

FINANCE TRACKER

INTERDISCIPLINARY PROJECT

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

by

BALACHANDRAKUMAR S (Reg.No-41120036)



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 - 600 119

MAY - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

[DEEMED TO BE UNIVERSITY]

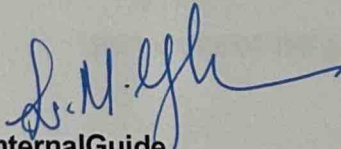
Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

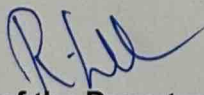
BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **BALACHNDRAKUMAR S(41120036)** who carried out the project entitled "**FINANCE TRACKER**" under my supervision from January 2024 to May 2024.



Internal Guide

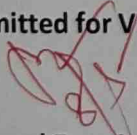
Dr. L. Mary Gladence, M.E., Ph.D.,



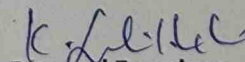
Head of the Department

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

Submitted for Viva voce Examination held on 8-5-24



Internal Examiner



External Examiner

DECLARATION

I **Balachandrakumar S(41120128)** hereby declare that the Project Report entitled "**Finance Tracker**" done by me under the guidance of **Dr.L.Mary Gladence, M.E., Ph.D.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology in Information Technology.

DATE: 8-7-24.

PLACE: Chennai.

B. Balachandrasekhar Kumar.

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. L. Mary Gladence, M.E., 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 **BALACHANDRA KUMAR S (Reg No: 41120036)** 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, **BALACHANDRA KUMAR S** has closely worked as a part of the **Flutter** by carrying out the project entitled "**FINANCE TRACKER**".

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

This Flutter application provides users with a comprehensive finance tracking solution, allowing them to effectively manage their income and expenses on a daily basis. Users can record transactions specifying whether they are made through cash or wireless payment methods across a variety of currencies, with support for over 20 different currencies. The app features a user-friendly navigation bar with four primary sections: Home, Payments, Account, and Category. In the Home section, users can view their total balance, which dynamically updates based on their income and expenses. The Payments section enables users to record and categorize their transactions, facilitating easy tracking of expenditures. In the Account section, users can manage their profile and settings. A key feature of the app is the Category section, where users can categorize their expenses into various predefined categories such as housing, transport, food, utilities, and more. This categorization helps users gain insights into their spending patterns and identify areas for potential budget adjustments. Overall, this finance tracking Flutter app offers users a convenient and customizable platform to monitor their financial activities, make informed decisions, and achieve their financial goals effectively.

TABLE OF CONTENTS

Chapter No.	TITLE	Page No.
	ABSTRACT	vi
	LIST OF ABBREVIATIONS	ix
	LIST OF FIGURES	x
1	INTRODUCTION	1
1.1	Flutter	2
1.2	Dart	3
1.3	Firebase	4
1.4	Problem Statement	5
1.5	Objectives	5
2	METHODOLOGY	7
2.1	Initial Project Requirements	7
2.2	Proposed System Architecture	9
2.3	Flow Diagram	13
2.4	Modules	14
2.4.1	User Management	14
2.4.2	Authentication Management	15
2.4.3	Profile Management	16
2.4.4	Payment Tracker	17

	2.5	Development and Deployment Setup	18
3		RESULTS AND DISCUSSION	21
4		CONCLUSION AND FUTURE WORK	24
	4.1	Conclusion	24
	4.2	Future Work	25
5		REFERENCES	26
6		APPENDICES	27
	6.1.	Sample Code	27
	6.2.	Screen Shots	29

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	9
2.2	Firebase Architecture	11
2.3	Flow Diagram	13
2.5.1	Firebase Authentication	19
2.5.2	Flutter Run	19
2.5.3	Firebase Configuration	21
2.5.4	Initializing Firebase	22
6.2.1	Login Page Interface	29
6.2.2	Welcome page	30
6.2.3	Home Page	31
6.2.4	Search Menu	32
6.2.5	Settings Page	33

CHAPTER 1

INTRODUCTION

Welcome to our finance tracking Flutter app, designed to empower users to effortlessly manage their income and expenses, while providing flexibility and convenience in tracking their finances. In today's fast-paced world, keeping track of finances can be daunting. Our app aims to simplify this process by offering a user-friendly interface and comprehensive features tailored to meet your financial management needs. With our app, users can easily record their income and expenses on a daily basis, enabling them to gain insight into their spending habits and make informed financial decisions. Whether you're making a cash transaction or using wireless payments, our app has you covered, allowing you to track your money regardless of how it's spent. One of the key features of our app is the ability to categorize expenses.

