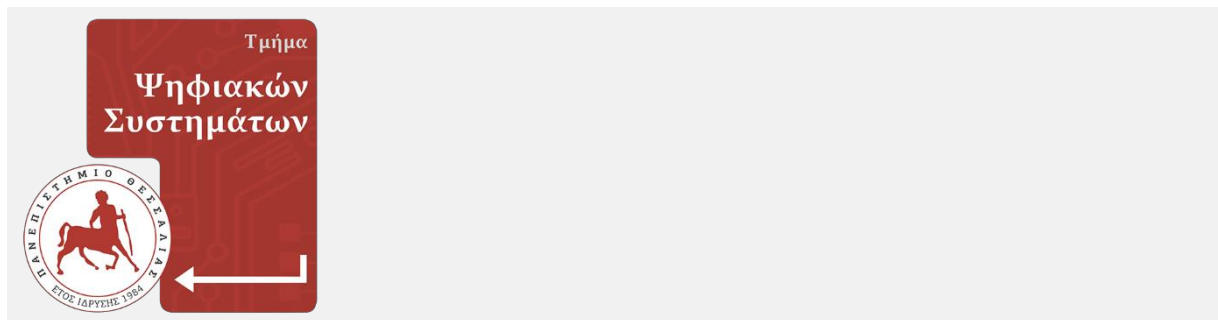




ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Σχολή Τεχνολογίας

Τμήμα Ψηφιακών Συστημάτων

Ανάπτυξη Εφαρμογής Ανταλλαγής Μηνυμάτων Σε Πραγματικό Χρόνο

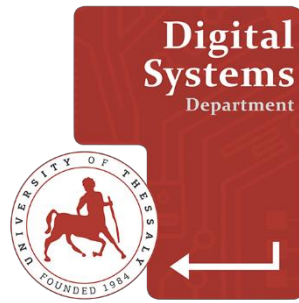
ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Αγορίτσα Μπινιάρη

(ΑΜ: 3119024)

Επιβλέπων: Γεώργιος Κακαρόντζας, Καθηγητής

ΛΑΡΙΣΑ, Μάρτιος 2024



UNIVERSITY OF THESSALY

School of Technology

Digital Systems Department

Development Of A Real-Time Messaging Application

BACHELOR THESIS

Agoritsa Biniari

(RN: 3119024)

Supervisor: *Georgios Kakarontzas, Professor*

LARISSA, March 2024

Υπεύθυνη Δήλωση περί Ακαδημαϊκής Δεοντολογίας και Πνευματικών Δικαιωμάτων

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα πτυχιακή εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον, και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο/Η Δηλών/ούσα

(Υπογραφή)

Αγορίτσα Μπινιάρη

Ημερομηνία

20-03-2024

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπων/πουσα	Γεώργιος Κακαρόντζας Καθηγητής, Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Θεσσαλίας
Μέλος	Αγορίτσα Μπινιάρη Φοιτήτρια, Τμήμα Ψηφιακών Συστημάτων/Πανεπιστήμιο Θεσσαλίας

Ημερομηνία έγκρισης: 31/10/2022

Περίληψη

Το θέμα της εργασίας πραγματεύεται την δημιουργία μίας εφαρμογής ανταλλαγής μηνυμάτων σε πραγματικό χρόνο χρησιμοποιώντας σύγχρονες τεχνολογίες ανάπτυξης εφαρμογών όπως το Angular Framework και υπηρεσίες της πλατφόρμας Firebase.

Ο στόχος ήταν να δημιουργηθεί μια σύγχρονη εφαρμογή, εύχρηστη και φιλική προς τον χρήστη, που ενσωματώνει τις βασικές λειτουργίες, όμοιες με αυτές που παρέχουν οι δημοφιλείς εφαρμογές ανταλλαγής μηνυμάτων. Για την επίτευξη του εγχειρήματος αυτού έγινε ενδελεχής έρευνα των εργαλείων και των πλεονεκτημάτων που προσφέρουν λαμβάνοντας υπόψιν ωστόσο και τα πιθανά μειονεκτήματα.

Ακόμα αναλύονται όλα τα πρότυπα της αρχιτεκτονικής και των πρακτικών που ακολουθήθηκαν. Γίνεται εκτενέστερη περιγραφή για όλη τη διαδικασία ανάπτυξης από την εγκατάσταση των εργαλείων και την δημιουργία του περιβάλλοντος ανάπτυξης σε τοπικό υπολογιστή, μέχρι την κατασκευή του τελικού προϊόντος και την δημοσίευσή του στον ιστό.

Η τελική έκβαση του εγχειρήματος ήταν επιτυχής καθώς κατασκευάστηκε μια λειτουργική εφαρμογή εύκολη στην χρήση η οποία παρέχει όλες τις δυνατότητες που σχεδιάστηκαν στην αρχή.

Η εφαρμογή δημοσιεύτηκε και είναι προσβάσιμη για τον αναγνώστη μέσω του παγκόσμιου ιστού. Έτσι, ο αναγνώστης μπορεί να γίνει χρήστης της εφαρμογής μας, καθώς έχει την δυνατότητα να αλληλεπιδράσει με την εφαρμογή και να αποκτήσει μια βιωματική εμπειρία διαπιστώνοντας στην πράξη όλα όσα περιγράφονται.

Abstract

The subject of the project deals with the creation of a real-time messaging application using modern application development technologies such as the Angular Framework and Firebase platform services. The goal was to create a modern, user-friendly application that incorporates basic functionalities similar to the ones provided by popular messaging applications. To achieve this project, a thorough investigation of the tools and advantages they offer was conducted, considering possible disadvantages as well.

Furthermore, all architectural standards and practices followed are analyzed. A detailed description is provided for the entire development process, from installing the tools and setting up the development environment on a local computer to building the final product and publishing it on the web.

The outcome of the project was successful, as a functional and user-friendly application was developed, providing all the capabilities originally planned. The application was published and is accessible to the reader via the World Wide Web. Thus, the reader can become a user of our application, as they have the opportunity to interact with it and gain an experiential understanding of everything that is described.

Ευχαριστίες

Θα ήθελα να εκφράσω τις ευχαριστίες μου στον επιβλέποντα καθηγητή μου, κ. Κακαρόντζα Γεώργιο, καθώς και στο υπόλοιπο διδακτικό προσωπικό για τη πολύτιμη βοήθεια τους .

Περισσότερο από όλους, οφείλω να ευχαριστήσω την οικογένεια μου, τον παππού και τις γιαγιάδες μου διότι χωρίς εκείνους η απόκτηση ενός πτυχίου θα ήταν αδύνατη ή έστω, πολύ δύσκολο εγχείρημα.

Τους ευχαριστώ που στάθηκαν αρωγοί για την πραγματοποίηση του στόχου μου !

Αγορίτσα Μπινιάρη

Μάρτιος 2024

Περιεχόμενα

ΠΕΡΙΛΗΨΗ.....	VII
ABSTRACT	IX
ΕΥΧΑΡΙΣΤΙΕΣ	XI
ΠΕΡΙΕΧΟΜΕΝΑ	XIII
1 ΕΙΣΑΓΩΓΗ	1
2 ΕΦΑΡΜΟΓΗ ΑΝΤΑΛΛΑΓΗΣ ΜΗΝΥΜΑΤΩΝ	3
2.1 ΕΠΙΚΟΙΝΩΝΙΑ ΜΕΣΩ ΔΙΑΔΙΚΤΥΟΥ	3
2.2 ΕΦΑΡΜΟΓΕΣ ΑΝΤΑΛΛΑΓΗΣ ΜΗΝΥΜΑΤΩΝ ΣΤΗ ΣΥΓΧΡΟΝΗ ΚΟΙΝΩΝΙΑ	8
3 ΚΑΤΑΣΚΕΥΗ ΕΦΑΡΜΟΓΗΣ.....	13
3.1 ΠΕΡΙΓΡΑΦΗ ΑΝΤΙΚΕΙΜΕΝΟΥ	13
3.2 ΤΕΧΝΟΛΟΓΙΕΣ & ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ	14
3.2.1 <i>Angular Framework - Εγκατάσταση Angular σε τοπικό υπολογιστή</i>	15
3.2.2 <i>Firebase BaaS Εγκατάσταση Εργαλείων του Firebase σε τοπικό υπολογιστή / console.....</i>	17
4 ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ.....	23
4.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΕΦΑΡΜΟΓΗΣ.....	23
4.2 ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΔΙΕΠΑΦΗΣ ΧΡΗΣΤΗ.....	25
4.2.1 <i>Επιχειρηματική Λογική και Services</i>	29
4.2.2 <i>Ενότητες και Δομικά Στοιχεία</i>	32
4.3 ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ FIREBASE	39
5 ΛΕΙΤΟΥΡΓΙΕΣ ΕΦΑΡΜΟΓΗΣ	45
5.1 ΔΥΝΑΤΟΤΗΤΕΣ ΧΡΗΣΤΗ	45
5.2 ΔΗΜΟΣΙΕΥΣΗ ΕΦΑΡΜΟΓΗΣ	50
6 ΣΥΜΠΕΡΑΣΜΑΤΑ.....	53
ΒΙΒΛΙΟΓΡΑΦΙΑ	55

ΠΑΡΑΡΤΗΜΑ	57
-----------------	----

1 Εισαγωγή

Στην κοινωνία της πληροφορίας στην οποία διαβιούμε, το διαδίκτυο και οι δυνατότητες που προσφέρει, διεκπεραιώνουν πληθώρα καθημερινών αναγκών. Μία βασική και από τις πιο σημαντικές δυνατότητες που παρέχει το διαδίκτυο είναι η επικοινωνία, η οποία επιτυγχάνεται μέσω των κατάλληλων εφαρμογών.

Το αντικείμενο της παρούσας πτυχιακής εργασίας είναι ο σχεδιασμός, η ανάπτυξη και η υλοποίηση μίας εφαρμογής άμεσης ανταλλαγής μηνυμάτων πραγματικού χρόνου.

Η εφαρμογή παρέχει βασικές δυνατότητες στους χρήστες, όπως την δημιουργία λογαριασμού χρήστη, την δυνατότητα σύνδεσης με άλλους χρήστες της εφαρμογής και την ανταλλαγή μηνυμάτων μεταξύ τους, σε πραγματικό χρόνο.

Η τεχνολογία που χρησιμοποιήθηκε για το κομμάτι της διεπαφής χρήστη είναι το Angular Framework. Επίσης, καθώς απαιτείται βάση δεδομένων πραγματικού χρόνου για την ανταλλαγή των μηνυμάτων και την αυθεντικοποίηση των χρηστών, χρησιμοποιήθηκε η πλατφόρμα BaaS (Backend As A Service) της Google Firebase.

Τέλος, για versioning management χρησιμοποιήθηκε Git και GitHubg ενώ το deployment και το hosting έγιναν επίσης στον server του Firebase.

2 Εφαρμογή Ανταλλαγής Μηνυμάτων

Η εποχή μας καθορίζεται σε μεγάλο βαθμό από την αλλαγή στους τρόπους επικοινωνίας μεταξύ των ανθρώπων. Αυτό συμβαίνει λόγω της εισβολής του διαδικτύου σε όλους τους τομείς της ζωής μας, άρα και στην επικοινωνία με τις πλατφόρμες ανταλλαγής μηνυμάτων.

2.1 Επικοινωνία μέσω διαδικτύου

Η διαδικτυακή επικοινωνία έχει αλλάξει άρδην τον τρόπο με τον οποίον επικοινωνούν άνθρωποι, οργανισμοί και κοινότητες μεταξύ τους. Εδώ θα εξετάσουμε την εξέλιξη της διαδικτυακής επικοινωνίας, ερευνώντας την προέλευση της και τα αποτελέσματα που έχει στη ζωή μας.

Η επικοινωνία μέσω διαδικτύου προήλθε από την ανάπτυξη του ARPANET (Advanced Research Projects Agency Network) τα τέλη της δεκαετίας του 1960 από το Υπουργείο Άμυνας των Ηνωμένων Πολιτειών. Το ARPANET δημιουργήθηκε για τη διευκόλυνση της επικοινωνίας και της ανταλλαγής δεδομένων μεταξύ ιδρυμάτων ερευνητικών και κρατικών φορέων. Ήταν ο πρόδρομος του σύγχρονου Διαδικτύου και έθεσε τα θεμέλια για την διαδικτυακή επικοινωνία.

Με το πέρασμα του χρόνου, οι αλλαγές που έγιναν στη διαδικτυακή επικοινωνία ήταν ριζικές όσο και σημαντικές σε σχέση με την τεχνολογία αλλά και με τον τρόπο που επικοινωνούσαν οι άνθρωποι. Μπορούμε να δούμε μερικές αλλαγές και να συγκρίνουμε τους διαδικτυακούς τρόπους επικοινωνίας σε σχέση με τις αρχικές προσπάθειες αυτού του θέματος.

Παλαιότερα, η ταχύτητα του διαδικτύου ήταν πολύ πιο αργή, σε σχέση με σήμερα. Αυτό οδηγούσε σε μεγάλους χρόνους φόρτωσης αποστολής και λήψης μηνυμάτων. Το πιο διαδεδομένο μέσο επικοινωνίας στο διαδίκτυο ήταν το ηλεκτρονικό ταχυδρομείο (E-mail), όπου απαιτούσε την εγκατάσταση ειδικών προγραμμάτων.

Επίσης, ο χρήστης έπρεπε να είναι συνδεδεμένος στον υπολογιστή του για να διαβάσει τα μηνύματα του, αν και υπήρχαν προγράμματα ανταλλαγής άμεσων μηνυμάτων (Instant messaging), η χρήση τους δεν ήταν τόσο διαδεδομένη και φιλική προς τον χρήστη, όπως σήμερα.

Στις μέρες μας, οι ταχύτητες του διαδικτύου είναι μεγαλύτερες ακόμα επιτυγχάνουν μεγάλες αποστάσεις και επιτρέπουν τη μεταφορά αρχείων και μηνυμάτων σε πραγματικό χρόνο το Messenger και το Telegram αποτελούν δημοφιλείς εφαρμογές ανταλλαγής μηνυμάτων, οι οποίες επιτρέπουν την ανταλλαγή φωτογραφιών ηχητικών ή γραπτών μηνυμάτων και βίντεο σε πραγματικό χρόνο, ανεξαρτήτως τοποθεσίας και συσκευής. Πρωτοπορούν, επίσης, τα μέσα κοινωνικής δικτύωσης ως προς την ανταλλαγή μηνυμάτων, δίνοντας στους χρήστες τους τη δυνατότητα να ανταλλάσσουν μηνύματα σε πραγματικό χρόνο, είτε σε ιδιωτικές είτε σε ομαδικές συνομιλίες, βοηθώντας ακόμα και στη διάδοση ειδήσεων.

Συνολικά, για ανταλλαγή μηνυμάτων στο διαδίκτυο έχει εξελιχθεί από ένα αργό, βασισμένο σε κείμενο είδος επικοινωνίας, σε μια ποικίλη αλληλεπίδραση σε πραγματικό χρόνο.

Κοινωνική Συνδεσιμότητα στην Ψηφιακή Εποχή

Με την εμφάνιση του διαδικτύου, οι τρόποι επικοινωνίας έχουν υποστεί μεγάλες αλλαγές, και οι αλληλεπίδραση μεταξύ των ανθρώπων ξεπερνά την επικοινωνία εντός συνόρων, και κάθε είδος μονότονης επικοινωνίας. Πλέον οι δυνατότητες επικοινωνίας προσφέρουν ποικίλες μεθόδους και είναι παγκόσμιας εμβέλειας. Οι εφαρμογές ανταλλαγής μηνυμάτων λειτουργούν ως κοινό σημείο συνάντησης, που μέσα από αυτές συνδέονται και δικτυώνονται άνθρωποι από όλο τον κόσμο, με διαφορετική καταγωγή, ιστορικό υπόβαθρο, ηλικίες, κι έτσι δίνουν τη δυνατότητα ο ένας στον άλλον να ανταλλάξουν τις προσωπικές τους ιστορίες, ακόμα και να δημιουργηθούν αληθινές σχέσεις. Με τη βοήθεια των εφαρμογών ανταλλαγής μηνυμάτων, οι χρήστες επιτυγχάνουν την διατήρηση των σχέσεων σε μεγάλες αποστάσεις, και καθιστούν πιο εύκολη την επανασύνδεση παλαιών φίλων και συγγενών.

Είναι γνωστό πως μειονότητες και περιθωριοποιημένες ομάδες χρησιμοποιούν τις εφαρμογές και γενικότερα το διαδίκτυο για τραβήξουν την προσοχή και να ακουστεί η φωνή τους.

Πολιτική συμμετοχή

Με την άνοδο του διαδικτύου οι πολίτες έχουν τη δυνατότητα να συμμετέχουν περισσότερο και με ενεργό τρόπο στην πολιτική διαδικασία και να ελέγχουν της κυβερνήσεις τους. Μια αξιοσημείωτη πτυχή αυτής της εξέλιξης είναι η εμφάνιση των πλατφορμών δικτύωσης, όπου οι πολίτες μπορούν να εκφράσουν ελεύθερα τις απόψεις τους, να διαφωνήσουν ή να συγκεντρώσουν υποστήριξη για διάφορους πολιτικούς λόγους. Έτσι, με την αξιοποίηση των διαδικτυακών εργαλείων επικοινωνίας, τα κινήματα βάσης οργανώνουν, με επιτυχία, διαδηλώσεις, συγκεντρώσεις και εκστρατείες υπεράσπισης, ενισχύοντας την επιρροή τους και φέρνοντας, τελικά, σημαντικούς και θετικούς μετασχηματισμούς στην κοινωνία. Ένας άλλος σημαντικός τομέας, όπου το διαδίκτυο έχει καταφέρει να διαδραματίσει κρίσιμο ρόλο, είναι η προώθηση της διαφάνειας και της λογοδοσίας στις κυβερνητικές υποθέσεις, με την έννοια ότι οι πολίτες έχουν πρόσβαση σε κυβερνητικά δεδομένα, μπορούν να παρακολουθούν δημόσια αξιωματικούς και να συμμετέχουν στις διαδικασίες λήψης αποφάσεων. Μέσω πρωτοβουλιών ανοιχτών δεδομένων και πλατφορμών τεχνολογίας πολιτών, μπορούν να συνεργαστούν με πιο αποτελεσματικό τρόπο, με κυβερνητικούς φορείς για την ανάπτυξη δημόσιων υπηρεσιών αλλά και την αντιμετώπιση των αναγκών των κοινοτήτων τους.

Διαμοιρασμός Γνώσης και Συνεργασία

Το διαδίκτυο έχει μεταμορφώσει την ανταλλαγή γνώσεων και τη συνεργασία με το να επιτρέπει τη μεγαλύτερη πρόσβαση σε πληροφορίες και την ευκαιρία για παγκόσμια συνεργασία. Δωρεάν προσφέρεται μια τεράστια γκάμα γνώσεων από διαδικτυακές πλατφόρμες, όπως η Wikipedia και τα περιοδικά ανοικτής πρόσβασης. Επιτρέπουν δε, να συμμετέχουν στη μάθηση, την έρευνα και τη δημιουργία, άτομα από διάφορα υπόβαθρα. Τα έγγραφα Google και το Slack είναι εργαλεία τα οποία διευκολύνουν τις ομάδες να συνεργάζονται σε πραγματικό χρόνο, οδηγώντας σε αυξημένη αποτελεσματικότητα, ανεξάρτητα από την τοποθεσία τους. Δημιουργείται ένα δυναμικό περιβάλλον για συνεχή προσωπική και επαγγελματική ανάπτυξη από τις διαδικτυακές κοινότητες μάθησης και τα φόρουμ, αφού προσφέρουν ένα χώρο για τα άτομα ώστε συμμετέχουν στη μάθηση ομότιμα και να ανταλλάσσουν γνώσεις.

