

COURSE HUB

INTERDISCIPLINARY PROJECT

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

by

KEERTHANA ANUSHA.C (Reg.No–41120093)



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA

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

CATEGORY-1 UNIVERSITY BY UGC

Accredited “A++” by NAAC

12B Status by UGC |Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI-600119

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 **KEERTHANA ANUSHA C(41120093)** who carried out the project entitled "**Course 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.SUBASHINI, M.E., Ph.D.,

Submitted for Viva voce Examination held on 08.05.24

Internal Examiner
External Examiner

DECLARATION

I Keerthana anusha hereby declare that the Project Report entitled "Course 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


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.SUBASHINI M.E.,Ph.D.**,Head of the Department Of the 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 **Keerthana Anusha C (Reg No: 41120093)** 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, **Keerthana Anusha C** has closely worked as a part of the **Flutter** by carrying out the project entitled "**Course 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

Course Hub is a cutting-edge educational platform designed to revolutionize the way we learn. With a vast array of courses spanning diverse subjects, the app caters to learners of all levels. Its intuitive interface empowers users to seamlessly explore, enroll, and navigate through courses, fostering a user-centric educational experience. Adaptive learning algorithms assess individual progress, tailoring content delivery for optimal comprehension. Social integration adds a collaborative dimension, enabling users to connect, share insights, and engage in discussions. Regular updates and curated content ensure courses remain current and captivating. Advanced analytics offer deep insights into learning patterns, empowering users to track and optimize their educational journey. The platform is designed for accessibility, ensuring a user-friendly experience across various devices. Course Hub emerges as a dynamic force, redefining education by blending innovation, connectivity, and personalized learning. As a beacon for the modern learner, Course Hub strives to make education accessible, interactive, and adaptable. Join the educational evolution and embark on a transformative learning journey with Course Hub, where knowledge meets the future."

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 System Architecture	12
	2.3 Flow Diagram	16
	2.4 Modules	17
	2.4.1 User Management	17
	2.4.2 Authentication Management	17
	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
3		RESULTS AND DISCUSSION	24
4		CONCLUSION AND FUTURE WORK	29
	4.1	Conclusion	29
	4.2	Future Work	30
		REFERENCES	32
		APPENDICES	35
	A.	Sample Code	35
	B.	Screen Shots	38

LIST OF ABBREVIATIONS

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

LIST OF FIGURES

Figure.	Name of the Figure	Page
No.		No.
2.1	Proposed Application Architecture	12
2.2	Firebase Architecture	14
2.3	Flow Diagram	16
2.4	Firebase Connection	22
2.5	Firebase Authentication	22
2.6	Flutter Run	24
3.1	firebase configuration	26
3.2	Dashboard Screen Display	27
3.3	Initializing dart code	28

CHAPTER 1

INTRODUCTION

Course hub is a digital platform designed to streamline and centralize educational resources for a specific academic course or program. It acts as a virtual hub where students and instructors can access course materials, assignments, and supplementary resources in one convenient location. These hubs often incorporate collaborative tools like discussion forums and messaging systems to facilitate communication among students and instructors. That is course hub encompasses several key components that collectively optimize the learning experience for students and instructors. At its core, it features a repository for course materials, offering a centralized space for lecture notes, slides, readings, and multimedia content. Assignments and assessments find a dedicated section with functionalities for submission and grading. The inclusion of discussion forums or message boards fosters collaborative communication among students and between students and instructors. Important updates and announcements are consolidated in a designated space, while tools for progress tracking and grade visibility provide transparency and accountability. Additionally, a calendar and schedule feature aids in organizing crucial dates and deadlines. User profiles offer personalized snapshots of individual progress and engagement. Ensuring accessibility for all, the hub incorporates features like alternative text and compatibility with assistive technologies. Mobile compatibility further enhances accessibility, allowing students to engage conveniently via smartphones or tablets

1.1 FLUTTER

Flutter, a robust cross-platform UI toolkit developed by Google, is at the heart of Men Course 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 Course 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, Course 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:

Course 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 Course 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 Course 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 Course 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 Course 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. CourseHub 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.

Course 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. Course Hub utilizes this feature to efficiently manage and serve multimedia.

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.

The CourseHub 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. The Course 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 Course 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. Course 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.

iii)Community Engagement:

Encouraging active participation from both students and mentors in discussions, answering queries, and contributing to the community.

