

**Tribhuvan University**

**Faculty of Humanities and Social Science**

**PocketPlan**

**A PROJECT REPORT**

**Submitted to:**

**Department of Computer Application**

**Swoyambhu International College**

Submitted by:

Roshan Shakya (6-2-927-48-2022)

Bibek Shenchuri (6-2-927-29-2022)

October, 2024

Under the supervision of

**Mr. Sujit Gyawali**



**Tribhuvan University**

**Faculty of Humanities and Social Science**

**Swoyambhu International College**

**Supervisor’s Recommendation**

I hereby recommend that this project prepared under my supervision by Roshan Shakya and Bibek Shenchuri “PocketPlan**”** in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

**SIGNATURE**

Mr. Sujit Gyawali

**SUPERVISOR**

Department of BCA

Swoyambhu International College

Lagankhel, Lalitpur



**Tribhuvan University**

**Faculty of Humanities and Social Science**

**Swoyambhu International College**

**LETTER OF APPROVAL**

This is to certify that this project was prepared by Roshan Shakya and Bibek Shenchuri. “PocketPlan**”** in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

|  |  |
| --- | --- |
| ………………………..……………….. | …………………………………………… |
| **Signature of Supervisor** | **Signature of Coordinator** |
| **Mr. Sujit Gyawali** | **Mr. Raj Kumar Sah** |
| Lecturer | Coordinator |
| Swoyambhu International College | Swoyambhu International College |
| Lagankhel, Lalitpur | Lagankhel, Lalitpur |
|  |  |
| ………………………………………… | …………………………………………… |
| **Signature of Internal Examiner** | **Signature of External Examiner** |
|  |  |

# ABSTRACT

PocketPlan is an innovative and user-friendly budget tracking application designed to simplify personal finance management and empower users with better control over their money. With an intuitive interface, users can effortlessly sign up, log in, and access a personalized financial dashboard to record income, track expenses, set budget limits, and monitor transaction history. Its automated tracking system categorizes expenses, eliminating the hassle of manual calculations while providing real-time insights into spending habits. Additionally, visual analytics, such as charts and graphs, make financial data easily understandable, helping users make informed decisions and develop smarter budgeting strategies. Whether managing daily expenses, saving for specific goals, or planning long-term financial stability, PocketPlan serves as a reliable financial companion that fosters better money management and simplifies financial tracking. By transforming budgeting into a seamless and insightful experience, PocketPlan helps users achieve financial security with minimal effort.

**Keywords**: *Budget, Expenses, Income, Personal Finance, Financial Stability*

# ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our Project Management Instructor/ Supervisor, Mr. Sujit Gyawali, for his invaluable guidance and continuous support throughout this project. His insights and encouragement played a crucial role in helping us navigate challenges and successfully complete our work.

We are also thankful to the entire faculty for their valuable recommendations and feedback, which enriched our learning experience and strengthened our understanding of project management principles.

Additionally, we extend our heartfelt appreciation to our families and friends for their unwavering support, motivation, and patience. Their encouragement kept us focused and determined throughout this journey.

Lastly, we are grateful to everyone who contributed, guided, or motivated us in any way. Their support has been instrumental in making this project a fulfilling and successful endeavor.

Roshan Shakya,

Bibek Shenchuri.

BCA, 4th Semester

Swoyambhu International College

**CONTENTS**

[ABSTRACT iv](#_Toc193876843)

[ACKNOWLEDGEMENT v](#_Toc193876844)

[LIST OF FIGURES viii](#_Toc193876845)

[LIST OF ABBREVIATION ix](#_Toc193876846)

[LIST OF TABLES x](#_Toc193876847)

[CHAPTER 1: INTRODUCTION TO PROJECT 1](#_Toc193876848)

[1.1 Introduction 1](#_Toc193876849)

[1.2 Problem Statement 2](#_Toc193876850)

[1.3 Objective 2](#_Toc193876851)

[1.4 Scope and limitation 2](#_Toc193876852)

[1.4.1 Scope 2](#_Toc193876853)

[1.5 Report Organization 3](#_Toc193876854)

[CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW 4](#_Toc193876855)

[2.1 Background Study 4](#_Toc193876856)

[2.2 Literature Review 4](#_Toc193876857)

[CHAPTER 3: SYSTEM ANALYSIS AND DESIGN 6](#_Toc193876858)

[3.1 System Analysis 6](#_Toc193876859)

[3.1.2 Feasibility Analysis 9](#_Toc193876860)

[3.1.3 Data Modeling (ER-Diagram) 12](#_Toc193876861)

[3.1.4 Process Modeling (DFD) 14](#_Toc193876862)

[3.2.1 Architectural Design 15](#_Toc193876863)

[3.2.2 Database Schema 16](#_Toc193876864)

[3.2.3 Interface Design (UI/UX) 17](#_Toc193876865)

[3.2.3 Physical DFD 23](#_Toc193876866)

[CHAPTER 4: IMPLEMENTATION AND TESTING 24](#_Toc193876867)

[4.1 IMPLEMENTATION 24](#_Toc193876868)

[4.1.1 Tools Used 24](#_Toc193876869)

[4.1.2 Implementation Details of Modules 25](#_Toc193876870)

[4.2 Testing 26](#_Toc193876871)

[4.2.1 Test Case for Unit Testing 26](#_Toc193876872)

[CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATIONS 29](#_Toc193876873)

[5.1. Lesson Learnt / Outcome 29](#_Toc193876874)

[5.2. Conclusion 29](#_Toc193876875)

[5.3. Future Recommendation 29](#_Toc193876876)

[REFERENCES 31](#_Toc193876877)

[APPENDICES 32](#_Toc193876878)

# LIST OF FIGURES

[Figure 3. 1: Waterfall Software Development Model……………………………………. 7](#_heading=h.c0al8hyyipsw)

[Figure 3. 2: Use Case Diagram of PocketPlan ………………………………………….... 8](#_heading=h.y24sm59d1a4)

[Figure 3. 3: Gantt Chart for PocketPlan ………………………………………………... 11](#_heading=h.6tcighqpjdd8)

[Figure 3. 4: ER-Diagram of PocketPlan … ………………………………………………12](#_heading=h.qdji34to2c60)

[Figure 3. 5: Level 0 DFD of PocketPlan ………………………………………………...14](#_heading=h.5m9x49bcfotv)

[Figure 3. 6: Level 1 DFD of PocketPlan ……………………………………………….. 14](#_heading=h.kd2xnqgj1o3j)

[Figure 3. 7: 3-Tier Architecture of PocketPlan …………………………………………. 15](#_heading=h.90ocn1qqv5r)

[Figure 3. 8: Database Schema of PocketPlan …………………………………………... 16](#_heading=h.sj0jra82h1d0)

[Figure 3. 9: Auth Pages of PocketPlan …………………………………………………. 17](#_heading=h.2yqcq2mfkm1r)

[Figure 3. 10: Dashboard Page of PocketPlan……………………………………………. 18](#_heading=h.sawf1z1e3nxf)

[Figure 3. 11: Income Page of PocketPlan……………………………………………….. 19](#_heading=h.5r1to6ciqgg9)

[Figure 3. 12: Expense Page of PocketPlan……………………………………………… 20](#_heading=h.lwhzyr1lmlaq)

[Figure 3. 13: Budget Page of PocketPlan……………………………………………….. 21](#_heading=h.bqtekm3pvkt6)

[Figure 3. 14: History Page of PocketPlan………………………………………………. 22](#_heading=h.sdd95jixe6hc)

[Figure 3. 15: Physical DFD of PocketPlan……………………………………………… 23](#_heading=h.vlrbzw6hxjfs)

