

ACTIVE LIFE (FITNESS-APP)

INTERDISCIPLINARY PROJECT

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

by

KAMALESH K (41120083)



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 **KAMALESH K (41120083)** who carried out the project entitled "Active life (fitness app)" 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 KAMALESH K hereby declare that the Project Report entitled "ACTIVE LIFE (FITNESS-APP)" 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

Kamalesh.K

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

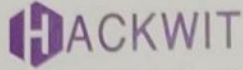
I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

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 **Kamalesh K (Reg No: 41120083)** 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, **Kamalesh K** has closely worked as a part of the **Flutter** by carrying out the project entitled "**Active Life (FITNESS-APP)**"

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

Active Life: A Comprehensive Fitness and Nutrition Tracking Application Active Life is a versatile mobile application designed to provide users with holistic support for their fitness and nutrition goals. The app offers a multifaceted approach, integrating workout videos, calorie tracking, body mass calculation, and administrative functionalities. Users can access a diverse collection of workout videos targeting various body parts, facilitating comprehensive fitness routines. Each section of the body offers a curated selection of exercises to cater to individual preferences and fitness levels. Through seamless integration with Firebase database, users can effortlessly log and monitor their daily food intake, enabling them to maintain a balanced diet. The app supports CRUD operations, empowering users to manage their calorie details with ease. Utilizing inputs such as weight, height, and gender, the app calculates Body Mass Index (BMI) values, providing users with valuable insights into their health and fitness status. This feature aids users in setting realistic goals and tracking their progress effectively. The application includes an administrative interface where users can view their registered email IDs, enhancing account management and security. Active Life aims to empower users in their journey towards a healthier lifestyle by offering comprehensive tools for fitness tracking

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 Calorie Tracking	19
	2.4.4 Workout Section	20
	2.4.5 BMI Calculator	21
	2.5 Development and Deployment Setup	22

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

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	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	HomePage and BMI Page	27
3.3	Profile Page	27
3.4	Calorie Tracker and EditData Page	28
3.5	Workout Home and Workout Page	28
3.6	Firebase Configuration	29
3.7	Initializing Firebase	30

CHAPTER 1

INTRODUCTION

Active Life, your comprehensive companion on the journey to a healthier, fitter you. Active Life is not just another fitness app; it's a holistic approach to fitness and nutrition, seamlessly integrating cutting-edge technology with user-friendly features to empower individuals in their pursuit of wellness. Active Life is more than just a workout app; it's a personalized fitness and nutrition hub designed to cater to the diverse needs and preferences of users. Whether you're a seasoned fitness enthusiast or just embarking on your wellness journey. With a diverse range of features, Active Life caters to every aspect of your fitness journey. From a comprehensive workout video library featuring exercises for every muscle group, to seamless calorie tracking and intuitive body mass calculation, Active Life offers a one-stop solution for all your fitness and nutrition needs. At the heart of Active Life is the belief that fitness should be accessible to everyone, regardless of age, gender, or fitness level. That's why we've designed Active Life to be user-friendly and inclusive, with features that empower users to take control of their health and well-being. Join the Active Life community today and embark on a journey to a healthier, happier you. Whether you're looking to build muscle, lose weight, or simply improve your overall fitness, Active Life is here to support you every step of the way

1.1.FLUTTER

Flutter, a robust cross-platform UI toolkit developed by Google, is at the heart of Active Life 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 Active Life 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, Active Life 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

Active Life 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 Active Life 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 Active Life. 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 Active Life. 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 Active Life, 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. Active Life 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. Active Life 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. Active Life 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

The aim of "Active Life" is to revolutionize the way individuals approach fitness and nutrition by providing a comprehensive platform that seamlessly integrates essential functionalities into a single, user-friendly interface. By offering features such as workout video libraries, calorie tracking, body mass calculation, and community support, "Active Life" empowers users to take control of their health and well-being with ease. Through intuitive design and robust technology, the app aims to streamline the user experience, making it simple and enjoyable for users to track their progress, set achievable goals, and stay motivated on their fitness journey

The modern lifestyle often leads to sedentary habits and poor dietary choices, contributing to a rise in health issues such as obesity, cardiovascular diseases, and lifestyle-related conditions. Many individuals struggle to maintain a balanced diet, engage in regular physical activity, and monitor their overall health due to various constraints such as lack of time, knowledge, or motivation.

Additionally, accessing personalized fitness guidance and resources tailored to individual needs and preferences can be challenging. There is a need for a comprehensive wellness platform that empowers users to track their dietary habits, access guided workout routines, monitor their health metrics, and engage with a supportive community to foster accountability and motivation.

Thus, the problem statement revolves around addressing these challenges by developing an intuitive and user-friendly app, "Active Life," that promotes healthy lifestyle choices and supports users in achieving their fitness and wellness goal

In today's fast-paced world, maintaining a healthy lifestyle amidst hectic schedules and tempting indulgences poses a significant challenge for many individuals. Sedentary habits, coupled with poor dietary choices, have led to a surge in health issues like obesity and cardiovascular diseases. Despite the growing awareness of the importance of fitness and nutrition, accessing personalized guidance and resources tailored to individual needs remains a daunting task.

The development of the "Active Life" app aims to address these issues by offering an intuitive and user-friendly solution that empowers users to make informed decisions about their health and embark on a path towards a healthier and happier lifestyle.