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 Course 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 Course Hub encompass various aspects such as functionalities, technologies, and packages.

i)Project Scope:

- 1) Develop a mobile application named "Course 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 jobrelated 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

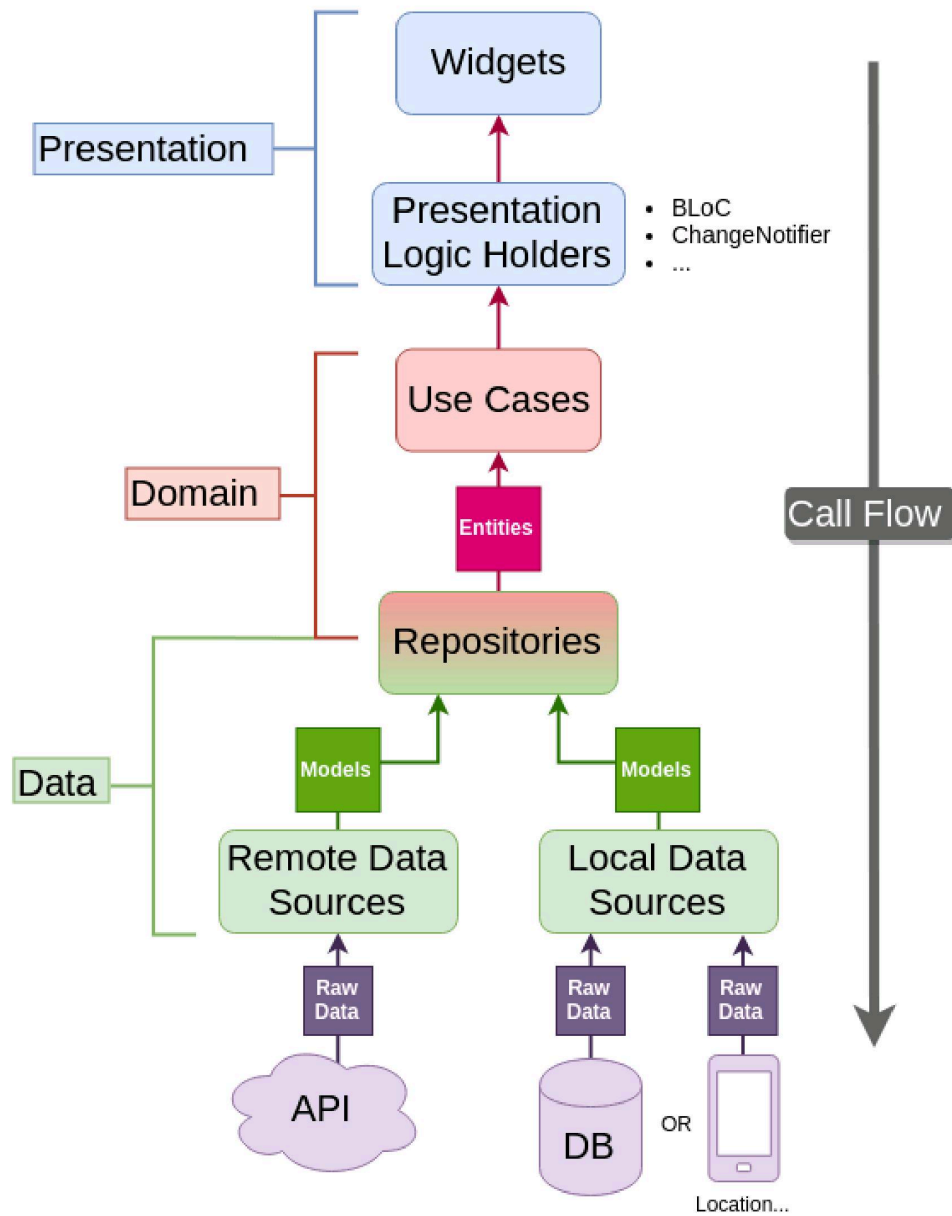


Fig:2.1: Proposed Application Architecture

1.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. The custom Dart code interacts with the Flutter framework to achieve various tasks.

2.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.

3.MethodChannel:

The MethodChannel allows communication between Dart code and native platform code (Android or iOS). When we invoke a method on the MethodChannel from Dart, it triggers a corresponding method call on the native side.

4.BinaryMessages:

BinaryMessages is another channel for communication between Dart and native code. It enables the exchange of binary data (such as images, files, or serialized objects) between the two sides.

5.Flutter Engine Android Shell:

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

- i)FlutterView: This is the Android view that displays your Flutter UI.
- ii)BinaryMessenger: It handles communication between the Flutter engine and the Android platform.

6.Flutter Engine iOS Shell:

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

i)FlutterViewController: The iOS view controller responsible for displaying your Flutter UI.

ii)FlutterBinaryMessenger: Handles communication between the Flutter engine and the iOS platform.

7.Custom Android Code and Custom iOS Code:

These represent platform-specific code written in Java/Kotlin (for Android) and Objective-C/Swift (for iOS).You'll write custom code here to interact with native features or APIs that aren't directly accessible from Flutter.

