να δημιουργεί μία ευχάριστη εμπειρία στον χρήστη.
2. Πλοήγηση: Η ιστοσελίδα πρέπει να έχει μια εύκολη και κατανοητή πλοήγηση, ώστε να βοηθάει τους χρήστες να βρίσκουν εύκολα τις απαραίτητες πληροφορίες.
3. Ασφάλεια: Η ιστοσελίδα πρέπει να είναι ασφαλής και να παρέχει προστασία στους χρήστες της. Η πρόσβαση σε ευαίσθητες πληροφορίες πρέπει να απαιτεί αυθεντικοποίηση.
4. Συμβατότητα: Η ιστοσελίδα πρέπει να είναι συμβατή με διάφορους φυλλομετρητές και συσκευές (όπως κινητό ή υπολογιστή).
5. Ταχύτητα: Η ιστοσελίδα πρέπει να φορτώνει στις ελάχιστες δυνατές χρονικές περιόδους, ώστε να μην αποθαρρύνει τους χρήστες.
6. Χρήση: Η ιστοσελίδα πρέπει να είναι εύκολη στη χρήση, χωρίς πολλά περίπλοκα μενού ή ανεπιθύμητες διαδικασίες.
7. Φυσική Παρουσία: Η ιστοσελίδα να απαιτεί τη λιγότερο δυνατή φυσική παρουσία του χρήστη, για διευκόλυνση των διαδικασιών.
8. Επεκτασιμότητα: Η ιστοσελίδα πρέπει να κατασκευαστεί με τρόπο που να μπορεί να επεκταθεί στο μέλλον με νέες λειτουργίες και χαρακτηριστικά, χωρίς να απαιτείται μείωση της απόδοσης ή της λειτουργικότητας.

## ΧΡΗΣΤΕΣ ΕΦΑΡΜΟΓΗΣ

### Ελεγκτής Αιτήσεων

Οι χρήστες της εφαρμογής χωρίζονται σε κατηγορίες ανάλογα με τον ρόλο και τις αρμοδιότητές τους. Οι κύριες ομάδες χρηστών περιλαμβάνουν:

1. **Υπάλληλοι** που ασχολούνται με τη σίτιση (υποδοχής καρτών, πώλησης κουπονιών, και ελέγχου αιτήσεων).
2. **Δικαιούχοι σίτισης**, που μπορεί να είναι είτε φοιτητές είτε προσωπικό του Πανεπιστημίου.

### Ελεγκτής Αιτήσεων

Ο ελεγκτής έχει τη δυνατότητα να δει τα στοιχεία του αιτούντα καθώς και τα δικαιολογητικά που έχει ανεβάσει.

#### ****Διαχείριση Δικαιολογητικών****

Για κάθε δικαιολογητικό, ο ελεγκτής μπορεί να:

1. **Προβολή:** Να δει το περιεχόμενο του δικαιολογητικού.
2. **Χαρακτηρισμός:** Να το χαρακτηρίσει ως:
   1. **Έγκυρο:** Εάν το δικαιολογητικό είναι σωστό και πλήρες.
   2. **Ελλιπές:** Εάν το δικαιολογητικό δεν πληροί τις απαιτούμενες προϋποθέσεις και απαιτεί επιπλέον πληροφορίες ή έγγραφα.
   3. **Απορριπτέο:** Εάν το δικαιολογητικό είναι λάθος ή μη αποδεκτό.

#### ****Αξιολόγηση Αίτησης****

Ο ελεγκτής έχει την επιλογή να χαρακτηρίσει την αίτηση ως:

1. **Εγκεκριμένη:** Εάν όλα τα δικαιολογητικά είναι σωστά και πληρούν τις προϋποθέσεις.
2. **Απορριπτέα:** Εάν η αίτηση δεν πληροί τις προϋποθέσεις για δωρεάν σίτιση.
3. **Ελλιπής:** Εάν τα δικαιολογητικά ή τα στοιχεία είναι ελλιπή και απαιτείται η συμπλήρωσή τους από τον αιτούντα.

Σε κάθε απόφαση, ο ελεγκτής πρέπει να παρέχει ένα σχόλιο, προκειμένου να αιτιολογήσει την απόφασή του στον αιτούντα.

### ****Υπάλληλος Υποδοχής Καρτών****

Ο υπάλληλος υποδοχής καρτών έχει ως αρμοδιότητα τον έλεγχο πρόσβασης των δικαιούχων σίτισης.

#### ****Διαδικασία Ελέγχου****:

1. Σαρώσει το barcode της κάρτας του δικαιούχου για να επιβεβαιώσει εάν διαθέτει δικαίωμα σίτισης.
2. Ενημερώνεται άμεσα για την έγκριση ή την απόρριψη της εισόδου.
3. Πράττει ανάλογα με την κατάσταση, π.χ., επιτρέπει ή απορρίπτει την είσοδο.