Recognizing the imperative for a holistic approach to wellness, encompassing physical activity, nutrition, mental health, and social support, there emerges a critical need for innovative solutions. The "Active Life" app endeavors to address these pressing challenges by offering a comprehensive platform that empowers users to make informed decisions about their health, foster sustainable lifestyle changes, and cultivate a community of support and encouragement for long-term well-being.

Challenges abound, from limited time for exercise and meal planning to a lack of motivation and accountability in adhering to healthier habits. Moreover, disparities in access to nutritious foods and fitness facilities, compounded by socioeconomic factors and cultural norms, perpetuate health inequities across communities.

1.5. OBJECTIVES

The objectives of "Active Life" are to develop a user-friendly platform that seamlessly integrates essential fitness and nutrition tracking features, including workout video libraries, calorie tracking, and body mass calculation. Through intuitive design and robust technology, the app aims to streamline user interaction, enhancing user experience and satisfaction. Additionally, "Active Life" seeks to foster a supportive community by implementing social features such as user profiles, progress sharing, and community forums, encouraging engagement, motivation, and support among users. The app will prioritize cross-platform compatibility, data security, and performance optimization to ensure a consistent and reliable experience across iOS and Android devices while safeguarding user privacy. By gathering user feedback and iterating on features

Key Challenges

Integration of Multiple Features

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

User Authentication and Security:

Implementing robust user authentication mechanisms using Firebase while ensuring the security and privacy of user data poses a critical challenge. Addressing potential vulnerabilities and ensuring compliance with data protection regulations is essential.

Cross-Platform Compatibility:

Ensuring consistent performance and user experience across iOS,

Android, and web platforms using Flutter, including optimization, The impact extends beyond physical health, with mental well-being also at stake as stress and anxiety mount in tandem with unhealthy lifestyles.platform-specific nuances, and feature parity.

Real-time Communication and Updates:

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.

Scalability and Performance

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

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 Active Life encompass various aspects such as functionalities, technologies, and packages.

i)Project Scope

- 1) Develop a mobile application named " Active Life".
- 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 secure user authentication mechanisms such as signup, login, and password reset functionalities to ensure data privacy and account security.
- 2) Calorie Tracking: Users can log their daily food intake and track calorie consumption to maintain a balanced diet and achieve their health goals.
- 3) BMI Calculator: Provide a dedicated section for job-related queries, interview preparation, and industry-specific challenges.
- 4) Progress Tracking: Enable users to track their progress towards fitness goals by visualizing calorie intake, workout history, BMI changes, and other relevant metrics over time.
- 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.