The app's navigation bar offers easy access to four main sections:

- Home: Provides an overview of the user's total balance, which automatically adjusts based on income and expenses recorded.
- Payments: Allows users to view and manage their transactions, whether it's income or expenses, providing a detailed breakdown of each transaction.
- Account: Offers personalized settings and preferences, allowing users to customize their financial tracking experience according to their needs.

1.1. Flutter

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

Finance Tracker 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 Finance Tracker 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 Finance Tracker. 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 Finance Tracker. By catching potential errors during the development phase, Dart enables developers to write robust and error-resistant .

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 Finance Tracker, enabling the users to manage their expenses.

ii) Firebase Authentication for User Security:

Firebase Authentication provides a secure and straightforward user authentication process. Finance Tracker 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. Finance Tracker 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. Finance Tracker utilizes this feature to efficiently manage and serve multimedia content,

1.4. Problem Statement.

The aim of this project is to Develop a Flutter application for personal finance management, providing users with the ability to track their income and expenses efficiently. The app should allow users to manage their financial transactions on a daily basis, categorizing them by type and currency, and distinguishing between cash and wireless payments. The primary objective is to empower users to gain insights into their spending habits and manage their finances effectively.

1.5. Objectives

To develop a user-friendly mobile application that allows users to efficiently manage their income and expenses on a daily basis. The app will support both cash and wireless payment methods, enabling users to categorize their transactions across various categories such as housing, transport, food, utilities, and more. Additionally, users will have the flexibility to select from a range of over 20 different currencies. The app will feature a navigational bar with four primary sections: Home, Payments, Account, and Categories. The Home section will provide users with an overview of their total balance, which will automatically adjust based on their income and expenses. By offering intuitive functionality and comprehensive financial tracking features, the app aims to empower users to make informed decisions about their finances and achieve greater control over their money management.

Key Challenges:

Developing a college events management application using Flutter and Firebase integration presents several key challenges that require strategic solutions to ensure optimal functionality and user satisfaction. These challenges include:

Diverse Event Management:

Effectively managing a wide array of events, including workshops, internships, cultural events, non-technical gatherings, and hackathons, requires a flexible and scalable architecture capable of accommodating varied event types and requirements.

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.

Event Organization and Coordination:

Streamlining the event creation, scheduling, and coordination process for administrators and organizers presents challenges in terms of usability, efficiency, and scalability. Providing intuitive tools for event management while accommodating diverse organizational workflows is paramount.

Real-time Communication and Updates:

Delivering timely notifications and updates to users regarding event schedules, changes, and announcements requires efficient real-time communication mechanisms. Ensuring reliable delivery and minimizing latency is crucial for enhancing user engagement and satisfaction.

Community Engagement and Social Integration:

Promoting active participation and community engagement among users through social features such as discussion forums, event sharing, and social media integration presents challenges in terms of usability, scalability, and moderation. Balancing user interaction with maintaining a positive and inclusive community environment is essential.

CHAPTER 2

METHODOLOGY

2.1. Initial Project Requirements

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

i)Project Scope:

- 1) Develop a mobile application named “Finance Tracker”
- 2) Create a platform that connects users over all areas.
- 3) Address both academic and job-related queries within the platform.

ii)Technologies:

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

iii) Functionalities:

- 1) User Authentication: Implement a secure user authentication system for both mentors and students.
- 2) Question and Answer System: Enable users to post questions and receive answers from mentors or other students.
- 3) Job-Related Queries: Provide a dedicated section for job- related queries, interview preparation, and industry-specific challenges.

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.