Πολιτισμική Ποικιλομορφία και Έκφραση

Η διαδικτυακή επικοινωνία διευκολύνει την πολιτιστική πολυμορφία και την προώθηση της διαπολιτισμικής ανταλλαγής και εμπλουτίζει το παγκόσμιο πολιτιστικό τοπίο. Τα άτομα μπορούν να μοιραστούν τις παραδόσεις, τις γλώσσες και τις μορφές τέχνης τους με

κοινό σε όλο τον κόσμο, μέσα από τις πλατφόρμες μέσω κοινωνικής δικτύωσης. Αυτή η ανταλλαγή πολιτιστικών εκφράσεων, ενισχύει την αμοιβαία κατανόηση και εκτίμηση που με τη σειρά της οδηγεί σε γεφύρωση των πολιτισμικών διαφορών και στην προώθηση της αίσθησης της παγκόσμιας ιθαγένειας. Οι άνθρωποι έχουν την ευκαιρία να εμβαθύνουν σε διαφορετικούς πολιτισμούς και ιστορίες από την άνεση του σπιτιού τους, λόγω του ότι η έλευση του διαδικτύου έχει φέρει επανάσταση στον τρόπο που έχουμε πρόσβαση και αλληλεπίδραση με πολιτιστικά αντικείμενα και τοποθεσίες πολιτιστικής κληρονομιάς. Τα ψηφιακά αρχεία, τα εικονικά μουσεία και οι διαδικτυακές εκθέσεις συμβάλλουν στην διατήρηση των πολύτιμων πολιτιστικών θησαυρών. Επίσης, δίνουν τη δυνατότητα να είναι διαθέσιμοι αυτοί οι πολιτιστικοί θησαυροί στις μελλοντικές γενιές για να τους εξερευνήσουν. Τέλος, οι πλατφόρμες περιεχομένου που έχουν δημιουργηθεί από χρήστες δίνουν την ευκαιρία στα άτομα να εκφράσουν τη δημιουργικότητά τους, να μοιραστούν τις δικές τους πολιτιστικές εκφράσεις και να υπάρξει έτσι ένα αναπτυσσόμενο περιβάλλον δημιουργικότητας και καινοτομίας.

Αξιοποίηση της διαδικτυακής επικοινωνίας για επιχειρηματική συνεργασία

Απαραίτητη για την αποτελεσματική συνεργασία στις επιχειρήσεις είναι η χρήση της διαδικτυακής επικοινωνίας, ιδιαίτερα μέσω πλατφορμών συνομιλίας. Με τον τρόπο αυτό, οι επαγγελματίες μπορούν να αντιμετωπίζουν επείγοντα ζητήματα, να ανταλλάσσουν ιδέες και να λαμβάνουν αποφάσεις σε πραγματικό χρόνο. Με την ανταλλαγή μηνυμάτων προωθείται ο απρόσκοπτος συντονισμός μεταξύ των μελών της ομάδας, χωρίς να έχει σημασία η γεωγραφική τους θέση. Ενισχύει την ευελιξία και την ανταπόκριση σε περιβάλλοντα εργασίας με γρήγορους ρυθμούς, συμβάλλει στη γρήγορη επίλυση προβλημάτων και στη διαδικασία λήψης αποφάσεων.

Στα αρχεία καταγραφής συνομιλιών βρίσκεται η τεκμηρίωση των συζητήσεων και των αποφάσεων, η οποία διασφαλίζει τη συνέχεια και τη διαφάνεια ώστε οι ομάδες να παρακολουθούν την πρόοδο, να διατηρούν την ευθυγράμμιση με κοινούς

επιχειρηματικούς στόχους και να επιλύουν συγκρούσεις. Ενισχύεται η παραγωγικότητα και καλλιεργείται μια κουλτούρα συνεργασίας και καινοτομίας, εντός των οργανισμών, με την υιοθέτηση εργαλείων διαδικτυακής επικοινωνίας για επαγγελματικές συνομιλίες. Έτσι επιτυγχάνεται, τελικά, μια βιώσιμη ανάπτυξη και επιτυχία.

Παρά τα πολυάριθμα οφέλη που προσφέρει η διαδικτυακή επικοινωνία, υπάρχουν επίσης πολλά μειονεκτήματα που πρέπει να ληφθούν υπόψη.

Διαδικτυακός εκφοβισμός(Cyberbullying)

Ένας τεράστιος εφιάλτης στη διαδικτυακή επικοινωνία είναι σήμερα ο εκφοβισμός μέσω διαδικτύου. Αυτός περιλαμβάνει τη χρήση ηλεκτρονικών μέσων για την παρενόχληση ή τον εκφοβισμό άλλων ατόμων. Ο παραδοσιακός εκφοβισμός περιορίζεται σε φυσικούς χώρους σε αντίθεση με τον διαδικτυακό όποιος δεν γνωρίζει σύνορα και γι' αυτό η διαφυγή των θυμάτων είναι πολύ πιο δύσκολη και ορισμένες φορές αδύνατη. Διάφορα άτομα κρύβονται πίσω από κοινωνικά μέσα, διαδικτυακά φόρουμ και εφαρμογές ανταλλαγής μηνυμάτων, συνήθως με ψευδείς ταυτότητες. Όλα αυτά κάνουν πιο εύκολο και πιο επικίνδυνο το να εμπλακούν τα άτομα σε πράξεις εκφοβισμού από διάφορα κοινωνικά μέσα, καθώς επίσης και να επιδεινώσει την ψυχική κατάσταση των θυμάτων. Κατ' επέκταση, μπορεί ο εκφοβισμός αυτός να οδηγήσει σε μακροχρόνια ψυχική τραυματική κατάσταση και ακόμα χειρότερα σε τραγικά αποτελέσματα, όπως αυτοκτονία.

Παραπληροφόρηση

Η διάδοση πληροφοριών στο διαδίκτυο δεν έχει πάντα θετικό πρόσημο. Αυτό σημαίνει ότι ενώ είναι μια τεράστια πηγή πληροφοριών και γνώσεων, παράλληλα επιτρέπει τη διάδοση ψευδών και ανυπόστατων ειδήσεων και πληροφοριών. Είναι πολύ εύκολο όλες ψευδείς και ανυπόστατες αυτές πληροφορίες να εξαπλωθούν μέσω των μέσων κοινωνικής δικτύωσης και πλατφορμών, προκαλώντας ποικίλα προβλήματα, όπως σύγχυση και έλλειψη εμπιστοσύνης. Υπάρχουν περιπτώσεις που η δημιουργία ψεύτικων ιστοριών γίνεται σκόπιμα για την εξαπάτηση άλλων ανθρώπων. Πολλές φορές όμως, η διάδοση αυτών γίνεται εν αγνοία των ατόμων που τις διαδίδουν. Ο κατακλυσμός παραπλανητικών πληροφοριών περιπλέκει την ικανότητα των ανθρώπων να διακρίνουν μεταξύ γεγονότος και φαντασίας.

Πολλές διαδικτυακές πλατφόρμες έχουν σχεδιαστεί ειδικά για να διαδίδουν ψεύτικες ειδήσεις, ενώ οι αλγόριθμοι των μέσων κοινωνικής δικτύωσης έχουν την τάση να μας παρουσιάζουν περιεχόμενο που ευθυγραμμίζεται με τις προϋπάρχουσες απόψεις μας, οδηγώντας στο σχηματισμό «echo chambers». Αυτοί οι «echo chambers» καθιστούν όλο και πιο δύσκολη τη διάκριση μεταξύ πραγματικότητας και ψεύδους. Λόγω των συνεπειών που προαναφέρθηκαν, τα άτομα μπορούν εύκολα να χειραγωγηθούν και αυτό μπορεί να οδηγήσει σε διαιρέσεις μεταξύ διαφορετικών ομάδων ανθρώπων. Κατά συνέπεια, η αντίληψή μας για

τον κόσμο γίνεται απλοϊκή, περιορίζεται σε μια διχοτομική άποψη, εμποδίζοντας έτσι την ικανότητά μας να κατανοούμε και να συμπάσχουμε με εναλλακτικές απόψεις. Αυτή η ατμόσφαιρα αβεβαιότητας και δυσπιστίας που επικρατεί παρεμποδίζει σημαντικά τη δυνατότητα συμμετοχής σε ουσιαστικές και παραγωγικές συζητήσεις εντός του ψηφιακού χώρου.

Εθισμός

Μία από τις πιο ανησυχητικές αρνητικές πτυχές της διαδικτυακής συνομιλίας είναι η ικανότητά της να καλλιεργεί τάσεις εθισμού. Η άμεση ικανοποίηση που παρέχει η διαδικτυακή επικοινωνία, μαζί με τη διαθεσιμότητα των πλατφορμών συνομιλίας 24 ώρες το 24ωρο, μπορεί να έχει ως αποτέλεσμα τα άτομα να ξοδεύουν υπερβολικό χρόνο συνομιλώντας αντί να συμμετέχουν σε αλληλεπιδράσεις πρόσωπο με πρόσωπο και να εκπληρώνουν τις υποχρεώσεις τους. Αυτός ο εθισμός στη διαδικτυακή συνομιλία, που χαρακτηρίζεται από ανεξέλεγκτη και καταναγκαστική χρήση, μπορεί να επηρεάσει αρνητικά την ψυχική υγεία και την παραγωγικότητα. Τέλος, η αύξηση της ντοπαμίνης, που οδηγεί σε αίσθημα ευχαρίστησης, που προκαλείται από τη λήψη ειδοποιήσεων και μηνυμάτων μπορεί να ενισχύσει περαιτέρω τις εθιστικές συμπεριφορές, παγιδεύοντας τα άτομα σε έναν επιβλαβή κύκλο εξάρτησης από εικονικές κοινωνικές αλληλεπιδράσεις.

2.2 Εφαρμογές ανταλλαγής μηνυμάτων στη σύγχρονη κοινωνία

Στον 21ο αιώνα, οι εφαρμογές ανταλλαγής μηνυμάτων, έχουν υποστεί ραγδαίες αλλαγές και εξελίξεις με συνέπεια να αλλάζει και ο τρόπος με τον οποίο οι χρησιμοποιούνται από τον άνθρωπο. Με μια αναδρομή στο παρελθόν, διαπιστώνουμε ότι εφαρμογές όπως, το Messenger, WhatsApp, και άλλες παρόμοιες, έχουν επαναστατήσει στον χώρο της επικοινωνίας μεταξύ των ανθρώπων. Σε σύγκριση με την υπηρεσία σύντομου μηνύματος (short message service, SMS), ηλεκτρονικό ταχυδρομείο (Email) και άλλες υπηρεσίες ανταλλαγής μηνυμάτων από το παρελθόν, παρατηρούμε αρκετές αλλαγές στις σύγχρονες εφαρμογές επικοινωνίας.

Η επικοινωνία σε πραγματικό χρόνο είναι μια σύγχρονη λειτουργία που εμφανίζεται στις εφαρμογές, σε αντίθεση με το ηλεκτρονικό ταχυδρομείο ή την υπηρεσία σύντομων μηνυμάτων που

μπορούν να φθάνουν στον παραλήπτη με μικρή χρονοκαθυστέρηση. Οι σύγχρονες εφαρμογές επιτρέπουν στον χρήστη ακόμα και την πρόσβαση στην επικαιρότητα. Επιπλέον, οι εφαρμογές στον 21ο αιώνα επιτρέπουν όχι μόνο την ανταλλαγή μηνυμάτων, αλλά και ανταλλαγή εικόνων, ηχητικών μηνυμάτων, βίντεο και αρχείων. Αυτό δίνει τη δυνατότητα για μια πιο ζωντανή και πληθώρα επικοινωνία. Σε αντίθεση με τον παλαιότερο τύπο ηλεκτρονικής και διαδικτυακής επικοινωνίας, οι σύγχρονες εφαρμογές προσφέρουν τη δυνατότητα ομαδικής επικοινωνίας και επιπλέον αποστολή τοποθεσίας, ώστε να είναι κατανοητή η τοποθεσία του χρήστη. Όντας φιλικές προς τον χρήστη, είναι δυνατή η χρήση των εφαρμογών όχι μόνο από υπολογιστή, αλλά και με κινητό τηλέφωνο ή τάμπλετ.

Οι εφαρμογές ανταλλαγής μηνυμάτων είναι ιδιαίτερα σημαντικές στην σύγχρονη κοινωνία γι' αυτό και υπάρχουν παντού αφού καλύπτουν ποικίλες ανάγκες. Ακολουθούν αρκετά αξιοσημείωτα παραδείγματα τέτοιων εφαρμογών:

- **Facebook Messenger:** Το Messenger, όπου ανήκει στο Facebook, είναι μία εφαρμογή ανταλλαγής μηνυμάτων που δίνει στους ανθρώπους τη δυνατότητα να επικοινωνούν με άλλα άτομα ανά τον κόσμο. Πιο συγκεκριμένα, η εν λόγω εφαρμογή υποστηρίζει διάφορες λειτουργίες όπως για παράδειγμα, μηνύματα κειμένου, φωνητικές κλήσεις, βιντεοκλήσεις καθώς και πολλές δυνατότητες κοινής χρήσης πολυμέσων. Με αυτό τον τρόπο το Messenger ανταποκρίνεται πλήρως σε όλες τις ανάγκες των χρηστών του και λειτουργεί ως ένα ισχυρό εργαλείο για την διευκόλυνση της επικοινωνίας.
- **Instagram:** Μια εξίσου ευρέως διαδεδομένη εφαρμογή είναι το Instagram ,όπου διαθέτει το Instagram Direct. Η εφαρμογή αυτή επιτρέπει στα άτομα που την χρησιμοποιούν να στέλνουν εύκολα μηνύματα κειμένου ,βίντεο, φωτογραφίες ,να κάνουν βιντεοκλήσεις με τους ακόλουθους τους και όχι μόνο. Τέλος, είναι προσβάσιμη και μέσω υπολογιστή κάνοντας την ακόμα πιο ελκυστική κυρίως για άτομα που δεν θέλουν ή δεν γνωρίζουν να χρησιμοποιήσουν το κινητό τους τηλέφωνο.

- WhatsApp & WhatsApp Web:** Το WhatsApp, όπου είναι κυρίως γνωστό στο εξωτερικό, είναι μία δωρεάν εφαρμογή η οποία προσφέρει στους χρήστες της την δυνατότητα να επικοινωνούν στέλνοντας μηνύματα κειμένου και μηνύματα πολυμέσων, φωτογραφίες, βίντεο και ήχους. Η εφαρμογή αυτή εξαγοράστηκε από το Facebook το 2014, το οποίο διαθέτει δισεκατομμύρια ενεργούς χρήστες παγκοσμίως. Η κρυπτογράφηση από άκρο σε άκρο της εφαρμογής (WhatsApp's end-to-end encryption) ,η οποία εγγυάται την ασφάλεια και το απόρρητο των χρηστών της, η ευκολία χρήσης και η δυνατότητα του να λειτουργεί ανεξαρτήτου συσκευής την καθιστά μία από τις πιο δημοφιλείς εφαρμογές για άτομα που επιθυμούν είτε προσωπικές είτε και επαγγελματικές συνομιλίες.
- LinkedIn Messaging:** Το LinkedIn είναι μία ιστοσελίδα/εφαρμογή η οποία δίνει την δυνατότητα , τόσο στις εταιρείες όσο και στους στα άτομα που αναζητούν εργασία, να έρχονται σε επαφή. Πιο συγκεκριμένα, το LinkedIn Messaging που παρέχεται από το LinkedIn, επιτρέπει στους χρήστες να επικοινωνούν μεταξύ τους και να δικτυώνονται στην συγκεκριμένη πλατφόρμα. Οι ενδιαφερόμενοι μπορούν να στέλνουν και να λαμβάνουν μηνύματα, ώστε να συνδέονται με συναδέλφους τους και υποψήφιους εργοδότες καθώς και να μοιράζονται άρθρα και ενημερώσεις. Το LinkedIn ουσιαστικά είναι, αν όχι καλύτερη, από τις καλύτερες εφαρμογές διαφήμισης του επαγγελματικού έργου ενός ανθρώπου.
- Ανώνυμες εφαρμογές ανταλλαγής μηνυμάτων:

Οι ανώνυμες εφαρμογές μηνυμάτων δίνουν τη δυνατότητα στους χρήστες να έρθουν σε επικοινωνία με άλλους χρήστες, χωρίς να είναι εμφανείς οι προσωπικές τους πληροφορίες. Οι εφαρμογές αυτές παρέχουν διάφορες λειτουργίες για να εξασφαλίσουν την ανωνυμία και να προστατεύσουν την ιδιωτικότητα των χρηστών ενώ πολλές φορές δεν απαιτούν τη δημιουργία λογαριασμού. Η επιλογή του συνομιλητή γίνεται τυχαία και το σύστημα φέρνει αυτόματα δύο χρήστες σε επικοινωνία. Συχνά τα μηνύματα των χρηστών διαγράφονται αυτόματα ώστε να εξασφαλίζεται ότι τα μηνύματα δεν αποθηκεύονται μόνιμα.

Οι παραπάνω εφαρμογές συνομιλίας αντιπροσωπεύουν μόνο ένα κλάσμα του διαφορετικού τοπίου των πλατφορμών ανταλλαγής μηνυμάτων που διατίθενται σήμερα. Με μια σειρά επιλογών που καλύπτουν διάφορες ανάγκες και σκοπούς, αυτές οι πλατφόρμες έχουν φέρει επανάσταση στον τρόπο επικοινωνίας ανθρώπων και οργανισμών στην ψηφιακή εποχή. Έχουν γίνει ένα ουσιαστικό συστατικό της σύγχρονης κοινωνίας, είτε πρόκειται για προσωπικές συνομιλίες είτε για επαγγελματικές συνεργασίες.

3 Κατασκευή Εφαρμογής

Το κεφάλαιο αυτό έχει ως σκοπό να παρουσιάσει την φιλοσοφία και την τεχνική διαδικασία δημιουργίας, της εφαρμογής ανταλλαγής μηνυμάτων πραγματικού χρόνου, που θα αναπτυχθεί στα πλαίσια της παρούσας πτυχιακή εργασίας, αλλά και γενικότερα να περιγράψει τον σύγχρονο τρόπο ανάπτυξης εφαρμογών διαδικτύου.

3.1 Περιγραφή Αντικειμένου

Η βασική φιλοσοφία και ο στόχος της εφαρμογής ανταλλαγής μηνυμάτων είναι η επικοινωνία και πως αυτή μπορεί να πραγματοποιηθεί με την δημιουργία ενός χώρου όπου οι χρήστες μπορούν να συνδεθούν, και να επικοινωνήσουν με άλλους χρήστες σε πραγματικό χρόνο χρησιμοποιώντας σύγχρονες τεχνολογίες καθολικά προσβάσιμες.

Η εφαρμογή εστιάζει στην απλότητα χρήσης και την ταχύτητα. Έχει σχεδιαστεί ώστε να είναι απλή και εύχρηστη, απευθύνεται σε απλούς και όχι τεχνικούς χρήστες, επιτρέποντάς τους να ανταλλάσσουν μηνύματα με ευκολία και άνεση.

Κύριο βασικό στοιχείο της εφαρμογής ανταλλαγής μηνυμάτων είναι η ασφάλεια και ιδιωτικότητα. Κάθε εν δυνάμει χρήστης της εφαρμογής θα πρέπει πρώτα να δημιουργήσει έναν προσωπικό λογαριασμό, με μοναδικό προσωπικό και κρυπτογραφημένο κωδικό.

