

# **MENTORSYNTH HUB**

## **INTERDISCIPLINARY PROJECT**

Submitted in partial fulfillment of the requirements for the award of  
Bachelor of Technology degree in Information Technology  
by

**KAMILATH RIFKA S(41120084)**



**DEPARTMENT OF INFORMATION TECHNOLOGY**  
**SCHOOL OF COMPUTING**

# **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY**  
**(DEEMED TO BE UNIVERSITY)**

**Category-1 University by UGC**

**Accredited with Grade “A++” by NAAC**

**12B Status by UGC |Approved by AICTE**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

**MAY 2024**



# **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)**

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

[www.sathyabama.ac.in](http://www.sathyabama.ac.in)

## **DEPARTMENT OF INFORMATION TECHNOLOGY**

### **BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of **KAMILATH RIFKA S(41120084)** who carried out the project entitled "**MentorSynth Hub**" under my supervision from **January 2024** to **May 2024**.

**Internal Guide**

**Dr.C.GEETHA, M.Tech., Ph.D.,**

**Head of the Department**

**Dr.R.SUBHASHINI, M.E., Ph.D.,**

Submitted for Viva voce Examination held on 8/5/2024

  
**Internal Examiner**  
**External Examiner**

## DECLARATION

I KAMILATH RIFKA S hereby declare that the Project Report entitled “MentorSynth Hub” done by me under the guidance of Dr.C.GEETHA, M.Tech., Ph.D., is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE: 8.5.2024

PLACE: CHENNAI

  
SIGNATURE OF THE CANDIDATE

## ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr.T.Sasikala, M.E., Ph.D., Dean**, School of Computing **Dr.R.Subhashini, M.E., Ph.D., Head** of the Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.C.Geetha, M.Tech., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

# CERTIFICATE OF COMPLETION



HACKWIT TECHNOLOGIES PVT LTD  
Redefine by Technologies

149, 1C, 2nd Floor, Mount Poonamallee Rd,  
Ramapuram, Chennai - 600089.  
admin@hackwittechnologies.com  
www.hackwittechnologies.com

10.02.2024

## CERTIFICATE FOR INTERNSHIP

TO WHOMSOEVER IT MAY CONCERN

This is to Certify that **Kamilath Rifka S (Reg No: 41120084)** student of Sathyabama Institute of Science and Technology, B. Tech Information Technology, has successfully completed the Internship at Hackwit Technologies from 22<sup>nd</sup> January to 30<sup>th</sup> January 2024.

During the internship, **Kamilath Rifka S** has closely worked as a part of the **FLUTTER** by carrying out the project entitled "**MentorSynth Hub**".

She demonstrated good developing and learning skill with self-motivated attitude to learn new things. Her contribution to implement the FLUTTER was good. We wish her all the best for her future endeavors.

Best Regards,

Ravikumar S  
CEO & Managing Director  
Hackwit Technologies Pvt Ltd



## CERTIFICATE OF COMPLETION



## **ABSTRACT**

MentorSynth Hub revolutionizes the traditional mentorship and learning experience by providing a comprehensive platform that addresses both academic and professional aspects. The app facilitates seamless interaction between students and mentors, fostering a community-driven approach to knowledge sharing. The platform transcends the conventional boundaries of academic Q&A, extending its capabilities to encompass job-related inquiries, interview preparation, and discussions on industry-specific challenges. Students benefit from MentorSynth Hub by gaining access to a diverse pool of experienced mentors who can provide guidance on academic subjects, career choices, and practical insights into the professional world. On the mentor side, professionals can leverage MentorSynth Hub to share their expertise, participate in discussions, and contribute to the development of aspiring talents. The platform serves as a space for mentors to engage with students, offer career advice, and even participate in hackathons, creating a dynamic ecosystem that bridges the gap between academia and real-world scenarios. MentorSynth Hub is not just a Q&A platform; it's a dynamic hub where users can actively engage in discussions, share knowledge, and collectively contribute to the growth of the community. The app's inclusive nature allows anyone to provide valuable answers, creating a collaborative learning environment that transcends geographical boundaries. By integrating mentorship with technology, MentorSynth Hub becomes a catalyst for skill, professional growth, and the exchange of innovative ideas.

## TABLE OF CONTENTS

| Chapter No. | TITLE                                 | Page No. |
|-------------|---------------------------------------|----------|
|             | ABSTRACT                              | vii      |
|             | LIST OF ABBREVIATIONS                 | x        |
|             | LIST OF FIGURES                       | xi       |
| 1           | INTRODUCTION                          | 1        |
|             | 1.1 Flutter                           | 2        |
|             | 1.2 Dart                              | 3        |
|             | 1.3 Firebase                          | 4        |
|             | 1.4 Problem Statement                 | 5        |
|             | 1.5 Objectives                        | 7        |
| 2           | METHODOLOGY                           | 9        |
|             | 2.1 Initial Project Requirements      | 9        |
|             | 2.2 Proposed Application Architecture | 12       |
|             | 2.3 Data Flow Diagram                 | 16       |
|             | 2.4 Modules                           | 17       |
|             | 2.4.1 User Management                 | 17       |
|             | 2.4.2 Authentication Management       | 18       |
|             | 2.4.3 Profile Management              | 19       |
|             | 2.4.4 Mentor Selection                | 20       |
|             | 2.4.5 Messaging and Collaboration     | 21       |
|             | 2.5 Development and Deployment Setup  | 22       |

|          |                                   |           |
|----------|-----------------------------------|-----------|
| <b>3</b> | <b>RESULTS AND DISCUSSION</b>     | <b>26</b> |
| <b>4</b> | <b>CONCLUSION AND FUTURE WORK</b> | <b>32</b> |
| 4.1      | Conclusion                        | 32        |
| 4.2      | Future Work                       | 33        |
|          | <b>REFERENCES</b>                 | <b>35</b> |
|          | <b>APPENDICES</b>                 | <b>37</b> |
| A.       | Sample Code                       | 37        |
| B.       | Screenshots                       | 39        |

## LIST OF ABBREVIATIONS

| ABBREVIATION | EXPANSION                            |
|--------------|--------------------------------------|
| UI           | - User Interface                     |
| UX           | - User Experience                    |
| iOS          | - iPhone Operating System            |
| IDE          | - Integrated Development Environment |

## LIST OF FIGURES

| <b>Figure No.</b> | <b>Name of the Figure</b>         | <b>Page No.</b> |
|-------------------|-----------------------------------|-----------------|
| 2.1               | Flutter Architecture              | 12              |
| 2.2               | Firebase Architecture             | 14              |
| 2.3               | Proposed Application Architecture | 14              |
| 2.4               | Data Flow Diagram                 | 16              |
| 2.5               | Project Creation                  | 22              |
| 2.6               | Firebase Connection               | 23              |
| 2.7               | Firebase Authentication           | 24              |
| 2.8               | Flutter Run                       | 24              |
| 3.1               | Login Page Interface              | 26              |
| 3.2               | Dashboard Screen Display          | 27              |
| 3.3               | Mentor Selection Interface        | 28              |
| 3.4               | Feedback Submission Screen        | 29              |
| 3.5               | Firebase Configurations           | 30              |
| 3.6               | Firebase Initialization           | 31              |

# **CHAPTER 1**

## **INTRODUCTION**

MentorSynth Hub stands as an innovative mobile application that redefines the landscape of collaborative learning and mentorship. Rooted in the principles of accessibility and inclusivity, the app serves as a dynamic platform connecting students and mentors across diverse educational and professional domains. One of the app's core strengths lies in its holistic learning approach. It seamlessly integrates academic support with professional development, recognizing the symbiotic relationship between education and employment. This unique blend positions MentorSynth Hub as a versatile tool catering to users at different stages of their learning and career journeys.

MentorSynth Hub addresses a spectrum of user queries ranging from academic doubts and subject-related discussions to job-related inquiries. Users can seek guidance on intricate academic subjects or navigate through the complexities of interview preparation and industry-specific challenges. This versatility makes the app a go-to resource for a myriad of informational needs. At the heart of MentorSynth Hub is the dynamic nexus between mentors and students. The app strives to break down barriers, allowing learners to connect with mentors seamlessly.

