International Burch University

# IT 309 SOFTWARE ENGINEERING

PROJECT DOCUMENTATION

# ChatApp

Prepared by:

**Student 1**: Almir Alic
**Student 2**: Hadzic Mirsad

Proposed to:
**Nermina Durmić, Assist. Prof. Dr.**
**Aldin Kovačević, Teaching Assistant**

Date of submission: 12.05.2023

## TABLE OF CONTENTS

## Introduction

This document serves as the project documentation for the ChatApp developed by Almir Alic and Hadzic Mirsad. The ChatApp is a global and private chat application that allows users to connect with each other and communicate in real-time. The application was developed with the aim of creating a platform where people from different parts of the world can come together and exchange ideas, thoughts, and experiences.

## About the project

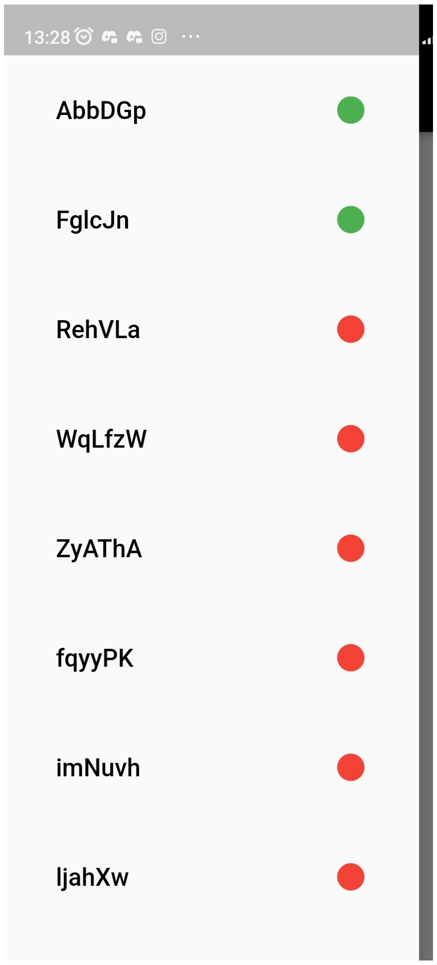
The ChatApp is available as .apk release at the GitHub release version ChatApp v1.0.0. (Link: Release Link).

The ChatApp provides the following main features:

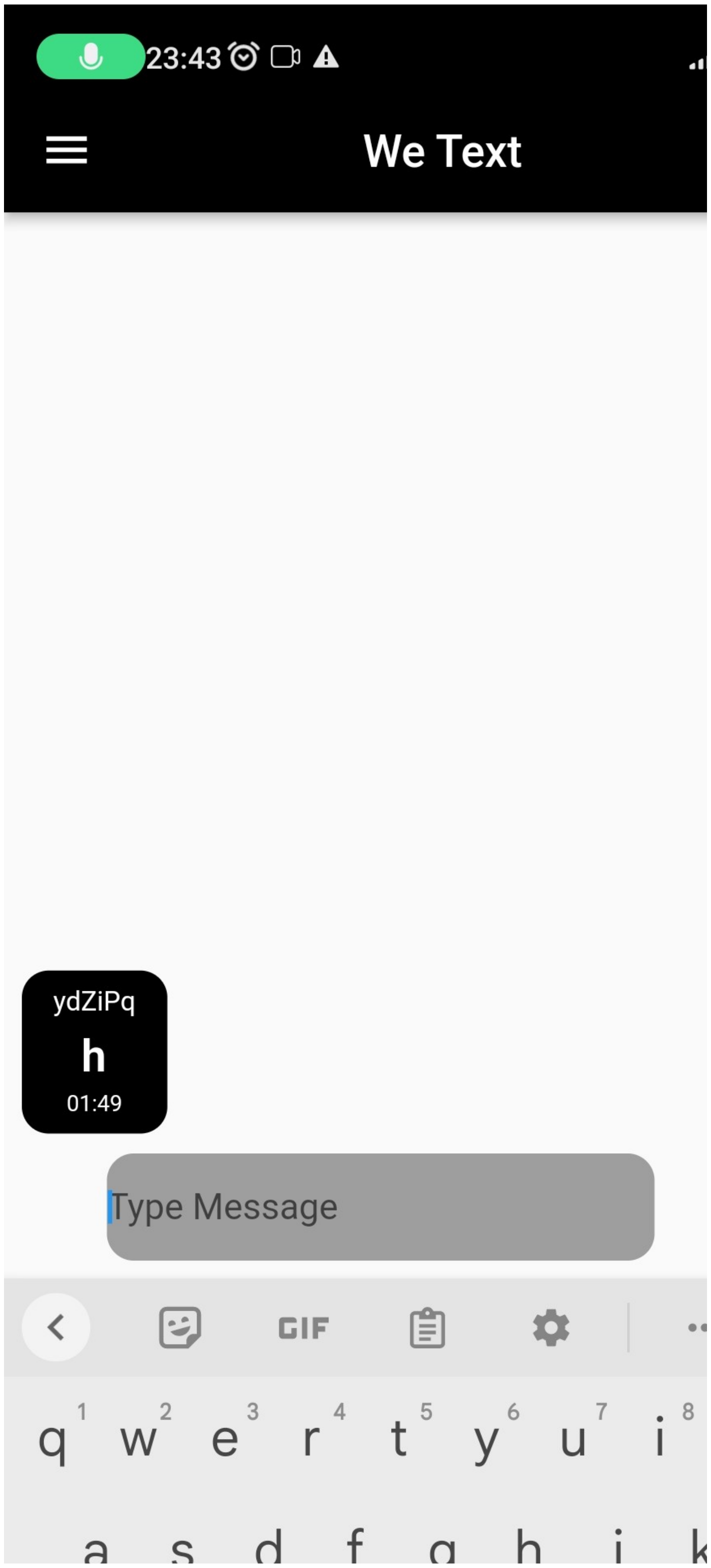
- Adding User: Allows users to create an account and add themselves to the user database.
- Private and Group Chat: Enables users to send and receive private messages as well as join global chat rooms.
- Real-time Messaging: Implements real-time messaging using Firebase to ensure instant message delivery.
- Online/Offline Status: Displays the online/offline status of users to indicate their availability for chat.