Μπορεί να επικοινωνήσει μόνο με συνδεδεμένους χρήστες, επιλέγοντας τον χρήστη που θέλει να συνομιλήσει και στην συνέχεια δημιουργώντας συνομιλίες δύο ατόμων.

Τα μηνύματα είναι κρυπτογραφημένα και η συνομιλία καθώς και το περιεχόμενό της είναι προσβάσιμα μόνο στους δύο χρήστες που την αποτελούν.

Ο σχεδιασμός και η υλοποίηση ακολουθούν όλα τα σύγχρονα πρότυπα τέτοιου είδους εφαρμογών όσον αφορά την δημιουργία λογαριασμού, την αυθεντικοποίηση και την σύνδεση του χρήστη, αλλά και τον τρόπο ανταλλαγής μηνυμάτων τόσο για την ευχρηστία όσο και για την ασφάλεια.

Γενικότερα, οι εφαρμογές ανταλλαγής μηνυμάτων χαρακτηρίζονται ως καταναλωτικές εφαρμογές και απευθύνονται σε όλους τους χρήστες. Αυτό συμβαίνει καθώς τέτοιου τύπου εφαρμογές εξυπηρετούν την ανάγκη της επικοινωνίας, και ως εκ τούτου όλοι οι χρήστες μπορούν να επωφεληθούν με την χρήση τους.

Η παρούσα εφαρμογή έχει την δυνατότητα να αποτελέσει την βάση για την ανάπτυξη ενός προϊόντος που μπορεί να χρησιμοποιηθεί από πραγματικούς χρήστες στον αληθινό κόσμο. Με τις κατάλληλες παραμετροποιήσεις, διαμορφώσεις και ελέγχους μπορεί να επεκταθεί σε πόρους και λειτουργίες και να εξυπηρετεί καθημερινά πληθώρα χρηστών.

Στην συνέχεια του κεφαλαίου, θα αναλυθούν τα βήματα και οι διαδικασίες που ακολουθήθηκαν ώστε να αναπτυχθεί η εφαρμογή. Τέλος, θα αναφερθούν τα εργαλεία και οι τεχνολογίες που χρησιμοποιήθηκαν, ο λόγος που προτιμήθηκαν έναντι των υπολοίπων διαθέσιμων, ποια προβλήματα υπερνικούν και πόσο συμβάλλουν στην ταχύτητα, στην αμεσότητα και στις συνολικές επιδόσεις.

3.2 Τεχνολογίες & Εργαλεία Ανάπτυξης

Η αρχιτεκτονική των εφαρμογών διαδικτύου υπαγορεύει κάποιες σχεδιαστικές αρχές και πρακτικές οι οποίες κρίνεται απαραίτητο να ακολουθηθούν για την βέλτιστη, γρήγορη και λιγότερο επιρρεπής σε σφάλματα, διαδικασία ανάπτυξης. Περισσότερα για την αρχιτεκτονική της εφαρμογής θα αναλυθούν σε επόμενο κεφάλαιο, ωστόσο κρίνεται σκόπιμο για την κατανόηση του κεφαλαίου να αναφέρουμε κάποιες γενικές αλλά σημαντικές έννοιες.

Κάθε εφαρμογή διαδικτύου χωρίζεται σε τρία βασικά επίπεδα. Αρχικά, είναι το επίπεδο της Παρουσίασης Presentation Layer, που αφορά το τμήμα λογισμικού που είναι υπεύθυνο για την αλληλεπίδραση με τον χρήστη και το οποίο χρησιμοποιεί ώστε να αλληλεπιδράσει με το επίπεδο της Λογικής Επιχειρήσεων.

Το επίπεδο της Λογικής επιχειρήσεων Business Logic Layer το οποίο είναι υπεύθυνο για την επεξεργασία των δεδομένων και την υλοποίηση της επιχειρηματικής λογικής της εφαρμογής.

Το επίπεδο των Δεδομένων (Data Layer) το οποίο ασχολείται με τη διαχείριση των δεδομένων της εφαρμογής.

Ο διαχωρισμός αυτών των επιπέδων είναι μέγιστης σημασίας καθώς έτσι επιτυγχάνεται η βέλτιστη συντήρηση, επέκταση και αλληλεπίδραση των μερών της εφαρμογής. Οι αλλαγές σε ένα

επίπεδο δεν θα πρέπει να επηρεάζουν τα υπόλοιπα, επιτρέποντας έτσι μεγαλύτερη ευελιξία και αποτελεσματικότητα στην ανάπτυξη, συντήρηση και λειτουργία της εφαρμογής.

Στις επόμενες παραγράφους θα παρουσιάσουμε τα εργαλεία που χρησιμοποιήθηκαν για να αναπτυχθούν τα παραπάνω επίπεδα.

3.2.1 Angular Framework - Εγκατάσταση Angular σε τοπικό υπολογιστή

Για την ανάπτυξη του επιπέδου Παρουσίασης – Διεπαφής Χρήστη, θα χρησιμοποιηθεί η Angular.

Η Angular είναι ένα framework (βλ. παράρτημα), σχεδιασμού και ανάπτυξης εφαρμογών, ανοικτού κώδικα (Open Source), βασισμένο στην γλώσσα προγραμματισμού TypeScript (βλ. παράρτημα) και υποστηρίζεται από την Google. Χρησιμοποιείται για την δημιουργία εξελιγμένων και αποτελεσματικών single-page εφαρμογών.

Ας δούμε τώρα τα βήματα για το πως μπορούμε να εγκαταστήσουμε την Angular σε τοπικό υπολογιστή. Αρχικά, θα χρειαστεί να έχουμε εγκαταστήσει το Node.JS τοπικά ώστε να μπορέσουμε να κατεβάσουμε τα απαραίτητα πακέτα. Μπορούμε να εγκαταστήσουμε το Node μεταβαίνοντας στην σελίδα του node (βλ. παράρτημα) και επιλέγοντας την τελευταία σταθερή έκδοση (latest stable version) για το λειτουργικό μας σύστημα. Στην συνέχεια θα πρέπει να εκτελέσουμε τον οδηγό εγκατάστασης και να ακολουθήσουμε τις οδηγίες. Για να ελέγξουμε εάν η εγκατάσταση έχει επιτευχθεί, μπορούμε να ανοίξουμε ένα τερματικό και να εκτελέσουμε τις παρακάτω εντολές, οι οποίες θα εμφανίσουν τις εγκαταστημένες εκδόσεις:

```
C:\>node -v
v18.18.2

C:\>npm -v
9.8.1
```

Αφού έχουμε εγκαταστήσει τον Node, θα χρειαστεί να δημιουργήσουμε ένα νέο Angular Project, χρησιμοποιώντας το πακέτο της διεπαφής εκτέλεσης εντολών (C.L.I. – βλ. παράρτημα) που μας παρέχει η Angular. Για να το χρησιμοποιήσουμε θα πρέπει πρώτα να το κατεβάσουμε και να το εγκαταστήσουμε καθολικά (globally) στον τοπικό μας υπολογιστή χρησιμοποιώντας τον Node μέσα από το τερματικό μας και εκτελώντας την παρακάτω εντολή:

```
npm install -g @angular/cli
```

Για να ελέγξουμε εάν η εγκατάσταση έχει επιτευχθεί μπορούμε να εκτελέσουμε την παρακάτω εντολή για να ελέγξουμε την έκδοση της Angular, όπως φαίνεται στην εικόνα 3.2.1.

```
C:\>ng version

Angular CLI
Angular CLI: 16.2.11
Node: 18.18.2
Package Manager: npm 9.8.1
OS: win32 x64

Angular:
...

Package                                Version
-----
@angular-devkit/architect              0.1602.11 (cli-only)
@angular-devkit/core                   16.2.11 (cli-only)
@angular-devkit/schematics             16.2.11 (cli-only)
@schematics/angular                   16.2.11 (cli-only)
```

Εικόνα 3.2.1 Έκδοση Angular

Σαν τελευταίο βήμα, θα δημιουργηθεί ένα νέο Angular έργο (project) τρέχοντας την εντολή με το όνομα της επιλογής μας:

```
C:\>ng new my-project-name
```

Ακολουθούμε τις οδηγίες και όταν η διαδικασία και η εγκατάσταση των πακέτων ολοκληρωθεί θα έχουμε δημιουργήσει έναν φάκελο με το όνομα του έργου μας και όλα τα απαραίτητα αρχεία και πακέτα που αποτελούν ένα αρχικό Angular έργο. Στην συνέχεια μπορούμε να ανοίξουμε το έργο στον code editor της επιλογής μας και να ξεκινήσουμε την συγγραφή κώδικα.

Η παρούσα εφαρμογή δημιουργήθηκε με την παραπάνω διαδικασία και στην συνέχεια αναπτύχθηκαν και προστέθηκαν τα δομικά στοιχεία που την αποτελούν όπως components, modules και services για τα οποία θα γίνει εκτενέστερη αναφορά σε επόμενο κεφάλαιο.

3.2.2 Firebase BaaS Εγκατάσταση Εργαλείων του Firebase σε τοπικό υπολογιστή / console

Για την ανάπτυξη του επιπέδου της λογικής επιχειρήσεων (Business Logic) αλλά και του επιπέδου των δεδομένων χρησιμοποιήθηκε η Firebase.

Η Firebase είναι μια πλατφόρμα ανάπτυξης εφαρμογών της Google που παρέχει έναν ολοκληρωμένο σκετο υπηρεσιών για τη δημιουργία και τη λειτουργία εφαρμογών. Αυτές οι υπηρεσίες περιλαμβάνουν μια πραγματικού χρόνου βάση δεδομένων (Realtime Database), αυθεντικοποίηση χρηστών, αποθήκευση αρχείων, φιλοξενία εφαρμογών στον ιστό (web hosting) και πολλά άλλα.

Διαθέτει, επίσης, ένα εύρος εργαλείων ανάπτυξης λογισμικού SDK, τα οποία παρέχουν ένα πιο ευέλικτο και εύχρηστο περιβάλλον ανάπτυξης, καθώς προσφέρουν έτοιμες λύσεις και εργαλεία για την αλληλεπίδραση με διάφορες υπηρεσίες και πόρους. Εκτός των βασικών υπηρεσιών του παρέχονται και άλλες δυνατότητες, όπως η διαχείριση του αναπτυξιακού περιβάλλοντος (CLI), εργαλεία αναλυτικών δεδομένων και περιβάλλοντα ελέγχου ανάπτυξης.

Το Firebase είναι δημοφιλές για την ευκολία χρήσης, τη σταθερότητά του και την ικανότητά του να ενσωματωθεί εύκολα σε διάφορες πλατφόρμες και γλώσσες προγραμματισμού, παρέχοντας ένα πιο απλοποιημένο περιβάλλον ανάπτυξης σε σχέση με την HTTP API αρχιτεκτονική. Εκτενέστερη αναφορά για την αρχιτεκτονική θα γίνει στο σχετικό κεφάλαιο.

Εγκατάσταση Εργαλείων

Στην συνέχεια θα περιγράψουμε τα βήματα που πρέπει να ακολουθηθούν ώστε να δημιουργήσουμε το τοπικό περιβάλλον ανάπτυξης με την χρήση των προϊόντων του Firebase.

Αρχικά θα πρέπει να κάνουμε έναν δωρεάν λογαριασμό στην διαδικτυακή εφαρμογή του Firebase και να επιλέξουμε τα προϊόντα που επιθυμούμε.

Στην παρούσα πτυχιακή εργασία χρησιμοποιήθηκε το authentication για την δημιουργία λογαριασμού χρήστη και για την αυθεντικοποίηση, το real-time database για την εγγραφή των συνομιλιών και των μηνυμάτων και τέλος το hosting για την φιλοξενία της εφαρμογής στον ιστό.

Για να εγκαταστήσουμε το Firebase στην Angular εφαρμογή μας θα χρησιμοποιήσουμε μια βιβλιοθήκη τρίτων (third-party library) το @angular/fire (βλ. παράρτημα).

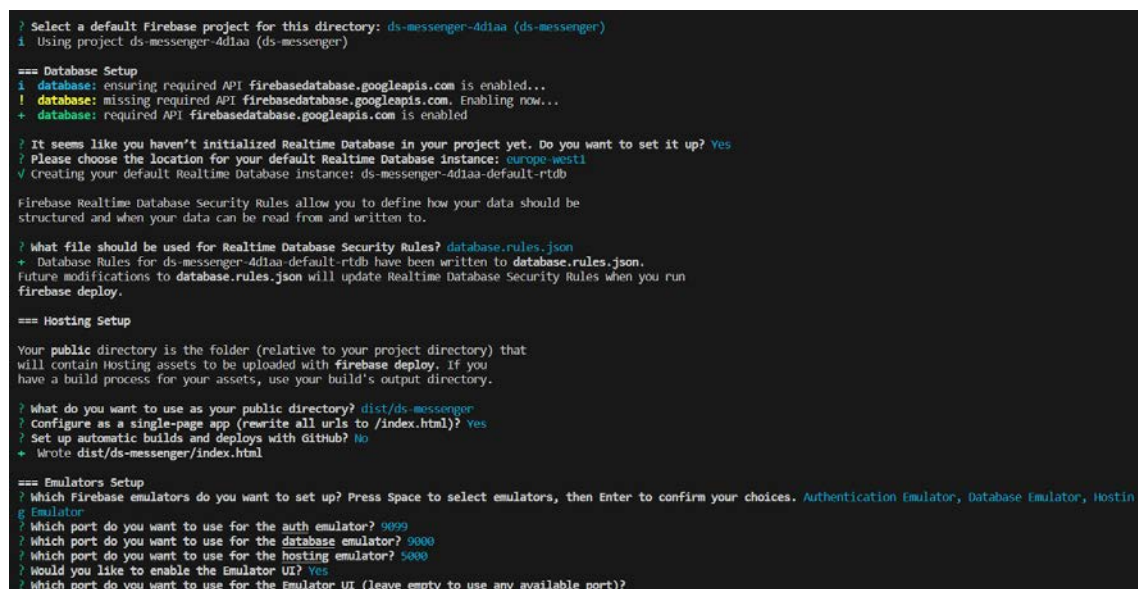
Θα απλουστεύσει την διαδικασία δημιουργίας ενός Firebase Project και θα κάνει τις απαραίτητες παραμετροποιήσεις ώστε να το συνδέσει με το Angular application μας.

Για την ολοκλήρωση της εγκατάστασης θα αναφέρουμε συνοπτικά τα απαραίτητα βήματα χωρίς να εμβαθύνουμε, καθώς είναι εκτός του θέματος της παρούσας πτυχιακής, υπάρχουν αρκετά λήμματα στο διαδίκτυο και επίσημα εγχειρίδια που ο αναγνώστης μπορεί να ανατρέξει για περισσότερες πληροφορίες, τα οποία θα αναφερθούν στο σχετικό κεφάλαιο.

Αρχικά, θα κάνουμε τις απαραίτητες εγκαταστάσεις πακέτων στο Angular application μας, εκτελώντας τις παρακάτω εντολές για συνδεθούμε με το Firebase λογαριασμό μας, για να εκκινήσουμε το Firebase και να το συνδέσουμε με το ήδη υπάρχον project που δημιουργήσαμε στην πλατφόρμα.

```
firebase login:ci  
  
firebase init
```

Στην συνέχεια, αφού ακολουθήσουμε τις οδηγίες και τα βήματα του οδηγού εγκατάστασης, θα πρέπει να επιλέξουμε τα προϊόντα που θα χρειαστούμε αλλά και τους Firebase Emulators. Το αποτέλεσμα θα είναι όμοιο με αυτό της εικόνας 3.2.2.



```
? Select a default Firebase project for this directory: ds-messenger-4d1aa (ds-messenger)
1 Using project ds-messenger-4d1aa (ds-messenger)

=== Database Setup
1 database: ensuring required API firebase.database.googleapis.com is enabled...
! database: missing required API firebase.database.googleapis.com. Enabling now...
+ database: required API firebase.database.googleapis.com is enabled

? It seems like you haven't initialized Realtime Database in your project yet. Do you want to set it up? Yes
? Please choose the location for your default Realtime Database instance: europe-west1
✓ creating your default Realtime Database instance: ds-messenger-4d1aa-default-rtdb

Firebase Realtime Database Security Rules allow you to define how your data should be
structured and when your data can be read from and written to.

? What file should be used for Realtime Database Security Rules? database.rules.json
+ Database Rules for ds-messenger-4d1aa-default-rtdb have been written to database.rules.json.
Future modifications to database.rules.json will update Realtime Database Security Rules when you run
firebase deploy.

=== Hosting Setup

Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? dist/ds-messenger
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
? Set up automatic builds and deploys with GitHub? No
+ Wrote dist/ds-messenger/index.html

=== Emulators Setup
? Which Firebase emulators do you want to set up? Press Space to select emulators, then Enter to confirm your choices. Authentication Emulator, Database Emulator, Hostin
g Emulator
? Which port do you want to use for the auth emulator? 9099
? Which port do you want to use for the database emulator? 9000
? Which port do you want to use for the hosting emulator? 5000
? Would you like to enable the Emulator UI? Yes
? Which port do you want to use for the Emulator UI (leave empty to use any available port)?
```

Εικόνα 3.2.2 Αποτέλεσμα Εγκατάστασης

Οι Firebase Emulators είναι μια σουίτα εργαλείων που παρέχονται από την Google Firebase για την ανάπτυξη και δοκιμή εφαρμογών που χρησιμοποιούν τις υπηρεσίες του Firebase. Αυτά

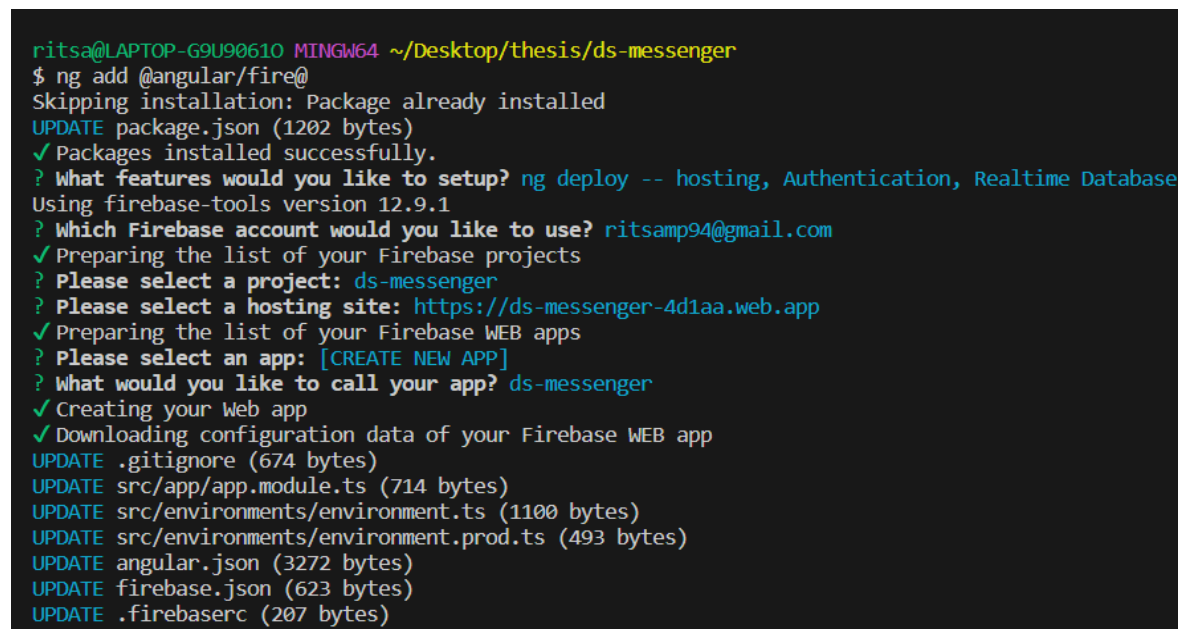
τα εργαλεία μας επιτρέπουν να τρέξουμε τοπικά τις υπηρεσίες του Firebase στον τοπικό υπολογιστή, αντί να χρησιμοποιήσουμε τις πραγματικές υπηρεσίες στον κεντρικό διακομιστή της Firebase.