8.Firebase:

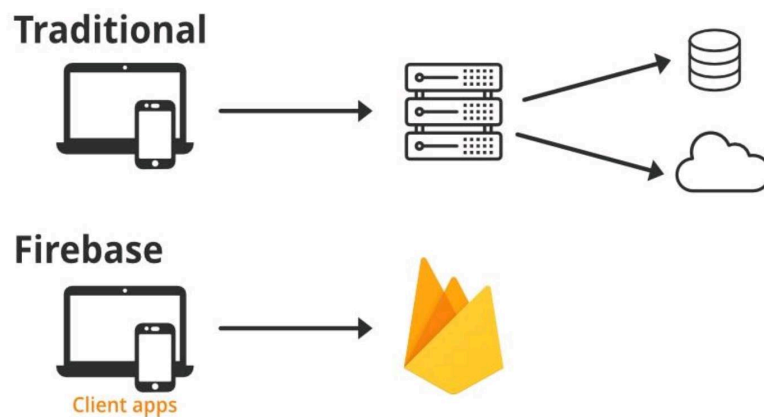


Fig:2.2: Firebase Architecture

i)Firebase Authentication:

Firebase Authentication offers ready-to-use authentication services. It integrates seamlessly with Flutter, allowing users to sign up, sign in, and manage their accounts. MFA (Multi-factor Authentication) can be enabled to enhance security by requiring additional verification steps (e.g., SMS codes, email, or biometrics).

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.

iii)Firebase Storage:

Firebase Storage provides secure cloud storage for files (e.g., images, videos, documents). You can upload and download files directly from your Flutter app. Data-at-rest encryption ensures sensitive files are protected.

iv)Firebase Analytics:

Firebase Analytics tracks user behavior, app usage, and events. You can gain insights into user engagement, retention, and conversion rates. Use analytics data to make informed decisions about app improvements.

9.To integrate Firebase with your 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). Use Flutter widgets to build UI that interact with Firebase services.

2.3 FLOW DIAGRAM

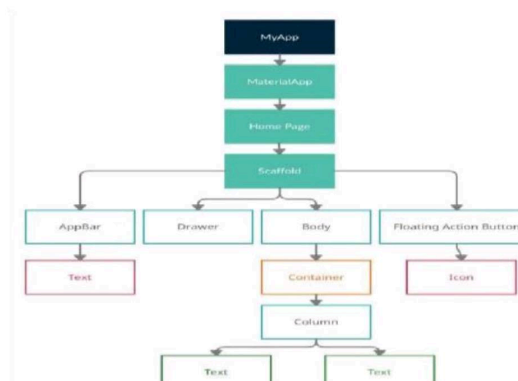


Fig:2.3: Flow Diagram

The following steps are used to describe the above 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.

Algorithm:

i)Display Registration Form:The signup process begins by displaying a user-friendly registration(Signup) form on the Course Hub platform. The form includes fields for personal information, contact details, and interests, ensuring a comprehensive profile setup.

ii)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.

iii)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).

iv)Create User Account: After successful validation, the system creates a new user account using the provided information user ID

2.4.2 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 Course Hub's 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.

2.4.3 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 CourseHub, 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.

2.4.4 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.

2.4.5 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.

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.

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. Dependency Management:

Project dependencies specified in the pubspec.yaml file.

Key packages include:

- firebase_core for Firebase integration.
- firebase_auth for user authentication.
- cloud_firestore for real-time data synchronization.
- Other packages like get, toast, image_picker, etc., for additional functionalities.

3. Firebase Configuration:

Firebase project created and configured for the application.

Firebase Hosting used for deploying the Flutter web application.

4. Firebase Authentication:

Firebase Authentication set up for secure user login and identity verification.

5.Development Tools:

Flutter commands used for testing and building the application.Realtime 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: This file likely contains the UI and logic for user login. It includes form fields for entering credentials and a mechanism to authenticate user

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: 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.

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: This file represents a page where mentors are listed 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: This file represents a feedback page where users can provide 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:

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.

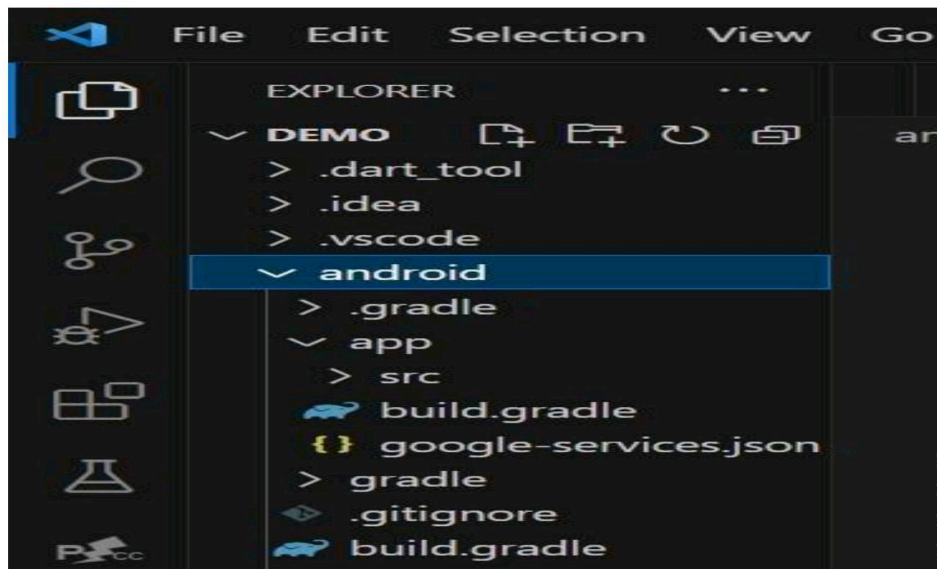


Fig:3.5: Firebase Configurations

iii)Add Firebase Dependencies:

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

```
yaml code: dependencies:  firebase_core:^2.25.3  
firebase_auth: ^4.17.5
```

iv)Initialize Firebase:

In your main.dart or wherever you initialize your Flutter app, make sure to initialize Firebase. If you want to implement user authentication, Firebase provides the firebase_auth package. You can use it to sign in users using various methods like email/password, Google Sign-In, etc.

Dart code:

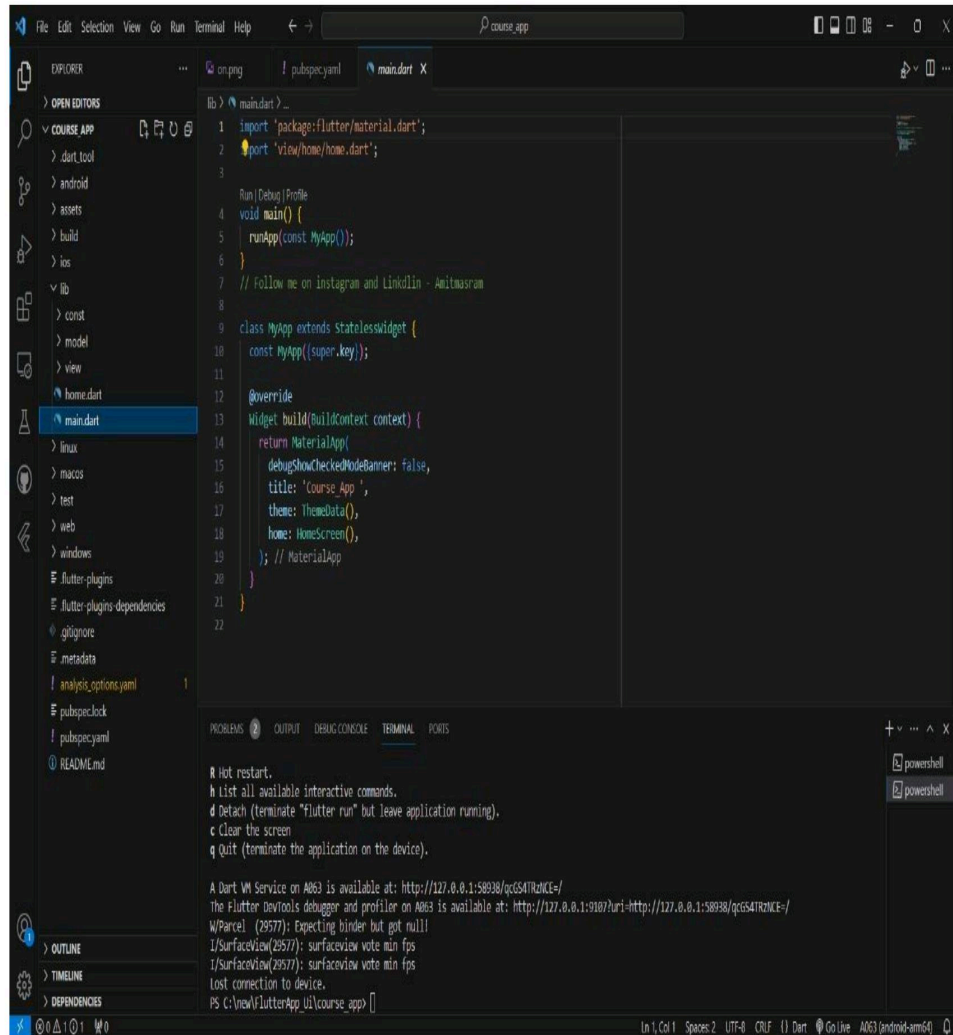


Fig:3.6: Initializing dartcode

CHAPTER 4

CONCLUSION AND FUTURE WORK

4.1. CONCLUSION

In conclusion, Course 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.

Course 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, Course 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 openended 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 Course 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 Course Hub comfortably.

REFERENCES

1.Journals:

Journal of Online Learning and Teaching:

This peer-reviewed journal publishes research articles, case studies, and reviews related to online learning and teaching, including topics such as instructional design, technology integration, and learning analytics.

Computers & Education:

As mentioned earlier, this journal covers research on the use of technology in education, including articles on online learning platforms, educational software, and instructional design.