#### ****Δυνατότητες****:

Εξαγωγή στατιστικών στοιχείων για τον αριθμό των ατόμων που σιτίστηκαν, αναλυμένα ανά τρόπο σίτισης (δωρεάν ή με κουπόνια).

### ****Πωλητής Κουπονιών****

Ο πωλητής κουπονιών είναι υπεύθυνος για την πώληση κουπονιών σίτισης.

#### ****Διαδικασία Πώλησης****:

1. Σαρώσει το barcode της κάρτας του δικαιούχου σίτισης.
2. Δηλώνει τον αριθμό των κουπονιών που αγοράστηκαν, ανά κατηγορία (πρωινό, μεσημεριανό, βραδινό).

#### ****Δυνατότητες****:

Εξαγωγή στατιστικών στοιχείων για τον αριθμό των ατόμων που σιτίστηκαν, αναλυμένα ανά τρόπο πληρωμής.

### ****Ελεγκτής Αιτήσεων Δωρεάν Σίτισης****

Ο ελεγκτής αιτήσεων είναι υπεύθυνος για την αξιολόγηση των αιτήσεων δωρεάν σίτισης.

#### ****Διαδικασία Ελέγχου****

1. Ελέγχει εάν ο αιτών πληροί τις προϋποθέσεις που απαιτούνται.
2. Αποφασίζει εάν η αίτηση θα εγκριθεί, απορριφθεί ή θα χαρακτηριστεί ως ελλιπής (λόγω έλλειψης δικαιολογητικών).

#### ****Δυνατότητες****:

1. Εύρεση και επεξεργασία αιτήσεων.
2. Ενημέρωση της κατάστασης μιας αίτησης (π.χ., έγκριση, απόρριψη, εκκρεμής).

### ****Δικαιούχος Σίτισης****

Οι δικαιούχοι σίτισης διαχωρίζονται σε δύο κατηγορίες: **φοιτητές** και **προσωπικό Δ.Ε.Π./Δ.Τ./Δ.Π.**

#### ****Φοιτητής****

#### ****Δικαιώματα Σίτισης****:

1. Είσοδος με κουπόνια που έχει αγοράσει.
2. Δωρεάν είσοδος, εάν έχει υποβάλει αίτηση για δωρεάν σίτιση και αυτή έχει εγκριθεί.

#### ****Δυνατότητες****:

1. Υποβολή αίτησης για δωρεάν σίτιση κάθε ακαδημαϊκό έτος, με επισύναψη των απαραίτητων δικαιολογητικών.
2. Ενημέρωση για την απόφαση του ελεγκτή σχετικά με την αίτησή του.
3. Δυνατότητα υποβολής επιπρόσθετων δικαιολογητικών σε περίπτωση που η αίτηση χαρακτηριστεί ως ελλιπής.
4. Προβολή ιστορικού κινήσεων στο εστιατόριο.

#### ****Προσωπικό Δ.Ε.Π./Δ.Τ./Δ.Π.****

#### ****Δικαιώματα Σίτισης****:

Είσοδος μόνο μέσω κουπονιών, τα οποία έχουν υψηλότερο κόστος από αυτά του φοιτητή.

#### ****Δυνατότητες****:

1. Προβολή του τρέχοντος υπολοίπου κουπονιών μέσα από την ιστοσελίδα.
2. Επισκόπηση προηγούμενων κινήσεων για σίτιση.
3. Δυνατότητα αποστολής κουπονιών σε άλλους δικαιούχους του ίδιου κόστους.

## Πορεία Αίτησης Δωρεάν Σίτησης

Η παρακάτω ανάλυση περιγράφει τα βήματα για την έγκριση μιας αίτησης σίτισης φοιτητή. Η διαδικασία χωρίζεται σε τρία βασικά στάδια: **υποβολή από τον φοιτητή**, **έλεγχος από τον ελεγκτή** και **χρήση της αίτησης από τον φοιτητή μετά την έγκριση**.

### ****Υποβολή Αίτησης από τον Φοιτητή****

#### ****Έλεγχος Στοιχείων:****

1. Ο φοιτητής εισέρχεται στο σύστημα και ελέγχει τα προσωπικά του δεδομένα.
2. Εάν υπάρχουν σφάλματα ή ελλείψεις, επικοινωνεί με το σύστημα του πανεπιστημίου για την τροποποίησή τους.
3. Εφόσον τα στοιχεία είναι σωστά, προχωρά στο επόμενο βήμα.