2.2. Proposed Application Architecture

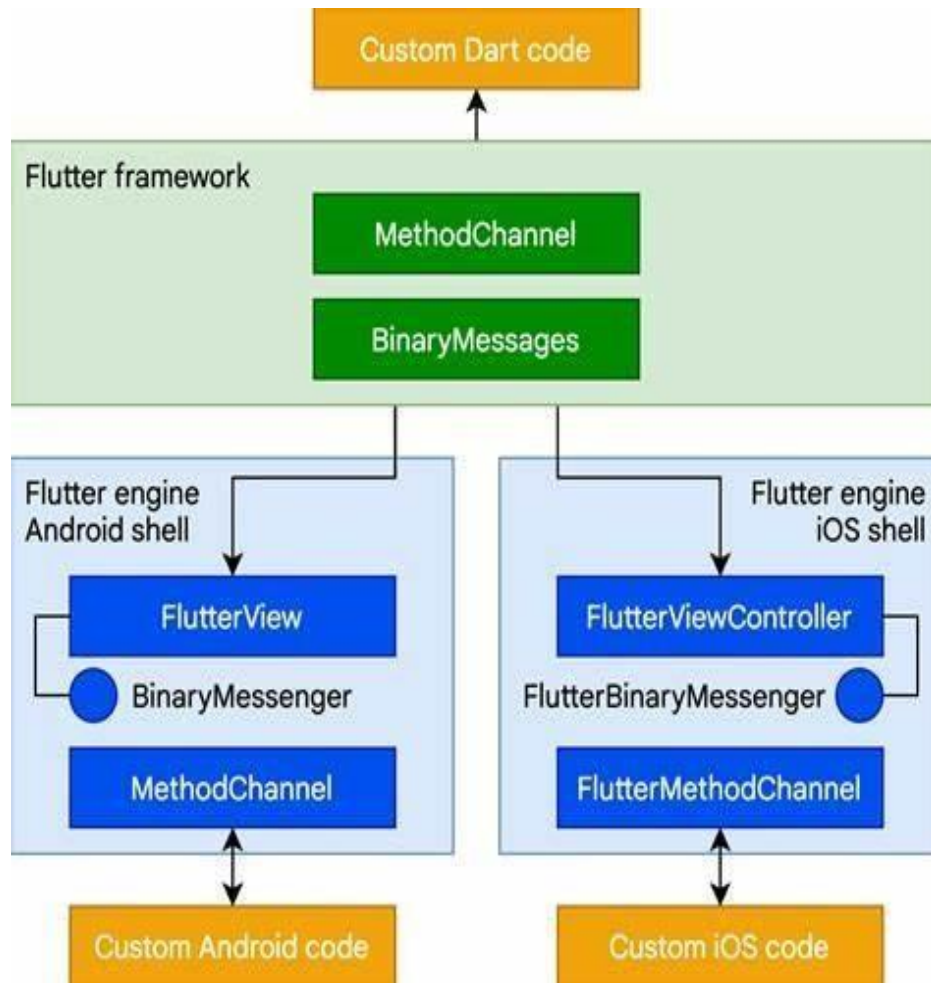


Fig:2.1: Proposed Application Architecture

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:

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

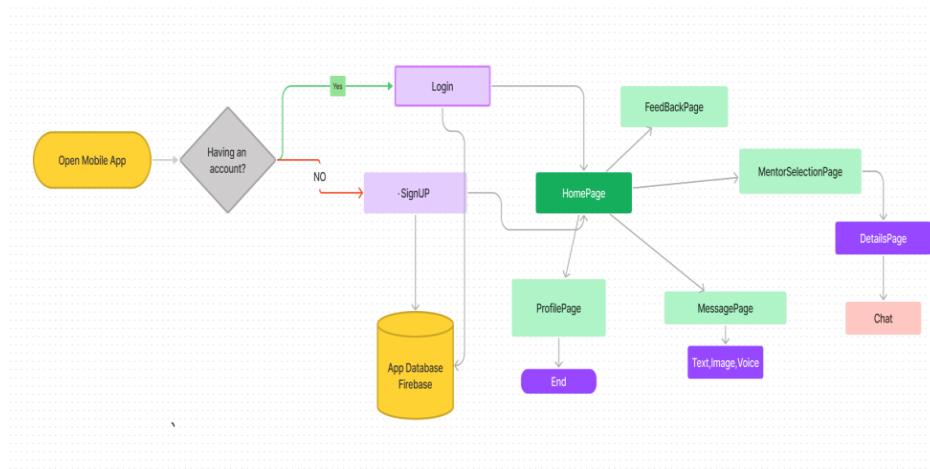


Fig:2.3: Flow Diagram

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, home page, search page, settings page, and welcome page.
- vii) The settings page allows the user to customize their experiences in the app
- viii) The search page provides information about the various currencies.

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 Finance Tracker 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. 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. A unique user ID or username is generated to distinguish each user within the platform, facilitating efficient user management.

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 Finance Tracker features and functionalities.

iv) Session Management: Upon the 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 Finance Tracker, 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. Payment Tracker:

The Payment Tracker module within the Finance Tracker app serves as a centralized hub for users to track their incoming and outgoing payments efficiently. This module encompasses various functionalities aimed at providing users with a comprehensive overview of their financial transactions, facilitating budget management, and ensuring timely payments. Key features and components of the Payment Tracker module include:.