Στην συνέχεια θα εγκαταστήσουμε την βιβλιοθήκη **@angular/fire**, με την οποία θα κάνουμε την σύνδεση του Angular project με τα προϊόντα της Firebase που εγκαταστήσαμε στο προηγούμενο βήμα.

Η εγκατάσταση αυτή θα δημιουργήσει αυτόματα τις βασικές παραμετροποιήσεις στα σχετικά αρχεία του project μας.

Στην συνέχεια θα κάνουμε τις περαιτέρω παραμετροποιήσεις για τις ανάγκες του project.

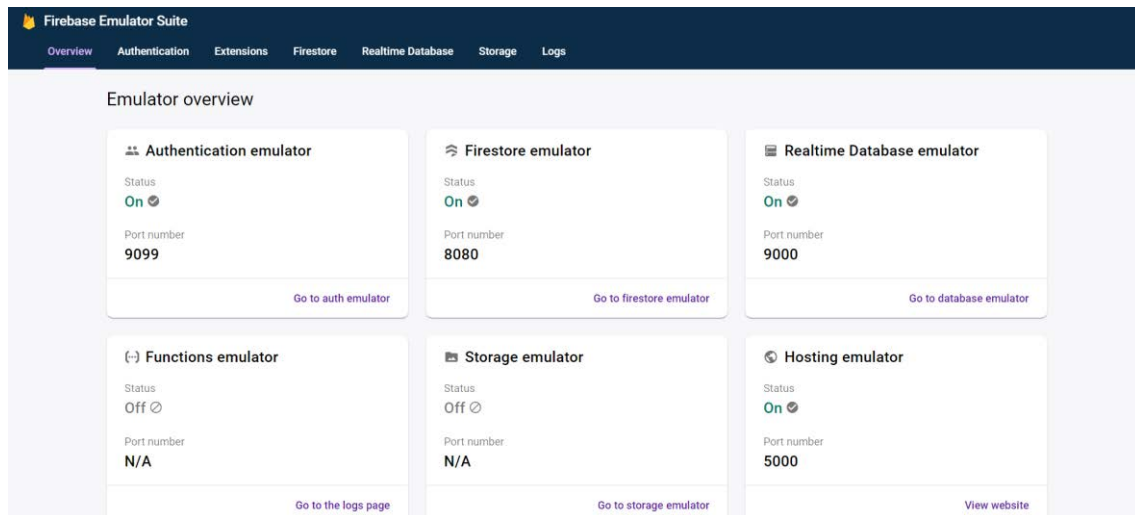
Το αποτέλεσμα της εγκατάστασης και αρχικοποίησης του Firebase θα είναι όμοιο με αυτό της εικόνας 3.2.3.



```
ritsa@LAPTOP-G9U90610 MINGW64 ~/Desktop/thesis/ds-messenger
$ ng add @angular/fire@
Skipping installation: Package already installed
UPDATE package.json (1202 bytes)
✓ Packages installed successfully.
? What features would you like to setup? ng deploy -- hosting, Authentication, Realtime Database
Using firebase-tools version 12.9.1
? Which Firebase account would you like to use? ritsamp94@gmail.com
✓ Preparing the list of your Firebase projects
? Please select a project: ds-messenger
? Please select a hosting site: https://ds-messenger-4d1aa.web.app
✓ Preparing the list of your Firebase WEB apps
? Please select an app: [CREATE NEW APP]
? What would you like to call your app? ds-messenger
✓ Creating your Web app
✓ Downloading configuration data of your Firebase WEB app
UPDATE .gitignore (674 bytes)
UPDATE src/app/app.module.ts (714 bytes)
UPDATE src/environments/environment.ts (1100 bytes)
UPDATE src/environments/environment.prod.ts (493 bytes)
UPDATE angular.json (3272 bytes)
UPDATE firebase.json (623 bytes)
UPDATE .firebaserc (207 bytes)
```

Εικόνα 3.2.3 Αποτέλεσμα Εγκατάστασης

Στην συνέχεια εκτελώντας αυτή την εντολή μπορούμε να τρέξουμε τοπικά τους emulators. Το αποτέλεσμα θα είναι όμοιο με αυτό της εικόνας.



Εικόνα 3.2.4 Emulators Suite

Σε αυτό το σημείο έχουν ολοκληρωθεί όλες οι απαραίτητες ενέργειες σεταρίσματος και αρχικοποίησης της εφαρμογής μας, οπότε και μπορούμε να αρχίσουμε την συγγραφή απαραίτητου κώδικα για την ανάπτυξη των λειτουργιών της εφαρμογής.

Για την πληρότητα της αναφοράς ας δούμε πως μπορεί να εκτελεστεί η εφαρμογή μας σε τοπικό υπολογιστή.

Αρχικά, θα πρέπει να έχουμε έναν code editor της επιλογής μας. Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκε το Visual Studio Code της Microsoft. Έτσι μπορούμε να ανοίξουμε τον φάκελο της εφαρμογής στον editor και σε ένα τερματικό, αφού μεταβούμε μέσα στον φάκελο να εκτελέσουμε την παρακάτω εντολή, για να εγκαταστήσουμε τα απαραίτητα πακέτα :

```
npm install
```

Στην συνέχεια, μπορούμε να εκτελέσουμε τα παρακάτω scripts για να «σερβίρουμε» το Angular έργο αλλά και να τρέξουμε τους emulators:

```
~/Desktop/ds-messenger-app> npm run start & npm run start-emulators
```

και η εφαρμογή θα ανοίξει αυτόματα, στον φυλλομετρητή που έχουμε εγκατεστημένο στον τοπικό υπολογιστή μας. Το αποτέλεσμα θα είναι όμοιο με αυτό της παρακάτω εικόνας 3.2.5.



⚠️ This application is developed for academic purposes! It's not recommended to enter sensitive data!!! ⚠️

Communicate with Your Friends in Real Time.

Create your account today for free and join the greatest chat community.

Login

Email *

Password *

Sign In

Don't have an account?
[Create one for free!](#)

Εικόνα 3.2.5 Landing Page

4 Περιγραφή εφαρμογής

Στο κεφάλαιο αυτό θα αναφέρουμε τις αποφάσεις που πήραμε για τις αρχές σχεδίασης και ποια πρότυπα θα ακολουθήσαμε, πως χρησιμοποιήσαμε τις διαθέσιμες τεχνολογίες στο έπακρο και πως θα επωφεληθούμε μέγιστα από αυτές καθώς και όλα εκείνα τα δομικά στοιχεία που απαρτίζουν την εφαρμογής μας.

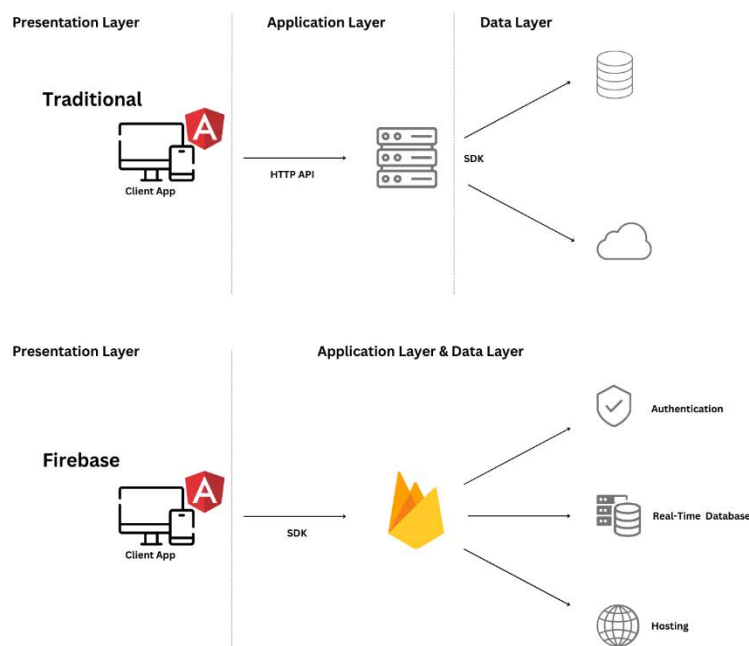
4.1 Αρχιτεκτονική Εφαρμογής

Η αρχιτεκτονική μιας εφαρμογής είναι το πρώτο και σημαντικότερο βήμα για την επιτυχημένη δημιουργία της. Αναλύοντας και σχεδιάζοντας της αρχιτεκτονική της εφαρμογής, βλέπουμε την μεγαλύτερη εικόνα του έργου και παίρνουμε σημαντικές αποφάσεις για τον τρόπο ανάπτυξης σύμφωνα με τις ανάγκες και τους στόχους της εφαρμογής.

Η αρχιτεκτονική της εφαρμογής μας διαφέρει από τα παραδοσιακά μοντέλα αρχιτεκτονικής, όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο. Σε ένα παραδοσιακό μοντέλο αρχιτεκτονικής το επίπεδο παρουσίασης (presentation layer) χρειάζεται το επίπεδο της επιχειρηματικής λογικής (business logic layer) ώστε να αλληλεπιδράσει με το επίπεδο των δεδομένων (data layer). Αυτό επιτυγχάνεται μέσω των HTTP APIs.

Το μοντέλο της αρχιτεκτονικής που θα ακολουθήσουμε δεν απαιτεί την δημιουργία επιπέδου επιχειρηματικής λογικής καθώς το Firebase επιτρέπει στο επίπεδο παρουσίασης να επικοινωνήσει κατευθείαν με το επίπεδο δεδομένων μέσω του SDK (Software Development Kit). Η απαραίτητη επιχειρηματική λογική είναι αναπτυγμένη από τα προϊόντα του Firebase που καθιστούν την διαδικασία αυτή εφικτή καθώς παρέχουν την απαραίτητη λειτουργικότητα που χρειάζεται. Τα SDKs παρέχουν ένα πιο ευέλικτο και εύχρηστο περιβάλλον ανάπτυξης, καθώς δίνουν έτοιμες λύσεις και εργαλεία για την αλληλεπίδραση με διάφορες υπηρεσίες και πόρους.

Τις περισσότερες φορές, η χρήση ενός SDK μειώνει τον απαιτούμενο χρόνο ανάπτυξης και τις πιθανές παραμέτρους λάθους, καθώς παρέχει συγκεκριμένες λειτουργίες και πρωτόκολλα. Συχνά, τα SDKs είναι σχεδιασμένα για να είναι συμβατά με συγκεκριμένες πλατφόρμες ή υπηρεσίες, προσφέροντας ένα πιο σταθερό και αξιόπιστο περιβάλλον ανάπτυξης. Τα δύο πρότυπα αρχιτεκτονικής αναπαρίστανται στην εικόνα 4.1.1.



Εικόνα 4.1.1 Πρότυπα Αρχιτεκτονικής

Αντίθετα, η παλιά αρχιτεκτονική με χρήση HTTP API συνήθως απαιτεί τη χειροκίνητη κωδικοποίηση και αποκωδικοποίηση δεδομένων, καθώς και τη διαχείριση των αιτημάτων και των απαντήσεων HTTP.

Αυτή η αρχιτεκτονική μπορεί να είναι πιο ευέλικτη για ορισμένες χρήσεις, αλλά μπορεί επίσης να είναι πιο περίπλοκη και χρονοβόρα, καθώς απαιτεί περισσότερο κώδικα και διαχείριση από τον προγραμματιστή.

Συνολικά, το SDK παρέχει ένα πιο συνεπές και απλοποιημένο περιβάλλον ανάπτυξης, ενώ η παλιά αρχιτεκτονική με HTTP API μπορεί να είναι πιο ευέλικτη αλλά πιο περίπλοκη.

Ωστόσο, το Firebase εκτός από το SDK προσφέρει δυνατότητες για τη δημιουργία εξειδικευμένης επιχειρηματικής λογικής μέσω της πλατφόρμας του. Αυτό περιλαμβάνει τη δυνατότητα δημιουργίας και διαχείρισης προσαρμοσμένων λειτουργιών Cloud Functions που μπορούν να λειτουργούν ως API για την επικοινωνία με τη βάση δεδομένων ή άλλες υπηρεσίες του Firebase.

Οι Cloud Functions είναι προσαρμοσμένες λειτουργίες που μπορούν να αναπτυχθούν με χρήση γλωσσών προγραμματισμού όπως η JavaScript, TypeScript ή ακόμα και η Python, που εκτελούνται αυτόματα στον cloud υπολογιστικό χώρο του Firebase.

Με τις Cloud Functions, μπορούμε να αντιδράμε σε συγκεκριμένα γεγονότα στο περιβάλλον του Firebase, όπως η δημιουργία, η ενημέρωση ή η διαγραφή δεδομένων στη βάση δεδομένων ή η αυθεντικοποίηση χρηστών.

Επίσης, μπορούμε να χρησιμοποιήσουμε τις Cloud Functions για να εκτελέσουμε προσαρμοσμένη επιχειρηματική λογική, να αποστέλλουμε ειδοποιήσεις, να επεξεργαστούμε δεδομένα ή να αλληλεπιδράσουμε με εξωτερικούς πόρους και υπηρεσίες, εκτός του Firebase περιβάλλοντος.

4.2 Ανάπτυξη Λογισμικού Διεπαφής Χρήστη

Για την ανάπτυξη του λογισμικού της διεπαφής χρήστη θα χρησιμοποιήσουμε τα δομικά στοιχεία που παρέχει το Angular framework, τα αρχιτεκτονικά μοντέλα, τις σχεδιαστικές αλλά και τις γενικότερες αρχές που προτείνει.

Η Angular, όπως αναφέρθηκε, χρησιμοποιεί την γλώσσα προγραμματισμού TypeScript. Η TypeScript είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού που αναπτύχθηκε από τη Microsoft και αποτελεί ένα υπερσύνολο της γλώσσας JavaScript. Βασίζεται στη συντακτική δομή της JavaScript, συμπεριλαμβάνοντας και πρότυπα από την έκδοση ES6 (ECMAScript 2015) και νεότερα, και προσθέτει δυνατότητες όπως τύπους δεδομένων, interfaces, generics και πολλά άλλα, καθιστώντας τον κώδικα πιο ασφαλή και ευανάγνωστο.

Με το πρότυπο ES6 η JavaScript προσφέρει πληθώρα νέων λειτουργιών και βελτιώσεων, όπως οι βελτιωμένες συντακτικές δομές (όπως οι βελτιωμένες δυνατότητες στον τομέα των λειτουργικών προγραμματισμών, π.χ. arrow functions και destructuring), καινούργιες μεθόδους για την δημιουργία κλάσεων, modules, Promises, αναβαθμισμένες δομές δεδομένων (π.χ. Sets και Maps), και πολλά άλλα.

Ένα από τα κύρια χαρακτηριστικά της TypeScript είναι η υποστήριξη τύπων δεδομένων, που επιτρέπουν την καθορισμένη δήλωση τύπων για μεταβλητές, παραμέτρους συναρτήσεων, επιστρεφόμενες τιμές κλπ., προσφέροντας ένα επίπεδο ασφάλειας και ευκολίας στην ανάπτυξη του κώδικα.

Η TypeScript είναι συμβατή με τη JavaScript, πράγμα που σημαίνει ότι μπορούμε να χρησιμοποιήσουμε υπάρχοντα JavaScript κώδικα σε ένα έργο TypeScript χωρίς καμία αλλαγή. Επιπλέον, η TypeScript μεταγλωττίζεται σε καθαρή JavaScript, επιτρέποντάς σας να τη χρησιμοποιήσετε σε οποιοδήποτε περιβάλλον που υποστηρίζει JavaScript.

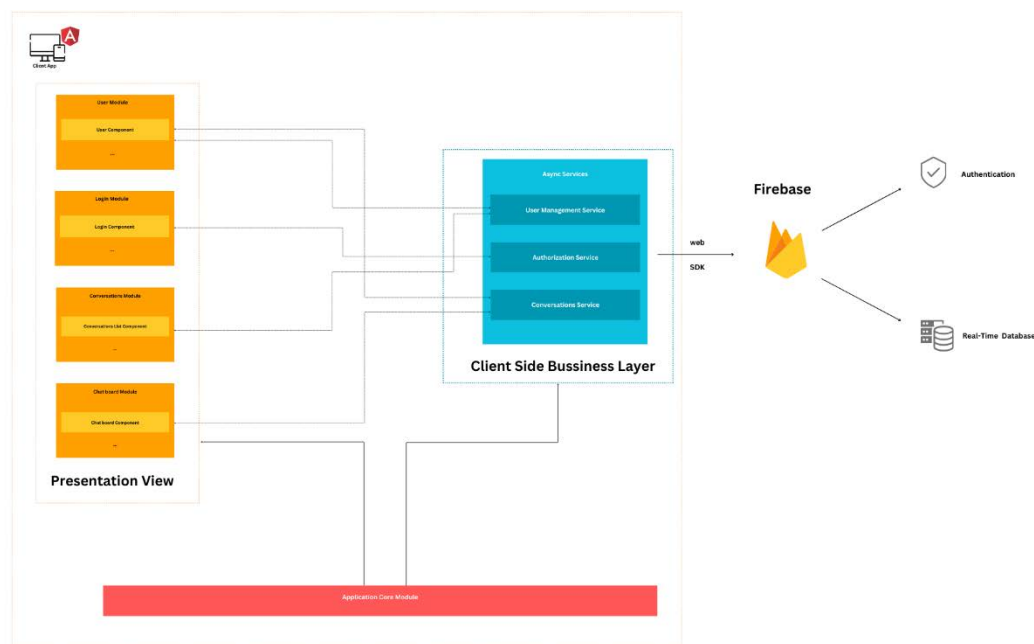
Τα βασικότερα δομικά στοιχεία που παρέχει η Angular είναι τα components, τα modules και τα services. Είναι ουσιαστικά κλάσεις με decorators οι οποίοι προσφέρουν συγκεκριμένη λειτουργικότητα. Ας τα περιγράψουμε αναλυτικά:

- **Components (Συστατικά):** Τα Components αντιπροσωπεύουν τα βασικά οπτικά στοιχεία του χρήστη στην Angular εφαρμογή. Κάθε Component αντιπροσωπεύει ένα συγκεκριμένο τμήμα του UI (Διεπαφής Χρήστη). Τα Components περιέχουν λογική TypeScript, δομή HTML και στυλ CSS.
- **Modules (Ενότητες):** Τα Modules είναι συλλογές συστατικών που συνδέονται λειτουργικά. Χρησιμοποιούνται για να οργανώσουν την εφαρμογή σε λογικές μονάδες. Κάθε Angular εφαρμογή έχει τουλάχιστον ένα κύριο module, το AppModule, που εκκινεί την εφαρμογή.
- **Services (Υπηρεσίες):** Τα Services παρέχουν λειτουργικότητα κοινή σε πολλά σημεία της εφαρμογής. Χρησιμοποιούνται για την ανακτήσεις δεδομένων από εξωτερικές πηγές, την κοινή χρήση λογικής μεταξύ Components, την αποθήκευση δεδομένων και άλλα. Τα Services είναι συνήθως ενσωματωμένα με την επιχειρηματική λογική της εφαρμογής.

Οι Decorators είναι μια σημαντική λειτουργία στην TypeScript που χρησιμοποιούνται στην Angular για να προσθέσουν μεταδεδομένα σε κλάσεις, όπως στα Components, Modules και Services. Με τους decorators, μπορούμε να προσθέσουμε metadata σε κλάσεις που η Angular θα χρησιμοποιήσει για να δημιουργήσει δυναμικά τα συστατικά της εφαρμογής μας.