2.2.PROPOSED APPLICATION ARCHITECTURE

1.Flutter

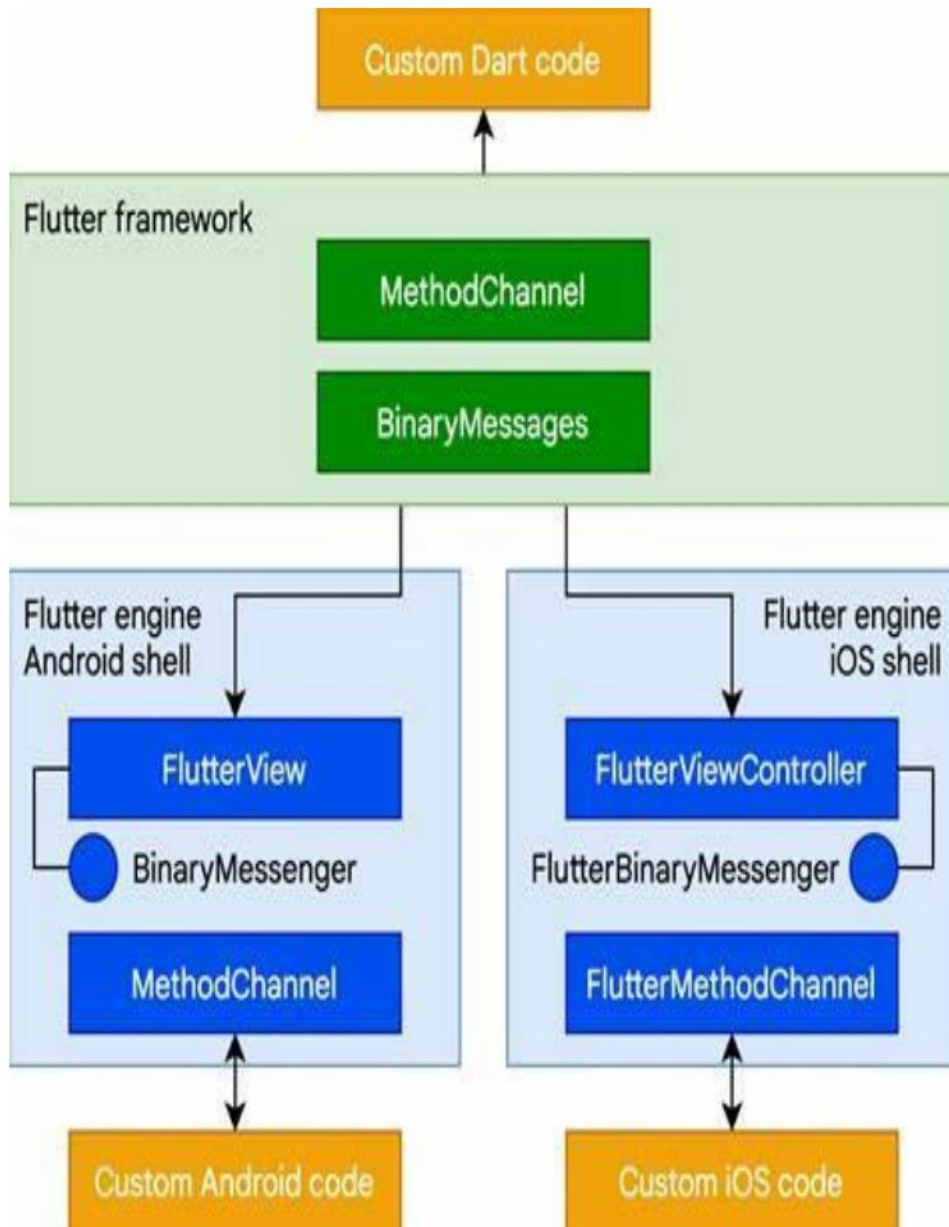


Fig:2.1: Flutter Architecture

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

Firebase 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).

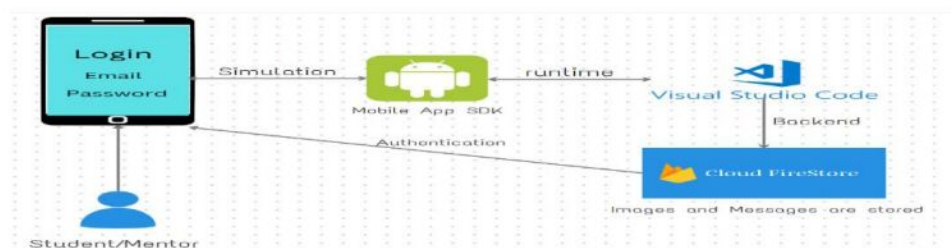


Fig:2.3: Proposed Application Architecture

i)User Interaction

Users (represented by the “user” 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

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

2.3.DATA FLOW DIAGRAM

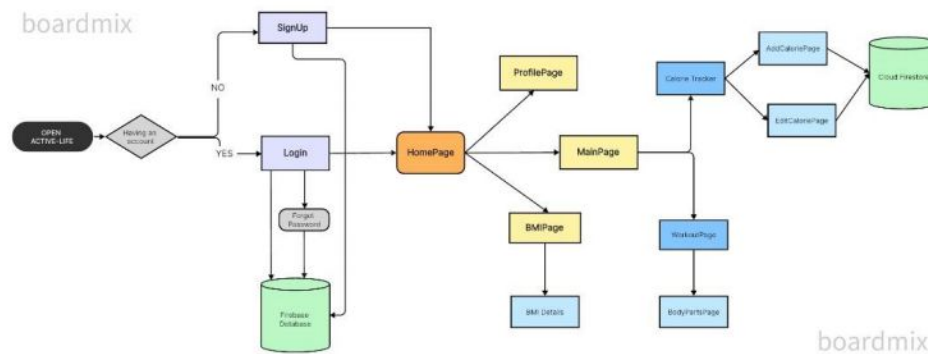


Fig:2.4: Data Flow Diagram

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) If the user forgot the password enter the registered email id in the forgot password page user can receive the forgot password link in the Gmail
- vi) Once the user is logged in, they will be taken to the home page.
- vii) From the home page, the user can access various features of the app such as their profile page, BMIPage, Calorie Tracker page, Workout page, and details page.
- viii) The Calorie Tracker page allows to user note the daily food they consumed it saved in cloud firestore database.
- ix) The Workout page contains all body parts workouts in categorized in manner
- x) The BMI page fetch the users basic information and show the bmi details.

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.CALORIE TRACKING

Calorie tracking in your app involves several key functionalities to help users monitor their daily food intake and manage their calorie consumption effectively

Functionalities

i)Calories tracking: The calorie tracking functionality in the app allows users to meticulously monitor their dietary habits and manage their calorie intake with precision. Users can effortlessly log their meals by entering details such as food items consumed, portion sizes, and meal times.

ii) Nutritional Information: Comprehensive nutritional information is provided for each logged food item, including calorie count, macronutrients, and micronutrients. Users can log their meals by entering details such as food items consumed, portion sizes, and meal times.

iii) Progress Tracking: Users can track their progress towards calorie goals over time, with insights and motivation provided for achieving desired outcomes. platform, allowing them to explore Active Life features and functionalities.

Algorithm

1. Design a calorie tracking page using a Scaffold widget.
2. Using Firebase database to add a entry in database
3. Click the edit calorie button to edit the Calorie entry in the database
4. In the main calorie Tracking page we can edit and delete the calories

2.4.4.WORKOUT SECTION

The workout clips section of the app offers users a dynamic and versatile resource for their fitness journey. With a diverse library of exercise videos spanning various fitness levels, interests, and goals, users can easily find workouts that suit their preferences and needs.

Functionalities

i)Video Demonstrations: Each exercise includes a video demonstration or animation illustrating proper form and technique for safe and effective execution. Users can create personalized workout routines by selecting individual exercises and setting parameters such as repetitions, sets, and rest intervals.

ii) Workout Tracking Users can log completed workouts, track progress, and record performance metrics such as weights lifted, repetitions performed, and duration of workouts. .

iii) Categorized Workouts: Exercises are organized into categories such as strength training, cardio, flexibility, and targeted muscle groups for easy browsing.