Educational Technology Research and Development:

This journal publishes research on educational technology, including studies on the design, development, and evaluation of technologyenhanced learning environments.

British Journal of Educational Technology:

A leading journal in the field of educational technology, publishing research on the design, implementation, and evaluation of educational technology tools and interventions.

International Review of Research in Open and Distributed Learning (IRRODL):

IRRODL is an open-access journal that publishes research on open and distance learning, including articles on online course design, pedagogy, and technology integration.

2. Books:

E-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning" by Ruth C. Clark and Richard E. Mayer:

This book provides evidence-based guidelines for designing effective e-learning experiences, covering principles of multimedia learning and instructional design.

"Design for How People Learn" by Julie Dirksen:

Dirksen's book offers practical insights into designing engaging and effective learning experiences, incorporating principles from cognitive science and user experience design.

"The Online Teaching Survival Guide: Simple and Practical Pedagogical Tips" by Judith V. Boettcher and Rita-Marie Conrad:

This guide offers practical strategies for designing and facilitating online courses, covering topics such as course design, communication, assessment, and technology integration. "Interaction

Design: Beyond Human-Computer Interaction" by Jenny Preece, Yvonne Rogers, and Helen Sharp: A comprehensive resource on interaction design, covering principles, methods, and techniques for designing interactive systems, including websites and educational software.

"Learning Management System Technologies and Software Solutions for Online Teaching:

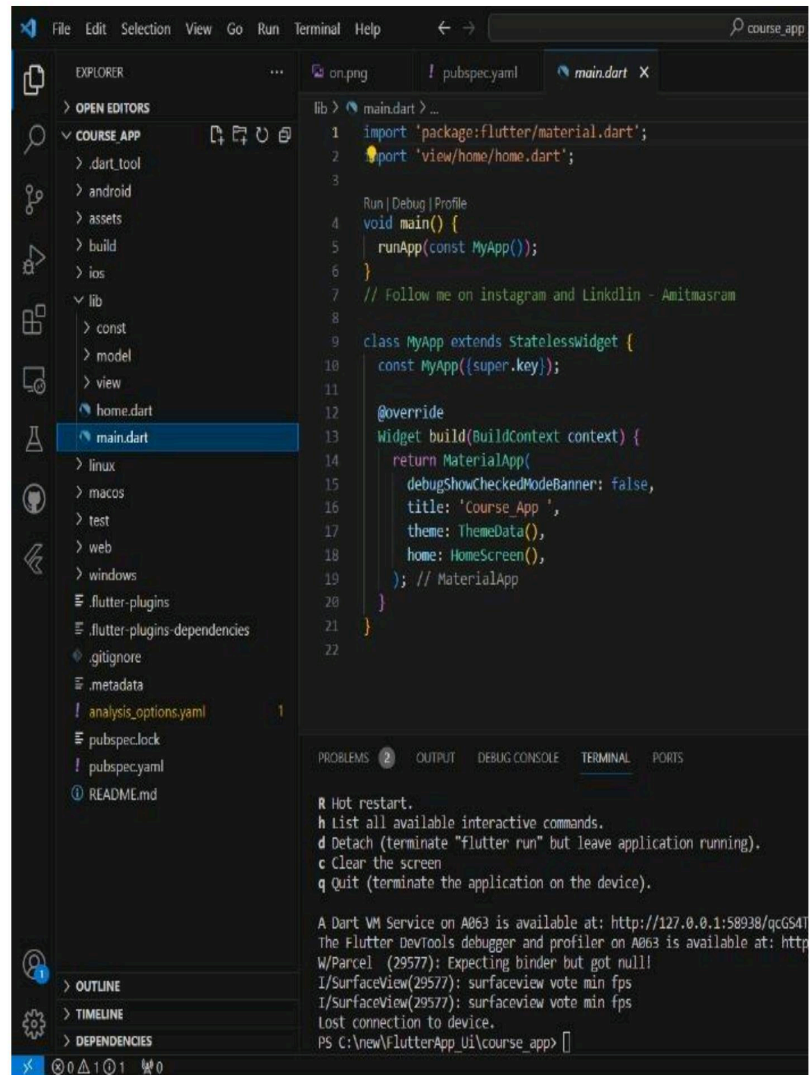
Tools and Applications" edited by Lawrence A. Tomei:

This book explores various learning management systems (LMS) and software solutions for online teaching, providing insights into features, functionalities, and best practices.

APPENDICES

A.SAMPLE CODE

1)Main Code:



The screenshot shows an IDE interface with the following components:

- EXPLORER:** A file tree on the left showing the project structure. The `main.dart` file is selected under the `lib` directory.
- EDITOR:** The main area displays the content of `main.dart`. The code includes imports for Flutter material and home views, a `main` function that runs `MyApp`, and the `MyApp` class which extends `StatelessWidget` and implements the `build` method to return a `MaterialApp` widget.
- TERMINAL:** A terminal window at the bottom right shows the output of a hot restart. It includes instructions for interactive commands (R, h, d, c, q) and logs from the Dart VM Service and Flutter DevTools.