Leveraging cutting-edge technology, MentorSynth Hub incorporates user-friendly features and an intuitive interface. The technological backbone of the app ensures a seamless and enriching user experience, empowering individuals to navigate the complexities of both academia and the professional realm.

## **1.1.FLUTTER**

Flutter, a robust cross-platform UI toolkit developed by Google, is at the heart of MentorSynth Hub, shaping the application's foundation and user interface. The utilization of Flutter brings a paradigm shift to the traditional app development approach, offering a single codebase that transcends platform-specific limitations. This innovation allows MentorSynth Hub to deliver a visually captivating and responsive user interface, adeptly adjusting to both iOS and Android environments.

The efficiency of Flutter lies in its ability to streamline the development process, significantly reducing the complexities associated with managing separate codebases for different platforms. Developers benefit from a cohesive workflow, resulting in accelerated development cycles and enhanced productivity. Moreover, Flutter's hot reload feature enables real-time code changes and instant visual feedback, facilitating quick iterations and adjustments.

One of the key advantages of Flutter is its capability to ensure a consistent and high-quality user experience across diverse devices. By leveraging Flutter's rich set of customizable widgets, MentorSynth Hub achieves a cohesive design language and smooth interactions, irrespective of the underlying operating system. Users can seamlessly transition between devices without compromising on the application's performance or aesthetics.

## **1.2.DART**

### **i)Unified Codebase**

MentorSynth Hub relies on Dart, a client-optimized programming language, for implementing backend logic and server-side operations. Dart's versatility enables developers to maintain a unified codebase for both the frontend (Flutter) and backend, fostering consistency and ease of collaboration.

### **ii)Asynchronous Programming**

Dart's support for asynchronous programming is crucial for handling concurrent operations efficiently. This feature ensures that MentorSynth Hub can seamlessly manage multiple tasks, such as handling user requests, database operations, and external API calls, without causing delays or disruptions in the application's responsiveness.

### **iii)Scalability**

Dart's scalability is instrumental in accommodating the potential growth of MentorSynth Hub. As the user base expands and the demand for mentorship increases, Dart's ability to scale allows the platform to handle higher traffic, ensuring a smooth and responsive experience for all users.

### **iv)Strong Typing System**

Dart's strong typing system enhances code reliability and maintainability in MentorSynth Hub. By catching potential errors during the development phase, Dart enables developers to write robust and error-resistant code, reducing the likelihood of runtime issues and enhancing the overall stability of the application.

## **1.3.FIREBASE**

### **i)Real -time Data Synchronization**

Firebase Cloud Firestore supports real-time data synchronization, allowing seamless updates across multiple devices. This feature is crucial for MentorSynth Hub, enabling mentors and students to experience instant changes in discussions, mentorship requests, and collaborative interactions.

### **ii)Firebase Authentication for User Security**

Firebase Authentication provides a secure and straightforward user authentication process. MentorSynth Hub leverages this feature to ensure that only authorized users, both mentors and students, can access the platform. Multi-factor authentication and identity verification contribute to a robust security layer.

### **iii)Serverless Cloud Functions**

Firebase offers serverless cloud functions that enable the execution of backend logic without managing servers. MentorSynth Hub utilizes these functions for various tasks, such as sending notifications, updating user data, and handling asynchronous operations. This serverless architecture enhances scalability and efficiency.

### **iv)Firebase Cloud Storage for Media Assets**

Storing media assets, such as profile pictures and discussion attachments, is streamlined with Firebase Cloud Storage. MentorSynth Hub utilizes this feature to efficiently manage and serve multimedia content, ensuring a smooth and responsive user experience for both mentors and students.

#### **1.4.PROBLEM STATEMENT**

In the current educational landscape, students often encounter challenges that extend beyond the traditional realms of academia. The existing platforms for learning and mentorship are often fragmented, lacking a comprehensive solution that addresses both academic queries and professional development. As students navigate through their educational journey, they are confronted with intricate academic subjects, job-related uncertainties, and a need for mentorship that spans beyond conventional boundaries.

In contemporary education, students grapple with a myriad of challenges ranging from intricate academic concepts to the complexities of transitioning into the professional realm. The existing educational landscape lacks a unified platform that effectively addresses the holistic needs of students and mentors, leaving critical gaps in academic support and career guidance. The absence of an integrated solution for both academic and professional development results in a fragmented learning experience, hindering the seamless synthesis of knowledge and skills.

Furthermore, the lack of inclusivity in existing platforms restricts the diversity of perspectives and experiences available to users. Many students, despite having valuable insights to share, find themselves on the periphery of collaborative learning environments. The current platforms also often fail to foster inclusivity, sidelining valuable contributors such as students with unique perspectives and experiences. This exclusionary nature stifles the collaborative potential of learning environments and limits the diverse range of insights available to users.

One of the primary challenges is the fragmentation of available resources, with separate platforms addressing academic queries, job-related challenges, and mentorship programs. This disjointed approach not only complicates the user experience but also fails to provide a holistic solution for individuals seeking guidance across various facets of their educational and professional journey.

Moreover, the existing platforms often lack inclusivity, leading to a limited diversity of perspectives and experiences. Many students, despite possessing valuable insights and experiences, find themselves on the fringes of collaborative learning environments. This lack of inclusivity hampers the potential for a vibrant exchange of ideas and knowledge, hindering the growth and development of both mentors and mentees.

MentorSynth Hub seeks to address these challenges by offering an all-encompassing platform that transcends the limitations of current educational and mentorship solutions. The problem at hand is the absence of a versatile and inclusive ecosystem that unifies academic support, professional mentorship, and collaborative learning, creating a transformative space where students and mentors can engage, learn, and thrive. MentorSynth Hub aims to revolutionize the educational experience by providing an integrated platform that fosters holistic development, promotes inclusivity, and empowers individuals to navigate the complexities of academia and professional growth with confidence.

## **1.5. OBJECTIVES**

The main objective of MentorSynth Hub is to establish a versatile and inclusive platform that seamlessly integrates academic support, professional mentorship, and collaborative learning. The platform aims to bridge the gap between traditional educational resources and the dynamic requirements of students, providing a comprehensive solution for both academic and professional development. MentorSynth Hub seeks to create a vibrant and diverse community where students and mentors can engage in meaningful interactions, share insights, and collectively contribute to each other's growth. The platform aspires to foster an environment that empowers individuals to navigate the complexities of academia and the professional world with confidence, ensuring holistic development.

### ***Key Challenges***

#### **i)Integration of Diverse Features**

Harmonizing academic support and professional mentorship functionalities within a single application. Ensuring a seamless and intuitive user experience while navigating through varied features.

#### **ii)Inclusive Platform Design**

Designing a user interface that accommodates a broad user base, considering users with varying levels of technological proficiency. Balancing inclusivity without compromising simplicity and clarity in the app's design.

### **iii)Community Engagement**

Encouraging active participation from both students and mentors in discussions, answering queries, and contributing to the community. Implementing effective community-building features to foster a sense of belonging and collaborative learning.

### **iv)Security and Privacy**

Ensuring robust security measures to protect user data, particularly in the context of mentorship interactions.

Adhering to privacy regulations and standards to build trust among users regarding the confidentiality of their information.

### **v)Scalability and Performance**

Designing the app architecture to accommodate potential growth in user base and content without compromising performance. Optimizing the performance of MentorSynth Hub to deliver a responsive and efficient experience, especially as the user community expands.

### **vi)Flutter Codebase Maintenance**

Managing the complexity of a Flutter-based codebase to ensure code readability, maintainability, and scalability. Keeping the codebase up-to-date with the latest Flutter releases and best practices to leverage new features and improvements.

## **CHAPTER 2**

### **METHODOLOGY**

#### **2.1. INITIAL PROJECT REQUIREMENTS**

The initial project requirements for MentorSynth Hub encompass various aspects such as functionalities, technologies, and packages.

##### **i)Project Scope**

- 1) Develop a mobile application named "MentorSynth Hub".
- 2) Create a platform that connects students seeking mentorship with experienced professionals.
- 3) Address both academic and job-related queries within the platform.

