A Lab Project Report on

# PERSONAL MONEY ASSISTANT

Submitted in partial fulfillment of the requirement of the lab

**UI DESIGN-FLUTTER**

**III-B. TECH. I SEM**

**in**

**Department of CSE(AI & ML)**

Submitted by

**CH.SAI GANESH (227R1A6681)**

**DEPARTMENT OF CSE(AI & ML)**

**CMR TECHNICAL CAMPUS**

**UGC AUTONOMOUS**

**(Accredited by NBA, Approved by AICTE, Affiliated to JNTUH)**

**Kandlakoya, Medchal Road, Hyderabad-501401**

**2024-2025**

**CMR TECHNICAL CAMPUS**

**UGC AUTONOMUS**

**(Approved by AICTE, Affiliated to JNTUH, Kukatpally, Hyderabad)**

**Kandlakoya, Medchal Road, Hyderabad-501401.**

**DEPARTMENT OF CSE(AI & ML)**

## CERTIFICATE

This is to certify that a Mini Project entitled with: "**PERSONAL MONEY ASSISTANT**" is being Submitted by **CH.SAI GANESH(227R1A6681)** in partial fulfillment of the requirement for completion of the **UI DESIGN-FLUTTER** III B.Tech I- Semester is a record of a bonfide work carried out under guidance and supervision.

**Signature of Faculty** **Signature of the HOD**

# ACKNOWLEDGEMENT

We are extremely grateful to **Dr. A. Raji Reddy, Director and Dr. S Rao Chintalapudi, Head of Department,** Department of CSE(AI & ML),CMR TECHNICAL CAMPUS for their inspiration and valuable guidance during entire duration. We are extremely thankful to our Assistant Professor **Dr.K.MAHESH**, Department of CSE(AI & ML)department for his constant guidance, encouragement and moral support throughout the project. We express our thanks to all staff members and friends for all the help and coordination extended in bringing out this mini project successfully in time.

**CH.SAI GANESH(227R1A6681)**

## TABLE OF CONTENTS

# ABSTRACT

Personal Money Assistant is a comprehensive and user-friendly money management application designed to help users take control of their finances. Built with Flutter, it combines seamless functionality with a visually appealing interface to provide a holistic financial tracking solution.

The app features secure user authentication through Firebase, supporting multiple login methods, including email and social media platforms like Google, Facebook, LinkedIn, and Discord. SQLite is used as the local database to ensure offline access to financial data, while shared preferences enable extensive customization options such as currency type, date format, and language selection.

Cashmate empowers users to track income and expenses efficiently, offering categorization, detailed transaction descriptions, and a deletion feature for data accuracy. Visual insights like pie charts and a calendar view simplify the understanding of spending habits, promoting better financial decisions. The app further enhances user experience by supporting custom categories, icons, and global adaptability through flexible currency and language settings.

To maintain clean records, CashMate includes data maintenance tools like resetting categories and deleting all data. Social interaction features, such as app sharing and feedback options, encourage community engagement and promote financial literacy. In summary, CashMate is an all-in-one solution for managing personal finances. With its secure, customizable, and intuitive features, it caters to a wide range of financial management needs, making it an indispensable tool for users worldwide.

# 1.INTRODUCTION

Managing personal finances is an essential aspect of modern life. With the growing complexities of daily expenses, income sources, and savings, a structured and efficient system to track and organize finances has become indispensable. CashMate: Your Personal Money Assistant is a revolutionary application designed to simplify financial management. By integrating cutting-edge technology with user-centric features, CashMate empowers individuals to take control of their monetary well-being.

CashMate is a Flutter-based application that combines security, customization, and insightful tools to cater to a diverse range of users. Whether you are a student managing a monthly allowance, a professional tracking expenses, or a small business owner overseeing transactions, CashMate adapts to your needs seamlessly. The app serves as a comprehensive solution for financial organization, offering features like income and expense tracking, visual data analysis, and detailed customization options.

One of the foundational pillars of CashMate is its emphasis on user security and accessibility. The app integrates Firebase for authentication, allowing users to log in securely using email or social media accounts such as Google, Facebook, LinkedIn, and Discord. This multi-faceted login system ensures that users can access their accounts conveniently without compromising on security. Furthermore, the Forgot Password feature ensures account recovery is straightforward and secure, adding an extra layer of reliability to the app's authentication system.