#### ****Κατάθεση Δικαιολογητικών:****

1. Ο φοιτητής επιλέγει και ανεβάζει τα απαιτούμενα δικαιολογητικά.
2. Έχει τη δυνατότητα να:
   1. **Διαγράψει** ένα δικαιολογητικό.
   2. **Αλλάξει κατηγορία** σε ένα ανεβασμένο δικαιολογητικό.
   3. Να δεί το δικαιολογητικό.
   4. Υποβάλλει την αίτηση.

#### Προβολή Κατατεθημένης αίτησης

Το σύστημα ενημερώνει τον φοιτητή ότι η αίτηση βρίσκεται σε **αναμονή ελέγχου**.

### ****Έλεγχος Αίτησης από τον Ελεγκτή****

Ο ελεγκτής εξετάζει τα δεδομένα της αίτησης και τα δικαιολογητικά του φοιτητή. Οι ενέργειες του ελεγκτή περιλαμβάνουν:

#### ****Αποφάσεις για τα Δικαιολογητικά:****

1. Αν κάποιο δικαιολογητικό είναι αιτία απόρριψης, μπορεί να το απορρίψει.
2. Εάν κάποιο δικαιολογητικό είναι ελλιπές, χαρακτηρίζει την αίτηση ως **ελλιπή**.
3. Αν όλα τα δικαιολογητικά είναι σωστά, προχωρά στην απόφαση για την αίτηση.

#### ****Αποφάσεις για την Αίτηση:****

1. **Έγκριση:** Ο ελεγκτής υποχρεούται να δηλώσει την **ημερομηνία λήξης της αίτησης**.
2. **Απόρριψη:** Πρέπει να καταχωρίσει ένα σχόλιο, ώστε ο φοιτητής να γνωρίζει τον λόγο της απόρριψης.
3. **Χαρακτηρισμός ως Ελλιπής:** Ο φοιτητής καλείται να διορθώσει την αίτηση ανεβάζοντας νέα δικαιολογητικά.

## ****Επεξεργασία Ελλιπούς Αίτησης από τον Φοιτητή****

Εάν η αίτηση χαρακτηριστεί ελλιπής:

1. Ο φοιτητής μπορεί να υποβάλει νέα δικαιολογητικά.
2. Μπορεί να διαγράψει δικαιολογητικά που δεν έχουν εγκριθεί ή απορριφθεί.
3. Υποβάλλει εκ νέου την αίτηση για επανέλεγχο.

## ****Χρήση της Έγκρισης για Σίτιση****

Εφόσον η αίτηση εγκριθεί: Ο αιτούντας χρησιμοποιεί την έγκριση για να σιτίζεται στο εστιατόριο. Κατά την είσοδο στο εστιατόριο:

1. Σαρώνεται το **barcode** του αιτούντα.
2. Καταγράφεται η είσοδός του και του αποδίδεται μία μερίδα ανά γεύμα.

## ****ΒΑΣΗ**** ΔΕΔΟΜΕΝΩΝ ---

Η βάση δεδομένων είναι απαραίτητη για την αποθήκευση, οργάνωση και διατήρηση μεγάλων όγκων δεδομένων μίας εφαρμογής. Επίσης, η ανάκτηση αυτών πρέπει να γίνεται με αποδοτικό τρόπο. Στην παρούσα εφαρμογή, τα δεδομένα αυτά αναφέρονται στους χρήστες, τις αιτήσεις τα αρχεία κλ. Για την διαχείριση των δεδομένων χρησιμοποιήθηκε η σχεσιακή βάση δεδομένων MySQL, που προσφέρει υψηλές επιδόσεις, αξιοπιστία και ασφάλεια.

## Γέμισμα Βάσης Δεδομένων ---

Για την βάση δεδομένων στη διπλωματική εργασία, αξιοποιήθηκαν τα **Seeders** και τα **Factories** που παρέχει το Laravel, σε συνδυασμό με τη βιβλιοθήκη **Faker**. Αυτή η διαδικασία αυτοματοποίησε τη δημιουργία και εισαγωγή μεγάλου όγκου τυχαίων δεδομένων στη βάση, προσφέροντας τα εξής πλεονεκτήματα:

1. ****Seeders****:
   1. Οι seeders χρησιμοποιήθηκαν για να γεμίσουν συγκεκριμένους πίνακες της βάσης δεδομένων με προκαθορισμένα ή τυχαία δεδομένα.
   2. Με τον Artisan Command php artisan db:seed, η βάση δεδομένων μπορούσε να γεμίζει εύκολα και γρήγορα σε περιβάλλον ανάπτυξης ή δοκιμών.
   3. Παράδειγμα: Δημιουργία αρχείων seeders για βασικά δεδομένα, όπως κατηγορίες προϊόντων, ρόλοι χρηστών ή σταθερές τιμές που απαιτούνται για τη λειτουργία της εφαρμογής.
2. ****Factories****:
   1. Τα factories χρησιμοποιήθηκαν για τη μαζική δημιουργία τυχαίων εγγραφών σε πίνακες, αξιοποιώντας το **Faker** για τη δημιουργία ψευδοτυχαίων δεδομένων.
   2. Κάθε factory ορίζεται με τρόπο που περιγράφει τη δομή του πίνακα και τον τρόπο γέμισματός του. Για παράδειγμα, για έναν πίνακα χρηστών, το factory μπορεί να παράγει τυχαία ονόματα, email, τηλέφωνα, και διευθύνσεις.
   3. Χρησιμοποιώντας τη μέθοδο factory()→for() μπορουσαν να διμιιουργηθουν νεεες εγρρααφεφς μου να συστετιζοντε με εγραφές αλλων πινάκων
3. ****Faker Library****:
   1. Η βιβλιοθήκη Faker είναι ενσωματωμένη στο Laravel και παρέχει πληθώρα επιλογών για τη δημιουργία ρεαλιστικών τυχαίων δεδομένων.
   2. Παραδείγματα:
      1. Ονόματα ($faker->name)
      2. Email ($faker->email)
      3. Ημερομηνίες ($faker->date)
      4. Τηλέφωνα ($faker->phoneNumber)
      5. Κείμενο ($faker->text)
   3. Αυτό επέτρεψε τη δημιουργία δεδομένων που μοιάζουν ρεαλιστικά, δίνοντας στις δοκιμές της εφαρμογής έναν πιο πρακτικό χαρακτήρα.

## ΛΕΙΤΟΥΡΓΙΕΣ ΕΦΑΡΜΟΓΗΣ ---

Σε αυτό το μέρος θα γίνει αναλυτική περιγραφή των λειτουργιών της εφαρμογής και θα μελετηθεί κάθε σελίδα ξεχωριστά με βάση το είδος του χρήστη.

### ****Διαχείριση Πρώτης Σύνδεσης Χρήστη****

Όταν ένας χρήστης συνδέεται για πρώτη φορά, το σύστημα δεν διαθέτει αποθηκευμένες πληροφορίες για αυτόν. Αντιλαμβανόμενο αυτήν την κατάσταση, το σύστημα τον ανακατευθύνει σε μια ειδική σελίδα προβολής των προσωπικών του δεδομένων, τα οποία αντλούνται από το σύστημα σύνδεσης.

Σε αυτήν τη σελίδα, ο χρήστης έχει δύο επιλογές:

1. **Αποδοχή Δεδομένων:** Εάν αποδεχτεί την αποθήκευση των δεδομένων του, αυτά καταχωρούνται στη βάση δεδομένων και ο χρήστης μπορεί να συνεχίσει κανονικά στην εφαρμογή.
2. **Απόρριψη Δεδομένων:** Εάν επιλέξει απόρριψη, τα δεδομένα δεν αποθηκεύονται στη βάση δεδομένων και ο χρήστης αποσυνδέεται αυτόματα από το σύστημα.
3. Εάν ο χρήστης είχε προσπαθήσει να μεταβεί σε μια συγκεκριμένη σελίδα πριν την ανακατεύθυνση, με την αποδοχή των δεδομένων ανακατευθύνεται σε αυτήν τη σελίδα. Διαφορετικά, παραμένει στη σελίδα προφίλ.

### ****Διαχείριση Ελέγχου Εισόδου μέσω Συστήματος Barcode****

Όταν συνδέεται ένα άτομο με δικαίωμα ελέγχου εισόδου (π.χ., ελεγκτής), εμφανίζεται η παρακάτω λειτουργικότητα:

1. Το σύστημα έχει ρυθμιστεί ώστε να επικεντρώνεται στο **πεδίο εισόδου για το ID**.
2. Όταν γίνεται **σκανάρισμα** του barcode από ειδική συσκευή, το ID του χρήστη εμφανίζεται αυτόματα στο πεδίο και αποστέλλεται για έλεγχο.
3. Ενημερώνετε ο χρήστης στο email του σε περιπτώση επιτυχούς κατανάλωσης

#### ****Διαδικασία Ελέγχου:****

1. **Έγκριση Εισόδου:**

Εάν ο χρήστης δικαιούται δωρεάν σίτιση ή διαθέτει έγκυρα κουπόνια, η είσοδός του εγκρίνεται και εμφανίζεται στην οθόνη σχετικό μήν­υμα επιβεβαίωσης.