##### **ii)Technologies**

- 1) Flutter: Utilize Flutter for cross-platform mobile app development, ensuring a consistent user experience on both iOS and Android.
- 2) Dart: Employ the Dart programming language as the primary language for Flutter app development.
- 3) Firebase: Integrate Firebase for backend services, including authentication, real-time database (Firestore), and cloud storage.

### **iii)Functionalities**

- 1) User Authentication: Implement a secure user authentication system for both mentors and students.
- 2) Question and Answer System: Enable users to post questions and receive answers from mentors or other students.
- 3) Job-Related Queries: Provide a dedicated section for job-related queries, interview preparation, and industry-specific challenges.
- 4) Mentor-Student Matching: Implement a system to match students with mentors based on expertise and interests.
- 5) Real-time Chat: Include a real-time chat feature for direct communication between mentors and students.

### **iv)Packages Used**

i)get

Version: 4.6.6

Purpose: State management and navigation framework.

ii)firebase\_core

Version: 2.25.3

Purpose: Firebase initialization and core functionality.

iii)firebase\_auth

Version: 4.17.3

Purpose: User authentication with Firebase.

iv)cloud\_firestore

Version: 4.15.3

Purpose: Interaction with Cloud Firestore for real-time data.

v)cloud\_firestore\_web

Version: 3.10.3

Purpose: Web implementation of Cloud Firestore.

vi)firebase\_auth\_web

Version: 5.9.3

Purpose: Web implementation of Firebase Authentication.

vii)firebase\_core\_web

Version: 2.11.3

Purpose: Web implementation of Firebase Core.

viii)google\_sign\_in

Version: 5.2.1

Purpose: Google Sign-In authentication.

ix)toast

Version: 0.3.0

Purpose: Displaying simple toast notifications.

x)fluttertoast

Version: 8.0.7

Purpose: Displaying customizable toast messages.

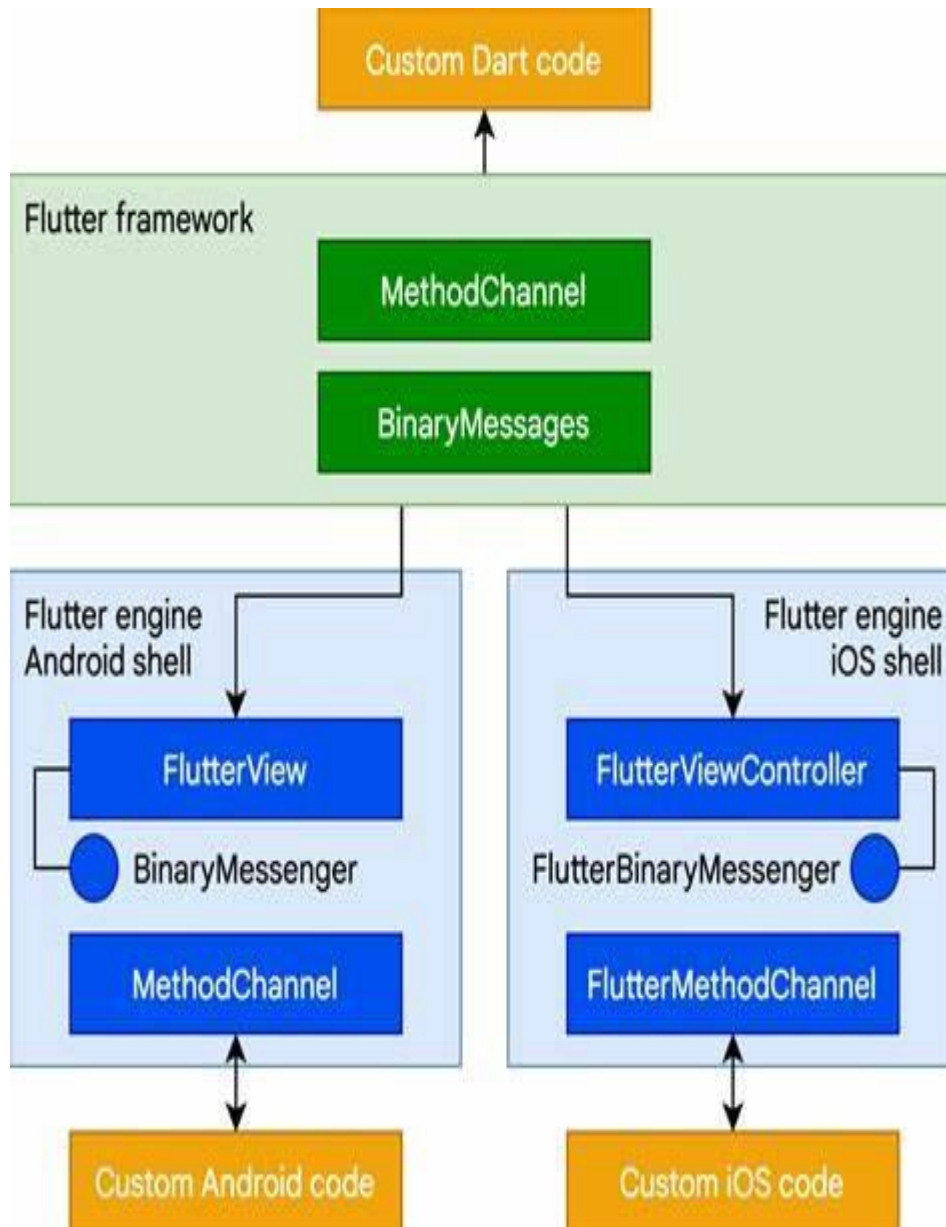
xi)image\_picker

Version: 1.0.7

Purpose: Picking images from the gallery or taking photos.

## 2.2.PROPOSED APPLICATION ARCHITECTURE

### 1.Flutter



**Fig:2.1: Flutter Architecture**

The following steps are used to describe Fig:2.1 Flutter Architecture:

**i)Custom Dart Code**

This represents any custom code you write in Dart specifically for your Flutter app. It could be business logic, UI components, or any other functionality unique to your application.

**ii)Flutter Framework**

The Flutter framework is the heart of your app. It provides a rich set of widgets, tools, and APIs for building cross-platform mobile applications.

**iii)MethodChannel**

The MethodChannel allows communication between Dart code and native platform code (Android or iOS).

**iv)BinaryMessages**

BinaryMessages is another channel for communication between Dart and native code.

**v)Flutter Engine**

On the Android side, the Flutter engine runs within an Android shell. On iOS, the Flutter engine runs within an iOS shell.

**vi)Custom Android Code and Custom iOS Code**

These represent platform-specific code written in Java/Kotlin (for Android) and Objective-C/Swift (for iOS).

## 2.Firebase



**Fig:2.2: Firebase Architecture**

The following steps are used to describe Fig:2.2 Firebase Architecture:

### i)Firebase Authentication

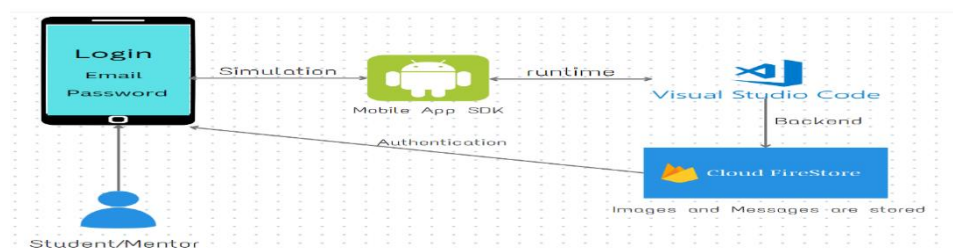
Firestore Authentication offers ready-to-use authentication services. It integrates seamlessly with Flutter, allowing users to sign up, sign in, and manage their accounts.

### ii)Firestore (Realtime Database)

Firestore is a NoSQL database provided by Firebase. It stores structured data in collections and documents. You can use Firestore to store user profiles, chat messages, app settings, and more.

## 3.To integrate Firebase with My Flutter app

Add the necessary Firebase plugins (e.g., `firebase_core`, `firebase_auth`, `cloud_firestore`, etc.) to your `pubspec.yaml`. Initialize Firebase in your app (usually in the `main` function).