Algorithm

1. Create a workout page to display a workout clips using a Scaffold widget.
2. Leads to body parts are categorized into cards
3. In Each Categorize 10 plus fitness video tutorials
4. Inside using video player dependencies clips will be played minimum 10sec clips and all videos are available in offline mode

2.4.5. BMI CALCULATOR:

The BMI (Body Mass Index) calculator feature in the app offers users a valuable tool for assessing their overall health status based on their weight and height. By inputting these basic measurements, users can quickly obtain their BMI score, which provides an indication of whether their weight falls within a healthy range relative to their height.

Functionalities

i)Input Parameters: Users can input their weight and height measurements into the BMI calculator. The calculator computes the BMI score based on the provided weight and height values.

ii) Integration: The BMI calculator may integrate with other features of the app, such as workout tracking and calorie monitoring, to provide a comprehensive health assessment. The app classifies the BMI score into categories such as underweight, normal weight, overweight, or obese.

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

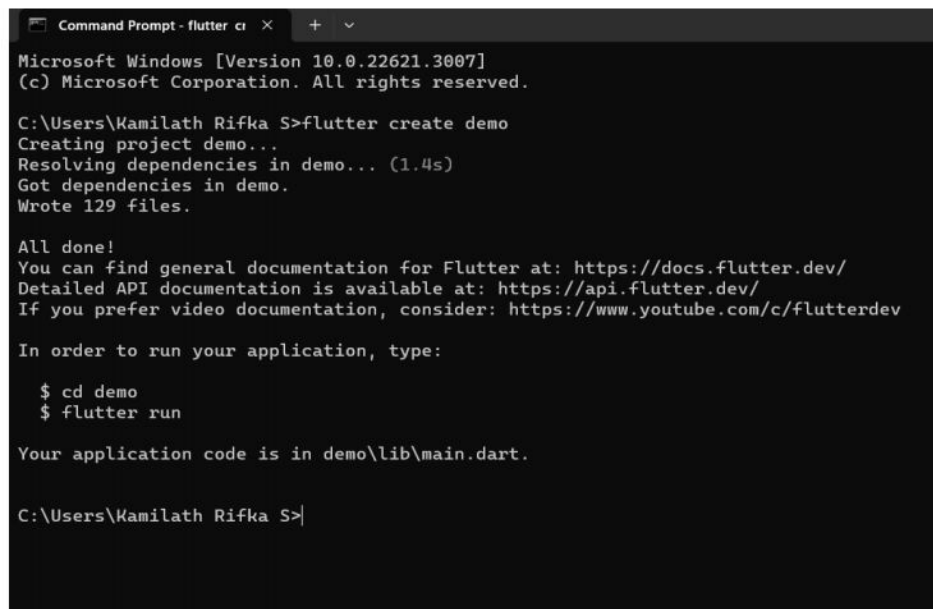
1. Implement BMI page for calculate to check body state or normal, obesity or weak
2. First fetch the information form the user like height ,weight ,age And gender
3. After enter the all details enter submit button the bmi formula calculate the bmi value and we can see the bmi ranges in bmi details in BMI Details page.

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:

- 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.

