



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

IHUCconnect: Διαδικτυακή Πύλη του Διεθνούς
Πανεπιστημίου της Ελλάδος για φοιτητές Erasmus -
Ανάπτυξη διαδικτυακής εφαρμογής και εφαρμογής για κινητές συσκευές



Των φοιτητών
Γιώτα Αλεξάνδρου
Αρ. Μητρώου: 073268

Θεοφανίδη Ηρακλή
Αρ. Μητρώου: 164664

Επιβλέπων
Μπράτσας Χαράλαμπος
Επίκουρος καθηγητής

Ιανουάριος 2024

Τίτλος Π.Ε.: IHUConnect: Διαδικτυακή Πύλη του Διεθνούς Πανεπιστημίου της Ελλάδος για φοιτητές
Erasmus - Ανάπτυξη διαδικτυακής εφαρμογής και εφαρμογής για κινητές συσκευές
Κωδικός Π.Ε. 23163

Όνοματεπώνυμο φοιτητών Γιώτας Αλέξανδρος – Θεοφανίδης Ηρακλής

Όνοματεπώνυμο εισηγητή Μπρατσάς Χαράλαμπος

Ημερομηνία ανάληψης Π.Ε. 22/03/2023

Ημερομηνία περάτωσης Π.Ε. 26/01/2024

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

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

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

Πρόλογος

Η παρούσα εργασία εκπονήθηκε από κοινού από τον Ηρακλή Θεοφανίδη και τον Αλέξανδρο Γιώτα υπό την καθοδήγηση και επίβλεψη κ. Παναγιώτη Αδαμίδη. Εμπνευσμένοι από την ανάγκη για ενίσχυση και βελτίωση της φοιτητικής εμπειρίας για τους φοιτητές Erasmus, η παρούσα εφαρμογή αποτελεί μια καινοτόμο προσέγγιση για την παροχή υψηλής ποιότητας υπηρεσιών και πληροφοριών.

Με ευαισθησία στις σύγχρονες ανάγκες των φοιτητών Erasmus, η εφαρμογή προσφέρει ένα ολοκληρωμένο περιβάλλον πληροφόρησης και διαχείρισης, καλύπτοντας εκτενώς την ποικιλία των αναγκών τους. Το διαχειριστικό σύστημα που συνοδεύει την εφαρμογή αποτελεί έναν καθοριστικό παράγοντα για την επιτυχία της, παρέχοντας αποτελεσματικό και φιλικό περιβάλλον για τους διαχειριστές.

Συνοψίζοντας, η εφαρμογή αντιπροσωπεύει ένα βήμα προς τη δημιουργία μιας πλήρους και προηγμένης πλατφόρμας που στοχεύει στη βελτίωση της φοιτητικής εμπειρίας. Μέσω της συνεχούς ανάπτυξης και εξέλιξης, η εφαρμογή επιδιώκει να ανταποκριθεί στις ποικίλες ανάγκες των φοιτητών Erasmus, προσφέροντας τους έναν ολοκληρωμένο και προσαρμοσμένο ψηφιακό βοηθό για τη διάρκεια της φοιτητικής τους περιόδου στο πλαίσιο του προγράμματος Erasmus του ΔΙ.ΠΑ.Ε.

Περίληψη

Η τρέχουσα πτυχιακή εργασία εστιάζει στην ανάπτυξη μιας ολοκληρωμένης εφαρμογής που αποτελείται από δύο μέρη: μια εφαρμογή για κινητά και μια για τον ιστό. Η εργασία επικεντρώνεται στην κάλυψη των αναγκών των εισερχομένων φοιτητών του προγράμματος Erasmus+ στο Διεθνές Πανεπιστήμιο της Ελλάδος. Ο κύριος στόχος της εργασίας είναι να βελτιώσει την εμπειρία των φοιτητών Erasmus+ παρέχοντάς τους μια φιλική προς τον χρήστη πλατφόρμα που διευκολύνει τον αρχικό προσανατολισμό τους και τη γρήγορη πρόσβαση σε σημαντικές πληροφορίες κατά την πλοήγηση στο Πανεπιστήμιο. Η εφαρμογή ενσωματώνει πολλά σημαντικά χαρακτηριστικά, όπως προσαρμογή με βάση τη γλώσσα, την πόλη και την πανεπιστημιούπολη, έναν διαδραστικό χάρτη της και έναν ψηφιακό βοηθό με δυναμικά προγραμματιζόμενο διάλογο. Το Flutter Framework χρησιμοποιήθηκε για την εφαρμογή για κινητά και το Angular Framework για το τμήμα διαχείρισης ιστού της εφαρμογής. Επιπλέον, η ενοποίηση αυτών των τεχνολογιών με αυτές της πλατφόρμας Firebase, όπως η βάση δεδομένων Firestore, συνέβαλε στον συγχρονισμό δεδομένων σε πραγματικό χρόνο και στην αποτελεσματική διαχείριση δεδομένων. Η συνολική προσέγγιση που υιοθετήθηκε σε αυτό το έργο ενθαρρύνει ένα περιβάλλον χωρίς αποκλεισμούς και υποστηρικτικό για τους χρήστες του, απλοποιώντας την πρόσβασή τους σε κρίσιμες υπηρεσίες και ενισχύοντας την αίσθηση της ζωντανής κοινότητας. Προσπαθώντας να αντιμετωπίσει με επιτυχία τις μοναδικές ανάγκες των φοιτητών και των διαχειριστών ταυτόχρονα, η εφαρμογή παρέχει τη δυνατότητα σημαντικής βελτίωσης της συνολικής εμπειρίας Erasmus+, προωθώντας την αποτελεσματική επικοινωνία, την πολιτιστική ενσωμάτωση και την ευκολία πρόσβασης σε βασικές υπηρεσίες. Μέσω αυτού του έργου ελπίζουμε να δείξουμε τη δέσμευσή μας στην αξιοποίηση της τεχνολογίας για τη βελτίωση των εκπαιδευτικών εμπειριών και να εμβαθύνουμε τη συμμετοχή της κοινότητας.

IHUConnect: IHU's Online Gateway for Erasmus students - Web and mobile application development

Giotas Alexandros - Theofanidis Iraklis

Abstract

This thesis focuses on developing an integrated application comprising two components: a mobile application and a web application. The work focuses on meeting the needs of the incoming Erasmus+ program students at International Hellenic University. The main goal of the project is to improve the experience and optimize the early integration of Erasmus+ students by providing them with a user-friendly platform that facilitates their initial orientation and quick access to important information navigating around the University. The app incorporates many important features such as customization based on language, city and campus, interactive campus maps and a digital assistant with dynamically programmable dialogue. We leverage Flutter framework for the mobile application and Angular framework for the web management of the application. In addition, the integration of these technologies with those of the Firebase suite, such as the Firestore database, contributed to real-time data synchronization and efficient data management. The overall approach taken in this project fosters an inclusive and supportive environment for its users, simplifying access to crucial services and enhancing a sense of vibrant community. Trying to successfully address the unique needs of both the Erasmus+ students and the administrators, the app provides the potential to significantly improve the overall Erasmus+ experience, promoting effective communication, cultural integration and ease of access to essential services. Through this project we hope to demonstrate our commitment to leveraging technology to enhance educational experiences and deepen community engagement.

Περιεχόμενα

Πρόλογος.....	3
Περίληψη.....	4
Abstract.....	5
Περιεχόμενα.....	6
Κατάλογος Εικόνων.....	9
Συντομογραφίες.....	12
Κεφάλαιο 1ο: Εισαγωγή.....	13
1.1 Το πρόβλημα προσαρμογής και προσανατολισμού.....	13
1.2 Υπάρχουσες εφαρμογές ενημέρωσης και υποστήριξης.....	13
1.2.1 Ιστοσελίδες ΔΙ.ΠΑ.Ε.....	13
1.2.2 Ειδικές εφαρμογές τμημάτων.....	13
1.2.3 Λοιπές υπηρεσίες.....	13
1.3 Στόχος εργασίας.....	14
1.4 Οργάνωση εργασίας.....	14
Κεφάλαιο 2ο: Τεχνολογίες.....	15
2.1 Εισαγωγή.....	15
2.2 Angular.....	15
2.2.1 Components.....	16
2.2.2 Services.....	19
2.2.3 Modules.....	20
2.2.4 Directives.....	21
2.2.5 Routes.....	23
2.2.6 Guards.....	24
2.3 Firebase.....	25
2.3.1 Authentication.....	25
2.3.2 Cloud Firestore.....	26
2.3.3 Realtime Database.....	26
2.3.4 Cloud Storage.....	26
2.3.5 Cloud Functions.....	27
2.3.6 Hosting.....	30
2.4 FLUTTER.....	31

2.4.1 Χαρακτηριστικά Flutter.....	31
2.4.2 Πως λειτουργεί το Flutter.....	32
2.4.3 Πλεονεκτήματα.....	32
Κεφάλαιο 3ο: Ανάπτυξη εφαρμογών.....	34
3.1 Εισαγωγή.....	34
3.2 Σχεδιασμός και υλοποίηση της βάσης δεδομένων.....	34
3.3 Εφαρμογή διαχείρισης.....	35
3.3.1 Δυνατότητες διαχειριστή.....	35
3.3.2 Ανάπτυξη διαχειριστικής εφαρμογής ιστού.....	36
3.3.3 Αυθεντικοποίηση χρηστών.....	36
3.3.4 Υλοποίηση της κύριας γραφικής διεπαφής χρήστη.....	37
3.3.5 Αρχικοποίηση των συστατικών (Angular Components) και των δρομολογίων (Angular Routes).....	38
3.3.6 Υλοποίηση της καρτέλας "Ρυθμίσεις".....	39
3.3.7 Υλοποίηση της καρτέλας "Μεταφράσεις".....	40
3.3.8 Υλοποίηση της καρτέλας "Διαχειριστές".....	41
3.3.9 Υλοποίηση των καρτελών "Πανεπιστημιούπολεις, Στατικά Κείμενα, Πρεσβείες, Μουσεία, Αξιοθέατα".....	41
3.3.10 Υλοποίηση της λειτουργικότητας Διαδραστικών χαρτών.....	42
3.3.11 Υλοποίηση ψηφιακού βοηθού (Chatbot).....	44
3.4 Εφαρμογή κινητών συσκευών.....	45
3.4.1 Δυνατότητες χρήστη.....	45
3.4.2 Περιβάλλον ανάπτυξης.....	46
3.4.3 Ταυτοποίηση χρήστη.....	46
3.4.4 Υλοποίηση κύριας γραφικής διεπαφής χρήστη.....	47
3.4.5 Αρχική διεπαφή.....	48
3.5 Διεπαφή διαδραστικού χάρτη.....	49
3.6 Ανακοινώσεις.....	51
3.7 Chatbot.....	52
3.8 Ρυθμίσεις.....	55
3.9 Δομή βάσης δεδομένων.....	56
Κεφάλαιο 4ο: Εγχειρίδιο περιβάλλοντος διαχείρισης.....	57
4.1 Εισαγωγή.....	57
4.2 Σύνδεση στο σύστημα.....	57
4.2.1 Σύνδεση.....	57

4.2.2 Επαναφορά κωδικού πρόσβασης.....	57
4.3 Ανακάλυψη του Διαχειριστικού Περιβάλλοντος.....	58
4.3.1 Κεντρικός Πίνακας Διαχείρισης.....	58
4.4 Περιγραφή των καρτελών.....	61
4.4.1 Μεταφράσεις.....	61
4.4.2 Προβολή μεταφράσεων.....	62
4.4.3 Διαχειριστές.....	63
4.4.4 Στατικά κείμενα.....	64
4.4.5 Ειδοποιήσεις.....	65
4.4.6 Πανεπιστημιούπολεις.....	66
4.4.7 Τμήματα.....	68
4.4.8 Διαδραστικοί Χάρτες.....	69
4.4.9 Πρεσβείες.....	71
4.4.10 Μουσεία.....	73
4.4.11 Αξιοθέατα.....	74
4.4.12 Γενικοί πόροι.....	75
4.4.13 Chatbot.....	76
4.4.14 Settings.....	79
Κεφάλαιο 5ο: Συμπεράσματα ή/και προτάσεις βελτίωσης.....	82
5.1 Συμπεράσματα.....	82
5.2 Προτάσεις.....	82
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	84

Κατάλογος Εικόνων

Figure 1: Τα αρχεία ενός component.....	16
Figure 2: Κύκλος ζωής ενός component.....	18
Figure 3: Παράδειγμα ενός service αρχείου.....	20
Figure 4: Παράδειγμα ενός module αρχείου.....	21
Figure 5: Παράδειγμα ενός directive component.....	22
Figure 6: Παράδειγμα ενός structural directive.....	22
Figure 7: Παράδειγμα ενός attribute directive.....	23
Figure 8: Παράδειγμα ενός route αρχείου.....	24
Figure 9: Παράδειγμα ενός guard αρχείου.....	25
Figure 10: Διεπαφή αυθεντικοποίησης εφαρμογής ιστού.....	37
Figure 11: Συναρτήσεις σύνδεσης και αποσύνδεσης με χρήση της AngularFire.....	37
Figure 12: Δομή αρχείων συστήματος σύμφωνα με την λογική των components.....	38
Figure 13: Μενού προσθαφαίρεσης γλωσσών συστήματος.....	39
Figure 14: Μενού επεξεργασίας πόλεων.....	39
Figure 15: Μενού επεξεργασίας κατηγοριών πόρων/κειμένων.....	40
Figure 16: Διεπαφή μεταγλώττισης κειμένων.....	40
Figure 17: Γραφική διεπαφή επισκόπησης μουσείων.....	41
Figure 18: Οντότητες πανεπιστημιούπολεων.....	42
Figure 19: Σύνδεση πανεπιστημιούπολης με φωτογραφία-χάρτη.....	42
Figure 20: Επικάλυψη φωτογραφίας με στοιχείο καμβά.....	43
Figure 21: Καταγραφή σημείων πολυγώνου με χειριστή συμβάντων κλικ.....	43
Figure 22: Επεξεργασία πληροφοριών περιοχής πολυγώνου.....	43
Figure 23: Δομή εγγράφου δεδομένων ψηφιακού βοηθού στην γραφική διεπαφή της σουίτας Firebase.....	44
Figure 24: Αποτέλεσμα συνομιλίας με τον ψηφιακό βοηθό, με κατάληξη την προβολή μουσείων στην επιλεγμένη πόλη του χρήστη.....	45
Figure 25: Έγγραφο χρήστη της συλλογής Users στην γραφική διεπαφή της Firebase.....	46
Figure 26: Υποσέλιδο (bottom navigation bar) εφαρμογής τελικών χρηστών.....	47
Figure 27: Μέθοδος παραγωγής ρεύματος δεδομένων.....	48
Figure 28: Δομητής ρεύματος δεδομένων.....	49
Figure 29: Έγγραφο δεδομένων διαδραστικού χάρτη.....	49

Figure 30: Κώδικας μετατροπής απόλυτης θέσης αγγίγματος σε σχετικής με βάση τις διαστάσεις της εικόνας και έλεγχος έγκλεισης σημείου σε πολύγωνο.....	50
Figure 31: Γραφική διεπαφή κατόπιν επιλογής σημείου εντός ορισμένου πολυγώνου.....	51
Figure 32: Μέθοδος παραγωγής ρεύματος δεδομένων.....	51
Figure 33: Γραφική διεπαφή ανακοινώσεων.....	52
Figure 34: Στοιχείο ListView.builder.....	52
Figure 35: Αρχική κατάσταση διαλόγου.....	53
Figure 36: Εξέλιξη κατάστασης κατόπιν ενέργειας χρήστη.....	54
Figure 37: Λήξη διαλόγου και προβολή επιθυμητών αποτελεσμάτων.....	55
Figure 38: Συνδυασμός ρευμάτων για ορθή ενημέρωση των στοιχείων γραφικής διεπαφής επιλογής.....	56
Figure 39: Μέθοδος παραγωγής ρεύματος δεδομένων με χρήση ερωτήματος (query) στην βάση δεδομένων.....	56
Figure 40: Χρήση της κλάσης Provider, για δυναμική μετάφραση.....	56
Figure 41: Σύνδεση στο σύστημα.....	57
Figure 42: Επαναφορά κωδικού.....	58
Figure 43: Logout.....	59
Figure 44: Καρτέλες διαχειριστικού.....	60
Figure 45: Κουμπί προσθήκη σελίδας.....	61
Figure 46: Φόρμα νέου κλειδιού για μετάφραση.....	61
Figure 47: Προβολή όλων των κλειδιών μιας σελίδας.....	62
Figure 48: Επεξεργασία ενός κλειδιού.....	62
Figure 49: Πίνακας διαχειριστών.....	63
Figure 50: Κουμπί - Προσθήκη διαχειριστή.....	63
Figure 51: Προβολή στατικών κειμένων.....	64
Figure 52: Κουμπί - προσθήκη στατικού κειμένου.....	64
Figure 53: Φόρμα ειδοποίησης για την κινητή εφαρμογή.....	65
Figure 54: Λίστα ειδοποιήσεων.....	66
Figure 55: Προβολή πανεπιστημιούπολεων.....	67
Figure 56: Κουμπί - προσθήκη πανεπιστημιούπολης.....	67
Figure 57: Προσθήκη ονόματος και περιγραφής της πανεπιστημιούπολης σε όλες τις γλώσσες της εφαρμογής.....	68
Figure 58: Προβολή τμημάτων του ΔΙ.ΠΑ.Ε.....	68
Figure 59: Επιλογή πανεπιστημιούπολης της εφαρμογής.....	69
Figure 60: Προβολή διαδραστικών χαρτών.....	69

Figure 61: Φόρμα διαδραστικού χάρτη πριν την επιλογή φωτογραφίας.....	70
Figure 62: Φόρμα διαδραστικού χάρτη μετά την επιλογή φωτογραφίας.....	70
Figure 63: Συμπληρωμένη φόρμα διαδραστικού χάρτη μετά την επιλογή φωτογραφίας.....	71
Figure 64: Προβολή πρεσβειών.....	72
Figure 65: Προσθήκη στίγματος.....	73
Figure 66: Προβολή μουσείων.....	73
Figure 67: Προβολή αξιοθέατων.....	74
Figure 68: Προβολή γενικών πόρων.....	75
Figure 69: Έναρξη δενδροδιαλόγου.....	77
Figure 70: Προβολή μηνυμάτων του δενδροδιαλόγου.....	78
Figure 71: Προσθήκη τελικού κόμβου.....	79
Figure 72: Προσθήκη τελικού κόμβου με κατηγορία γενικοί πόροι.....	79
Figure 73: Προβολή του αποτελέσματος του τελικού κόμβου του δενδροδιαλόγου.....	79
Figure 74: Προσθήκη και διαγραφή γλωσσών στο σύστημα.....	80
Figure 75: Φόρμα πόλης.....	80
Figure 76: Φόρμα κατηγορίας στατικού κειμένου.....	81

Συντομογραφίες

Π.Ε.	Πτυχιακή Εργασία
ΔΙ.ΠΑ.Ε	Διεθνές Πανεπιστήμιο Ελλάδος
API	Application Programming Interface
VPN	Virtual Private Network
I.H.U./IHU	International Hellenic University
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
SASS	Syntactically Awesome Style Sheets
LESS	Leaner Style Sheets
DOM	Document Object Model
URL	Uniform Resource Locator
SDK	Software Development Kit
JSON	JavaScript Object Notation
CLI	Command Line Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
SSL	Secure Sockets Layer
CDN	Content Delivery Network
NPM	Node Package Manager
ARM	Advanced RISC Machines/Acorn RISC Machine
GPS	Global Positioning System
SVG	Scalable Vector Graphics
UI	User Interface
i18n	Internationalization and Localization

Κεφάλαιο 1ο: Εισαγωγή

1.1 Το πρόβλημα προσαρμογής και προσανατολισμού

Η αλλαγή περιβάλλοντος είναι από τα σημαντικότερα και επαχθέστερα επαναλαμβανόμενα γεγονότα στην ζωή κάθε ανθρώπου. Ειδικότερα, στην περίπτωση της μετάβασης νεαρών ανθρώπων από την ανώτερη στην ανώτατη εκπαίδευση περιλαμβάνει διάφορες προκλήσεις. Μεταξύ άλλων περιορισμών, η ανώτερη εκπαίδευση περιορίζει τους μαθητές οργανωτικά. Οι μαθητές δεν χρειάζεται παρά να περνούν το κατώφλι του κτηρίου καθημερινά και να ακολουθούν το πρόγραμμα το οποίο τους ανατίθεται. Σε πόλεις με μεγάλο πληθυσμό φοιτητών, όπως η Θεσσαλονίκη, μπορούμε να υποθέσουμε ότι ένα σημαντικό ποσοστό αυτών προέρχονται από άλλες πόλεις. Η απότομη αλλαγή από την έλλειψη αυτοδιοίκησης στην πλημμύρα επιλογών, που προσφέρει το πανεπιστήμιο, μπορεί να παραλύσει τους φοιτητές. Σε ένα σύντομο χρονικό διάστημα πρέπει να οικειοποιηθούν το κοινωνικό σύστημα της πόλης και τις εγκαταστάσεις της πανεπιστημιούπολης, να ενημερωθούν για τις ευθύνες τους ως φοιτητές του ΔΙ.ΠΑ.Ε., όπως και τις παροχές του πανεπιστημίου προς αυτούς κ.α. Επιπροσθέτως, πολλοί εξ' αυτών καλούνται να προσαρμοστούν και σε νέο πολιτικό περιβάλλον χωρίς φίλους και γνωριμίες. Έτσι, η αναζήτηση καθημερινών δραστηριοτήτων όπως εστίαση, αθλητικών προγραμμάτων ή ο τουρισμός της πόλης, προσθέτουν άγχος και βάρος στην διαδικασία προσαρμογής στο νέο περιβάλλον.

Εν ολίγοις, οι νέοι φοιτητές καλούνται να μεταβούν από ένα περιβάλλον εκπαίδευσης όπου οι πόροι που τους παρέχονται βρίσκονται ανά πάσα στιγμή στο οπτικό τους πεδίο, και η οργάνωση της εκπαίδευσης αναλαμβάνεται από δασκάλους και γονείς, σε ένα όπου οι ίδιοι είναι σχεδόν αποκλειστικά υπεύθυνοι για την αναζήτηση και οργάνωση αυτών.

1.2 Υπάρχουσες εφαρμογές ενημέρωσης και υποστήριξης

1.2.1 Ιστοσελίδες ΔΙ.ΠΑ.Ε

Το ΔΙ.ΠΑ.Ε διαθέτει ιστοσελίδες για το πανεπιστήμιο και για κάθε τμήμα του, οι οποίες περιέχουν βασικές πληροφορίες για την οργάνωση αυτών, γενικές ανακοινώσεις, πληροφορίες εκδηλώσεων, ανάλυση των προγραμμάτων σπουδών, τρόπους επικοινωνίας κ.α. Οι πληροφορίες που περιέχονται σε αυτές τις ιστοσελίδες παίζουν σημαντικό ρόλο στην διαδικασία ένταξης των φοιτητών στο ΔΙ.ΠΑ.Ε.

1.2.2 Ειδικές εφαρμογές τμημάτων

Τμήματα του ΔΙ.ΠΑ.Ε ενδέχεται να διαθέτουν δικές τους εφαρμογές διασύνδεσης. Για παράδειγμα, το τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων διαθέτει τις *Apps*(apps.iee.ihu.gr) και *Aboard*(aboard.iee.ihu.gr και *Aboard* στο *Google Play Store*). Με την χρήση αυτών, οι φοιτητές του τμήματος έχουν πρόσβαση στον εξειδικευμένο πίνακα ανακοινώσεων του τμήματος, πρόσβαση σε ειδικές υπηρεσίες που παρέχει το τμήμα σε κάθε φοιτητή (υπηρεσία VPN, προσωπική βάση δεδομένων κ.α.) κ.α.

1.2.3 Λοιπές υπηρεσίες

Εκτός από τις εφαρμογές του ΔΙ.ΠΑ.Ε., οι νεοεισερχόμενοι φοιτητές μπορούν να επωφεληθούν και από άλλες πληροφορίες άλλων φορέων, όπως τα δρομολόγια δημόσιων συγκοινωνιών, οδηγούς τουρισμού πόλεων (πχ. thessaloniki.gr/episkeptis/) κ.α.

1.3 Στόχος εργασίας

Στόχος της παρούσας πτυχιακής εργασίας είναι η ανάπτυξη εφαρμογής διαδικτύου και κινητών συσκευών η οποία ενοποιεί χρήσιμες πληροφορίες, μεταξύ άλλων, από πόρους που αναφέρονται στο κεφάλαιο 1.2, σε έναν εύχρηστο οδηγό για νέους φοιτητές του ΔΙ.ΠΑ.Ε. Έτσι, οι χρήστες θα έχουν ένα κεντρικό σημείο, όπου θα μπορούν, μεταξύ άλλων, να:

- Συστήνονται με τις βασικές αρχές του ΔΙ.ΠΑ.Ε.
- Ενημερώνονται για νέα του πανεπιστημίου, όπως και του εκάστοτε τμήματος φοίτησης, για θέματα στοχευμένα σε αυτούς.
- Ανακαλύπτουν πολιτιστικούς προορισμούς, δομές υποστήριξης και λοιπών υπηρεσιών που προσφέρει η πόλη στην οποία φοιτούν.
- Έχουν άμεση πρόσβαση σε πληροφορίες όπως στοιχεία επικοινωνίας για κάθε τμήμα, ωράρια λειτουργίας δομών, τοποθεσίες εστιών και λοιπών εγκαταστάσεων.

1.4 Οργάνωση εργασίας

Στο επόμενο κεφάλαιο θα αναφέρουμε όλες τις τεχνολογίες και εφαρμογές που απαιτήθηκαν για την σχεδίαση και ανάπτυξη των εφαρμογών. Αυτές συμπεριλαμβάνουν γλώσσες προγραμματισμού, επεξεργαστή κειμένων, μεθοδολογίες προγραμματισμού, frameworks κλπ.

Στο τρίτο κεφάλαιο θα αναλύσουμε τον τρόπο ανάπτυξης των εφαρμογών. Θα αναφερθούμε στην αρχιτεκτονική της εφαρμογής, τον σχεδιασμό της δομής βάσης δεδομένων, καθώς και την διαδικασία ανάπτυξης και αξιοποίησης εργαλείων.

Το τέταρτο κεφάλαιο αποτελεί ένα εγχειρίδιο χρήσης με κάποια πρακτικά παραδείγματα.

Τέλος, στο πέμπτο κεφάλαιο αναφέρουμε τα συμπεράσματα για την διαδικασία ανάπτυξης της εφαρμογής, τα αποτελέσματα της εργασίας, και προτάσεις για βελτίωση ή μελλοντική επέκταση του αποτελέσματος.

Κεφάλαιο 2ο: Τεχνολογίες

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει αναφορά και ανάλυση των τεχνολογιών που χρησιμοποιήθηκαν για την ανάπτυξη των εφαρμογών. Η εφαρμογή αποτελείται από δύο μέρη: μια εφαρμογή για κινητές συσκευές που έχει αναπτυχθεί με τη χρήση του πλαισίου (framework) Flutter και ένα διαχειριστικό περιβάλλον ιστού που έχει αναπτυχθεί με τη χρήση του πλαισίου Angular. Το υποστηρικτικό σύστημα (backend) υλοποιήθηκε με τη βοήθεια της σουίτας Firebase.

Η εφαρμογή για κινητές συσκευές διαθέτει μια φιλική προς τον χρήστη διεπαφή χρήστη, η οποία παρέχει πρόσβαση σε σημαντικές λειτουργίες της εφαρμογής. Το διαχειριστικό περιβάλλον στο παγκόσμιο ιστό (web) παρέχει στους διαχειριστές της εφαρμογής τη δυνατότητα να διαχειριστούν τους χρήστες και τα δεδομένα της εφαρμογής.

Η σουίτα Firebase παρέχει έναν εύκολο και αξιόπιστο τρόπο για την υλοποίηση του υποστηρικτικού συστήματος της εφαρμογής. Η χρήση της σουίτας Firebase επιτρέπει στους αναπτυσσόμενους να συγκεντρωθούν στην υλοποίηση της εφαρμογής και όχι με άλλες εκκρεμότητες όπως είναι η φιλοξενία ιστοσελίδων (hosting). Επιπλέον, παρέχει πολλές δυνατότητες όπως εξοικονόμηση χρόνου στην ανάπτυξη, αυτόματη επικύρωση των δεδομένων, αντιγραφή ασφαλείας και πολλά άλλα.

Συνολικά, η εφαρμογή αυτή παρέχει μια σύγχρονη και πλήρη εμπειρία χρήστη για τους χρήστες της, ενώ παράλληλα παρέχει στους διαχειριστές της τη δυνατότητα να διαχειρίζονται την εφαρμογή με αποτελεσματικό τρόπο. Η χρήση των πλαισίων ανάπτυξης λογισμικού Flutter και Angular στην ανάπτυξη των διαφόρων μερών της εφαρμογής εξασφαλίζει την εύκολη συντήρηση της εφαρμογής και τη δυνατότητα επέκτασης της σε μελλοντικές ανάγκες.

2.2 Angular

Το πλαίσιο Angular είναι ένα πλαίσιο σύνολο προγραμματιστικών εργαλείων και βιβλιοθηκών στην γλώσσα Javascript, ανοιχτού κώδικα γραμμένο στην γλώσσα TypeScript, ένα υπερσύνολο της Javascript. Το πλαίσιο αυτό δημιουργήθηκε από την Google και αναπτύσσεται συνεχώς από μια μεγάλη κοινότητα προγραμματιστών. Ως πλαίσιο, το Angular έχει σαφή πλεονεκτήματα, ενώ παρέχει επίσης μια τυπική δομή για να συνεργαστούν οι προγραμματιστές. Επιτρέπει στους χρήστες να δημιουργούν μεγάλες εφαρμογές με τρόπο συντηρήσιμο [1].

Στο πλαίσιο Angular, τα τμήματα του κώδικα μπορούν να οργανωθούν και να καταμεριστούν σε κλάσεις οι οποίες βαφτίζονται ανάλογα με την λειτουργία που επιτελούν ως εξής:

1. Modules,
2. Services,
3. Components,
4. Directives.

Τα modules στο πλαίσιο Angular ομαδοποιούν συναφή λειτουργικότητα μαζί για την οργάνωση του κώδικα και την εύκολη επαναχρησιμοποίηση της. Πιο συγκεκριμένα μπορούν να περιέχουν components, services και άλλα modules. Τα services παρέχουν έναν τρόπο να μοιράζονται δεδομένα και λειτουργικότητα μεταξύ components και άλλων σημείων της εφαρμογής. Τα components είναι τα

βασικά στοιχεία του πλαισίου Angular, αντιπροσωπεύοντας τμήματα της διεπαφής χρήστη, και τα directives χρησιμοποιούνται για να προσθέτουν συμπεριφορά στην διεπαφή χρήστη [2]. Πιο αναλυτικά:

2.2.1 Components

Ένα component της Angular χωρίζεται, σε 4 κομμάτια :

1. Σε ένα αρχείο HTML, όπου βρίσκονται τα στοιχεία (elements) που το απαρτίζουν.
2. Σε ένα αρχείο CSS, όπου ορίζεται με σαφήνεια και ιδιαίτερη ευελιξία ο τρόπος με τον οποίο εμφανίζονται διάφορα στοιχεία σε μία ιστοσελίδα.
3. Σε ένα αρχείο Typescript, στο οποίο βρίσκεται η λογική.
4. Σε ένα ακόμα Typescript αρχείο για δοκιμαστική διαδικασία (testing).

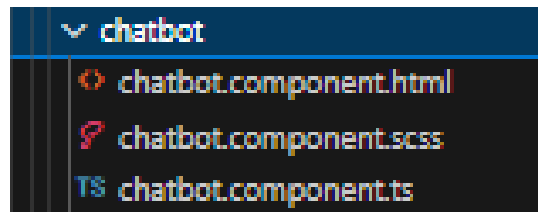


Figure 1: Τα αρχεία ενός component

2.2.1.1 HTML

Το HTML σημαίνει Γλώσσα Χαρακτηρισμού Υπερ-Κειμένου (Hypertext Markup Language), και τα αρχεία HTML υλοποιούνται κατά κύριο λόγο ως στατικές σελίδες των δικτυακών τόπων. Ουσιαστικά, δεν είναι γλώσσα προγραμματισμού, αλλά γλώσσα περιγραφής ιδιοτήτων των στοιχείων που αποτελούν μία ιστοσελίδα. Χρησιμοποιώντας ειδικές ετικέτες (tags) δίνονται εντολές, για το πως να διαχειριστεί ένα έγγραφο HTML το περιεχόμενό του και πως να το εμφανίσει στον χρήστη ένας φυλλομετρητής (web browser), το πρόγραμμα δηλαδή που χρησιμοποιήθηκε για να προβληθούν οι διάφορες σελίδες στο διαδίκτυο [3].

2.2.1.2 CSS

Το CSS είναι ένα σύνολο εντολών που γράφονται για να μορφοποιηθούν “αντικείμενα” μέσα σε μία ιστοσελίδα. Με τις CSS εντολές μπορεί να δοθεί χρώμα, να γίνει bold, να προστεθεί περίγραμμα, να υπογραμμιστεί και να εφαρμοστούν πολλές άλλες παρόμοιες ενέργειες σε διάφορα στοιχεία μιας ιστοσελίδας [4],[5].

Το πλαίσιο Angular παρέχει επίσης υποστήριξη για άλλες γλώσσες παρόμοιες με το css:

1. Το Sass, το οποίο μεταγλωττίζεται σε CSS και αποθηκεύεται με την επέκταση .sass. Το Sass αποτελείται από δύο συντάξεις, το πρωτότυπο που βασίζεται σε εσοχές που χρησιμοποιεί την επέκταση .sass και το νεότερο SCSS με μορφοποίηση μπλοκ όπως το CSS που χρησιμοποιεί την επέκταση .scss [6].
2. Το SCSS είναι η δεύτερη σύνταξη του Sass που χρησιμοποιεί αγκύλες αντί για εσοχές. Το SCSS σχεδιάστηκε με τέτοιο τρόπο ώστε ένα έγκυρο αρχείο CSS3 να είναι επίσης έγκυρο αρχείο SCSS. Τα αρχεία SCSS αποθηκεύονται με την επέκταση .scss.[7]

3. Το Less το οποίο επιτρέπει στο Angular CLI να χρησιμοποιηθεί μέσα στα πρότυπα συνιστωσών του πλαισίου Angular. Τα αρχεία Less αποθηκεύονται με την επέκταση .less. Το πλεονέκτημα του Less από το Sass είναι στο ότι όλος ο έγκυρος κώδικας CSS είναι επίσης έγκυρος κώδικας Less. Αυτό καθιστά την εκμάθηση του Less πολύ εύκολη αν γνωρίζετε ήδη CSS [6].