**Fig:2.3: Proposed Application Architecture**

The following steps are used to describe Fig:2.3 Proposed Application Architecture:

#### **i)User Interaction**

Users (represented by the “Student/Mentor” icon) log in via a mobile device. The login process involves entering an email and password.

#### **ii)Authentication**

Firebase Authentication handles user sign-in. Methods include Google Sign-In, email/password, and phone number authentication.

#### **iii)Development Components**

Simulation (Mobile App SDK): Represents the mobile app development environment. Developers use tools like Android Studio or Xcode.

Runtime (Visual Studio Code Backend): Represents the backend development environment. Backend logic, APIs, and data storage are implemented here.

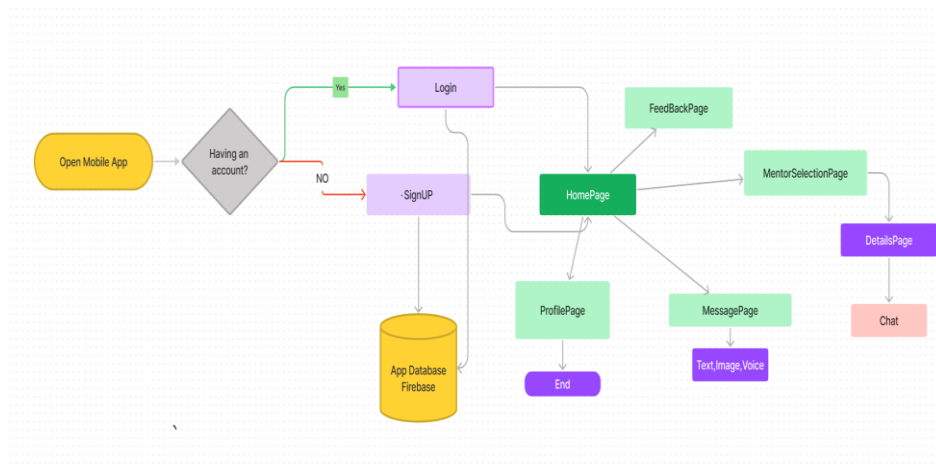
#### **iv)Data Storage**

Cloud Firestore: Data (images, messages) is stored in Cloud Firestore. Real-time synchronization and efficient querying are supported.

#### **v)Flow and Connections**

1. Arrows connect components, indicating data flow. User login triggers authentication.
2. Authenticated users interact with the app (Mobile App SDK).

### 2.3.DATA FLOW DIAGRAM



**Fig:2.4: Data Flow Diagram**

The following steps are used to describe Fig:2.4 Data Flow Diagram:

- i)The user opens the mobile app.
- ii)The app prompts the user to log in or sign up if they don't have an account yet.
- iii)If the user chooses to sign up, they will be taken to the sign-up page.
- iv)If the user is already registered, they can log in directly.
- v)Once the user is logged in, they will be taken to the home page.
- vi)From the home page, the user can access various features of the app such as their profile page, feedback page, message page, mentor selection page, and details page.
- vii)The mentor selection page allows the user to choose a mentor and view their details.
- viii)The details page provides information about the mentor.
- ix)The user can also leave feedback on the app.
- x)The chat feature allows the user to communicate with their mentor.

## **2.4.MODULES**

### **2.4.1.USER MANAGEMENT**

Provides a comprehensive registration form(signup) with fields for personal information, contact details, and interests, enhancing user profile completeness.

#### **Functionalities**

i)Collect User Information: When a user accesses the registration form, they are prompted to fill in essential information such as their full name, email address, password, date of birth, and areas of interest. These details are crucial for creating a personalized and relevant user profile.

ii)Validate Input Data: As users enter their information, the system performs validation checks to ensure the accuracy and completeness of the data. For example, it verifies that the email format is correct and that the password meets security requirements (e.g., minimum length, inclusion of special characters).

#### **Algorithm**

1. Create the main registration form using a Scaffold widget.
2. Include fields for personal information (name, date of birth), contact details (email, password), and interests.
3. Validate input data (email format, password strength) using Flutter's form validation techniques.
4. Upon successful validation, create a new user account and store the information in Firebase Authentication.

### **2.4.2.AUTHENTICATION**

Implements secure login functionality with email/password or social login options, providing users with seamless access to the platform.

#### **Functionalities**

i)Display Login Interface: Upon accessing the platform, users are presented with a login interface that allows them to enter their credentials for authentication.

ii)Verify User Credentials: When users attempt to log in, the system verifies their credentials, such as their email address and password, against the stored data in the database. For social login, the system validates the authentication tokens provided by the respective platforms.

iii)Grant Access: If the entered credentials or authentication tokens are valid, the system grants the user access to the platform, allowing them to explore MentorSynth Hub's features and functionalities.

#### **Algorithm**

1. Create a login\_page.dart with a Scaffold widget for the login interface.
2. Implement email/password and social login options.
3. Verify user credentials against Firebase Authentication.
4. Upon successful authentication, grant access and manage user sessions using Firebase Authentication.

### **2.4.3.PROFILE MANAGEMENT**

Enables users to update profiles, upload pictures, set availability, and manage notifications, enhancing personalization and user control.

#### **Functionalities**

i)Access Profile Settings: Users can access their profile settings through a dedicated interface within MentorSynth Hub, providing them with centralized control over their profile information and preferences.

ii)Update Profile Information: The profile management functionality allows users to update their profile details, such as their name, contact information, interests, and any additional information relevant to their mentorship experience.

iii)Grant Access: If the entered credentials or authentication tokens are valid, the system grants the user access to the platform, allowing them to explore MentorSynth Hub's features and functionalities.

#### **Algorithm**

1. Design a profile settings page using a Scaffold widget.
2. Allow users to update profile information (name, contact details, interests).
3. Implement profile picture upload functionality and store images in Firebase Storage.
4. Include options to set availability status (online/offline) for mentorship sessions.

#### **2.4.4.MENTOR SELECTION**

Implements collaborative filtering algorithms considering user preferences, mentor expertise, availability, and ratings, enhancing accurate mentor-mentee matches.

##### **Functionalities**

i)Data Collection: Gather user preferences, mentor attributes, availability schedules, and historical ratings or feedback from the database. Capture user interests, mentor expertise areas, preferred session timings, and past interaction patterns.

ii)User-Driven Selection Process: Design a user-friendly interface for browsing and exploring recommended mentors, showcasing relevant information such as mentor profiles, expertise areas, availability, ratings, and reviews.

iii)User Feedback: Allow users to review and provide feedback on recommended mentors, enabling them to express preferences and refine their selection criteria.

##### **Algorithm**

1. Create a page to display recommended mentors using a Scaffold widget.
2. Fetch mentor data from Firebase Firestore based on collaborative filtering algorithms.
3. Display detailed mentor information (skills, ratings) in a ListView with interactive features for filtering and sorting.
4. Allow users to provide feedback and ratings for mentors.

#### **2.4.5.MESSAGING AND COLLABORATION**

Integrates real-time messaging using WebSockets or Firebase, enabling instant communication and collaboration between mentors and mentees. Facilitates file uploads and downloads during messaging sessions, enhancing collaboration and resource sharing.

##### **Functionalities**

i)Real-Time Messaging Setup: Initialize real-time messaging using WebSockets or Firebase Realtime Database. Connect users to messaging channels based on their roles (mentors, mentees) and conversations (individual chats, group chats).

ii)Message Sending: Mentors or mentees compose and send messages through the messaging interface. Messages are formatted, timestamped, and sent to the recipient(s) within the messaging channel.

iii)File Uploads: Users initiate file uploads by selecting files from their devices. Uploaded files are transmitted securely and stored temporarily or permanently based on user preferences.

##### **Algorithm**

1. Implement real-time messaging using Firebase Cloud Firestore and Firebase Realtime Database.
2. Design a chat interface with a ListView to display messages.
3. Enable message sending/receiving functionalities in real time.
4. Implement file upload/download capabilities securely using Firebase Storage.