Η χρήση των δομικών στοιχείων στην Angular βοηθάει στην οργάνωση, στην ευκολία συντήρησης και επέκτασης της εφαρμογής σας, καθώς και στη διαχωρισμό της λογικής της εφαρμογής σε μικρότερα και επαναχρησιμοποιήσιμα κομμάτια.

Στην παρακάτω εικόνα (4.2.1) αναπαρίσταται η αρχιτεκτονική της εφαρμογής. Στις επόμενες παραγράφους θα αναλύσουμε τα δομικά στοιχεία, την χρήση τους και τις λειτουργίες που συντελούν την εφαρμογή.



Εικόνα 4.2.1 Αρχιτεκτονική Εφαρμογής

Στην παραπάνω εικόνα (4.2.1) αρχικά παρατηρούμε τα modules, και ένα από τα components που περιέχουν. Σε αυτό το τμήμα έχει ακολουθηθεί το σχεδιαστικό πρότυπο "Feature Module Pattern" της Angular.

Σύμφωνα με αυτό το πρότυπο, η εφαρμογή διασπάται σε διάφορα feature modules, όπου κάθε ένα από αυτά τα modules είναι υπεύθυνο για μία συγκεκριμένη δυνατότητα ή λειτουργία της εφαρμογής.

Έτσι, για κάθε μέρος ή λειτουργία της εφαρμογής υπάρχει ένα αντίστοιχο module. Έτσι στην εφαρμογή μας, έχουμε ένα module για την εγγραφή και σύνδεση χρηστών **Login Module**, ένα άλλο module για την διαχείριση και αναπαράσταση του χρήστη **User Module**, ένα άλλο module για τη διαχείριση και αναπαράσταση των συνομιλιών **Conversations Module** και τέλος το **Chat board Module** για την αναπαράσταση της ανταλλαγής μηνυμάτων.

Τα feature modules προσφέρουν έναν τρόπο οργάνωσης της εφαρμογής που είναι επαναχρησιμοποιήσιμος, επιτρέποντας την ευκολότερη διαχείριση, συντήρηση και διαχωρισμό του κώδικα. Επίσης, βοηθούν στην αποφυγή της υπερφόρτωσης του κύριου AppModule και στην αποφυγή της δημιουργίας ενός μονολιθικού, μεγάλου module που είναι δύσκολο να διαχειριστεί.

Το Feature Module Pattern, επίσης, ενθαρρύνει την ανάπτυξη της εφαρμογής με βάση τον προσανατολισμό της λειτουργικότητας (feature-oriented development), κάτι που μπορεί να βοηθήσει στην καλύτερη οργάνωση και κατανόηση του κώδικα ακόμα και από διαφορετικές ομάδες ανάπτυξης.

Στην συνέχεια, παρατηρούμε τα services τα οποία περιλαμβάνουν την επιχειρηματική λογική της εφαρμογής μας, λαμβάνουν και διαχειρίζονται τα απαραίτητα δεδομένα σε δύο κατευθύνσεις, δηλαδή εκείνα που έρχονται από τη Firebase και τα διαμοιράζουν στα δομικά στοιχεία που τα χρειάζονται, αλλά και εκείνα που έρχονται από τις εισόδους του χρήστη και αποστέλλονται στην Firebase.

Τέλος, υπάρχει το core module το οποίο παρέχει δευτερεύοντα δομικά στοιχεία (components), βοηθητικές μεθόδους κλάσεις και μοντέλα τα οποία διαμοιράζονται σε όλη την εφαρμογή μας στα μέρη που τα χρειάζονται.

Το Core Module είναι ένα επιπλέον module που προστίθεται στην αρχιτεκτονική μιας εφαρμογής Angular, εκτός από τα Feature Modules που αναφέρθηκαν προηγουμένως. Περιέχει λειτουργικότητα που είναι κοινή για ολόκληρη την εφαρμογή και δεν ανήκει σε κάποια συγκεκριμένη δυνατότητα ή λειτουργία.

Έχοντας ένα Core Module είναι σημαντικό για διάφορους λόγους:

- **Οργάνωση κώδικα:** Το Core Module επιτρέπει την οργάνωση της κοινής λειτουργικότητας όπως διαχείριση διεθνούς στυλ, ρυθμίσεις, αυθεντικοποίηση, αναχώρηση, αυθεντικοποίηση κλπ., σε έναν ξεχωριστό χώρο, κάνοντας τον κώδικα πιο οργανωμένο.
- **Διαχείριση εξαρτήσεων:** Το Core Module μπορεί να χρησιμοποιηθεί για τη δήλωση και διαχείριση των εξαρτήσεων που είναι κοινές για ολόκληρη την εφαρμογή, όπως η χρήση τρίτων βιβλιοθηκών.
- **Κοινόχρηστα στοιχεία:** Το Core Module μπορεί να περιέχει στοιχεία όπως υπηρεσίες, interceptors, guards κλπ., που χρησιμοποιούνται από πολλά σημεία της εφαρμογής.

- Προστασία από διπλή φόρτωση: Χρησιμοποιώντας ένα Core Module, μπορούμε να αποφύγουμε την διπλή φόρτωση των εξαρτημένων υπηρεσιών, interceptors και άλλων στοιχείων, καθώς το Core Module θα φορτώνεται μόνο μία φορά σε όλη την εφαρμογή.

Συνολικά, το Core Module παρέχει μια καλή οργανωτική δομή και επιτρέπει την ευκολότερη διαχείριση της κοινής λειτουργικότητας σε μια εφαρμογή Angular.

4.2.1 Επιχειρηματική Λογική και Services

Η επιχειρηματική λογική στο μέρος της διεπαφής χρήστη, αποτελείται από τρεις υπηρεσίες (services).

Αρχικά, το σημαντικότερο service είναι το authorization-service. Το εν λόγω service περιλαμβάνει την επιχειρηματική λογική για την αυθεντικοποίηση του χρήστη, την δημιουργία λογαριασμού, την σύνδεση και αποσύνδεση στην εφαρμογή. Πιο συγκεκριμένα, περιλαμβάνει τις μεθόδους που επικοινωνούν με τις Firebase υπηρεσίες, Authorization και Real-Time Database. Αναλυτικότερα, παρέχει την μέθοδο **singUpUser()** που δημιουργεί έναν λογαριασμό για τον χρήστη χρησιμοποιώντας email, password και first name. Την **signInUser()** η οποία συνδέει τον χρήστη όταν εισάγει το σωστό email και password, την **signOutUser()** η οποία αποσυνδέει τον χρήστη όταν αποσυνδεθεί πατώντας το σχετικό κουμπί ή όταν κλείσει την καρτέλα του browser. Επίσης, έχει την **setAuthPersistence()** η οποία θέτει το persistence policy και διασφαλίζει ότι ο χρήστης θα παραμείνει συνδεδεμένος στην εφαρμογή μας, όσο διαρκεί το session, δηλαδή όσο έχει ανοικτό τον browser με τον οποίο έχει συνδεθεί στην εφαρμογή μας. Τέλος, οι υπόλοιπες ασύγχρονες μέθοδοι ελέγχουν την κατάσταση αυθεντικοποίησης του χρήστη, δηλαδή εάν ο χρήστης είναι συνδεδεμένος, και στην συνέχεια διαμοιράζουν την κατάσταση αυτή σε άλλα κομμάτια της εφαρμογής μας που την χρειάζονται. Αναφορικά θα αναλύσουμε την μέθοδο **singUpUser()** που αναπαρίσταται παρακάτω, που η λειτουργία της είναι η δημιουργία ενός λογαριασμού για έναν νέο χρήστη.

```
signUpUser(email: string, password: string, firstName: string) {
  return this.fireAuth.createUserWithEmailAndPassword(
    email.trim(),
    password.trim()
  ).then((user) => {
    const newUser = {
      userId: user.user?.uid,
      userName: firstName
    }
    this.userLoggedIn = false;
    this.fireAuth.signOut().then(() =>
```

```

        this.loginDialog.open(LoginDialogComponent, { width: '400' });
        this.firebaseio.list('users').push(newUser)
    })
}

```

Αρχικά χρησιμοποιούμε την μέθοδο `createUserWithEmailAndPassword()`, η οποία παρέχεται από την κλάση `AngularFireAuth`, με τις κατάλληλες παραμέτρους, και η οποία μας επιστρέφει ένα promise. Όταν το promise γίνεται resolve, δημιουργούμε έναν νέο χρήστη με το μοναδικό του ID, και το username του. Ενώ στην αποσυνδέουμε τον χρήστη και του ζητάμε να συνδεθεί στον login component ενώ σαν τελευταίο βήμα προσθέτουμε τον νέο χρήστη στην βάση δεδομένων μας στην λίστα users. Στο login component που καλεί αυτή την μέθοδο κάνουμε και διαχείριση σφαλμάτων του promise και ενημερώνουμε τον χρήστη με τα αντίστοιχα μηνύματα λάθους όπως φαίνεται παρακάτω.

```

createNewUser() {
    const { email, password, firstName } = this.loginForm.value;
    this.authorizationService.signUpUser(email, password, firstName)
        .then(() => {
            this.loginForm.reset();
            this.newUser = true;
            this.loginForm.removeControl('firstName');
        })
        .catch((error: { code: string, message: string }) => {
            if (error.code === 'auth/weak-password')
                this.error = 'The password should at least have 6 characters';

            if (error.code === 'auth/email-already-in-use') {
                this.error = 'The email address is already in use by another account.'
            }
        })
    }
}

```

Στην συνέχεια, θα αναλύσουμε την μέθοδο `signInUser()` που αναπαρίσταται παρακάτω και η λειτουργία της είναι η αυθεντικοποίηση του χρήστη.

```

signInUser(email: string, password: string) {
    return this.fireAuth.signInWithEmailAndPassword(email, password)
        .then((user: any) => {
            localStorage.setItem('user', user.user?.uid);
            this.userLoggedIn = true;
            this.router.navigate(['/chat-board'])
        })
    }
}

```

Στην περίπτωση αυτή θα χρησιμοποιούμε την μέθοδο `signInWithEmailAndPassword()` η οποία παρέχεται επίσης από την κλάση `AngularFireAuth`, με τις κατάλληλες παραμέτρους, και η οποία μας επιστρέφει ένα promise. Όταν το promise γίνεται resolve, θέτουμε ότι ο χρήστης είναι

συνδεδεμένος και κάνουμε ανακατεύθυνση στο chat board ώστε να μπορεί να ξεκινήσει συνομιλίες. Στο login component που καλεί αυτή την μέθοδο κάνουμε και διαχείριση σφαλμάτων του promise και ενημερώνουμε τον χρήστη με τα αντίστοιχα μηνύματα λάθους όπως φαίνεται παρακάτω.

```
login(): void {
  const { email, password } = this.loginForm.value
  this.authorizationService.signInUser(email, password)
    .catch((error: { code: string, message: string }) => {
      switch (error.code) {
        case 'auth/user-not-found':
          this.error = 'The email address is invalid. The user does not exist or has been removed.';
          break;
        case 'auth/wrong-password':
          this.error = 'The password is invalid.';
          break;
        case 'auth/invalid-email':
          this.error = 'The email address is invalid.';
          break;
        default:
          this.error = 'The credentials are invalid.';
          break;
      }
    });
}
```

Στην συνέχεια, ακολουθεί το conversations-service. Το εν λόγω service περιλαμβάνει την επιχειρηματική λογική για την δημιουργία συζητήσεων και την ανταλλαγή μηνυμάτων. Πιο συγκεκριμένα περιλαμβάνει τις μεθόδους που επικοινωνούν με την Firebase υπηρεσία Real-Time Database. Ας αναλύσουμε τις κύριες μεθόδους.

Αρχικά, έχουμε την createChatRoom() η οποία δημιουργεί ένα conversation στην λίστα conversations στη βάση δεδομένων και ανακατευθύνει τον χρήστη στην αντίστοιχη σελίδα ώστε να ανταλλάξει μηνύματα. Στην συνέχεια έχουμε την getConversations() και την getConversationsById() που επιστρέφουν τα διαθέσιμα conversations από την βάση δεδομένων αλλά και ένα συγκεκριμένο conversation στέλνοντας ένα query στην βάση με το μοναδικό id της συνομιλίας. Τέλος, έχουμε την sendMessage() η οποία προσθέτει το σώμα του μηνύματος στην αντίστοιχη συνομιλία στην λίστα των συνομιλιών και την getMessagesForConversation() ή οποία επιστρέφει όλα τα μηνύματα για μία συγκεκριμένη συνομιλία μεταξύ δύο χρηστών.

Εν κατακλείδι, έχουμε το user-management service το οποίο έχει απλά μια μέθοδο για να ανακτούμε την λίστα με τους διαθέσιμους χρήστες από την βάση δεδομένων ώστε ο κάθε χρήστης να έχει την δυνατότητα να επιλέξει τον χρήστη που επιθυμεί για να συνομιλήσει.

4.2.2 Ενότητες και Δομικά Στοιχεία

Όπως αναφέρθηκε παραπάνω, η εφαρμογή αποτελείται από ενότητες και δομικά στοιχεία. Οι ενότητες είναι δομημένες ακολουθώντας το Feature Module Pattern μαζί με τα αντίστοιχα components τους. Ωστόσο τα components ακολουθούν και εκείνα ένα συγκεκριμένο πρότυπο, το "Container Presentation Pattern".

Το "Container Presentation Pattern" είναι ένας σχεδιαστικός τρόπος που χρησιμοποιείται συχνά στην ανάπτυξη εφαρμογών με Angular. Αυτό το πρότυπο διαχωρίζει τα συστατικά της εφαρμογής σε δύο κατηγορίες: τα Containers και τα Presentational Components.

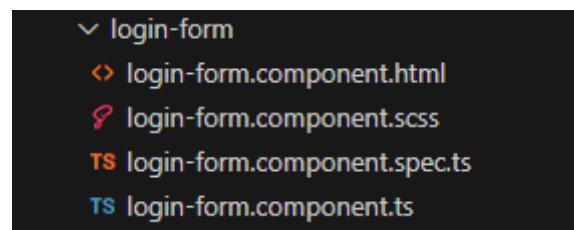
- **Containers (Ελεγκτές):** Τα Containers είναι συνήθως Angular Components που έχουν ως κύριο ρόλο τη διαχείριση της λογικής της εφαρμογής. Αυτά τα συστατικά είναι υπεύθυνα για την ανάκτηση δεδομένων από τις υπηρεσίες, τη διαχείριση της κατάστασης της εφαρμογής και τη μεταφορά δεδομένων στα Presentational Components για την εμφάνισή τους.
- **Presentational Components (Δομικά Στοιχεία Παρουσίασης):** Τα Presentational Components είναι Angular Components που είναι υπεύθυνα αποκλειστικά για την παρουσίαση δεδομένων και την αλληλεπίδραση με τον χρήστη. Δεν έχουν καμία γνώση του πώς λαμβάνονται τα δεδομένα ή ποια λογική εφαρμόζεται για να τα επεξεργαστεί.

Τα πλεονεκτήματα του Container Presentation Pattern στην Angular περιλαμβάνουν:

- **Διαχωρισμός της Ευθύνης (Separation of Concerns):** Η διάκριση της λογικής της εφαρμογής από την παρουσίαση των δεδομένων καθιστά τον κώδικα πιο οργανωμένο και ευανάγνωστο.
- **Επαναχρησιμοποίηση Κώδικα (Code Reusability):** Τα Presentational components μπορούν να επαναχρησιμοποιηθούν εύκολα σε διαφορετικά μέρη της εφαρμογής καθώς δεν είναι συνδεδεμένα με συγκεκριμένη λογική.

- **Ευκολία Μονάδος Δοκιμών (Unit Testing):** Το Container Presentation Pattern καθιστά ευκολότερη τη δοκιμή των διάφορων μονάδων του κώδικα, καθώς η λογική και η παρουσίαση είναι διαχωρισμένες.
- **Βελτιωμένη Συντηρησιμότητα (Maintainability):** Ο διαχωρισμός της λογικής της εφαρμογής από την παρουσίαση των δεδομένων καθιστά τον κώδικα πιο ευέλικτο και εύκολο στη συντήρηση.

Ένα component όπως αναφέρθηκε αποτελείται από ένα HTML αρχείο όπου περιέχει τον HTML κώδικα, ένα TypeScript αρχείο που ουσιαστικά είναι η κλάση του component και εκεί εμπεριέχεται η λογική του component και ένα SCSS αρχείο που περιέχει SCSS κώδικα που αφορά τα styles του component. Επίσης έχει και ένα αρχείο με την κατάληξη (.spec.ts) το οποίο είναι ένα TypeScript αρχείο όπου εκεί μπορούμε να αναπτύξουμε τον κώδικα για τα διάφορα unit και integration tests του component μας. Το αρχείο δημιουργείται αυτόματα όταν τρέχουμε εντολές για την δημιουργία των components μέσω του Angular CLI που αναφέραμε σε προηγούμενο κεφάλαιο. Στην παρούσα πτυχιακή εργασία δεν γράφτηκαν test cases, καθώς είναι εκτός θέματος της πτυχιακής μας. Στην παρακάτω εικόνα (4.2.2) παρατηρούμε τα αρχεία που απαρτίζουν ένα component.



Εικόνα 4.2.2 Αρχεία Component

Αναφορικά, θα περιγράψουμε δύο components της εφαρμογής μας για να αναδείξουμε την παραπάνω αρχιτεκτονική. Έτσι έχουμε τα login-form και login-page components. Το login form είναι ένα component που περιέχει τα πεδία φόρμας που ο χρήστης πρέπει να συμπληρώσει ώστε να εγγραφεί αλλά και να συνδεθεί στην εφαρμογή μας. Περιέχει επίσης τις απαραίτητες μεθόδους ενώ επικοινωνεί και λαμβάνει state και δεδομένα και από τα αντίστοιχα services όπως φαίνεται παρακάτω.

```
import { Component, OnInit } from '@angular/core';
```

```

import { FormBuilder, FormControl, FormGroup, Validators } from '@angular/forms';
import { AuthorizationService } from 'src/app/core/authorization.service';

@Component({
  selector: 'app-login-form',
  templateUrl: './login-form.component.html',
  styleUrls: ['./login-form.component.scss']
})
export class LoginFormComponent implements OnInit {

  loginForm: FormGroup;
  error!: string;
  newUser: boolean = true;
  showPassword = false;
  images: File[] = [];

  firstNameControl: FormControl;

  constructor(private fb: FormBuilder,
    private authorizationService: AuthorizationService) {

    this.firstNameControl = this.fb.control('', this.newUser ? null : Validators.required),
    this.loginForm = this.fb.group({
      email: this.fb.control('', Validators.compose([Validators.required, Validators.pattern('^[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,4}$'))),
      password: this.fb.control('', Validators.required),
      firstName: this.firstNameControl
    })
  }

  ngOnInit(): void { }

  createNewUser() {
    const { email, password, firstName } = this.loginForm.value;
    this.authorizationService.signUpUser(email, password, firstName)
      .then(() => {
        this.loginForm.reset();
        this.newUser = true;
        this.loginForm.removeControl('firstName');
      })
      .catch((error: { code: string, message: string }) => {
        if (error.code === 'auth/weak-password')
          this.error = 'The password should at least have 6 characters';

        if (error.code === 'auth/email-already-in-use') {
          this.error = 'The email address is already in use by another account.'
        }
      })
  }

  login(): void {
    const { email, password } = this.loginForm.value
    this.authorizationService.signInUser(email, password)
      .catch((error: { code: string, message: string }) => {
        switch (error.code) {
          case 'auth/user-not-found':
            this.error = 'The email address is invalid. The user does not exist or has been removed.';
            break;
          case 'auth/wrong-password':

```

```

        this.error = 'The password is invalid.';
        break;
    case 'auth/invalid-email':
        this.error = 'The email address is invalid.';
        break;
    default:
        this.error = 'The credentials are invalid.';
        break;
    }
    });
}

newUserChange() {
    this.newUser = !this.newUser;
    this.loginForm.reset();
    if(this.error) {this.error=''}
    this.newUser ? this.loginForm.removeControl('firstName') : this
    if (this.newUser) {
        this.loginForm.removeControl('firstName');
    } else {
        this.loginForm.addControl('firstName', this.firstNameControl)
    }
}

togglePasswordVisibility() {
    this.showPassword = !this.showPassword
}

onSelect(event: any) {
    this.images.push(...event.addedFiles);
}

onRemove(f: any) {
    this.images.pop()
}
}

```

Το login page component είναι το landing page της εφαρμογής μας. Δηλαδή η πρώτη σελίδα που έρχεται σε επαφή ο χρήστης ώστε να αλληλεπιδράσει με την εφαρμογή μας, όταν δηλαδή πληκτρολογεί την διεύθυνση στον περιηγητή του για να μεταβεί στην εφαρμογή μας. Το component αυτό χρησιμοποιεί το login form component αλλά δεν περιέχει καθόλου λογική, και επομένως αγνοεί το πως γίνεται η αυθεντικοποίηση ή η εγγραφή του χρήστη κλπ. Το μόνο που περιέχει είναι SCSS styles και HTML κώδικα ενώ το TypeScript αρχείο του έχει απλά μία άδεια κλάση, όπως φαίνεται παρακάτω.

```

import { Component } from '@angular/core';

@Component({
    selector: 'app-login-page',
    templateUrl: './login-page.component.html',
    styleUrls: ['./login-page.component.scss']
})
export class LoginPageComponent { }

```

Η τεχνική αυτή χρησιμοποιήθηκε για την πλειονότητα των components, ωστόσο υπάρχουν και components τα οποία είναι διαμοιραζόμενα και γενικά και επαναχρησιμοποιήσιμα ενώ άλλα όπως το navigation bar είναι βασικά components τα οποία βρίσκονται σε όλες τις σελίδες της εφαρμογής, η λειτουργία τους δεν είναι συνδεδεμένη αποκλειστικά με κάποιο feature και επομένως γι' αυτό βρίσκονται και στο core module της εφαρμογής.