1. **Απόρριψη Εισόδου:**

Εάν ο χρήστης δεν δικαιούται είσοδο, εμφανίζεται στην οθόνη μήνυμα απόρριψης.

#### ****Επιπλέον Δυνατότητες για Ελεγκτή:****

Ο ελεγκτής μπορεί να πατήσει τον συνδυασμό πλήκτρων **Ctrl + Alt + Space** για να δει ξανά το αποτέλεσμα της τελευταίας συναλλαγής

### ****Δυνατότητα Εξαγωγής Στατιστικών****

Τόσο ο ελεγκτής όσο και ο υπέυθυνος κουπονιών έχει τη δυνατότητα να εξάγει στατιστικά στοιχεία σχετικά με την είσοδο χρηστών στο σύστημα σίτισης. Οι επιλογές εξαγωγής περιλαμβάνουν:

1. **Τρέχον Γεύμα:** Στατιστικά μόνο για το τρέχον γεύμα. ( Μόνο στον ελεγκτή)
2. **Τρέχουσα Ημέρα:** Στατιστικά για όλα τα γεύματα της τρέχουσας ημέρας.
3. **Προσαρμοσμένη Περίοδος:**
   * Επιλογή συγκεκριμένης χρονικής περιόδου.
   * Δυνατότητα περιορισμού των δεδομένων σε συγκεκριμένο γεύμα (π.χ., μόνο πρωινό, μεσημεριανό, βραδινό).

#### ****Απεικόνιση Στατιστικών:****

Το αποτέλεσμα εμφανίζεται σε ένα αναδυόμενο παράθυρο και παρουσιάζεται αναλυτικά:

* **Ανά Γεύμα:** Διαχωρισμός στατιστικών ανά πρωινό, μεσημεριανό, βραδινό.
* **Ανά Κατηγορία Εισόδου:**
  + **Δωρεάν Σίτιση:** Χρήστες που δικαιούνται δωρεάν είσοδο. Μόνο για τον ελενκτή.
  + **Κουπόνια Φοιτητή:** Χρήστες με κουπόνια φοιτητών.
  + **Κουπόνια Εργαζομένων:** Χρήστες με κουπόνια εργαζομένων του πανεπιστημίου.

### ****Πώληση κουπονιών****

#### ****Καταχώριση Αγοράς Κουπονιών:****

Ο υπεύθυνος μπορεί να καταχωρίσει νέες αγορές κουπονιών μέσω μιας ειδικής φόρμας. Η διαδικασία περιλαμβάνει:

1. **Καταχώριση Πληροφοριών:**
   * **Αριθμός Κουπονιών:** Το ποσό των κουπονιών που αγοράστηκαν.
   * **ID Χρήστη:** Ο μοναδικός κωδικός του χρήστη που πραγματοποίησε την αγορά.

#### ****Επιλογές Διαχείρισης Συναλλαγής - Επιβεβαίωση:****

Ενημέρωση Στοιχείων Χρήστη:

Μετά την καταχώριση, το σύστημα εμφανίζει το όνομα του πελάτη (με ορισμένους χαρακτήρες αποκρυμμένους για λόγους ασφάλειας) και τον ρόλο του στο πανεπιστήμιο.

1. **Πίσω:** Επιστροφή για διόρθωση των στοιχείων της συναλλαγής.
2. Επιβεβαίωση**:** Αποθήκευση της αγοράς κουπονιών στη βάση δεδομένων.

#### ****Επιλογές Διαχείρισης Συναλλαγής - Ολοκλήρωση:****

**Προβολή Συναλλαγής και Επιστροφή για Νέα Πώληση:** Μετά την ολοκλήρωση, ο υπεύθυνος μπορεί να δει την καταγεγραμμένη συναλλαγή και να επιστρέψει στη φόρμα για καταχώριση νέας πώλησης.

Σε αυτό το σημείο ο αποδέκτης λαμβάνει ειδοποίηση στο ηλεκρονικό ταχυδρομείο του.

### ****Λειτουργίες Κατόχου Κουπονιών****

#### ****Ιστορικό Συναλλαγών****

Ο κάτοχος κουπονιών έχει τη δυνατότητα να βλέπει το ιστορικό των συναλλαγών του, το οποίο περιλαμβάνει όλες τις ενέργειες που αφορούν τη χρήση και τη διαχείριση των κουπονιών του.

#### ****Μεταφορά Κουπονιών****

Ο κάτοχος κουπονιών μπορεί να μεταφέρει κουπόνια σε άλλο άτομο που διαθέτει κουπόνια της ίδιας κατηγορίας. Η διεπαφή για τη μεταφορά είναι ίδια με αυτήν της πώλησης κουπονιών.