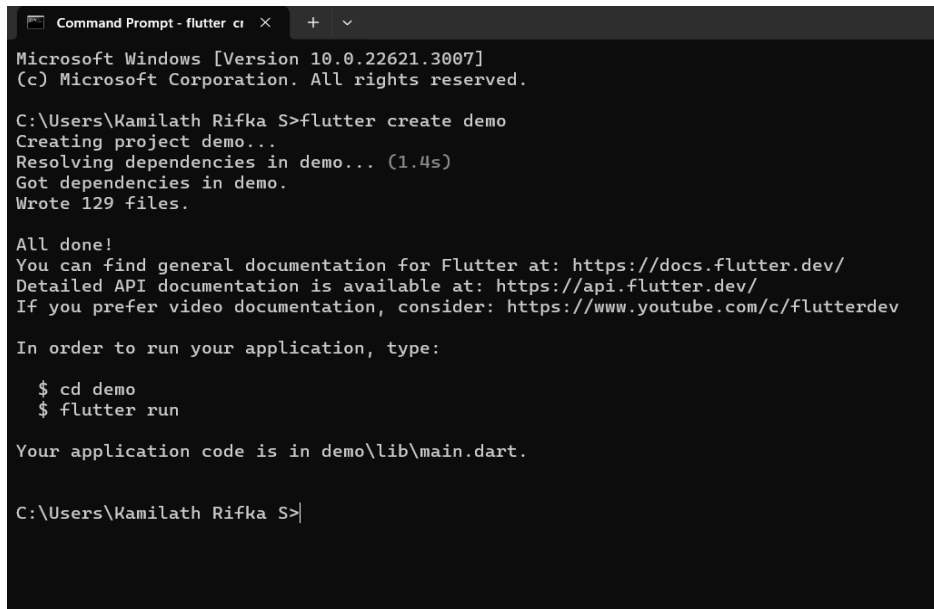
## 2.5.DEVELOPMENT AND DEPLOYMENT SETUP

### 1.Development Environment

Flutter and Dart need to be installed on the developer's machine.

IDEs like Visual Studio Code or Android Studio for coding.

### 2.Create Project



```
Command Prompt - flutter ci x + v
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Kamilath Rifka S>flutter create demo
Creating project demo...
Resolving dependencies in demo... (1.4s)
Got dependencies in demo.
Wrote 129 files.

All done!
You can find general documentation for Flutter at: https://docs.flutter.dev/
Detailed API documentation is available at: https://api.flutter.dev/
If you prefer video documentation, consider: https://www.youtube.com/c/flutterdev

In order to run your application, type:

$ cd demo
$ flutter run

Your application code is in demo\lib\main.dart.

C:\Users\Kamilath Rifka S>
```

***Fig:2.5: Project Creation***

The following steps are used to describe Fig:2.5 Project Creation:

1. >>flutter create project name
2. >>cd project name
3. >>flutter run

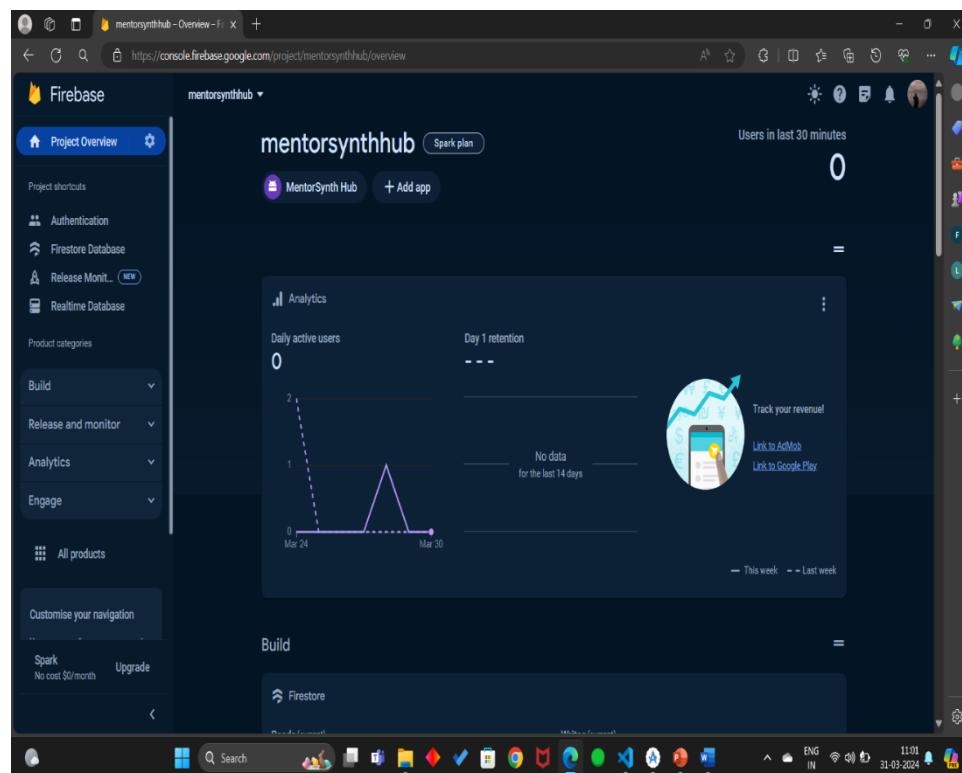
### 3.Dependency Management

Project dependencies specified in the pubspec.yaml file.

Key packages include:

1. firebase\_core for Firebase integration.
2. firebase\_auth for user authentication.
3. cloud\_firestore for real-time data synchronization.
4. Other packages like get, toast, image\_picker, etc., for additional functionalities.

### 4.Firebase Configuration

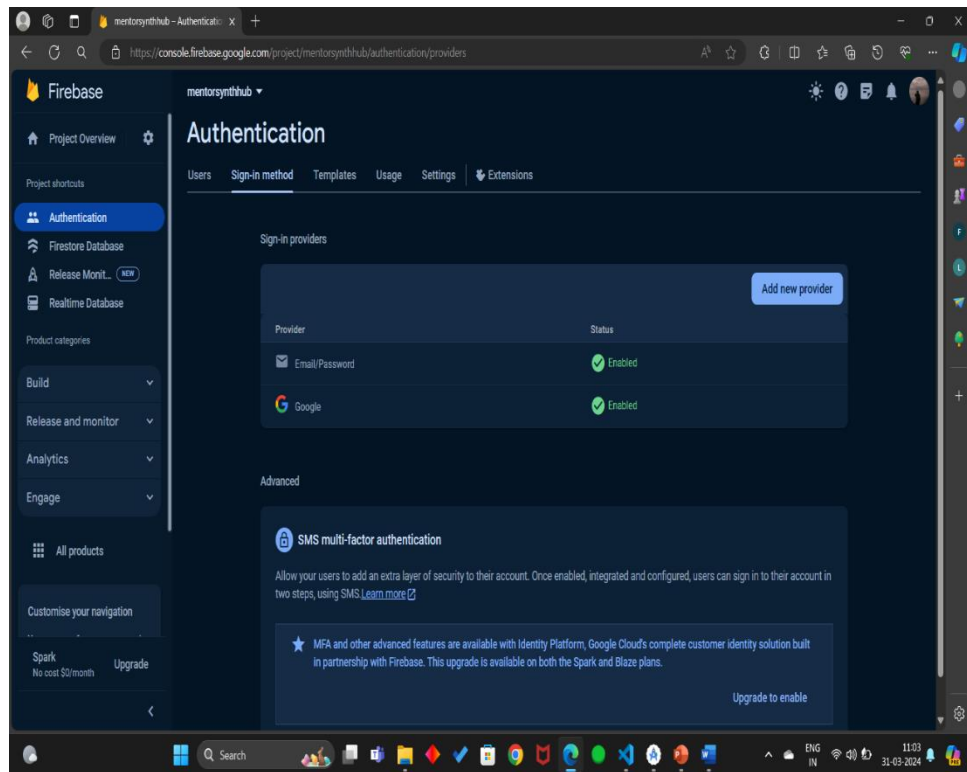


**Fig:2.6: Firebase Connection**

The following steps are used to describe Fig:2.6 Firebase Connection:

1. Firebase project created and configured for the application.
2. Firebase Hosting used for deploying the Flutter application.

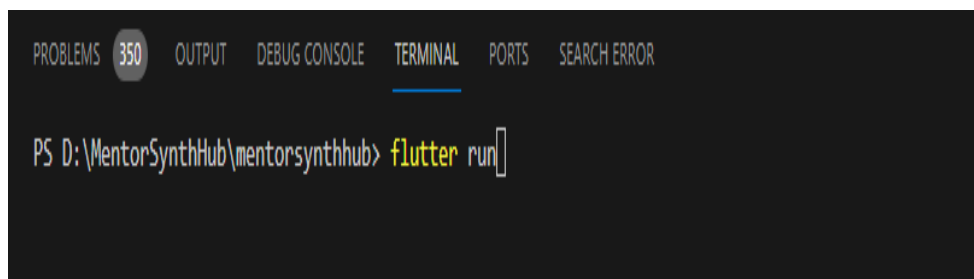
## 5. Firebase Authentication



**Fig:2.7: Firebase Authentication**

Fig:2.7 Firebase Authentication details the setup of Firebase Authentication for secure user login and identity verification.

## 6. Development Tools



**Fig:2.8: Flutter Run**

Fig:2.8 Flutter Run demonstrates Flutter commands used for testing and building the application. Real-time testing on Android and iOS devices.

## **7.Deployment Process**

Firebase used for hosting the web version of the application.

Seamless deployment using Flutter and Firebase commands.

## **8.Supported Platforms**

Cross-platform support for Android, iOS, and web.

## **9.Integrated Development Environment (IDE)**

Developers can choose between Visual Studio Code or Android Studio for coding.

## **10.User Authentication**

Firebase Authentication ensures secure user login.

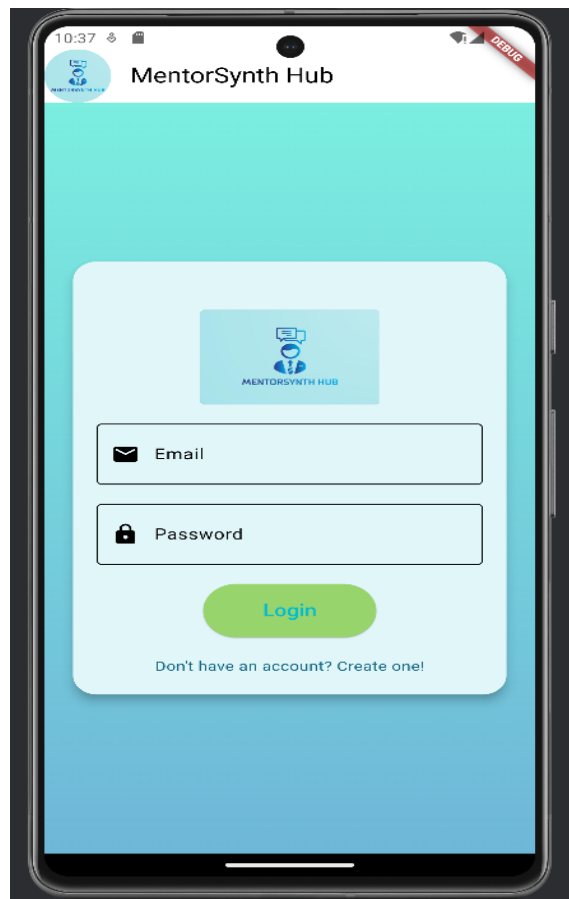
## CHAPTER 3

### RESULTS AND DISCUSSION

#### DART CODES

i)main.dart: This file is typically the entry point of a Flutter application. It initializes the app, sets up necessary configurations, and runs the main application widget.

ii)loginPage.dart:



***Fig:3.1: Login Page Interface***

Fig:3.1 Login Page Interface illustrates the UI and logic for user login. It includes form fields for entering credentials and a mechanism to authenticate users.

iii)signupPage.dart: Similar to the login file, signupPage.dart is likely responsible for handling user registration. It includes form fields for users to create accounts.

iv)homePage.dart: This file represents the main content of your app after a successful login or signup. It contains messages, profilePage, feedbackPage and mentorSelection Page with various features.

v)profilePage.dart:



***Fig:3.2: Dashboard Screen Display***

Fig:3.2 Dashboard Screen Display illustrates the functionality responsible for displaying and editing user profiles. This feature encompasses information about the user, such as their name, profile picture, and additional details that can be modified or updated as needed.

vi)messagePage.dart: This file could represent a page where users can exchange messages.

vii)message\_model.dart: This file defines the structure of a Message model.

viii)messageForm.dart: This file represents a form for users to send messages.

ix)messageList.dart: This file creates a list to display messages.

x)messageCard.dart: This file defines a card to display individual messages.

xi)mentor.dart: This file defines the structure of a Mentor model.

xii)mentorSelectionPage.dart:



**Fig:3.3: Mentor Selection Interface**

Fig:3.3 Mentor Selection Interface depicts a page dedicated to listing mentors available for selection.

xiii) mentorCard.dart: This file represents a card to display individual mentors

xiv) mentorDetailsPage.dart: This file creates a details page for an individual mentor.

xv) chat\_model.dart: This file defines the structure of a Chat model.

xvi) chat\_screen.dart: This file represents a screen where the chat messages are displayed.

xvii) feedBackPage.dart:



**Fig:3.4: Feedback Submission Screen**

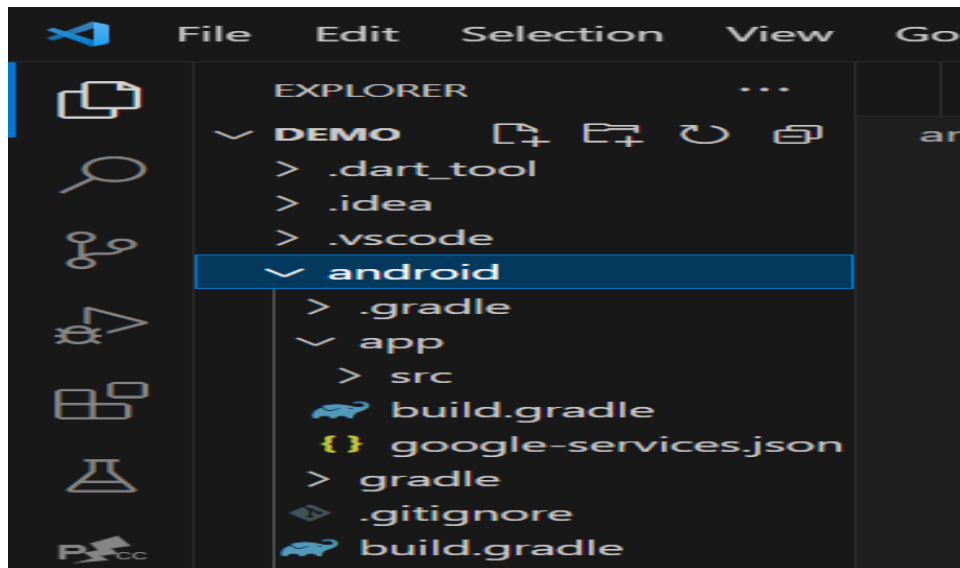
Fig:3.4 Feedback Submission Screen showcases a feedback page designed for users to provide their feedback.

## FIREBASE

### i)Create a Firebase Project

1. Go to the Firebase Console.
2. Click on "Add Project" and follow the setup instructions.

### ii)Configure Flutter Project



**Fig:3.5: Firebase Configurations**

The following steps are used to describe Fig:3.5 Firebase Configurations:

1. Add the Firebase configuration files to your Flutter project.
2. Follow the instructions provided in the Firebase Console to add google-services.json for Android and GoogleService-Info.plist for iOS.
3. These files contain configuration details needed for your app to connect to Firebase services.