2.2.1.3 Typescript

Η TypeScript είναι μια γλώσσα προγραμματισμού που αναπτύχθηκε και συντηρείται από τη Microsoft. Πρόκειται για μία επέκταση της Javascript [9]. Διαφέρει από τους ανταγωνιστές της, όπως CoffeScript και Dart στο ότι απλός κώδικας JavaScript μπορεί να χρησιμοποιηθεί ταυτόχρονα με την TypeScript. Ως εκ τούτου, η Javascript είναι Typescript. Εν αντιθέσει, η Typescript πρέπει να μεταγλωττιστεί σε Javascript ώστε να μπορέσει να τρέξει σε κάποια Javascript μηχανή, όπως το πρόγραμμα περιήγησης (browser) ή ένας node.js διακομιστής (server). Αυτό σημαίνει ότι δεν μπορούμε να ενσωματώσουμε Typescript απευθείας σε μια web σελίδα μέσω της ετικέτας 'script', αλλά τα αρχεία της Typescript(.ts) θα πρέπει να μεταγλωττιστούν σε Javascript(.js) για χρήση. Ως εκ τούτου, η Typescript δεν είναι Javascript [8].

2.2.1.3.1 LifeCycle hooks

Στο πλαίσιο Angular, κάθε component έχει έναν κύκλο ζωής, μια σειρά από διαφορετικά στάδια που περνά από την προετοιμασία έως την καταστροφή. Υπάρχουν 8 διαφορετικά στάδια στον κύκλο ζωής ενός component. Κάθε στάδιο ονομάζεται γεγονός κύκλου ζωής (life cycle hook event) και χρησιμοποιείται σε διαφορετικές φάσεις μιας εφαρμογής για να αποκτηθούν κάποιοι έξτρα έλεγχοι στα component. Δεδομένου ότι ένα component είναι μια Typescript κλάση, για αυτό το λόγο κάθε component πρέπει να έχει ένας δομητής (constructor). Πριν από την εκτέλεση οποιουδήποτε άλλου κύκλου ζωής εκτελείται ο δομητής. Εάν χρειάζεται να εισάγουμε εξαρτήσεις (dependencies), στον δομητή είναι το ιδανικό μέρος για να γίνει αυτή η εισαγωγή. Μετά την εκτέλεση του δομητή, εκτελούνται και οι υπόλοιποι κύκλοι ζωής μέθοδοι με μία συγκεκριμένη σειρά [17,21]:

1. ngOnChanges
2. ngOnInit
3. ngDoCheck
4. ngAfterContentInit
5. ngAfterContentChecked
6. ngAfterViewInit
7. ngAfterViewChecked
8. ngOnDestroy

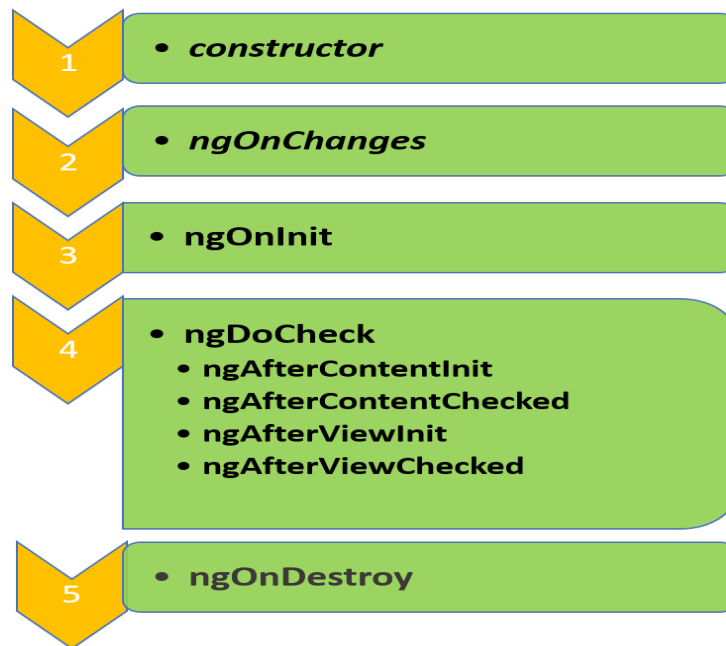


Figure 2: Κύκλος ζωής ενός component

2.2.1.3.2 ngOnChanges

Η ngOnChanges μέθοδος είναι ένας κύκλος ζωής, ο οποίος ενεργοποιείται κάθε φορά όταν ανιχνεύει αλλαγές στην ιδιότητα εισαγωγής δεσμευμένων δεδομένων. Αυτή η μέθοδος λαμβάνει ένα αντικείμενο SimpleChanges, το οποίο περιέχει τις τρέχουσες και τις προηγούμενες τιμές ιδιοτήτων [18].

2.2.1.3.3 ngOnInit

Η ngOnInit μέθοδος χρησιμοποιήσετε για να εκτελεστεί ένα κομμάτι κώδικα κατά την δημιουργία του component. Η ngOnInit ενεργοποιείται μόνο μία φορά μετά τον ορισμό των ιδιοτήτων δεσμευμένων δεδομένων. Αυτό σημαίνει ότι η ngOnInit θα εκτελεστεί εάν ανανεωθεί το πρόγραμμα περιήγησης ή αρχικοποιηθεί πρώτα ένα component αλλά όχι όταν προκύψουν άλλα συμβάντα [19].

2.2.1.3.4 ngDoCheck

Η ngDoCheck μέθοδος ενεργοποιείται κάθε φορά που ελέγχονται οι ιδιότητες ενός στοιχείου. Η διαφορά της με την ngOnChanges είναι ότι η DoCheck τρέχει κάθε φορά που ανιχνεύει οποιαδήποτε αλλαγή σε οποιοδήποτε component, ενώ η onChanges κάνει και αυτή έλεγχο, μόνο στις δικές της ιδιότητες εισόδου και τρέχει μόνο αν εντοπίσει εκεί πέρα κάποια αλλαγή [20].

2.2.1.3.5 ngAfterContentInit

Η μέθοδος ngAfterContentInit στο πλαίσιο Angular χρησιμοποιείται για να επιτελέσουν λειτουργίες αμέσως μετά την αρχική εισαγωγή περιεχομένου στο component. Η μέθοδος αυτή καλείται αφού το πλαίσιο Angular ανακαλύψει τα περιεχόμενα που εισήχθησαν στο component μέσω της διαδικασίας ανίχνευσης αλλαγών και τα εγκαθιστά στον αντίστοιχο πίνακα αντικειμένων περιεχομένου. Είναι χρήσιμη για λειτουργίες που απαιτούν την πρόσβαση σε περιεχόμενο που εισήχθη στο component,

όπως η επικοινωνία με άλλα component ή η αρχικοποίηση μεταβλητών που εξαρτώνται από το περιεχόμενο. Καλείται μόνο μία φορά μετά την ngDoCheck μέθοδο [17,21].

2.2.1.3.6 ngAfterContentChecked

Η μέθοδος ngAfterContentChecked στο πλαίσιο Angular χρησιμοποιείται για να εκτελέσει λειτουργίες αμέσως μετά τον έλεγχο των αλλαγών που έχουν γίνει στο περιεχόμενο που εισάγεται στο component. Η μέθοδος αυτή καλείται μετά από κάθε έλεγχο των αλλαγών στο component που εισήχθη στο στοιχείο μέσω της διαδικασίας ανίχνευσης αλλαγών. Χρησιμοποιείται συνήθως σε λειτουργίες που εξαρτώνται από το component που εισήχθη, όπως η ενημέρωση του περιεχομένου του component με βάση τις αλλαγές σε μεταβλητές που εξαρτώνται από το περιεχόμενο. Είναι σημαντικό να σημειωθεί ότι η μέθοδος αυτή καλείται πολλές φορές κατά τη διάρκεια του κύκλου ζωής ενός component, όποτε θα πρέπει να προσέχουμε ώστε οι λειτουργίες που εκτελούνται να είναι ελαφριές και αποδοτικές [17,21].

2.2.1.3.7 ngAfterViewInit

Η μέθοδος ngAfterViewInit χρησιμοποιείται για να εκτελέσει λειτουργίες αμέσως μετά την αρχικοποίηση της διεπαφής χρήστη (view) που συνδέεται με το component, αλλά και των child views. Είναι σημαντικό να σημειωθεί ότι η μέθοδος αυτή καλείται μόνο μία φορά κατά τη διάρκεια του κύκλου ζωής του component, οπότε μπορούμε να εκτελέσουμε πιο απαιτητικές λειτουργίες σε αυτήν τη μέθοδο [17,21].

2.2.1.3.8 ngAfterViewChecked

Η μέθοδος ngAfterViewChecked στο πλαίσιο Angular καλείται κάθε φορά που ο αλγόριθμος ανίχνευσης αλλαγών του πλαισίου Angular ελέγχει τη θέση του component και επιβεβαιώνει ότι όλα τα δεδομένα στη διεπαφή χρήστη (view) είναι ενημερωμένα. Αυτή η μέθοδος καλείται μετά την ngAfterViewInit. Είναι χρήσιμη για διάφορους σκοπούς όπως η εκτέλεση επιπλέον λειτουργιών πάνω στην παρουσίαση δεδομένων, ο έλεγχος αν το στοιχείο έχει ανανεωθεί σωστά και η ενημέρωση της διεπαφής χρήστη του component. Είναι σημαντικό να σημειωθεί ότι αυτή η μέθοδος εκτελείται πολλές φορές κατά τη διάρκεια του κύκλου ζωής του component και μπορεί να οδηγήσει σε απόδοση χαμηλότερης ποιότητας εάν δεν χρησιμοποιηθεί σωστά [17,21].

2.2.1.3.9 ngOnDestroy

Η μέθοδος ngOnDestroy καλείται αμέσως πριν από την καταστροφή ενός component στο πλαίσιο Angular. Είναι πολύ χρήσιμη για την αποσύνδεση των observables (μηχανισμό ασύγχρονης παραλαβής δεδομένων) και τους χειριστές συμβάντων (event handlers) προκειμένου να αποφευχθούν διαρροές μνήμης και άλλα προβλήματα. Καλείται αμέσως πριν από την καταστροφή του συγκεκριμένου στιγμιότυπου (instance) του component. Αυτή η μέθοδος καλείται μόνο μια φορά, αμέσως πριν από την αφαίρεση του component από το DOM και είναι καλό να χρησιμοποιηθεί για να αποδεσμεύσουμε τους πόρους που χρησιμοποιήθηκαν κατά τη διάρκεια του κύκλου ζωής του [17,21].

2.2.2 Services

Στο πλαίσιο Angular, ένα service είναι μία Typescript κλάση που παρέχει λειτουργικότητα σε πολλαπλά components. Είναι ένας τρόπος κοινής χρήσης δεδομένων και λειτουργικότητας μεταξύ component και χρησιμοποιείται συχνά για τη συγκέντρωση κοινής λειτουργικότητας ή λογικής που απαιτείται σε πολλά μέρη μίας εφαρμογής.

Ένα service στο πλαίσιο Angular έχει συνήθως μια συγκεκριμένη ευθύνη, όπως η διαχείριση της ανάκτησης δεδομένων ή εκτέλεση υπολογισμών και μπορεί να πραγματοποιήσει ενσωμάτωση (Inject) σε components τα οποία χρειάζονται πρόσβαση στη λειτουργικότητα της. Αυτό έχει ως αποτέλεσμα να γίνει η εφαρμογή πιο ευανάγνωστη και ευκολότερη στη συντήρηση.

Τα Angular services μπορούν να δημιουργηθούν χρησιμοποιώντας τον διακοσμητή (decorator) '@Injectable()', το οποίο τους επιτρέπει να **πραγματοποιήσουν ενσωμάτωση (Inject)** σε άλλα component ή services.

Σε ένα service επίσης μπορούν να πραγματοποιηθούν κλήσεις API, ή να γίνει η αυθεντικοποίηση των χρηστών [11][12].

```
@Injectable({
  providedIn: 'root' // You, 9 months ago * auth service
})
export class AuthService {

  currentFirstName$: Observable<Admin>;
  authState$ = authState(this.auth);
  uid$ = this.authState$.pipe(map(user => {
    const uid = user?.uid;
    if (!uid) {
      return '';
    }
    return uid;
  }));

  constructor(
    private auth: Auth,
    private adminsService: AdminsService,
    private snackBar: SnackBarService,
  ) {}

  isAuthenticated(): Observable<boolean> {
    return this.authState$.pipe(map(user => !!user), shareReplay(1));
  }
}
```

Figure 3: Παράδειγμα ενός service αρχείου

2.2.3 Modules

Τα modules αποτελούν έναν τρόπο δομής του κώδικα που επιτρέπει στους προγραμματιστές να οργανώνουν τη συναφή λειτουργικότητα μαζί, δημιουργώντας ένα καθαρό και οργανωμένο περιβάλλον ανάπτυξης λογισμικού. Με τη βοήθεια των modules, γίνεται να διαχωριστεί ο κώδικας σε μικρότερα, πιο ευανάγνωστα και επαναχρησιμοποιήσιμα κομμάτια, τα οποία μπορούν να κληθούν από άλλα μέρη του προγράμματος.

Στο πλαίσιο Angular, ένα module είναι ένας μηχανισμός για την ομαδοποίηση των components, directives, pipes, services που είναι σχετικά μεταξύ τους, με τέτοιο τρόπο ώστε να μπορούν να συνδυαστούν με άλλα modules για να δημιουργηθεί μια εφαρμογή Angular. Μια εφαρμογή Angular μπορεί να θεωρηθεί ως ένα παζλ όπου κάθε κομμάτι (ή κάθε module) απαιτείται για να μπορέσουμε να δούμε την πλήρη εικόνα.

Ένα άλλο παράδειγμα για να κατανοήσουμε τα Angular modules είναι οι κλάσεις. Σε μια κλάση, μπορούμε να ορίσουμε public ή private μεθόδους. Οι public μέθοδοι είναι τα API που άλλα τμήματα του κώδικα μας μπορούν να χρησιμοποιούν για να αλληλεπιδράσουν με την κλάση, ενώ οι private μέθοδοι είναι λεπτομέρειες υλοποίησης που είναι κρυμμένες. Με τον ίδιο τρόπο, ένα module μπορεί να εξάγει ή να κρύβει components, directives, pipes και services. Τα εξαγόμενα στοιχεία προορίζονται

για χρήση από άλλα modules, ενώ αυτά που δεν εξάγονται (κρυφά) χρησιμοποιούνται μόνο μέσα στο module και δεν μπορούν να αποκτηθούν πρόσβαση από άλλα modules της εφαρμογής μας [13].

```

const components = [
  ContainerComponent,
  HeaderComponent,
  FormFieldComponent,
  InputSearchComponent,
  SelectBoxComponent,
  ItemCardComponent,
  CustomMapComponent,
];

@NgModule({
  declarations: [
    ...components,
  ],
  imports: [
    CommonModule,
    FormsModule,
    MaterialModule,
    SvgIconsModule,
    GoogleMapsModule,
    CKEditorModule,
    DirectivesModule,
    PipesModule,
  ],
  exports: [
    MaterialModule,
    SvgIconsModule,
    GoogleMapsModule,
    CKEditorModule,
    DirectivesModule,
    PipesModule,
    ...components,
  ],
})
export class SharedModule {}

```

Figure 4: Παράδειγμα ενός module αρχείου

2.2.4 Directives

Τα Directives είναι κλάσεις που προσθέτουν νέες συμπεριφορές ή τροποποιούν τις υπάρχουσες συμπεριφορές στα στοιχεία του προτύπου. Γενικά, τα Directives χρησιμοποιούνται για την διαχείριση του DOM, για παράδειγμα προσθέτοντας ή αφαιρώντας στοιχεία από το DOM ή αλλάζοντας την εμφάνιση των στοιχείων του DOM [14].

Υπάρχουν 3 είδη directive στην Angular:

- Τα component directives τα οποία χρησιμοποιούνται για να καθορίσουν τα HTML templates. Έχουν το σχεδιασμό δομής και το πρότυπο λειτουργίας για το πως πρέπει το component να επεξεργάζεται, να δημιουργείται και να χρησιμοποιείται κατά την διάρκεια της εκτέλεσης του.

```

You, 7 months ago | 1 author (You)
@Directive({
  selector: '[appTypeOfNotification]'
})
export class TypeOfNotificationDirective implements OnChanges {

  @Input() target?: TypeOfNotification;
  constructor(
    private el: ElementRef,
  ) {}

  ngOnChanges() {
    switch (this.target) {
      case ('notification'): {
        this.write('Είδος ειδοποίησης: Notification');
        break;
      }
      case ('email'): {
        this.write('Είδος ειδοποίησης: Email');
        break;
      }
      case ('both'): {
        this.write('Είδος ειδοποίησης: Email και Notification');
        break;
      }
      default: {
        this.write('');
      }
    }
  }

  write(text: string) {
    this.el.nativeElement.innerHTML = text;
  }
}

```

Figure 5: Παράδειγμα ενός directive component

- Τα structural directive αλλάζουν το σχεδιασμό και τη δομή του DOM. Ο βασικός σκοπός των structural directive είναι η διαχείριση και η πραγματοποίηση αλλαγών στα στοιχεία του DOM κατά τη διάρκεια της εκτέλεσης. Μπορείτε εύκολα να την αναγνωρίσετε από το σύμβολο αστερίσκου (*). Μερικά παραδείγματα structural directives είναι τα εξής:
 1. *ngIf: Χρησιμοποιείται για να εμφανίσει ή να αποκρύψει μια ετικέτα HTML βάσει μιας συνθήκης.
 2. *ngFor: Χρησιμοποιείται για να επαναλάβει μια ετικέτα HTML για κάθε αντικείμενο σε μια λίστα.
 3. *ngSwitch: Χρησιμοποιείται για να εμφανίσει διαφορετικά τμήματα HTML βάσει μιας συνθήκης.

```

<ng-container *ngIf="formFieldType==='text'">
  <input [readOnly]="readonly" [(ngModel)]="value" class="custom-inputs" type="text" (input)="changeInput()">
</ng-container>
<ng-container *ngIf="formFieldType==='textarea'">
  <textarea [(ngModel)]="value" [rows]="rows" class="custom-inputs" type="text"
  (input)="changeInput()"></textarea>
</ng-container>

```

Figure 6: Παράδειγμα ενός structural directive

- Τα Attribute directives είναι μια σειρά απο ενσωματωμένα directive που μπορούμε να προσθέσουμε στα στοιχεία HTML μας για να τους δώσουμε δυναμική συμπεριφορά. Συνολικά, ένα attribute directive αλλάζει την εμφάνιση ή συμπεριφορά ενός στοιχείου του DOM. Μερικά παραδείγματα attribute directives είναι τα εξής:
 1. `ngClass`: Χρησιμοποιείται για να προσθέσει ή να αφαιρέσει κλάσεις CSS από μια ετικέτα HTML βάσει μιας συνθήκης.
 2. `ngStyle`: Χρησιμοποιείται για να ορίσει CSS σε μια ετικέτα HTML βάσει μιας συνθήκης [15,16].

```

<div *ngFor="let chatText of chatBreadcrumbs; index as i;"
  [ngClass]="i%2===0 ? 'chatBotPrompt' : 'userOptionSelection">
  <p class="self-center">{!!chatText[activeLanguage.code] ?
  chatText[activeLanguage.code] : "No translation available"}</p>
</div>

```

Figure 7: Παράδειγμα ενός attribute directive

2.2.5 Routes

Η διαχείριση σελίδων της εφαρμογής γίνεται μέσω του συστήματος δρομολόγησης (Router). Η δρομολόγηση χρησιμοποιείται για την πλοήγηση σε διαφορετικές σελίδες μιας εφαρμογής, αλλά και για να επιτευχθεί η συμπεριφορά εφαρμογής μιας σελίδας (Single Page Application), τεχνική που επιτρέπει την πλοήγηση σε διαφορετικές σελίδες χωρίς πλήρη επαναφόρτωση της εφαρμογής. Το routing δρομολογεί την εφαρμογή σε διαφορετικές σελίδες μέσω της αλλαγής του URL χωρίς να φορτώνει ξανά ολόκληρη την εφαρμογή.

Η διαχείριση της δρομολόγησης στο πλαίσιο Angular γίνεται μέσω της χρήσης του RouterModule, το οποίο μπορεί να παρέχει μια δομή για τη δρομολόγηση σε σελίδες και υπηρεσίες, ενώ το RouterOutlet είναι ένας διανομέας (placeholder) για το περιεχόμενο που θα φορτωθεί κατά τη δρομολόγηση.

Με τη δρομολόγηση στο πλαίσιο Angular, μπορείτε να καθορίσετε ποια σελίδα θα εμφανίζεται στο πρόγραμμα περιήγησης όταν ο χρήστης κάνει κλικ σε ένα σύνδεσμο ή εισάγει ένα συγκεκριμένο URL. Επίσης, μπορούν να οριστούν δυναμικά μερικά μέρη του URL ως παραμέτρους και να χρησιμοποιηθούν για να προβληθούν διαφορετικά δεδομένα στη σελίδα. Οι διαδρομές αυτές μπορούν να προστατευτούν μέσω Guards [22].

```

const routes: Routes = [
  {
    path: '',
    pathMatch: 'full',
    redirectTo: 'login',
  },
  {
    path: 'login',
    component: LoginComponent,
  },
  {
    path: 'main',
    component: MainNavComponent,
    canActivate: [AuthGuard],
    children: [
      {
        path: '',
        pathMatch: 'full',
        redirectTo: 'translations',
      },
      {
        path: 'translations',
        loadChildren: () => import('./views/main-nav/pages/translations/translations.component').then(m => m.TranslationsComponent),
      },
      {
        path: 'staticTexts',
        loadChildren: () => import('./views/main-nav/pages/static-texts/static-texts.module').then(m => m.StaticTextsModule),
      },
      {
        path: 'admins',
        loadChildren: () => import('./views/main-nav/pages/admins/admins.component').then(m => m.AdminsComponent),
      },
    ],
  },
];

```

Figure 8: Παράδειγμα ενός route αρχείου.

2.2.6 Guards

Τα guards είναι μια λειτουργία στο πλαίσιο Angular που χρησιμοποιείται για την προστασία διαδρομών (routes) από μη εξουσιοδοτημένους χρήστες. Μπορούν να χρησιμοποιηθούν για να επιτρέπουν ή να αποτρέπουν πρόσβαση σε μία συγκεκριμένη διαδρομή ανάλογα με τον ρόλο του χρήστη ή άλλες παραμέτρους. Στο πλαίσιο Angular υπάρχουν τέσσερα είδη guards:

1. CanActivate: Ελέγχει αν ο χρήστης έχει πρόσβαση στη διαδρομή.
2. CanActivateChild: Ελέγχει αν ο χρήστης έχει πρόσβαση στα παιδικά μονοπάτια της διαδρομής.
3. CanDeactivate: Ελέγχει αν ο χρήστης μπορεί να φύγει από τη διαδρομή.
4. Resolve: Φορτώνει δεδομένα πριν από την φόρτωση της διαδρομής [23].

```

import { inject } from '@angular/core';
import { ActivatedRouteSnapshot, CanActivateFn, Router, RouterStateSnapshot, UrlTree } from '@angular/router';
import { map, Observable, of, switchMap } from 'rxjs';
import { AuthService } from '../services/auth.service';

export const AuthGuard = (
  route: ActivatedRouteSnapshot,
  state: RouterStateSnapshot
): Observable<boolean | UrlTree> | Promise<boolean | UrlTree> | boolean | UrlTree => {

  const router: Router = inject(Router);
  const authService: AuthService = inject(AuthService);

  return authService.isAuthenticated().pipe(switchMap(isAuthenticated => {
    // check if user is authenticated
    if (!isAuthenticated) {
      return of(router.parseUrl(''));
    }

    // check if user exists in firestore
    return authService.getCurrentAdmin().pipe(
      map(admin => {
        if (admin) {
          return true;
        } else {
          return router.parseUrl('');
        }
      })
    ),
  ));
});

const canActivate: CanActivateFn = AuthGuard;

```

Figure 9: Παράδειγμα ενός guard αρχείου

2.3 Firebase

Η Firebase αποτελείται από μια σειρά υπηρεσιών που, συνδυαζόμενες, δίνουν τη δυνατότητα ανάπτυξης εφαρμογών υψηλής ποιότητας σε σχέση με τον απαιτούμενο χρόνο. Ωντας χωρίς διακομιστή (serverless) μπορεί να κρατήσει το κόστος της εφαρμογής χαμηλό και να κάνει κλιμακωθεί αυτόματα, ανάλογα με τους χρήστες.

2.3.1 Authentication

Η ταυτοποίηση Firebase (Firebase Authentication) είναι μία υπηρεσία που επιτρέπει τον εύκολο έλεγχο στην σύνδεση και την αναγνώριση των χρηστών σε μια εφαρμογή. Αυτό το προϊόν είναι απαραίτητο για τη σωστή διαμόρφωση ορισμένων από τα άλλα προϊόντα ειδικά εάν πρέπει να περιοριστεί η πρόσβαση στα δεδομένα ανά χρήστη. Η Firebase δίνει την δυνατότητα να ταυτοποιηθεί ένας χρήστης με διάφορες μεθόδους, όπως το ηλεκτρονικό ταχυδρομείο/κωδικός πρόσβασης, αριθμός τηλεφώνου, Facebook, Google, Twitter, GitHub κ.α.

Η ταυτοποίηση Firebase παρέχει ένα σετ ανάπτυξης λογισμικού (SDK) από την πλευρά του πελάτη (client-side SDK) που δίνει την δυνατότητα να ενσωματωθεί εύκολα η πιστοποίηση σε μία εφαρμογή. Επίσης παρέχει έναν SDK από την πλευρά του διακομιστή (server-side SDK) που δίνει την δυνατότητα να διαχειρίζονται οι χρήστες την κατάσταση πιστοποίησης τους, καθώς επίσης και να επιβληθούν κανόνες ασφαλείας για την πρόσβαση στις υπηρεσίες της Firebase.

Η αυθεντικοποίηση μπορεί να χρησιμοποιηθεί επίσης για την διαχείριση από λογαριασμούς χρηστών και την κατάσταση πιστοποίησης σε πολλές πλατφόρμες, συμπεριλαμβανομένων των iOS, Android και web. Με αυτόν τον τρόπο δημιουργούνται εύκολα εφαρμογές πολλαπλών πλατφορμών (cross-platform apps) με συνεπείς μηχανισμούς πιστοποίησης και εξουσιοδότησης χρηστών.

Συνολικά, η αυθεντικοποίηση της Firebase είναι μια ισχυρή και ευέλικτη υπηρεσία που επιτρέπει στους προγραμματιστές να προσθέτουν γρήγορα και εύκολα την πιστοποίηση χρηστών στις εφαρμογές τους, χωρίς να ανησυχούν για τις υποκείμενες πολυπλοκότητες της ταυτοποίησης και της ασφάλειας [24,25].

2.3.2 Cloud Firestore

Η Cloud Firestore είναι η υπηρεσία βάσης δεδομένων της Firebase, μαζί με τις απαραίτητες λειτουργίες εγγραφής και ανάγνωσης στη βάση από τους χρήστες της εφαρμογής. Η δομή της βάσης είναι NoSQL, δηλαδή ξεφεύγει από την κλασική οργάνωση σε πίνακες. Αυτή η δομή καθιστά τη βάση σε πολλές περιπτώσεις πιο εύχρηστη, ευέλικτη και αποδοτική. Τα δεδομένα της Firestore οργανώνονται σε συλλογές που περιέχουν έγγραφα. Τα έγγραφα μπορούν να περιέχουν πολύπλοκα ένθετα αντικείμενα ή/και υποσυλλογές.

Σε αντίθεση με άλλες λύσεις του παρελθόντος, μέσω της Firestore οι συσκευές ενημερώνονται άμεσα για αλλαγές στα δεδομένα, χωρίς να χρειάζεται ο χρήστης να κάνει κάποια επιπλέον ενέργεια. Επίσης όσα δεδομένα της βάσης έχουν ληφθεί από το νέφος (cloud) διατηρούνται στις συσκευές των χρηστών, ώστε να είναι δυνατή η άμεση εμφάνισή τους, ανεξαρτήτως της διαθεσιμότητας και της ταχύτητας του διαδικτύου, ενώ όταν γίνεται οποιαδήποτε τροποποίηση στα δεδομένα στο νέφος (cloud), οι συσκευές να κατεβάζουν από το διαδίκτυο μόνο τις αλλαγές, και όχι τις πληροφορίες που ήδη έχουν. Αυτές οι λειτουργίες έχουν ως αποτέλεσμα η εφαρμογή να λειτουργεί εντυπωσιακά γρήγορα, κάνοντας παράλληλα πολύ αποδοτική χρήση του Διαδικτύου. Για την προστασία δεδομένων από μη εξουσιοδοτημένους χρήστες, η Firestore δίνει την δυνατότητα στους διαχειριστές να ορίζουν δυναμικούς κανόνες πρόσβασης, βάσει της ταυτότητας χρήστη και άλλων συνιστωσών.

2.3.3 Realtime Database

Η Realtime Database της Firebase είναι και αυτή όπως και η Firestore μία NoSQL βάση δεδομένων χωρίς να έχει όμως αυστηρούς τύπους, δηλαδή δεν χρειάζεται να οριστούν οι τύποι δεδομένων για κανένα σημείο της βάσης. Η δομή της είναι μια δομή δέντρου ξεκινώντας από μία ρίζα (root) και φτάνοντας μέχρι 32 επίπεδα. Επί της ουσίας είναι ένα μεγάλο JSON αρχείο το οποίο οι προγραμματιστές μπορούν να διαχειριστούν σε πραγματικό χρόνο. Χρησιμοποιώντας μία μόνο διεπαφή προγραμματισμού εφαρμογής (API), η βάση δεδομένων Firebase παρέχει στην εφαρμογή την τρέχουσα τιμή των δεδομένων και τις ενημερώσεις σε αυτά τα δεδομένα. Για να μην έχει πρόσβαση ο οποιοσδήποτε χρησιμοποιούνται κανόνες οι οποίοι γράφονται από τον προγραμματιστή [26].

2.3.4 Cloud Storage

Το Cloud Storage έχει σχεδιαστεί για να αποθηκεύει περιεχόμενο που δημιουργείται από τους χρήστες, συνήθως μεγάλα αρχεία όπως φωτογραφίες ή βίντεο. Συνήθως χρησιμοποιείται ή αναπτύσσεται για φωτογραφίες και βίντεο, αλλά μπορούμε να το χρησιμοποιήσουμε και για άλλα πράγματα, όπως αρχεία κειμένου.

Πρόκειται για ένα ισχυρό, οικονομικό και απλό μέσο αποθήκευσης αντικειμένων που έχει σχεδιαστεί ώστε να:

- α. Εκμεταλλεύονται τα συστήματα ασφαλείας της Google κατά τη μεταφόρτωση ή λήψη αρχείων στις εφαρμογές στο Firebase.
- β. Μπορούν να αποθηκευτούν εικόνες, αρχεία ήχου, βίντεο ή άλλο περιεχόμενο που δημιουργούν οι χρήστες.
- γ. Παρέχεται ταυτόχρονη πρόσβαση στα ίδια αρχεία στον διακομιστή.

Χρησιμοποιεί σύστημα με bucket, δηλαδή έναν χώρο που αποθηκεύονται μεγάλα αρχεία. Για την οργάνωση χρησιμοποιείται το σύστημα φακέλων/αρχείων που όλοι γνωρίζουν. Για να μην έχει πρόσβαση ο οποιοσδήποτε χρησιμοποιούνται κανόνες οι οποίοι γράφονται από τον προγραμματιστή [28].

2.3.5 Cloud Functions

Οι Firebase cloud functions είναι μία χωρίς διακομιστή υπηρεσία υπολογισμού (serverless) που προσφέρεται από την Google ως μέρος της πλατφόρμας Firebase. Δίνουν την δυνατότητα να γραφτεί λογική στην πλευρά του διακομιστή μιας εφαρμογής χωρίς να υπάρχει ανησυχία για τη διαχείριση της υποδομής του διακομιστή ή την κλιμάκωση των πόρων. Οι Cloud functions ενεργοποιούνται από γεγονότα όπως οι αλλαγές στα δεδομένα σε μία βάση δεδομένων Firebase, ή πιστοποίηση χρήστη (user authentication) ή HTTP αιτήματα (requests). Μπορούν να γραφτούν σε Node.js, Python, Go, Java ή .Net.

Οι Cloud Functions μπορούν να χρησιμοποιηθούν για διάφορες εργασίες, όπως η επεξεργασία δεδομένων χρήστη, η αποστολή ειδοποιήσεων, η δημιουργία αναφορών και η ενσωμάτωση με άλλες υπηρεσίες τρίτων. Είναι κλιμακωτές και μπορούν να χειριστούν ένα μεγάλο αριθμό αιτημάτων ταυτόχρονα, καθιστώντας τις ιδανικές για τη δημιουργία υψηλά αποκρίσιμων και κλιμακωτών εφαρμογών.

Μπορούν επίσης να αναπτυχθούν και να τοποθετηθούν απευθείας από την κονσόλα της Firebase (Firebase console) ή χρησιμοποιώντας την γραμμή εντολών Firebase (Firebase CLI). Μπορούν επίσης να δοκιμαστούν τοπικά χρησιμοποιώντας εργαλεία όπως την σουίτα προσομοίωσης Firebase (Firebase emulator suite). Η τιμολόγηση των Firebase Cloud Functions βασίζεται στη χρήση, με ένα γενναιόδωρο δωρεάν επίπεδο και επίπεδα τιμολόγησης που βασίζονται στον αριθμό των εκτελέσεων των λειτουργιών, το χρόνο εκτέλεσης και την εξερχόμενη κίνηση δικτύου.[31]

Οι Cloud Functions, χωρίζονται σε ορισμένες κατηγορίες ανάλογα με τον τρόπο χρήσης τους:

- α. Callable functions
- β. Http request functions
- γ. Schedule functions
- δ. Database triggers
- ε. Firestore triggers
- στ. Authentication triggers
- ζ. Storage triggers
- η. Google Analytics triggers
- θ. Remote Config triggers
- ι. Cloud Pub/Sub triggers

2.3.5.1 Callable Functions

Οι Cloud Functions για το Firebase σερβιρ ανάπτυξης λογισμικού (Software Development Kit ή SDK) των πελατών (client) επιτρέπουν να καλούνται συναρτήσεις (functions) απευθείας από μία εφαρμογή Firebase. Για να κληθεί μία συνάρτηση από μια εφαρμογή με αυτόν τον τρόπο, θα πρέπει να γραφτεί και να υλοποιηθεί μια συνάρτηση κλήσης μέσω HTTPS (HTTPS Callable function) στις Cloud Functions, και στη συνέχεια να προστεθεί η λογική του πελάτη για να κληθεί η συνάρτηση από την εφαρμογή.