4.Firebase Configuration

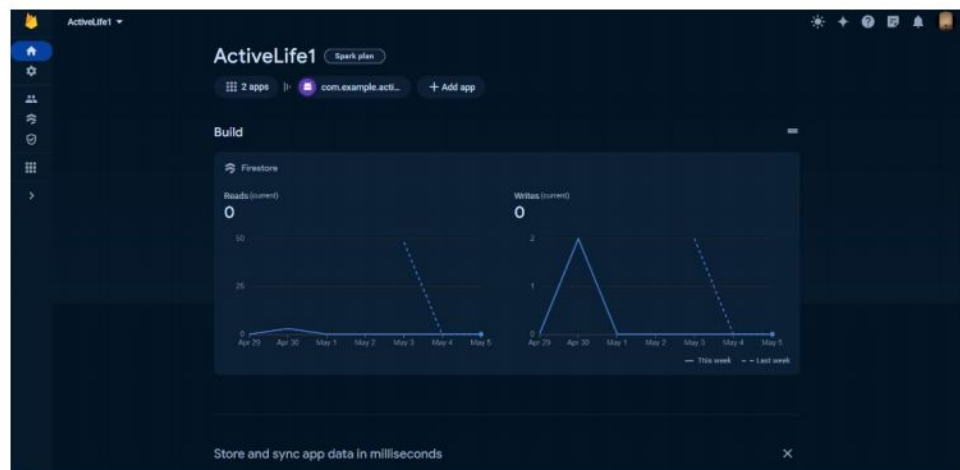


Fig:2.6: Firebase Connection

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

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

5. Firebase Authentication

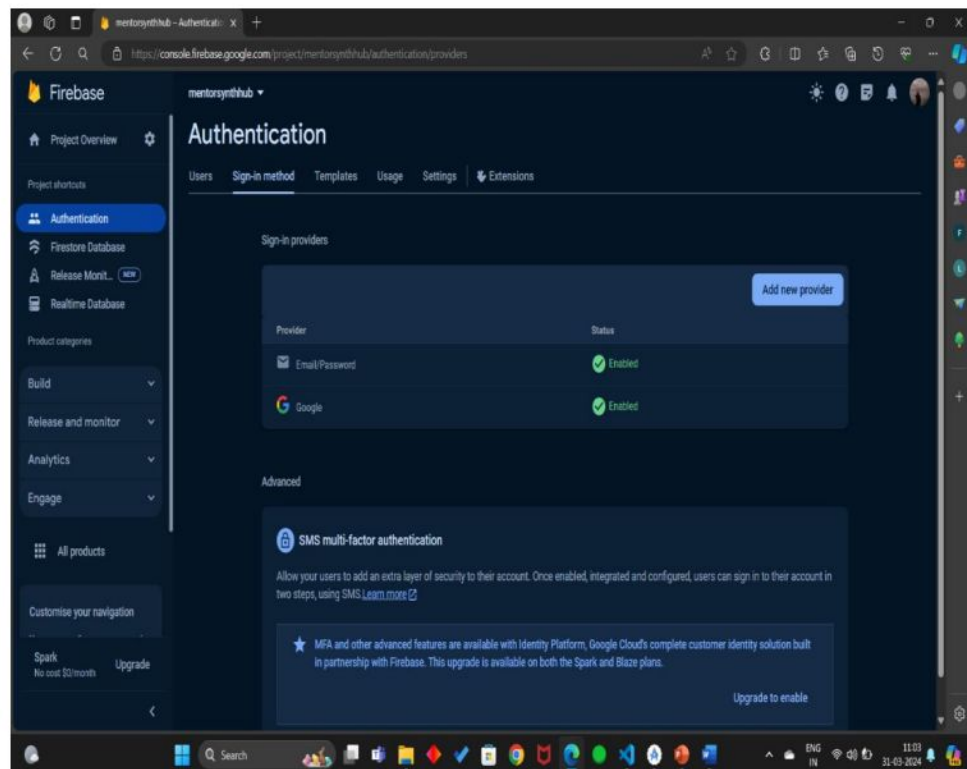


Fig:2.7: Firebase Authentication

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

6. Development Tools

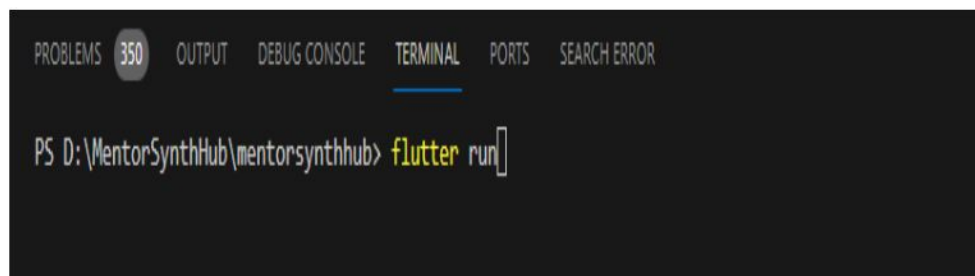


Fig:2.8: Flutter Run

Fig:2.8 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:

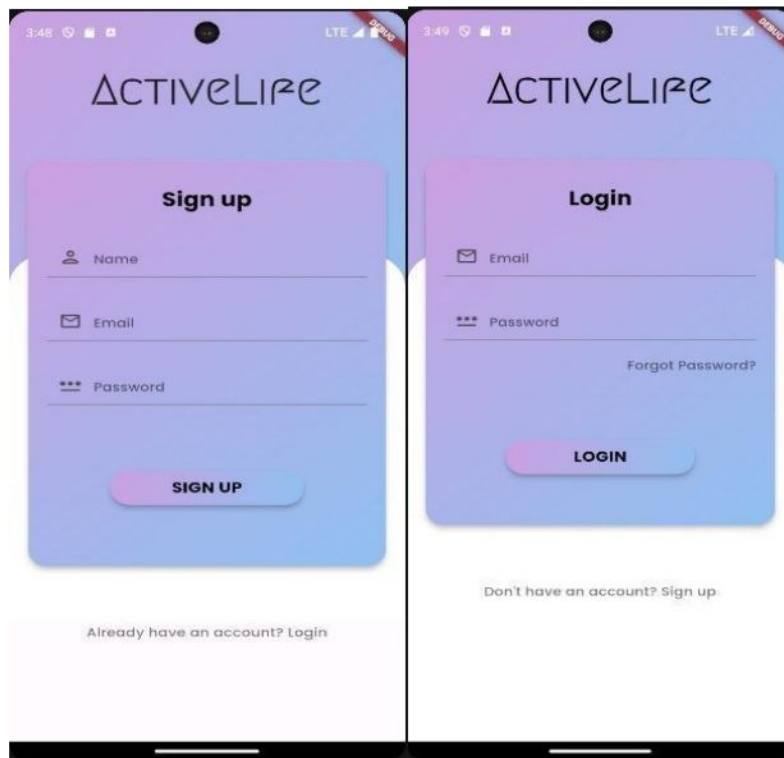


Fig:3.1: Login Page Interface

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

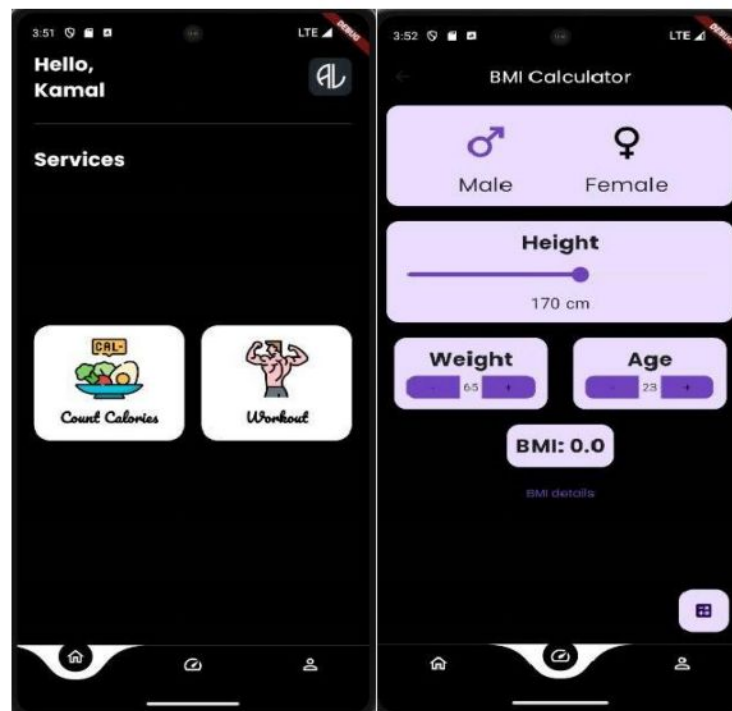


Fig:3.2: HomePage and BMI Page

Fig:3.2 above page shows the HomePage BmiPage It contains a calories and workout activity and bmi



Fig 3.3 Profile Page

Fig 3.3 above page show the profile page we can see the username and registered mail and delete an account

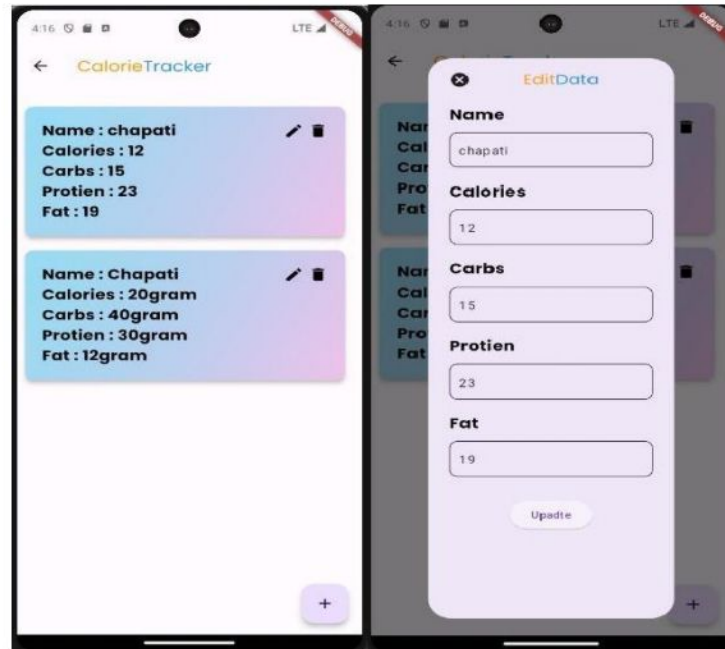


Fig:3.4: CalorieTracker and EditData Page

Fig:3.4 above page show the calorie entry page user can Enter the calorie details in the page and edit the data in the edit page