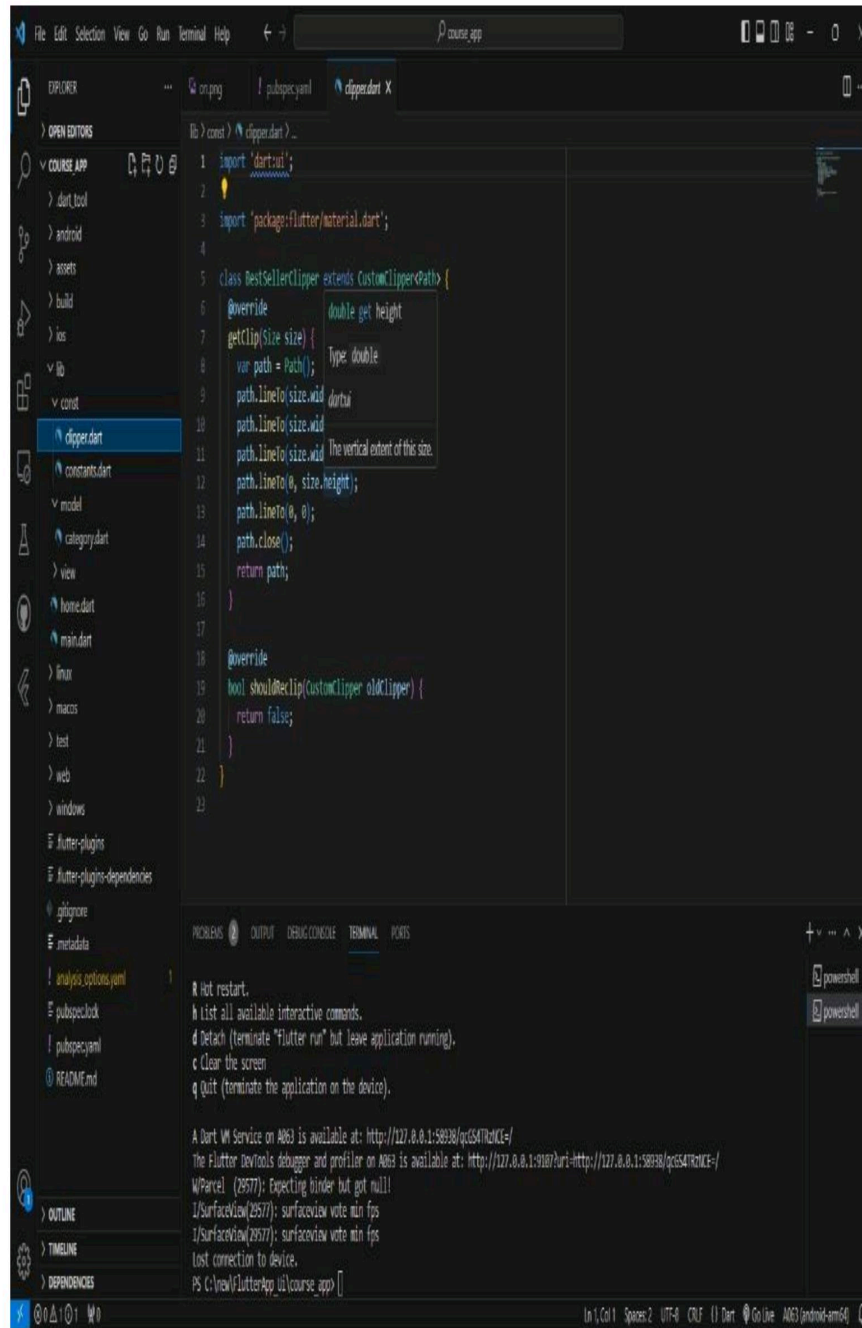
```
lib > main.dart > ...
1 import 'package:flutter/material.dart';
2 import 'view/home/home.dart';
3
4 void main() {
5   runApp(const MyApp());
6 }
7 // Follow me on instagram and Linkdin - Amitmasram
8
9 class MyApp extends StatelessWidget {
10   const MyApp({super.key});
11
12   @override
13   Widget build(BuildContext context) {
14     return MaterialApp(
15       debugShowCheckedModeBanner: false,
16       title: 'Course_App ',