Είναι σημαντικό να αναφερθεί ότι οι HTTPS Callable Functions είναι παρόμοιες αλλά όχι ίδιες με τις HTTP Functions. Για να χρησιμοποιηθούν οι HTTP CALLABLE Functions πρέπει να χρησιμοποιηθεί το client SDK για την πλατφόρμα με το functions.https backend API.

Για να δημιουργηθεί μια HTTPS callable πρέπει να χρησιμοποιηθεί η συνάρτηση functions.https.onCall. Αυτή η μέθοδος δέχεται δύο παραμέτρους:

- α. τα δεδομένα (data)
- β. προαιρετικά, το context το οποίο περιέχει πληροφορίες εξουσιοδότησης χρήστη (user auth information).

2.3.5.2 Http Request Functions

Μπορεί να ενεργοποιηθεί μια συνάρτηση μέσω ενός HTTP αιτήματος χρησιμοποιώντας το functions.https. Αυτό σημαίνει ότι μπορεί να κληθεί μία συγχρονισμένη συνάρτηση μέσω των ακολουθών υποστηριζόμενων μεθόδων HTTP: GET, POST, PUT, DELETE ΚΑΙ OPTIONS.

Ο χειριστής συμβάντων (event handler) για μια HTTP συνάρτηση ακούει για το συμβάν onRequest(), το οποίο υποστηρίζει δρομολογητές (routers) και εφαρμογές (apps) που διαχειρίζονται από το πλαίσιο Express web .[31]

2.3.5.3 Schedule Functions

Οι Schedule functions είναι μέθοδοι οι οποίες τρέχουν συγκεκριμένες χρονικές στιγμές, χρησιμοποιώντας το functions.pubsub.schedule().onRun(). Αυτή η μέθοδος εκτελεί μια συνάρτηση σύμφωνα με το προγραμματισμένο πρόγραμμα.[31]

2.3.5.4 Database Triggers

Με τις Cloud Functions, μπορεί να γίνει διαχείριση συμβάντων στη βάση Δεδομένων της Firebase Realtime Database χωρίς να χρειάζεται η ενημέρωση του κώδικα του πελάτη (client code). Οι Cloud Functions δίνουν την δυνατότητα να εκτελεστούν λειτουργίες στην Βάση Δεδομένων σε πραγματικό χρόνο με πλήρη διαχειριστικά δικαιώματα και εξασφαλίζουν ότι κάθε αλλαγή στη βάση δεδομένων σε πραγματικό χρόνο επεξεργάζεται ξεχωριστά.

Οι συναρτήσεις επιτρέπουν στον προγραμματιστή να διαχειριστεί συμβάντα στη Βάση Δεδομένων σε πραγματικό χρόνο σε δύο επίπεδα ειδικότητας. Υποστηρίζουν συμβάντα όπως η δημιουργία, η ενημέρωση και η διαγραφή. Οι Cloud Functions υποστηρίζουν τους χειριστές συμβάντων για τη Βάση Δεδομένων σε πραγματικό χρόνο:

- α. onWrite() που ενεργοποιείται όταν δημιουργούνται, ενημερώνονται ή διαγράφονται δεδομένα στην Realtime Database.
- β. onCreate() που ενεργοποιείται όταν δημιουργούνται νέα δεδομένα στην Realtime Database.

- γ. `onUpdate()` που ενεργοποιείται όταν ενημερώνονται νέα δεδομένα στην Realtime Database.
- δ. `onDelete()` που ενεργοποιείται όταν διαγράφονται νέα δεδομένα στην Realtime Database.

2.3.5.5 Firestore Triggers

Στο Cloud Firestore, όπως και στη Realtime Database δίνεται η δυνατότητα διαχείρισης συμβάντων χωρίς να χρειάζεται η ενημέρωση του κώδικα του πελάτη. Μπορούν να γίνουν αλλαγές στο Cloud Firestore μέσω της διεπαφής DocumentSnapShot ή μέσω του σετ ανάπτυξης διαχειριστή (Admin SDK).

Το SDK των Cloud Functions της Firebase εξάγει ένα αντικείμενο `functions.firestore` που επιτρέπει την δημιουργία χειριστών όπου συνδέονται με συγκεκριμένα συμβάντα:

- α. `onCreate()` που ενεργοποιείται όταν ένα document γράφεται για πρώτη φορά
- β. `onUpdate()` που ενεργοποιείται όταν ένα υπάρχον document έχει αλλάξει οποιαδήποτε τιμή του
- γ. `onDelete()` που ενεργοποιείται όταν διαγράφεται ένα document
- δ. `onWrite()` που ενεργοποιείται εάν κάποιο από τα 3 παραπάνω ενεργοποιηθεί.

2.3.5.6 Authentication triggers

Μπορεί να δημιουργηθεί μια Cloud Function όταν δημιουργηθεί η διαγραφεί ένας λογαριασμός χρήστη στην Firebase. Για παράδειγμα, να σταλεί ένα email καλωσορίσματος σε έναν χρήστη που μόλις δημιούργησε έναν λογαριασμό στην εφαρμογή.

Το SDK των Cloud Functions της Firebase εξάγει ένα αντικείμενο `functions.auth.user()` που επιτρέπει την δημιουργία χειριστών που συνδέονται με τα δύο συμβάντα που προαναφέρθηκαν:

- α. `onCreate()` που ενεργοποιείται μόλις δημιουργηθεί ένας χρήστης.
- β. `onDelete()` που ενεργοποιείται μόλις διαγραφεί ένας χρήστης.

2.3.5.7 Cloud Storage Triggers

Μπορεί να δημιουργηθεί μια Cloud Function στην περίπτωση ενημέρωσης ή διαγραφής αρχείων και φακέλων στο Cloud Storage.

Το Cloud Storage υποστηρίζει τα παρακάτω συμβάντα:

- α. `onArchive` Αποστέλλεται μόνο όταν ένα bucket έχει ενεργοποιήσει τον εκδοτικό έλεγχο αντικειμένου (object versioning). Αυτό το γεγονός υποδηλώνει ότι η ζωντανή έκδοση ενός αντικειμένου έχει γίνει αρχειοθετημένη έκδοση, είτε επειδή αυτό αρχειοθετήθηκε είτε επειδή αντικαταστάθηκε από τη μεταφόρτωση ενός αντικειμένου με το ίδιο όνομα.
- β. `onDelete` Αποστέλλεται όταν ένα αντικείμενο έχει διαγραφεί μόνιμα. Αυτό περιλαμβάνει αντικείμενα που αντικαταστάθηκαν ή διαγράφηκαν ως μέρος της διαμόρφωσης κύκλου ζωής του bucket.
- γ. `onFinalize` Αποστέλλεται όταν ένα νέο αντικείμενο (ή μια νέα γενιά ενός υπάρχοντος αντικειμένου) δημιουργείται με επιτυχία στο bucket. Αυτό περιλαμβάνει την αντιγραφή ή την επανεγγραφή ενός υπάρχοντος αντικειμένου.

- δ. `onMetadataUpdate` Αποστέλλεται όταν τα μεταδεδομένα ενός υπάρχοντος αντικειμένου αλλάζουν.

2.3.5.8 Google Analytics triggers

Το Google Analytics παρέχει αναφορές γεγονότων που βοηθούν στην κατανόηση της αλληλεπίδρασης των χρηστών μιας εφαρμογής. Αυτό συμβαίνει επειδή δίνεται πρόσβαση πρόσβαση στα συμβάντα μετατροπής που έχουν καταγραφεί από συσκευές Apple και Android και να ενεργοποιηθούν λειτουργίες βάσει αυτών των συμβάντων.

Το Cloud Functions υποστηρίζει το `AnalyticsEvent` του Google Analytics. Αυτό το συμβάν ενεργοποιείται κάθε φορά που η δραστηριότητα του χρήστη παράγει ένα συμβάν μετατροπής.

2.3.5.9 Remote Config Triggers

Είναι συνάρτηση ανταπόκρισης σε συμβάντα απομακρυσμένης διαμόρφωση της Firebase (Firebase Remote Config), συμπεριλαμβανομένης της δημοσίευσης μιας νέας έκδοσης ρυθμίσεων ή της επαναφοράς σε μια παλαιότερη έκδοση. Το αντικείμενο `TemplateVersion` που επιστρέφεται από το `onUpdate` περιέχει τα βασικά μεταδεδομένα ενός ενημερωμένου προτύπου, όπως ο αριθμός έκδοσης και η ώρα ενημέρωσης. Δίνεται επίσης η δυνατότητα ανάκτησης του email του χρήστη που έκανε την ενημέρωση, με το όνομα και μια εικόνα αν είναι διαθέσιμα.

2.3.5.10 Pub/Sub Triggers

Το Pub/Sub είναι ένα κατακευματισμένο δίκτυο μηνυμάτων που κλιμακώνεται αυτόματα. Μια τέτοια συνάρτηση μπορεί να ενεργοποιηθεί κάθε φορά που στέλνεται ένα νέο μήνυμα Pub/Sub σε ένα συγκεκριμένο θέμα.

2.3.6 Hosting

Η φιλοξενία Firebase (Firebase hosting) δίνει την δυνατότητα στους προγραμματιστές να ανεβάσουν την εφαρμογή/ιστοσελίδα στους διακομιστές της Google. Αποτελεί μια φιλοξενία ιστοσελίδων υψηλής παραγωγικότητας για τους προγραμματιστές. Δίνει την δυνατότητα να ανέβει πολύ γρήγορα και εύκολα μια ιστοσελίδα και να παρέχει τόσο στατικό όσο και δυναμικό περιεχόμενο σε ένα παγκόσμιο δίκτυο παράδοσης περιεχομένου με μόνο μία εντολή. Ο προεπιλεγμένος σύνδεσμος της εφαρμογής μέσω της υπηρεσίας Hosting είναι του τύπου 'όνομα εφαρμογής'.web.app.

Προς το παρόν, το μεγαλύτερο μέρος της φιλοξενία ιστοσελίδων είναι με κόστος και οι διαθέσιμες δωρεάν υπηρεσίες φιλοξενίας είναι αργές. Επιπλέον υπάρχει έξτρα κόστος για την λήψη του πιστοποιητικού SSL (Επίπεδο σύνδεσης ασφαλείας) για να μετατρέψει τον ιστότοπο σε ασφαλή με HTTPS.

Το Firebase Hosting είναι δωρεάν. Επίσης από προεπιλογή, παρέχει πιστοποιητικό SSL και προσφέρει εντυπωσιακή ταχύτητα σε διάφορες γεωγραφικές τοποθεσίες χωρίς καμία απαίτηση για ξεχωριστό δίκτυο παροχής περιεχομένου (CDN) στην κορυφή.

Υπάρχουν οι ακόλουθες απαιτήσεις για τη χρήση του Firebase Hosting:

- α. **Λογαριασμός Google** - Στην παρούσα εποχή, όλοι έχουν λογαριασμό Google/Gmail. Ο λογαριασμός Gmail είναι αρκετός για να χρησιμοποιηθεί.
- β. **Firebase-CLI** - Για την εγκατάσταση της Firebase-CLI, απαιτείται να έχουμε το Node.js στον υπολογιστή μας. Έτσι, πρώτα θα εγκατασταθεί το Node.js στον υπολογιστή και ύστερα θα

εγκατασταθεί το Firebase CLI χρησιμοποιώντας το npm (Node Package Manager) εκτελώντας την ακόλουθη εντολή ‘npm install -g firebase-tools’.

- γ. **Domain** - Αυτή είναι η προαιρετική απαίτηση. Υπάρχουν αρκετοί πάροχοι ονομάτων domain στον κόσμο. Ένα παράδειγμα είναι από το Godaddy.com.

2.4 FLUTTER

Το πλαίσιο Flutter είναι μία πλατφόρμα ανάπτυξης εφαρμογών κινητών τηλεφώνων που δημιουργήθηκε από τη Google. Επιτρέπει στους προγραμματιστές να δημιουργούν εφαρμογές ιστού web εφαρμογές, desktop εφαρμογές αλλά και εφαρμογές που τρέχουν σε συσκευές Android και iOS χρησιμοποιώντας τον ίδιο κώδικα, καθιστώντας την ανάπτυξη κινητών εφαρμογών πιο γρήγορη και εύκολη από παραδοσιακές μεθόδους. Το πλαίσιο Flutter χρησιμοποιεί την γλώσσα προγραμματισμού Dart.

2.4.1 Χαρακτηριστικά Flutter

Μερικά από τα κορυφαία χαρακτηριστικά του πλαισίου Flutter περιλαμβάνουν:

- α. Γλώσσα πραγματισμού Dart: Το πλαίσιο Flutter χρησιμοποιεί τη γλώσσα προγραμματισμού Dart, η οποία είναι εύκολη στην εκμάθηση και επιτρέπει την ανάπτυξη σε εφαρμογές υψηλής ποιότητας.
- β. Hot reload: Η δυνατότητα “hot reload” του πλαισίου Flutter επιτρέπει την γρήγορη και εύκολη αλλαγή στις εφαρμογές χωρίς να χρειάζεται η επανεκκίνηση. Αυτή η δυνατότητα επιτρέπει στους προγραμματιστές να δουν άμεσα τις αλλαγές που κάνουν στον κώδικα να εκδηλώνονται αμέσως στην τρέχουσα εφαρμογή. Αυτό σημαίνει ότι οι προγραμματιστές μπορούν να πειραματίζονται, να επαναλαμβάνουν και να διορθώνουν προβλήματα χωρίς την ανάγκη επανεκκίνησης της εφαρμογής με αποτέλεσμα να επιταχύνει τη διαδικασία ανάπτυξης και της αποσφαλμάτωσης (debugging) και βελτιώνει συνολικά την παραγωγικότητα.
- γ. Ευέλικτη διεπαφή χρήστη: Τα στοιχεία διεπαφής χρήστη του πλαισίου Flutter κατασκευάζονται χρησιμοποιώντας τις ίδιες αρχές με τις κατευθυντήριες γραμμές σχεδιασμού της Google, το Material Design, παρέχοντας μια ευέλικτη διαδικασία δημιουργίας όμορφων εφαρμογών.
- δ. Native performance: Οι εφαρμογές Flutter μεταγλωττίζονται σε εγγενή κώδικα (native), παρέχοντας την καλύτερη δυνατή απόδοση και στα δύο συστήματα iOS και Android. Αντίθετα με άλλα πλαίσια που εξαρτώνται από γέφυρες επικοινωνίας με κώδικα εγγενή κώδικα (native), το Flutter μεταγλωττίζεται άμεσα σε κώδικα μηχανής ARM και x86. Αυτή η προσέγγιση οδηγεί σε ταχύτερη εκτέλεση και εξαλείφει τα προβλήματα απόδοσης που σχετίζονται με τις γέφυρες. Οι εφαρμογές του πλαισίου Flutter επιδεικνύουν εξαιρετική απόδοση, ανταγωνιζόμενες με εκείνες που αναπτύσσονται χρησιμοποιώντας τεχνολογίες που είναι ειδικές για την κάθε πλατφόρμα.
- ε. Open source: Το Flutter είναι ένα έργο ανοιχτού κώδικα, που σημαίνει ότι μπορεί να χρησιμοποιηθεί δωρεάν αλλά υπάρχει και η δυνατότητα να συνεισφέρει ο οποιοσδήποτε στην ανάπτυξη της πλατφόρμας. Το πλαίσιο Flutter είναι ένα έργο ανοιχτού κώδικα, που σημαίνει ότι μπορεί να χρησιμοποιηθεί δωρεάν αλλά υπάρχει και η δυνατότητα να συνεισφέρει ο οποιοσδήποτε στην ανάπτυξη της πλατφόρμας. Έτσι έχει συγκεντρωθεί μια εκτεταμένη και δραστήρια κοινότητα προγραμματιστών. Αυτή η κοινότητα συνεισφέρει ενεργά στην

πλατφόρμα δημιουργώντας και διατηρώντας πακέτα, πρόσθετα και επεκτάσεις που ενισχύουν τις δυνατότητες του Flutter. Η ανοικτού κώδικα φύση του πλαισίου Flutter σημαίνει επίσης ότι οι προγραμματιστές μπορούν να έχουν πρόσβαση στον πηγαίο κώδικα, παρέχοντας διαφάνεια και επιτρέποντας τους να κατανοήσουν πώς λειτουργεί το πλαίσιο εσωτερικά.

στ. Ανταποκρίσιμη Διεπαφή Χρήστη και Δυνατότητα Προσαρμογής στη Συσκευή

Το πλαίσιο Flutter επιτρέπει στους προγραμματιστές να δημιουργούν διεπαφές χρήστη υψηλής ανταπόκρισης και προσαρμόσιμες, προσαρμόζονται αυτόματα σε διάφορα μεγέθη οθονών και προσανατολισμούς. Αυτό εξασφαλίζει ότι η εφαρμογή φαίνεται εξαιρετική και λειτουργεί άψογα σε διάφορες συσκευές, από κινητά και tablets έως προγράμματα περιήγησης ιστού και οθόνες επιτραπέζιων υπολογιστών. [30,32]

2.4.2 Πως λειτουργεί το Flutter

Κατά τη δημιουργία μιας εφαρμογής Flutter, οι προγραμματιστές θα χρησιμοποιήσουν τα “widgets”. Τα widgets είναι οι βασικά εργαλεία μιας εφαρμογής Flutter και χρησιμοποιούνται για να δημιουργήσουν τα οπτικά στοιχεία μιας εφαρμογής (όπως κουμπιά και κείμενο) και τα λειτουργικά στοιχεία (όπως τα Stateless Widgets). Υπάρχουν δύο είδη widgets: τα Stateless Widgets και τα Stateful Widgets.

2.4.2.1 Stateless Widgets

Όπως υποδηλώνει το όνομα τους, τα Stateless Widgets είναι εκείνα που δεν έχουν εσωτερική κατάσταση. Αυτά είναι τα πιο απλά widgets και χρησιμοποιούνται συχνά για κουμπιά ή κείμενο και το περιεχόμενο και η κατάσταση τους δεν μπορούν να ενημερωθούν αφού δημιουργηθούν.

Για να παρέχουμε ένα πραγματικό παράδειγμα, τα Stateless Widgets είναι σαν μια πινακίδα ή μια αφίσα. Μια πινακίδα παρουσιάζει ένα σταθερό μήνυμα ή πληροφορίες που παραμένουν αμετάβλητα. Οι άνθρωποι περνούν από την πινακίδα, βλέπουν το περιεχόμενό της και μπορεί να συγκεντρώνουν πληροφορίες από αυτήν. Ωστόσο, η ίδια η πινακίδα δεν αλλάζει. Παρουσιάζει το ίδιο μήνυμα όπως έκανε αρχικά. Με παρόμοιο τρόπο, τα Stateless Widgets δημιουργούν μια διεπαφή χρήστη με τα δεδομένα που δίνονται αρχικά και παραμένουν αμετάβλητα στη συνέχεια.

2.4.2.2 Stateful Widgets

Από την άλλη πλευρά, τα Stateful Widgets έχουν εσωτερική κατάσταση και αυτή η κατάσταση μπορεί να αλλάξει με τον χρόνο και θα αντανakλάται στο πως το widget φαίνεται και συμπεριφέρεται. Τα Stateful widgets χρησιμοποιούνται συχνά για πεδία εισόδου χρήστη ή για τη διαχείριση κινούμενων γραφικών (animations).

Για να παρέχουμε ένα πραγματικό παράδειγμα ένα Stateful Widget μοιάζει περισσότερο με ένα παιχνίδι σκακιού. Εδώ, πολλά κομμάτια μετακινούνται, αλλά αρκεί μόνο η μετακίνηση ενός κομματιού για να θεωρηθεί αλλαγή κατάστασης. Δεν χρειάζεστε πολλές διάφορες αλλαγές. Χρειάζεστε μόνο μία αλλαγή για να απαιτηθεί η μετατροπή του Stateless Widget σας σε ένα Stateful Widget. Τα Stateless Widgets δεν έχουν καμία αλλαγή.[33]

2.4.3 Πλεονεκτήματα

Υπάρχουν διάφορα σημαντικά οφέλη από τη χρήση του πλαισίου Flutter για την ανάπτυξη κινητών εφαρμογών. Αυτά περιλαμβάνουν:

- α. Είναι γρήγορο: Χρησιμοποιεί τη γλώσσα προγραμματισμού Dart που μεταγλωττίζεται σε native κώδικα. Αυτό έχει ως αποτέλεσμα εφαρμογές που είναι γρήγορες και ανταποκρίνονται γρήγορα στη χρήση τους.
- β. Δημιουργεί εφαρμογές πολλαπλών πλατφορμών (cross-platform): Ο ίδιος κώδικας μπορεί να χρησιμοποιηθεί για την δημιουργία εφαρμογών για συσκευές iOS και Android από μια μόνο βάση κώδικα αντί να αλλάζετε μεταξύ διαφορετικών πλατφορμών. Αυτό μπορεί να εξοικονομήσει πολύ χρόνο και προσπάθεια κατά την ανάπτυξη εφαρμογών για κινητές συσκευές. Επιπλέον, η Flutter μπορεί να χρησιμοποιηθεί για ανάπτυξη ιστοσελιδών (web development) για την δημιουργία web εφαρμογών.
- γ. Διαθέτει μια πλούσια συλλογή από widgets: τα widgets είναι τα βασικά στοιχεία κατασκευής των εφαρμογών Flutter και είναι διαθέσιμα σε μεγάλη ποικιλία. Αυτό καθιστά εύκολη τη δημιουργία όμορφων και εξατομικευμένων διεπαφών χρήστη.
- δ. Είναι Open Source: Οποιοσδήποτε μπορεί να συνεισφέρει στην ανάπτυξη του Flutter και μια αναπτυσσόμενη κοινότητα προγραμματιστών το χρησιμοποιεί. Επιπλέον, πολλές χρήσιμες τεκμηριώσεις και εκπαιδευτικά υλικά είναι διαθέσιμα στο διαδίκτυο, δημιουργημένα από την κοινότητα του Flutter σε ιστότοπους όπως το Github.
- ε. Είναι δωρεάν: Δεν υπάρχει καμία χρέωση για την ανάπτυξη εφαρμογών σε Flutter. Αυτό το καθιστά μια ελκυστική επιλογή για νεοσύστατες (startup) επιχειρήσεις και προγραμματιστές που θέλουν να δημιουργήσουν υψηλής ποιότητας εφαρμογές χωρίς να δαπανήσουν πολλά χρήματα.
- στ. Υποστηρίζεται από την Google. Ως προϊόν της Google, λαμβάνει σημαντική υποστήριξη, η οποία δουλεύει συνεχώς για να το βελτιώσει. Αυτό σημαίνει ότι οι προγραμματιστές μπορούν να είναι βέβαιοι ότι το Flutter θα συνεχίσει να αναπτύσσεται και να υποστηρίζεται.
- ζ. Εύκολη αποσφαλμάτωση (debugging): Η γλώσσα προγραμματισμού Dart διαθέτει εξαιρετικά εργαλεία για την αποσφαλμάτωση, όπως ο αναλυτής του Dart και η σουίτα DevTools. Αυτό καθιστά εύκολο τον εντοπισμό και την επιδιόρθωση σφαλμάτων στις εφαρμογές Flutter.
- η. Αυτόματο testing: Η γλώσσα προγραμματισμού Dart έχει καλή υποστήριξη για αυτόματες δοκιμές (testing), και το πλαίσιο του Flutter έχει επίσης τα δικά του εργαλεία για τις δοκιμές.
- θ. Η χρήση της τεχνολογίας Flutter επιτρέπει στις εφαρμογές να έχουν πρόσβαση στις πλήρεις δυνατότητες του hardware υλικού αλλά και του software. Αυτό σημαίνει ότι μπορούν να αξιοποιούν χαρακτηριστικά όπως η κάμερα, το GPS και ο αναγνώστης δακτυλικών αποτυπωμάτων.
- ι. Διαφορετική προσαρμοστικότητα οθόνης: Οι εφαρμογές Flutter μπορούν να σχεδιαστούν έτσι ώστε να λειτουργούν σε διάφορα μεγέθη οθόνης και κατανομές οθόνης. Αυτό καθιστά εύκολη τη δημιουργία εφαρμογών που φαίνονται εξαιρετικές τόσο σε τηλέφωνα όσο και σε tablet [30].

Κεφάλαιο 3ο: Ανάπτυξη εφαρμογών

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα αναφερθούμε στην αρχιτεκτονική και την μεθοδολογία ανάπτυξης των εφαρμογών. Για τον σκοπό αυτό, θα αναλύσουμε την ανάπτυξη των εφαρμογών, την δομή της βάσης δεδομένων καθώς και την διασύνδεση με αυτήν, όπως και ότι API αναπτύχθηκε.

Η εφαρμογή απαρτίζεται από δύο βασικά κομμάτια.

Την διαχειριστική εφαρμογή ιστού, η οποία αναπτύχθηκε με χρήση του πλαισίου ανάπτυξης εφαρμογών ιστού Angular.

Την εφαρμογή κινητών συσκευών, η οποία απευθύνεται στους τελικούς χρήστες, η οποία αναπτύχθηκε με την χρήση του πλαισίου ανάπτυξης εφαρμογών κινητών συσκευών Flutter.

Η εφαρμογή ιστού αποτελεί το κέντρο ελέγχου του συστήματος, προσφέροντας μια προηγμένη διεπαφή χρήστη για την αποτελεσματική διαχείριση των πόρων και των δεδομένων. Η ανάπτυξη της έγινε με το πλαίσιο Angular, προϊόν της Google, και ένα από τα σταθερά πιο διαδεδομένα πλαίσια ανάπτυξης εφαρμογών ιστού, το οποίο ακολουθεί τις αρχές της σύγχρονης ανάπτυξης εφαρμογών ιστού. Με βασική φιλοσοφία την δημιουργία μιας ευέλικτης και ευανάγνωστης δομής, η εφαρμογή ιστού προσφέρει στον χρήστη τη δυνατότητα εύκολης και αποτελεσματικής πλοήγησης και αλληλεπίδρασης με το σύστημα με σκοπό την παραμετροποίηση της εφαρμογής χρηστών και την διαχείριση οντοτήτων της βάσης δεδομένων.

Η εφαρμογή κινητών συσκευών αναπτύχθηκε με το πλαίσιο Flutter, επίσης προϊόν της Google. Βασικοί πυλώνες του πλαισίου Flutter είναι η δυνατότητα ανάπτυξης μέσω μίας ενιαίας βάσης κώδικα για τον ανάπτυξη εφαρμογών για Android και iOS, εξασφαλίζοντας εύχρηστη και ομοιογενή εμπειρία ανάπτυξης, και παράγει εγγενή (native) στοιχεία για κάθε λειτουργικό σύστημα, τα οποία αξιοποιούνται με τον καλύτερο τρόπο από το εκάστοτε λειτουργικό σύστημα.

Ως βάση δεδομένων επιλέξαμε την Firestore, η οποία χρησιμοποιεί αρχιτεκτονική NoSQL, αξιοποιώντας μία δομή συλλογών και εγγράφων για την διατήρηση δεδομένων. Η Firestore είναι εργαλείο της σουίτας Firebase, μιας ολοκληρωμένης πλατφόρμας ανάπτυξης από τη Google η οποία παρέχει λύσεις για την δωρεάν, και μη, αποθήκευση δεδομένων στο νέφος (cloud), την επιτήρηση και ανάλυση, καθώς και την αυθεντικοποίηση χρηστών.

Συνοψίζοντας, η πτυχιακή εργασία μας παρουσιάζει ένα παράδειγμα ενοποίησης προηγμένων τεχνολογιών ιστού και κινητών συσκευών, υπό την αιγίδα και αξιοπιστίας της Google. Η συνεργασία μεταξύ των πλαισίων Angular και Flutter, μαζί με την σουίτα Firebase δημιουργεί ένα ολοκληρωμένο οικοσύστημα που συνδυάζει την ευελιξία, την απόδοση και την αξιοπιστία.

3.2 Σχεδιασμός και υλοποίηση της βάσης δεδομένων

Κατά την αρχική φάση σχεδίασης του έργου μας, καθορίσαμε ως πρωταρχικό στόχο τη δημιουργία μιας αξιόπιστης και αποδοτικής δομής βάσης δεδομένων. Η ανάπτυξη αυτής της βάσης αποτέλεσε το θεμέλιο για τη συνέχιση του έργου μας, ενισχύοντας τη δυνατότητα αποθήκευσης, ανάκτησης και ενημέρωσης πληροφοριών, καθώς και την μελλοντική επεκτασιμότητα των εφαρμογών.

Σε αυτό το πλαίσιο, επιλέξαμε να χρησιμοποιήσουμε προχωρημένες τεχνικές μοντελοποίησης δεδομένων, δημιουργώντας ευέλικτα και στιβαρά μοντέλα δεδομένων (interfaces). Τα μοντέλα αυτά

διαδραματίζουν καθοριστικό ρόλο στον τρόπο με τον οποίο τα δεδομένα αλληλεπιδρούν με τη βάση δεδομένων, παρέχοντας μια ευανάγνωστη και αποτελεσματική διεπαφή με το σύστημα.

Κατά τον σχεδιασμό των μοντέλων, λάβαμε υπόψη μας την ανάγκη για επεκτασιμότητα και αποτελεσματική διαχείριση των πληροφοριών. Τα μοντέλα σχεδιάστηκαν λαμβάνοντας υπόψη τις σύγχρονες απαιτήσεις της ανάπτυξης λογισμικού, ενώ παράλληλα εξασφαλίζουν τη συμβατότητα και την αποδοτικότητα των εφαρμογών που θα χρησιμοποιήσουν αυτά τα μοντέλα.

Η διαδικασία αυτή αντιπροσωπεύει τη σύνθεση ενός ολοκληρωμένου και συνεκτικού περιβάλλοντος για τη διαχείριση των δεδομένων, ενισχύοντας την αποτελεσματικότητα της εφαρμογής μας και παρέχοντας μια σταθερή βάση για την ανάπτυξη και την επέκταση του συστήματός μας.

3.3 Εφαρμογή διαχείρισης

Σκοπός της εφαρμογής διαχείρισης είναι η διαχείριση κειμένων, πόρων και οποιαδήποτε άλλη παραμετροποίηση απαιτείται, για την ομαλή λειτουργία της εφαρμογής χρηστών.

3.3.1 Δυνατότητες διαχειριστή

Ο διαχειριστής έχει την δυνατότητα να:

- Ρυθμίσει τις γλώσσες στις οποίες θα είναι διαθέσιμες οι εφαρμογές.
- Ρυθμίσει τις πόλεις στις οποίες θα υπάρχουν πόροι, όπως μουσεία, σημεία ενδιαφέροντος, αξιοθέατα, οι οποίες θα είναι διαθέσιμες στην εφαρμογή, με σκοπό την παραμετροποίηση αυτής.
- Δημιουργήσει και να επεξεργαστεί πληροφορίες για τις πανεπιστημιούπολεις του ΔΙ.ΠΑ.Ε καθώς και τα τμήματα αυτών, για περαιτέρω παραμετροποίηση της εφαρμογής τελικών χρηστών
- Δημιουργήσει και να επεξεργαστεί πληροφορίες για τις πρεσβείες που βρίσκονται σε όλη την Ελλάδα.
- Δημιουργήσει, επεξεργαστεί και να διαγράψει στατικά κείμενα και πόρους. Αυτά αναφέρονται κυρίως σε πόρους οι οποίοι παραμένουν κυρίως αμετάβλητοι όπως το νομοθετικό πλαίσιο το οποίο διέπει το πρόγραμμα Erasmus+, διευθύνσεις/τοποθεσίες δομών του πανεπιστημίου, στοιχεία επικοινωνίας κ.ο.κ.
- Αναζητήσει, δημιουργήσει και να αναρτήσει, επεξεργαστεί και να διαγράψει δυναμικά κείμενα και πόρους. Αυτά αναφέρονται σε ευαίσθητες στον χρόνο ανακοινώσεις που αφορούν το πρόγραμμα Erasmus+, εκδηλώσεις του πανεπιστημίου, προγράμματα υποτροφίας ή επιμόρφωσης, για τα οποία έχει υλοποιηθεί ειδική γραφική διεπαφή στην εφαρμογή τελικών χρηστών.
- Προσθέσει διαδραστικούς χάρτες για τις πανεπιστημιούπολεις, ορίζοντας περιοχές με ειδικές πληροφορίες για αυτές, με σκοπό την γρήγορη αναζήτηση πληροφοριών και εξοικείωση του χρήστη με τους χώρους του πανεπιστημίου.
- Δημιουργήσει, επεξεργαστεί και να διαγράψει κόμβους στον δενδροδιάλογο του ψηφιακού βοηθού (Chatbot).
- Προσθέσει και να επεξεργαστεί μεταφράσεις για τα κείμενα που αφορούν όλα τα παραπάνω καθώς και λέξεις κλειδιά που αναφέρονται σε στατικά στοιχεία της εφαρμογής κινητών

συσκευών, προσφέροντας την δυναμική μεταγλώττιση σύμφωνα με την γλώσσα επιλογής του χρήστη.

Επίσης, ο υπερ-διαχειριστής επιπλέον των επάνω δυνατοτήτων έχει την δυνατότητα να:

- Δημιουργήσει και να επεξεργαστεί τους υπόλοιπους διαχειριστές της εφαρμογής

3.3.2 Ανάπτυξη διαχειριστικής εφαρμογής ιστού

3.3.2.1 Περιβάλλον ανάπτυξης

Κατά την ανάπτυξη του διαχειριστικού μέρους της εφαρμογής, το πρώτο βήμα αποτέλεσε τη δημιουργία του αποθετηρίου κώδικα (repository) στην εφαρμογή ελέγχου εκδόσεων (version control system) GitHub, ένα βήμα κρίσιμο για τον αποδοτικό συντονισμό και τη συνεργασία μεταξύ των μελών της ομάδας ανάπτυξης. Με βάση τη φιλοσοφία της εξελιγμένης ανάπτυξης, επικεντρωθήκαμε στην οργάνωση και τη διαχείριση του κώδικα, διασφαλίζοντας τη συνέπεια και την αποτελεσματικότητα της διαδικασίας ανάπτυξης.