### iii)Add Firebase Dependencies

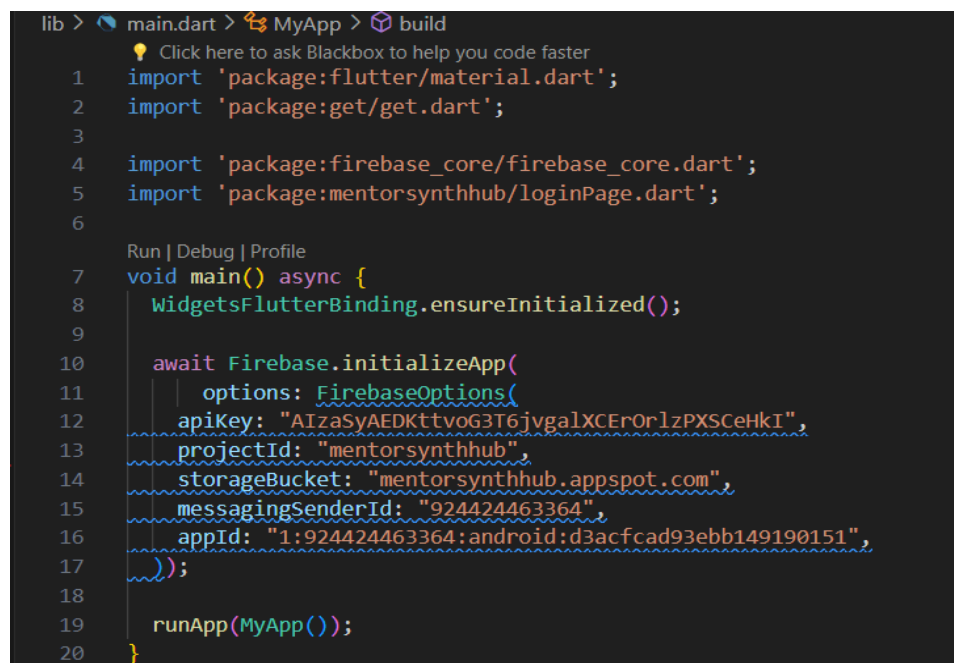
Open the pubspec.yaml file and add the following dependencies:

dependencies:

firebase\_core: ^2.25.3

firebase\_auth: ^4.17.5

### iv)Initialize Firebase



```
lib > main.dart > MyApp > build
  Click here to ask Blackbox to help you code faster
1  import 'package:flutter/material.dart';
2  import 'package:get/get.dart';
3
4  import 'package:firebase_core/firebase_core.dart';
5  import 'package:mentorsynthhub/loginPage.dart';
6
7  Run | Debug | Profile
8  void main() async {
9
10     WidgetsFlutterBinding.ensureInitialized();
11
12     await Firebase.initializeApp(
13       options: FirebaseOptions(
14         apiKey: "AIzaSyAEDKttvoG3T6jvgalXCEROrlzPXSCeHkI",
15         projectId: "mentorsynthhub",
16         storageBucket: "mentorsynthhub.appspot.com",
17         messagingSenderId: "924424463364",
18         appId: "1:924424463364:android:d3acfcad93ebb149190151",
19       ));
20
21     runApp(MyApp());
22 }
```

**Fig:3.6: Firebase Initialization**

Fig:3.6 Firebase Initialization highlights the initialization of Firebase within your main.dart file or wherever you initialize your Flutter app. It emphasizes the importance of initializing Firebase to enable functionalities like user authentication. Firebase offers the firebase\_auth package for implementing user authentication, allowing you to sign in users using methods such as email/password authentication, Google Sign-In.

## **CHAPTER 4**

### **CONCLUSION AND FUTURE WORK**

#### **4.1. CONCLUSION**

In conclusion, MentorSynth Hub represents a cutting-edge mobile application that seamlessly blends academic learning with professional mentorship. Through the innovative use of Flutter and Dart, our project provides an intuitive and cross-platform user interface, ensuring a consistent and engaging experience for both students and mentors on iOS and Android devices. The integration of Firebase further enhances the app's functionality, offering secure user authentication.

MentorSynth Hub serves as a dynamic hub for collaborative learning, addressing not only academic queries but also facilitating discussions on job-related challenges, interview preparation, and industry-specific topics. The inclusive platform encourages a diverse community of learners and mentors to actively contribute, share expertise, and collectively bridge the gap between theoretical knowledge and practical applications.

With its user-friendly design, real-time communication features, and the ability to foster meaningful connections, MentorSynth Hub redefines mentorship in the digital age. The project stands as a testament to the power of technology in creating an enriched learning environment that connects individuals from various backgrounds, ultimately promoting professional growth and knowledge synthesis.

## **4.2. FUTURE WORK**

### **1.Enhanced User Profiles**

Implement additional fields and customization options in user profiles, allowing users to showcase more details about their skills, interests, and experiences.

### **2.Advanced Mentor Matching Algorithm**

Develop a more sophisticated algorithm for mentor-student matching, considering factors like specific expertise, project interests, and availability to enhance the quality of mentorship connections.

### **3.Real-time Collaboration Tools**

Integrate real-time collaboration tools, such as collaborative document editing, code sharing, or video conferencing, to facilitate more interactive mentor-student sessions.

### **4.Community Forums and Discussions**

Expand the platform by introducing community forums or discussion boards where users can engage in broader discussions, ask open-ended questions, and share knowledge with a wider audience.

### **5.Integration with Learning Resources**

Collaborate with educational platforms or content providers to integrate learning resources directly into the app, providing users with access to tutorials, courses, and industry-specific materials.

## **6.Gamification Elements**

Introduce gamification elements, such as achievement badges, leaderboards, or challenges, to incentivize user engagement and recognize contributions within the MentorSynth Hub community.

## **7.Machine Learning for Recommendation**

Explore the use of machine learning algorithms to enhance mentor recommendations based on historical user data, preferences, and successful mentor-student interactions.

## **8.Mobile App Accessibility**

Ensure the app's accessibility by incorporating features like voiceover support, larger text options, and other accessibility improvements to cater to a diverse user base.

## **9.Feedback and Rating System**

Implement a comprehensive feedback and rating system for both mentors and students, allowing users to provide reviews and ratings after mentorship sessions to build a reputation system.

## **10.Multi-Language Support**

Expand the application's reach by incorporating multi-language support, allowing users from different linguistic backgrounds to use MentorSynth Hub comfortably.

## REFERENCES

### 1.Journals

[1] Payne, R. (2019). Using firebase with flutter. *Beginning App Development with Flutter*, 12(1), 255–285.

[2] Tyagi, P. (2021). Building layouts. *Pragmatic Flutter*, 2(3), 87-126.

[3] Tyagi, P. (2021). Setting up environment. *Pragmatic Flutter*, 1(2), 23–34.

### 2.Books

[1] A. K. S. (2018). *Mastering Firebase for Android Development: Build Real-Time, Scalable, and Cloud-Enabled Android Apps with Firebase*. Birmingham, UK: Packt Publishing, Ltd.

[2] Iversen, J. H., & Eierman, M. A. (2014). *Learning Mobile App Development: A Hands-on Guide to Building Apps with IOS and Android*. Upper Saddle River, New Jersey: Addison-Wesley.

[3] Miola, A. (2023). *Flutter Complete Reference 2.0: The Ultimate Reference for Dart and Flutter*. Independently published.

### **3.Internet Articles**

[1] Camp, D. (2024). Best flutter blog articles from around the web. Flutter articles, <https://www.devglan.com/programming/flutter-article-feeds>.

[2] Wilson, W. (2014). The ultimate list of flutter resources(over 200!).Flutter Resources, <https://wilsonwilson.dev/articles/flutter-resources> .