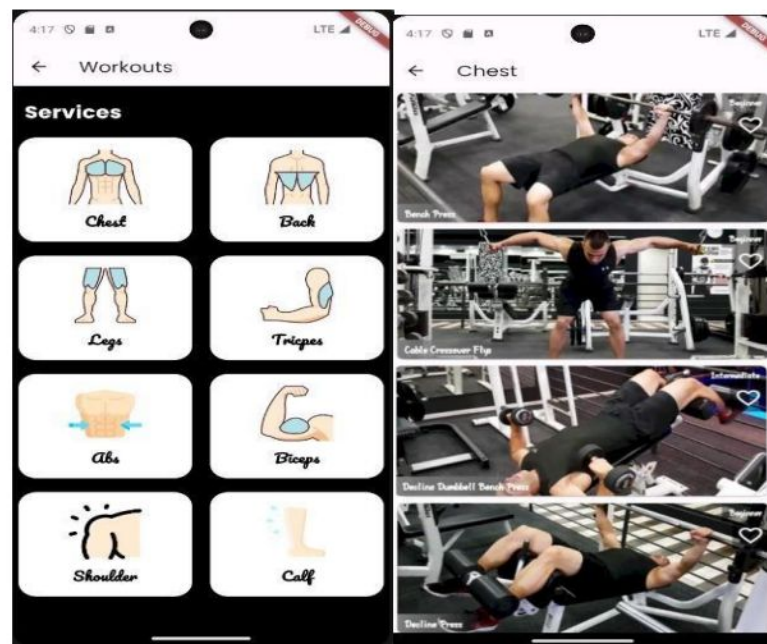


Fig:3.5: Workout home and Workout Page

Fig:3.5 above page user can select the body parts workouts vidoes

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

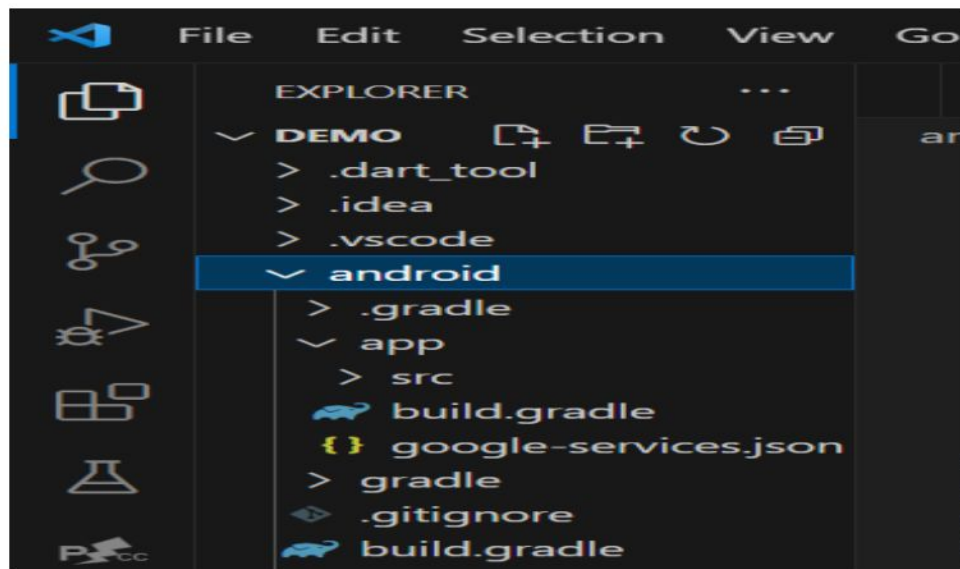


Fig:3.6: Firebase Configurations

The following steps are used to describe Fig:3.6 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

iv) Initialize Firebase

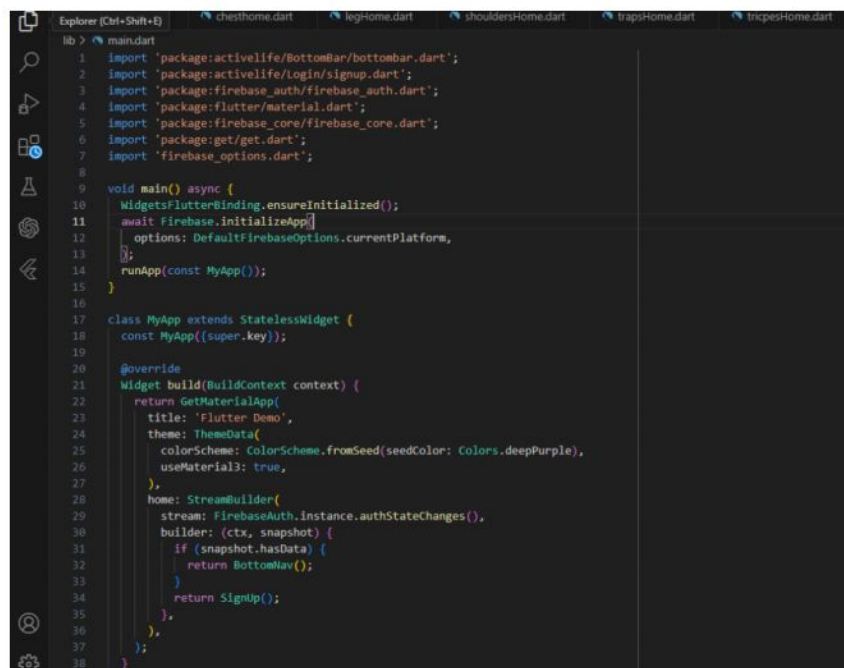


Fig:3.7: Initializing Firebase

Fig:3.7 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, "Active Life" emerges as a comprehensive solution for individuals seeking to improve their health and fitness. With its robust features including calorie tracking, workout videos, BMI calculation, and community engagement, the app offers a holistic approach to wellness. By providing users with personalized recommendations, expert guidance, and a supportive community, "Active Life" empowers individuals to set and achieve their fitness goals effectively. The frontend's visually appealing design, intuitive navigation, and seamless integration with backend functionalities ensure a delightful user experience. As users embark on their wellness journey with "Active Life," they are equipped with the tools and resources needed to lead healthier, happier lives. Overall, "Active Life" stands as a beacon of health and wellness, inspiring individuals to make positive lifestyle changes and embark on a journey towards a more active and fulfilling life.. Additionally, its commitment to accessibility and usability ensures that all users, regardless of their abilities or preferences, can seamlessly navigate and engage with the app's functionalities. Moreover, the incorporation of offline functionality and data synchronization capabilities guarantees uninterrupted access to critical event information and actions, even in low or no internet connectivity scenarios, thereby maximizing user convenience and productivity. Overall, the college events management app stands as a testament to the power of innovative technology in facilitating efficient event planning, fostering community engagement, and enriching the overall campus experience. With its user-centric d

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 Active Life 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 Active Life 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