#### ****Ενημέρωση Υπολοίπου****

Ανεξάρτητα από το εάν βρίσκεται στη σελίδα μεταφοράς κουπονιών ή στο ιστορικό, ο κάτοχος ενημερώνεται σε πραγματικό χρόνο για το τρέχον υπόλοιπο των κουπονιών του.

### ****Λειτουργίες Δικαιούχου Δωρεάν Σίτισης****

#### ****Παρακολούθηση Τρέχουσας Αίτησης****

Ένα άτομο με δικαίωμα υποβολής αιτήσεων για δωρεάν σίτιση:

* Βλέπει την κατάσταση της τρέχουσας αίτησής του, εφόσον έχει υποβάλει μία.
* Ενημερώνεται για την ημερομηνία λήξης του δικαιώματός του για δωρεάν σίτιση.

#### ****Ιστορικό Δωρεάν Σίτισης****

Στη σελίδα ιστορικού, ο δικαιούχος μπορεί να δει αναλυτικά τις εισόδους του στο εστιατόριο, όπως πότε και για ποιο γεύμα χρησιμοποιήθηκε το δικαίωμά του.

### ****Υποβολη Αιτήσεων Δωρεάν Σίτισης****

#### ****Τρέχουσα Αίτηση****

Ο πιθανός δικαιούχος δωρεάν σίτισης μπορεί:

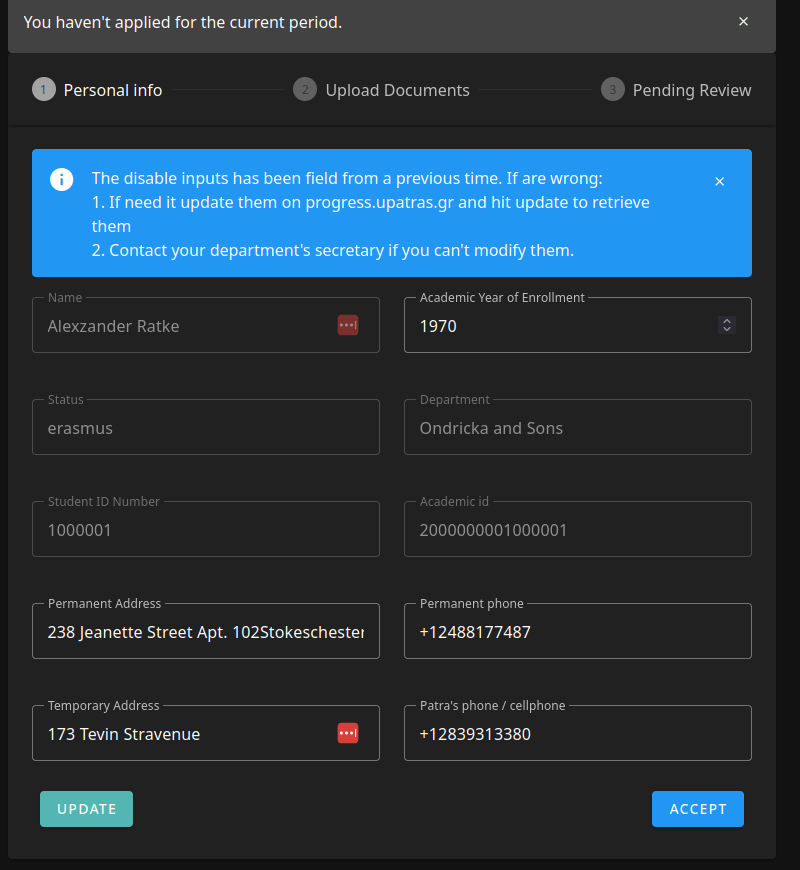
* Να υποβάλει νέα αίτηση.
* Να τροποποιήσει ή να δει την τρέχουσα αίτησή του.

#### ****Διαδικασία για Νέα Αίτηση****

Αν δεν υπάρχει αίτηση για την τρέχουσα περίοδο, ο χρήστης:

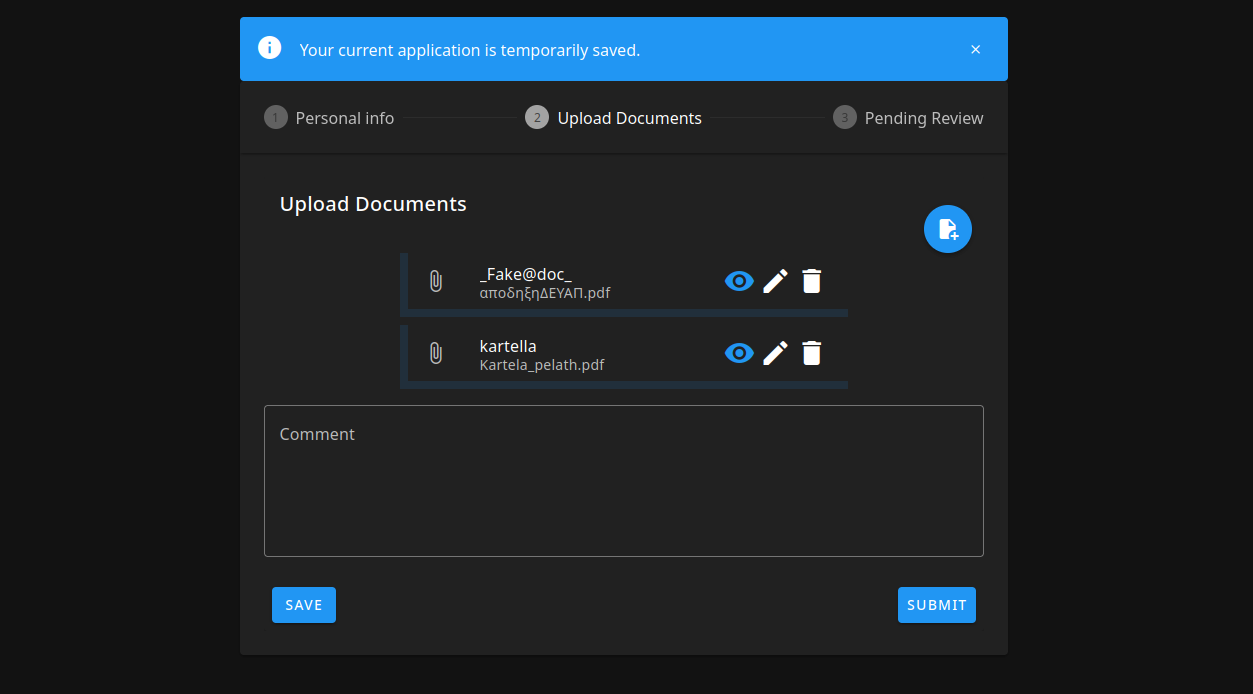
**Βήμα 1: Επεξεργασιά στοιχείων**

* 1. **Ελέγχει τα Αποθηκευμένα Στοιχεία:**
  + Αν πρόκειται για την πρώτη αίτηση ή τα στοιχεία έχουν διαγραφεί λόγω παλαιότητας, θα χρειαστεί να αντλήσει τα δεδομένα του από το σύστημα του πανεπιστημίου.
  + Εμφανίζονται τα στοιχεία του που είναι αποθηκευμένα στον διακομιστή.
  + Εάν τα στοιχεία είναι σωστά, μπορεί να συνεχίσει στο επόμενο βήμα.



* 1. **Επεξεργασία Στοιχείων:**
  + Τα πεδία που δεν συμπληρώνονται αυτόματα από το σύστημα του πανεπιστημίου έχουν δυνατότητα επεξεργασίας.
  + Εάν χρειάζεται αλλαγή στοιχείων, ο χρήστης ενημερώνεται να επικοινωνήσει με τη γραμματεία ή να συνδεθεί σε σχετική υπηρεσία του πανεπιστημίου.

**Βημα 2: Διαχείρηση Δικαιολογητικών**



**O χρήστης μπορεί να ανεβάσει νέα έγγραφα**

Nα δει τα ήδη υπάρχοντα, ή να διαγράψει όσα δεν έχουν ακόμα ελεγχθεί ή έχουν χαρακτηριστεί ελλιπή .

**Βήμα 3: Υποβολή**

Ο χρήστης μετα την υποβλη βλέπει τα δεδομένα και τα δικαιολογητικά που έχει υποβάλει, καθώς και την τρέχουσα κατάσταση της αίτησης.

### Έ****λεγχος Αιτήσεων Δωρεάν Σίτισης****

**Ο ελεγκτής έχει τη δυνατότητα να δει τα στοιχεία του αιτούντα καθώς και τα δικαιολογητικά που έχει ανεβάσει.**

#### ****Διαχείριση Δικαιολογητικών****

Για κάθε δικαιολογητικό, ο ελεγκτής μπορεί να:

* Να δει το περιεχόμενο του δικαιολογητικού.
* Να το χαρακτηρίσει ως:
  + **Έγκυρο:** Εάν το δικαιολογητικό είναι σωστό και πλήρες.
  + **Ελλιπές:** Εάν το δικαιολογητικό δεν πληροί τις απαιτούμενες προϋποθέσεις και απαιτεί επιπλέον πληροφορίες ή έγγραφα.
  + **Απορριπτέο:** Εάν το δικαιολογητικό είναι λάθος ή μη αποδεκτό.