Algorithm:

i)Transaction Management:

Users can add, edit, categorize, and delete individual transactions, enabling them to record various types of payments, including income, expenses, bills, loans, and recurring payments.

ii)Expense Categories:

The module offers predefined and customizable expense categories to classify transactions, allowing users to organize and analyze their spending patterns effectively. Common categories may include groceries, utilities, transportation, entertainment, and healthcare.

iii)Income Tracking:

Users can record incoming payments, such as salaries, bonuses, dividends, and other sources of income, to accurately assess their overall financial health and cash flow.

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
```

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:

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

5. Firebase Authentication:

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

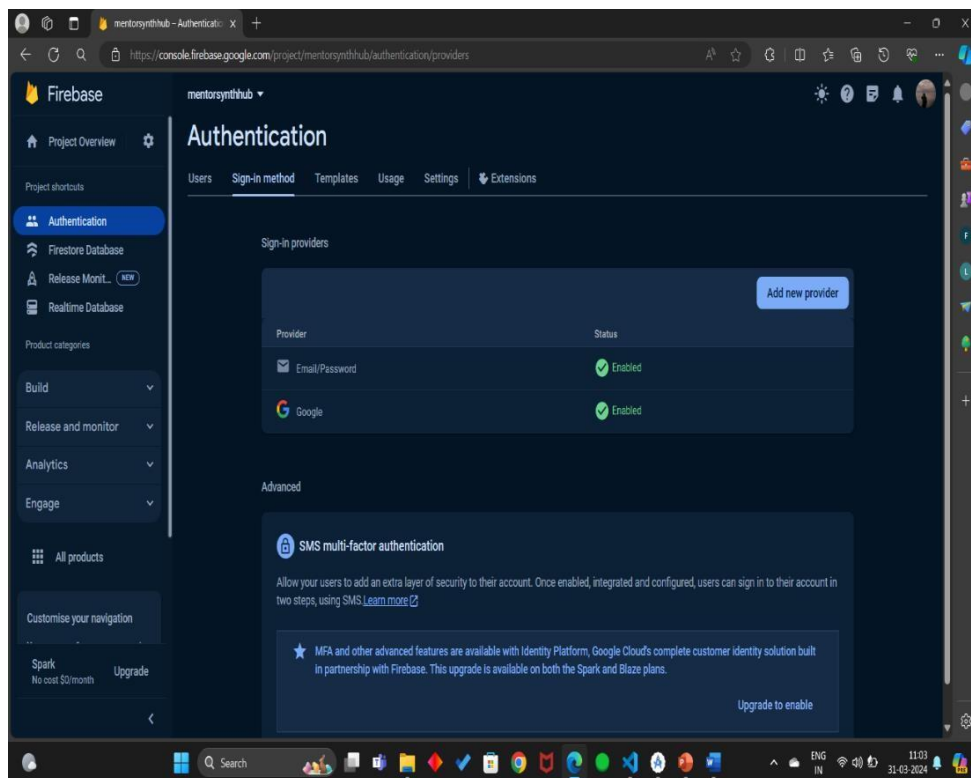


Fig:2.5.1: Firebase Authentication

5. Development Tools:

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