Τέλος, ένα βασικό δομικό στοιχείο της εφαρμογής μας είναι μια ιδιαίτερη κλάση που έχει σημαντικό ρόλο στην εφαρμογή μας. Αυτή είναι η κλάση AuthGuard και είναι ένα Angular Route Guards.

Τα Angular Route Guards είναι μηχανισμοί που χρησιμοποιούνται για την προστασία και τον έλεγχο της πλοήγησης σε μια εφαρμογή Angular. Βασικά, επιτρέπουν την εφαρμογή της λογικής που καθορίζει εάν ένας χρήστης μπορεί να προβεί σε μια συγκεκριμένη μετάβαση σε μια σελίδα ή σε ένα route.

Καθώς οι χρήστες εκτελούν εργασίες στην εφαρμογή, χρειάζεται να μετακινούνται μεταξύ των διαφορετικών προβολών που έχουμε ορίσει. Για να διαχειριστούμε την πλοήγηση από μια προβολή στην επόμενη, χρησιμοποιούμε τον Angular Router. Ο Router επιτρέπει την πλοήγηση με την ερμηνεία ενός URL περιηγητή ως οδηγία για αλλαγή προβολής.

Οι βασικοί τύποι των Angular Route Guards είναι οι εξής:

- **CanActivate:** Επιτρέπει ή αποτρέπει την πλοήγηση σε ένα route βάσει μιας συνθήκης. Χρησιμοποιείται για να ελέγξει αν ένας χρήστης έχει πρόσβαση σε μια σελίδα.
- **CanLoad:** Επιτρέπει ή αποτρέπει την φόρτωση ενός module βάσει μιας συνθήκης. Χρησιμοποιείται για να ελέγξει αν ένας χρήστης έχει πρόσβαση σε μια σελίδα.
- **CanActivateChild:** Παρόμοιο με το CanActivate, αλλά ελέγχει την πρόσβαση σε child routes.
- **CanDeactivate:** Επιτρέπει ή αποτρέπει την αποχώρηση από ένα route. Χρησιμοποιείται, για παράδειγμα, για επιβεβαίωση αποχώρησης από μια φόρμα πριν την υποβολή.
- **Resolve:** Χρησιμοποιείται για να φορτώσει δεδομένα πριν από την εμφάνιση μιας σελίδας. Εξασφαλίζει ότι τα απαραίτητα δεδομένα είναι διαθέσιμα πριν από την εμφάνιση της σελίδας.

Οι Route Guards είναι χρήσιμοι για την επιβολή αυθεντικοποίησης και αδειών πρόσβασης σε συγκεκριμένες σελίδες, τη διαχείριση αλλαγών κατά την πλοήγηση και τον έλεγχο των δεδομένων που φορτώνονται πριν από την εμφάνιση μιας σελίδας.

Στην δική μας περίπτωση χρησιμοποιούνται 2 μέθοδοι για την ανακατεύθυνση του χρήστη. Η πρώτη η `canActivate()`, ελέγχει εάν ο χρήστης είναι συνδεδεμένος και εάν προσπαθήσει να μεταβεί στην σελίδα για να κάνει login η εγγραφή τον ανακατευθύνει στο chat-board που λειτουργεί και ως landing page αφού ο χρήστης συνδεθεί. Η δεύτερη η `canLoad()` ελέγχει εάν ο χρήστης είναι συνδεδεμένος ώστε να μπορεί να μεταβεί στο chat board, να δει την λίστα των διαθέσιμων χρηστών, την λίστα των συνομιλιών και να ανταλλάξει μηνύματα. Εάν για οποιοδήποτε λόγο ο χρήστης δεν είναι συνδεδεμένος, ή γενικότερα δεν υπάρχει πρόσβαση, τον ανακατευθύνει στην αρχική σελίδα ώστε να συνδεθεί ή να δημιουργήσει λογαριασμό. Ο λόγος που χρησιμοποιήθηκε το `canLoad()` είναι για λόγους καλύτερης απόδοσης και λειτουργίας καθώς η εφαρμογή δεν θα φορτώσει το module που δεν χρειάζεται.

Δομικά Στοιχεία και Βιβλιοθήκες

Για την ανάπτυξη του project χρησιμοποιήθηκαν και βιβλιοθήκες που αξίζει να αναφερθούν καθώς παρέχουν έτοιμα δομικά στοιχεία και δυνατότητες που συμβάλουν στην αποδοτικότερη και γρηγορότερη ανάπτυξη του έργου. Όπως αναφέρθηκε σε προηγούμενο κεφάλαιο χρησιμοποιήθηκε η `@angular/fire` ώστε να ενσωματώσουμε την Firebase στο Angular application μας.

Στην συνέχεια, χρησιμοποιήθηκε η Angular Material. Η Angular Material είναι μια βιβλιοθήκη για το Angular Framework που παρέχει ένα πλήρες σετ από δομικά στοιχεία, σχεδιασμένα με τις αρχές του υλικού σχεδιασμού της Google, γνωστές και ως Material Design.

Με τη χρήση της Angular Material, μπορούμε να δημιουργήσουμε εύκολα και γρήγορα εφαρμογές φιλικές προς τον χρήστη, με μοντέρνο, καθαρό και ευέλικτο σχεδιασμό, που είναι συμβατές με διάφορες πλατφόρμες και συσκευές.

Η Angular Material περιλαμβάνει μια πληθώρα στοιχείων, όπως κουμπιά, φόρμες, μενού, πλαίσια διαλόγου, πίνακες, καρτέλες, sliders, κ.ά., που είναι έτοιμα για χρήση και συνήθως απαιτούν λίγο ή καθόλου προσαρμογή για να ενσωματωθούν στις εφαρμογές.

Η χρήση της Angular Material μας βοηθάει να δημιουργήσουμε ένα προσαρμοσμένο θέμα (Custom Theming) προσφέροντας έτσι μια συνεπή εμπειρία χρήστη. Το Theming αναφέρεται στη διαδικασία προσαρμογής της εμφάνισης μιας εφαρμογής ή ιστοσελίδας, συμπεριλαμβανομένων των χρωμάτων, των γραμματοσειρών και των στοιχείων σχεδιασμού, ώστε να αντιστοιχεί σε ένα συγκεκριμένο στυλ ή brand.

Στην Angular Material, η προσαρμογή του theming είναι αρκετά εύκολη και γίνεται μέσω του Angular Material Theming System. Με αυτό το σύστημα, μπορούμε να προσαρμόσουμε το

στυλ των στοιχείων της Angular Material κατά το δοκούν, συμπεριλαμβανομένων των χρωμάτων, των τυπογραφικών στοιχείων και των στυλ επιφάνειας (surface styles). Μερικοί βασικοί τρόποι προσαρμογής του theming στην Angular Material περιλαμβάνουν την ακόλουθη διαδικασία:

- **Προσαρμογή των Χρωμάτων (Color customization):** Μπορούμε να ορίσουμε τα δικά μας χρώματα ή να επιλέξουμε από ένα σύνολο προκαθορισμένων θεμάτων χρωμάτων.
- **Τυπογραφική Προσαρμογή (Typography customization):** Μπορούμε να ορίσουμε διαφορετικές γραμματοσειρές, μεγέθη γραμμάτων και γραμματοσειρές για το κείμενο της εφαρμογής.
- **Προσαρμογή των Στοιχείων Επιφάνειας (Surface customization):** Μπορούμε να ορίσετε τα στυλ των διάφορων στοιχείων επιφάνειας, όπως κάρτες, κουμπιά και φόρμες.

Αυτή η παραμετροποίηση επιτρέπει στις εφαρμογές Angular Material να συμβαδίζουν με το γενικό σχεδιασμό ή το brand της εταιρείας ή του προϊόντος μας. Για το προσαρμοσμένο theme επιλέχθηκαν μερικές από τις διαθέσιμες χρωματικές παλέτες που παρέχει η Angular Material.

Τέλος, χρησιμοποιήθηκε η Bootstrap. Η Bootstrap είναι μια ανοιχτού κώδικα βιβλιοθήκη CSS και JavaScript που παρέχει έτοιμα στυλ και συστατικά για τη δημιουργία αντικειμένων διαδικτυακής περιήγησης. Δημιουργήθηκε από τους προγραμματιστές του Twitter και αναπτύχθηκε για να διευκολύνει τη δημιουργία ενός σύγχρονου και ανταγωνιστικού σχεδιασμού ιστοσελίδων.

Τα βασικότερα πλεονεκτήματα της βιβλιοθήκης αυτής είναι:

- **Ταχύτητα Ανάπτυξης:** Η Bootstrap παρέχει ένα σύνολο έτοιμων στυλ και συστατικών που μας επιτρέπουν να αναπτύξουμε γρήγορα και εύκολα ιστοσελίδες χωρίς να ξεκινούμε από το μηδέν.
- **Απόκριση (Responsiveness):** Η Bootstrap προσφέρει ένα σύστημα grid που είναι εύκολο να προσαρμοστεί σε διάφορες συσκευές και οθόνες, κάνοντας τις ιστοσελίδες να φαίνονται καλά σε κινητά, τάμπλετ και υπολογιστές.

- **Προσαρμοσμένος Σχεδιασμός:** Με την Bootstrap, μπορούμε εύκολα να προσαρμόσουμε τα styles και να γράψουμε αποδοτικότερη CSS χρησιμοποιώντας έτοιμες κλάσεις και δυνατότητες από εκείνες που προσφέρει η βιβλιοθήκη.
- **Συνοχή:** Χρησιμοποιώντας το Bootstrap, οι ιστοσελίδες μπορούν να διατηρήσουν συνοχή στον σχεδιασμό τους, καθώς το ίδιο στυλ και οι ίδιοι κανόνες εφαρμόζονται σε διάφορα μέρη της ιστοσελίδας.

Στην συνέχεια θα αναφερθούμε στην παραμετροποίηση του Firebase.

4.3 Παραμετροποίηση Firebase

Όπως αναφέρθηκε σε προηγούμενα κεφάλαια θα χρησιμοποιηθούν οι υπηρεσίες του Firebase και για ανάπτυξη σε τοπικό υπολογιστή θα χρησιμοποιηθούν οι emulators. Καθώς για τις ανάγκες της εφαρμογής μας δεν θα αναπτύξουμε κάποιο API ή κάποια βάση δεδομένων θα αναφερθούμε στις παραμετροποιήσεις που έγιναν στις υπηρεσίες της Firebase όπως την υπηρεσία του Authentication και την υπηρεσία του Real-Time Database.

Αυθεντικοποίηση Χρήστη - Authentication

Η αυθεντικοποίηση του χρήστη αφορά την δημιουργία λογαριασμού και την σύνδεση του χρήστη στην εφαρμογή μας, ώστε να μπορεί να ανταλλάσσει μηνύματα με άλλους συνδεδεμένους χρήστες της εφαρμογής. Η Firebase μας προσφέρει μια πληθώρα επιλογών για τον τρόπο που ένας χρήστης μπορεί να δημιουργήσει λογαριασμό και να συνδεθεί. Ο πιο απλός και συνήθης τρόπος, που χρησιμοποιήθηκε και για τις ανάγκες αυτής της εφαρμογής είναι η εγγραφή του χρήστη μέσω e-mail και password.

Η υπηρεσία αυτή μας δίνει την δυνατότητα να στέλνουμε αυτοματοποιημένα επίσης e-mail επιβεβαίωσης ώστε να ελέγχουμε ότι το e-mail υπάρχει και ο χρήστης έχει πρόσβαση σε αυτό και στην συνέχεια να επιτρέπουμε την τελική αυθεντικοποίησή του και σύνδεση στην εφαρμογή, λειτουργία που διαθέτουν οι περισσότερες δημοφιλείς σύγχρονες εφαρμογές. Καθώς η εφαρμογή μας είναι στα πλαίσια ακαδημαϊκής εργασίας, κάτι τέτοιο δεν χρησιμοποιήθηκε.

Μεταξύ άλλων δημοφιλών και σύγχρονων τρόπων σύνδεσης που παρέχει η Firebase είναι και η δυνατότητα σύνδεσης με λογαριασμό Google, η σύνδεση με λογαριασμό Facebook και μερικών άλλων εφαρμογών. Τέλος, παρέχει τον τρόπο σύνδεσης με αριθμό τηλεφώνου αλλά και με προσαρμοσμένους τρόπους όπως με Identity Platforms (OpenID Connect και SAML πρωτόκολλα) και πολυπαραγοντική αυθεντικοποίηση (multi-factor authentication), που όμως δεν είναι διαθέσιμοι στο δωρεάν πλάνο χρήσης του Firebase.

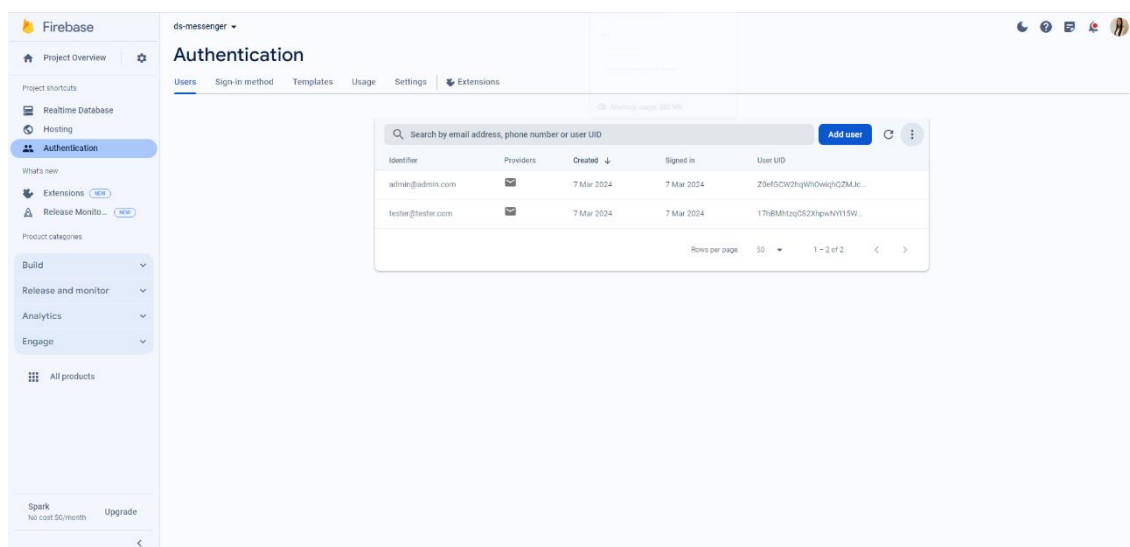
Το Firebase Authentication παρέχει, επίσης, λειτουργίες για τη διαχείριση των χρηστών, όπως η διαχείριση των προφίλ χρηστών, ο έλεγχος πρόσβασης, η ανάκτηση κωδικού πρόσβασης και άλλα.

Με το Firebase Authentication, οι προγραμματιστές μπορούμε να ενσωματώσουμε γρήγορα και ασφαλώς λειτουργίες αυθεντικοποίησης στην εφαρμογή μας χωρίς την ανάγκη να υλοποιήσουμε ένα πλήρες σύστημα αυθεντικοποίησης από την αρχή.

Για τη σύνδεση με email και κωδικό πρόσβασης, το Firebase χρησιμοποιεί ασφαλείς μεθόδους και αλγορίθμους κρυπτογράφησης για την αποθήκευση των κωδικών πρόσβασης των χρηστών, συχνά με τη χρήση μοντέλων όπως το SCRYPT. Αυτό σημαίνει ότι τα προσωπικά δεδομένα του χρήστη, συμπεριλαμβανομένων των κωδικών πρόσβασης, αποθηκεύονται στη βάση δεδομένων του Firebase με έναν τρόπο που είναι ασφαλής και ανθεκτικός σε επιθέσεις.

Ως αποτέλεσμα, τα προσωπικά δεδομένα των χρηστών παραμένουν προστατευμένα και δεν είναι προσβάσιμα από άλλους χωρίς την κατάλληλη αυθεντικοποίηση και εξουσιοδότηση.

Στην παρακάτω εικόνα βλέπουμε το περιβάλλον διαχείρισης στην κονσόλα (console) του firebase, που περιλαμβάνει την λίστα των εγγεγραμμένων χρηστών της εφαρμογής μας καθώς και τα διάφορα μενού με τις επιλογές παραμετροποίησης και τις διαθέσιμες λειτουργίες.



Εικόνα 4.3.1 Firebase Console

Κάθε χρήστης που εγγράφεται στην εφαρμογή μας θα προστίθεται σε αυτή την λίστα αλλά και στην βάση δεδομένων μας με ένα μοναδικό αναγνωριστικό το οποίο δημιουργείται αυτόματα από το σύστημα.

Βάση Δεδομένων Πραγματικού Χρόνου – Real-Time Database

Όπως αναφέραμε η βάση δεδομένων που θα χρησιμοποιηθεί θα είναι η υπηρεσία Realtime Database.

Η Firebase Realtime Database είναι μια cloud βάση δεδομένων NoSQL που μας επιτρέπει να αποθηκεύουμε και να συγχρονίζουμε δεδομένα σε πραγματικό μεταξύ του διακομιστή (server) και του client. Τα δεδομένα αποθηκεύονται σε μορφή δέντρου JSON και είναι προσβάσιμα ακόμα και όταν η εφαρμογή είναι εκτός σύνδεσης.