## APPENDICES

### A.SAMPLE CODE

#### i)Main Code

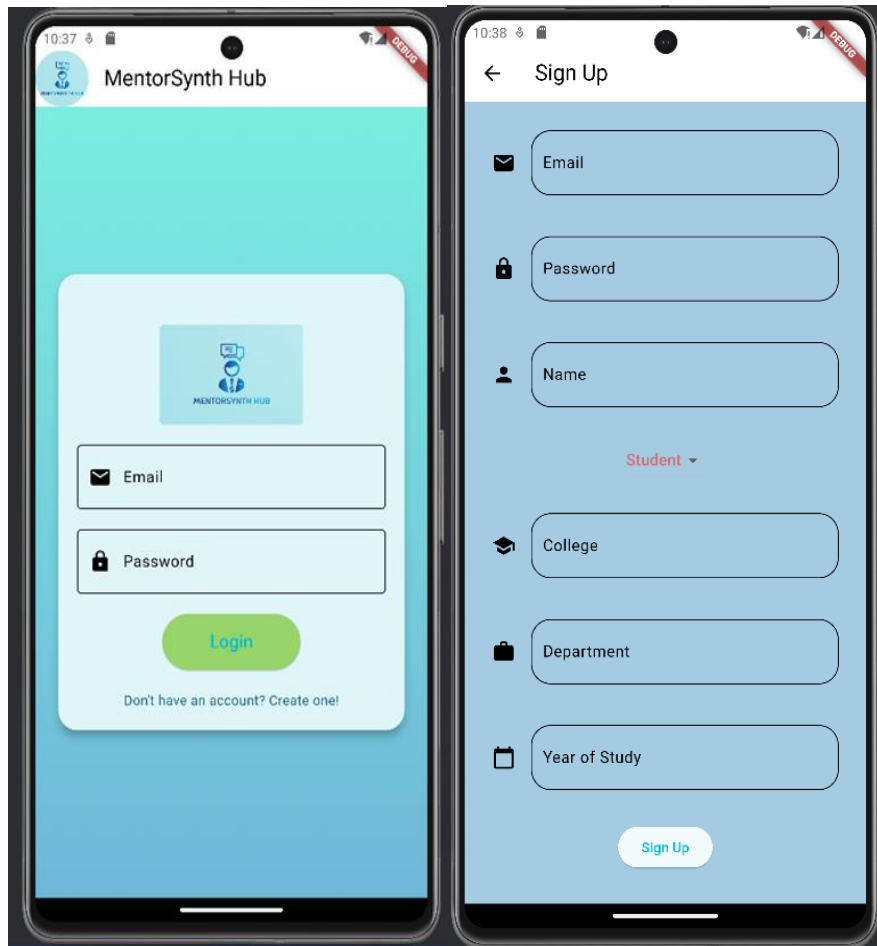
```
lib > main.dart > MyApp > build
  Click here to ask Blackbox to help you code faster
1  import 'package:flutter/material.dart';
2  import 'package:get/get.dart';
3
4  import 'package:firebase_core/firebase_core.dart';
5  import 'package:mentorsynthhub/loginPage.dart';
6
7  void main() async {
8    WidgetsFlutterBinding.ensureInitialized();
9
10   await Firebase.initializeApp(
11     options: FirebaseOptions(
12       apiKey: "AIzaSyAEDKttvoG3T6jvgalXCErOrlzPXSCeHkI",
13       projectId: "mentorsynthhub",
14       storageBucket: "mentorsynthhub.appspot.com",
15       messagingSenderId: "924424463364",
16       appId: "1:924424463364:android:d3acfcad93ebb149190151",
17     ));
18
19   runApp(MyApp());
20 }
```

ii)pubspec.yaml

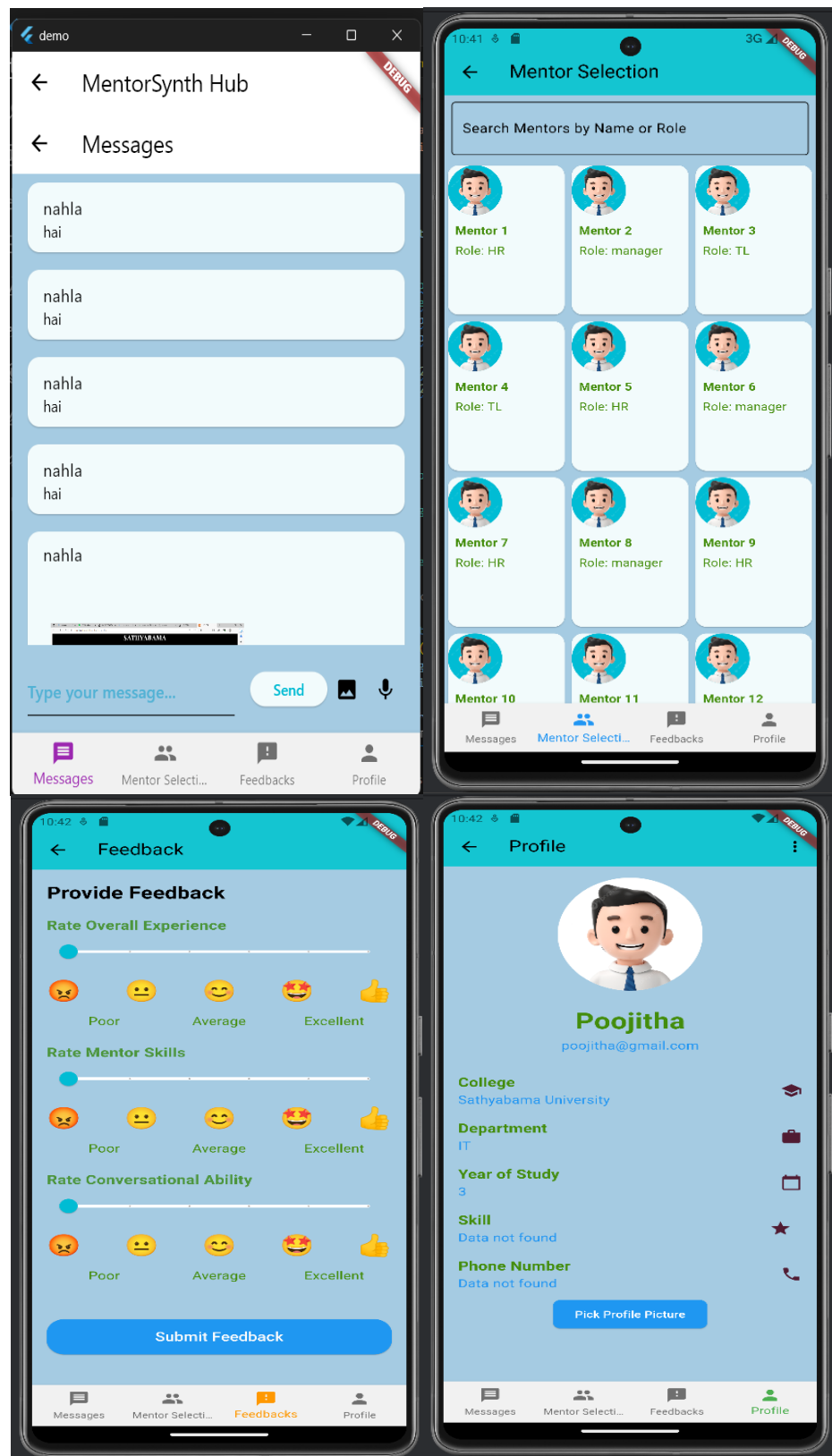
```
! pubspec.yaml ●
D: > MentorSynthHub > mentorsynthhub > ! pubspec.yaml
  ⚡ Click here to ask Blackbox to help you code faster
1  name: mentorsynthhub
2  description: "A new Flutter project."
3  publish_to: 'none'
4  version: 1.0.0+1
5  environment:
6    sdk: '>=3.2.6 <4.0.0'
7  dependencies:
8    flutter:
9      sdk: flutter
10   cupertino_icons: ^1.0.2
11   get: ^4.6.6
12   firebase_core: ^2.25.3
13   firebase_auth: ^4.17.3
14   cloud_firestore: ^4.15.3
15   cloud_firestore_web: ^3.10.3
16   firebase_auth_web: ^5.9.3
17   firebase_core_web: ^2.11.3
18   google_sign_in: ^5.2.1
19   toast: ^0.3.0
20   fluttertoast: ^8.0.7
21   image_picker: ^1.0.7
22   permission_handler: ^11.2.0
23   provider: ^6.1.1
24   flutter_rating_bar: ^4.0.0
25  dev_dependencies:
26    flutter_test:
27      sdk: flutter
28    flutter_lints: ^3.0.1
29  flutter:
30    uses-material-design: true
31    assets:
32      - asset/
33      - asset/R (1).jpg
34      - asset/profile_photo.png
35      - asset/image.jpg
36
```

## B.SCREENSHOTS

i)Login page And Signup Page



ii) Homepage & Navigation bar



### iii) Select & Chat with Mentors