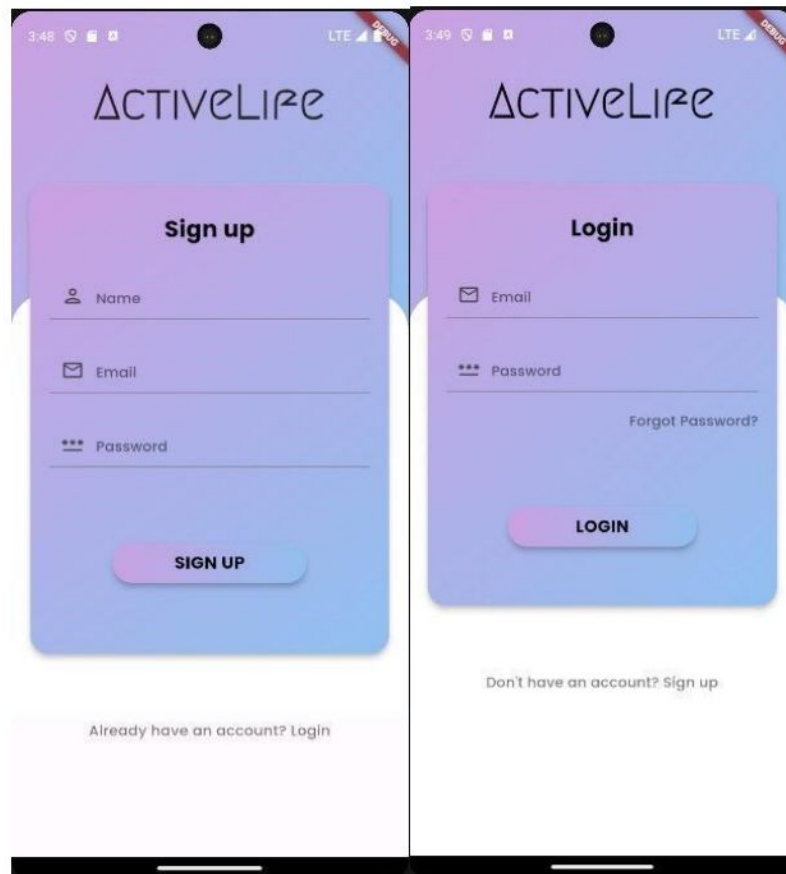
```
1  import 'package:activelife/BottomBar/bottombar.dart';
2  import 'package:activelife/Login/signup.dart';
3  import 'package:firebase_auth/firebase_auth.dart';
4  import 'package:flutter/material.dart';
5  import 'package:firebase_core/firebase_core.dart';
6  import 'package:get/get.dart';
7  import 'firebase_options.dart';
8
9  Run | Debug | Profile
10 void main() async {
11   WidgetsFlutterBinding.ensureInitialized();
12   await Firebase.initializeApp(
13     options: DefaultFirebaseOptions.currentPlatform,
14   );
15   runApp(const MyApp());
16 }
17 class MyApp extends StatelessWidget {
18   const MyApp({super.key});
19
20   @override
21   Widget build(BuildContext context) {
22     return GetMaterialApp(
23       title: 'Flutter Demo',
24       theme: ThemeData(
25         colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
26         useMaterial3: true,
27       ), // ThemeData
28       home: StreamBuilder(
29         stream: FirebaseAuth.instance.authStateChanges(),
30         builder: (ctx, snapshot) {
31           if (snapshot.hasData) {
32             return BottomNav();
33           }
34           return SignUp();
35         },
36       ), // StreamBuilder
37     ); // GetMaterialApp
```

ii)pubspec.yaml

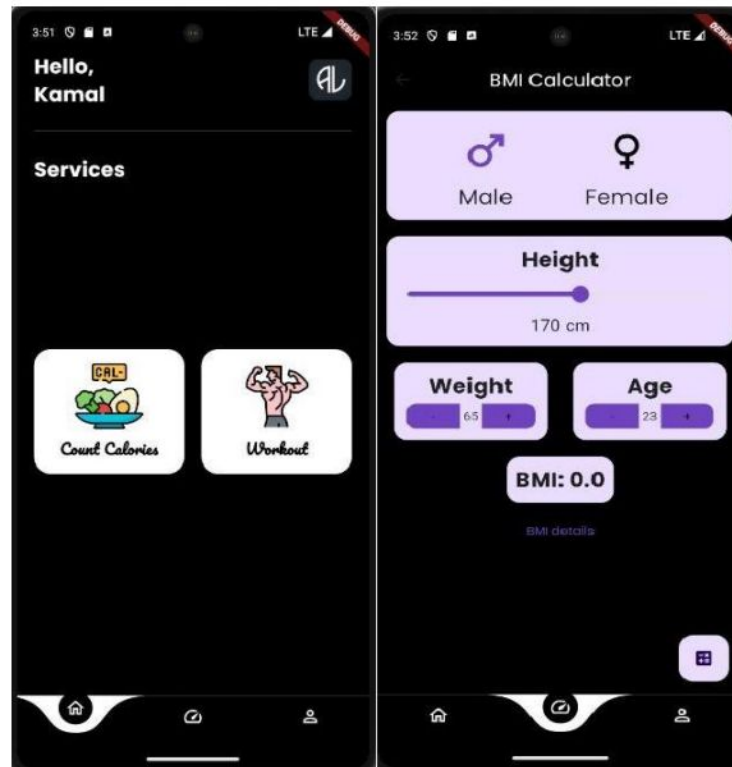
```
! pubspec.yaml ●
D: > MentorSynthHub > mentorsynthhub > ! pubspec.yaml
  🔦 Click here to ask Blackbox to help you code faster
1  name: mentorsynthhub
2  description: "A new Flutter project."
3  publish_to: 'none'
4  version: 1.0.0+1
5  environment:
6    sdk: '>=3.2.6 <4.0.0'
7  dependencies:
8    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) Calorie Tracker & Edit Data