Στη συνέχεια, επικεντρωθήκαμε στην ενσωμάτωση απαραίτητων εργαλείων και πλαισίων για τη βελτιστοποίηση της χρήσης του πλαισίου ανάπτυξης Angular. Ενσωματώσαμε την βιβλιοθήκη Angular Material, η οποία προσφέρει ένα πλούσιο σύνολο στοιχείων γραφικών, και μη, διεπαφών χρήστη. Επιπλέον, ενσωματώσαμε το πλαίσιο μορφοποίησης στοιχείων HTML Tailwind (Tailwind CSS), επιτρέποντας την ευέλικτη σχεδίαση και βελτίωση της εμπειρίας του χρήστη σε όλη την εφαρμογή.

Ένα σημαντικό στοιχείο της ανάπτυξης αποτέλεσε το σετ εργαλείων SVG Icons module, το οποίο δημιουργήθηκε για τον χειρισμό προσαρμοσμένων εικονιδίων μορφής SVG. και επιτρέπει την εύκολη ενσωμάτωση και χειρισμό αυτών, ενισχύοντας την αισθητική και την αναγνωρισιμότητα της εφαρμογής.

Με αυτά τα βασικά και απαραίτητα εργαλεία ενσωματωμένα στο έργο μας, προχωρήσαμε στην επόμενη φάση της ανάπτυξης, επιδιώκοντας τη δημιουργία μιας πλήρως λειτουργικής και εντυπωσιακής διαχειριστικής εφαρμογής.

3.3.3 Αυθεντικοποίηση χρηστών

Ο πρώτος μας στόχος ήταν η εξασφάλιση ενός ασφαλούς αλλά και χρηστικού συστήματος εισόδου για τους χρήστες. Για να το πετύχουμε αυτό, ενσωματώσαμε την βιβλιοθήκη AngularFire στο έργο μας. Η χρήση αυτής της βιβλιοθήκης επέτρεψε την ομαλή ενσωμάτωση με την υπηρεσία αυθεντικοποίησης Firebase. Με τη βοήθεια της AngularFire, δημιουργήσαμε μία υπηρεσία αυθεντικοποίησης (Auth Service), η οποία αποτελεί τον πυρήνα για όλες τις λειτουργίες σύνδεσης και αυθεντικοποίησης.

Στην υπηρεσία αυθεντικοποίησης ενσωματώσαμε λειτουργίες σύνδεσης (sign-in), αποσύνδεσης (logout), καθώς και τη δυνατότητα επαναφοράς κωδικού πρόσβασης (reset password). Με αυτόν τον τρόπο, παρέχεται μια πλήρης σουίτα λειτουργιών ασφαλούς εισόδου και διαχείρισης του λογαριασμού για τους χρήστες.

Ο συνδυασμός της AngularFire με την υπηρεσία αυθεντικοποίησης μας επέτρεψε να υλοποιήσουμε όχι μόνο μια λειτουργική λύση σύνδεσης αλλά και να ενισχύσουμε την ασφάλεια του συστήματός μας. Αυτό το επίτευγμα ανοίγει τον δρόμο για την περαιτέρω ανάπτυξη της εφαρμογής μας, προσφέροντας ένα αξιόπιστο και ασφαλές περιβάλλον για τους διαχειριστές της εφαρμογής.

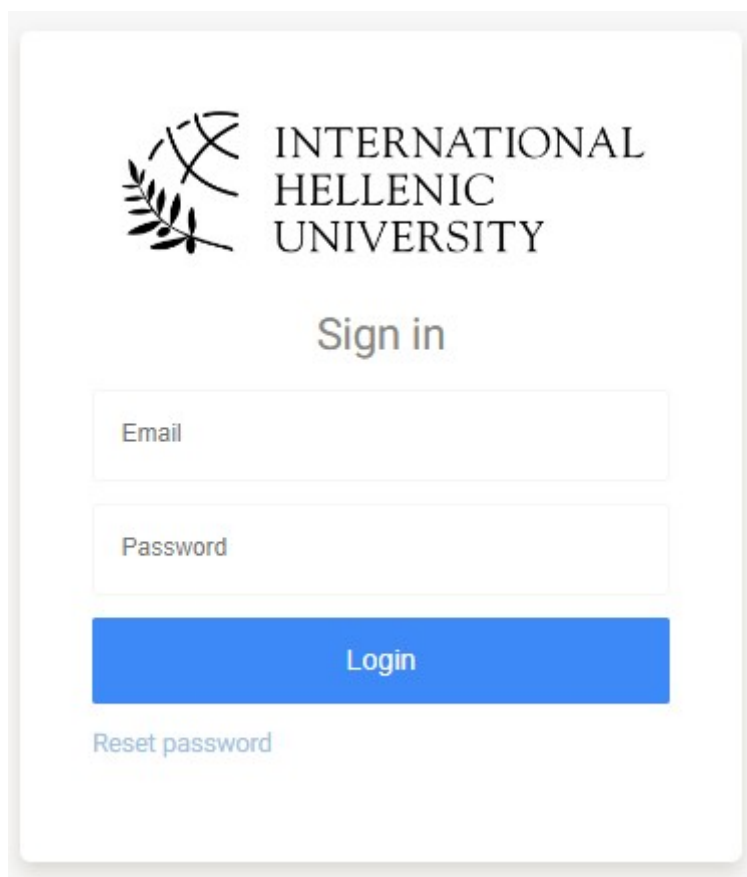


Figure 10: Διεπαφή αυθεντικοποίησης εφαρμογής ιστού

```
async signIn(email: string, password: string): Promise<void> {
  try {
    await signInWithEmailAndPassword(this.auth, email, password);
    this.snackBar.snackPositionTopCenter('success', 'Συνδεθήκατε με επιτυχία!');
  } catch (error) {
    this.snackBar.snackPositionTopCenter('error', 'Δεν υπάρχει λογαριασμός με αυτά τα στοιχεία!');
    console.error(error);
  }
}

async signOut(): Promise<void> {
  try {
    await this.auth.signOut();
    this.snackBar.snackPositionTopCenter('success', 'Αποσυνδεθήκατε με επιτυχία!');
  } catch (error) {
    this.snackBar.snackPositionTopCenter('error', 'Κάτι πήγε στραβά!');
  }
}
```

Figure 11: Συναρτήσεις σύνδεσης και αποσύνδεσης με χρήση της AngularFire

3.3.4 Υλοποίηση της κύριας γραφικής διεπαφής χρήστη

Έχοντας ολοκληρώσει την υλοποίηση των λειτουργιών ταυτοποίησης, κατευθυνθήκαμε προς τον σχεδιασμό της γραφικής διεπαφής του χρήστη (UI), βασισμένοι στην ιδέα της εξατομικευμένης και ασφαλούς εμπειρίας.

Η γραφική διεπαφή ελέγχεται από έναν μηχανισμό του πλαισίου Angular γνωστό ως φύλακα ταυτοποίησης (Auth Guard). Σκοπός αυτού είναι η εξασφάλιση πως μόνο συνδεδεμένοι χρήστες

μπορούν να αποκτήσουν πρόσβαση στην πλήρη λειτουργικότητα της εφαρμογής. Επιπλέον, για να επιτευχθεί μια επιπλέον στρώση ασφαλείας, προσθέσαμε έναν μηχανισμό ελέγχου ο οποίος ελέγχει εάν υπάρχει μία καταγραφή στην βάση, με την μορφή ενός αρχείου στην συλλογή “admins” με το μοναδικό αναγνωριστικό του χρήστη (id). Αυτό διασφαλίζει πως ο χρήστης ο οποίος ζητά πρόσβαση στην συγκεκριμένη όψη της γραφικής διεπαφής είναι όντως ένας από τους διαχειριστές της εφαρμογής.

Στον σχεδιασμό της γραφικής διεπαφής, περιλαμβάνεται ένα αριστερό μενού πλοήγησης (side navigation/sidenav) για γρήγορη πλοήγηση, καθώς και μία κεφαλίδα (header) στα δεξιά που περιλαμβάνει χρήσιμες πληροφορίες και επιλογές. Κάτω από αυτά, το κύριο περιεχόμενο αντιπροσωπεύεται από ένα component το οποίο αλλάζει δυναμικά ανάλογα με το επιλεγμένο δρομολόγιο (route), προσφέροντας ένα προσαρμοσμένο περιβάλλον χρήστη.

3.3.5 Αρχικοποίηση των συστατικών (Angular Components) και των δρομολογίων (Angular Routes)

Κατά το επόμενο κρίσιμο βήμα της ανάπτυξης, προχωρήσαμε στην αρχικοποίηση όλων των συστατικών (components) τα οποία θα αντιπροσωπεύονται στο αριστερό μενού πλοήγησης (sidenav), δημιουργώντας έτσι ένα συνεκτικό και λειτουργικό μενού πλοήγησης. Η διαδικασία αυτή συνέβαλε στη διαμόρφωση μιας πλήρους εικόνας της εφαρμογής, παρέχοντας τη δυνατότητα πρόσβασης σε κάθε σημαντική λειτουργία.

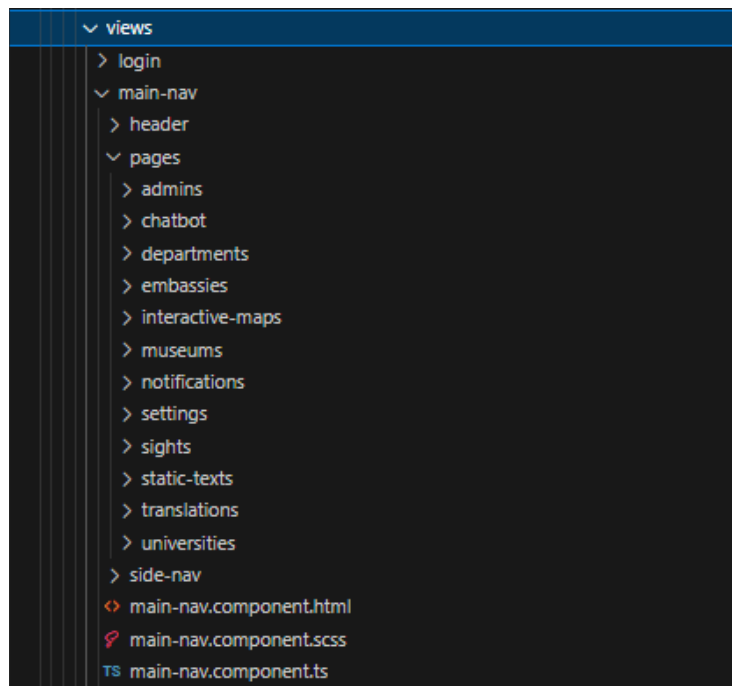


Figure 12: Δομή αρχείων συστήματος σύμφωνα με την λογική των components

Συγχρόνως, δημιουργήσαμε τα αντίστοιχα routes για κάθε component, επιτρέποντας την εύκολη πλοήγηση και αλλαγή μεταξύ των σελίδων της εφαρμογής. Αυτή η ανάπτυξη των routes βοήθησε στο να αξιοποιηθεί πλήρως ο δρομολογητής Angular (Angular Router), προσφέροντας παράλληλα μια ενιαία δομή για την οργάνωση του κώδικα και τη διαχείριση των σελίδων.

Κάθε component αντιστοιχεί σε μια συγκεκριμένη λειτουργία ή οθόνη της εφαρμογής, ενώ τα routes διευκολύνουν την πλοήγηση ανάμεσα σε αυτά με έναν ευέλικτο και ευανάγνωστο τρόπο. Η αρχικοποίηση αυτή συνεισέφερε σημαντικά στη συνολική συνεπή και ευχάριστη εμπειρία των χρηστών, ενώ παράλληλα δημιούργησε την απαραίτητη βάση για την περαιτέρω ανάπτυξη και επέκταση της εφαρμογής μας.

3.3.6 Υλοποίηση της καρτέλας "Ρυθμίσεις"

Μία από τις πρώτες καρτέλες που αναπτύξαμε ήταν οι "Ρυθμίσεις", η οποία αποτελεί το κρισιμότερο κομμάτι για την παραμετροποίηση της εφαρμογής. Στο πλαίσιο αυτό, επικεντρωθήκαμε στη δημιουργία μιας ολοκληρωμένης λειτουργικότητας παραμετροποίησης, της οποίας η χρήση θα επιτρέπει στους τελικούς χρήστες να προσαρμόζουν την εφαρμογή και το περιεχόμενο της σύμφωνα με τις προτιμήσεις και τις ανάγκες τους.

Στην καρτέλα "Ρυθμίσεις", προσθέσαμε τρεις βασικές λειτουργίες που καθιστούν την εφαρμογή μας πραγματικά προσαρμόσιμη.

1. Ενσωματώσαμε τη δυνατότητα προσθήκης και αφαίρεσης γλώσσας από την εφαρμογή με την χρήση μιας βιβλιοθήκης βασισμένη στο διεθνές πρότυπο i18n. Αυτή η λειτουργία είναι ιδιαίτερα κρίσιμη για τους φοιτητές Erasmus+, καθώς τους επιτρέπει να προσαρμόζουν το περιεχόμενο σύμφωνα με τις γλώσσες που γνωρίζουν.

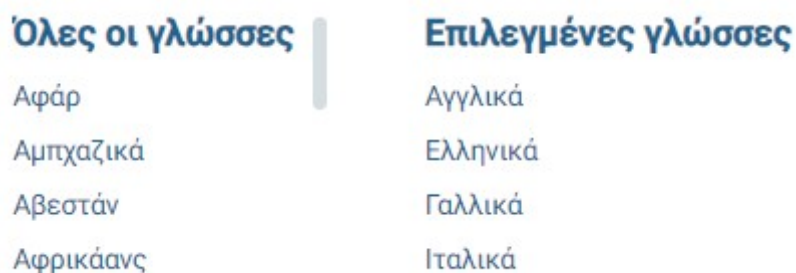


Figure 13: Μενού προσθαφαίρεσης γλωσσών συστήματος

2. Ενσωματώσαμε τη δυνατότητα προσαρμογής των διαθέσιμων πόλεων στην εφαρμογή. Αυτό επιτρέπει στους χρήστες να επιλέγουν την πόλη που τους ενδιαφέρει, προσφέροντας μια προσαρμοσμένη εμπειρία βασισμένη στην γεωγραφική τους θέση.



Figure 14: Μενού επεξεργασίας πόλεων

3. Ενσωματώσαμε τη δυνατότητα διαχείρισης κατηγοριών, για την κατάλληλη απόδοσης αυτών σε πόρους και κείμενα, με στόχο την κατηγοριοποίηση αυτών ώστε να βελτιώσουμε την προσπέλαση πληροφοριών κατά την αλληλεπίδραση των χρηστών με τις εφαρμογές.



Figure 15: Μενού επεξεργασίας κατηγοριών πόρων/κειμένων

3.3.7 Υλοποίηση της καρτέλας "Μεταφράσεις"

Ακολουθώντας την ανάπτυξη της καρτέλας "Ρυθμίσεις" και την προσθήκη της δυνατότητας επιλογής γλώσσας, το φυσικό ακόλουθο ήταν η ανάπτυξη μιας διεπαφής για την καρτέλα "Μεταφράσεις". Βασισμένο στις γλώσσες που καθορίζονται στην προαναφερθείσα καρτέλα "Ρυθμίσεις", η καρτέλα "Μεταφράσεις" προσφέρει τη δυνατότητα προσθήκης στατικών κειμένων, όπως και λέξεις κλειδιών που χρησιμοποιούν οι εφαρμογές στα στοιχεία γραφικών διεπαφών σε όλες τις γλώσσες που υποστηρίζονται. Αυτό το χαρακτηριστικό αποδεικνύεται ιδιαίτερα σημαντικό για μια εφαρμογή που αναφέρεται σε χρήστες με διαφορετικές γλωσσικές προτιμήσεις.

Ο χρήστης έχει τη δυνατότητα να προσθέσει και να επεξεργαστεί κείμενα για κάθε σελίδα της εφαρμογής τελικών χρηστών σε πολλές γλώσσες, βοηθώντας έτσι στη διαμόρφωση μιας πραγματικά πολυγλωσσικής εμπειρίας. Αυτό το χαρακτηριστικό παρέχει στους χρήστες τη δυνατότητα να αλληλεπιδρούν με την εφαρμογή στη γλώσσα της επιλογής τους, βελτιώνοντας σημαντικά την προσωπική τους εμπειρία.

key	Ελληνικά	Ιταλικά	Αγγλικά	Γαλλικά
tap	Πατήστε σε ένα κτίριο στο χάρτη για πληροφορίες	Tocca un edificio sulla mappa per ottenere informazioni	Tap on a building on the map for information	Appuyez sur un bâtiment sur la carte pour obtenir des informations
building_description	Περιγραφή κτιρίου	Descrizione dell'edificio	Building description	Descriptif du bâtiment
no_data	Δεν βρέθηκαν δεδομένα για αυτήν την πανεπιστημιούπολη.	Nessun dato trovato per questo campus.	No data found for this campus.	Aucune donnée trouvée pour ce campus.
building_name	Όνομα κτιρίου	Nome immobile	Building name	Nom du bâtiment
office	Αίθουσες/Γραφεία	Sale/Uffici	Halls/Offices	Halls/Bureaux

Figure 16: Διεπαφή μεταγλώττισης κειμένων

3.3.8 Υλοποίηση της καρτέλας "Διαχειριστές"

Συνεχίσαμε την ανάπτυξη με τη δημιουργία της καρτέλας "Διαχειριστές", ένα εξίσου σημαντικό κομμάτι της εφαρμογής μας που επικεντρώνεται στον έλεγχο της πρόσβασης και τη διαχείριση των διαχειριστών.

Σε αυτή την διεπαφή δώσαμε αποκλειστικό δικαίωμα στον υπερ-διαχειριστή να δημιουργεί νέους διαχειριστές και να διαγράφει υπάρχοντες.

Σημειώνουμε ότι οι υπόλοιποι χρήστες δεν έχουν τη δυνατότητα να εκτελέσουν αυτές τις ενέργειες, προσφέροντας έτσι ένα επίπεδο περαιτέρω προστασίας και ελέγχου στον τρόπο πρόσβασης στην εφαρμογή.

3.3.9 Υλοποίηση των καρτελών "Πανεπιστημιούπολεις, Στατικά Κείμενα, Πρεσβείες, Μουσεία, Αξιοθέατα"

Στην συνέχεια, προχωρήσαμε στην υλοποίηση των καρτελών "Πανεπιστημιούπολεις", "Στατικά Κείμενα", "Πρεσβείες", "Μουσεία", και "Αξιοθέατα", τα οποία αποτελούν σημαντικά κομμάτια της εφαρμογής μας. Όλα σχεδιάστηκαν με ένα κοινό μοτίβο, προσφέροντας συνολικά και συνεκτικά δεδομένα για τους χρήστες και μία μορφή δεδομένων η οποία μπορεί να αξιοποιηθεί από τον ψηφιακό βοηθό [εδώ].

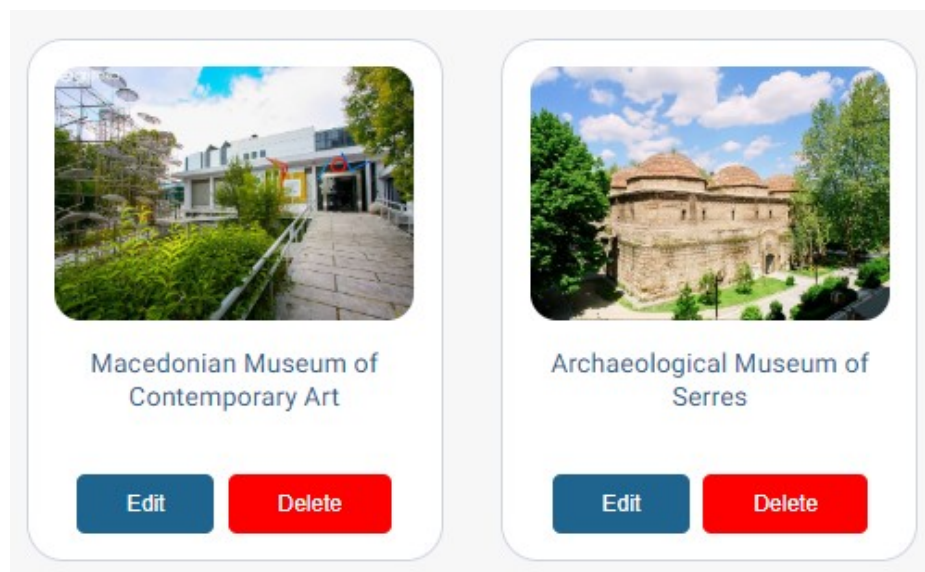


Figure 17: Γραφική διεπαφή επισκόπησης μουσείων

Κατά τον σχεδιασμό των περιεχομένων τους, ορίσαμε συγκεκριμένα κοινά στοιχεία όπως το όνομα και η περιγραφή, τα οποία είναι διαθέσιμα σε όλες τις γλώσσες οι οποίες έχουν προστεθεί στις ρυθμίσεις. Αυτό εξασφαλίζει ότι οι πληροφορίες είναι προσβάσιμες και κατανοητές για κάθε χρήστη, ανεξάρτητα από τη γλώσσα προτίμησής του.

Επιπλέον, στην δομή κάθε οντότητας συμπεριλάβαμε πεδία για έναν υπερσύνδεσμο και φωτογραφίες, επιτρέποντας στους χρήστες να αποκτούν εκτενείς και πολυπολιτισμικές πληροφορίες για κάθε κατηγορία. Επιπλέον, η προσθήκη της πόλης ως στοιχείου εμπλουτίζει την εμπειρία του χρήστη, προσφέροντας πληροφορίες που είναι τοπικά προσαρμοσμένες.

Με την υλοποίηση αυτών των διεπαφών, παρέχουμε στους χρήστες εκτενείς και ποικίλες πληροφορίες για τις πόλεις και τις δραστηριότητες που μπορούν να εξερευνήσουν, αποτελώντας έτσι μια βάση για μία από τις σημαντικότερες επεκτάσεις, την υλοποίηση του ψηφιακού βοηθού (Chatbot).

3.3.10 Υλοποίηση της λειτουργικότητας Διαδραστικών χαρτών

Μία από τις πρωταρχικές ενέργειες ενός νέου φοιτητή σε ένα πανεπιστημιακό ίδρυμα είναι η εξοικείωση αυτού με τις εγκαταστάσεις του χώρου του ιδρύματος. Για την επίτευξη του στόχου αυτού, με κύρια κριτήρια την ευχρηστία και ταχύτητα εύρεσης πληροφοριών, υλοποιήσαμε τους διαδραστικούς χάρτες πανεπιστημιούπων.

3.3.10.1 Προσθήκη φωτογραφίας ως βάση διαδραστικού χάρτη

Οποιοσδήποτε χρήστης διαθέτει πρόσβαση στην διαχειριστική εφαρμογή ιστού μπορεί να επεξεργαστεί τις οντότητες των πανεπιστημιούπων καθώς και των τμημάτων αυτών. Στην καρτέλα “Διαδραστικοί χάρτες”, δίνεται η δυνατότητα ανάρτησης μιας φωτογραφίας η οποία θα λειτουργήσει ως η βάση του καμβά του διαδραστικού χάρτη, και εφόσον έχει καταχωρηθεί τουλάχιστον μία πανεπιστημιούπολη ο ορισμός της φωτογραφίας ως ο χάρτης της πανεπιστημιούπολης.

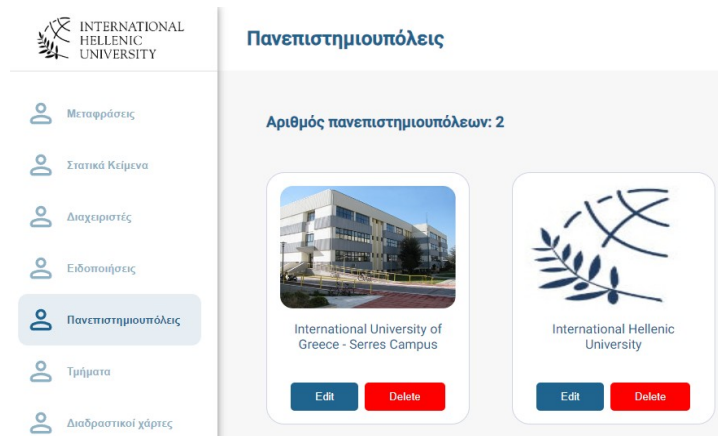


Figure 18: Οντότητες πανεπιστημιούπων

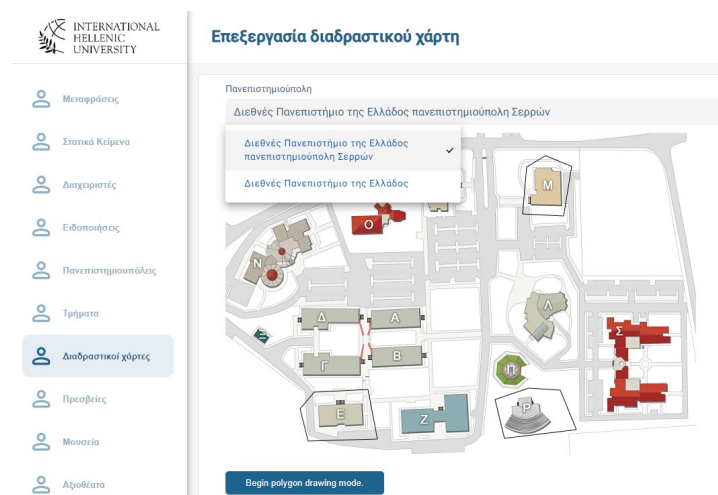


Figure 19: Σύνδεση πανεπιστημιούπολης με φωτογραφία-χάρτη

3.3.10.2 Ορισμός διαδραστικών περιοχών

Κατόπιν των παραπάνω, ο διαχειριστής μπορεί να ξεκινήσει τον σχεδιασμό πολύγωνων επί της φωτογραφίας-χάρτη, τα οποία θα ορίζουν τα όρια της εκάστοτε περιοχής. Για τον σκοπό αυτό τοποθετήσαμε ένα στοιχείο καμβά HTML (canvas element) το οποίο επικαλύπτει απόλυτα την φωτογραφία. Με αυτό τον τρόπο οι συντεταγμένες που ορίζουν τα πολύγωνα πάνω στον καμβά αντιστοιχίζονται με τα pixels της φωτογραφίας.

```
<div id="map_div relative">
  <img #map_image [src]="imgSrc" alt="">
  <canvas class="absolute top-0 left-0" #map_overlay></canvas>
</div>
```

Figure 20: Επικάλυψη φωτογραφίας με στοιχείο καμβά

```
canvas.addEventListener('click', (event) => {
  const x = event.clientX - canvas.getBoundingClientRect().left;
  const y = event.clientY - canvas.getBoundingClientRect().top;
  if (this.isDrawing) {
    this.currentPolygon.push({ x, y });
  }
});
```

Figure 21: Καταγραφή σημείων πολυγώνου με χειριστή συμβάντων κλικ

Έχοντας πλέον ορίσει πολύγωνα, επιλέγοντας ένα σημείο στην φωτογραφία-χάρτη (με αριστερό κλικ), προσδιορίζεται εάν το σημείο βρίσκεται εντός κάποιου πολυγώνου με τον αλγόριθμο ray-casting, και εφόσον βρίσκεται εντός, επιτρέπονται οι επεξεργασίες των στοιχείων του κτιρίου όπως όνομα, περιγραφή, συστεγαζόμενες σχολές, αίθουσες/γραφεία και πληροφορίες αυτών σε οσοδήποτε γλώσσες έχουν οριστεί από τις Ρυθμίσεις της εφαρμογής όπως αναφέρθηκαν στο κεφάλαιο 3.3.6.

Όνομα κτιρίου	Γλώσσα	
Κτίριο P	en	
Περιγραφή κτιρίου	Γλώσσα	
test2	en	
1ο Γραφείο/Αίθουσα		
Αριθμός γραφείου/αίθουσας	Όνομα γραφείου/αίθουσας	Υπεύθυνος γραφείου/αίθουσας
100	Αμφιθέατρο	
2ο Γραφείο/Αίθουσα		
Αριθμός γραφείου/αίθουσας	Όνομα γραφείου/αίθουσας	Υπεύθυνος γραφείου/αίθουσας
101	Εργαστήριο Πληροφορικής	Αδαμίδης Παναγιώτης

Figure 22: Επεξεργασία πληροφοριών περιοχής πολυγώνου.

3.3.11 Υλοποίηση ψηφιακού βοηθού (Chatbot)

Για τους σκοπούς της αποδοτικής οργάνωσης του μεγάλου όγκου δεδομένων, της εύκολης προσπέλασης αυτών καθώς και εξοικονόμησης χώρου στις διεπαφές της εφαρμογής των τελικών χρηστών όπως και για την αποφυγή πλοήγησης σε διάφορες οθόνες ή προγράμματα περιηγητών, αναπτύξαμε έναν ψηφιακό βοηθό. Σκοπός αυτού είναι η ψευδοσυνομιλία με τους τελικούς χρήστες κάνοντας χρήση ενός δυναμικά καθορισμένου δενδροδιαλόγου ο οποίος μπορεί να οδηγήσει τον χρήστη στις πηγές που ψάχνει παραμένοντας στην ίδια γραφική διεπαφή.

Στην καρτέλα “Chatbot” της διαχειριστικής εφαρμογής ιστού, ο διαχειριστής μπορεί να δημιουργήσει μία σειρά από προτροπές διαλόγου και επιλογές του τελικού χρήστη, οι οποίες καταλήγουν στην δημιουργία ενός ερωτήματος (query) στην βάση δεδομένων το οποίο εμφανίζει τα κατάλληλα αποτελέσματα. Με αυτό τον τρόπο, ξεκινώντας από μία προκαθορισμένη προτροπή ρίζα, μπορούν να δημιουργηθούν πολλαπλά μονοπάτια διαλόγου τα οποία καταλήγουν σε διαφορετικά ερωτήματα με αποτέλεσμα να καλυφθούν όλες οι υπάρχουσες πληροφορίες που διαθέτει η βάση δεδομένων της εφαρμογής.

Κάθε προτροπή διαλόγου είναι ένα αρχείο της συλλογής “Chatbot”, στην βάση δεδομένων. Κάθε επιλογή προτροπής διαλόγου συνδέεται με ένα έγγραφο στην ίδια συλλογή με βάση το μοναδικό αναγνωριστικό του (nextDocumentId) στην ίδια συλλογή σαν μία δομή δεδομένων συνδεδεμένης λίστας (Linked List). Με εξαίρεση την ρίζα του διαλόγου, όλα τα έγγραφα διαθέτουν και δείκτη προς το προηγούμενο αρχείο διαλόγου με βάση το μοναδικό αναγνωριστικό του (previousDocumentId), σαν δομή διπλά συνδεδεμένης λίστας (Doubly Linked List). Έτσι, κάθε επιλογή επιφέρει την συνέχεια του διαλόγου μέσω του αναγνωριστικού του επομένου εγγράφου, και του προηγούμενου σημείου στον διάλογο με το αναγνωριστικό του προηγούμενου εγγράφου.

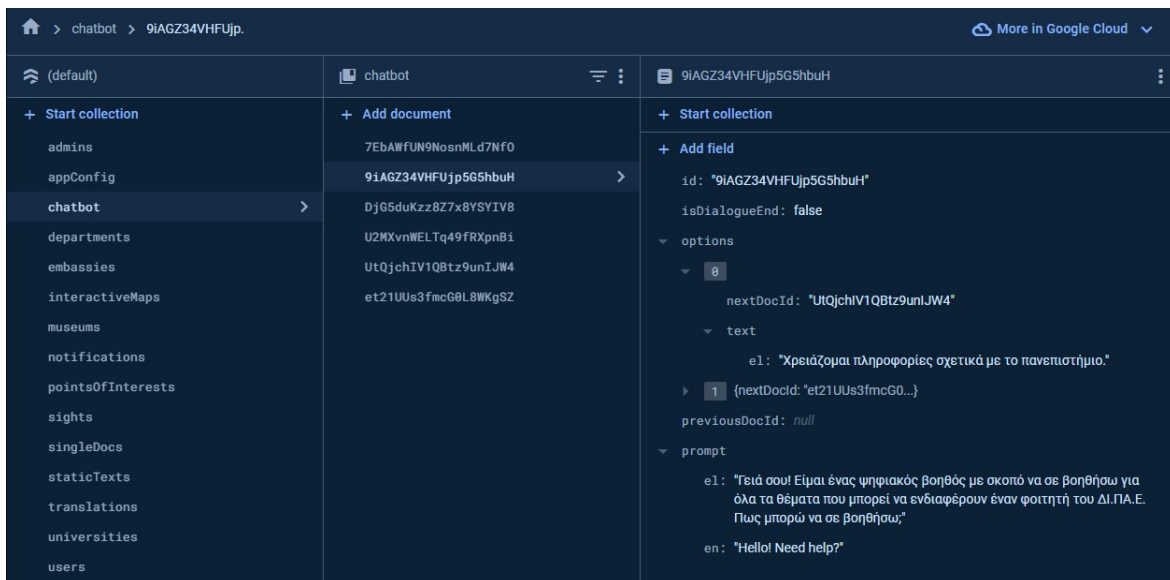


Figure 23: Δομή εγγράφου δεδομένων ψηφιακού βοηθού στην γραφική διεπαφή της σουίτας Firebase.

Όλες οι προτροπές του ψηφιακού βοηθού καθώς και οι επιλογές του τελικού χρήστη, έχουν την δυνατότητα να μεταφραστούν σε όλες τις γλώσσες που έχουν οριστεί από την διαχειριστική εφαρμογή, όπως αναφέρθηκε στο κεφάλαιο 3.3.6.

Επιπλέον, κάθε αρχείο διαθέτει ένα πεδίο σημαία (boolean), ονόματι “isDialogueEnd” το οποίο χρησιμοποιείται για να ενημερώσει την εφαρμογή πως ο διάλογος έχει φτάσει σε τερματικό κόμβο και πρέπει να εκτελεστεί ένα ερώτημα προς την βάση με τις παραμέτρους που ορίζονται στο πεδίο “result” και να τερματιστεί ο διάλογος. Τα πεδία του πεδίου “result” είναι τα εξής:

- Το όνομα συλλογής στην οποία θα εκτελεστεί το ερώτημα.
- Μία σημαία για το αν τα αποτελέσματα θα περιοριστούν σε πόρους/πληροφορίες που υπάρχουν σε μία πόλη.
- Μία αντίστοιχη σημαία για πανεπιστημιούπολη.
- Ένα πεδίο με τις υπόλοιπες, προαιρετικές παραμέτρους, όπως ονόματα πεδίων και τιμές που πρέπει να έχουν αυτά, στα έγγραφα στα οποία εκτελείται το ερώτημα, ώστε να περιοριστούν τα αποτελέσματα στα επιθυμητά.

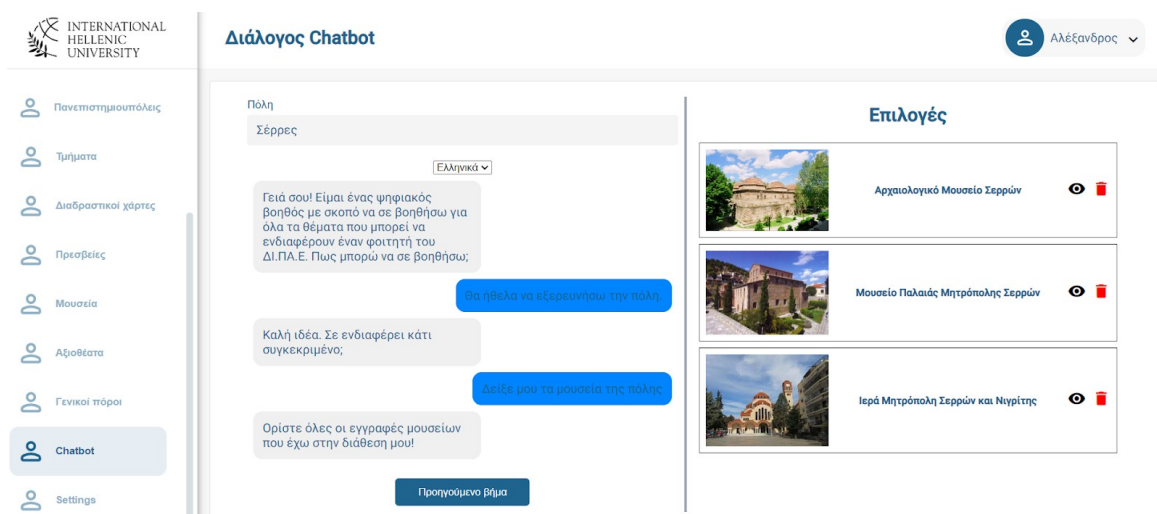


Figure 24: Αποτέλεσμα συνομιλίας με τον ψηφιακό βοηθό, με κατάληξη την προβολή μουσείων στην επιλεγμένη πόλη του χρήστη.

Για την επίτευξη της ομαλής προβολής αποτελεσμάτων ανεξαρτήτως των οντοτήτων οι οποίες προβάλλονται, η δομή της βάσης σχεδιάστηκε με τέτοιο τρόπο ώστε όλες να μοιράζονται συγκεκριμένα πεδία τα οποία θεωρούμε απαραίτητα για την ελάχιστη επεξήγηση των αποτελεσμάτων καθώς και κάποια ακόμα προαιρετικά, τα οποία εμφανίζονται μόνο υπό τις κατάλληλες συνθήκες και εφόσον δικαιολογείται η χρήση τους.

3.4 Εφαρμογή κινητών συσκευών

Σκοπός της εφαρμογής χρηστών είναι ο προσανατολισμός και η ενημέρωση των φοιτητών για την δομή, τις υπηρεσίες και τους πόρους υποστήριξης φοιτητών του ΔΙ.ΠΑ.Ε, η διαφάνεια των φοιτητικών τους υποχρεώσεων, καθώς και η εξερεύνηση της πόλης.

3.4.1 Δυνατότητες χρήστη

Μέσα από την εφαρμογή χρηστών, οι φοιτητές μπορούν να:

- Μάθουν για το πανεπιστήμιο και το σπουδαστικό πλαίσιο μέσω των στατικών κειμένων που ορίζει ο διαχειριστής.
- Γνωρίσουν την πανεπιστημιούπολη μέσω διαδραστικού χάρτη.

- Συνομιλήσουν με Chatbot για επίλυση αποριών ή ταχύτερη αναζήτηση πόρων.
- Παρακολουθούν πίνακα ανακοινώσεων με αποκλειστικό περιεχόμενο για την εφαρμογή.

3.4.2 Περιβάλλον ανάπτυξης

Ξεκινώντας την ανάπτυξη της εφαρμογής τελικών χρηστών με το πλαίσιο Flutter, δημιουργήσαμε το αποθετήριο (repository) στο GitHub για τους λόγους που προαναφέρθηκαν στο κεφάλαιο 3.3.2.1. Τα πρώτα βήματα περιλάμβαναν την εγκατάσταση του πλαισίου Flutter και την λήψη του λογισμικού Android Studio. Δύο απαραίτητα εργαλεία για την ανάπτυξη και εκτέλεση της εφαρμογής σε εξομοιωτή ή φυσική συσκευή.

Η εγκατάσταση του πλαισίου Flutter παρέχει το περιβάλλον ανάπτυξης που χρειάζεται για τη δημιουργία εφαρμογών με αυτό, ενώ το πρόγραμμα Android Studio είναι ένα ισχυρό εργαλείο ανάπτυξης εφαρμογών Android.

3.4.3 Ταυτοποίηση χρήστη

Πρώτο βήμα ήταν η σελίδα ταυτοποίησης, παρέχοντας στους χρήστες έναν εύχρηστο τρόπο να διαχειριστούν την εγγραφή και είσοδο τους στην εφαρμογή. Η σελίδα αυτή διασφαλίζει ότι ο χρήστης είναι εγγεγραμμένος και έχει πρόσβαση σε εξατομικευμένες ρυθμίσεις.

Στη διαδικασία εισόδου, ο χρήστης παρέχει τα στοιχεία του, όπως την διεύθυνση ηλεκτρονικού ταχυδρομείου του (email) και τον κωδικό πρόσβασης, και εκτελεί είσοδο μέσω της υπηρεσίας ταυτοποίησης της σουίτας Firebase. Αν ο χρήστης δεν έχει ήδη λογαριασμό, του δίνεται η δυνατότητα να δημιουργήσει έναν νέο.

Αυτό σημαίνει ότι η βασική λειτουργία της εφαρμογής εξαρτάται από τη συλλογή των πληροφοριών που ο χρήστης εισάγει κατά τη διαδικασία εγγραφής. Εκείνος επιλέγει την πόλη, την πανεπιστημιούπολη και τη γλώσσα προτίμησής του. Αυτά τα στοιχεία αποθηκεύονται σε ένα έγγραφο (document) στη συλλογή (collection) "users" στη βάση δεδομένων Firestore της Firebase. Αυτή η προσαρμοσμένη αποθήκευση επιτρέπει στην εφαρμογή να παρέχει προσαρμοσμένες εμπειρίες στον χρήστη, λαμβάνοντας υπόψη τις προτιμήσεις του για την πόλη, την πανεπιστημιούπολη και τη γλώσσα.

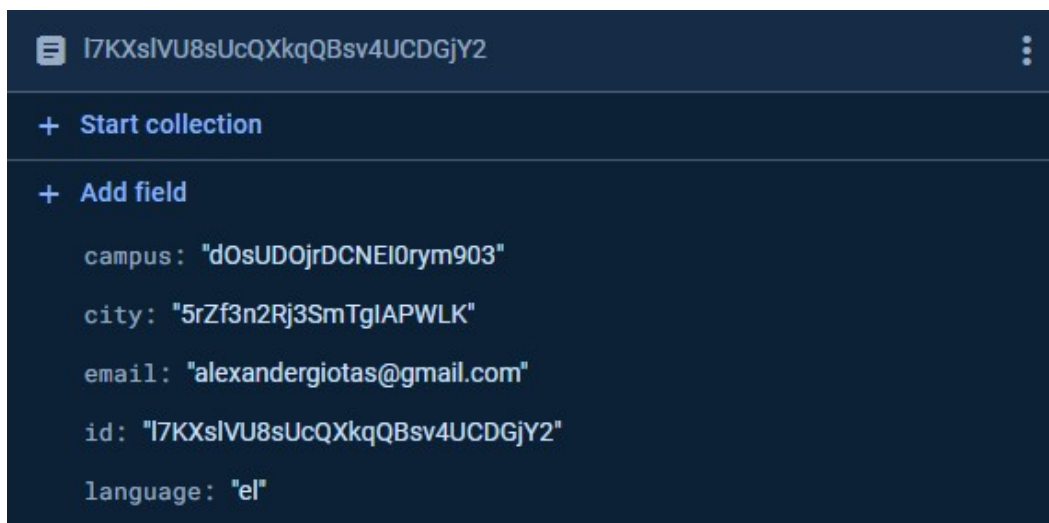


Figure 25: Έγγραφο χρήστη της συλλογής Users στην γραφική διεπαφή της Firebase.

3.4.4 Υλοποίηση κύριας γραφικής διεπαφής χρήστη

Στη συνέχεια, δομήσαμε την κύρια διεπαφή χρήστη, λαμβάνοντας υπόψη τις διεπαφές που θα περιλαμβάνει, και τον τρόπο οργάνωσης της δομής της. Κάθε διεπαφή σχεδιάστηκε να περιλαμβάνει μία κεφαλίδα (header ή appBar στην ορολογία του πλαισίου Flutter), η οποία αναφέρει το όνομά της καθώς και ένα κύριο περιεχόμενο (body) το οποίο θα αλλάζει δυναμικά ανάλογα με την επιλεγμένη διεπαφή.

Για την ευκολία της πλοήγησης, προσθέσαμε ένα υποσέλιδο (footer ή bottomNavigationBar στην ορολογία του πλαισίου Flutter) που επιτρέπει στον χρήστη να επιλέξει εύκολα την επιθυμητή διεπαφή. Αυτή η διάταξη εξασφαλίζει ένα συνεκτικό και ευανάγνωστο περιβάλλον για τους χρήστες, επιτρέποντάς τους να αλληλεπιδρούν εύκολα με την εφαρμογή και να πλοηγούνται απροβλημάτιστα ανάμεσα στις διάφορες λειτουργίες της.

Πιο συγκεκριμένα το υποσέλιδο περιλαμβάνει τις ακόλουθες επιλογές για να διευκολύνει την πλοήγηση του χρήστη:

1) Αρχική Διεπαφή (Home Page):

Στην αρχική διεπαφή, ο χρήστης μπορεί να βρει γενικές πληροφορίες για την πανεπιστημιούπολη που έχει επιλέξει, προσφέροντας μια συνολική εικόνα του περιβάλλοντος του.

2) Διαδραστικός Χάρτης Πανεπιστημιούπολης:

Ο χρήστης έχει πρόσβαση σε έναν διαδραστικό χάρτη της πανεπιστημιούπολης που έχει επιλέξει, επιτρέποντάς του να εξερευνήσει την περιοχή με ευκολία.

3) Ανακοινώσεις:

Εδώ, ο χρήστης μπορεί να προβάλει ανακοινώσεις που έχουν δημιουργηθεί και αποσταλεί από διαχειριστές, παρέχοντας ενημερώσεις για σημαντικά γεγονότα.

4) Ψηφιακός Βοηθός (Chatbot):

Ο χρήστης έχει πρόσβαση σε έναν ψηφιακό βοηθό (Chatbot), ο οποίος προσφέρει εύκολη προσπέλαση σε διάφορες πληροφορίες που μπορεί να τον ενδιαφέρουν.

5) Ρυθμίσεις (Settings):

Στις ρυθμίσεις, ο χρήστης μπορεί να προσαρμόσει τις προτιμήσεις του, συμπεριλαμβανομένης της αλλαγής της γλώσσας της εφαρμογής, της πόλης και της πανεπιστημιούπολης για τις οποίες ενδιαφέρεται.

Με αυτόν τον τρόπο, ο χρήστης έχει άμεση πρόσβαση στις σημαντικές λειτουργίες της εφαρμογής, ενισχύοντας την ευκολία περιήγησης του και την προσαρμοστικότητα της εφαρμογής στις προτιμήσεις του.



Figure 26: Υποσέλιδο (bottom navigation bar) εφαρμογής τελικών χρηστών.

3.4.5 Αρχική διεπαφή

Στην διεπαφή εκκίνησης (launch page), παρουσιάζονται πληροφορίες που έχουν συμπληρωθεί από τους διαχειριστές της εφαρμογής ιστού. Αυτές οι πληροφορίες αφορούν την πανεπιστημιούπολη που έχει επιλέξει ο χρήστης κατά την εγγραφή του στην εφαρμογή τελικών χρηστών. Έτσι, στην αρχική διεπαφή, ο χρήστης λαμβάνει σημαντικές πληροφορίες που αφορούν το περιβάλλον της πανεπιστημιούπολης που έχει επιλέξει.

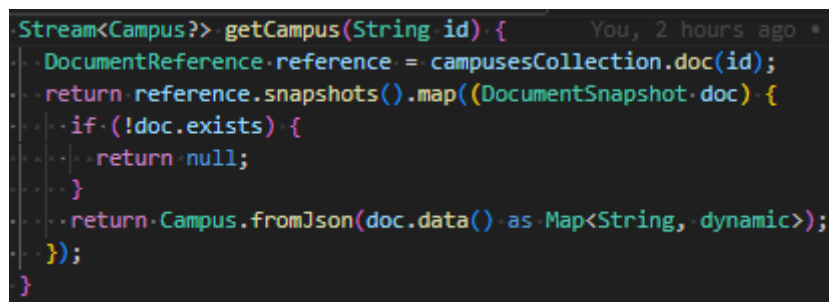
Στην διεπαφή εκκίνησης (launch page), ο χρήστης μπορεί να ανακτήσει σημαντικές πληροφορίες για την επιλεγμένη πανεπιστημιούπολη. Παρέχονται τα εξής στοιχεία:

- Όνομα Πανεπιστημιούπολης: Το όνομα της πανεπιστημιούπολης που έχει επιλέξει ο χρήστης.
- Περιγραφή: Μια συνοπτική περιγραφή του περιβάλλοντος και των επιλογών που προσφέρει η συγκεκριμένη πανεπιστημιούπολη.
- Εικόνα: Μια αντιπροσωπευτική εικόνα που απεικονίζει την πανεπιστημιούπολη.
- Email: Η ηλεκτρονική διεύθυνση επικοινωνίας με την πανεπιστημιούπολη.
- Τηλέφωνο: Ο αριθμός τηλεφώνου για περαιτέρω επικοινωνία.
- Link: Ένας υπερσύνδεσμος που κατευθύνει τον χρήστη στον ιστότοπο της πανεπιστημιούπολης.

Τα παραπάνω υλοποιήθηκαν με τους εξής τρόπους:

1. Λήψη Πληροφοριών μέσω Stream:

1. Η μέθοδος `getCampus` από την υπηρεσία `campusesService` επιστρέφει ένα ρεύμα δεδομένων (stream) αντικειμένου `Campus`.
2. Ο χρήστης έχει επιλέξει μια συγκεκριμένη πανεπιστημιούπολη, και οι πληροφορίες ανανεώνονται δυναμικά όταν υπάρχουν αλλαγές.



```

Stream<Campus?> getCampus(String id) {
    DocumentReference reference = campusesCollection.doc(id);
    return reference.snapshots().map((DocumentSnapshot doc) {
        if (!doc.exists) {
            return null;
        }
        return Campus.fromJson(doc.data() as Map<String, dynamic>);
    });
}

```

Figure 27: Μέθοδος παραγωγής ρεύματος δεδομένων.

2. Εμφάνιση Στοιχείων στο UI:

1. Οι πληροφορίες της πανεπιστημιούπολης, όπως το όνομά της, η εικόνα, το link, το email, και το τηλέφωνο, εμφανίζονται σε σχετικά πεδία του στοιχείου γραφικής διεπαφής (widget).
2. Χρησιμοποιήθηκε ένας δομητής ρεύματος δεδομένων (StreamBuilder) για την παρακολούθηση του ρεύματος της προαναφερθείσας μεθόδου (`getCampus`) και την ανανέωση της γραφικής διεπαφής μόλις ανιχνευθούν αλλαγές στα δεδομένα.

```

Widget build(BuildContext context) {
  final student = Provider.of<Student?>(context);
  final allTranslations = Provider.of<List<TranslationPage?>>(context);
  final translation = allTranslations.firstWhere((tr) => tr?.id == 'launch');
  Campus? campus;
  return StreamBuilder<Campus?>(
    stream: campusesService.getCampus(student?.campus ?? ''),
    builder: (context, snapshot) {

```

Figure 28: Δομητής ρεύματος δεδομένων.

3. Δυναμικά Στοιχεία γραφικής διεπαφής:
 1. Η εικόνα της πανεπιστημιούπολης εμφανίζεται μόνο αν μία είναι διαθέσιμη.
 2. Τα δεδομένα όπως το link, το email και το τηλέφωνο παρουσιάζονται με δυναμικό τρόπο, με σύνδεσμο που είναι παρατηρητής για άνοιγμα συνδέσμου κατά τον πάτημα.

3.5 Διεπαφή διαδραστικού χάρτη

Όπως προαναφέρθηκε στο κεφάλαιο 3.3.10, εφόσον ο τελικός χρήστης έχει επιλέξει την πανεπιστημιούπολη στην οποία ανήκει το τμήμα σπουδών του από τις ρυθμίσεις χρήση της εφαρμογής κινητών συσκευών, μπορεί να επιλέξει το εικονίδιο του χάρτη ώστε να πλοηγηθεί στην διεπαφή του διαδραστικού χάρτη.

Η εφαρμογή εντοπίζει τον κατάλληλο χάρτη από την συλλογή “interactiveMaps” της βάσης δεδομένων αναζητώντας το αρχείο με τιμή το μοναδικό αλφαριθμητικό αναγνωριστικό της πανεπιστημιούπολης στο πεδίο “campus_id” όπως φαίνεται στην ακόλουθη εικόνα.

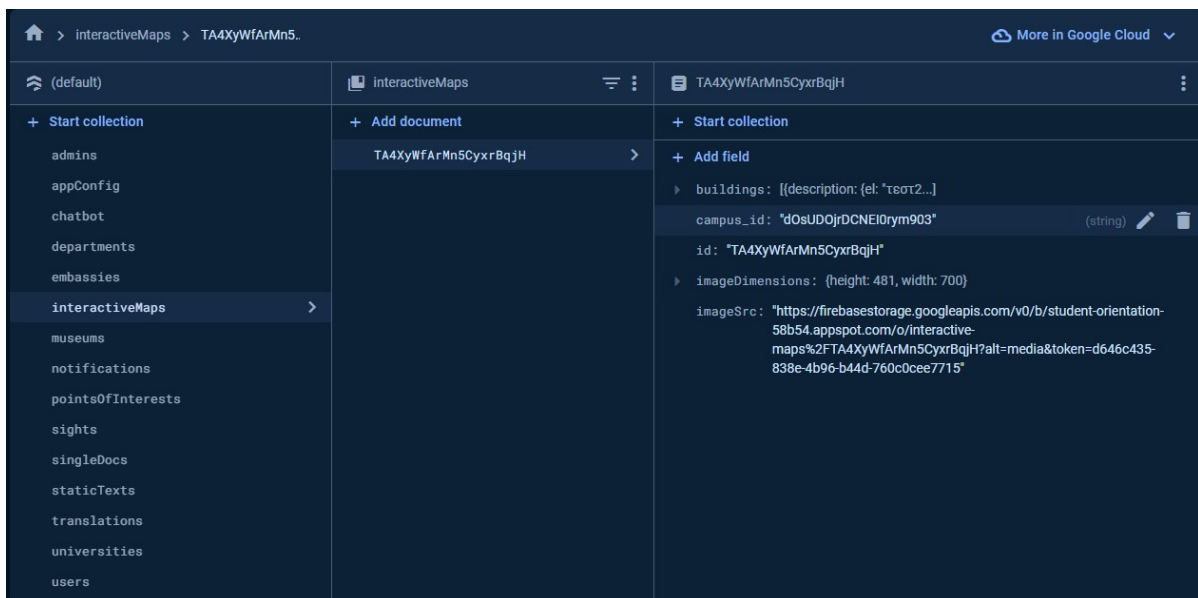


Figure 29: Έγγραφο δεδομένων διαδραστικού χάρτη.

Στην συνέχεια, μεταμορφώνει ασύγχρονα την εικόνα, η οποία βρίσκεται αποθηκευμένη στον αποθηκευτικό χώρο της εφαρμογής (Google Cloud Storage bucket), στην διεύθυνση (URL) που υποδεικνύει η τιμή του πεδίου “imageSrc”.

Το σύνολο της εικόνας επικαλύπτεται από ένα στοιχείο ανίχνευσης επαφής της γλώσσας Dart (GestureDetector), απαραίτητο για την εξαγωγή των συντεταγμένων του σημείου επαφής επί της οθόνης.

Αρχικά, η γραφική διεπαφή εμφανίζει την φωτογραφία του χάρτη και ένα μήνυμα οδηγίας χρήσης προς τον χρήστη.

Για την ορθή αποτύπωση των σημείων που ορίζουν τα πολύγωνα, ανεξάρτητα με το μέγεθος της οθόνης της εκάστοτε κινητής συσκευής, οι αρχικές διαστάσεις της εικόνας αποθηκεύονται στο έγγραφο του χάρτη, στην βάση δεδομένων, κατά την διαδικασία μεταμόρφωσης αυτής από την διαχειριστική εφαρμογή. Με αυτό τον τρόπο, μπορούμε να υπολογίσουμε το σχετικό σημείο επαφής επί της αρχικής εικόνας. Μόλις ο χρήστης πιάσει την οθόνη πυροδοτείται μία κλήση σε μέθοδο που εκτελεί τον αλγόριθμο Ray-Casting (όπως και στην διαχειριστική εφαρμογή), ώστε να προσδιοριστεί εν ο χρήστης επέλεξε σημείο το οποίο περιλαμβάνεται σε ένα ορισμένο πολύγωνο. Στην περίπτωση που το σημείο βρίσκεται εντός πολυγώνου, το μήνυμα προτροπής χρήστη αντικαθίσταται με τις πληροφορίες που έχουν οριστεί για αυτό το πολύγωνο.

```
child: GestureDetector(
  onTapDown: (details) {
    // Calculate tap position relative to image container
    RenderBox containerRenderBox =
      imageContainerKey.currentContext!.findRenderObject()
        as RenderBox;
    Offset localOffset = containerRenderBox
      .globalToLocal(details.globalPosition);
    // Calculate the equivalent tap position on the image
    double relativeX =
      localOffset.dx / containerRenderBox.size.width;
    double relativeY =
      localOffset.dy / containerRenderBox.size.height;
    tapX = (relativeX * map.imageDimensions.width).toInt();
    tapY = (relativeY * map.imageDimensions.height).toInt();
    // Use imageX and imageY as needed
    // Check if the tap position is inside any of the polygons
    for (final building in map.buildings) {
      if (pointIsInPolygon(tapX, tapY, building.mapPolygon)) {
        selectedBuilding = building;
        selectedBuildingNotifier.value = selectedBuilding;
        break;
      } else {
        selectedBuildingNotifier.value = null;
      }
    }
  },
),
```

Figure 30: Κώδικας μετατροπής απόλυτης θέσης αγγίγματος σε σχετικής με βάση τις διαστάσεις της εικόνας και έλεγχος έγκλεισης σημείου σε πολύγωνο.

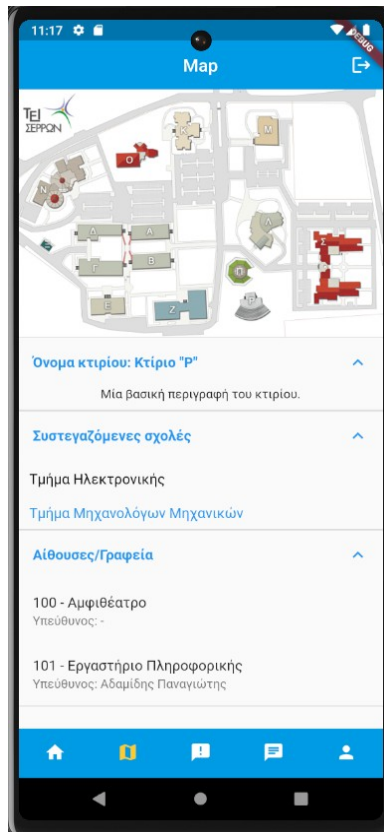


Figure 31: Γραφική διεπαφή κατόπιν επιλογής σημείου εντός ορισμένου πολυγώνου.

3.6 Ανακοινώσεις

Στη σελίδα των ανακοινώσεων, προβάλλονται όλες οι ανακοινώσεις που έχουν δημιουργηθεί από τους διαχειριστές μέσω της διαχειριστικής εφαρμογής ιστού. Για την ανάκτηση των ανακοινώσεων, χρησιμοποιήθηκε η μέθοδος παραγωγής ρεύματος δεδομένων `notifications()`.

```

final CollectionReference notificationsCollection =
    FirebaseFirestore.instance.collection('notifications');

Stream<List<AppNotification>> notifications() {
    return notificationsCollection.snapshots().map((QuerySnapshot query) {
        return query.docs.map((DocumentSnapshot doc) {
            return serializeAppNotification(doc);
        }).toList();
    });
}

```

Figure 32: Μέθοδος παραγωγής ρεύματος δεδομένων.

Κάθε ανακοίνωση περιλαμβάνει τα ακόλουθα στοιχεία:

- Δημιουργός: Ο χρήστης που δημιούργησε την ανακοίνωση από τη εφαρμογή ιστού.
- Ημερομηνία Δημιουργίας: Καταγράφεται η ημερομηνία που δημιουργήθηκε η ανακοίνωση.
- Τίτλος: Ο τίτλος της ειδοποίησης.

- Περιεχόμενο: Το περιεχόμενο της ειδοποίησης.

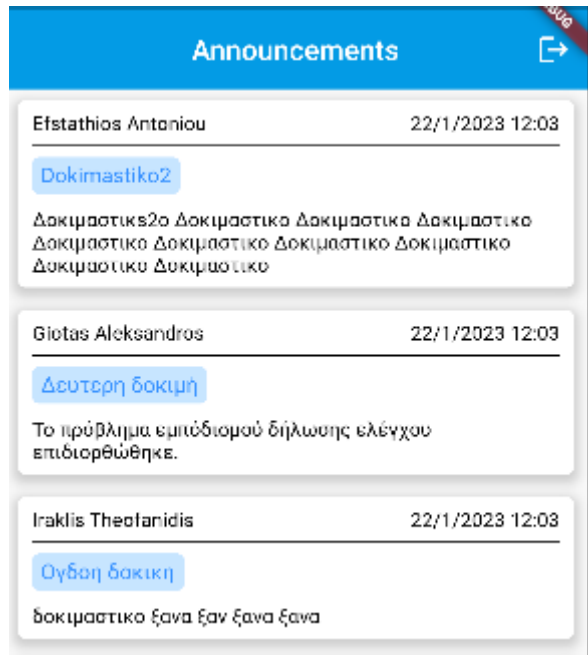


Figure 33: Γραφική διεπαφή ανακοινώσεων

Για την υλοποίηση της γραφικής διεπαφής στη σελίδα ανακοινώσεων, χρησιμοποιήσαμε το στοιχείο γραφικής διεπαφής `ListView.builder`. Αυτό επιτρέπει τη δημιουργία μίας δυναμικής λίστας στην γραφική διεπαφή, βασισμένο σε μια λίστα δεδομένων.

Στο εσωτερικό του στοιχείου `ListView.builder`, χρησιμοποιήσαμε ένα επαναχρησιμοποιήσιμο στοιχείο γραφικής διεπαφής με όνομα "Gesture" για κάθε ειδοποίηση. Αυτό καθιστά δυνατή τη δυναμική δημιουργία και ανανέωση των ειδοποιήσεων, επιτρέποντας την απευθείας αλληλεπίδραση του χρήστη με κάθε ειδοποίηση.

Το στοιχείο `ListView.builder` εξοικονομεί πόρους καθώς φορτώνει μόνο τα στοιχεία που είναι ορατά στην οθόνη, βελτιώνοντας έτσι την απόδοση της εφαρμογής. Με αυτόν τον τρόπο, δημιουργείται μια αποτελεσματική και απευθείας ανανεωμένη λίστα ειδοποιήσεων για τον χρήστη.

```
return Center(
  child: ListView.builder(
    itemCount: notifications.length,
    itemBuilder: (context, index) {
      return GestureDetector(

```

Figure 34: Στοιχείο `ListView.builder`

Με αυτόν τον τρόπο, ο χρήστης έχει πλήρη επισκόπηση όλων των ανακοινώσεων που έχουν δημιουργηθεί από τους διαχειριστές, παρέχοντας του σημαντικές ενημερώσεις σχετικά με τα γεγονότα και τις εξελίξεις στο περιβάλλον του πανεπιστημίου.

3.7 Chatbot

Το τελευταίο εικονίδιο στην μπάρα πλοήγησης της εφαρμογής κινητών συσκευών αντιστοιχεί στην διεπαφή του ψηφιακού βοηθού (Chatbot).

Την στιγμή που ο χρήστης πλοηγηθεί σε αυτήν πυροδοτείται μία ασύγχρονη κλήση στην βάση δεδομένων για την λήψη του εγγράφου που αντιστοιχεί στην ρίζα του διαλόγου. Μόλις ληφθούν τα δεδομένα δημιουργούνται 2 οριζόντια πλαίσια στην γραφική διεπαφή. Το πρώτο εμφανίζει το αρχικό μήνυμα προτροπής του ψηφιακού βοηθού, ενώ στο δεύτερο εμφανίζονται όλες οι πιθανές επιλογές του χρήστη για αυτό το επίπεδο του διαλόγου, όπως έχουν οριστεί προηγουμένως από την εφαρμογή διαχείρισης.

Εφόσον ο χρήστης επιλέξει μία από τις επιλογές διαλόγου, το κείμενο του διαλόγου προστίθεται στο πρώτο πλαίσιο, ώστε να φαίνεται καθαρά το ιστορικό συνομιλίας και πυροδοτείται μία νέα κλήση στην βάση δεδομένων για το έγγραφο με το αναγνωριστικό που αντιστοιχεί στην επιλογή που έκανε ο χρήστης.

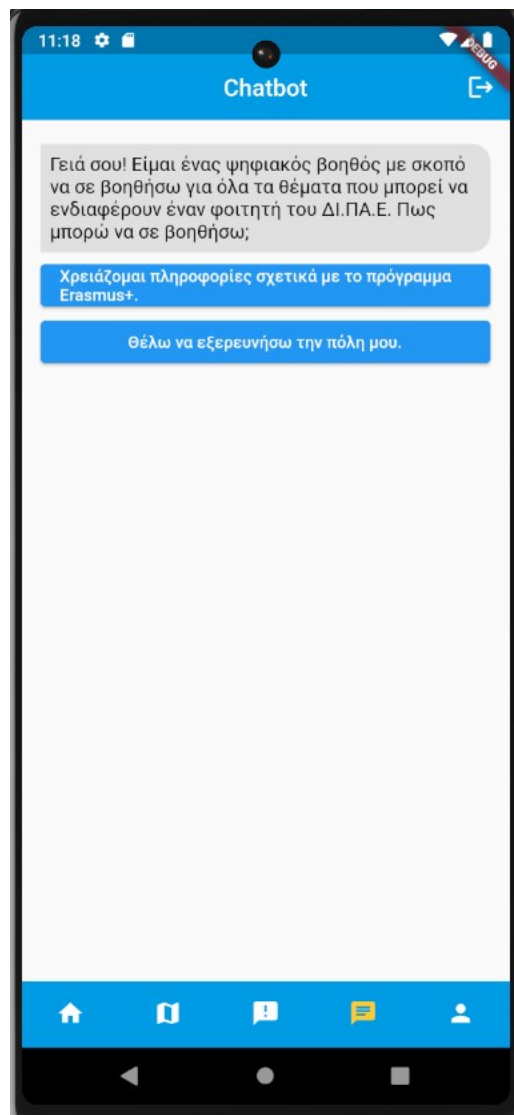


Figure 35: Αρχική κατάσταση διαλόγου.

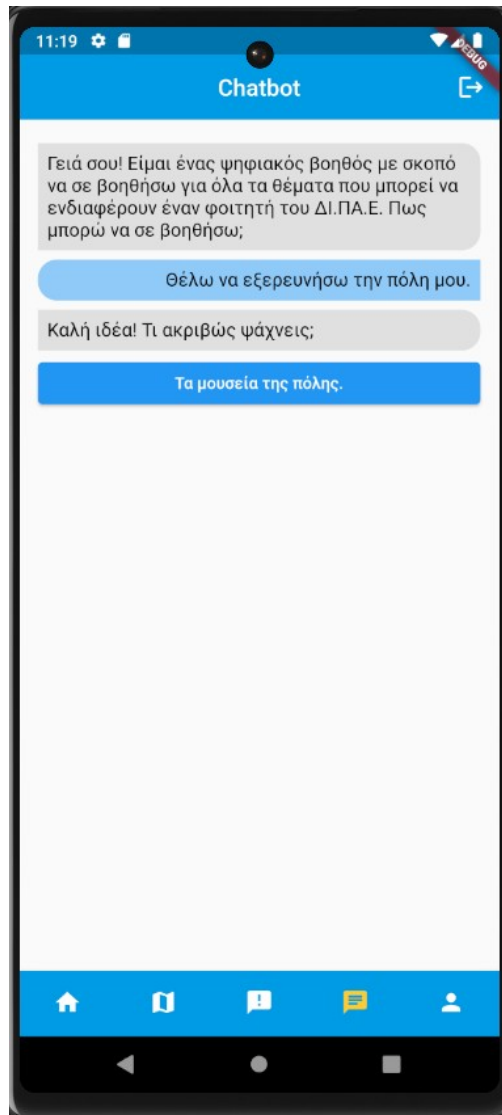


Figure 36: Εξέλιξη κατάστασης κατόπιν ενέργειας χρήστη.

Η παραπάνω διαδικασία επαναλαμβάνεται μέχρι ότου το έγγραφο που λαμβάνεται από την βάση έχει ορισμένο το πεδίο “isDialogueEnd” στην τιμή true. Σε αυτή την περίπτωση η εφαρμογή δημιουργεί ένα στοχευμένο ερώτημα προς την βάση δεδομένων χρησιμοποιώντας τις τιμές που βρίσκονται στο πεδίο “result” του εγγράφου και λαμβάνει τα θεμιτά έγγραφα ώστε να τα προβάλλει. Μόλις αυτά ληφθούν, η εφαρμογή αντικαθιστά το πλαίσιο των επιλογών χρήστη με ένα νέο πλαίσιο προβολής αποτελεσμάτων.

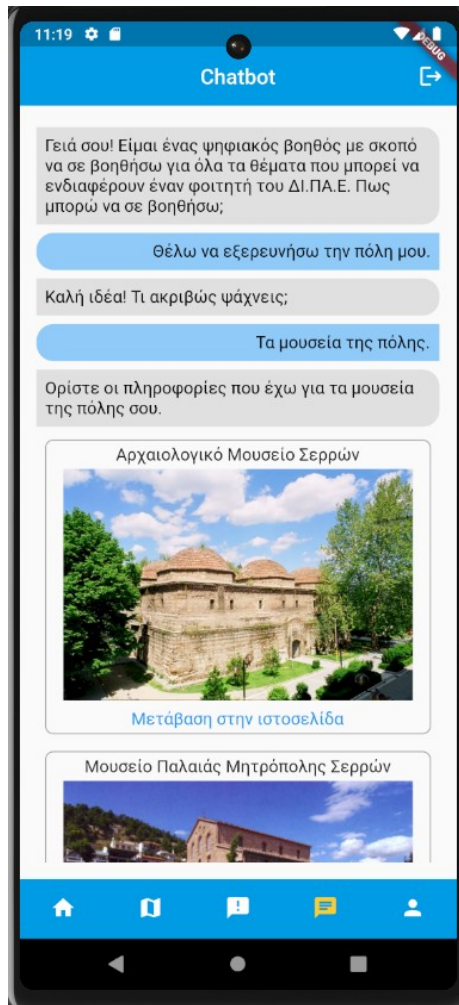


Figure 37: Αήξη διαλόγου και προβολή επιθυμητών αποτελεσμάτων.

Στην περίπτωση όπου οι τερματικοί κόμβοι της συνομιλίας έχουν οριστεί ώστε να χρησιμοποιείται η επιλεγμένη πανεπιστημιούπολη ή πόλη του χρήστη, λαμβάνεται υπόψη η τιμή που έχει ορίσει ο χρήστης στην διεπαφή “Ρυθμίσεις”, όπως αναφέρεται στο επόμενο κεφάλαιο.

3.8 Ρυθμίσεις

Στις ρυθμίσεις, ο χρήστης μπορεί να προσαρμόσει τις προτιμήσεις του, συμπεριλαμβανομένης της αλλαγής της γλώσσας της εφαρμογής, της πόλης και της πανεπιστημιούπολης που τον ενδιαφέρουν. Με αυτόν τον τρόπο, ο χρήστης έχει άμεση πρόσβαση στις σημαντικές λειτουργίες της εφαρμογής, ενισχύοντας την ευκολία περιήγησης του και την προσαρμοστικότητα της εφαρμογής στις προτιμήσεις του.

Για να επιτευχθεί αυτό η εφαρμογή χρησιμοποιήθηκε η βιβλιοθήκη RxDart για να ενώσει τρία διαφορετικά ρεύματα (streams) που παρακολουθούν συνεχώς τη βάση δεδομένων. Αυτά τα τρία ρεύματα αντιστοιχούν στις γλώσσες, τις πόλεις και τις πανεπιστημιούπολεις. Αυτή η σύνδεση επιτρέπει την άμεση ενημέρωση των πληροφοριών που εμφανίζονται σε κάθε στοιχείο επιλογής (select box), προσφέροντας έτσι πάντα τις κατάλληλες επιλογές στον χρήστη.

```

final campusesStream = CampusesService().campuses();
final citiesStream = AppConfigService().getCities();
final languagesStream = AppConfigService().getLanguages();

late final combinedStream =
  Rx.combineLatest3<List<Campus>, List<City>, dynamic, dynamic>(
    campusesStream,
    citiesStream,
    languagesStream,
    (campuses, cities, languages) =>
      {'campuses': campuses, 'cities': cities, 'languages': languages},
  );

```

Figure 38: Συνδυασμός ρευμάτων για ορθή ενημέρωση των στοιχείων γραφικής διεπαφής επιλογής.

```

final CollectionReference campusesCollection =
  FirebaseFirestore.instance.collection('universities');

Stream<List<Campus>> campuses() {
  return campusesCollection.snapshots().map((QuerySnapshot query) {
    return query.docs.map((DocumentSnapshot doc) {
      return Campus.fromJson(doc.data() as Map<String, dynamic>);
    }).toList();
  });
}

```

Figure 39: Μέθοδος παραγωγής ρεύματος δεδομένων με χρήση ερωτήματος (query) στην βάση δεδομένων.

Παράλληλα, χρησιμοποιήθηκε το στοιχείο Provider του πλαισίου Flutter για να εμφανίζονται δυναμικά όλες αυτές οι πληροφορίες στην κύρια διεπαφή χρήστη. Με τη χρήση του Provider.of<Student?>(context), οι πληροφορίες που αναφέρονται στις γλώσσες, τις πόλεις και τις πανεπιστημιούπολεις παρουσιάζονται δυναμικά, ενσωματώνοντας τις επιλογές του χρήστη.

Εάν, κατά τη διάρκεια της παραμονής στις ρυθμίσεις, ο χρήστης αλλάξει τη γλώσσα και πατήσει το κουμπί "Αποθήκευση", όλα τα κείμενα στην εφαρμογή προσαρμόζονται άμεσα στη νέα επιλεγμένη γλώσσα. Αυτό επιτυγχάνεται μέσω του χρήστη των ρευμάτων που συνδέονται με τη βάση δεδομένων και του RxDart, παρέχοντας έτσι άμεση ανταπόκριση στις αλλαγές προτιμήσεων του χρήστη.

```

final student = Provider.of<Student?>(context);
final allTranslations = Provider.of<List<TranslationPage?>>(context);
final translation =
  allTranslations.firstWhere((tr) => tr?.id == 'settings');
return StreamBuilder(
  stream: combinedStream,
  builder: (context, snapshot) {

```

Figure 40: Χρήση της κλάσης Provider, για δυναμική μετάφραση.

3.9 Δομή βάσης δεδομένων

Όπως προαναφέρθηκε στο κεφάλαιο 3.2, για την υλοποίηση της βάσης δεδομένων, επιλέχθηκε η Firebase Firestore. Γρήγορη, διαδεδωμένη και ανθεκτική σε αδύναμη ή και ανύπαρκτη σύνδεση στο διαδίκτυο, με την έξυπνη χρήση της μνήμης cache. Ακολουθεί ένα διάγραμμα με τις βασικές οντότητες που θα υπάρχουν σε αυτήν.

Κεφάλαιο 4ο: Εγχειρίδιο περιβάλλοντος διαχείρισης

4.1 Εισαγωγή

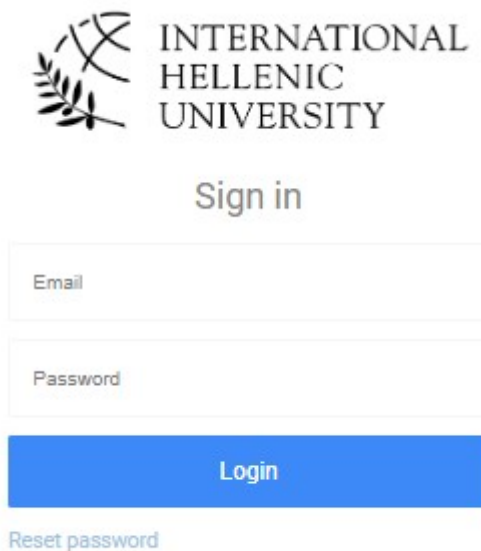
Σε αυτό το κεφάλαιο, θα εξερευνηθούν βήμα προς βήμα το Διαχειριστικό Περιβάλλον, παρέχοντας λεπτομερείς οδηγίες για να χρησιμοποιηθεί αποτελεσματικά. Αυτό το κεφάλαιο σας προσφέρει μια πλήρη κατανόηση των λειτουργιών και των δυνατοτήτων που προσφέρει το διαχειριστικό περιβάλλον.

Ξεκινώντας από τον κεντρικό πίνακα διαχείρισης και κινούμενοι σε κάθε καρτέλα, θα εξερευνηθούν οι λειτουργίες του συστήματος, προσφέροντάς αναλυτικές οδηγίες για κάθε βήμα. Επιπλέον, θα εξηγηθεί η διαδικασία προσθήκης, επεξεργασίας, και διαγραφής στοιχείων, ώστε να υπάρξει σωστή διαχείριση της εφαρμογής.

4.2 Σύνδεση στο σύστημα

4.2.1 Σύνδεση

Το πρώτο πράγμα που φαίνεται είναι η σελίδα σύνδεσης. Εκεί, θα πρέπει να συμπληρωθεί το email και ο κωδικός πρόσβασης του διαχειριστή, τα οποία θα έχουν δοθεί. Μόλις εισαχθούν τα στοιχεία, πατώντας το κουμπί "Login" θα αποκτηθεί η πρόσβαση στο σύστημα για τον διαχειριστή.



INTERNATIONAL
HELLENIC
UNIVERSITY

Sign in

Email

Password

Login

[Reset password](#)

Figure 41: Σύνδεση στο σύστημα

4.2.2 Επαναφορά κωδικού πρόσβασης

Αν κάποιος διαχειριστής ξεχάσει τον κωδικό πρόσβασης του, μπορεί να επαναφέρει τον κωδικό του ακολουθώντας τα παρακάτω βήματα:

- Συμπλήρωση του email: Στη σελίδα εισόδου, πριν πατήσετε "Σύνδεση", συμπληρώστε το email που συνδέεται με τον λογαριασμό σας.
- Επιλέξτε "Reset password": Κάτω από τη φόρμα εισόδου, υπάρχει η επιλογή "Reset password".
- Ακολουθήστε τις Οδηγίες: Θα λάβετε ένα email με οδηγίες για το πώς να επαναφέρετε τον κωδικό πρόσβασης σας. Ακολουθήστε τα βήματα που παρέχονται στο email.
- Ορίστε Νέο Κωδικό: Αφού ακολουθήσετε τις οδηγίες, θα σας ζητηθεί να ορίσετε έναν νέο κωδικό πρόσβασης.
- Συνδεθείτε με το Νέο Κωδικό: Επιστρέψτε στη σελίδα εισόδου και συνδεθείτε χρησιμοποιώντας το νέο σας κωδικό.

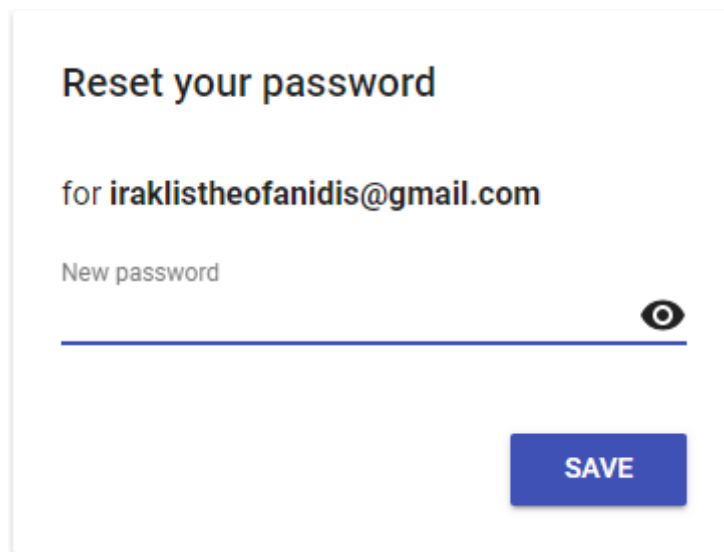
The image shows a web form titled "Reset your password" for the email address "iraklistheofanidis@gmail.com". Below the title, there is a text input field labeled "New password" with a blue underline. To the right of the input field is a small circular icon with an eye, used for toggling password visibility. At the bottom right of the form is a blue button with the text "SAVE" in white capital letters.

Figure 42: Επαναφορά κωδικού

Τώρα είστε έτοιμοι να συνεχίσετε τη διαχείριση του συστήματος με τον ενημερωμένο σας κωδικό πρόσβασης.

4.3 Ανακάλυψη του Διαχειριστικού Περιβάλλοντος

4.3.1 Κεντρικός Πίνακας Διαχείρισης

Στον σχεδιασμό της γραφικής διεπαφής, περιλαμβάνεται ένα αριστερό μενού πλοήγησης για γρήγορη πλοήγηση, καθώς και μία κεφαλίδα στα δεξιά που περιλαμβάνει χρήσιμες πληροφορίες και επιλογές. Κάτω από αυτά, το κύριο περιεχόμενο αντιπροσωπεύεται από περιεχόμενο το οποίο αλλάζει δυναμικά ανάλογα με την λειτουργία που έχετε επιλέξει στο μενού που υπάρχει αριστερά.

Στην κεφαλίδα, στα δεξιά, υπάρχει το όνομα του συνδεδεμένου διαχειριστή. Με ένα απλό κλικ πάνω σε αυτό, ανοίγει ένα μενού με διάφορες επιλογές, συμπεριλαμβανομένης της δυνατότητας αποσύνδεσης από το σύστημα.

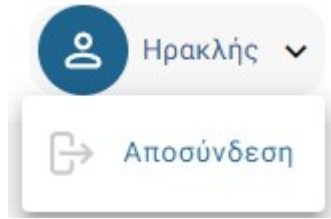


Figure 43: Logout

Το αριστερό μενού περιλαμβάνει τις παρακάτω επιλογές, που αντιπροσωπεύουν διάφορες λειτουργίες του διαχειριστικού περιβάλλοντος:

1. Μεταφράσεις: Στην καρτέλα "Μεταφράσεις," υπάρχουν όλες μεταφράσεις για τα στατικά κείμενα της mobile εφαρμογής.
2. Στατικά κείμενα: Στην καρτέλα "Στατικά κείμενα" υπάρχουν διάφορα κείμενα της εφαρμογής που αφορούν το ΔΙ.ΠΑ.Ε.
3. Διαχειριστές: Στην καρτέλα "Διαχειριστές" που είναι προσβάσιμη μόνο από τον super-admin και περιλαμβάνει τη δυνατότητα επεξεργασίας των διαχειριστών.
4. Ειδοποιήσεις: Στην καρτέλα "Ειδοποιήσεις," γίνεται η διαχείριση των ειδοποιήσεων που στέλλονται στους χρήστες της εφαρμογής.
5. Πανεπιστημιούπολεις: Στην καρτέλα "Πανεπιστημιούπολεις" περιέχονται πληροφορίες για τις πανεπιστημιούπολεις του ΔΙ.ΠΑ.Ε.
6. Τμήματα: Στην καρτέλα "Τμήματα," υπάρχουν πληροφορίες για τα τμήματα του ΔΙ.ΠΑ.Ε.
7. Διαδραστικοί Χάρτες: Στην καρτέλα "Διαδραστικοί Χάρτες" υπάρχουν οι διαδραστικούς χάρτες του ΔΙ.ΠΑ.Ε που έχει σχέση 1-1 με τις πανεπιστημιούπολεις.
8. Πρεσβείες: Στην καρτέλα "Πρεσβείες," υπάρχουν πληροφορίες για όλες τις πρεσβείες στην Ελλάδα.
9. Μουσεία: Στην καρτέλα "Μουσεία" περιέχονται πληροφορίες που έχουν προστεθεί στο σύστημα για κάποια μουσεία.
10. Αξιοθέατα: Στην καρτέλα "Αξιοθέατα" είναι παρόμοιο με το "Μουσεία."
11. Γενικοί πόροι: Στην καρτέλα "Γενικοί πόροι" περιέχονται πληροφορίες για γενικούς πόρους όπως η εύκολη αναζήτηση του τηλεφώνου της αστυνομίας σε περίπτωση ανάγκης.
12. Chatbot: Στην καρτέλα "Chatbot," ο χρήστης θα βρει απαντήσεις σχετικές με την κινητή εφαρμογή.
13. Settings: Στην καρτέλα "Settings" περιλαμβάνει σημαντικές λειτουργίες για ολόκληρη την εφαρμογή.

Κάθε καρτέλα έχει τις δικές του λεπτομερείς λειτουργίες που θα αναλύσουμε σε επόμενες παραγράφους πιο αναλυτικά.

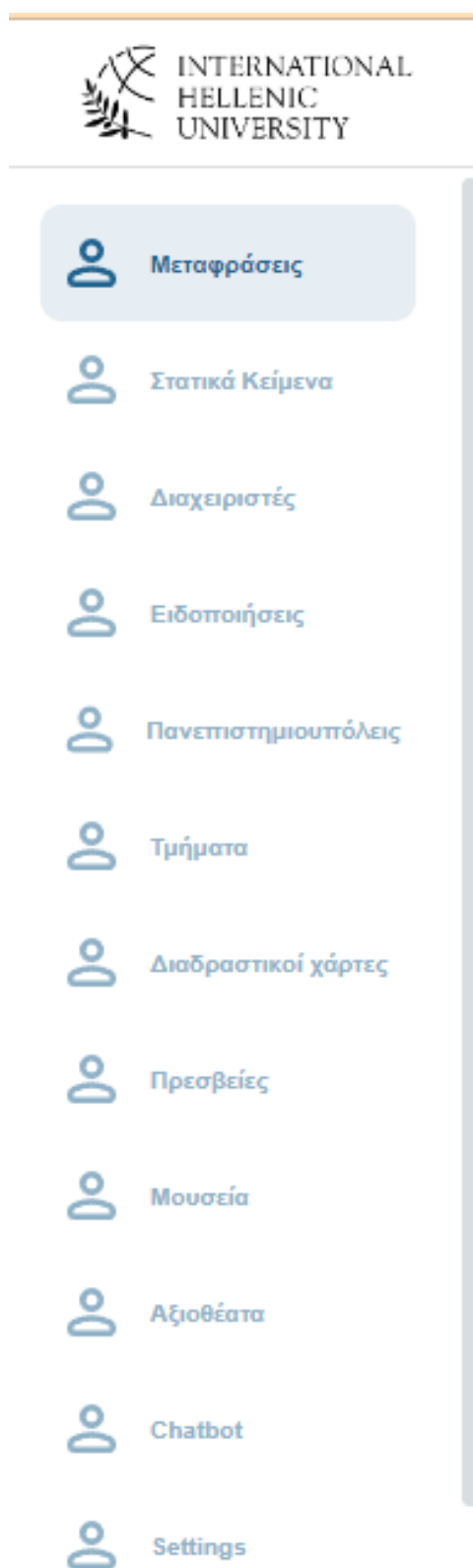


Figure 44: Καρτέλες διαχειριστικού

4.4 Περιγραφή των καρτελών

4.4.1 Μεταφράσεις

Στην καρτέλα "Μεταφράσεις," γίνεται η διαχείριση των στατικών κειμένων για κάθε σελίδα της κινητής εφαρμογής σε διάφορες γλώσσες.

4.4.1.1 Προσθήκη νέας σελίδας

Πάνω δεξιά, υπάρχει το κουμπί "Προσθήκη Σελίδας." Όταν το πατήσετε, εμφανίζεται ένα pop-up παράθυρο όπου μπορείτε να εισάγετε το όνομα της νέας σελίδας που θέλετε να προσθέσετε.

Με αυτόν τον τρόπο, είναι δυνατή η διαχείριση των μεταφράσεων για τις σελίδες και η εύκολη προσθήκη νέων σελίδων στην εφαρμογή.



Figure 45: Κουμπί προσθήκη σελίδας

4.4.1.2 Προσθήκη ή επεξεργασία κλειδιού

Για να προστεθεί μια νέα μετάφραση, ακολουθήστε τα παρακάτω βήματα:

- Συμπληρώστε ένα μοναδικό κείμενο ως αναγνωριστικό για τη μετάφραση.
- Πατήστε το κουμπί "Create" για να δημιουργήσετε τη νέα μετάφραση.
- Θα εμφανιστεί έπειτα ένα pop-up για να συμπληρώσετε την λέξη σε όλες τις γλώσσες του συστήματος.

Figure 46: Φόρμα νέου κλειδιού για μετάφραση

Με αυτόν τον τρόπο, μπορείτε να προσθέσετε νέες μεταφράσεις για το κείμενο της εφαρμογής.

4.4.2 Προβολή μεταφράσεων

Στον ίδιο πίνακα "Μεταφράσεις," φαίνεται μια ολοκληρωμένη λίστα με όλες τις μεταφράσεις που έχουν δημιουργηθεί για όλες τις σελίδες της εφαρμογής. Αυτό καθιστά ευκολότερη τη διαχείριση και τον έλεγχο των μεταφράσεων.

key	Ιταλικά	Γαλλικά	Ελληνικά	Αγγλικά
building_description	Descrizione dell'edificio	Descriptif du bâtiment	Περιγραφή κτιρίου	Building description
tap	Tocca un edificio sulla mappa per ottenere informazioni	Appuyez sur un bâtiment sur la carte pour obtenir des informations	Πατήστε σε ένα κτίριο στο χάρτη για πληροφορίες	Tap on a building on the map for information
office	Sale/Uffici	Halls/Bureaux	Αίθουσες/Γραφεία	Halls/Offices
no_data	Nessun dato trovato per questo campus.	Aucune donnée trouvée pour ce campus.	Δεν βρέθηκαν δεδομένα για αυτήν την πανεπιστημιούπολη.	No data found for this campus.
building_name	Nome Immobile	Nom du bâtiment	Όνομα κτιρίου	Building name

Figure 47: Προβολή όλων των κλειδιών μιας σελίδας

Στη λίστα αυτή, φαίνονται τα αναγνωριστικά των μεταφράσεων, το κείμενο που έχετε εισαγάγει, και να προβείτε σε ενέργειες όπως η επεξεργασία ή η διαγραφή τους πατώντας πάνω στην γραμμή που θέλετε να επεξεργαστείτε.

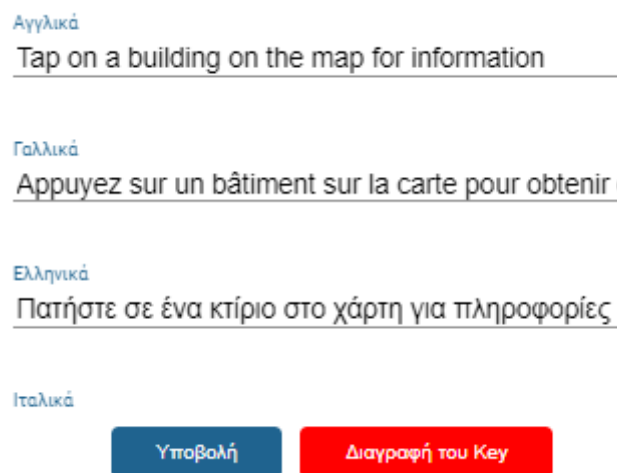


Figure 48: Επεξεργασία ενός κλειδιού

4.4.3 Διαχειριστές

4.4.3.1 Προβολή διαχειριστών

Στην καρτέλα "Διαχειριστές," μπορείτε να προβάλλετε όλους τους διαχειριστές που έχουν δημιουργηθεί. Ωστόσο, η δυνατότητα διαγραφής ή δημιουργίας νέου διαχειριστή προσφέρεται μόνο στον super Admin.

Ο super Admin έχει την εξουσία να διαγράψει υπάρχοντες διαχειριστές ή να δημιουργήσει νέους, ενώ οι υπόλοιποι διαχειριστές διαθέτουν μόνο τη δυνατότητα προβολής της λίστας με τους διαχειριστές.

Με αυτόν τον τρόπο, εξασφαλίζεται η ασφαλής διαχείριση των διαχειριστών της εφαρμογής, με μοναδικά προνόμια για τον super Admin.



Figure 49: Πίνακας διαχειριστών

4.4.3.2 Προσθήκη διαχειριστή

Για να προσθέσετε έναν νέο διαχειριστή, ακολουθήστε τα παρακάτω βήματα:

- Στην επικεφαλίδα της καρτέλας "Διαχειριστές," εντοπίστε το κουμπί "Προσθήκη Διαχειριστή" που βρίσκεται στην πάνω δεξιά γωνία.
- Κάντε κλικ στο κουμπί "Προσθήκη Διαχειριστή" για να ανοίξετε τη φόρμα προσθήκης νέου διαχειριστή.



Figure 50: Κουμπί - Προσθήκη διαχειριστή

Στη συνέχεια του βήματος 2, κατά τη διαδικασία προσθήκης νέου διαχειριστή, θα εμφανιστεί ένα popup όπου πρέπει να συμπληρώσετε τα απαραίτητα στοιχεία του διαχειριστή. Ακολουθήστε τα παρακάτω βήματα:

- Συμπληρώστε το email του νέου διαχειριστή.
- Συμπληρώστε το όνομα και το επώνυμο του διαχειριστή.

Μετά τη συμπλήρωση των παραπάνω πεδίων, πατήστε το κουμπί "Αποθήκευση" για να ολοκληρώσετε τη διαδικασία προσθήκης.

Με αυτόν τον τρόπο, ο νέος διαχειριστής προστίθεται στο σύστημα με τα κατάλληλα δικαιώματα πρόσβασης που έχουν οριστεί.

4.4.4 Στατικά κείμενα

Στην καρτέλα "Στατικά κείμενα," μπορούν να προβληθούν όλα τα στατικά κείμενα που έχουν καταχωρηθεί στο σύστημα.

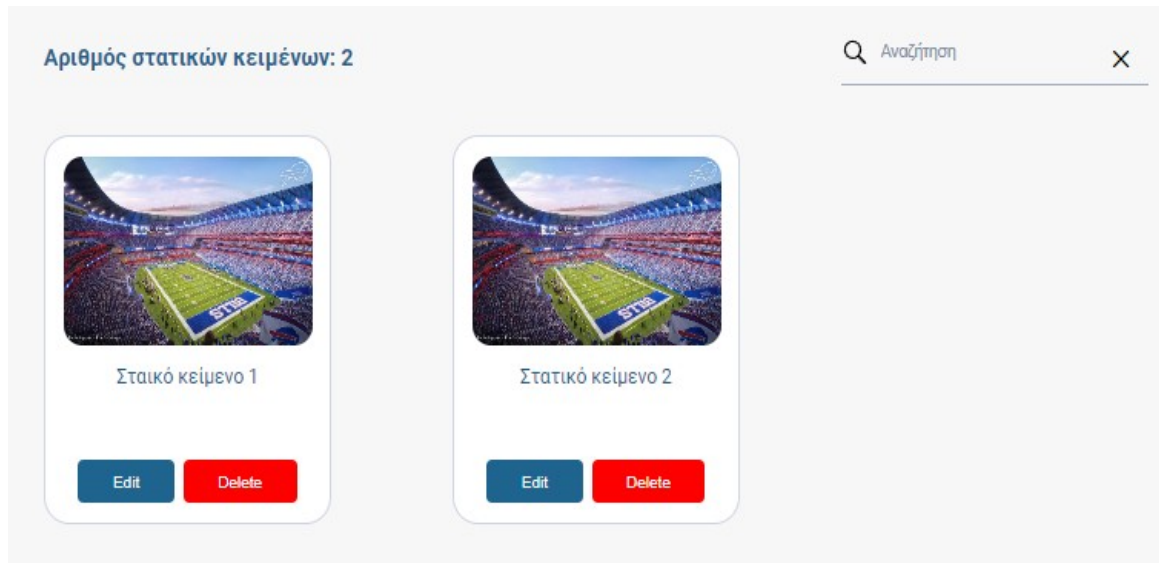


Figure 51: Προβολή στατικών κειμένων

Παρέχονται οι εξής επιλογές:

- Κάνοντας κλικ στο κουμπί "Delete," μπορείτε να διαγράψετε αμέσως ένα στατικό κείμενο από το σύστημα.
- Κάνοντας κλικ στο κουμπί "Edit," μπορείτε να επεξεργαστείτε το στατικό κείμενο μέσω της αντίστοιχης φόρμας.
- Πατώντας στην επικεφαλίδα δεξιά το κουμπί "Προσθήκη στατικού κειμένου," θα μεταφερθείτε στη φόρμα δημιουργίας νέου στατικού κειμένου, όπου μπορείτε να εισάγετε νέα δεδομένα.



Figure 52: Κουμπί - προσθήκη στατικού κειμένου

Στη φόρμα δημιουργίας νέας πρεσβείας, πρέπει να προσθέσετε τα ακόλουθα δεδομένα:

1. Όνομα: Εισαγωγή τίτλου για το στατικό κείμενο. Η εγγραφή του τίτλου θα γίνει στη γλώσσα που έχει επιλεγεί από τα Settings (βλ. 4.4.14.1).
2. Περιγραφή: Καταχώρηση μιας σύντομης περιγραφής που επίσης θα είναι στη γλώσσα που έχει επιλεγεί από τα Settings.
3. Κατηγορία: Καταχώρηση της κατηγορίας στατικού κειμένου. Η καταγραφή της κατηγορίας θα είναι στη γλώσσα που έχει επιλεγεί από τα Settings.
4. Link: Εισαγωγή ενός προαιρετικό συνδέσμου (URL) που οδηγεί στην ιστοσελίδα που έχει περισσότερες πληροφορίες για το θέμα του κειμένου.

5. Φωτογραφία: Επισύναψη μιας προαιρετικής φωτογραφίας που αντιπροσωπεύει το στατικό κείμενο.

4.4.5 Ειδοποιήσεις

Στη διαχείριση των ειδοποιήσεων χρησιμοποιούνται δύο στήλες:

4.4.5.1 Φόρμα δημιουργίας ειδοποίησης

- Δημιουργός: Συμπληρώνετε το όνομα του δημιουργού της ειδοποίησης.
- Τίτλος: Συμπληρώνετε έναν τίτλο για την ειδοποίηση.
- Περιεχόμενο: Καταχωρείτε το κείμενο της ειδοποίησης.

Περιεχόμενο ειδοποίησης

Δημιουργός

Τίτλος ειδοποίησης

Περιεχόμενο ειδοποίησης

Φίλτρετε τους χρήστες

➤ Αποστολή ειδοποίησης

Figure 53: Φόρμα ειδοποίησης για την κινητή εφαρμογή

4.4.5.2 Λίστα ειδοποιήσεων

- Εμφανίζονται όλες οι δημιουργηθείσες ειδοποιήσεις
- Πιθανότητα διαγραφής όλες κάθε ειδοποίησης από τη λίστα

Με αυτόν τον τρόπο, ο διαχειριστής μπορεί να δημιουργήσει νέες ειδοποιήσεις μέσω της φόρμας και να διαχειριστεί όλες τις υπάρχουσες ειδοποιήσεις στη λίστα.

Λίστα ειδοποιήσεων

03/10/2023 - 18:40:56



titlos eidopioisis

Periexomeno eidopioisis

Δημιουργός: Dimiourgos

Εστάλη σε όλους τους χρήστες

Είδος ειδοποίησης: Notification

28/05/2023 - 18:22:53



Dokimastiko2

Δοκιμαστικs2o Δοκιμαστικο Δοκιμαστικο
Δοκιμαστικο Δοκιμαστικο Δοκιμαστικο
Δοκιμαστικο Δοκιμαστικο Δοκιμαστικο
Δοκιμαστικο

Δημιουργός: Efstathios Antoniou

Εστάλη σε όλους τους χρήστες

Είδος ειδοποίησης: Email και Notification

Figure 54: Λίστα ειδοποιήσεων

4.4.6 Πανεπιστημιούπολεις

Στην καρτέλα "Πανεπιστημιούπολεις," μπορούν να προβληθούν όλες οι πανεπιστημιούπολεις που έχουν καταχωρηθεί στο σύστημα.

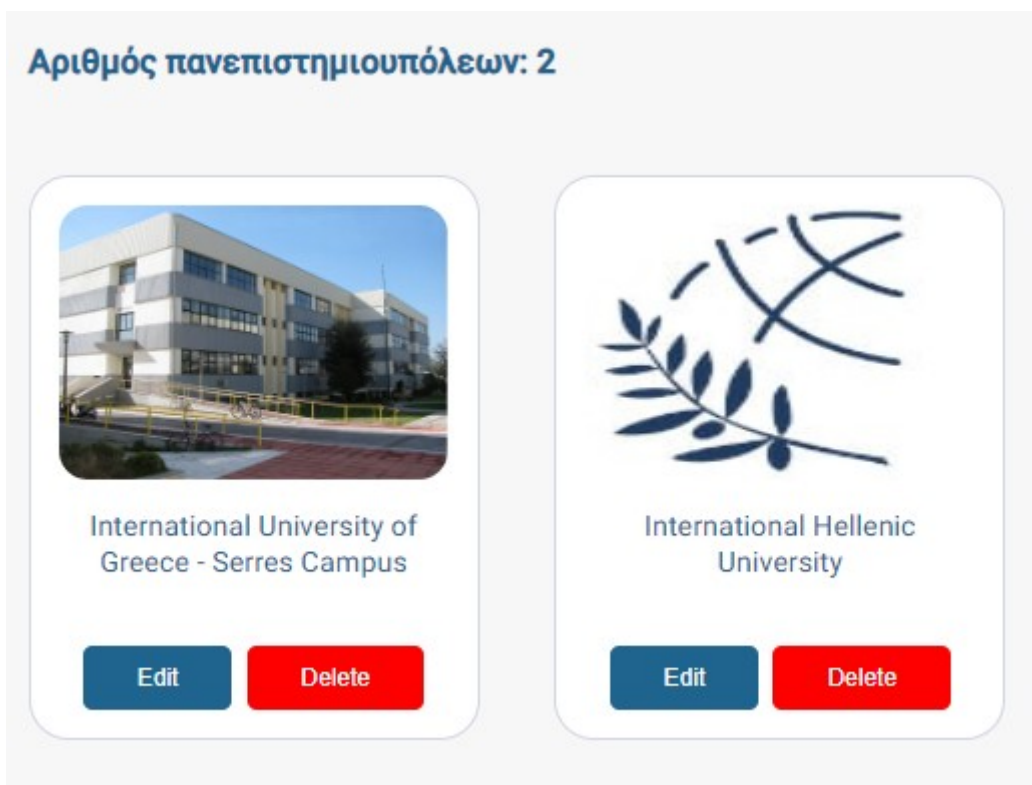


Figure 55: Προβολή πανεπιστημιούπων

Παρέχονται οι εξής επιλογές:

- Κάνοντας κλικ στο κουμπί "Delete," μπορείτε να διαγράψετε αμέσως μια πανεπιστημιούπολη από το σύστημα.
- Κάνοντας κλικ στο κουμπί "Edit," μπορείτε να επεξεργαστείτε τα στοιχεία μιας πανεπιστημιούπολης μέσω της αντίστοιχης φόρμας.
- Πατώντας στην επικεφαλίδα δεξιά το κουμπί "Προσθήκη Πανεπιστημιούπολης," θα μεταφερθείτε στη φόρμα δημιουργίας νέας πανεπιστημιούπολης, όπου μπορείτε να εισάγετε νέα δεδομένα.