Μια από τις βασικές της δυνατότητες είναι η άμεση συγχρονισμένη και πραγματική χρονική ανταλλαγή δεδομένων, η οποία επιτρέπει την άμεση ενημέρωση των clients μετά από αλλαγές στη βάση δεδομένων. Με άλλα λόγια, κάθε φορά που ένα δεδομένο αλλάζει, κάθε συνδεδεμένη συσκευή λαμβάνει αυτή την ενημέρωση μέσα σε εκατοστά του δευτερολέπτου.

Ως εκ τούτου, η Realtime Database είναι κατάλληλη για εφαρμογές που απαιτούν άμεση συγχρονισμένη ανταλλαγή δεδομένων σε πραγματικό χρόνο, όπως τα chat applications.

Ωστόσο υπάρχουν και κάποιες προκλήσεις:

- Καθυστέρηση (Latency): Η αύξηση της κίνησης και των χρηστών μπορεί να οδηγήσει σε αυξημένη καθυστέρηση, καθώς η Realtime Database μεταφέρει δεδομένα αμέσως μετά από αλλαγές στη βάση δεδομένων.

- **Κόστος:** Η χρήση της Realtime Database στο δωρεάν πρόγραμμα έχει κάποια όρια κίνησης δεδομένων. Αν η εφαρμογή μας γίνει δημοφιλής και χρησιμοποιείται από πολλούς χρήστες η στοχεύουμε στον να γίνει επαγγελματική θα χρειαστεί να μεταβούμε σε κάποιο επί πληρωμή πλάνο για να μπορούμε να εξυπηρετήσουμε αποτελεσματικά τις ανάγκες των χρηστών χωρίς προβλήματα. Σε αυτό το πλάνο πρέπει να λάβουμε καθοριστικές αποφάσεις για τις ανάγκες και το budget, γιατί το κόστος μπορεί να αυξηθεί ραγδαία, καθώς χρεώνεται βάσει της κίνησης δεδομένων και του όγκου των δεδομένων.
- **Δομή Δεδομένων:** Λόγω της δομής JSON, η διαχείριση σύνθετων δεδομένων μπορεί να είναι πιο δύσκολη και περίπλοκη για αυτό συνίσταται να έχουμε όσο το δυνατόν πιο απλή και επίπεδη δομή και να αποφύγουμε την δημιουργία εμφωλιασμένων δομών δεδομένων.

Η δομή της βάσης δεδομένων της εφαρμογής μας, καθώς αποθηκεύει τα δεδομένα σε δένδροειδή μορφή JSON αποτελείται από τρεις (3) κόμβους με τα σχετικά αντιπροσωπευτικά κλειδιά τους (key-value pair) οι οποίοι είναι ο κόμβος των χρηστών (users) και ο κόμβος των συζητήσεων conversations και ο κόμβος των μηνυμάτων. Μπορούμε να ομοιάσουμε αυτές τις δομές με λίστες οι οποίες περιέχουν χρήστες και συνομιλίες αντίστοιχα.

Μια εγγραφή στην δομή των χρηστών έχει τις ιδιότητες (properties) `userId` και `userName` που είναι αλφαριθμητικά (strings). Το `userId` δημιουργείται αυτόματα όταν εγγράφεται ο χρήστης.

Μια εγγραφή στην δομή των συζητήσεων έχει τις παρακάτω ιδιότητες (properties) με τύπο δεδομένων αλφαριθμητικού (string):

- **userOne:** με το id του ενός συνομιλητή που συμμετέχει στην συνομιλία
- **userTwo:** με το id του δεύτερου συνομιλητή που συμμετέχει στην συνομιλία

Μια εγγραφή στην δομή των μηνυμάτων, που περιλαμβάνει τις εγγραφές των μηνυμάτων που ανταλλάσσουν οι χρήστες, έχει τις παρακάτω ιδιότητες:

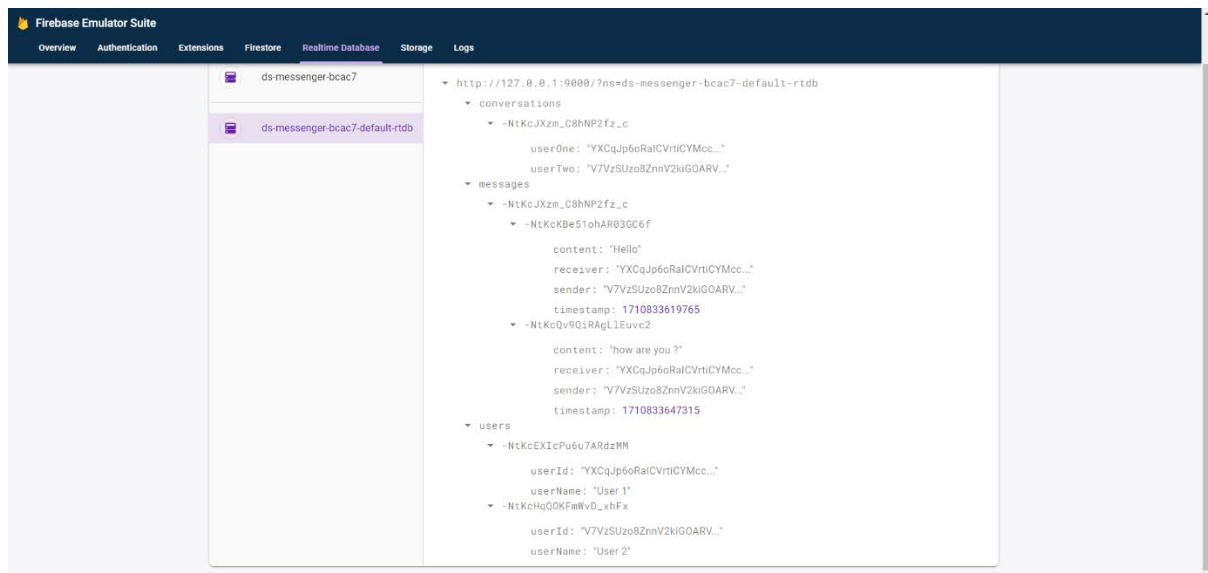
- **content:** με το περιεχόμενο του μηνύματος
- **receiver:** με το id του αποστολέα
- **sender:** με το id του παραλήπτη

- **timestamp:** με τον χρόνο και την ημερομηνία αποστολής του μηνύματος (εγγραφή στη βάση).

Στον κόμβο των μηνυμάτων που δημιουργείται, προστίθεται ένας κόμβος για την κάθε συνομιλία, ο οποίος εγγράφεται στην βάση με το unique id της συνομιλίας (conversation) με την οποία συσχετίζονται τα μηνύματα. Τα μηνύματα της συνομιλίας προστίθενται στον κόμβο αυτόν, όπως φαίνεται και στην παρακάτω εικόνα 4.3.2.

Αυτό έχει ως αποτέλεσμα την αποτελεσματική και γρήγορη ανάκτηση των μηνυμάτων για την αντίστοιχη συνομιλία.

Η δομή της βάσης και των κόμβων της είναι απλή και επίπεδη ενώ οι εγγραφές περιλαμβάνουν μόνο τα απαραίτητα που χρειάζεται η εφαρμογή μας. Η παρακάτω εικόνα (4.3.2 παρουσιάζει την δομή της βάσης μας στην κονσόλα διαχείρισης των emulators.



Εικόνα 4.3.2 Δομή Βάσης Δεδομένων

Στο Angular application δεν θα χρειαστεί να κάνουμε όλες τις απαραίτητες παραμετροποιήσεις σε θέμα Firebase. Οι περισσότερες έχουν γίνει αυτόματα όταν εκτελέσαμε τις απαραίτητες εντολές.

Ωστόσο θα πρέπει να συνδέσουμε την εφαρμογή μας με το Firebase για το production και να θέσουμε τα values σε κάποια properties του

TypeScript αρχείου περιβάλλοντος για την παραγωγή (production environment file) το οποίο θα συμπεριληφθεί στο τελικό «πακέτο διανομής» (distribution bundle), όταν χτίσουμε την εφαρμογή μας ώστε να την δημοσιεύσουμε (deploy) και να την φιλοξενήσουμε στον διακομιστή ιστού (web host), για να είναι προσβάσιμη από τους χρήστες μέσω του παγκόσμιου ιστού. Εκτενέστερη αναφορά για την διαδικασία αυτή θα πραγματοποιηθεί σε επόμενο κεφάλαιο.

Στο κεφάλαιο αυτό αναφερθήκαμε στην αρχιτεκτονική και τις πρακτικές που ακολουθήθηκαν για την ανάπτυξη του project. Επίσης, κάναμε εκτενέστερη αναφορά σε τμήματα του κώδικα που είχαν την μεγαλύτερη σημασία για τον αναγνώστη, ώστε να κατανοήσει καλύτερα την δομή και τον τρόπο ανάπτυξης της εφαρμογής. Στην συνέχεια, περιγράψαμε τα πρότυπα σχεδίασης της Angular που εφαρμόσαμε, την δημιουργία θέματος καθώς και τις βιβλιοθήκες που χρησιμοποιήσαμε. Τέλος αναφερθήκαμε και στις τεχνολογίες της Firebase που παρέχουν και αποτελούν την λογική της επιχείρησης και το επίπεδο δεδομένων της εφαρμογής μας και πως όλα αυτά συνδυάστηκαν και συντέλεσαν στην δημιουργία του τελικού προϊόντος. Στο επόμενο κεφάλαιο θα περιγράψουμε τις λειτουργίες της εφαρμογής και τις δυνατότητες που έχει ο χρήστης.

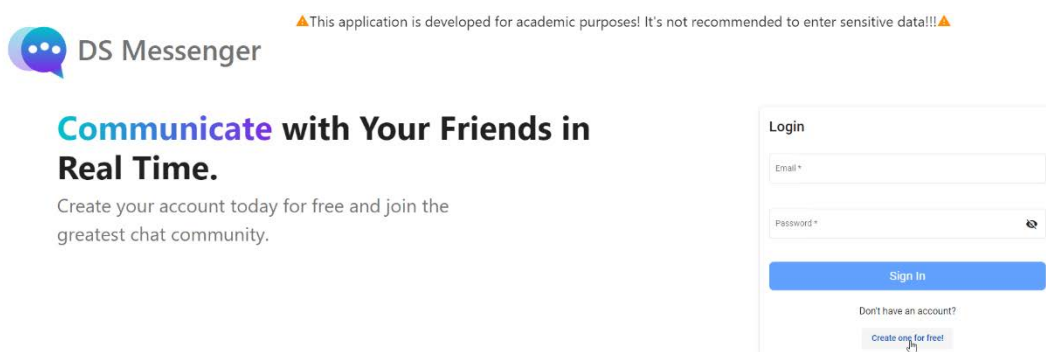
5 Λειτουργίες Εφαρμογής

Σε αυτήν την ενότητα, θα εξετάσουμε τις λειτουργίες και τις δυνατότητες που είναι ενσωματωμένες στην εφαρμογή ανταλλαγής μηνυμάτων. Καθώς η τεχνολογία προχωρά, η αποδοτικότητα και η αποτελεσματικότητα των εφαρμογών αυτών διαδραματίζουν κρίσιμο ρόλο στη διευκόλυνση των αλληλεπιδράσεων μεταξύ των χρηστών. Αυτό το κεφάλαιο έχει ως στόχο να παρουσιάσει τις βασικές λειτουργίες της εφαρμογής, τα χαρακτηριστικά της και τις δυνατότητες που έχει ο χρήστης.

5.1 Δυνατότητες Χρήστη

Αρχική Σελίδα – Landing Page

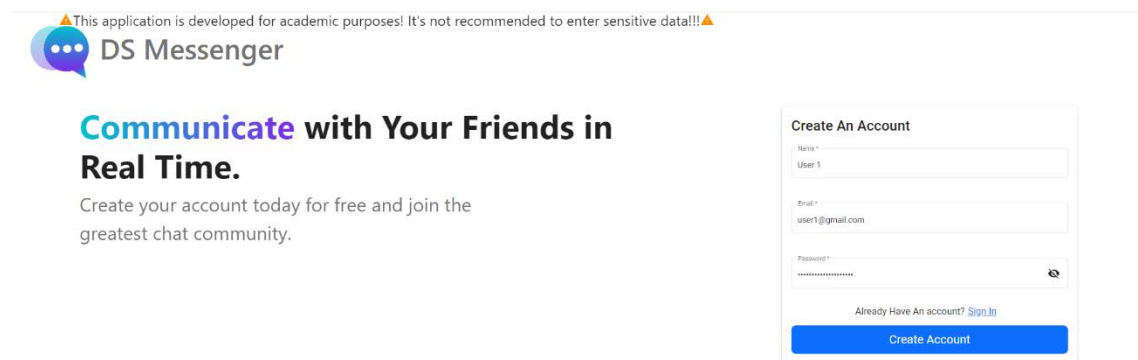
Όταν ένας εν δυνάμει χρήστης μεταβεί στην σελίδα μας από τον περιηγητή του, θα έρθει σε επαφή με την αρχική μας σελίδα, η οποία αποτελείται από αντιπροσωπευτικό κείμενο για το τι προσφέρει η εφαρμογή μας αλλά μια φόρμα σύνδεσης και δημιουργίας λογαριασμού χρήστη. Εφόσον ο χρήστης δεν είναι εγγεγραμμένος και επομένως δεν έχει λογαριασμό για να χρησιμοποιήσει την εφαρμογή μας, πρέπει να επιλέξει την επιλογή Create one for free, όπως φαίνεται και στην παρακάτω εικόνα 5.1.1.



Εικόνα 5.1.1 Αρχική Σελίδα

Εγγραφή Χρήστη

Αφού ο χρήστης κατευθυνθεί στην σχετική φόρμα θα πρέπει να συμπληρώσει τα απαραίτητα πεδία όπως το όνομα χρήστη, τη διεύθυνση ηλεκτρονικού ταχυδρομείου και τον κωδικό πρόσβασης που επιθυμεί, όπως φαίνεται και στην εικόνα 5.1.2. Η διαδικασία εγγραφής περιλαμβάνει μηχανισμούς επικύρωσης για τη διασφάλιση της αυθεντικότητας των πληροφοριών που παρέχονται από τον χρήστη και την αποτροπή τυχόν παραπλανητικών ενεργειών.

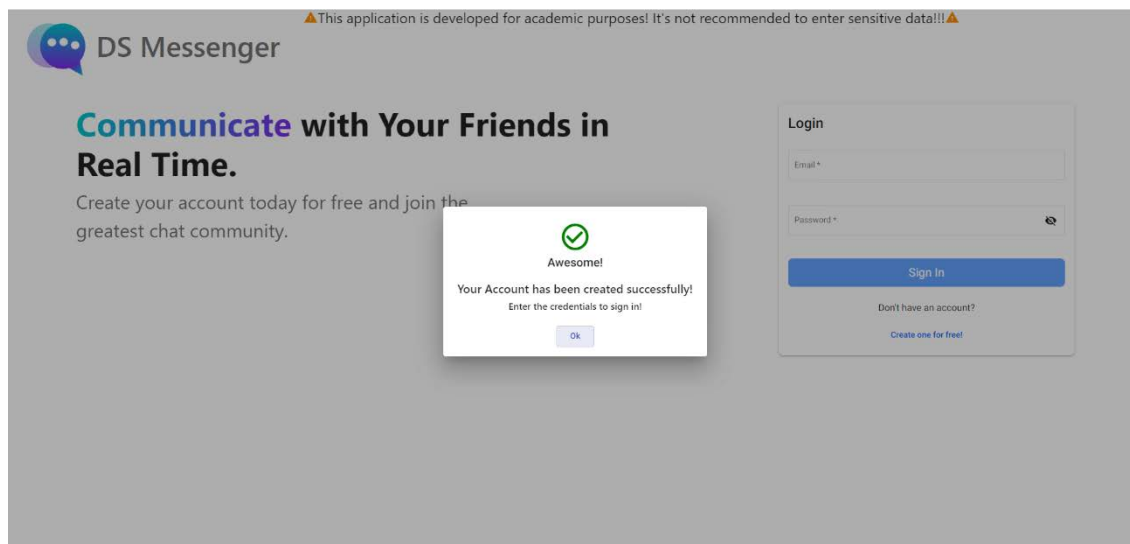


The screenshot shows the DS Messenger registration interface. At the top, a warning message states: "This application is developed for academic purposes! It's not recommended to enter sensitive data!!!". Below this is the DS Messenger logo and the text "Communicate with Your Friends in Real Time." followed by "Create your account today for free and join the greatest chat community." On the right, the "Create An Account" form is displayed. It includes input fields for "Name" (with "User 1" as a placeholder), "Email" (with "user1@gmail.com" as a placeholder), and "Password" (with a masked password and an eye icon to toggle visibility). Below the password field is a link "Already Have An account? Sign In" and a blue "Create Account" button.

Εικόνα 5.1.2 Δημιουργία Λογαριασμού

Σύνδεση

Μόλις οι χρήστες εγγραφούν επιτυχώς, θα εμφανιστεί ένα αναδυόμενο παράθυρο που θα ενημερώνει τον χρήστη για την επιτυχή δημιουργία λογαριασμού ενώ θα τον παροτρύνει να εισάγει τα στοιχεία του ώστε να συνδεθεί στην εφαρμογή, όπως φαίνεται και στην εικόνα 5.1.3.



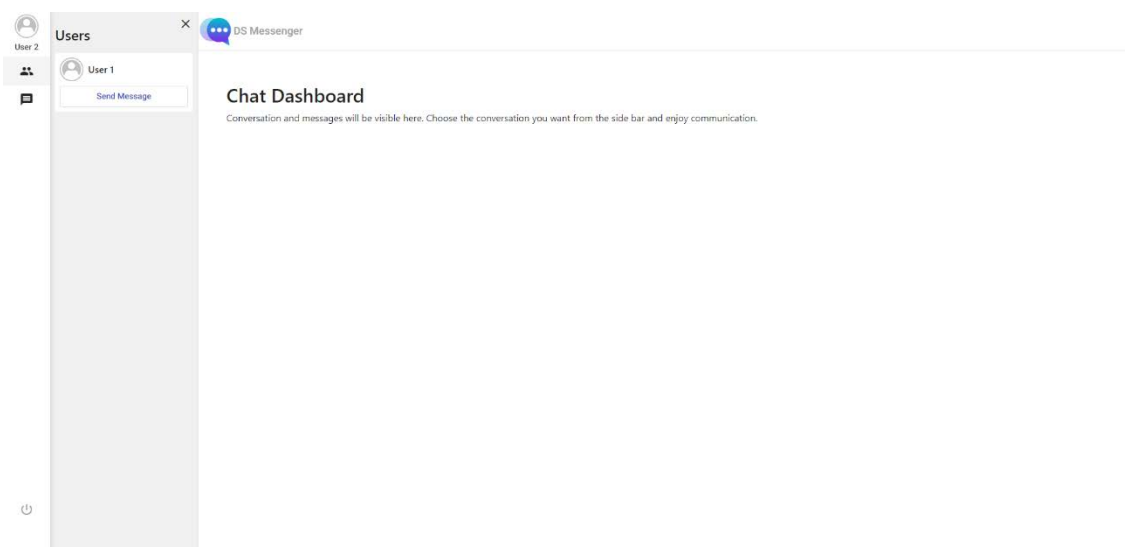
Εικόνα 5.1.3 Επιτυχής Δημιουργία Χρήστη

Η διαδικασία σύνδεσης περιλαμβάνει την υποβολή του ονόματος χρήστη και του κωδικού πρόσβασης που σχετίζονται με τον λογαριασμό του χρήστη. Στη συνέχεια, η εφαρμογή ταυτοποιεί τα στοιχεία συγκρίνοντάς τα με εκείνα που είναι αποθηκευμένα στην βάση.