17       theme: ThemeData(),
18       home: HomeScreen(),
19     ); // MaterialApp
20   }
21 }
22
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

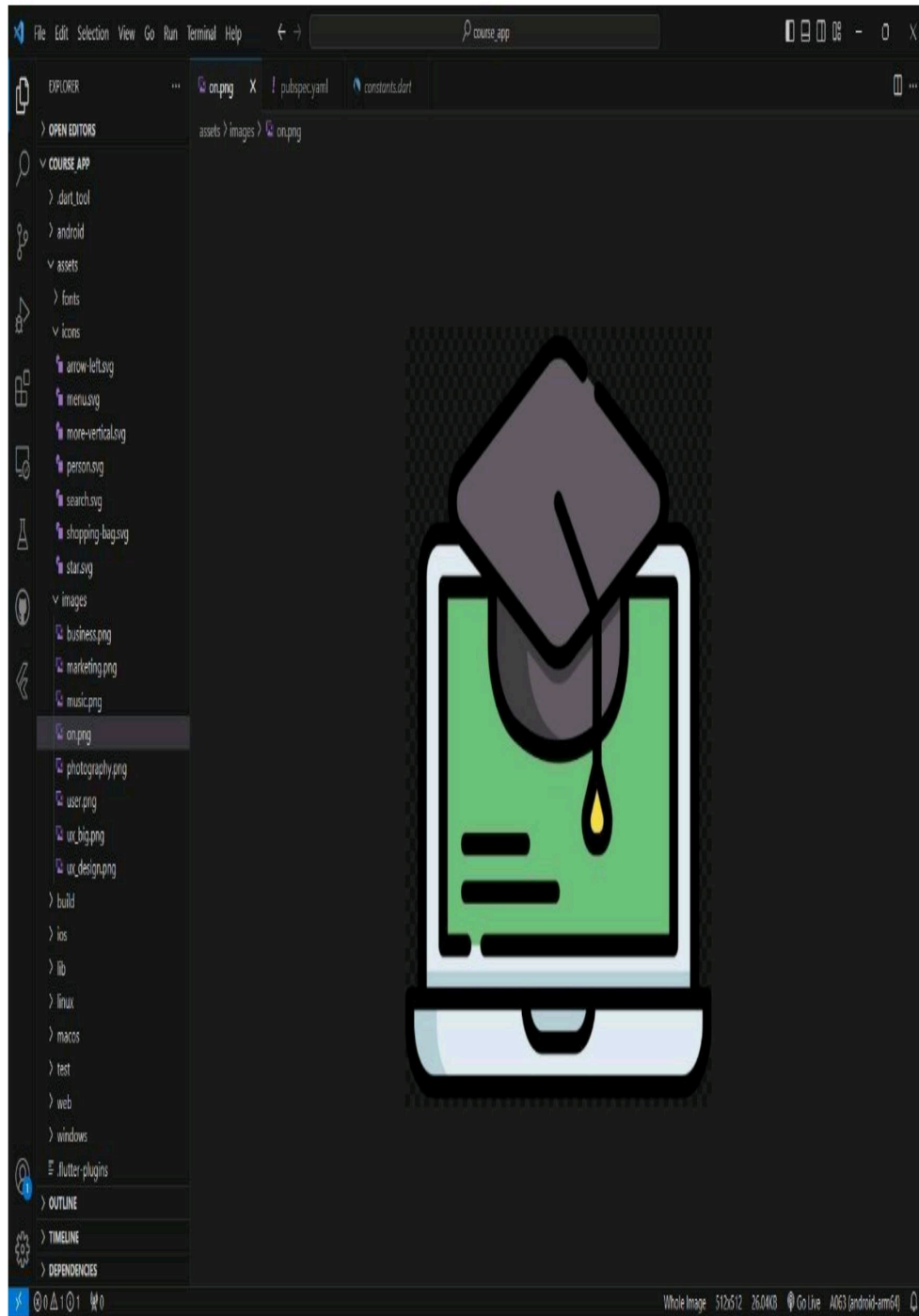
R Hot restart.
h list all available interactive commands.
d Detach (terminate "flutter run" but leave application running).
c Clear the screen
q Quit (terminate the application on the device).

A Dart VM Service on A063 is available at: http://127.0.0.1:58938/qcGS4T