To enhance user convenience, CashMate employs SQLite as its local database. This ensures that users can store and access their financial data offline, making the app dependable even in the absence of an internet connection. The use of SQLite also supports rapid data retrieval and manipulation, enabling users to record, edit, or delete transactions with ease. Coupled with shared preferences for customization, the app allows users to tailor their experience by setting their preferred currency type, date format, and language. This adaptability makes CashMate a global financial tool suitable for users from diverse regions and backgrounds.

CashMate goes beyond simple record-keeping by offering robust tools for financial analysis and planning. The app organizes transactions into income and expense categories, each customizable to suit individual needs. With features like detailed descriptions for transactions, users gain a deeper understanding of their financial activities. For instance, a grocery expense can be categorized under "Food" with an optional note specifying the store or items purchased. This level of granularity ensures that every transaction has context, making financial reviews more meaningful.

One of the standout features of CashMate is its visual analytics. The app provides users with a pie chart analyzer that visually represents spending and income patterns over customizable timeframes. Whether you want to analyze your expenses for the week, month, or year, the visual insights make it easy to identify trends, such as overspending in specific categories or consistent savings. These insights help users make informed financial decisions, promoting better budgeting and resource allocation.

Another unique aspect of CashMate is its Calendar View feature. This allows users to view their financial history in a date-wise format, adding a temporal dimension to their financial tracking. The calendar not only enhances transaction visibility but also aids in planning for recurring expenses, such as monthly bills or annual subscriptions. This feature simplifies the process of correlating transactions with specific dates, making it easier to identify patterns and plan for the future.

CashMate's commitment to personalization extends to its category management system. Users can create custom categories for both income and expenses, choosing from a wide array of icons to visually represent their financial activities. For example, a freelancer can create a category for "Client Payments" with an icon representing professional work, while a fitness enthusiast might add a category for "Gym Expenses." This customization ensures that the app aligns perfectly with individual lifestyles, enhancing its usability and relevance.

In addition to its analytical and customization features, CashMate prioritizes data maintenance and privacy. Users can reset all categories or delete all data at their discretion, ensuring that their financial records remain tidy and up-to-date. This flexibility is particularly useful for individuals who want to start fresh or discontinue using the app without leaving behind any data traces. The app also supports transaction deletion, allowing users to remove incorrect or duplicate entries with ease, thereby maintaining the accuracy of their financial records.

# 2.Literature Survey

A literature survey serves as a foundation to understand the existing landscape of personal finance management tools and the methodologies employed in their development. CashMate: Your Personal Money Assistant is positioned as an advanced financial management solution, leveraging the capabilities of Flutter, Firebase, and SQLite to deliver secure and customizable features. To contextualize its development, this survey examines the evolution of personal finance tools, relevant technological frameworks, and user-centric features in similar applications.

## 2.1 Evolution of Personal Finance Management Tools

Personal finance management has transitioned significantly from manual bookkeeping to advanced digital tools. Historically, individuals maintained handwritten records of income and expenses. However, with the advent of computers, spreadsheet applications such as Microsoft Excel introduced automated calculations and basic data visualization. This evolution laid the groundwork for the development of dedicated financial management software.

The proliferation of smartphones brought a paradigm shift, making finance management accessible on-the-go. Applications like Mint, YNAB (You Need A Budget), and PocketGuard have set benchmarks for features such as automatic transaction tracking, budgeting, and real-time notifications. However, these applications often lack robust offline functionality or require subscriptions for full access, presenting an opportunity for tools like CashMate to fill this gap.

## 2.2 Technological Frameworks and Methodologies

The backbone of CashMate lies in the integration of state-of-the-art technologies, including Flutter, Firebase, and SQLite. This section explores the significance of these frameworks in personal finance applications.

### 2.2.1. Flutter

Flutter, an open-source UI toolkit developed by Google, has gained traction for its cross-platform development capabilities. Applications built with Flutter achieve a nativelike performance while maintaining a single codebase for both Android and iOS.

- **Advantages for Financial Apps**: Flutter's fast rendering and widget-based architecture allow developers to design responsive and interactive financial dashboards, such as CashMate's pie chart analyzer.

- **Customization**: Shared preferences in Flutter enable personalized user settings, such as currency type, date format, and language, making apps versatile for global users.

- **Adoption in Industry**: Applications like Nubank, one of the largest digital banks in Brazil, have demonstrated the potential of Flutter for financial services due to its scalability and speed.

### 2.2.2 Firebase

Firebase, a backend-as-a-service (BaaS) platform by Google, offers robust tools for authentication, database management, and analytics. For personal finance applications, Firebase ensures data security and real-time synchronization.

- **User Authentication**: Firebase Authentication supports multi-factor and social logins, which are crucial for secure access to financial data. Applications like CashMate leverage Firebase's capabilities for seamless user onboarding.