# LIST OF ABBREVIATION

CRUD: Create, Retrieve, Update, Delete

CSS: Cascading Style Sheet

DFD: Data Flow Diagram

ER-Diagram: Entity Relationship Diagram

HTML: Hypertext Markup Language

JS: JavaScript

MySQL: My Structured Query Language

PHP: Hypertext Preprocessor

SDLC: System Development Life Cycle

UI: User Interface

UX: User Experience

VS Code: Visual Studio Code

XAMPP: Cross(X)-Platform Apache MySQL PHP Perl

# LIST OF TABLES

[Table 1: Scheduling of PocketPlan 2](#_Toc193873990)

[Table 2: Register Page Test Case 2](#_Toc193873991)

[Table 3: Login Page Test Case 2](#_Toc193873992)

[Table 4: Income / Expense Page Test Case 2](#_Toc193873993)

# CHAPTER 1: INTRODUCTION TO PROJECT

## 1.1 Introduction

Managing personal finances efficiently is essential in today’s fast-paced world. PocketPlan is a budget tracking application designed to help users record, manage, and analyze their income and expenses effectively. The goal of this system is to provide an easy-to-use platform where users can log financial transactions, categorize expenses, and set budget limits for better financial planning.

The application follows a structured approach using the Waterfall Model for system development, ensuring clarity in requirements and a systematic workflow. The development process includes phases such as Requirement Gathering, System Design, Implementation, Integration & Testing, and System Documentation. These stages help in delivering a well-structured and reliable financial tracking system.

PocketPlan offers key features such as income and expense tracking, a budget allocation system, and a transaction history overview. The system allows users to register and log in securely, ensuring data protection. Once logged in, users can add income details, categorize expenses, and monitor their financial activities through an interactive dashboard. The system stores data efficiently in a MySQL database, ensuring quick access and smooth operations.

This project is developed as part of a structured academic initiative to enhance financial literacy and budgeting habits. By leveraging PHP for backend operations, MySQL for database management, and a responsive frontend, PocketPlan aims to be a comprehensive and user-friendly budget tracking tool. The structured development approach ensures smooth implementation and reliable functionality, making it a practical solution for effective financial management.

## 1.2 Problem Statement

Many people face difficulties in managing their finances due to the lack of a structured and automated system. Traditional methods such as manual record-keeping or using spreadsheets can be time-consuming and prone to errors. Some common challenges include:

* Lack of a centralized system to record transactions.
* Difficulty in tracking expenses and income over time.
* No clear visualization of financial habits and trends.

## 1.3 Objective

The platform aims to provide users with an easy-to-use, web-based tool for tracking income and expenses. By allowing users to set and manage budgets across various categories, it helps them stay on top of their finances and maintain control over their spending. With a focus on security, the system ensures that financial records are safely stored while remaining accessible for users to review and manage at any time. The primary objectives of PocketPlan are:

* To provide a user-friendly web-based platform for tracking income and expenses.
* To allow users to set and manage budgets for different categories.
* To maintain a secure and accessible financial record for users.

## 1.4 Scope and limitation

Every application comes with its own set of features and limitations. The PocketPlan project offers the following scope and has certain restrictions:

### 1.4.1 Scope

Our finance management system ensures secure login and registration while keeping user data safe. Users can track income and expenses under different categories and set budgets to manage spending effectively. A clear transaction history helps in reviewing past financial activities, and the simple interface makes tracking finances easy and accessible.

* Users can log in and register securely.
* Ability to record income and expenses under different categories.
* Users can allocate budgets to specific expense categories.

**1.4.2 Limitation**

The current version of the system doesn’t include advanced data visualization like graphs and charts, but future updates may add these features for better insights. It also doesn’t support multi-user accounts or collaborative budgeting, though this is planned for future releases. Automation for recurring expenses and income tracking is not available yet, but it may be introduced in later versions. Additionally, future updates could integrate AI-powered financial insights to help users optimize their finances

* No advanced data visualization (graphs and charts) in the initial version.
* Does not support multi-user accounts or collaborative budgeting.
* Lacks automation for recurring expenses and income tracking.