The Flutter DevTools debugger and profiler on A063 is available at: http://127.0.0.1:58938/?url=flutter://A063
W/Parcel (29577): Expecting binder but got null!
I/SurfaceView(29577): surfaceview vote min fps
I/SurfaceView(29577): surfaceview vote min fps
lost connection to device.
PS C:\new\FlutterApp_UI\course_app>

2)Clipper..dart:

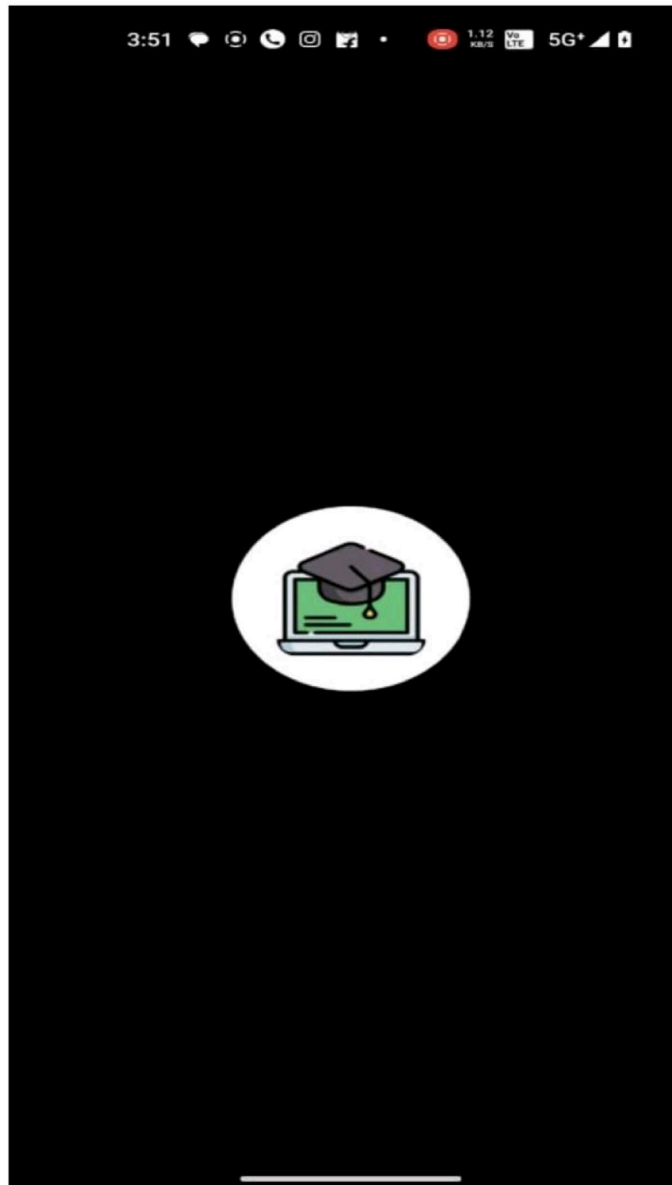


3)on.png:

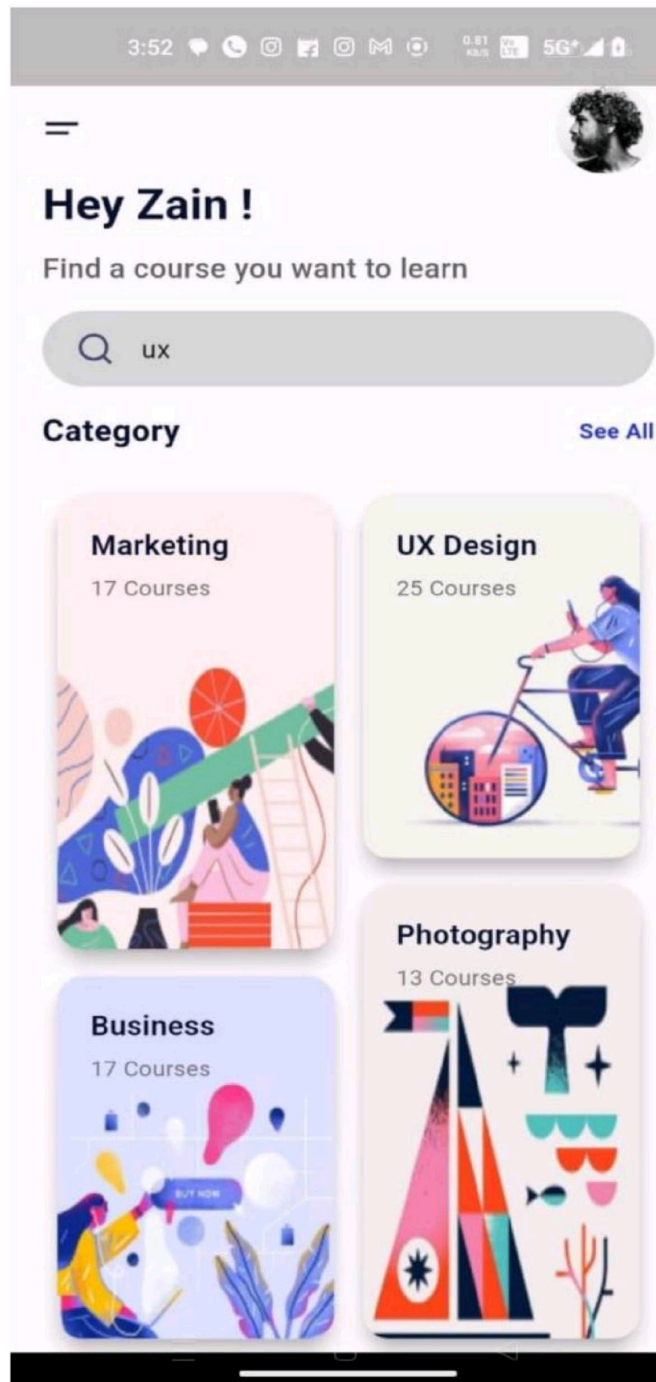


B.SCREEN SHOTS

i) App icon:



2)home page:



3) Navigation Bar:

