

PROJECT PAIRING HUB

INTERDISCIPLINARY PROJECT

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

HENNY STEPHEN M(41120070)



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

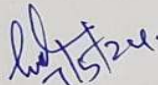
[DEEMED TO BE UNIVERSITY]

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

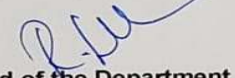
DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **HENNY STEPHEN M(41120070)** who carried out the project entitled "**PROJECT PAIRING HUB**" under my supervision from January 2024 to May 2024.

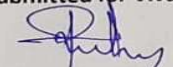

Internal Guide

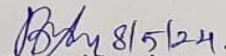
Dr.C.GEETHA,M.Tech.,Ph.D.,


Head of the Department

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

Submitted for Viva voce Examination held on 08-05-2024


Internal Examiner


External Examiner

DECLARATION

I **Henny Stephen M** hereby declare that the Project Report entitled "PROJECT PAIRING 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: 08-05-24

PLACE:CHENNAI

M. Henny Stephen
SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

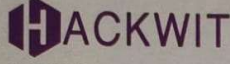
I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

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**
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

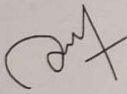
TO WHOMSOEVER IT MAY CONCERN

This is to Certify that **M. Henny Stephen (Reg No: 41120070)** student of Sathyabama Institute of Science and Technology, B. Tech Information Technology, has successfully completed the Internship at Hackwit Technologies from 22nd January to 30th January 2024.

During the internship, **M. Henny Stephen** has closely worked as a part of the **Flutter** by carrying out the project entitled **"Project Pairing Hub"**.

He demonstrated good developing and learning skill with self-motivated attitude to learn new things. His contribution to implement the Flutter was good. We wish him all the best for his future endeavors.

Best Regards,



Ravikumar S
CEO & Managing Director
Hackwit Technologies Pvt Ltd

CERTIFICATE OF COMPLETION



ABSTRACT

"Project Pairing Hub" is an innovative mentorship platform developed using Flutter, a cross-platform framework, aiming to foster connections between students and specialized mentors within their fields of interest. This comprehensive ecosystem enables students to conveniently identify, engage, and collaborate with mentors, enhancing their learning experiences and professional growth. Driven by user-centric design principles, "Project Pairing Hub" offers an intuitive interface to optimize user engagement and satisfaction. Leveraging Flutter's versatile widget library ensures a consistent and visually appealing experience across Android and iOS platforms. At the core of the platform lies a robust backend system, employing sophisticated algorithms to match students with mentors based on factors like expertise, availability, and shared interests. This matching process ensures compatibility and maximizes the effectiveness of mentorship relationships. Key features include comprehensive user profiles, advanced search capabilities, secure messaging and video conferencing functionalities, along with a feedback and review system for continuous improvement. These features foster dynamic and interactive mentorship relationships, contributing to impactful learning experiences. Prioritizing scalability, data privacy, and security, "Project Pairing Hub" provides a reliable and secure platform for mentorship activities. Through iterative development and user feedback, the platform evolves to offer an ever-improving experience for both students and mentors.

TABLE OF CONTENTS

Chapter No.	TITLE	Page No.
	ABSTRACT	vii
	LIST OF ABBREVIATIONS	x
	LIST OF FIGURES	xi
1	INTRODUCTION	1
	1.1 Flutter	3
	1.2 Dart	4
	1.3 Firebase	5
	1.4 Problem Statement	6
	1.5 Objectives	8
2	METHODOLOGY	10
	2.1 Initial Project Requirements	10
	2.2 Proposed System Architecture	14
	2.3 Flow Diagram	17
	2.4 Modules	18
	2.4.1 Authentication Management	18
	2.4.2 Profile Management	19
	2.4.3 Mentor Selection	20
	2.4.4 Messaging and Collaboration	21

	2.5	Development and Deployment Setup	22
3		RESULTS AND DISCUSSION	25
4		CONCLUSION AND FUTURE WORK	30
	4.1	Conclusion	30
	4.2	Future Work	31
		REFERENCES	33
		APPENDICES	35
	A.	Sample Code	35
	B.	Screen Shots	37

LIST OF ABBREVIATIONS

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

LIST OF FIGURES

Figure No.	Name of the Figure	Page No.
2.1	Proposed Application Architecture	14
2.2	Flow Diagram	17
2.3	Project Creation	22
2.4	Flutter run	23
2.5	User Authentication	23
2.6	Data Model	24
3.1	Main Dart	25
3.2	Login Page	26
3.3	Home Page	27
3.4	Profile Page	28
3.5	Explore Page	28
3.6	Chat Page	29
3.7	Explore Page	29

CHAPTER 1

INTRODUCTION

In today's rapidly evolving academic and professional landscape, mentorship plays a pivotal role in guiding individuals towards success. Recognizing the importance of personalized guidance and support, "Project Pairing Hub" emerges as an innovative solution designed to facilitate mentorship connections between students and specialized guides within their respective domains. Built upon the versatile Flutter framework, this platform represents a fusion of cutting-edge technology and human-centric design principles, aimed at revolutionizing the way mentorship relationships are formed and nurtured. The concept of mentorship has long been revered for its ability to empower individuals to reach their full potential by providing them with the knowledge, skills, and guidance necessary to navigate their academic and professional journeys successfully. However, traditional mentorship models often suffer from limitations such as geographical constraints, limited access to expertise, and lack of scalability. "Project Pairing Hub" seeks to address these challenges by harnessing the power of technology to create a dynamic and accessible ecosystem where mentorship flourishes unhindered by physical barriers. At its core, "Project Pairing Hub" is driven by a commitment to democratizing access to mentorship opportunities and fostering a culture of collaboration and knowledge-sharing. By leveraging sophisticated algorithms and intuitive user interfaces, the platform empowers students to discover mentors whose expertise aligns closely with their interests and aspirations. Whether seeking guidance on academic pursuits, career development, or personal growth, students can find mentors who possess the specialized

knowledge and experience needed to support them on their journey. Furthermore, "Project Pairing Hub" places a strong emphasis on user experience, ensuring that interactions within the platform are seamless, intuitive, and rewarding. Through features such as comprehensive user profiles, advanced search capabilities, secure messaging and video conferencing functionalities, and a robust feedback system, users can engage with confidence, knowing that they have access to the tools and resources needed to facilitate meaningful mentorship relationships. In the pages that follow, we will explore the key features, functionalities, and design principles that underpin "Project Pairing Hub," as well as the potential impact it stands to have on the landscape of mentorship and education.

Through a combination of technological innovation, human-centered design, and a commitment to continuous improvement, "Project Pairing Hub" seeks to empower individuals to unlock their full potential and achieve their goals with the guidance and support of experienced mentors.

he 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 Project Pairing 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 Project Pairing 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, Project Pairing 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.

In essence, Flutter empowers Project Pairing Hub to provide a unified cross-platform experience, where the focus remains on delivering an exceptional user interface and optimal performance. The adoption of Flutter aligns with the project's commitment to technological excellence, enabling Project Pairing Hub to stand out as a modern and user-friendly mentorship platform.

1.2. DART

i)Unified Codebase:

Project Pairing 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 Project Pairing 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 Project Pairing 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 Project Pairing 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

Mentorship often fail to provide a comprehensive solution that addresses the diverse needs of students in both academic and professional spheres. As students navigate their educational journey, they encounter complex academic concepts, uncertainties regarding future career paths, and a growing demand for mentorship that transcends conventional boundaries.

Contemporar

i)Real -time Data Synchronization:

Firebase Cloud Firestore supports real-time data synchronization, enabling instant updates across multiple devices. This functionality is essential for Project Pairing Hub, ensuring mentors and students experience immediate changes in discussions, mentorship requests, and collaborative interactions.

ii)Firebase Authentication for User Security:

Project Pairing Hub prioritizes user security with Firebase Authentication, offering a secure and streamlined authentication process. Utilizing features like multi-factor authentication and identity verification, the platform ensures that only authorized users can access its services, enhancing trust and confidentiality.

iii)Serverless Cloud Functions:

Leveraging Firebase's serverless cloud functions, Project Pairing Hub executes backend logic seamlessly without managing servers. These functions handle tasks like sending notifications, updating user data, and managing asynchronous operations, enhancing scalability and operational efficiency.

iv)Firebase Cloud Storage for Media Assets:

Storing media assets, such as profile pictures and discussion attachments, is simplified with Firebase Cloud Storage. Project Pairing Hub utilizes this feature to efficiently manage and serve multimedia content, ensuring a seamless user experience for mentors and students alike.

1.4. PROBLEM STATEMENT

In the modern educational landscape, students face multifaceted challenges that extend beyond traditional academic boundaries. Existing platforms for learning and education presents students with a myriad of challenges, from mastering intricate academic subjects to navigating the transition into the professional realm. However, current educational platforms lack integration, leaving critical gaps in academic support and career guidance. This fragmentation hinders the seamless synthesis of knowledge and skills, impeding students' holistic development.

Additionally, the existing platforms suffer from a lack of inclusivity, limiting the diversity of perspectives and experiences available to users. Many students, despite having valuable insights to contribute, find themselves marginalized in collaborative learning environments. This exclusionary nature stifles the potential for vibrant exchange and hampers the overall growth and development of individuals.

The primary challenge addressed by Project Pairing Hub lies in the fragmentation of available resources,

By creating a unified platform that integrates academic support, career guidance, and mentorship opportunities, Project Pairing Hub seeks to revolutionize the educational experience, empowering individuals to navigate their academic and professional endeavors with confidence and success..

1.5. OBJECTIVES

The primary objective of Project Pairing Hub is to establish a cohesive and inclusive platform that seamlessly integrates academic support, professional mentorship, and collaborative learning. The platform aims to address the fragmented nature of existing educational resources by providing a comprehensive solution for both academic excellence and professional growth. Project Pairing Hub seeks to cultivate a vibrant community where students and mentors can engage in meaningful interactions, share valuable insights, and collectively contribute to each other's development. Through this platform, individuals will be empowered to navigate the complexities of academia and the professional world with confidence, fostering holistic growth and success.

Key Challenges:

i) Integration of Comprehensive Features:

Seamlessly integrating diverse functionalities for academic support and professional mentorship within a unified platform. Ensuring a smooth and intuitive user experience while navigating through a wide array of features tailored to meet users' needs effectively.

ii) Inclusive User Interface Design:

Developing a user-friendly interface that caters to a broad spectrum of users, including those with varying levels of technological proficiency. Balancing inclusivity with simplicity and clarity in the platform's design to ensure accessibility for all users.

iii) **Fostering Community**

Engagement: Promoting active engagement from both students and mentors through robust community-building features. Encouraging participation in discussions, query responses, and collaborative learning activities to create a vibrant and supportive community environment.

iv) **Enhancing Security and Privacy Measures:** Implementing stringent security protocols to safeguard user data, particularly in the context of mentorship interactions. Adhering to stringent privacy regulations and standards to instill trust among users regarding the confidentiality of their personal information.

v) **Ensuring Scalability and Optimal Performance:** Designing a scalable app architecture capable of accommodating potential growth in user base and content volume. Optimizing the performance of the platform to deliver a responsive and efficient user experience, especially as the user community expands.

vi) **Streamlining Flutter Codebase Maintenance:** Effectively managing the complexities of a Flutter-based codebase to ensure code readability, maintainability, and scalability. Regularly updating the codebase in line with the latest Flutter releases and best practices to leverage new features and enhancements for continuous improvement.

CHAPTER 2

METHODOLOGY

2.1. INITIAL PROJECT REQUIREMENTS

The initial project requirements for Project Pairing Hub encompass various aspects such as functionalities, technologies, and design considerations. Here are the key requirements:

i)Project Scope:

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

ii)Technologies:

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

iii)Functionalities:

- User Authentication: Implement a secure user authentication system for both mentors and students.
- Question and Answer System: Enable users to post questions and receive answers from mentors or other students.
- Job-Related Queries: Provide a dedicated section for job-related queries, interview preparation, and industry-specific challenges.
- Mentor-Student Matching: Implement a system to match students with mentors based on expertise and interests.

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

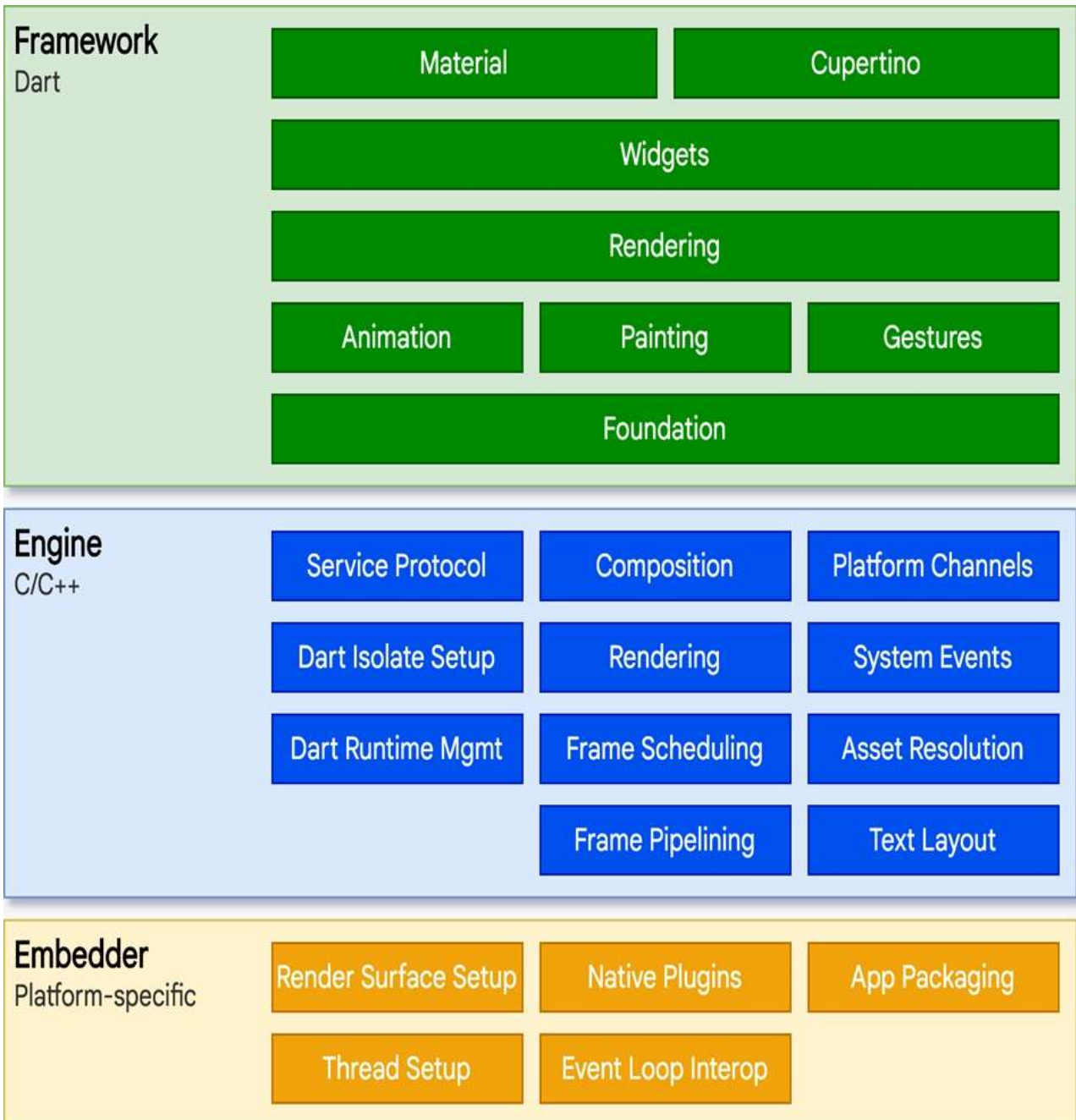


Fig:2.1: Proposed Application Architecture

Fig:2.1 shows the following steps of the Application Architecture:

i) This is the lowest level of the framework and provides an API for embedding Dart into a native application.

ii) This is the core of the Dart programming language, which includes the Dart virtual machine, garbage collector, and just-in-time compiler.

iii) This is the environment in which Dart code runs, including the Dart engine and any necessary runtime services

iv) This is a set of core libraries for building mobile apps, including APIs for assets, collections, files, network requests, and more.

v) This is a set of UI components that can be combined to build mobile app screens. Widgets can be customized and extended to create complex UIs.

vi) This is a layer that handles the layout, painting, and compositing of widgets on the screen

vii) This is a layer that provides APIs for creating complex animations in mobile apps.

viii) This is a set of UI components and styles that follow the design language of iOS.

ix) This is a component that provides a set of UI components and services specific to Android.

x) This is a set of UI components and styles that follow the design language of Android.

xi) This is a layer that provides APIs for detecting and handling user gestures, such as taps, swipes, and pinches.

xii) This is a layer that provides a set of UI styles and themes that can be customized for mobile apps.

xiii) This is a layer that provides a mechanism for communicating between Dart and native platform code.

2.3. FLOW DIAGRAM

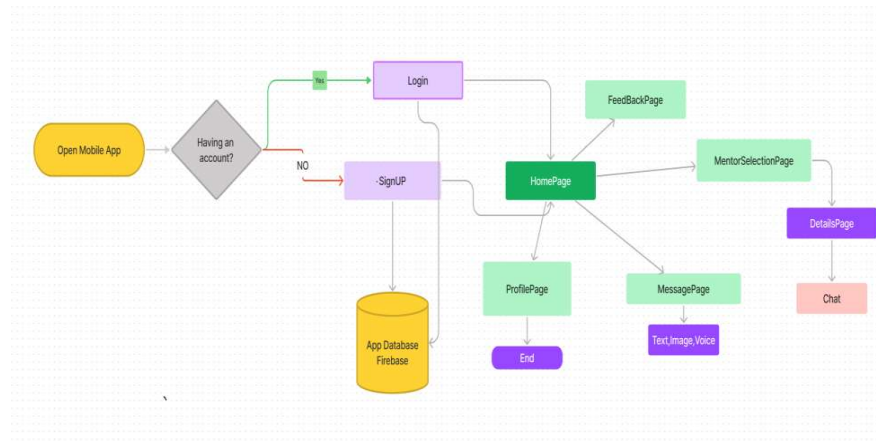


Fig:2.2: Data Flow Diagram

The following steps are used to describe the Fig:2.2 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.AUTHENTICATION:

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

Algorithm:

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 Project Pairing Hub features and functionalities.

iv)Session Management:Upon successful authentication, the system creates a session or issues a token to maintain the user's logged-in state. This session/token management ensures secure and seamless user interactions throughout their session on the platform.

2.4.2.PROFILE MANAGEMENT:

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

Algorithm:

i)Access Profile Settings:Users can access their profile settings through a dedicated interface within Project Pairing 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)Upload Profile Picture:Users have the option to upload profile pictures or avatars, adding a personal touch to their profiles. The system securely stores these images and associates them with the respective user profiles.

iv)Manage Availability:Users can set their availability status (e.g., online/offline) to indicate when they are open for mentorship sessions.

2.4.3.MENTOR SELECTION:

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

Algorithm:

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.

iv)Interactive Selection: Implement interactive features such as filtering, sorting, and bookmarking mentors to facilitate informed and personalized mentorship selection.

v)Real-Time Assistance:Integrate a chat support feature to assist users in their mentor selection process, answering questions, providing guidance, and addressing concerns in real time.

2.4.4.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.

Algorithm:

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)Message Reception:Recipients receive incoming messages in real time.Messages are displayed in the conversation thread with sender details, timestamps, and message content.

iv)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.

v)Collaborative Features:Implement collaborative tools like shared whiteboards or document editors within specific conversation threads.Allow real-time collaboration on shared documents, with version control and concurrent editing capabilities.

2.5. DEVELOPMENT AND DEPLOYMENT SETUP

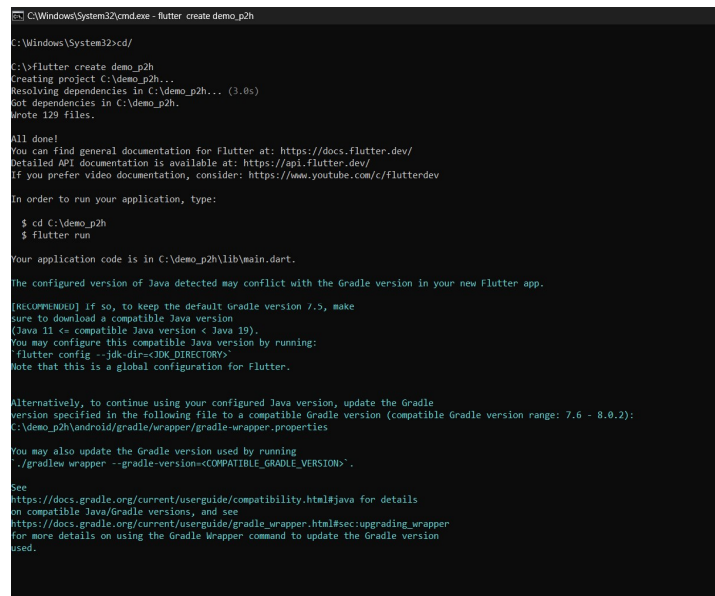
1. Development Environment:

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:

>flutter create project_name



```
C:\Windows\System32\cmd.exe - flutter create demo_p2h

C:\Windows\System32>cd/

C:\>flutter create demo_p2h
Creating project C:\demo_p2h...
Resolving dependencies in C:\demo_p2h... (3.0s)
Got dependencies in C:\demo_p2h.
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 C:\demo_p2h
$ flutter run

Your application code is in C:\demo_p2h\lib\main.dart.

The configured version of Java detected may conflict with the Gradle version in your new Flutter app.

[RECOMMENDED] If so, to keep the default Gradle version 7.5, make
sure to download a compatible Java version
(Java 11 <= compatible Java version < Java 19).
You may configure this compatible Java version by running:
'flutter config --jdk-dir=<JDK_DIRECTORY>'
Note that this is a global configuration for Flutter.

Alternatively, to continue using your configured Java version, update the Gradle
version specified in the following file to a compatible Gradle version (compatible Gradle version range: 7.6 - 8.0.2):
C:\demo_p2h\android\gradle\wrapper\gradle-wrapper.properties

You may also update the Gradle version used by running
'./gradlew wrapper --gradle-version=<COMPATIBLE_GRADLE_VERSION>'.

See
https://docs.gradle.org/current/userguide/compatibility.html#java for details
on compatible Java/Gradle versions, and see
https://docs.gradle.org/current/userguide/gradle_wrapper.html#sec:upgrading_wrapper
for more details on using the Gradle Wrapper command to update the Gradle version
used.
```

Fig:2.3: Project Creation

Fig 2.3 Shows the steps to create project

3. Development Tools:

Flutter commands used for testing and building the application. Real-time testing on Android and iOS devices.

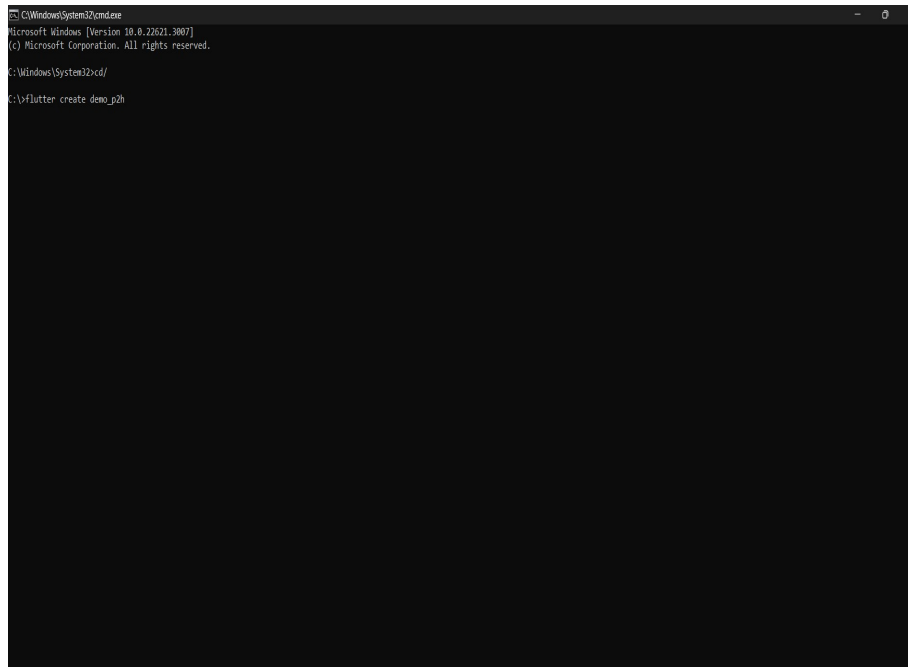


Fig:2.4: Flutter Run

Fig 2.4 Illustrates flutter command to run the the projects

2.6 ALGORITHMIC APPROACH:

To develop a OCTANE application, you can follow a high-level algorithmic approach.

User Authentication:

```

13 class _LoginPageState extends State<LoginPage> {
14   final _formKey = GlobalKey<FormState>();
15
16   TextEditingController EMAIL = TextEditingController();
17   TextEditingController PASSWORD = TextEditingController();
18
19   void loginAuth() {
20     String login mail = "H";
21     String login pass = "1";
22     String email = EMAIL.text.toString();
23     String password = PASSWORD.text.toString();
24

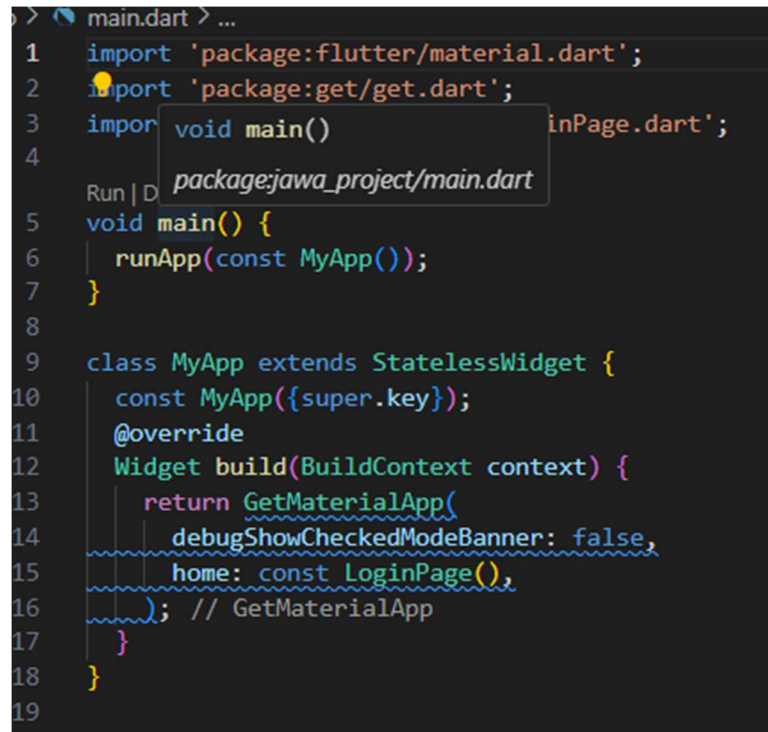
```

Example using LoginPage Authentication dart code:

Fig:2.5:User Authentication

Data Model:

Example using Main.dart code



```
1 import 'package:flutter/material.dart';
2 import 'package:get/get.dart';
3 import 'package:jawa_project/main.dart';
4
5 void main() {
6   runApp(const MyApp());
7 }
8
9 class MyApp extends StatelessWidget {
10   const MyApp({super.key});
11   @override
12   Widget build(BuildContext context) {
13     return GetMaterialApp(
14       debugShowCheckedModeBanner: false,
15       home: const LoginPage(),
16     ); // GetMaterialApp
17   }
18 }
```

Fig:2.6:Dta Model

Testing:

Conduct thorough testing of the application, including unit tests, integrationtests, and user testing to ensure functionality and usability.

Deployment:

Deploy the application on relevant platforms (iOS, Android, web)and make itaccessible to users.

CHAPTER 3

RESULTS AND DISCUSSION

3.1 FRONT END

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.

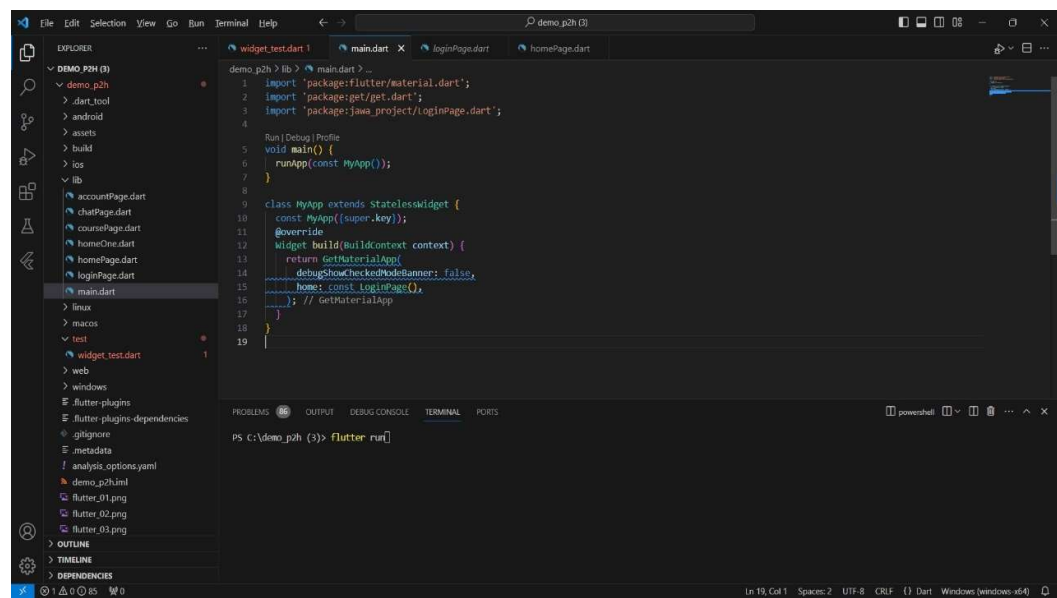


Fig.3.1 Main Dart

ii) **loginPage.dart**

This file likely contains the UI and logic for user login. It includes form fields for entering credentials and a mechanism to authenticate users.

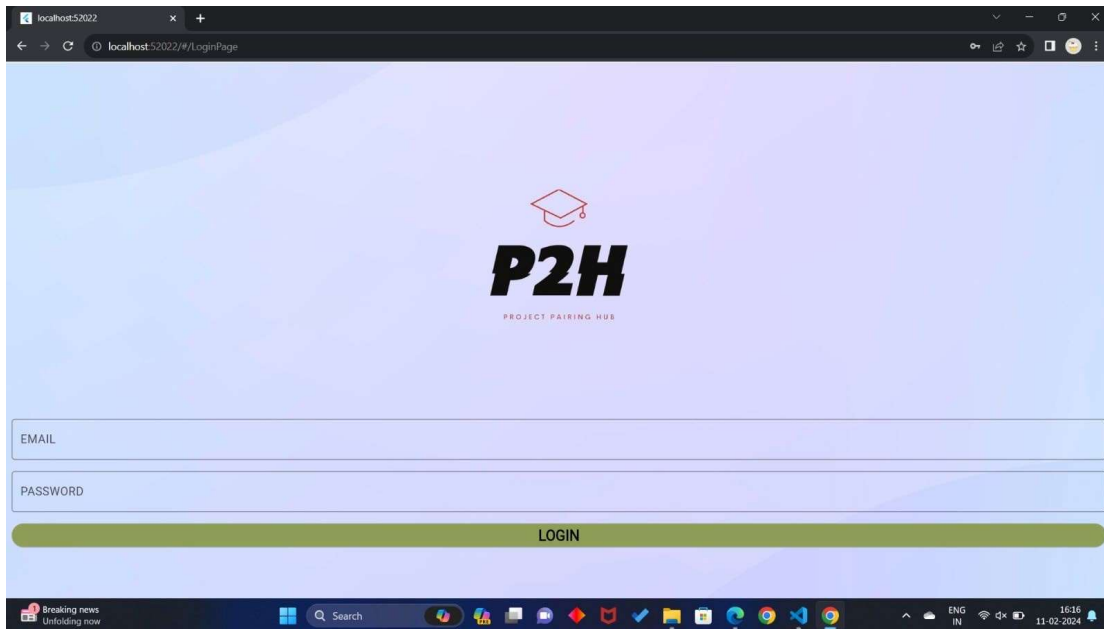


Fig.3.2 Login Page

iii) **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 withvarious features.

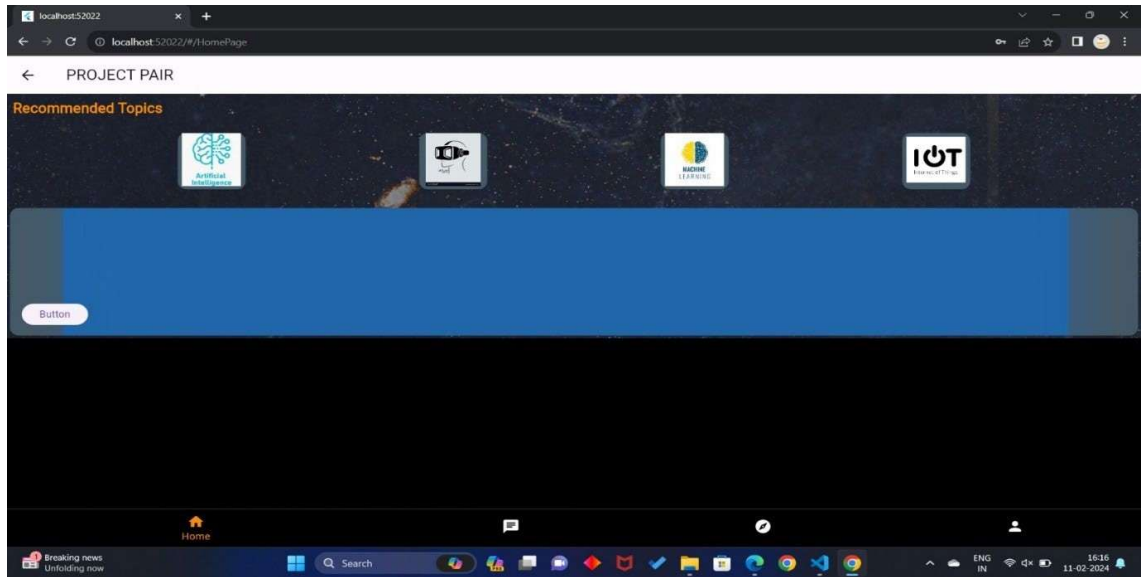


Fig.3.3 Home Page

iv) **profilePage.dart:**

The file is likely responsible for displaying and editing user profiles. It includes information about the user, such as their name, profile picture, and additional details.

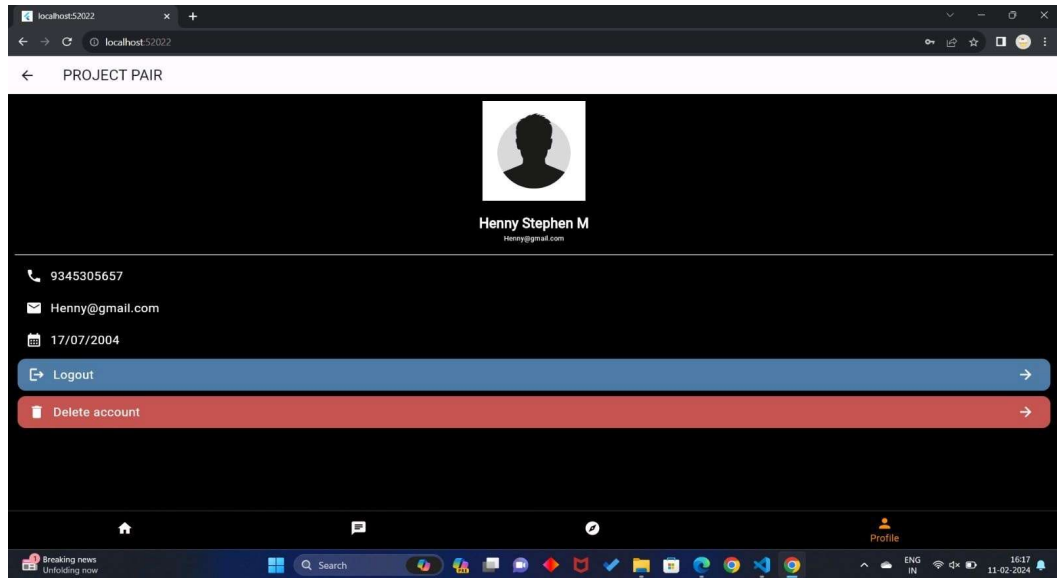


Fig.3.4 Profile Page

v) explorePage.dart:

This file represents a detail page for explorePage.dart

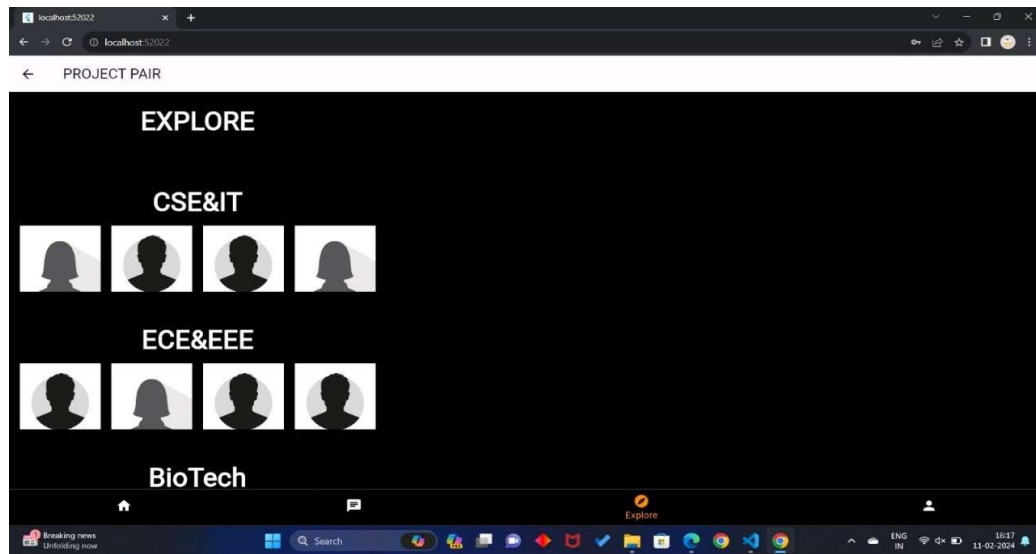


Fig.3.5 Explore Page

vi) chatPage.dart:

This file represents a detail page for chatPage.dart

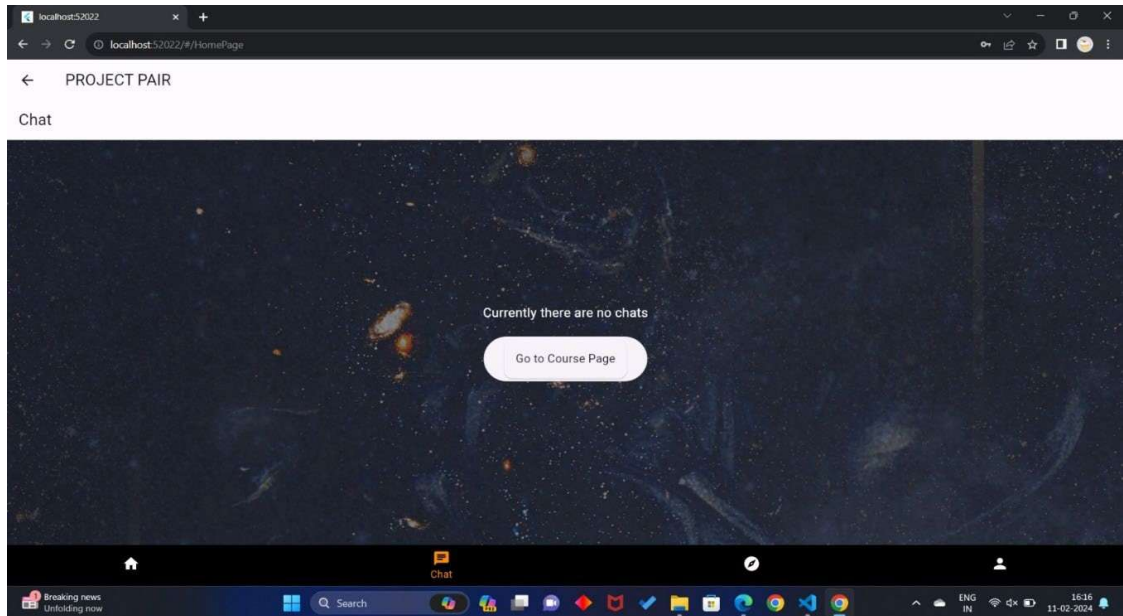


Fig.3.6 Chat Page.

When you click Go to Course page will redirect to explore page

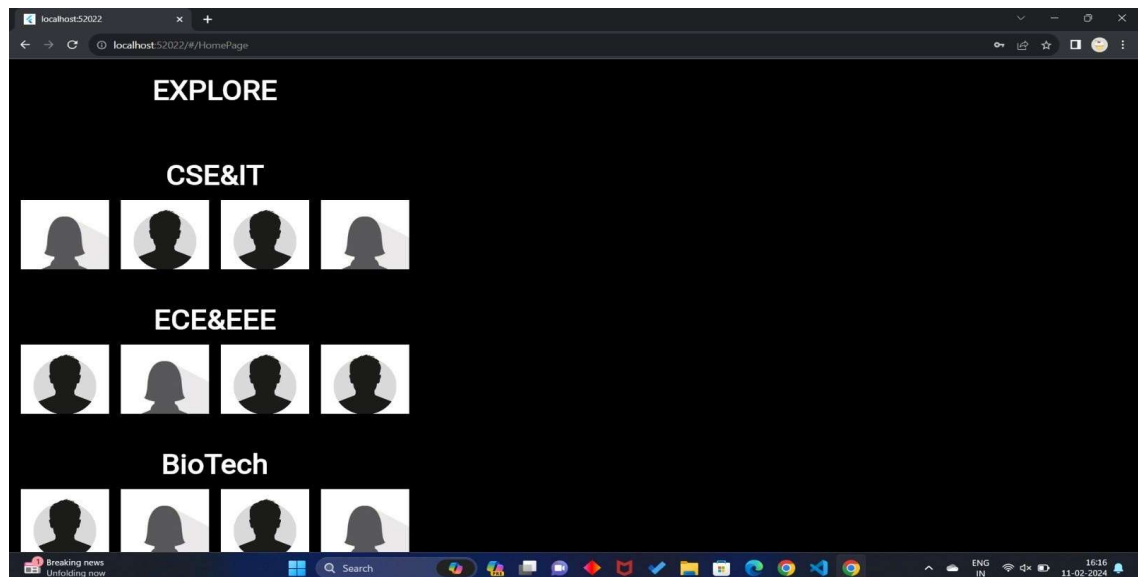


Fig.3.7 Explore Page.

CHAPTER 4

CONCLUSION AND FUTURE WORK

4.1. CONCLUSION

In conclusion, Project Pairing 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.

Project Pairing 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, Project Pairing 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 Project Pairing 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 Project Pairing Hub comfortably.

REFERENCES

Academic Papers and Journals:

- [1] C.-L. Ho, W.-L. Chen, and C.-H. Ou, "Constructing a personalized travel itinerary recommender system with the internet of things," *Wireless Networks*, pp. 321–330, Aug. 2023. doi:10.1007/s11276-023-03453-y
- [2] P. Tiwari, R. P. Kaurav, and K. Y. Koay, "Understanding Travel Apps Usage intention: Findings from PLS and NCA," *Journal of Marketing Analytics*, vol. 12, no. 1, pp. 25–41, Oct. 2023. doi:10.1057/s41270-023-00258-y
- [3] P. Y.-H. Sia, S. S. Saidin, and Y. H. Iskandar, "Systematic review of Mobile Travel Apps and their smart features and challenges," *Journal of Hospitality and Tourism Insights*, vol. 6, no. 5, pp. 2115–2138, Nov. 2022. doi:10.1108/jhti-02-2022-0087

Books:

- [1] A. K. S., *Mastering Firebase for Android Development: Build Real-Time, Scalable, and Cloud-Enabled Android Apps with Firebase*. Birmingham, UK: Packt Publishing, Ltd., 2018.
- [2] A. Miola, *Flutter Complete Reference 2.0: The Ultimate Reference for Dart and Flutter*. Independently published, 2023.
- [3] J. H. Iversen and M. A. Eierman, *Learning Mobile App Development: A Hands-on Guide to Building Apps with IOS and Android*. Upper Saddle River, New Jersey: Addison-Wesley, 2014.

Internet Articles:

- [1] D. Camp, "Best flutter blog articles from around the web," Flutter articles, <https://www.devglan.com/programming/flutter-article-feeds> (accessed Apr. 25, 2024).
- [2] Google, "Firebase documentation," Flutter Documentations, <https://firebase.google.com/docs> (accessed Apr. 25, 2024).
- [3] W. Wilson, "The ultimate list of flutter resources (over 200!)," Flutter Resources, <https://wilsonwilson.dev/articles/flutter-resources> (accessed Feb. 20, 2024).

APPENDICES

A.SAMPLE CODE

i)Main Code:

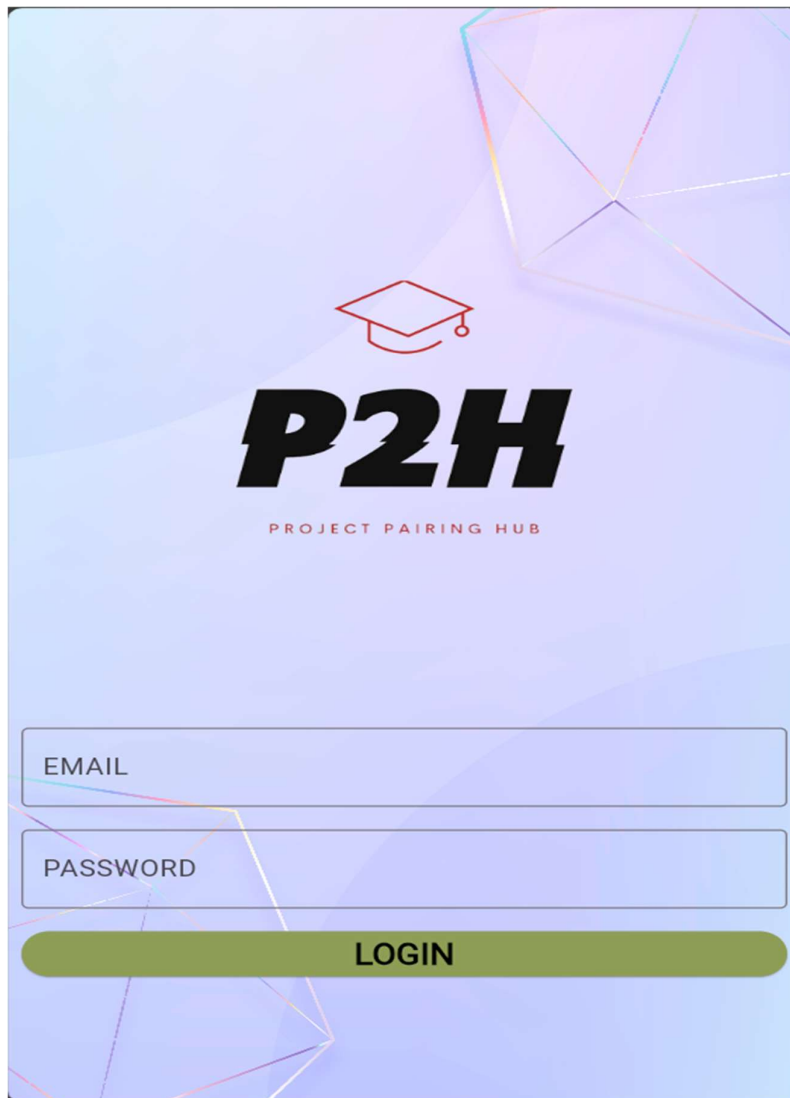
```
> main.dart library package:jawa_project/LoginPage.da
1 import
2 import package:jawa_project/LoginPage.dart
3 import 'package:jawa_project/LoginPage.dart';
4
Run | Debug | Profile
5 void main() {
6   runApp(const MyApp());
7 }
8
9 class MyApp extends StatelessWidget {
10   const MyApp({super.key});
11   @override
12   Widget build(BuildContext context) {
13     return GetMaterialApp(
14       debugShowCheckedModeBanner: false,
15       home: const LoginPage(),
16     ); // GetMaterialApp
17   }
18 }
19
```

ii)pubspec.yaml

```
! pubspec.yaml
1  name: java_project
2  description: "A new Flutter project."
3  publish_to: 'none'
4  version: 1.0.0+1
5
6  environment:
7    sdk: '>=3.2.5 <4.0.0'
8
9  dependencies:
10   flutter:
11     sdk: flutter
12
13   cupertino_icons: ^1.0.2
14   get: ^4.6.6
15   fluttertoast: ^8.2.4
16
17
18  dev_dependencies:
19   flutter_test:
20     sdk: flutter
21
22   flutter_lints: ^2.0.0
23
24
25
26  uses-material-design: true
27
28  assets:
29    - assets/AI.jpeg
30    - assets/ML.jpeg
31    - assets/ARVR.jpeg
32    - assets/BG1.jpg
33    - assets/BG2.jpg
34    - assets/logo.png
35    - assets/R.png
36    - assets/men.jpg
37    - assets/women.jpeg
38    - assets/BR.png
39    - assets/iot.png
40    - assets/mech.png
41    - assets/biotech.jpg
42    - assets/Mar.png
43    - assets/ee.jpg
```

B.SCREEN SHOTS

i)Login page



The login page features a light blue background with a subtle geometric pattern of colorful lines. At the top center is a red graduation cap icon. Below it, the text "P2H" is displayed in a large, bold, black font. Underneath "P2H", the words "PROJECT PAIRING HUB" are written in a smaller, red, sans-serif font. The login form consists of two white rectangular input fields with thin black borders. The first field is labeled "EMAIL" in a small, black, sans-serif font. The second field is labeled "PASSWORD" in a small, black, sans-serif font. Below these fields is a green button with rounded corners, labeled "LOGIN" in a bold, black, sans-serif font.



P2H

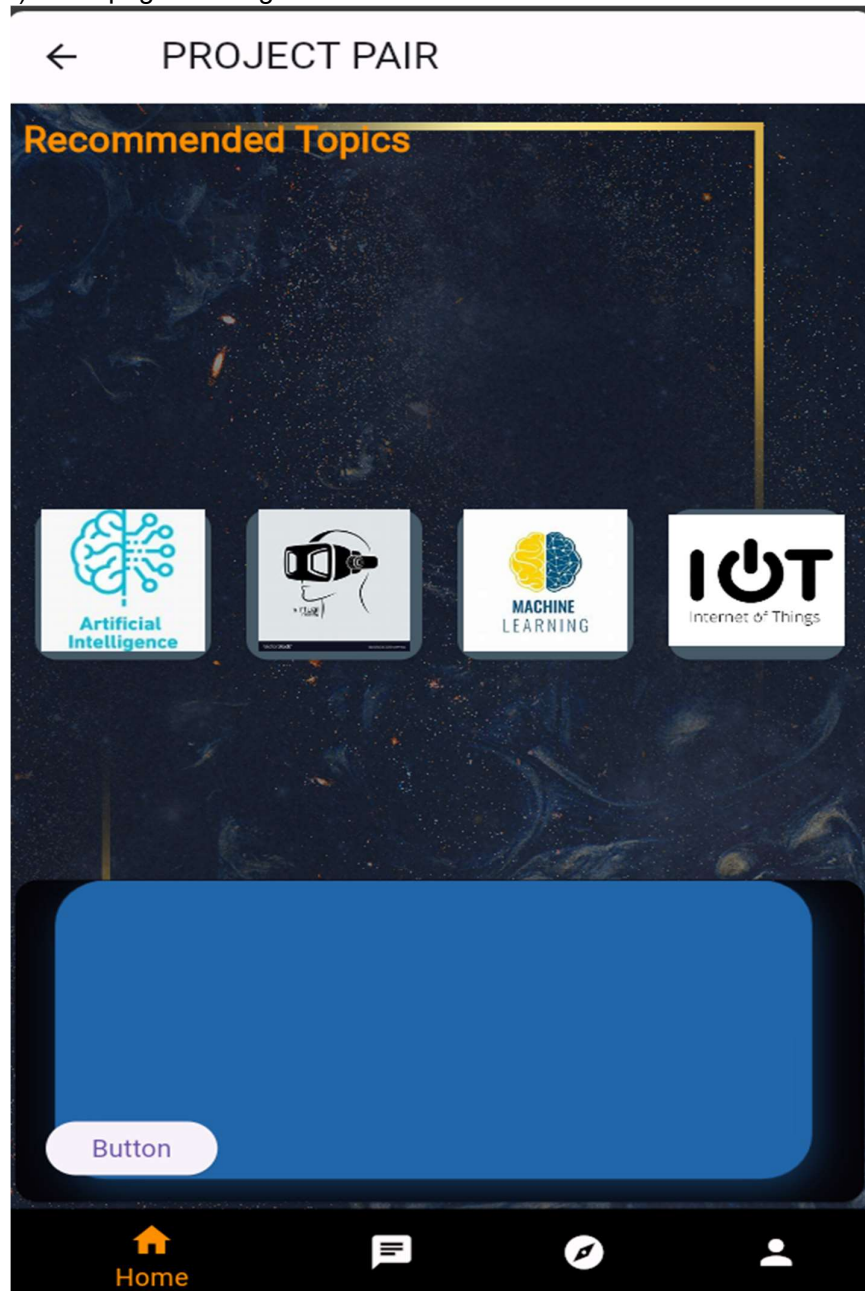
PROJECT PAIRING HUB

EMAIL

PASSWORD

LOGIN

ii) Homepage & Navigation bar



iii)Select & Chat with Mentors