Μόνο όταν τα διαπιστευτήρια ταιριάζουν με τα αποθηκευμένα δεδομένα, ο χρήστης αποκτά πλήρη πρόσβαση στις λειτουργίες που έχει να προσφέρει η εφαρμογή, σε διαφορετική περίπτωση εμφανίζονται στον χρήστη τα αντίστοιχα μηνύματα σφάλματος.

Πίνακας Συνομιλιών – Chat Board

Στην συνέχεια, ο χρήστης θα ανακατευθυνθεί στο chat board. Εκεί βρίσκεται πλευρικό πάνελ, στο οποίο ο χρήστης μπορεί να δει και να επιλέξει τους διαθέσιμους χρήστες ώστε να συνομιλήσει αλλά και να δει και να μεταβεί στις συνομιλίες, ώστε να διαβάσει τα μηνύματά του, όπως φαίνεται και στην εικόνα 5.1.4.



Εικόνα 5.1.4 Πίνακας Συνομιλιών και Λίστα Χρηστών

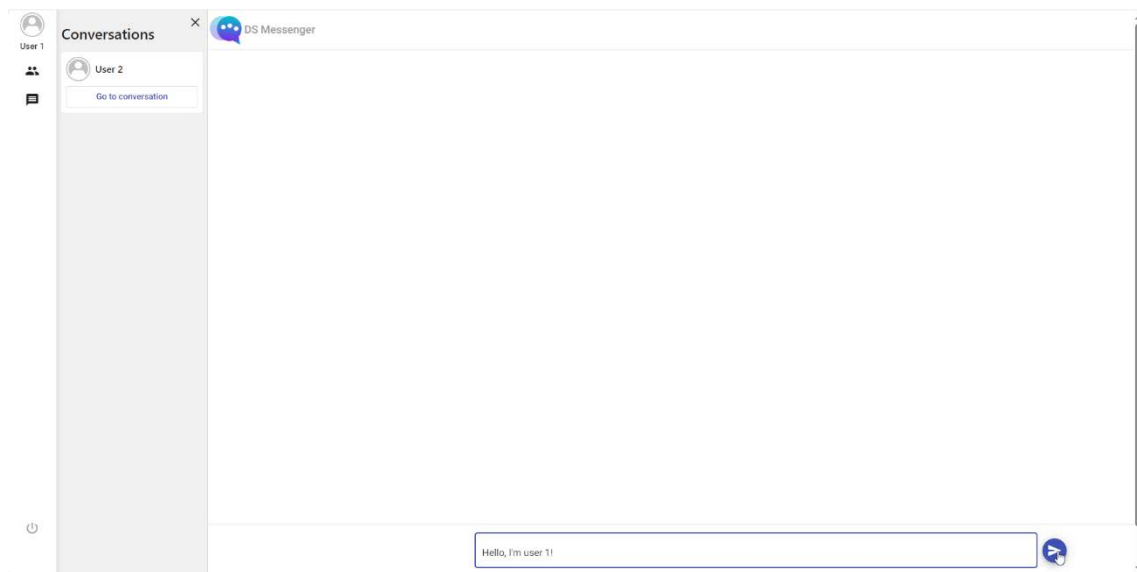
Ο χρήστης μπορεί να δημιουργήσει πολλά δωμάτια συνομιλίας με διαφορετικούς χρήστες της εφαρμογής

Οι χρήστες μπορούν να στέλνουν και να λαμβάνουν μηνύματα σε πραγματικό χρόνο, επιτρέποντας την άμεση επικοινωνία.

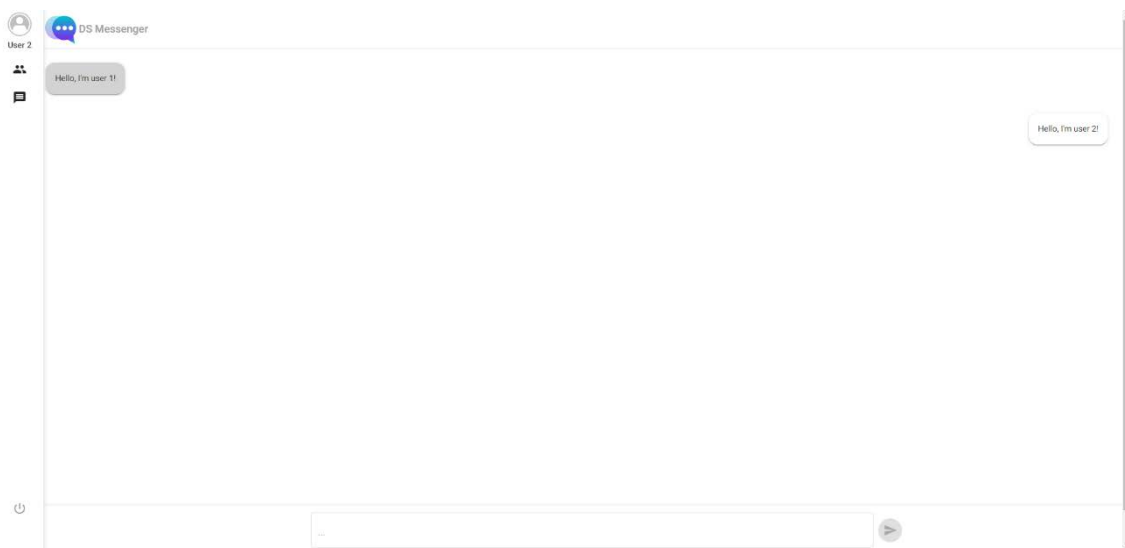
Στην σελίδα συνομιλίας ο χρήστης μπορεί να πληκτρολογήσει και να αποστείλει μηνύματα κειμένου αλλά και να δει το ιστορικό της συνομιλίας με τα μηνύματα και των δύο χρηστών.

Όταν ο χρήστης δεν έχει πρόσβαση στην εφαρμογή, είτε γιατί αποσυνδέθηκε είτε γιατί δεν έχει λογαριασμό, θα ανακατευθυνθεί πάντα στην αρχική σελίδα αυθεντικοποίησης.

Όταν ο χρήστης εισάγει έναν σύνδεσμο για την εφαρμογή μας που δεν υπάρχει θα οδηγηθεί σε σελίδα σφάλματος 404. Στις παρακάτω εικόνες αναπαρίσταται η ανταλλαγή ενός μηνύματος μεταξύ των χρηστών User 1 και User 2.

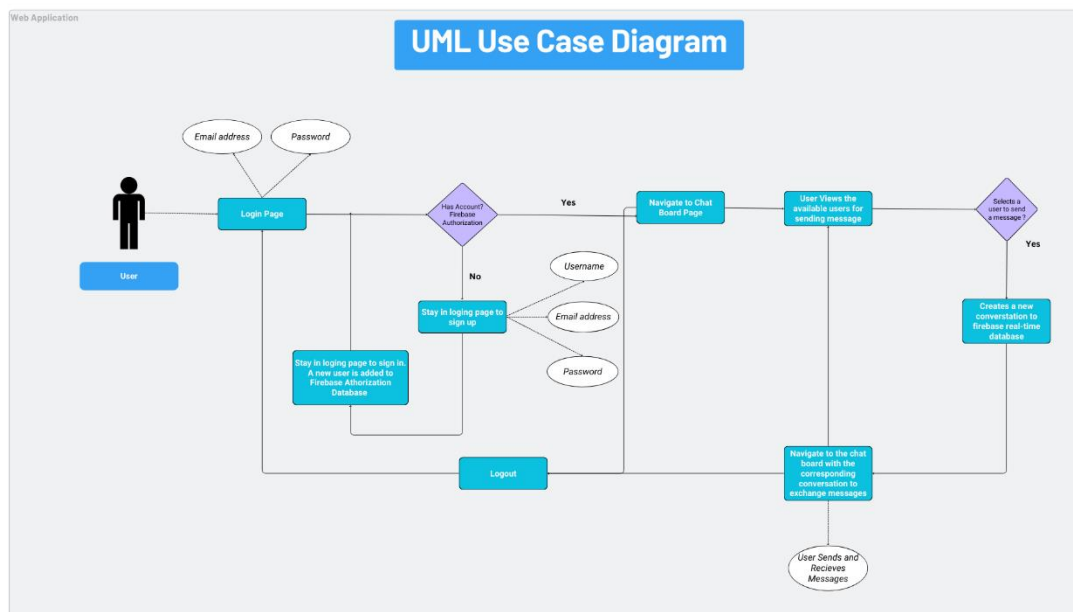


Εικόνα 5.1.5 Ο User 1 στέλνει μήνυμα



Εικόνα 5.1.6 Ο User 2 στέλνει μήνυμα

Στην συνέχεια, υπάρχει στην εικόνα 5.1.7 ένα UML διάγραμμα περιπτώσεων χρήσης που αναπαριστά τις περιπτώσεις χρήσης που είναι διαθέσιμες στην εφαρμογή μας .



Εικόνα 5.1.7 UML Διάγραμμα Χρήσης

5.2 Δημοσίευση Εφαρμογής

Σε αυτή την υποενότητα θα περιγράψουμε αναφορικά πως μπορούμε να δημοσιεύσουμε την εφαρμογή μας ώστε να είναι προσβάσιμη από τον παγκόσμιο ιστό. Όπως αναφέραμε σε προηγούμενο κεφάλαιο, η Firebase μας προσφέρει την υπηρεσία hosting. Το Firebase Hosting είναι μια υπηρεσία φιλοξενίας ιστοσελίδων που προσφέρεται από την Google στο πλαίσιο της πλατφόρμας Firebase. Παρέχει αρκετά πλεονεκτήματα και χαρακτηριστικά:

- **Δωρεάν SSL Πιστοποιητικά:** Το Firebase Hosting προσφέρει δωρεάν SSL πιστοποιητικά για όλες τις ιστοσελίδες που φιλοξενούνται σε αυτό, εξασφαλίζοντας έτσι ασφαλή σύνδεση μεταξύ του χρήστη και της ιστοσελίδας.
- **Γρήγορη Φόρτωση:** Η υπηρεσία διαθέτει παγκόσμιο δίκτυο παράδοσης περιεχομένου (CDN), που επιτρέπει τη γρήγορη φόρτωση των ιστοσελίδων από οπουδήποτε στον κόσμο.
- **Διαχείριση Έκδοσης:** Μπορούμε να δημοσιεύουμε εκδόσεις της εφαρμογής μας με ευκολία και να επιλέξουμε ποια έκδοση θα είναι διαθέσιμη για τους χρήστες μας.

- **Εύκολη Σύνδεση με το Domain:** Η Firebase παρέχει δύο δωρεάν domains για την εφαρμογή μας ωστόσο είναι εφικτό να χρησιμοποιήσουμε το δικό μας εφόσον έχουμε αγοράσει από τους διαθέσιμους παρόχους. Η σύνδεση του είναι εύκολη και γίνεται με λίγα κλικ μέσα από την console του Firebase .
- **Στατιστικά Κίνησης:** Μπορείτε να παρακολουθείτε την επίδοση της ιστοσελίδας σας μέσω των στατιστικών που παρέχονται από το Firebase Hosting.

Συνολικά, το Firebase Hosting προσφέρει μια αξιόπιστη, γρήγορη και εύκολη λύση φιλοξενίας για ιστοσελίδες και εφαρμογές, με ένα ευρύ φάσμα χρηστικών χαρακτηριστικών.

Για να δημιουργήσουμε το τελικό «πακέτο διανομής» (distribution bundle), θα πρέπει αρχικά να χτίσουμε το Angular project για την παραγωγή, εκτελώντας την παρακάτω εντολή:

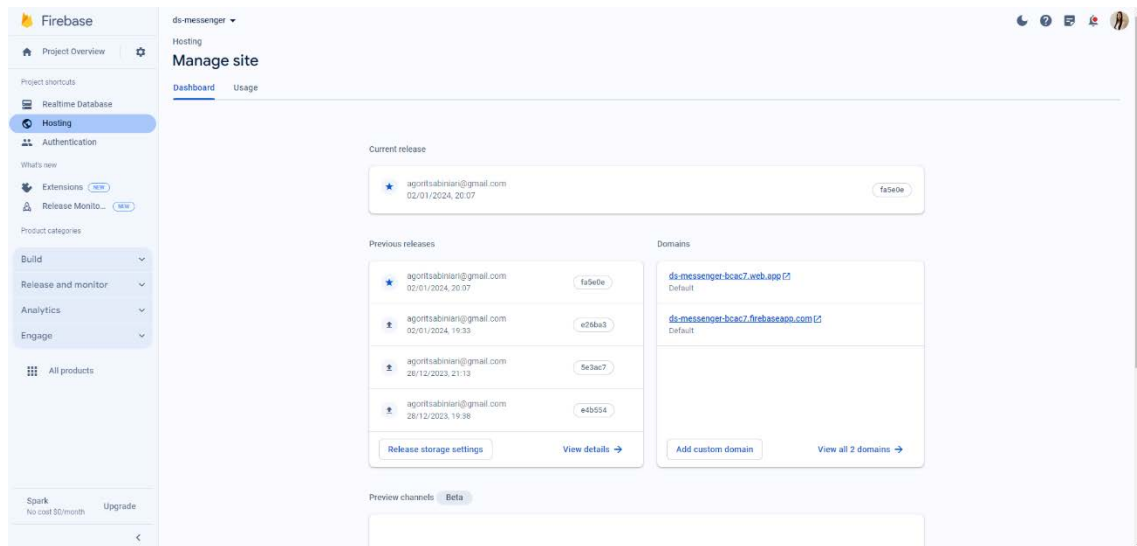
```
ng build --configuration production --aot
```

στην συνέχεια, αφού χτίσουμε την εφαρμογή μας για να την δημοσιεύσουμε (deploy) και να την φιλοξενήσουμε στον διακομιστή ιστού (web host), για να είναι προσβάσιμη από τους χρήστες μέσω του παγκόσμιου ιστού, θα πρέπει να εκτελέσουμε την εντολή:

```
firebase deploy
```

Μόλις ολοκληρωθεί η διαδικασία μπορούμε να μεταβούμε στην Firebase console, ώστε να δούμε τα deployments και releases, πληροφορίες για κίνηση δεδομένων κ.λ.π όπως φαίνεται και στην παρακάτω εικόνα 5.2.1. Η εφαρμογή μας είναι πλέον προσβάσιμη από τον παγκόσμιο ιστό με την χρήση των δύο domains:

- ds-messenger-bcac7.web.app
- ds-messenger-bcac7.firebaseio.com



Εικόνα 5.2.1 Firebase Console - Hosting

6 Συμπεράσματα

Στα πλαίσια της εργασίας αυτής, η εφαρμογή ανταλλαγής μηνυμάτων που κατασκευάστηκε προσφέρει μια σύγχρονη και ευχάριστη εμπειρία για τον χρήστη, καθώς καθιστά εύκολη και γρήγορη την συνομιλία δύο χρηστών με ασφάλεια.

Η εφαρμογή που δημιουργήθηκε ξεπερνά τα όρια της πτυχιακής εργασίας και του ακαδημαϊκού οικοσυστήματος καθώς προσφέρει ένα καταναλωτικό προϊόν που μπορεί να εξυπηρετήσει πραγματικούς καθημερινούς χρήστες. Για την ανάπτυξη της εφαρμογής έγινε ενδελεχής έρευνα στα εργαλεία που χρησιμοποιήθηκαν.

Αποκτήθηκαν νέες γνώσεις και εμπειρία, τόσο από την μελέτη για την απόκτηση της απαραίτητης θεωρητικής και τεχνικής γνώσης, αλλά και από την επίλυση των προβλημάτων και των προκλήσεων.

Αξίζει να σημειωθεί πως οι τεχνολογίες που χρησιμοποιήθηκαν χρησιμοποιούνται για την κατασκευή επαγγελματικών web εφαρμογών στην σημερινή κοινωνία οι οποίες είναι η ισχύουσα σύγχρονη τάση, καθώς αναπτύσσονται και χρησιμοποιούνται ευρέως, ενώ έχουν εφαρμογή σε όλους τους τομείς και κλάδους.

Εν κατακλείδι, η παρούσα εφαρμογή είναι επεκτάσιμη καθώς είναι απλή, κατανοητή, και διέπεται από βέλτιστες σχεδιαστικές και αρχιτεκτονικές πρακτικές, αποτελώντας έτσι μια στέρεα βάση για μελλοντικές επεκτάσεις και προσθήκη λειτουργιών.

Βιβλιογραφία

- [1] <https://www.wikipedia.org>
- [2] <https://firebase.google.com/docs>
- [3] <https://angular-gr.web.app>
- [4] <https://material.angular.io>
- [5] Καλογρίδου-Κολυβά, Μαργαρίτα, *“Οι πολλές όψεις του διαδικτύου, χρήση, κατάχρηση, εθισμός, αντίλογος, διάλογος*, Αθήνα: Δρόμων, 2011

Παράρτημα

Κεφ.	Περιεχόμενο
3.2.1 Angular Framework	Ένα framework όπως η Angular είναι ένα εργαλείο ανάπτυξης λογισμικού που παρέχει μια δομή και μια σειρά από λειτουργίες προκειμένου να διευκολύνει τον προγραμματιστή στη δημιουργία εφαρμογών. Συγκεκριμένα, η Angular είναι ένα πλαίσιο εργασίας για τη δημιουργία ενιαίων, δυναμικών ιστοσελίδων ή εφαρμογών.
3.2.1 TypeScript	Η TypeScript είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού που παρέχει ένα υποσύνολο της σύνταξης της JavaScript, επεκτείνοντας τη με στοιχεία που βοηθούν στην ανάπτυξη μεγάλων εφαρμογών.
3.2.1 Node.js και npm	Το Node.js είναι μια ανοιχτού κώδικα πλατφόρμα που χρησιμοποιείται για την ανάπτυξη διαδικτυακών εφαρμογών. Βασίζεται στην JavaScript και εκτελείται στην πλευρά του διακομιστή, δηλαδή στον server. Όταν εγκαθιστούμε το Node.js στον υπολογιστή μας, το npm (Node Package Manager) περιλαμβάνεται αυτόματα ως μέρος της εγκατάστασης. Το npm είναι το εργαλείο διαχείρισης πακέτων του Node.js και χρησιμοποιείται για να εγκαταστήσουμε, να ενημερώσουμε και να διαχειριστούμε πακέτα (προγράμματα, βιβλιοθήκες, εργαλεία κ.λπ.) στο περιβάλλον του Node.js. Για την εγκατάσταση του Node.js μπορείτε να μεταβείτε στην σελίδα: https://nodejs.org/en/download
3.2.1 C.L.I.	Ένα CLI (Command Line Interface) είναι ένα πρόγραμμα που εκτελείται από το τερματικό ή τη γραμμή εντολών του υπολογιστή και χρησιμοποιείται για τη διαχείριση διαφόρων εργασιών και λειτουργιών στο σύστημα ή σε μια εφαρμογή. Πιο συγκεκριμένα, το Angular CLI είναι ένα CLI που παρέχεται από την ομάδα ανάπτυξης του Angular, που παρέχει ένα σύνολο εντολών για τη διευκόλυνση της δημιουργίας, της διαχείρισης και της ανάπτυξης εφαρμογών Angular.
3.2.1 @angular/fire	Το @angular/fire είναι ένα επίσημο εργαλείο που παρέχεται από την ομάδα ανάπτυξης της Angular σε συνεργασία με την ομάδα ανάπτυξης του Firebase. Χρησιμοποιείται για την εύκολη ενσωμάτωση και αλληλεπίδραση με τις υπηρεσίες του Firebase σε ένα έργο Angular. Για περισσότερες πληροφορίες μπορείτε να μεταβείτε στην σελίδα: https://github.com/angular/angularfire