- **Scalability**: Firebase's cloud storage and analytics ensure that the app can handle a growing user base without performance degradation.

### 2.2.3 SQLite

SQLite is a lightweight, serverless database that provides offline functionality, making it ideal for personal finance apps.

- **Offline Access**: Unlike cloud-based tools, SQLite enables users to store and retrieve data locally, ensuring uninterrupted access even without an internet connection.:

# 3.Analysis and Design

## 3.1 System Analysis:

System analysis is a critical phase in the software development lifecycle (SDLC), wherein the requirements, challenges, and potential solutions for the system are thoroughly examined. For CashMate, the system analysis involves identifying key functional and non-functional requirements, studying user behavior, and evaluating existing solutions to design a system that addresses current gaps in personal finance management applications.

### 3.1.1 Identifying Functional Requirements

Functional requirements are the core capabilities that the system must deliver to meet user expectations. For CashMate, the functional requirements include:

- **User Authentication**: Secure login and registration using Firebase, with support for email, Google, Facebook, LinkedIn, and Discord accounts.
- **Transaction Management**: Enabling users to add, edit, and delete transactions, including income and expenses, with associated descriptions, dates, and times.
- **Category Customization**: Allowing users to create, edit, and delete categories with custom icons to suit individual financial activities.
- **Data Visualization**: Providing pie charts and analytics to represent income and expense patterns over various timeframes (daily, weekly, monthly).
- **Calendar View**: Displaying transactions date-wise to enhance visibility and planning.
- **Personalization Settings**: Options for setting preferred currency, language, and date formats.
- **Data Maintenance**: Tools to reset categories or delete all data for privacy and record-keeping purposes.

### 3.1.2 Identifying Non-Functional Requirements

Non-functional requirements define the quality attributes of the system and ensure that it meets performance, scalability, and usability expectations. Key non-functional requirements for CashMate include:

- **Performance**: The app must provide quick responses for all user interactions, such as retrieving transactions and generating visual insights.
- **Security**: Implementation of robust authentication mechanisms and secure data storage using Firebase and SQLite.
- **Usability**: A simple and intuitive interface designed using Flutter to ensure user-friendliness for all demographics.
- **Scalability**: The system must handle an increasing number of users and transactions without degradation in performance.
- **Reliability**: The app must ensure data consistency and provide seamless offline access using SQLite.

### 3.1.3 Problem Analysis

Existing personal finance applications often fall short in addressing the following user concerns:

- **Limited Offline Functionality**: Many apps rely heavily on cloud-based systems, rendering them unusable without an internet connection.
- **Lack of Customization**: Users frequently encounter rigid category systems that do not align with their unique financial activities.
- **High Subscription Costs**: Premium features, such as detailed analytics, are often hidden behind paywalls.

### 3.1.4 Feasibility Study

A feasibility study evaluates the practicality of implementing the CashMate system:

- **Technical Feasibility**: Leveraging proven technologies like Flutter, Firebase, and SQLite ensures that the app is technically viable.
- **Operational Feasibility**: The app is designed to align with user habits and needs, making it practical for everyday financial management.
- **Economic Feasibility**: As an MIT-licensed app, CashMate minimizes costs for users and provides features without additional charges.

## 3.2 System Design:

System design translates the insights gained from system analysis into a structured framework that guides the implementation of the application. For CashMate, the design involves both architectural and component-level considerations, ensuring that the system is modular, maintainable, and scalable.

### 3.2.1 High-Level Architecture

CashMate employs a three-tier architecture, which separates the system into presentation, business logic, and data layers for better modularity and scalability.

1. **Presentation Layer:**
   - Built using Flutter, this layer manages user interactions and displays financial data, including transaction lists, pie charts, and the calendar view.
   - Features a responsive design to ensure a seamless experience across devices.
2. **Business Logic Layer:**
   - Handles core functionalities, such as transaction management, data processing, and chart generation.
   - Implements shared preferences for customization options like currency type, language, and date format.

### 3.Data Layer:

- Utilizes Firebase for user authentication and SQLite for local data storage.
- Ensures secure and efficient data retrieval and manipulation through well-structured database schemas.

### 3.2.2 Database Design

The database design is crucial for efficient data storage and retrieval. For CashMate, SQLite is structured to support the following entities:

| Entity | Attributes |
|---|---|
| User | UserID, Email, Password, PreferredCurrency, PreferredLanguage, DateFormat |
| Transaction | TransactionID, UserID, CategoryID, Amount, Description, Date, Time |
| Category | CategoryID, UserID, Name, Icon |

Relationships:

- The User table has a one-to-many relationship with the Transaction and Category tables.
- Each Transaction is linked to a specific Category.

## 3.3 User Interface Design

The UI is designed with simplicity and accessibility in mind. Key design considerations include:

- Home Screen: Displays a summary of transactions and a pie chart for quick insights.
- Add Transaction Screen: Provides fields for entering transaction details, including category selection and description.
- Settings Screen: Offers customization options for currency, language, and date format.

## 3.4 Component-Level Design

Each feature of CashMate is developed as a modular component:

- Authentication Component: Integrates Firebase for secure user login and registration.
- Transaction Component: Manages transaction entry, editing, deletion, and categorization.

# 4.Implementation

The implementation phase of CashMate focuses on translating the design and analysis into a functional application that meets user requirements. By leveraging modern technologies like Flutter, Firebase, and SQLite, CashMate delivers a secure, intuitive, and robust platform for personal finance management. This section outlines the implementation details for key components of the system, emphasizing the practical steps and technologies used to realize the features and functionalities.

## 4.1 Development Environment

CashMate is developed using Flutter, a cross-platform framework, ensuring consistent performance across Android and iOS devices. Key tools and technologies used in the implementation include:

**Flutter SDK**: For building the user interface and implementing app logic.

**Firebase**: For authentication and secure backend integration.

**SQLite**: For local data storage to ensure offline functionality.

**Dart Programming Language**: The primary language for writing the application logic.

**IDE**: Visual Studio Code with Flutter and Dart plugins for streamlined development.

## 4.2 Implementation of Core Features

### 4.2.1 User Authentication

The authentication module is implemented using Firebase Authentication. It supports multiple sign-in methods, including email and social media platforms like Google, Facebook, LinkedIn, and Discord.

**Firebase Setup:**

- The Firebase project is configured with the app’s unique bundle identifier.
- OAuth credentials are generated for social media logins.

**Authentication Workflow:**

- Users enter login credentials.
- Firebase authenticates the input and grants access if valid.

### 4.2.2 Transaction Management

Transaction management is a core feature of CashMate, allowing users to add, edit, delete, and view transactions. The implementation involves:

**Database Schema:**

- Tables for storing transactions, categories, and user preferences are created in SQLite.
- The schema includes relationships to link transactions to categories and users.

**CRUD Operations:**

- Flutter’s sqflite package is used for database operations.
- Functions like addTransaction(), updateTransaction(), and deleteTransaction() handle respective database operations.

**UI Integration:**

- Forms are implemented for users to input transaction details.
- Transactions are displayed using ListView, categorized by date and type.

**Visual Insights with Pie Charts:**

The visual insights module provides users with a graphical representation of their financial data.

**Chart Implementation:**

- The charts_flutter package is used to render pie charts.
- Data from SQLite is aggregated to calculate category-wise spending.

**Dynamic Filtering:**

- Filters for timeframes (daily, weekly, monthly) are implemented using dropdown menus.
- Data is dynamically recalculated based on the selected timeframe and fed into the chart widget.

## 4.3 Calendar View

The calendar view enhances transaction visibility by displaying records date-wise.

**Implementation:**

- The table_calendar package is used to render the calendar.
- Transactions are fetched from SQLite for the selected date and displayed below the calendar.

**User Interaction:**

- Users can tap on a date to view transactions for that day.
- Swiping through months allows navigation across different periods.

**Workflow:**

- Users enter their registered email.
- Firebase sends a password reset link to the email.
- Users follow the link to reset their password securely.

## 4.4 Social Interaction

- To enhance user engagement, CashMate includes sharing and feedback options.

**Share App:**

Flutter’s share package allows users to share the app link via messaging or social media platforms.

**Rate App:**

- Links to app stores are provided for users to leave feedback.

## 4.5 User Authentication:

- On launching the app, users are prompted to log in or sign up.

**Home Screen:**

- After successful login, users land on the home screen displaying a transaction summary and pie chart.

**Adding a Transaction:**

- Users navigate to the "Add Transaction" screen, fill in details, and save the transaction.

**Viewing Data:**

- The calendar view and transaction lists provide insights into spending habits.

**Customization**:

- Users access the settings menu to modify preferences such as currency and language.

**Maintenance:**

- Reset or delete options are available for managing data.

# 5.Testing and debugging

## 5.1 Unit Testing

Ensures individual modules, such as transaction management, work correctly. Integration Testing: Verifies the interaction between components, such as UI and database. User Testing: Real users test the app to provide feedback on usability and functionality. Debugging tools like Flutter’s debug Print and the Firebase console help identify and resolve issues during development.

## 5.2 Code:

**HTML PAGE**

**LOGIN PAGE:**

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="css/admin_login.css">
    <title>Admin Login</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <script rel="javascript" src="js/admin_login.js">
    </script>

  </head>

  <body>
    <h1 id="portal">Online Library Portal</h1>
    <hr>
    <button id="back_button" onclick="location.href='index.html'">Back</button>
    <div class="login-page">
          <div class="form">
            <hr>
            <h3>Admin Authentication</h3>
            <hr>
            <br>
            <!-- login  -->
            <form class="login-form">
              <input type="text" placeholder="username" id = "username" required/>
              <input type="password" placeholder="password" id = "password" required/>
              <button type="button" id="submit_data"> Login </button>
            </form>
          </div>
    </div>

<!-- The core Firebase JS SDK is always required and must be listed first -->
  <script src="https://www.gstatic.com/firebasejs/6.0.2/firebase-app.js"></script>
  <script defer src="https://www.gstatic.com/firebasejs/6.0.2/firebase-firestore.js"></script>
  <script defer src="https://www.gstatic.com/firebasejs/6.0.2/firebase-auth.js"></script>

</body>
</html>
```

**ADMIN_LOGIN**:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="css/admin_login.css">
    <title>Admin Login</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <script rel="javascript" src="js/admin_login.js">
    </script>

  </head>

  <body>
    <h1 id="portal">Online Library Portal</h1>
    <hr>
    <button id="back_button" onclick="location.href='index.html'">Back</button>
    <div class="login-page">
          <div class="form">
            <hr>
            <h3>Admin Authentication</h3>
            <hr>
            <br>
            <!-- login  -->
            <form class="login-form">
              <input type="text" placeholder="username" id = "username" required/>
              <input type="password" placeholder="password" id = "password" required/>
              <button type="button" id="submit_data"> Login </button>
            </form>
          </div>
    </div>

  <script src="https://www.gstatic.com/firebasejs/6.0.2/firebase-app.js"></script>
  <script defer src="https://www.gstatic.com/firebasejs/6.0.2/firebase-
firestore.js"></script>
  <script defer src="https://www.gstatic.com/firebasejs/6.0.2/firebase-auth.js"></script>
```

```
</body>
</html>
```

**JAVASCRIPT**

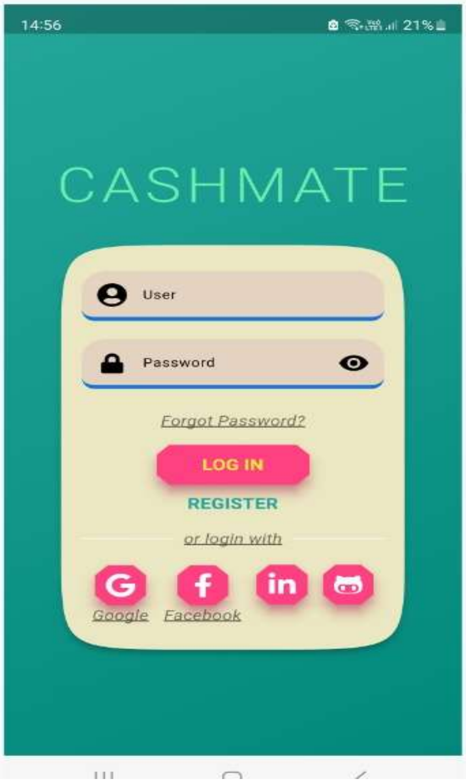
**ADMIN_LOGIN**

```
//admin@123

$(document).ready(function(){
  var firebaseConfig = {
    apiKey: "AIzaSyBin1evT-H6jfR49WIhtVPsGMLzbEklIQY",
    authDomain: "library-management-syste-f2a85.firebaseapp.com",
    databaseURL: "https://library-management-syste-f2a85.firebaseio.com",
    projectId: "library-management-syste-f2a85",
    storageBucket: "library-management-syste-f2a85.appspot.com",
    messagingSenderId: "914416876417",
    appId: "1:914416876417:web:bf9e7762c1c283ba"
   };
   // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
  var db = firebase.firestore();

  $("#login-form").submit(function(e) {
    e.preventDefault();
  });

  $('#submit_data').click(function() {
   login();
  });

  $('#back_button').click(function()
  {
    logout();
  });

  firebase.auth().onAuthStateChanged(user => {
    if(user) {
         window.location = 'admin_portal.html'; //After successful login, user will be redirected to home.html
      }
    });

 });

function login(){
   var email = document.getElementById("username").value;
   var password = document.getElementById("password").value;

   if(email === "admin@gmail.com")
   {
       firebase.auth().signInWithEmailAndPassword(email, password).catch(function(error)
{
         // Handle Errors here.
         var errorCode = error.code;
         var errorMessage = error.message;
         window.alert(errorMessage);
      });
   }

}
function logout()
{
   firebase.auth().signOut().then(function() {
   // Sign-out successful.
   }).catch(function(error) {
   // An error happened.
   });
}
```

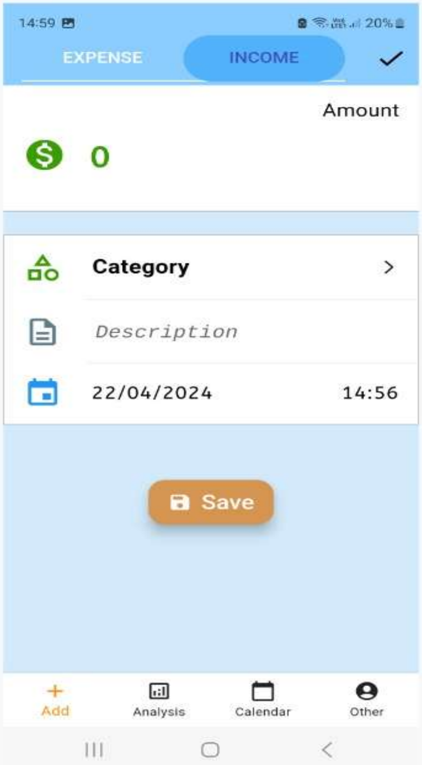
**CSS_LOGIN PAGE**

```
@import url(https://fonts.googleapis.com/css?family=Roboto:300);

#portal{
   text-align: center;
   font-size:300%;
   font-family: "Trebuchet MS", Helvetica, sans-serif;
}

.login-page {
  width: 360px;
  padding: 8% 0 0;
  margin-left: 35%;
  margin-top: -10%;
}
.form {
  position: relative;
  z-index: 1;
  background: #FFFFFF;
  max-width: 360px;
  margin: 0 auto 100px;
  padding: 45px;
  text-align: center;
  box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0 rgba(0, 0, 0, 0.24);
}
.form input {
  font-family: "Roboto", sans-serif;
  outline: 0;
  background: #f2f2f2;
  width: 100%;
  border: 0;
  margin: 0 0 15px;
  padding: 15px;
  box-sizing: border-box;
  font-size: 14px;
}
button {
  background-color: #86d1e0; /* white */
  border: solid;
  border-radius: 5px;
  color: white;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  cursor: pointer;
  padding: 15px;
}

#back_button:hover{
  background: green;
}
.form button:hover,.form button:active,.form button:focus {
  background: green;
  width: 100%;
}
.form .message {
  margin: 15px 0 0;
  color: #b3b3b3;
  font-size: 12px;
}
.form .message a {
  color: #4CAF50;
  text-decoration: none;
}
.form .register-form {
  display: none;
}
.container {
  position: relative;
  z-index: 1;
  max-width: 300px;
  margin: 0 auto;
}
.container:before, .container:after {
  content: "";
  display: block;
  clear: both;
}
.container .info {
  margin: 50px auto;
  text-align: center;
}
.container .info h1 {
  margin: 0 0 15px;
  padding: 0;
  font-size: 36px;
  font-weight: 300;
  color: #1a1a1a;
}
.container .info span {
  color: #4d4d4d;
  font-size: 12px;
}
.container .info span a {
  color: #000000;
  text-decoration: none;
}
.container .info span .fa {
  color: #EF3B3A;
}
body {
  font-family: "Roboto", sans-serif;
}
```

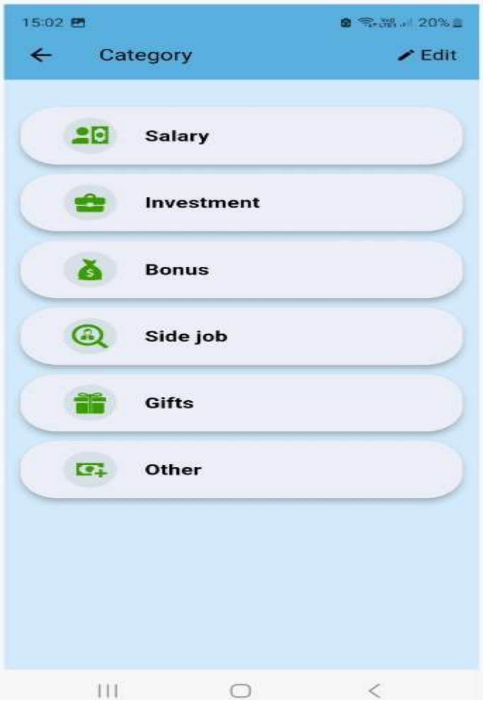
Login Screen



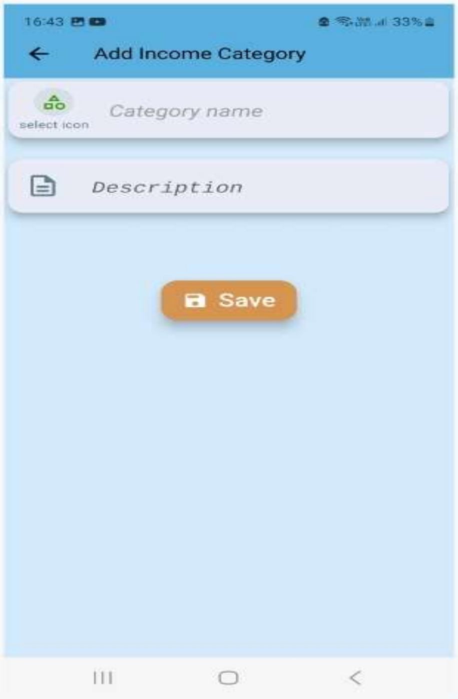
Income Adding Screen



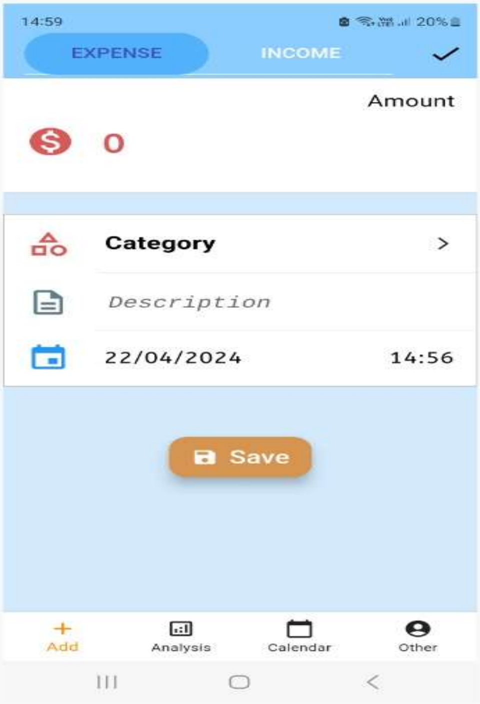
Income Category Screen



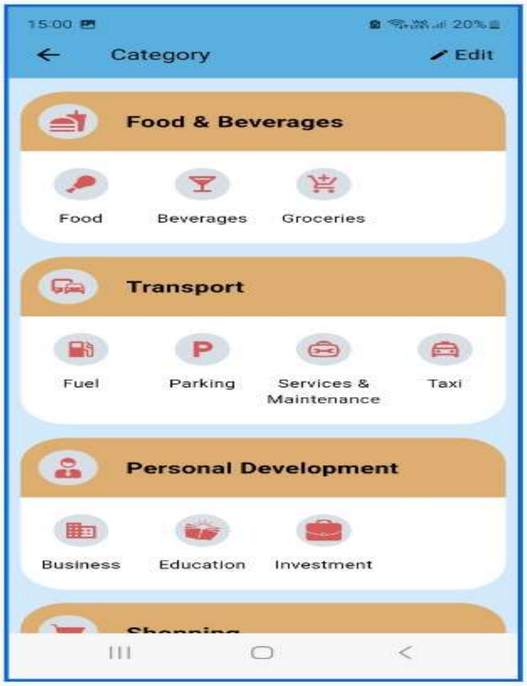
Add Income Category Screen



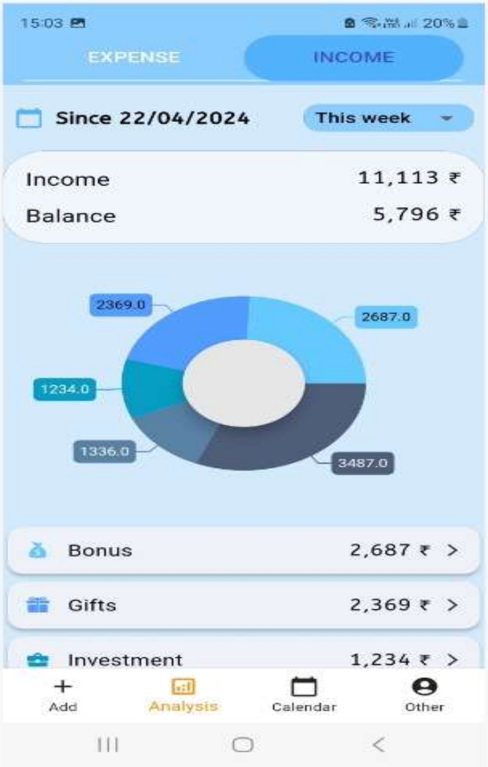
Expense Adding Screen



Expense Category Screen

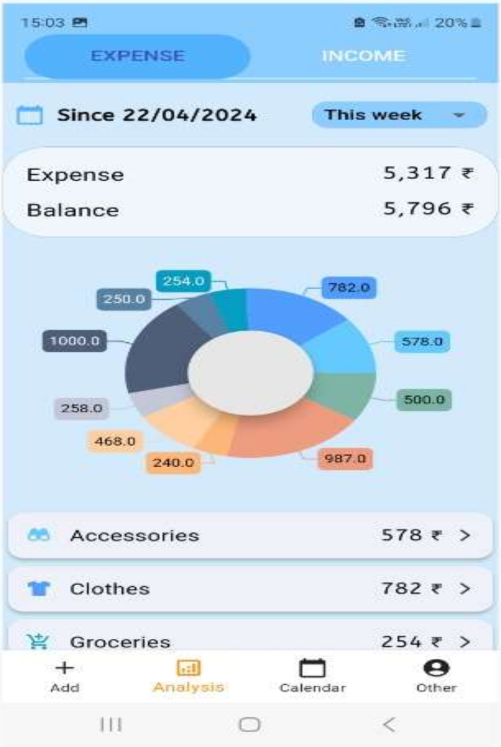


Income Analysis Screen



| | |
|---|---|
| Income | 11,113 ₹ |
| Balance | 5,796 ₹ |
| Bonus | 2,687 ₹ |
| Gifts | 2,369 ₹ |
| Investment | 1,234 ₹ |

Expense Analysis Screen



| | |
|---|---|
| Expense | 5,317 ₹ |
| Balance | 5,796 ₹ |
| Accessories | 578 ₹ |
| Clothes | 782 ₹ |
| Groceries | 254 ₹ |

# Conclusion

Personal Money Assistant is a comprehensive and user-centric solution designed to simplify financial management. It effectively addresses the common challenges faced by users, such as limited offline functionality, lack of customization, and high costs associated with advanced features in many existing applications. By leveraging cutting-edge technologies like Flutter for cross-platform compatibility, Firebase for secure authentication, and SQLite for offline data storage, CashMate ensures a seamless and reliable experience.

The application stands out for its dynamic features, including customizable categories, transaction descriptions, visual insights through pie charts, calendar-based tracking, and multilingual support. These functionalities not only enhance usability but also empower users to take control of their finances with a personalized approach. Additionally, the app's adherence to the MIT license and commitment to free access make it a valuable tool for a global audience, democratizing financial management tools.

The iterative development process, encompassing rigorous system analysis, thoughtful design, and meticulous implementation, ensures that CashMate is robust, scalable, and adaptable to future enhancements. It addresses the gaps in the market by offering a secure, intuitive, and feature-rich platform for personal finance tracking.

In conclusion, CashMate redefines personal finance management by blending technology, user-focused design, and accessibility. It is not just a financial tool but a step toward fostering financial literacy and empowering individuals to make informed financial decisions. With its strong foundation and adaptability, CashMate has the potential to grow and evolve, continuing to meet the changing needs of its users effectively.

# Reference

**1.***The Daily Star*, [online] Available: https://www.thedailystar.net/supplements/30th-anniversary-supplements/aspirations-the-next-50-years/news/bangladesh-and-the-ongoing-technological-revolution-2047309.

**2.***The Business Standard* Available: https://www.tbsnews.net/thoughts/quest-technology-driven-economic-development-222340.

*3.The Daily Star*, [online] Available: https://www.thedailystar.net/business/news/bangladesh-economy-grows-725-fy22-bbs-3020436.

**4.***The Daily Observer*, [online] Available: https://www.observerbd.com/news.php?id=397415.