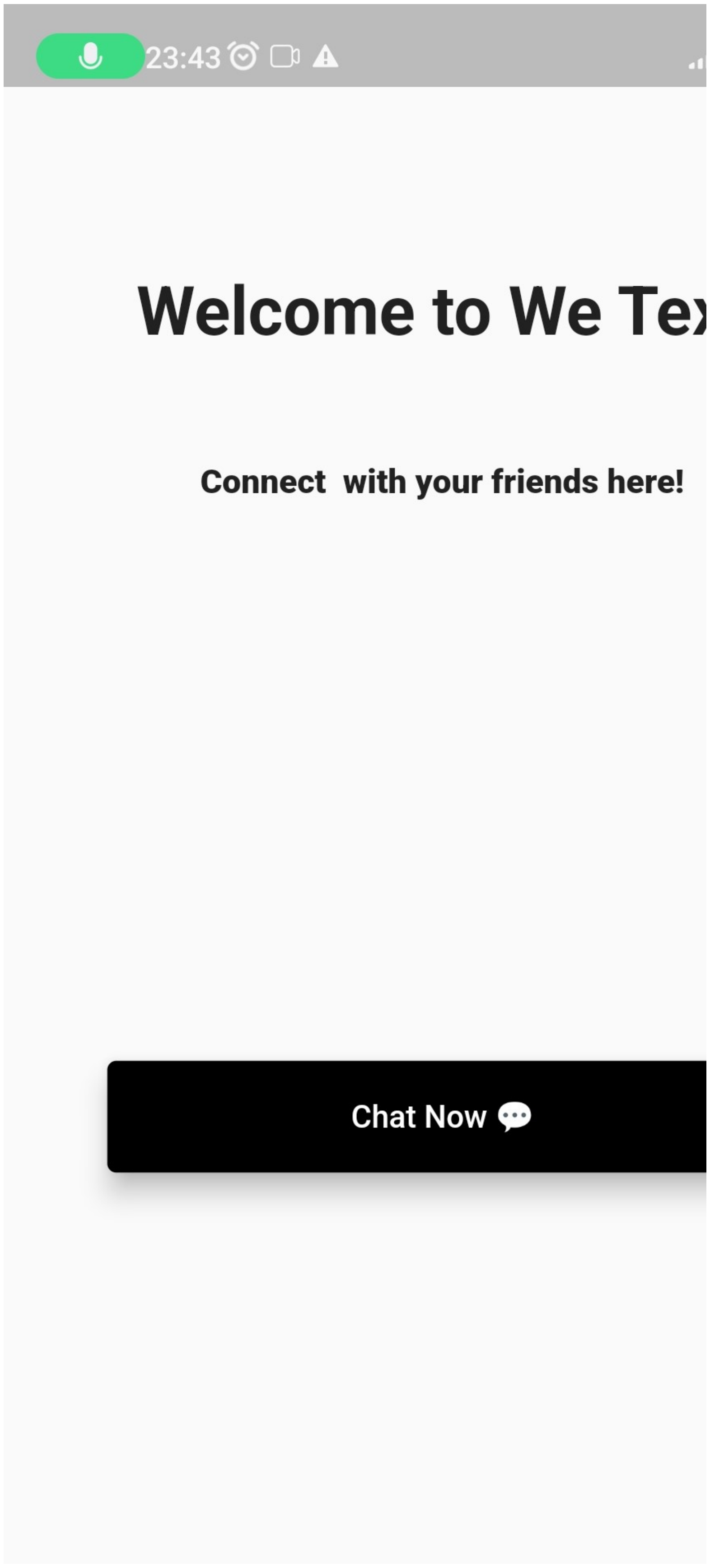
- Push Notifications: Sends push notifications to users for new private messages or when someone joins the ChatApp.
- Emoji Support: Allows users to send emojis in their chats.

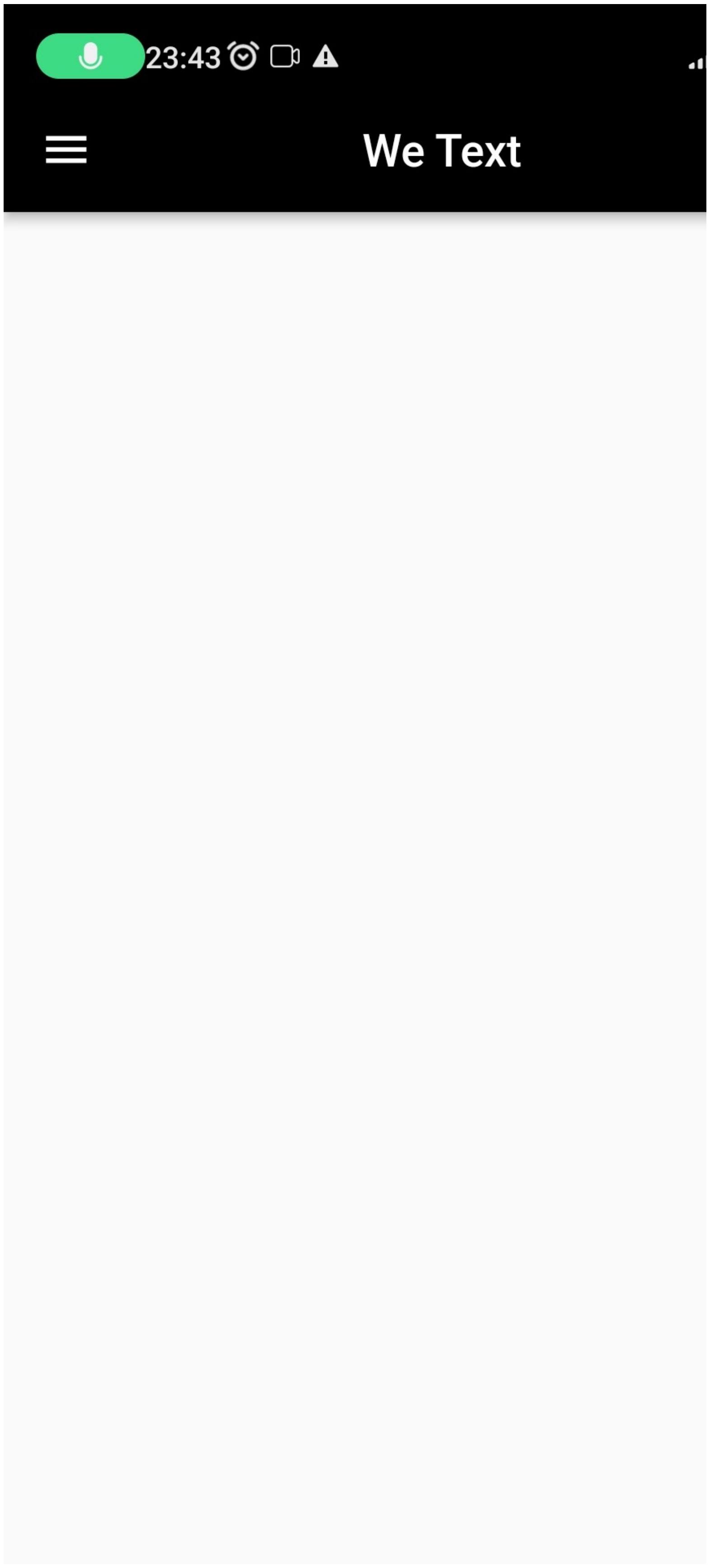
## Project Functionalities and Screenshots

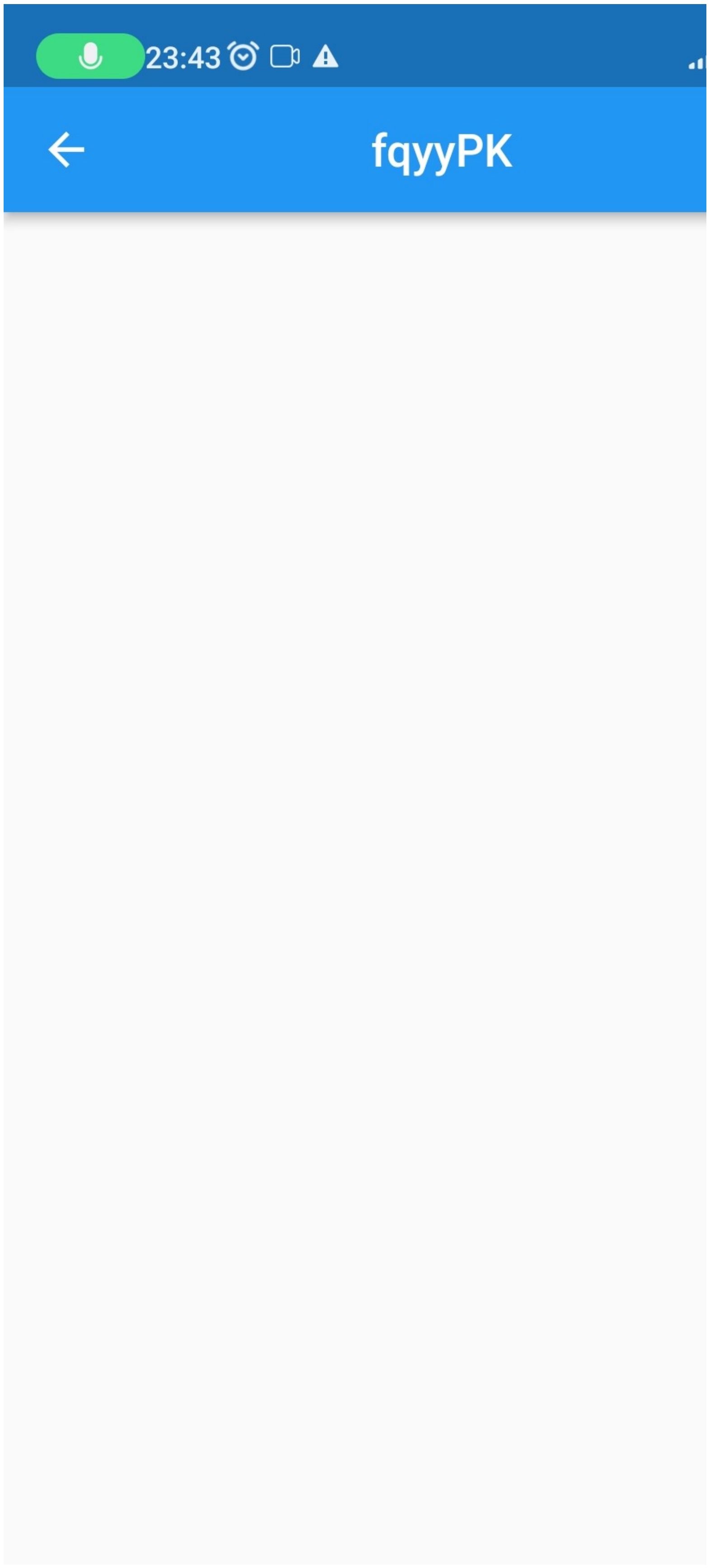


| | |
|---|---|
| AbbDGp | 🟢 |
| FglcJn | 🟢 |
| RehVLa | 🔴 |
| WqLfzW | 🔴 |
| ZyAThA | 🔴 |
| fqyyPK | 🔴 |
| imNuvh | 🔴 |
| ljahXw | 🔴 |

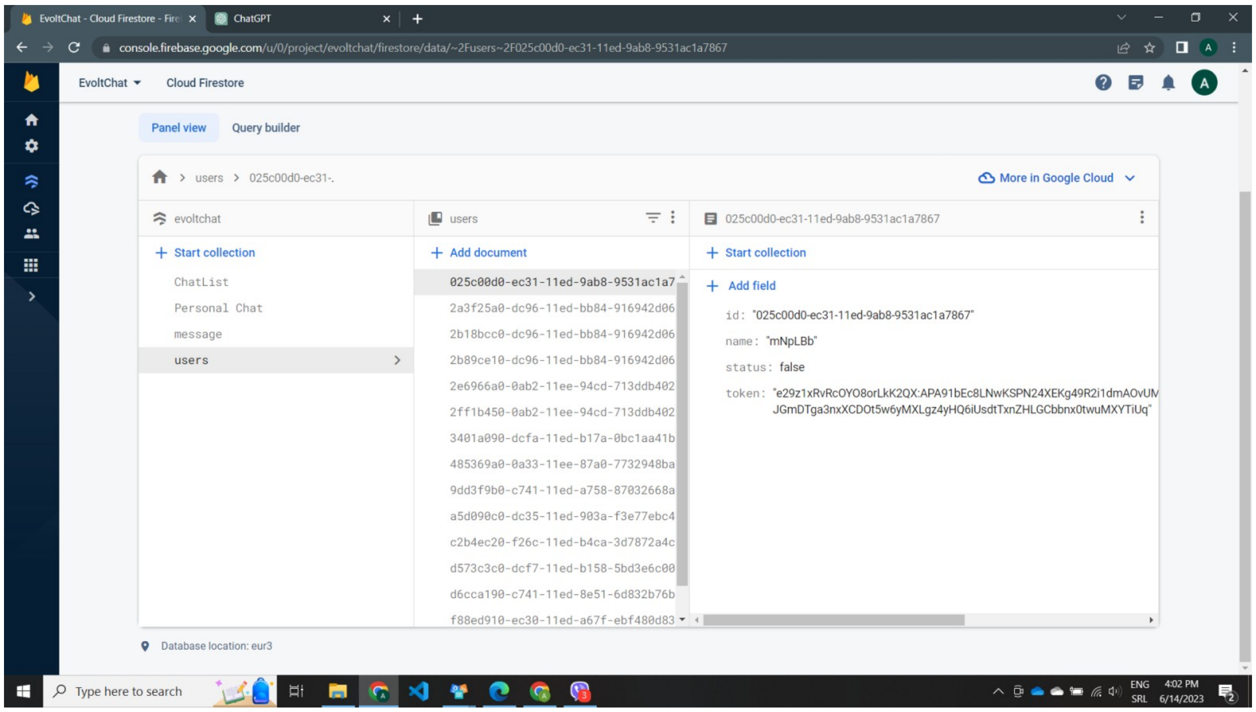

23:43
We Text
ydZiPq
h
01:49
Type Message
GIF
q w e r t y u i
a s d f g h j


23:43
Welcome to We Tex
Connect  with your friends here!
Chat Now 💬


23:43
We Text


23:43
fqyyPK

# Firebase Screenshots



console.firebase.google.com/u/0/project/evoltchat/firestore/data/~2Fusers~2F025c00d0-ec31-11ed-9ab8-9531ac1a7867

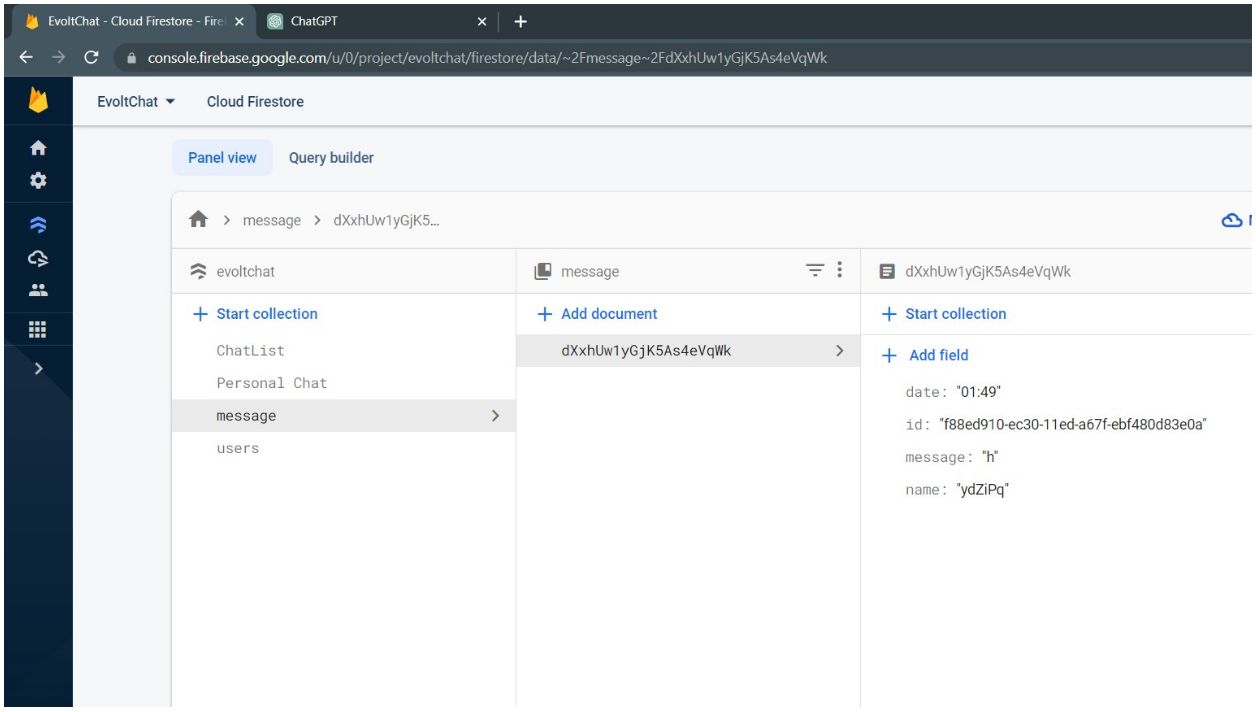
EvoltChat · Cloud Firestore

Panel view · Query builder

users > 025c00d0-ec31-.

| evoltchat | users | 025c00d0-ec31-11ed-9ab8-9531ac1a7867 |
|---|---|---|
| + Start collection | + Add document | + Start collection |
| ChatList | 025c00d0-ec31-11ed-9ab8-9531ac1a7 | + Add field |
| Personal Chat | 2a3f25a0-dc96-11ed-bb84-916942d06 | id: "025c00d0-ec31-11ed-9ab8-9531ac1a7867" |
| message | 2b18bcc0-dc96-11ed-bb84-916942d06 | name: "mNpLBb" |
| users | 2b89ce10-dc96-11ed-bb84-916942d06 | status: false |
| | 2e6966a0-0ab2-11ee-94cd-713ddb402 | token: "e29z1xRvRcOYO8orLkK2QX:APA91bEc8LNwKSPN24XEKg49R2i1dmAOvUM JGmDTga3nxXCDOt5w6yMXLgz4yHQ6iUsdtTxnZHLGCbbnx0twuMXYTiUq" |
| | 2ff1b450-0ab2-11ee-94cd-713ddb402 | |
| | 3401a090-dcfa-11ed-b17a-0bc1aa41b | |
| | 485369a0-0a33-11ee-87a0-7732948ba | |
| | 9dd3f9b0-c741-11ed-a758-87032668a | |
| | a5d090c0-dc35-11ed-903a-f3e77ebc4 | |
| | c2b4ec20-f26c-11ed-b4ca-3d7872a4c | |
| | d573c3c0-dcf7-11ed-b158-5bd3e6c00 | |
| | d6cca190-c741-11ed-8e51-6d832b76b | |
| | f88ed910-ec30-11ed-a67f-ebf480d83 | |

Database location: eur3



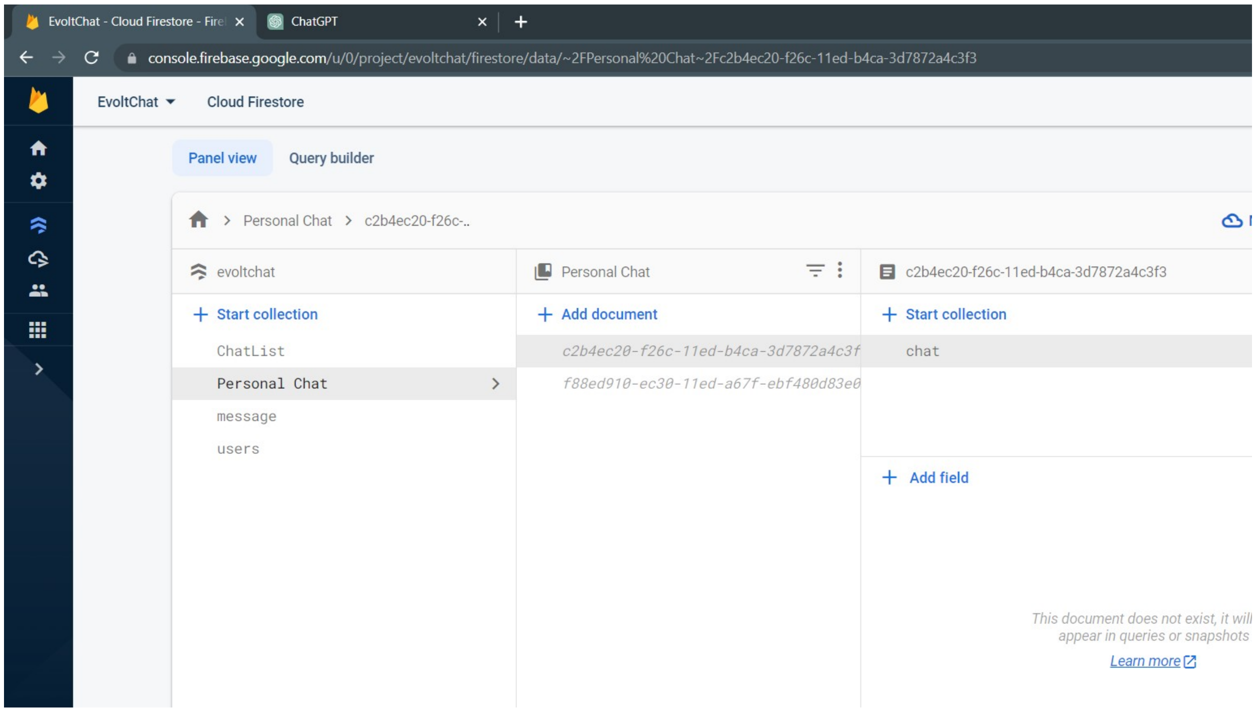
console.firebase.google.com/u/0/project/evoltchat/firestore/data/~2Fmessage~2FdXxhUw1yGjK5As4eVqWk

EvoltChat · Cloud Firestore

Panel view · Query builder

message > dXxhUw1yGjK5...

| evoltchat | message | dXxhUw1yGjK5As4eVqWk |
|---|---|---|
| + Start collection | + Add document | + Start collection |
| ChatList | dXxhUw1yGjK5As4eVqWk | + Add field |
| Personal Chat | | date: "01:49" |
| message | | id: "f88ed910-ec30-11ed-a67f-ebf480d83e0a" |
| users | | message: "h" |
| | | name: "ydZiPq" |


EvoltChat - Cloud Firestore - Fire | ChatGPT

console.firebase.google.com/u/0/project/evoltchat/firestore/data/~2FPersonal%20Chat~2Fc2b4ec20-f26c-11ed-b4ca-3d7872a4c3f3

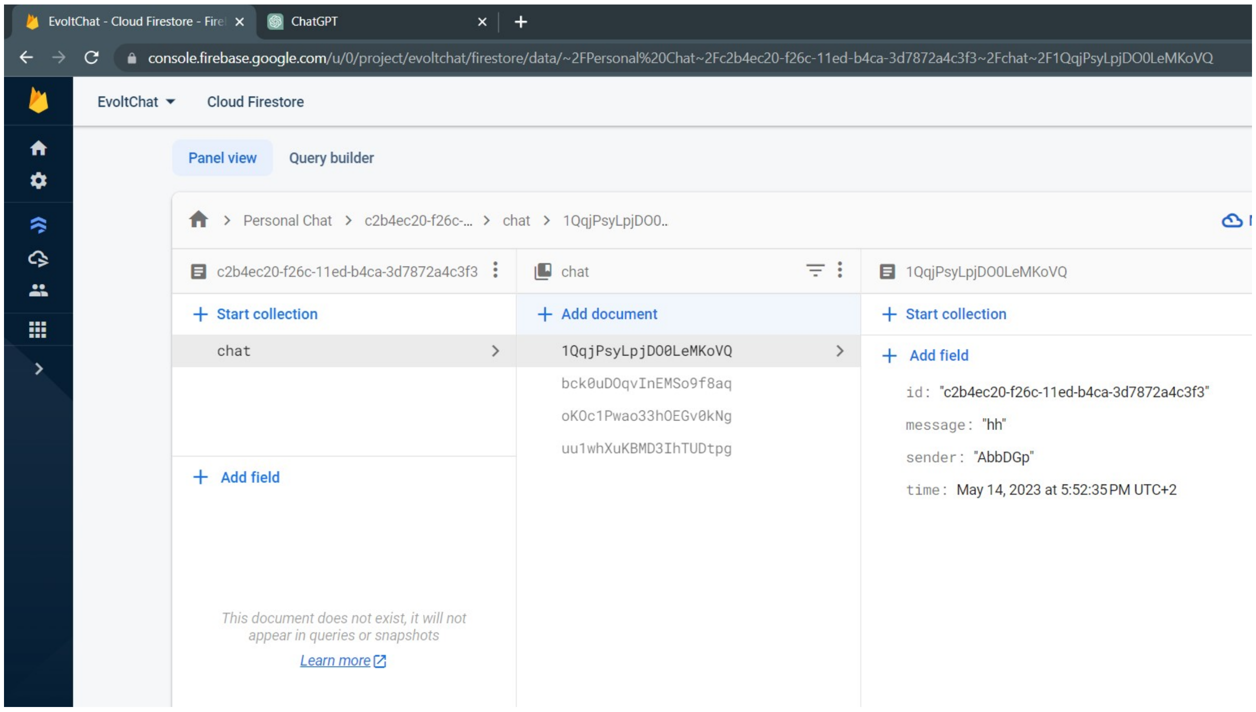
EvoltChat | Cloud Firestore

Panel view | Query builder

Personal Chat > c2b4ec20-f26c-...

| evoltchat | Personal Chat | c2b4ec20-f26c-11ed-b4ca-3d7872a4c3f3 |
|---|---|---|
| + Start collection | + Add document | + Start collection |
| ChatList | c2b4ec20-f26c-11ed-b4ca-3d7872a4c3f | chat |
| Personal Chat | f88ed910-ec30-11ed-a67f-ebf480d83e0 | |
| message | | + Add field |
| users | | |

This document does not exist, it will not appear in queries or snapshots
Learn more


EvoltChat - Cloud Firestore - Fire | ChatGPT

console.firebase.google.com/u/0/project/evoltchat/firestore/data/~2FPersonal%20Chat~2Fc2b4ec20-f26c-11ed-b4ca-3d7872a4c3f3~2Fchat~2F1QqjPsyLpjDO0LeMKoVQ

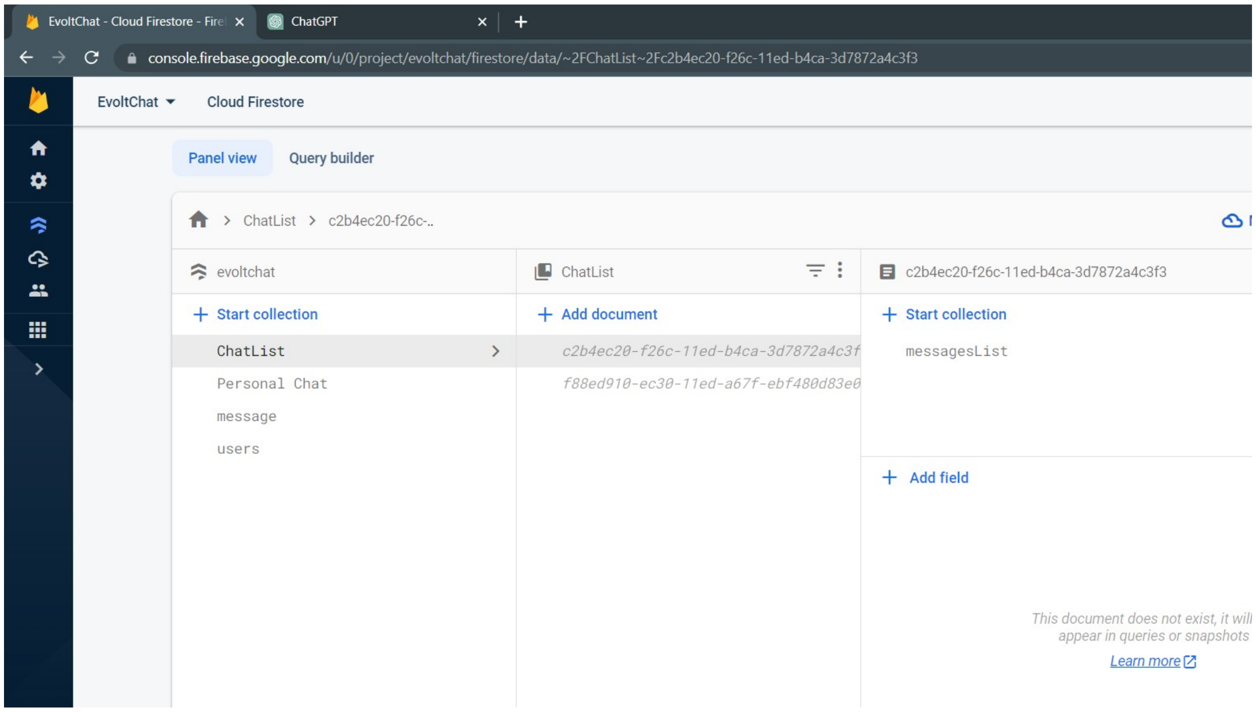
EvoltChat | Cloud Firestore

Panel view | Query builder

Personal Chat > c2b4ec20-f26c-... > chat > 1QqjPsyLpjDO0..

| c2b4ec20-f26c-11ed-b4ca-3d7872a4c3f3 | chat | 1QqjPsyLpjDO0LeMKoVQ |
|---|---|---|
| + Start collection | + Add document | + Start collection |
| chat | 1QqjPsyLpjDO0LeMKoVQ | + Add field |
| | bck0uDOqvInEMSo9f8aq | id: "c2b4ec20-f26c-11ed-b4ca-3d7872a4c3f3" |
| | oKOc1Pwao33hOEGv0kNg | message: "hh" |
| | uu1whXuKBMD3IhTUDtpg | sender: "AbbDGp" |
| + Add field | | time: May 14, 2023 at 5:52:35 PM UTC+2 |

This document does not exist, it will not appear in queries or snapshots
Learn more

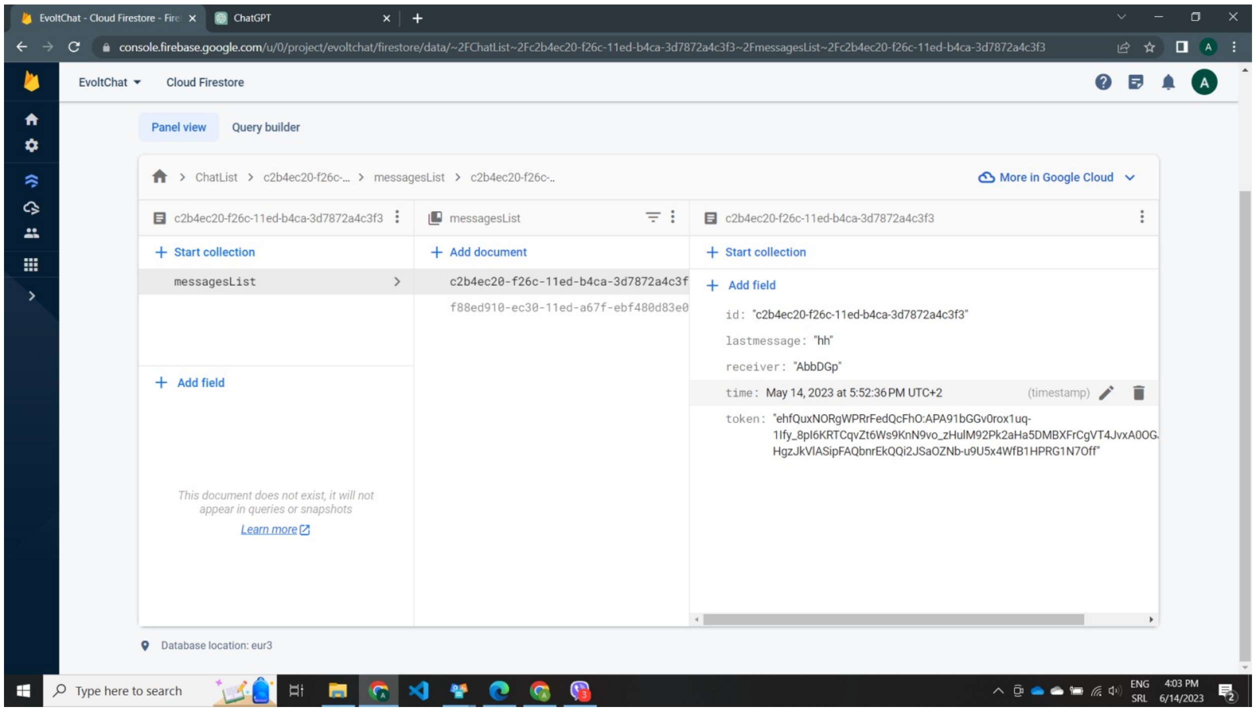

console.firebase.google.com/u/0/project/evoltchat/firestore/data/~2FChatList~2Fc2b4ec20-f26c-11ed-b4ca-3d7872a4c3f3

EvoltChat › Cloud Firestore

ChatList › c2b4ec20-f26c-..

| evoltchat | ChatList | c2b4ec20-f26c-11ed-b4ca-3d7872a4c3f3 |
|---|---|---|
| + Start collection | + Add document | + Start collection |
| ChatList | c2b4ec20-f26c-11ed-b4ca-3d7872a4c3f | messagesList |
| Personal Chat | f88ed910-ec30-11ed-a67f-ebf480d83e0 | |
| message | | + Add field |
| users | | |

This document does not exist, it will not appear in queries or snapshots
Learn more


console.firebase.google.com/u/0/project/evoltchat/firestore/data/~2FChatList~2Fc2b4ec20-f26c-11ed-b4ca-3d7872a4c3f3~2FmessagesList~2Fc2b4ec20-f26c-11ed-b4ca-3d7872a4c3f3

EvoltChat › Cloud Firestore

ChatList › c2b4ec20-f26c-... › messagesList › c2b4ec20-f26c-..

| c2b4ec20-f26c-11ed-b4ca-3d7872a4c3f3 | messagesList | c2b4ec20-f26c-11ed-b4ca-3d7872a4c3f3 |
|---|---|---|
| + Start collection | + Add document | + Start collection |
| messagesList | c2b4ec20-f26c-11ed-b4ca-3d7872a4c3f | + Add field |
| | f88ed910-ec30-11ed-a67f-ebf480d83e0 | |

```
id: "c2b4ec20-f26c-11ed-b4ca-3d7872a4c3f3"
lastmessage: "hh"
receiver: "AbbDGp"
time: May 14, 2023 at 5:52:36 PM UTC+2    (timestamp)
token: "ehfQuxNORgWPRrFedQcFhO:APA91bGGv0rox1uq-
       1Ify_8pI6KRTCqvZt6Ws9KnN9vo_zHulM92Pk2aHa5DMBXFrCgVT4JvxA0OG
       HgzJkVIASipFAQbnrEkQQi2JSaOZNb-u9U5x4WfB1HPRG1N7Off"
```

This document does not exist, it will not appear in queries or snapshots
Learn more

Database location: eur3

## Project Structure

## Technologies

The technologies used in the ChatApp project include:

- Dart: A client-optimized programming language used for mobile, web, and desktop application development.
- Flutter: An open-source UI software development kit (SDK) by Google for building natively compiled applications for mobile, web, and desktop.
- Firebase: A mobile and web application development platform by Google that provides services such as authentication, real-time databases, cloud storage, and messaging.

The coding standards followed in the project are as follows:

- Dart: The coding standard used is the Dart Style Guide, which provides guidelines for code layout, naming, documentation, and more. The guide can be found on the official Dart website.
- Flutter: The coding standard is based on the Dart Style Guide, along with Flutter-specific conventions and guidelines documented in the Flutter Style Guide, available on the official Flutter website.

## Database Entities

The ChatApp project utilizes Firebase Firestore as the database. The following entities/tables are present in the database:

- Users: Stores user information and credentials.
- Messages: Stores chat messages exchanged between users.
- Personal Chat: Stores personal chat conversations between two users.
- ChatList: Stores the list of chats for each user.

## Architectural Pattern

MVC (Model-View-Controller) Architecture: This architectural pattern was chosen for its ability to separate the application's concerns and improve code organization.

## Design Patterns

In our project, we utilized the following design patterns:

- Stateful Widget Pattern (UI Design Pattern): We employed the Stateful Widget pattern in Flutter, which is a variation of the classic State pattern used in object-oriented programming.

- Singleton Pattern (Firebase Messaging Instance): To establish efficient communication between our app and Firebase, we implemented the Singleton pattern for the Firebase messaging instance.

  These design patterns played a crucial role in structuring and organizing our codebase, providing us with flexibility, reusability, and maintainability throughout the development process.

Stateful Widget Pattern:

- Location: lib/presentation/chat_list.dart
- Description: The Stateful Widget pattern is used when you need to change the state of a widget dynamically. It creates a StatefulWidget that creates a corresponding State object which is responsible for holding the state of the widget. The state object is then used to rebuild the widget tree as the state changes.

  In this particular code, the UsersList widget is a StatefulWidget that creates an instance of _UsersListState, which holds the state of the widget. The state is then updated through the getname() method, which retrieves the user name from shared preferences, and through the StreamBuilder widget, which retrieves the list of chats from Firebase. The state is finally used to rebuild the widget tree in the build() method.

Singleton Pattern:

- Location: lib/firebase/chat.dart
- Description: Line of code where this pattern is in use is 36.

  ```
  String? sendertoken = await firebaseMessaging.getToken();
  ```

  The firebaseMessaging class follows the Singleton pattern by ensuring that only one instance of the class is created and shared across the application.

## Tests

In our project, we conducted various tests to ensure the quality and reliability of our application. The following types of tests were implemented:

- Unit Tests: We are excited to announce our latest release, which focuses on enhancing our unit testing framework. In accordance with project requirements, we have added three separate unit test files to verify specific functionalities of our application. These tests include:

1. Elevated Button Test: This test validates the presence and behavior of an elevated button in our user interface. By simulating user interactions and examining the resulting state, we ensure the button functions as intended.
2. Welcome Text Test: To guarantee the proper display of a welcome text, we created a unit test that verifies the correct rendering and positioning of this important component.
3. Connect Text Test: Similarly, we implemented a unit test to confirm the accurate display of a connect text, ensuring that it appears correctly and provides the expected information to the users.

By organizing our tests into separate files, we improved maintainability and allowed for targeted testing. This approach facilitated the identification and resolution of issues specific to each functionality, resulting in a more robust and reliable application.

## How does it work?

Step 1: Install the .apk version here.

Step 2: Enter the App.

Step 3: Click on Chat Now.

Step 4: Random username is displayed on the top right part. To chat, click on the burger menu in the top left corner.

Step 5: Choose an Online user.

Step 6: Start chatting!

Step 7: Click on the group icon. Global chat is displayed. Chat!

Step 8: Click on Chat icon.

Step 9: Private chats are displayed. Chat!

## Conclusion

In conclusion, we are satisfied with the overall implementation of the ChatApp project. We believe that we have successfully achieved the main objectives of creating a global and private chat application. The project allowed us to gain valuable experience in developing real-time communication features using Flutter and Firebase.

Throughout the development process, we encountered some challenges, such as managing real-time updates and optimizing the user experience. However, through thorough research and collaboration, we were able to overcome these challenges and deliver a functional and user-friendly chat application.

In future iterations of the project, we aim to further enhance the user interface, introduce additional features, and optimize the performance of the application. We also plan to conduct more comprehensive testing to ensure the stability and reliability of the ChatApp.

Overall, the ChatApp project has been an excellent opportunity for us to apply software engineering principles and gain practical experience in developing a real-world application. We look forward to further improving and expanding the ChatApp in the future.