Figure 56: Κουμπί - προσθήκη πανεπιστημιούπολης

Στη φόρμα δημιουργίας νέας πανεπιστημιούπολης, πρέπει να προσθέσετε τα ακόλουθα δεδομένα:

1. Όνομα: Εισαγωγή του ονόματος της πανεπιστημιούπολης. Η εγγραφή του ονόματος θα γίνει στη γλώσσα που έχει επιλεγεί από τα settings.
2. Περιγραφή: Καταχώρηση μιας σύντομης περιγραφής που επίσης θα είναι στη γλώσσα που έχει επιλεγεί από τα settings.
3. Link: Εισαγωγή ενός συνδέσμου (URL) που οδηγεί στην ιστοσελίδα της πανεπιστημιούπολης.
4. Τηλέφωνο: Καταχώρηση του τηλεφώνου επικοινωνίας της πανεπιστημιούπολης.

5. Email: Καταχώρηση της διεύθυνσης email για επικοινωνία.
6. Πόλη: Καταχώρηση της πόλης όπου βρίσκεται η πανεπιστημιούπολη. Η καταγραφή της πόλης θα είναι στη γλώσσα που έχει επιλεγεί από τα settings.



Figure 57: Προσθήκη ονόματος και περιγραφής της πανεπιστημιούπολης σε όλες τις γλώσσες της εφαρμογής

Για να αποθηκευτούν τα στοιχεία που συμπληρώσατε, παρακαλείστε να πατήσετε το κουμπί "Αποθήκευση" στην επικεφαλίδα στο πάνω δεξιά μέρος της οθόνης.

4.4.7 Τμήματα

Στην καρτέλα "Τμήματα," μπορούν να προβληθούν όλα τα τμήματα που έχουν καταχωρηθεί στο σύστημα. Η αναζήτηση που βλέπετε στην εικόνα από κάτω χρησιμοποιεί κώδικα αμφίδρομου μετασχηματισμού ελληνικών-λατινικών χαρακτήρων, από μία δική μας βιβλιοθήκη που υλοποιήσαμε και ανεβάσαμε στο NPM, και μπορεί ο οποιοσδήποτε να την χρησιμοποιήσει.

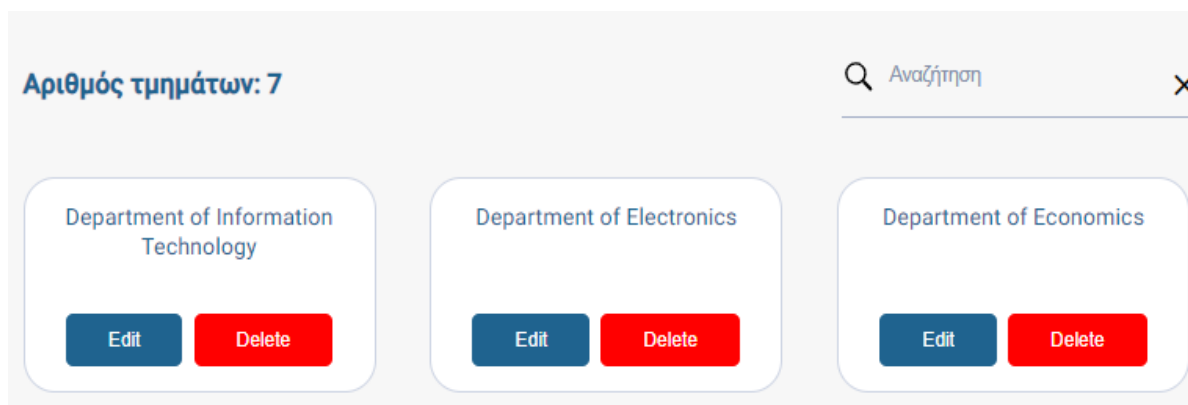


Figure 58: Προβολή τμημάτων του ΔΙ.ΠΑ.Ε

Παρέχονται οι εξής επιλογές:

- Κάνοντας κλικ στο κουμπί "Delete," μπορείτε να διαγράψετε αμέσως τμήμα από το σύστημα.
- Κάνοντας κλικ στο κουμπί "Edit," μπορείτε να επεξεργαστείτε τα στοιχεία ενός τμήματος μέσω της αντίστοιχης φόρμας.

- Πατώντας στην επικεφαλίδα δεξιά το κουμπί "Προσθήκη Τμήματος," θα μεταφερθείτε στη φόρμα δημιουργίας νέου τμήματος, όπου μπορείτε να εισάγετε νέα δεδομένα.

Στη φόρμα δημιουργίας νέας πανεπιστημιούπολης, πρέπει να προσθέσετε τα ακόλουθα δεδομένα:

1. Όνομα: Εισαγωγή του ονόματος της τμήματος. Η εγγραφή του ονόματος θα γίνει στη γλώσσα που έχει επιλεγεί από τα settings.
2. Πανεπιστημιούπολη: Επιλογή πανεπιστημιούπολης από αυτές που έχουν ήδη περαστεί στο σύστημα.

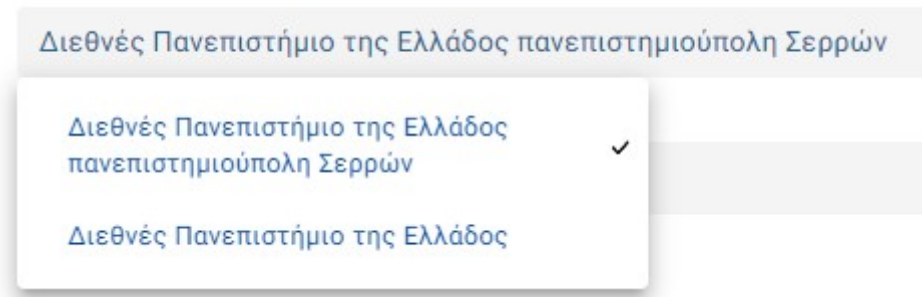


Figure 59: Επιλογή πανεπιστημιούπολης της εφαρμογής

Για να αποθηκευτούν τα στοιχεία που συμπληρώσατε, παρακαλείστε να πατήσετε το κουμπί "Αποθήκευση" στην επικεφαλίδα στο πάνω δεξιά μέρος της οθόνης.

4.4.8 Διαδραστικοί Χάρτες

Στην καρτέλα “Διαδραστικοί χάρτες”, προβάλλονται οι χάρτες των πανεπιστημιούπολων.

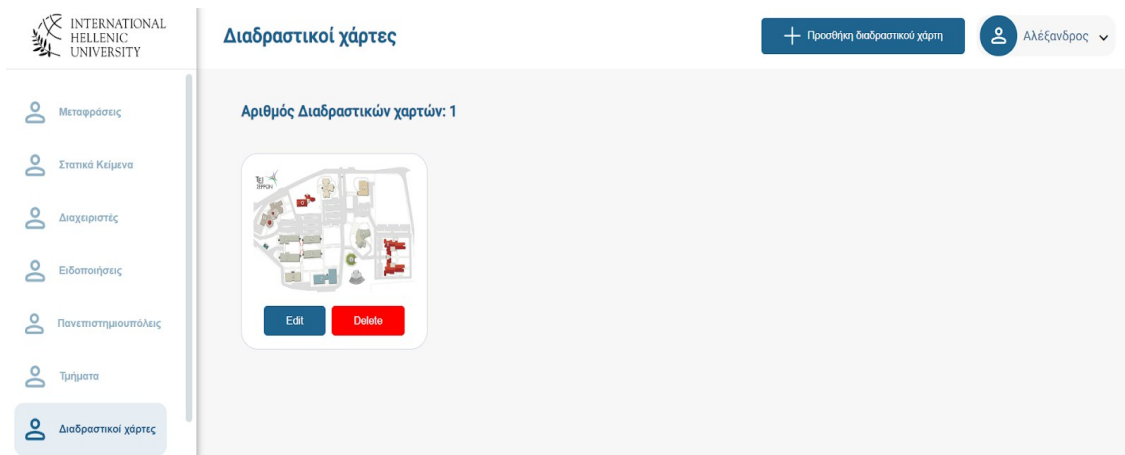


Figure 60: Προβολή διαδραστικών χαρτών

- Κάνοντας κλικ στο κουμπί "Delete," μπορείτε να διαγράψετε τον χάρτη από το σύστημα.
- Κάνοντας κλικ στο κουμπί "Edit," μπορείτε να επεξεργαστείτε τις πληροφορίες του χάρτη.
- Πατώντας στην επικεφαλίδα δεξιά το κουμπί "Προσθήκη διαδραστικού χάρτη", θα μεταφερθείτε στη φόρμα δημιουργίας νέας πρεσβείας, όπου μπορείτε να εισάγετε νέα δεδομένα.

Για την προσθήκη διαδραστικού χάρτη επιβάλλονται οι δύο παρακάτω ενέργειες, όπως φαίνονται στην φωτογραφία που ακολουθεί:

- Επιλογή πανεπιστημιούπολης την οποία θα αντιπροσωπεύει ο χάρτης.
- Προσθήκη φωτογραφίας η οποία θα αποτελέσει την βάση του καμβά του χάρτη.

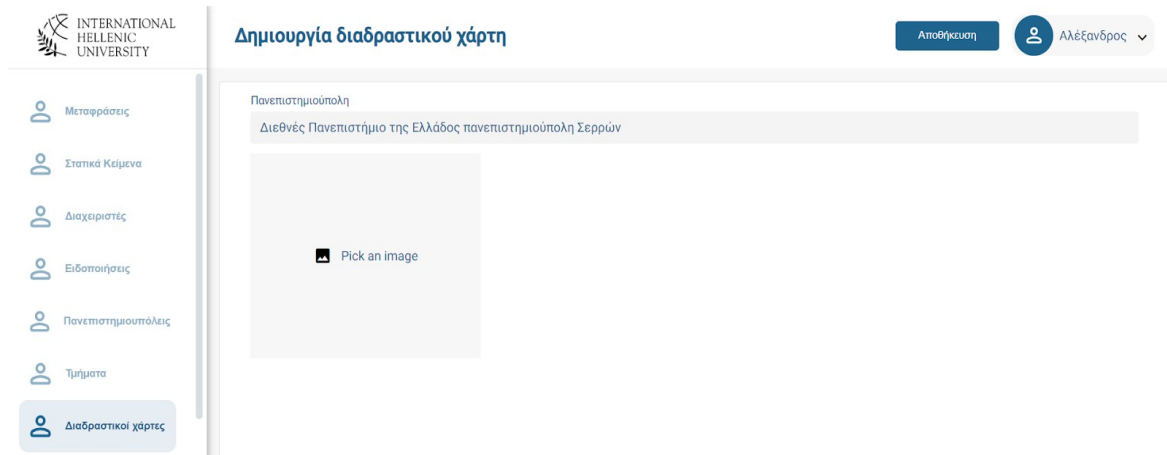


Figure 61: Φόρμα διαδραστικού χάρτη πριν την επιλογή φωτογραφίας

Κατόπιν επιλογής εικόνας, εμφανίζεται ένα κουμπί, στο κάτω αριστερά μέρος του πλαισίου, με το κείμενο “Begin polygon drawing mode”. Κάνοντας κλικ σε αυτό, δίνεται η δυνατότητα ορισμού σημείων ενός πολυγώνου επί της φωτογραφίας κάνοντας κλικ πάνω σε αυτήν. Μόλις οριστούν τα σημεία, κάνοντας κλικ στο κουμπί με το κείμενο “End drawing mode.”, το πολύγωνο ορίζεται και εμφανίζεται η φόρμα εισαγωγής δεδομένων του πολυγώνου, όπως φαίνεται στην φωτογραφία.



Figure 62: Φόρμα διαδραστικού χάρτη μετά την επιλογή φωτογραφίας

Σε αυτή την φόρμα, υπάρχουν τα εξής πεδία:

- Πεδίο κειμένου για το όνομα του κτιρίου που περικλείει το πολύγωνο.
- Πεδίο κειμένου για την περιγραφή του κτιρίου που περικλείει το πολύγωνο.
- Φόρμα πολλαπλής επιλογής τμημάτων (από τα ορισμένα στην καρτέλα “Τμήματα”) τα οποία συστεγάζονται στο κτίριο.

- Κουμπί για την προσθήκη συστεγαζόμενων αιθουσών/γραφείων.

Κάνοντας κλικ στο κουμπί “Προσθήκη αίθουσας/γραφείου” εμφανίζεται μία φόρμα με 3 πεδία τα οποία αφορούν την αίθουσα.

Κτίριο Πληροφορικής Delete

Όνομα κτηρίου Γλώσσα

Περιγραφή κτηρίου Γλώσσα

1ο Γραφείο/Αίθουσα

Αριθμός γραφείου/αίθουσας	Όνομα γραφείου/αίθουσας	Υπεύθυνος γραφείου/αίθουσας	
<input type="text" value="101"/>	<input type="text" value="Αμφιθέατρο Α"/>	<input type="text" value="Κώστας Γουλιάνας"/>	❌

2ο Γραφείο/Αίθουσα

Αριθμός γραφείου/αίθουσας	Όνομα γραφείου/αίθουσας	Υπεύθυνος γραφείου/αίθουσας	
<input type="text" value="214"/>	<input type="text" value="Γραφείο ΣΤ'"/>	<input type="text" value="Παναγιώτης Αδαμίδης"/>	❌

Προσθήκη αίθουσας/γραφείου

Figure 63: Συμπληρωμένη φόρμα διαδραστικού χάρτη μετά την επιλογή φωτογραφίας

Επιπλέον, δίνεται η δυνατότητα προβολής των πληροφοριών και επεξεργασίας αυτών, οποιουδήποτε άλλου πολυγώνου, κάνοντας κλικ εντός αυτού. Τέλος, έχοντας επιλεγμένο ένα πολύγωνο, δίνεται η δυνατότητα διαγραφής αυτού, κάνοντας κλικ στο κουμπί “Delete”.

4.4.9 Πρεσβείες

Στην καρτέλα "Πρεσβείες," μπορούν να προβληθούν οι πρεσβείες που έχουν καταχωρηθεί στο σύστημα.

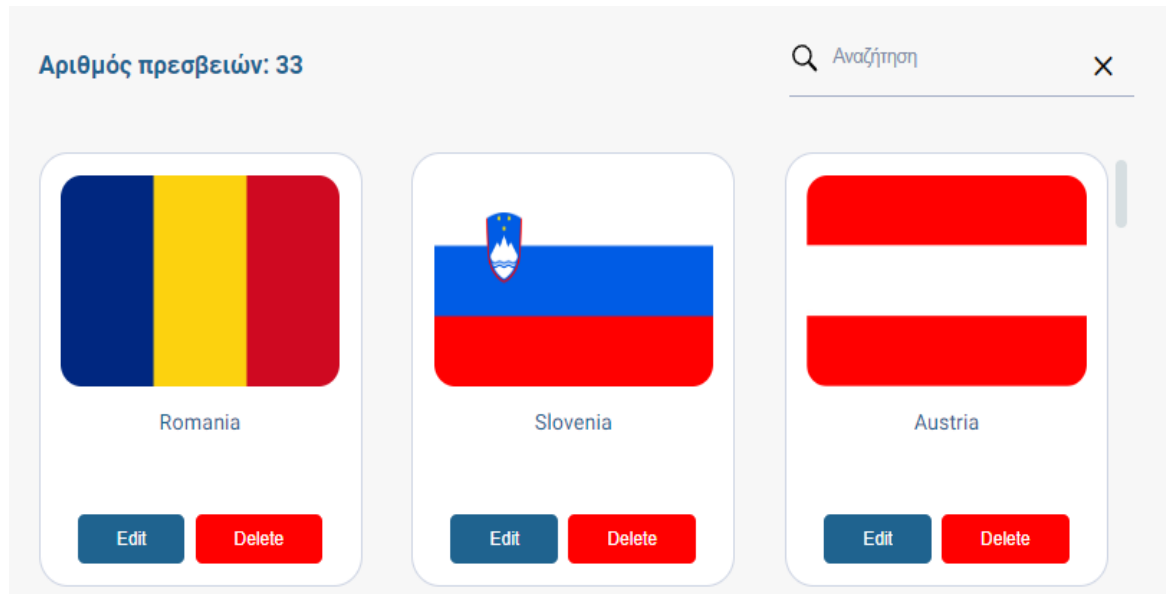


Figure 64: Προβολή πρεσβειών

Παρέχονται οι εξής επιλογές:

- Κάνοντας κλικ στο κουμπί "Delete," μπορείτε να διαγράψετε αμέσως μια πρεσβεία από το σύστημα.
- Κάνοντας κλικ στο κουμπί "Edit," μπορείτε να επεξεργαστείτε τα στοιχεία μιας πρεσβείας μέσω της αντίστοιχης φόρμας.
- Πατώντας στην επικεφαλίδα δεξιά το κουμπί "Προσθήκη Πρεσβείας," θα μεταφερθείτε στη φόρμα δημιουργίας νέας πρεσβείας, όπου μπορείτε να εισάγετε νέα δεδομένα.

Στη φόρμα δημιουργίας νέας πρεσβείας, πρέπει να προσθέσετε τα ακόλουθα δεδομένα:

1. Χώρα: Εισαγωγή της χώρας στην οποία βρίσκεται η πρεσβεία.
2. Περιγραφή: Καταχώρηση μιας σύντομης περιγραφής για την πρεσβεία. Μπορείτε να πληροφορήσετε για οτιδήποτε θεωρείτε σημαντικό, όχι μόνο τα προεπιλεγμένα πεδία που περιέχει το σύστημα.
3. Πόλη: Καταχώρηση της πόλης όπου βρίσκεται η πρεσβεία.
4. Ταχυδρομικός Κώδικας: Καταχώρηση του ταχυδρομικού κώδικα της περιοχής.
5. Τηλέφωνο: Καταχώρηση του τηλεφώνου επικοινωνίας της πρεσβείας.
6. Φωτογραφία της Χώρας: Επισύναψη μιας φωτογραφίας που αντιπροσωπεύει τη χώρα στην οποία βρίσκεται η πρεσβεία.
7. Προσθήκη Στίγματος με Google Maps: Επισύναψη στίγματος της τοποθεσίας της πρεσβείας χρησιμοποιώντας το Google Maps.

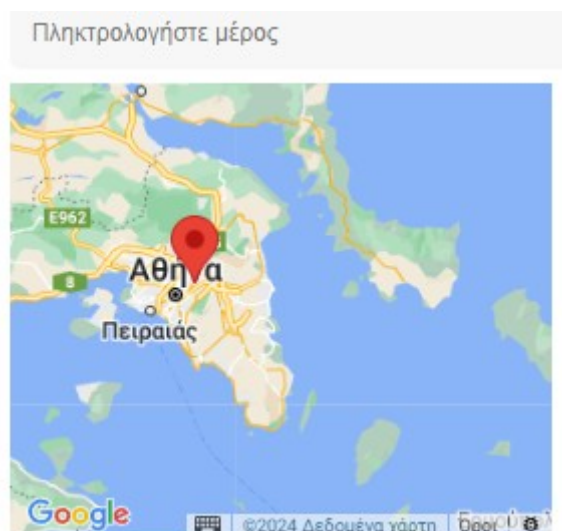


Figure 65: Προσθήκη στίγματος

Για να αποθηκευτούν τα στοιχεία που συμπληρώσατε, παρακαλείστε να πατήσετε το κουμπί "Αποθήκευση" στην επικεφαλίδα στο πάνω δεξιά μέρος της οθόνης.

4.4.10 Μουσεία

Στην καρτέλα "Μουσεία," μπορείτε να προβάλλετε όλα τα μουσεία που έχουν καταχωρηθεί στο σύστημα.

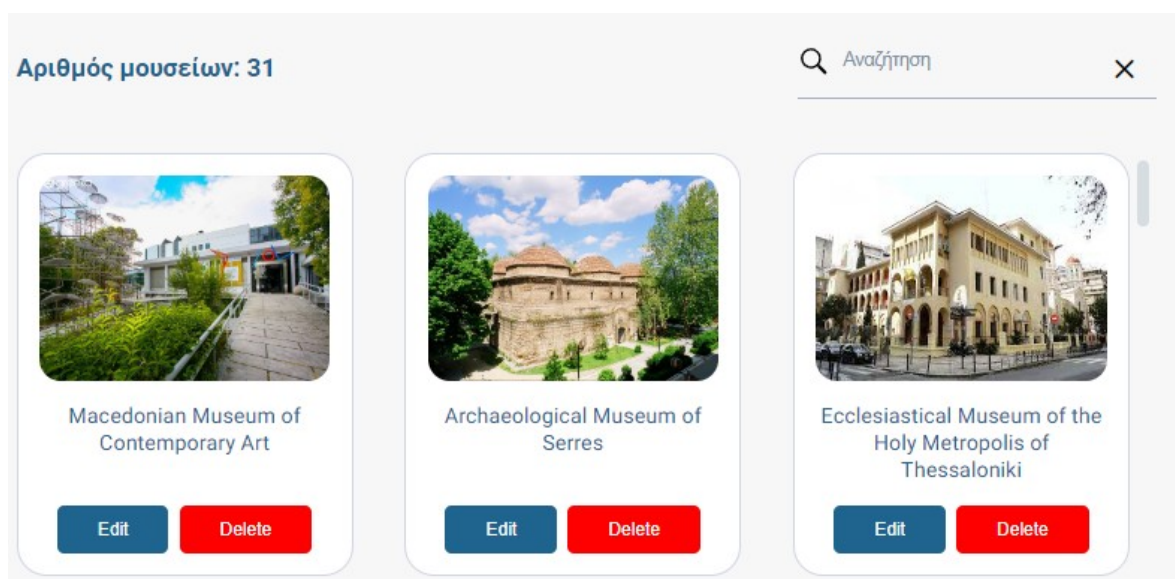


Figure 66: Προβολή μουσείων

Παρέχονται οι εξής επιλογές:

- Κάνοντας κλικ στο κουμπί "Delete," μπορείτε να διαγράψετε αμέσως ένα μουσείο από το σύστημα.
- Κάνοντας κλικ στο κουμπί "Edit," μπορείτε να επεξεργαστείτε τα στοιχεία ενός μουσείου μέσω της αντίστοιχης φόρμας.

- Πατώντας στην επικεφαλίδα δεξιά το κουμπί "Προσθήκη Μουσείου," θα μεταφερθείτε στη φόρμα δημιουργίας νέου μουσείου, όπου μπορείτε να εισάγετε νέα δεδομένα.

Στη φόρμα δημιουργίας νέου μουσείου, πρέπει να προσθέσετε τα ακόλουθα δεδομένα:

1. Όνομα: Εισαγωγή του ονόματος του μουσείου. Η εγγραφή του ονόματος θα γίνει στη γλώσσα που έχει επιλεγεί από τα settings.
2. Περιγραφή: Καταχώρηση μιας σύντομης περιγραφής που επίσης θα είναι στη γλώσσα που έχει επιλεγεί από τα settings.
3. Link: Εισαγωγή ενός συνδέσμου (URL) που οδηγεί στην ιστοσελίδα του μουσείου.
4. Πόλη: Καταχώρηση της πόλης όπου βρίσκεται το μουσείο. Η καταγραφή της πόλης θα είναι στη γλώσσα που έχει επιλεγεί από τα settings.
5. Προσθήκη Στίγματος με Google Maps: Επισύναψη στίγματος της τοποθεσίας του μουσείου χρησιμοποιώντας το Google Maps.
6. Φωτογραφία: Επισύναψη μιας φωτογραφίας που αντιπροσωπεύει το μουσείο.

Για να αποθηκευτούν τα στοιχεία που συμπληρώσατε, παρακαλείστε να πατήσετε το κουμπί "Αποθήκευση" στην επικεφαλίδα στο πάνω δεξιά μέρος της οθόνης.

4.4.11 Αξιοθέατα

Στην καρτέλα "Αξιοθέατα," μπορείτε να προβάλλετε όλες τα αξιοθέατα που έχουν καταχωρηθεί στο σύστημα.

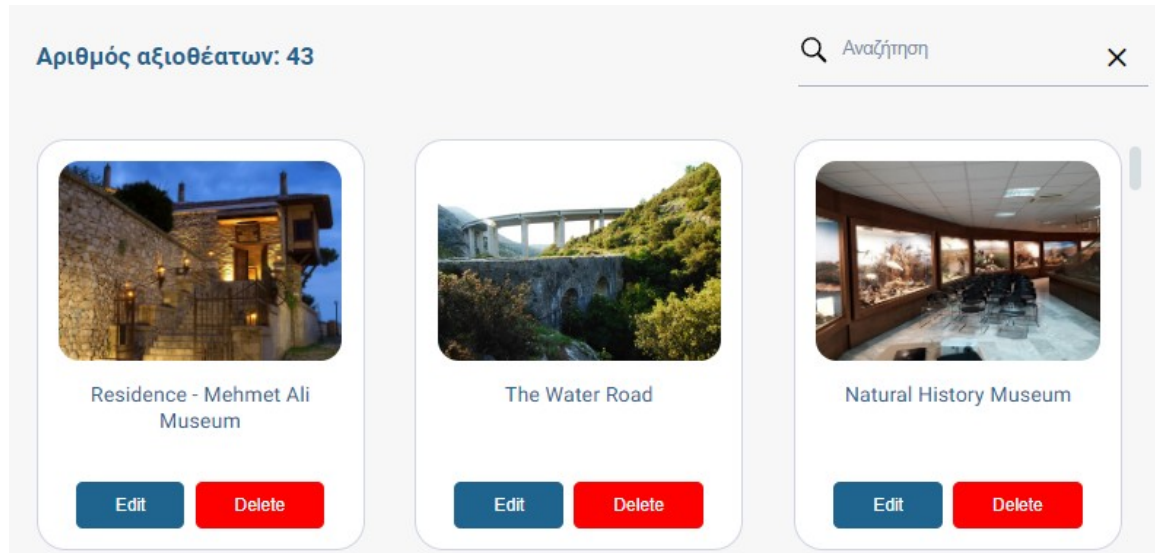


Figure 67: Προβολή αξιοθέατων

Παρέχονται οι εξής επιλογές:

- Κάνοντας κλικ στο κουμπί "Delete," μπορείτε να διαγράψετε αμέσως ένα αξιοθέατο από το σύστημα.
- Κάνοντας κλικ στο κουμπί "Edit," μπορείτε να επεξεργαστείτε τα στοιχεία ενός αξιοθέατου μέσω της αντίστοιχης φόρμας.

- Πατώντας στην επικεφαλίδα δεξιά το κουμπί "Προσθήκη Αξιοθέατου," θα μεταφερθείτε στη φόρμα δημιουργίας νέου αξιοθέατου, όπου μπορείτε να εισάγετε νέα δεδομένα.

Στη φόρμα δημιουργίας νέου αξιοθέατου, πρέπει να προσθέσετε τα ακόλουθα δεδομένα:

1. Όνομα: Εισαγωγή του ονόματος του μουσείου. Η εγγραφή του ονόματος θα γίνει στη γλώσσα που έχει επιλεγεί από τα settings.
2. Περιγραφή: Καταχώρηση μιας σύντομης περιγραφής που επίσης θα είναι στη γλώσσα που έχει επιλεγεί από τα settings.
3. Πόλη: Καταχώρηση της πόλης όπου βρίσκεται το μουσείο. Η καταγραφή της πόλης θα είναι στη γλώσσα που έχει επιλεγεί από τα settings.
4. Προσθήκη Στίγματος με Google Maps: Επισύναψη στίγματος της τοποθεσίας του μουσείου χρησιμοποιώντας το Google Maps.
5. Φωτογραφία: Επισύναψη μιας φωτογραφίας που αντιπροσωπεύει το μουσείο.

Για να αποθηκευτούν τα στοιχεία που συμπληρώσατε, παρακαλείστε να πατήσετε το κουμπί "Αποθήκευση" στην επικεφαλίδα στο πάνω δεξιά μέρος της οθόνης.

4.4.12 Γενικοί πόροι

Στην καρτέλα "Γενικοί πόροι," μπορείτε να προβάλλετε όλους τους γενικούς πόρους που έχουν καταχωρηθεί στο σύστημα.

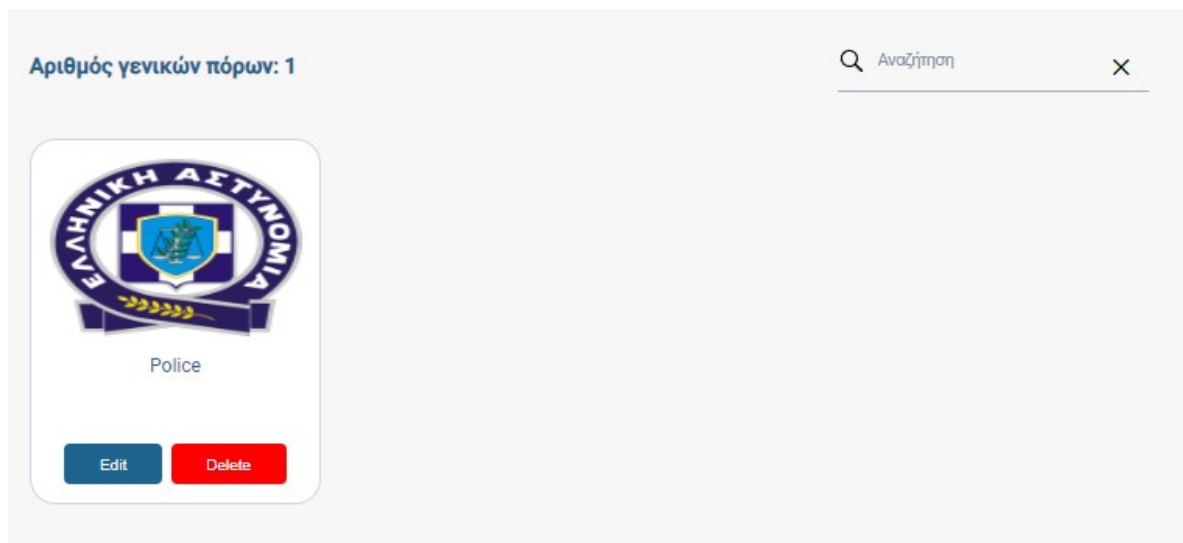


Figure 68: Προβολή γενικών πόρων

Παρέχονται οι εξής επιλογές:

- Κάνοντας κλικ στο κουμπί "Delete," μπορείτε να διαγράψετε αμέσως έναν γενικό πόρο από το σύστημα.
- Κάνοντας κλικ στο κουμπί "Edit," μπορείτε να επεξεργαστείτε τα στοιχεία ενός πόρου μέσω της αντίστοιχης φόρμας.
- Πατώντας στην επικεφαλίδα δεξιά το κουμπί "Προσθήκη Γενικού πόρου," θα μεταφερθείτε στη φόρμα δημιουργίας νέου πόρου, όπου μπορείτε να εισάγετε νέα δεδομένα.

Στη φόρμα δημιουργίας νέου αξιοθέατου, πρέπει να προσθέσετε τα ακόλουθα δεδομένα:

- Όνομα: Εισαγωγή τίτλου του γενικού πόρου. Η εγγραφή του τίτλου θα γίνει στη γλώσσα που έχει επιλεγεί από τα settings.
- Περιγραφή: Καταχώρηση μιας σύντομης περιγραφής που επίσης θα είναι στη γλώσσα που έχει επιλεγεί από τα settings.
- Πόλη: Καταχώρηση της πόλης όπου βρίσκεται το μουσείο. Η καταγραφή της πόλης θα είναι στη γλώσσα που έχει επιλεγεί από τα settings.
- Κατηγορία: Καταχώρηση της κατηγορίας στατικού κειμένου. Η καταγραφή της κατηγορίας θα είναι στη γλώσσα που έχει επιλεγεί από τα settings.
- Link: Εισαγωγή ενός συνδέσμου (URL) που οδηγεί στην ιστοσελίδα που έχει περισσότερες πληροφορίες για το θέμα του κειμένου.
- Email: Καταχώρηση της διεύθυνσης email για επικοινωνία.
- Τηλέφωνο: Καταχώρηση του τηλεφώνου επικοινωνίας του πόρου. Π.χ. Το τηλέφωνο της αστυνομίας.
- Προσθήκη Στίγματος με Google Maps: Επισύναψη στίγματος της τοποθεσίας του μουσείου χρησιμοποιώντας το Google Maps.
- Φωτογραφία: Επισύναψη μιας φωτογραφίας που αντιπροσωπεύει το μουσείο.

Για να αποθηκευτούν τα στοιχεία που συμπληρώσατε, παρακαλείστε να πατήσετε το κουμπί "Αποθήκευση" στην επικεφαλίδα στο πάνω δεξιά μέρος της οθόνης.

4.4.13 Chatbot

4.4.13.1 Επισκόπηση

Στην καρτέλα "Chatbot", δίνεται η δυνατότητα επεξεργασίας του δενδροδιαλόγου του ψηφιακού βοηθού. Στο αριστερό μέρος του πλαισίου περιεχομένου, προβάλλονται οι προτροπές και οι επιλογές του χρήστη με την σειρά που επιλέγονται. Αρχικά, ο διάλογος ξεκινάει με μια προκαθορισμένη προτροπή από τον ψηφιακό βοηθό, όπως φαίνεται στην εικόνα.

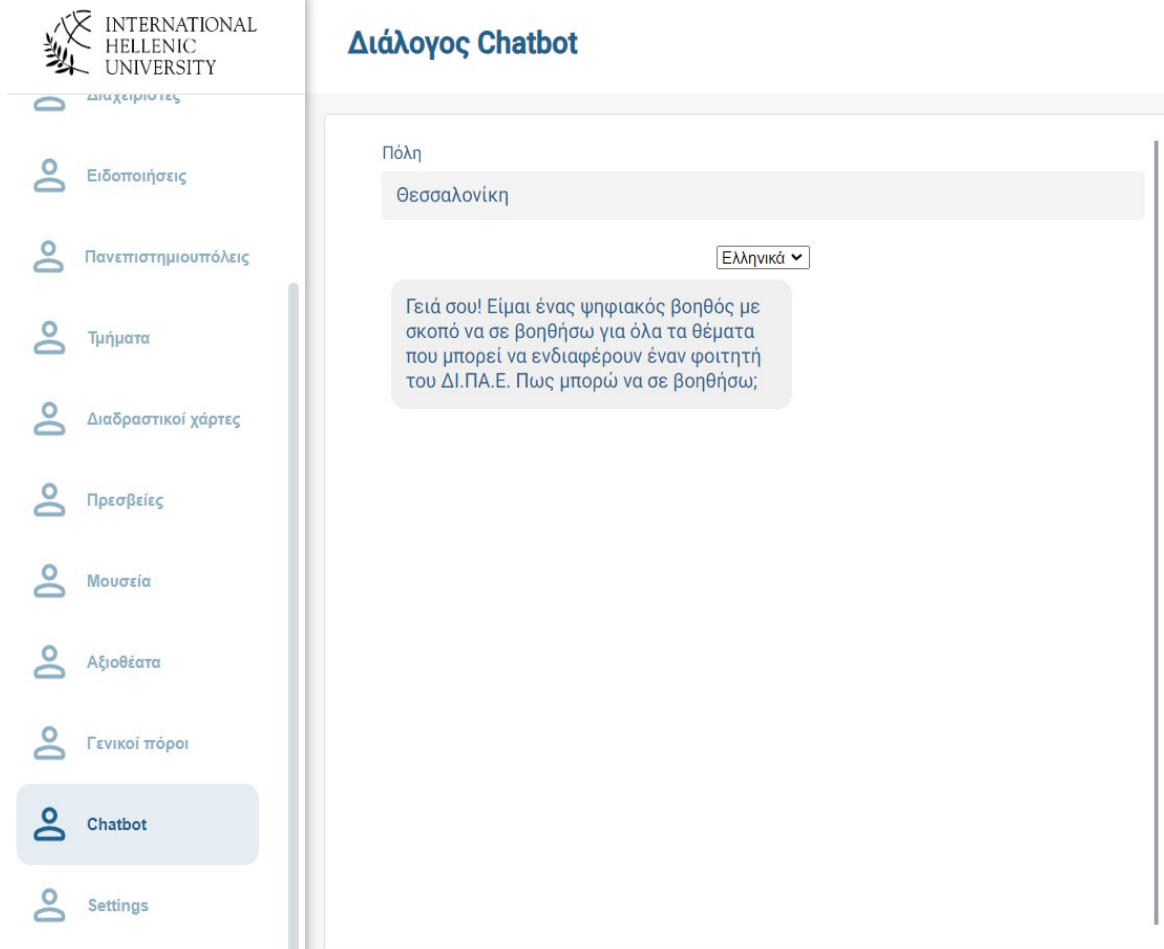


Figure 69: Έναρξη δένδροδιαλόγου

Στο πάνω μέρος του πλαισίου, εμφανίζεται η πόλη που έχει επιλεγμένη ο χρήστης, κάτι που βοηθάει στην προσομοίωση της ομιλίας με τον βοηθό, από την εφαρμογή κινητών συσκευών. Ακριβώς κάτω από το πεδίο επιλογής πόλης, βρίσκεται το πεδίο επιλογής γλώσσας, το οποίο αλλάζει την γλώσσα προβολής, για τον ίδιο σκοπό. Σε αυτό το σημείο, υπάρχουν 2 επιλογές.

4.4.13.2 Προσθήκη νέας επιλογής

Για την προσθήκη νέας επιλογής στο ισχύων επίπεδο του διαλόγου, μπορείτε να κάνετε κλικ στο κουμπί “Προσθήκη νέας επιλογής”. Με την ενέργεια αυτή, εμφανίζεται ένα παράθυρο με μία φόρμα επιλογής γλώσσας και εισαγωγής 2 κειμένων, ένα για το κείμενο που εκφράζει την επιθυμία του χρήστη και ένα για την επόμενη προτροπή του ψηφιακού βοηθού, σε περίπτωση επιλογής αυτής.

Κατόπιν αποθήκευσης αυτού του κόμβου διαλόγου, η επιλογή αυτή γίνεται διαθέσιμη στο τρέχον επίπεδο του δένδροδιαλόγου.

Στο δεξιό μέρος του πλαισίου περιεχομένων, εμφανίζονται οι πιθανές επιλογές του χρήστη ή, στην περίπτωση που το δέντρο του διαλόγου έχει φτάσει στο τέλος του, τα αποτελέσματα της συνομιλίας.

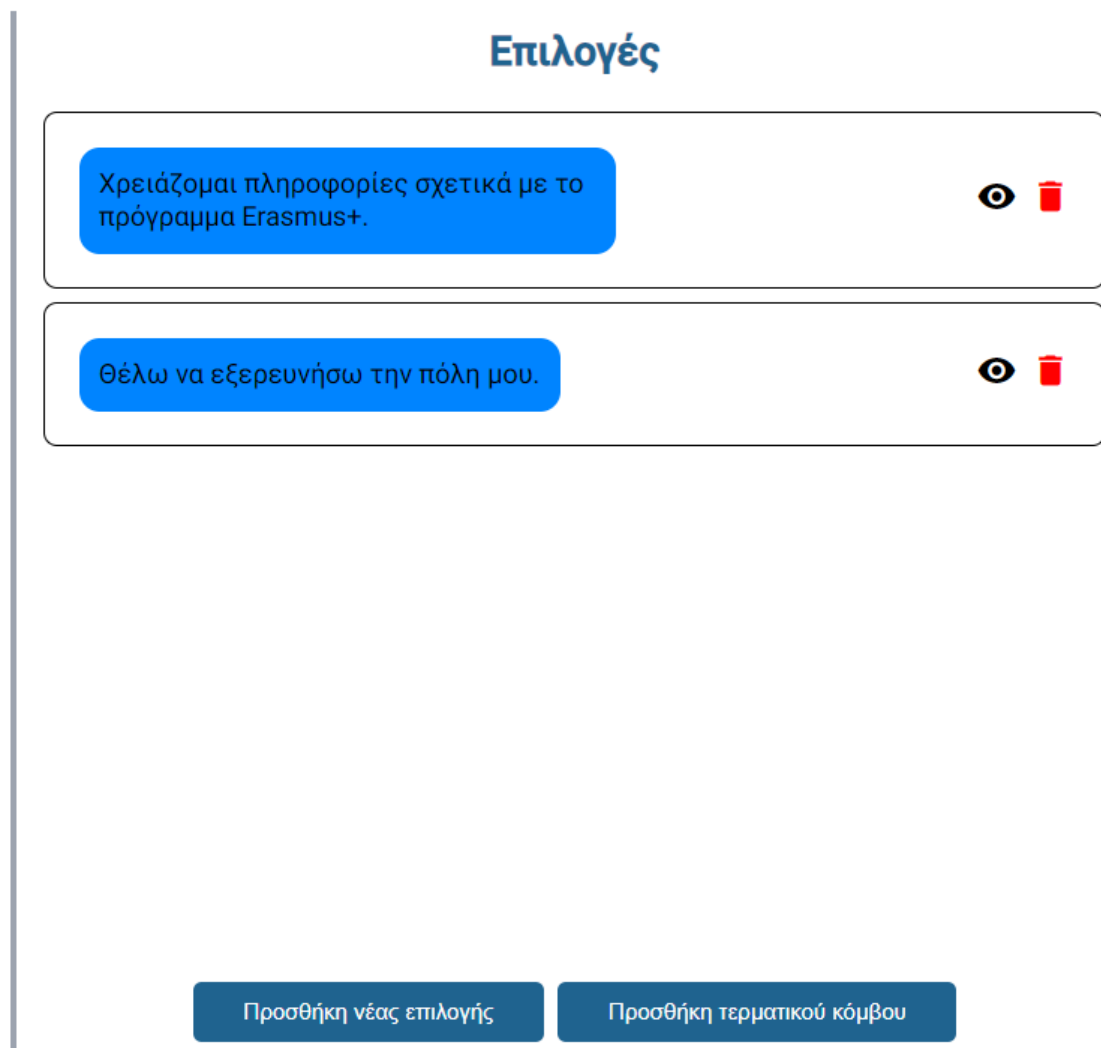


Figure 70: Προβολή μηνυμάτων του δενδροδιαλόγου

Κάθε επιλογή χρήστη ακολουθείται από 2 εικονίδια. Κάνοντας κλικ στο εικονίδιο του ματιού, ο διάλογος προχωράει με την επιλογή αυτή, το κείμενο της οποίας εμφανίζεται στο αριστερό πλαίσιο μαζί με την επόμενη προτροπή διαλόγου από την μεριά του ψηφιακού βοηθού. Κάνοντας κλικ στον κάδο, διαγράφεται αυτό το μονοπάτι διαλόγου από την βάση δεδομένων.

4.4.13.3 Προσθήκη τερματικού κόμβου

Εφόσον ένα μονοπάτι διαλόγου έχει οριστεί και είμαστε ικανοποιημένοι με την πληροφορία που παράγει ώστε να φτάσει στο τέλος του και να προβάλουμε τα θεμιτά αποτελέσματα, επιλέγουμε “Προσθήκη τερματικού κόμβου”, όπου εμφανίζεται ένα παρόμοιο παράθυρο εισαγωγής κειμένου για τον χρήστη και η απάντηση του ψηφιακού βοηθού, μαζί με κάποια πεδία παραμετροποίησης των αποτελεσμάτων.

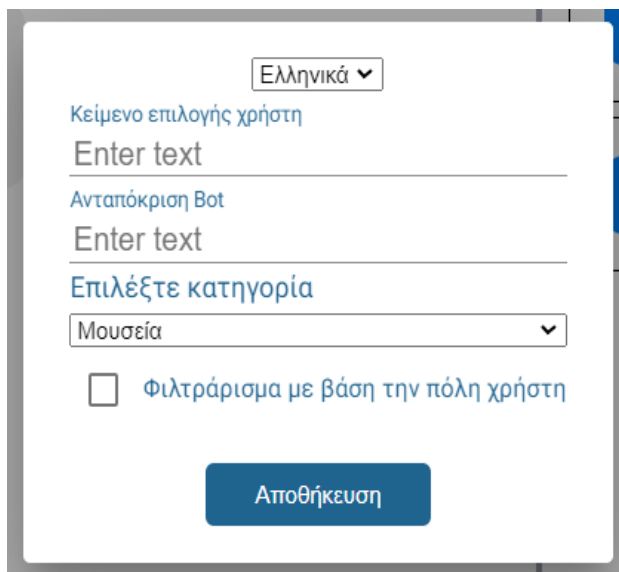


Figure 71: Προσθήκη τελικού κόμβου

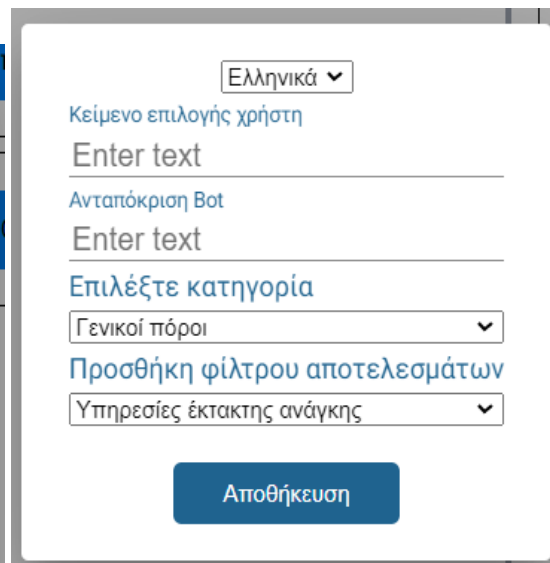


Figure 72: Προσθήκη τελικού κόμβου με κατηγορία γενικοί πόροι

Η πρώτη απαραίτητη ενέργεια είναι η επιλογή κατηγορίας αποτελεσμάτων. Για παράδειγμα, μία από τις κατηγορίες είναι η “Μουσεία”. Επιλέγοντας αυτήν τα αποτελέσματα που θα εμφανιστούν θα είναι όλες οι πληροφορίες των μουσείων στην βάση δεδομένων. Στην περίπτωση που θέλουμε τα αποτελέσματα να φιλτράρονται αυτόματα με βάση την επιλεγμένη πόλη του χρήστη, αρκεί να επιλέξουμε το κουτάκι “Φιλτράρισμα με βάση την πόλη του χρήστη”. Σε άλλες κατηγορίες, όπως στις “Στατικά κείμενα” ή “Γενικοί πόροι”, δίνεται η δυνατότητα επιλογής υποκατηγορίας (βλ. 4.4.14.3).

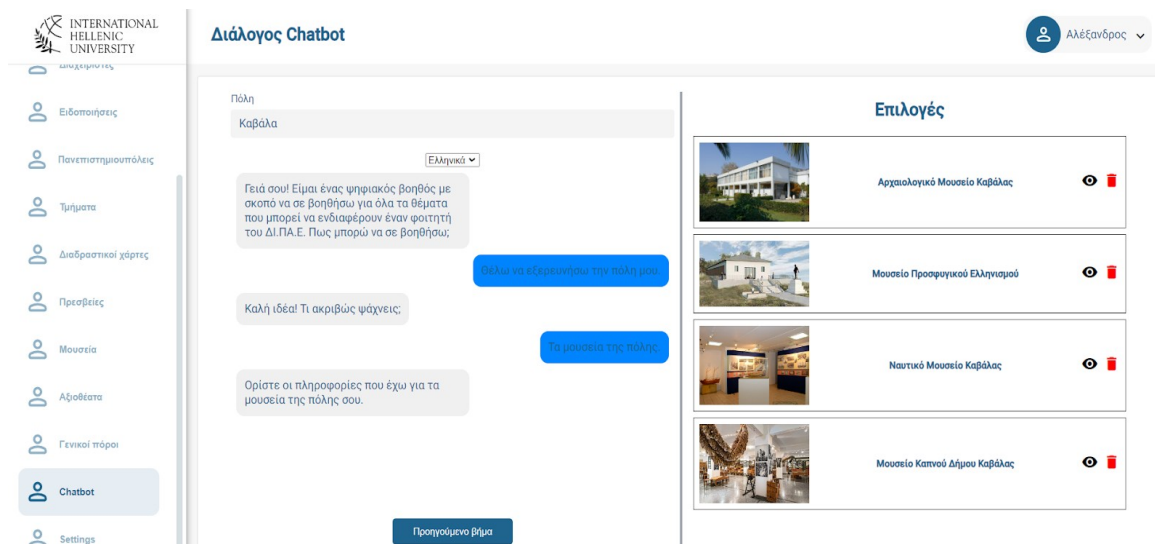


Figure 73: Προβολή του αποτελέσματος του τελικού κόμβου του δενδροδιαλόγου

4.4.14 Settings

Στις ρυθμίσεις (settings), υπάρχουν τρεις καρτέλες: “Γλώσσες”, “Πόλεις” και “Κατηγορίες”.

4.4.14.1 Γλώσσες

Στην καρτέλα "Γλώσσες," υπάρχουν δύο στήλες:

- Διαθέσιμες Γλώσσες:
 - Περιέχει μια λίστα με όλες τις διαθέσιμες γλώσσες.
 - Εάν κάνετε κλικ σε μια από τις διαθέσιμες γλώσσες, αυτή προστίθεται στη λίστα των επιλεγμένων γλωσσών.
- Επιλεγμένες Γλώσσες:
 - Περιέχει τις γλώσσες που έχουν επιλεγεί για χρήση στην εφαρμογή.
 - Εάν κάνετε κλικ σε μια από τις επιλεγμένες γλώσσες, αυτή αφαιρείται από τις επιλεγμένες και προστίθεται στις διαθέσιμες γλώσσες.

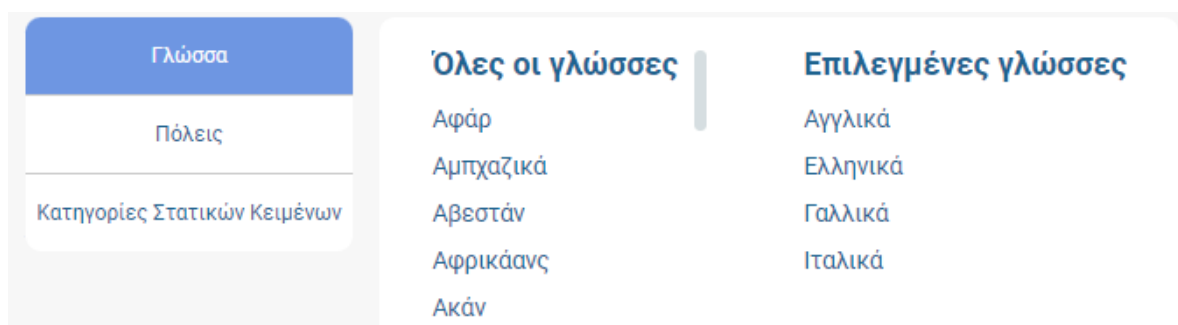


Figure 74: Προσθήκη και διαγραφή γλωσσών στο σύστημα

4.4.14.2 Πόλεις

Στην καρτέλα "Πόλεις," μπορείτε να προσθέσετε, να επεξεργαστείτε πόλεις σε όλες τις γλώσσες που έχουν επιλεχθεί και να διαγράψετε πληροφορίες πόλεων. Ο κωδικός της πόλης δημιουργείται αυτόματα και δεν μπορείτε να τον τροποποιήσετε.



Figure 75: Φόρμα πόλης

Για να προσθέσετε πόλη πρέπει να σκρολάρετε κάτω και να πατήσετε κάτω δεξιά το κουμπί προσθήκη.

4.4.14.3 Κατηγορίες στατικών κειμένων

Στην καρτέλα "Κατηγορίες στατικών κειμένων," μπορείτε να προσθέσετε, να επεξεργαστείτε κατηγορίες σε όλες τις γλώσσες που έχουν επιλεγεί και να διαγράψετε πληροφορίες κατηγοριών. Ο κωδικός της κατηγορίας δημιουργείται αυτόματα και δεν μπορείτε να τον τροποποιήσετε.



The screenshot shows a form with two input fields and a dropdown menu. The first field is labeled "Όνομα κατηγορίας" (Category Name) and contains the text "Υπηρεσίες έκτακτης ανάγκης" (Emergency Services). The second field is labeled "Κωδικός κατηγορίας" (Category Code) and contains the alphanumeric string "79LIAiK2RppmdemQ92fE". To the right of these fields is a dropdown menu labeled "Γλώσσα" (Language). The dropdown is currently open, showing three options: "el" (selected with a checkmark), "fr", and "en". A red trash icon is visible next to the "el" option in the dropdown.

Figure 76: Φόρμα κατηγορίας στατικού κειμένου

Για να προσθέσετε πόλη πρέπει να κυλίσετε την σελίδα προς τα κάτω και να πατήσετε κάτω δεξιά το κουμπί προσθήκη.

Ότι αλλαγή και να κάνετε στις πόλεις και στις γλώσσες και στις κατηγορίες για να γίνει αποθήκευση παρακαλείστε να πατήσετε το κουμπί "Αποθήκευση" στην επικεφαλίδα στο πάνω δεξιά μέρος της οθόνης.

Κεφάλαιο 5ο: Συμπεράσματα ή/και προτάσεις βελτίωσης

5.1 Συμπεράσματα

Η υλοποίηση της εφαρμογής για τους φοιτητές Erasmus+ του ΔΙ.ΠΑ.Ε αποτελεί έναν πολύτιμο πόρο για την κοινότητα φοιτητών που συμμετέχουν στο πρόγραμμα ανταλλαγής. Προσφέρει μια πρακτική λύση για τη βελτίωση της καθημερινής ζωής των φοιτητών Erasmus+, προσφέροντας εξειδικευμένες πληροφορίες, διαχείριση ρυθμίσεων και επικοινωνία. Οι δυνατότητες που προσφέρονται στους χρήστες, όπως η προσαρμογή των ρυθμίσεων για γλώσσα, πόλη και πανεπιστημιούπολη, η δυνατότητα πλοήγησης σε διαδραστικούς χάρτες και η ενημέρωση μέσω ανακοινώσεων τους ενισχύουν τη χρηστικότητα και την ευκολία χρήσης της εφαρμογής.

Το διαχειριστικό περιβάλλον που υλοποιήθηκε σε Angular, παρέχει ένα αποτελεσματικό και φιλικό περιβάλλον για τους διαχειριστές, επιτρέποντας τους να δημιουργούν και να επεξεργάζονται περιεχόμενο το οποίο θα είναι χρήσιμο για τους χρήστες της εφαρμογής, καθώς και να δημιουργούν ανακοινώσεις και να παρέχουν συνεχώς ενημερώσεις.

Αναγνωρίζουμε πως το φάσμα των φοιτητών που συμμετέχουν στο πρόγραμμα Erasmus, συνεπώς και οι ανάγκες και προτιμήσεις αυτών, είναι ιδιαίτερα ευρύ. Ο σχεδιασμός και η ανάπτυξη μίας πραγματικά ολοκληρωμένης εφαρμογής, για όλους τους φοιτητές είναι ιδιαίτερα μεγάλο έργο. Πιστεύουμε πως οι εφαρμογές που αναπτύξαμε καταφέρνουν να παρέχουν μία σημαντική πλατφόρμα ώστε να εξυπηρετηθεί ένα σημαντικό μερίδιο των βασικών και κοινών αναγκών, και αναπτύχθηκαν με την φιλοσοφία της εύκολης επεκτασιμότητας. Καταβάλαμε σημαντική προσπάθεια ώστε να σχεδιάσουμε και να υλοποιήσουμε αρχιτεκτονικά εύκαμπτες εφαρμογές, καθώς πιστεύουμε πως είναι πιθανό να ενεργήσουν ως θεμέλιο για μία ακόμα πιο ισχυρή εφαρμογή στο μέλλον. Με αυτή την σκέψη, παραθέτουμε τις προτάσεις μας για βελτιώσεις, επεκτάσεις και εναλλακτικές λύσεις.

5.2 Προτάσεις

Σε περαιτέρω εξέλιξη του έργου, θα ήταν επωφελές να εξεταστεί η δυνατότητα επέκτασης/διαμόρφωσης των παρακάτω λειτουργιών των εφαρμογών.

- Ψηφιακός βοηθός: Χωρίς ενεργούς χρήστες στην εφαρμογή ή κάποια κατάταξη προτιμήσεων/αναγκών από τους φοιτητές που θα ευνοηθούν από το έργο μας, λάβαμε την απόφαση να ορίσουμε κάποιες οντότητες ως τις κύριες κατηγορίες αποτελεσμάτων της συνομιλίας με τον βοηθό. Για την επίδειξη της εύκολης επεκτασιμότητας της λειτουργίας του ψηφιακού βοηθού, προσθέσαμε τις κατηγορίες στατικών κειμένων και πόρων. Έτσι, οι διαχειριστές μπορούν να παράγουν προσαρμοσμένες κατηγορίες στις οποίες υπάγονται οι γενικοί πόροι και τα στατικά κείμενα που δημιουργούν. Οι κατηγορίες μπορούν, εφόσον κριθεί μελλοντικά συνετό, να επεκταθούν με την δημιουργία υποκατηγοριών, και όλες οι οντότητες που μπορούν να δράσουν ως αποτελέσματα συνομιλίας με τον ψηφιακό βοηθό να υπαχθούν στην κλάση των γενικών πόρων, συμπεριλαμβάνοντας έτσι την λειτουργικότητα. Κάτι τέτοιο θα βελτιώσει την ευχρηστία της εφαρμογής, θα μειώσει τον πήχη κατανόησης της αρχιτεκτονικής από μη τεχνικά καταρτημένους χρήστες και θα απομονώσει την αρχιτεκτονική του ψηφιακού βοηθού σημαντικά από την υπόλοιπη εφαρμογή.

- Διαδραστικοί χάρτες: Οι χάρτες διευκολύνουν ιδιαίτερα την πλοήγηση των πανεπιστημιούπολεων και την εξοικείωση των φοιτητών με αυτήν. Μία σημαντική επέκταση που θα μπορούσε να υλοποιηθεί είναι η ψηφιακή ξενάγηση των δομών του πανεπιστημίου.
- Ειδοποιήσεις: Καθώς η εφαρμογές μας αξιοποιούν το δωρεάν πακέτο υπηρεσιών της σουίτας Firebase, δεν ήταν εφικτό να αναπτύξουμε λειτουργικότητες που απαιτούν έναν αποκλειστικό διακομιστή, όπως παρακολούθηση συμβάντων στην βάση δεδομένων για την αποστολή ειδοποιήσεων (Push notifications) στην εφαρμογή κινητών συσκευών. Εφόσον υπάρχει δυνατότητα παροχής ενός αποκλειστικού διακομιστή, ή μίας αμελητέας χρηματοδότησης η εφαρμογή θα επωφεληθεί ιδιαίτερα.
- Ανακοινώσεις: Κατά τις συζητήσεις μας με καθηγητές και συμφοιτητές, μάθαμε πως πολλά τμήματα δεν έχουν διαδικτυακές πύλες για να ενημερώνονται για σπουδαστικά θέματα οι φοιτητές αυτών, σε αντίθεση με το τμήμα Ηλεκτρολόγων και Μηχανικών Πληροφορικής (apps/aboard), αλλά χρησιμοποιούν τους πίνακες ανακοινώσεων του moodle για την ενημέρωσή τους. Εφόσον το πανεπιστήμιο παρέχει έναν διακομιστή με σταθερή διεύθυνση IP, μπορεί να αναπτυχθεί η δυνατότητα σύνδεσης με τα στοιχεία του moodle, καθιστώντας δυνατή την ανάπτυξη λειτουργικότητας προβολής των πινάκων ανακοινώσεων, στους οποίους εγγράφονται οι φοιτητές, διαθέσιμους στην εφαρμογή των κινητών.
- Ενίσχυση της αλληλεπίδρασης μεταξύ των χρηστών, όπως η δημιουργία φόρουμ ή η δυνατότητα ανταλλαγής εμπειριών, θα συμβάλλει στη δημιουργία μιας περαιτέρω προσβάσιμης κοινότητας.

Συνοψίζοντας, η εφαρμογή αποτελεί ένα πολύτιμο εργαλείο που συμβάλλει στην επιτυχημένη οργάνωση και προώθηση της εκπαιδευτικής εμπειρίας των φοιτητών Erasmus+ του ΔΙ.ΠΑ.Ε.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Internet Site

- [1] Chinmayee Deshpande, (Jul 24, 2023) What is Angular? Architecture, Features, and Advantages. Accessed on: Nov. 2, 2015. [Online]. Available:<https://www.simplilearn.com/tutorials/angular-tutorial/what-is-angular>
- [2] Angular Documentation [Online]. Available:<https://angular.io/guide/built-in-directives>
- [3] ReviverSoft. .HTML Επέκταση αρχείου. Accessed on: Nov. 11, 2023. [Online]. Available: <https://www.reviversoft.com/el/file-extensions/html>.
- [4] IPGlobal IKE. Τι είναι η HML. Accessed on: Nov. 11, 2023. [Online]. Available: https://www.ip.gr/Web_Development/%CF%84%CE%B9-%CE%B5%CE%AF%CE%BD%CE%B1%CE%B9-%CE%B7-html-61.html
- [5] Μιχαήλια Κομβούτη – Βέρου. Εισαγωγή στο css. Accessed on: Nov. 11, 2023. [Online]. Available: http://pages.cs.aueb.gr/courses/epl131/files/CSS_notes.pdf
- [6] Pluralsight SASS Documentation Accessed on: Nov. 11, 2023. [Online]. Available:<https://www.pluralsight.com/guides/use-sass-less-or-css-within-your-angular-component-template>
- [7] FileFormat SCSS Documentation Accessed on: Nov. 11, 2023. [Online]. Available:<https://docs.fileformat.com/web/scss/>
- [8] Firebit, (Feb 23, 2017). Τι είναι η Typescript? Accessed on: Nov. 11, 2023. [Online]. Available: <https://medium.com/@firebit/%CF%84%CE%B9-%CE%B5%CE%AF%CE%BD%CE%B1%CE%B9-%CE%B7-typescript-5da6935041f7>
- [9] Hiren Dhaduk, (November 28, 2022). Best Frontend Frameworks for Web Development in 2023. Accessed on: Nov. 11, 2023. [Online]. Available: <https://www.simform.com/blog/best-frontend-frameworks/>
- [10] Testim, (September 03, 2021). Angular Component Testing: A Detailed How-To With Examples. Accessed on: Nov. 11, 2023. [Online]. Available: <https://www.testim.io/blog/angular-component-testing-detailed-guide/#:~:text=component-,spec.,spec.>
- [11] JavaTpoint. What is an Angular Service. Accessed on: Nov. 18, 2023. [Online]. Available: <https://www.javatpoint.com/what-is-an-angular-service>
- [12] TekTutorialsHub (March 9, 2023). Intoduction to Angular Services. Accessed on: Nov. 18, 2023. [Online]. Available: <https://www.tektutorialshub.com/angular/angular-services/>.
- [13] Rangle.io. What is an Angular Module. Accessed on: Nov. 18, 2023. [Online]. Available: <https://angular-training-guide.rangle.io/modules/introduction>.
- [14] Makesh Kumar, (Apr 20, 2022). What are Directives in Angular. Accessed on: Nov. 18, 2023. [Online]. Available: <https://blog.bitsrc.io/directives-in-angular-6160ce805416>
- [15] W3resource, (August 19 2022). Attribute Directives. Accessed on: Nov. 18, 2023. [Online]. Available: <https://www.w3resource.com/angular/attributes-directives.php#:~:text=Angular%20attribute%20directives%20are%20a,behavior%20of%20a%20DOM%20element.>

- [16] Rajkumar M G, (September 23, 2019). Components vs Directives in Angular with examples. Accessed on: Nov. 18, 2023. [Online]. Available: <https://www.agiratech.com/components-vs-directives-in-angular#:~:text=Component%20directive%20is%20used%20to,directive%20in%20any%20Angular%20project.>
- [17] Debasis Saha, (Jul 23, 2018). Life Cycle of Angular Components. Accessed on: Nov. 18, 2023. [Online]. Available: <https://www.c-sharpcorner.com/article/life-cycle-of-angular-components/>
- [18] TekTutorialsHub, (March 9, 2023). Angular ngOnChanges Life Cycle Hook. Accessed on: Nov. 18, 2023. [Online]. Available: <https://www.tektutorialshub.com/angular/angular-ngonchanges-life-cycle-hook/#:~:text=The%20ngOnChnages%20is%20a%20life,current%20and%20previous%20property%20values.&text=The%20child%20Component%20decorates%20the%20property%20using%20the%20%40Input%20decorator.>
- [19] StackChief LLc, (Dec 10, 2023). NgOnInit Example | Angular Accessed on: Nov. 18, 2023. [Online]. Available: <https://www.stackchief.com/blog/ngOnInit%20Example%20%7C%20Angular>
- [20] inDepth Angular Tutorial Accessed on: Nov. 11, 2023. [Online]. Available:<https://indepth.dev/tutorials/angular/track-changes-in-ngdocheck-with-keyvaluediffers>
- [21] Chidume Nnamdi, (Mar 26, 2021). Angular Lifecycle hooks explained. Accessed on: Nov. 25, 2023. [Online]. Available: <https://blog.logrocket.com/angular-lifecycle-hooks/>
- [22] Angular Router Guide Accessed on: Nov. 11, 2023. [Online]. Accessed on: Nov. 11, 2023. [Online]. Available:<https://angular.io/guide/router>
- [23] Nwose Lotanna Victor, (March 08, 2022). Angular Basics: CanActivate - Introduction to Routing Guards. Accessed on: Nov. 25, 2023. [Online]. Available: <https://www.telerik.com/blogs/angular-basics-canactivate-introduction-routing-guards>
- [24] Jessica Clark. What is Firebase Authentication? Accessed on: Nov. 25, 2023. [Online]. Available: <https://blog.back4app.com/firebase-authentication/>
- [25] Doug Stevenson, (Sep 25, 2018). What is firebase? The Complete Story, abridged. Accessed on: Nov. 25, 2023. [Online]. Available: <https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0>
- [26] JavaTpoint: Firebase: Realtime Database. Accessed on: Nov. 25, 2023. [Online]. Available: <https://www.javatpoint.com/firebase-realtime-database>
- [27] JavaTpoint. Firebase: Firestore. Accessed on: Nov. 25, 2023. [Online]. Available: <https://www.javatpoint.com/firebase-firestore>
- [28] JavaTpoint. Firebase: Cloud Storage. Accessed on: Nov. 25, 2023. [Online]. Available: <https://www.javatpoint.com/firebase-cloud-storage>
- [29] JavaTpoint. Firebase: Hosting. Accessed on: Nov. 25, 2023. [Online]. Available: <https://www.javatpoint.com/firebase-hosting.>
- [30] Adservio group, (July 20, 2022). What is Flutter and its Advantage. Accessed on: Nov. 25, 2023. [Online]. Available: <https://www.adservio.fr/post/what-is-flutter-and-what-are-its-advantages#:~:text=Flutter%20is%20a%20mobile%20app,and%20easier%20than%20traditional%20methods.>

[31] Firebase. Cloud Functions For Firebase. Accessed on: Nov. 25, 2023. [Online]. Available: <https://firebase.google.com/docs/functions>.

[32] Jacksonmateo, (Sep 20, 2023). Key characteristics of flutter development. Accessed on: Nov. 25, 2023. [Online]. Available: <https://medium.com/@jacksonmateo625/key-characteristics-of-flutter-application-development-63e10f5d2d33#:~:text=Flutter's%20key%20characteristics%2C%20such%20as,relia%20Flutter%20application%20development%20company>.

[33] GünseliÜnsal,(Jul 2, 2023). Stateless vs Stateful Widgets in Flutter. Accessed on: Nov. 25, 2023. [Online]. Available: <https://medium.com/@gunseliunsal/stateless-vs-stateful-widgets-in-flutter-852741b6046e>