**Αξιολόγηση Αίτησης**

Ο ελεγκτής έχει την επιλογή να χαρακτηρίσει την αίτηση ως:

* **Εγκεκριμένη:** Εάν όλα τα δικαιολογητικά είναι σωστά και πληρούν τις προϋποθέσεις.
* **Απορριπτέα:** Εάν η αίτηση δεν πληροί τις προϋποθέσεις για δωρεάν σίτιση.
* **Ελλιπής:** Εάν τα δικαιολογητικά ή τα στοιχεία είναι ελλιπή και απαιτείται η συμπλήρωσή τους από τον αιτούντα.

Σε κάθε απόφαση, ο ελεγκτής πρέπει να παρέχει ένα σχόλιο, προκειμένου να αιτιολογήσει την απόφασή του στον αιτούντα.

# ΑΞΙΟΛΟΓΗΣΗ ΕΦΑΡΜΟΓΗΣ ---

## ΜΕΘΟΔΟΣ ΑΞΙΟΛΟΓΗΣΗΣ ---

u[28]

## ΧΡΗΣΤΕΣ ΑΞΙΟΛΟΓΗΣΗΣ ---

## ΣΕΝΑΡΙΑ ΧΡΗΣΗΣ ---

## ΑΠΟΤΕΛΕΣΜΑΤΑ ---

## ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΕΠΕΚΤΑΣΕΙΣ ---

## ΣΥΜΠΕΡΑΣΜΑΤΑ ---

## ΒΕΛΤΙΩΣΕΙΣ & ΕΠΕΚΤΑΣΕΙΣ ---

## ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] ‘Σίτιση’, Πανεπιστήμιο Πατρών. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://www.upatras.gr/foitites/foititiki-merimna/sitisi/

[2] ‘About npm’. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://docs.npmjs.com/about-npm

[3] ‘Why Vite | Vite’. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://vite.dev/guide/why.html

[4] ‘JavaScript - Βικιπαίδεια’. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://el.wikipedia.org/wiki/JavaScript

[5] ‘Vue.js - The Progressive JavaScript Framework | Vue.js’. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://vuejs.org/

[6] ‘What is Vuex? | Vuex’. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://vuex.vuejs.org/

[7] ‘Vuetify — A Vue Component Framework’. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://vuetifyjs.com/en/

[8] ‘What is Vue I18n? | Vue I18n’. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://vue-i18n.intlify.dev/guide/introduction.html

[9] ‘Introduction | i18next documentation’. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://www.i18next.com

[10] ‘Introduction | Vue Router’. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://router.vuejs.org

[11] ‘echo/README.md at master · laravel/echo’. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://github.com/laravel/echo/blob/master/README.md

[12] ‘Getting Started | Axios Docs’. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://axios-http.com/docs/intro

[13] ‘CSS - Wikipedia’. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://en.wikipedia.org/wiki/CSS

[14] ‘GitHub’, *Wikipedia*. 26 Δεκέμβριος 2024. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://en.wikipedia.org/w/index.php?title=GitHub&oldid=1265299480

[15] ‘Apache HTTP Server Project’. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://httpd.apache.org/ABOUT\_APACHE.html

[16] ‘MySQL’. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://www.mysql.com/

[17] ‘PHP: Introduction - Manual’. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://www.php.net/manual/en/introduction.php

[18] ‘phpMyAdmin’. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://www.phpmyadmin.net/

[19] ‘Introduction - Composer’. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://getcomposer.org/doc/00-intro.md

[20] ‘Laravel - The PHP Framework For Web Artisans’. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://laravel.com/

[21] ‘Laravel - Telescope’. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://laravel.com/docs/10.x/telescope

[22] ‘FakerPHP / Faker’. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://fakerphp.org/

[23] B. vd Heuvel, *barryvdh/laravel-dompdf*. (28 Δεκέμβριος 2024). PHP. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://github.com/barryvdh/laravel-dompdf

[24] *tighten/ziggy*. (28 Δεκέμβριος 2024). JavaScript. Tighten. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://github.com/tighten/ziggy

[25] Propaganistas, *Propaganistas/Laravel-Phone*. (28 Δεκέμβριος 2024). PHP. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://github.com/Propaganistas/Laravel-Phone

[26] ‘Laravel Sanctum’. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://laravel.com/docs/10.x/sanctum

[27] *laravel/ui*. (25 Δεκέμβριος 2024). PHP. The Laravel Framework. Ημερομηνία πρόσβασης: 28 Δεκέμβριος 2024. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://github.com/laravel/ui

[28] Ν. Αβούρης *κ.ά.*, *Εισαγωγή στην Αλληλεπίδραση Ανθρώπου - Υπολογιστή*. 2016.