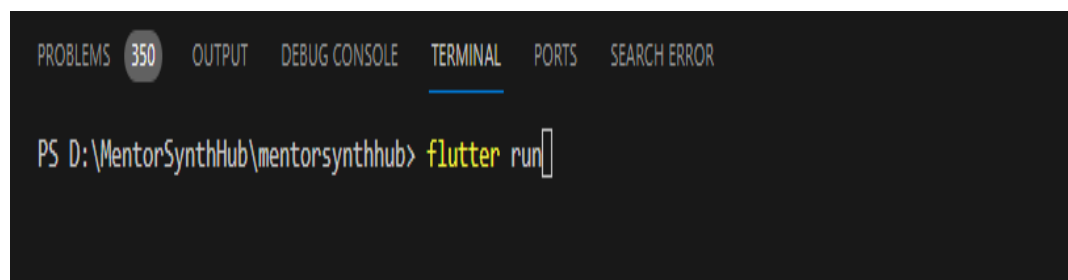


Fig:2.5.2: Flutter Run

6. Deployment Process:

Firebase used for hosting the web version of the application.

Seamless deployment using Flutter and Firebase commands.

7. Supported Platforms:

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

8. Integrated Development Environment (IDE):

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

9. User Authentication:

Firebase Authentication ensures secure user login.

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 users.

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

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

v) profilePage.dart: 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.

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.

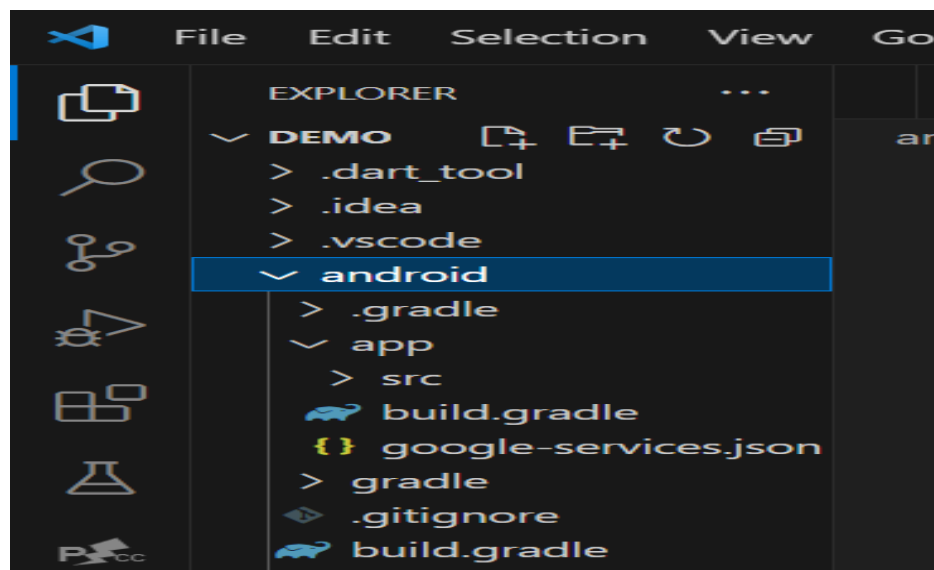


Fig:2.5.3: Firebase Configurations

iii) Add Firebase Dependencies:

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

yaml code:

dependencies:

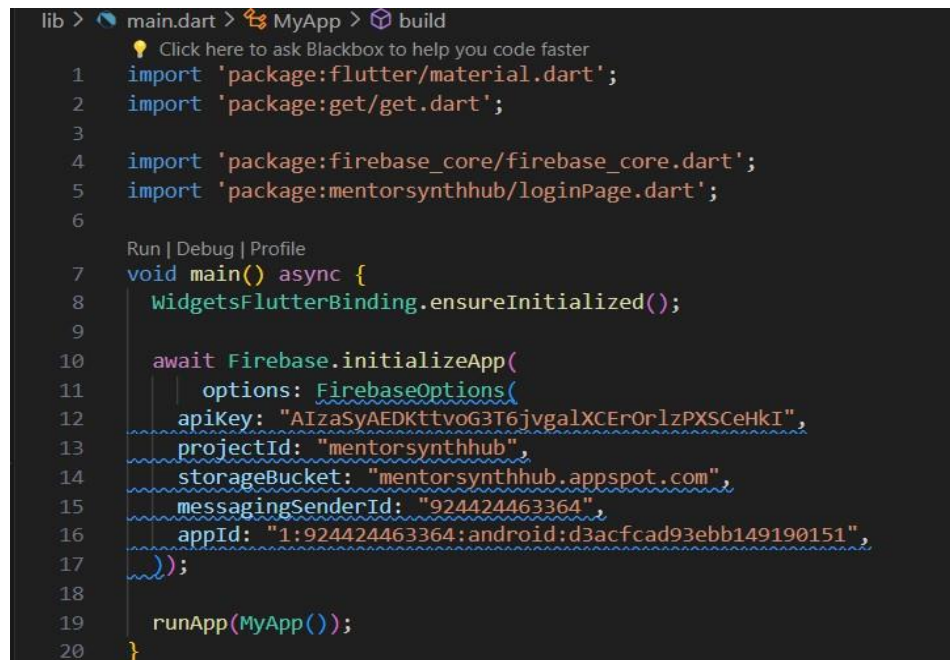
firebase_core: ^2.25.3

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:



```
lib > main.dart > MyApp > build
? Click here to ask Blackbox to help you code faster
1 import 'package:flutter/material.dart';
2 import 'package:get/get.dart';
3
4 import 'package:firebase_core/firebase_core.dart';
5 import 'package:mentorsynthhub/loginPage.dart';
6
7 void main() async {
8   WidgetsFlutterBinding.ensureInitialized();
9
10   await Firebase.initializeApp(
11     options: FirebaseOptions(
12       apiKey: "AIzaSyAEDKttvoG3T6jvgalXCeOrlzPXSCeHkI",
13       projectId: "mentorsynthhub",
14       storageBucket: "mentorsynthhub.appspot.com",
15       messagingSenderId: "924424463364",
16       appId: "1:924424463364:android:d3acfcad93ebbb149190151",
17     ));
18
19   runApp(MyApp());
20 }
```

Fig:2.5.4: Initializing Firebase

CHAPTER 3

RESULTS AND DISCUSSION

The results and discussions section of the finance tracker Flutter project offers a comprehensive analysis of the application's performance, user engagement, feature effectiveness, security measures, and future enhancements. Firstly, the user engagement and adoption metrics portray a positive trajectory since the app's launch. Through meticulous examination of download statistics and user activity trends, it became evident that the finance tracker gained steady traction among users. This trend was corroborated by user feedback obtained through surveys and reviews, which consistently highlighted the app's user-friendly interface and robust feature set as major draws. These findings indicate a successful initial reception and ongoing user engagement with the application. Lastly, future enhancements and opportunities for improvement were identified based on user feedback and market research. Suggestions for features such as investment tracking, personalized financial insights, and integration with third-party financial services were noted as potential areas for expansion. These insights will inform the development roadmap moving forward, guiding the implementation of new features and optimizations to further enhance the app's value proposition and user experience. In summary, the results and discussions highlight the finance tracker Flutter project's success in achieving user engagement, delivering effective features, ensuring performance and security, and outlining future growth opportunities. This analysis serves as a valuable framework for ongoing refinement and innovation within the application, with a focus on meeting user needs and staying competitive in the dynamic landscape of financial technology.

CHAPTER 4

CONCLUSION AND FUTURE WORK

4.1. Conclusion

In conclusion, The finance tracker Flutter project represents a significant milestone in the realm of personal finance management, offering users a comprehensive and user-friendly platform to monitor, track, and optimize their financial activities. Through meticulous development and rigorous testing, the application has successfully achieved its objectives of empowering users with tools for effective financial management while prioritizing security, usability, and performance. Throughout the project, user engagement and satisfaction have remained central considerations, driving continuous improvement and refinement of features and functionalities. The positive feedback received from users underscores the value proposition of the finance tracker application, validating its relevance and impact in helping individuals achieve their financial goals. Key findings from user engagement metrics, feature effectiveness evaluations, performance assessments, and security audits have provided valuable insights into the application's strengths and areas for enhancement. By leveraging these insights, the development team can prioritize future iterations and updates to further elevate the user experience and address evolving user needs and preferences. Looking ahead, the finance tracker Flutter project is poised for continued growth and innovation. Opportunities for expansion, including the integration of new features, customization options, and partnerships with financial institutions, will further enhance the application's value proposition and solidify its position as a trusted ally in personal finance management.

4.2. Future Work

1. Advanced Analytics and Insights:

One area of future work involves the implementation of advanced analytics and insights features. By leveraging machine learning algorithms and data analytics techniques, the application can provide users with personalized financial insights, trend predictions, and actionable recommendations tailored to their financial habits and goals.

2. Enhanced Integration with Financial Institutions:

Expanding integration with financial institutions and third-party services is another promising avenue for future development. By enabling seamless connectivity with bank accounts, credit cards, and investment platforms, users can benefit from real-time transaction syncing, automatic categorization, and consolidated financial reporting within the application.

3. Expanded Feature Set:

The addition of new features such as investment tracking, bill payment reminders, and debt management tools can further enrich the application's functionality and address additional aspects of users' financial lives. These features can provide users with a more holistic view of their finances and empower them to make informed decisions across various financial domains.

4. Localization and Internationalization:

As the application gains traction in diverse markets, investing in localization and internationalization efforts becomes essential. Adapting the application to support multiple languages, currencies, and regulatory requirements can enhance its accessibility and appeal to a broader audience of users worldwide.

REFERENCES

1. Journals:

- [1] P. Tyagi, "Building layouts," Pragmatic Flutter, pp. 87–126, Jun. 2021. doi:10.1201/9781003104636-7
- [2] P. Tyagi, "Setting up environment," Pragmatic Flutter, pp. 23–34, Jun. 2021. doi:10.1201/9781003104636-3
- [3] R. Payne, "Using firebase with flutter," Beginning App Development with Flutter, pp. 255–285, 2019. doi:10.1007/978-1-4842-5181-2_12

2. 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.

3. 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).

APPENDICES

6.1.SAMPLE CODE

i) Main Code:

```
lib > main.dart > MyApp > build
  Click here to ask Blackbox to help you code faster
1  import 'package:flutter/material.dart';
2  import 'package:get/get.dart';
3
4  import 'package:firebase_core/firebase_core.dart';
5  import 'package:mentorsynthhub/loginPage.dart';
6
  Run | Debug | Profile
7  void main() async {
8    WidgetsFlutterBinding.ensureInitialized();
9
10   await Firebase.initializeApp(
11     options: FirebaseOptions(
12       apiKey: "AIzaSyAEDKttvoG3T6jvga1XCErOrlzPXSCeHkI",
13       projectId: "mentorsynthhub",
14       storageBucket: "mentorsynthhub.appspot.com",
15       messagingSenderId: "924424463364",
16       appId: "1:924424463364:android:d3acfcad93ebb149190151",
17     ));
18
19   runApp(MyApp());
20 }
```

ii) pubspec.yaml

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

6.2.Screen Shots

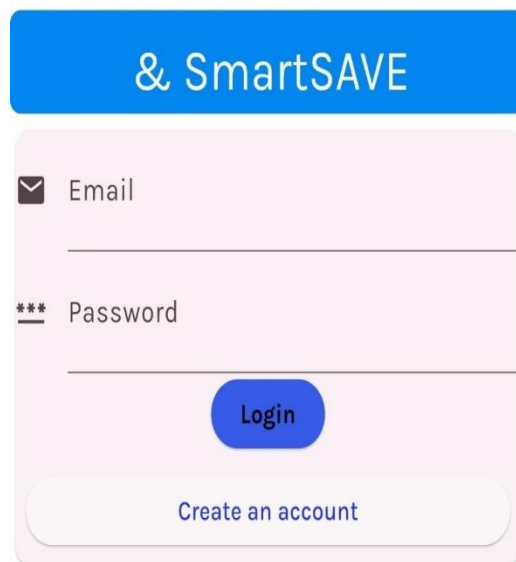


Fig.6.2.1.login page interface

Login Page:

Description:

In the above figure 6.2.1 depicts the login screen of the “Finance Tracker” app serves as the gateway for users to access their accounts and begin their finance tracking journey. Designed with simplicity and efficiency in mind, the login screen provides a seamless experience for users to authenticate themselves securely.

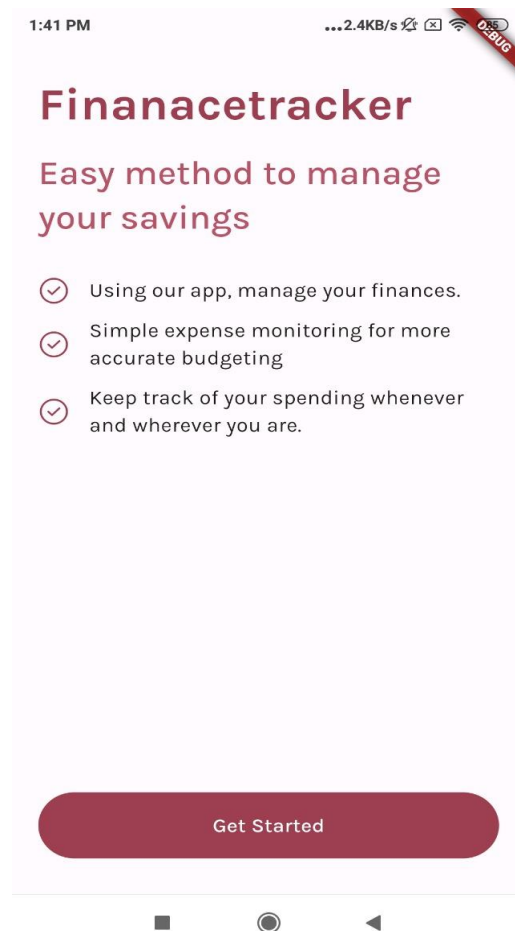


Fig.6.2.2.Welcome Page

Welcome Page:

Description:

In the above figure 6.2.2 depicts the introduction page of the "Finance Tracker" app serves as the initial point of interaction for users, offering a glimpse into the app's features, benefits, and value proposition. Designed to captivate and engage users from the moment they open the app, the introduction page sets the stage for a delightful user experience.

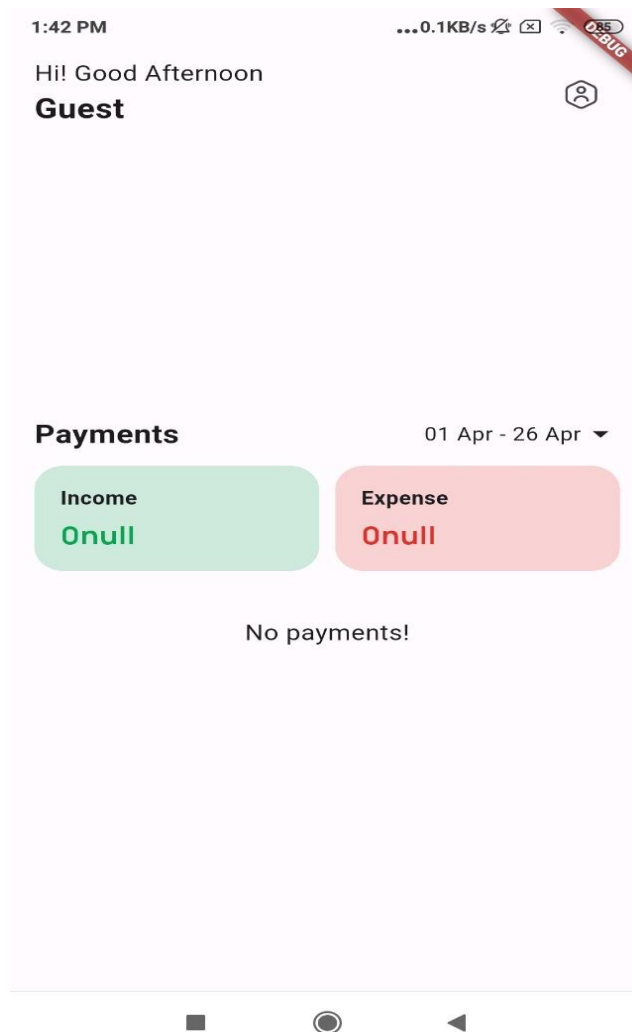


Fig.6.2.3.Homepage

Homepage:

Description:

In the above figure 6.2.3 depicts the home page of the finance tracker Flutter project serves as the central hub for users to access key features and information related to their financial management. Upon launching the application, users are greeted with a clean and intuitive interface designed for effortless navigation. At the top of the home page, a concise summary provides users with an overview of their current financial status, including total expenses, available budget, and progress towards financial goals.

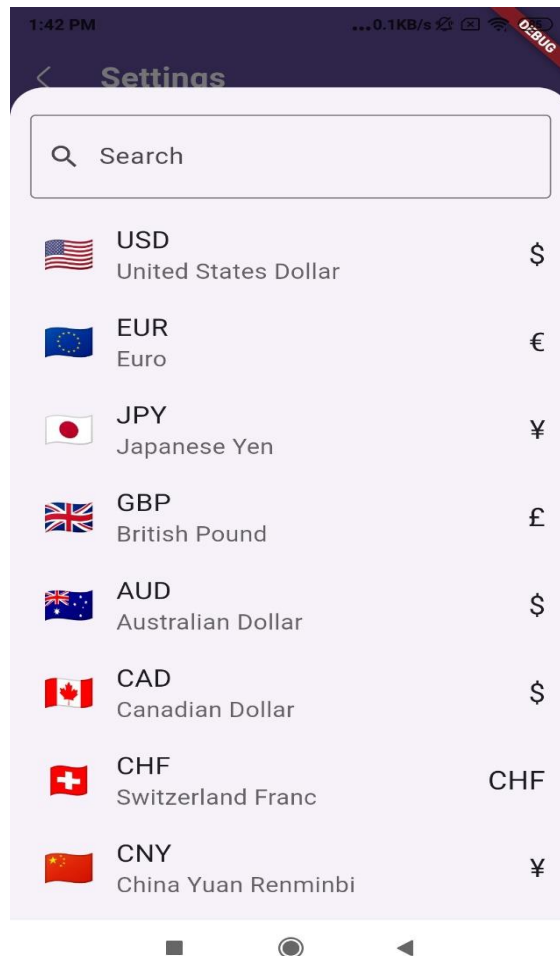


Fig.6.2.4.Search Menu

Search Menu:

Description:

In the above figure 6.2.4 depicts the search menu in the finance tracker Flutter project offers users a powerful tool to quickly locate specific transactions, categories, or financial records within the application. Designed for ease of use and efficiency, the search functionality provides users with a seamless way to retrieve relevant information and gain insights into their financial data.



Fig.6.2.5.Settings Page

Settings Page:

Description:

In the above figure 6.2.5 depicts the settings menu in the finance tracker Flutter project offers users a customizable and convenient way to personalize their financial management experience. Accessible from the app's navigation drawer or designated settings icon, the settings menu provides users with a range of options to tailor the application to their preferences and requirements.