## 1.5 Report Organization

This report consists of five chapters. The first chapter introduces the project, outlining its objectives, scope, and limitations. It provides an overview of the problem being addressed and the significance of developing an effective solution. The second chapter provides a background study and literature review related to financial management systems, highlighting existing solutions, industry standards, and key challenges. It explores relevant theories, methodologies, and technologies that form the foundation of the proposed system. The third chapter discusses the system analysis and design, including requirement analysis, feasibility study, and system architecture. It details the functional and non-functional requirements, system components, and the overall structure that ensures scalability and efficiency. The fourth chapter covers implementation and testing, describing the development process, tools used, and methods applied to validate system performance. The final chapter presents the conclusion and future recommendations, summarizing key findings, evaluating system effectiveness, and suggesting potential improvements or further research directions.

# CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

## 2.1 Background Study

Financial management is a critical aspect of modern life, and with the rise of digital tools, individuals are increasingly relying on technology to manage their finances. However, many existing solutions either lack simplicity or are overloaded with features that make them inaccessible to the average user. PocketPlan aims to bridge this gap by offering a balanced solution that combines ease of use with essential budgeting features.

The concept of budget tracking and expense management has evolved significantly over the past decade. With the advent of smartphones and the internet, financial management tools have become more accessible to the general public. However, the challenge lies in creating a tool that is both powerful and easy to use. Many users find existing tools either too basic or too complex, leading to a gap in the market for a solution that strikes the right balance.

In the context of Nepal, where digital literacy is growing but still limited in some areas, a simple and intuitive financial management tool like PocketPlan can have a significant impact. By providing a platform that is easy to navigate and requires minimal technical knowledge, PocketPlan can help users take control of their finances without feeling overwhelmed.

## 2.2 Literature Review

Financial management tools have been the subject of extensive research and development over the years. Below is a review of existing tools, research findings, and trends in the field:

1. **Basic Budgeting Apps**

* Examples: Mint, YNAB (You Need A Budget), and Wallet.
* Features: These apps focus on tracking income and expenses, categorizing transactions, and providing basic budgeting tools.
* Limitations: While they are user-friendly, they often lack advanced features such as predictive analytics, multi-user support, or detailed financial insights.

1. **Comprehensive Financial Tools**

* Examples: Quicken, Personal Capital, and PocketGuard.
* Features: These tools offer advanced functionalities such as investment tracking, retirement planning, and AI-driven financial insights.
* Limitations: Their complexity can overwhelm users who only need basic budgeting features, and they often come with a steep learning curve.

1. **PocketPlan’s Contribution**

PocketPlan addresses these gaps by focusing on simplicity, accessibility, and essential features. It provides a user-friendly interface for tracking income and expenses, allocating budgets, and viewing transaction history. By avoiding unnecessary complexity, PocketPlan ensures that users can easily adopt and maintain the system.

Additionally, PocketPlan incorporates a secure authentication system to protect user data, addressing the growing concern around data privacy. The platform also includes a feedback system that allows users to provide suggestions for improvement, ensuring that the tool evolves to meet user needs.

1. **Relevance in the Nepalese Context**

In Nepal, where digital transformation is still in its early stages, PocketPlan can play a crucial role in promoting financial literacy and responsibility. By providing a tool that is both easy to use and effective, PocketPlan can help individuals and families take control of their finances, leading to better financial outcomes.

Furthermore, the platform’s focus on simplicity makes it accessible to users with varying levels of technical expertise, ensuring that it can be adopted by a wide range of individuals, from students to working professionals.

# CHAPTER 3: SYSTEM ANALYSIS AND DESIGN

## 3.1 System Analysis

Given that this project involves the design and implementation of a software system, it is essential to carefully consider the various models used in software development and deployment before proceeding with further development.

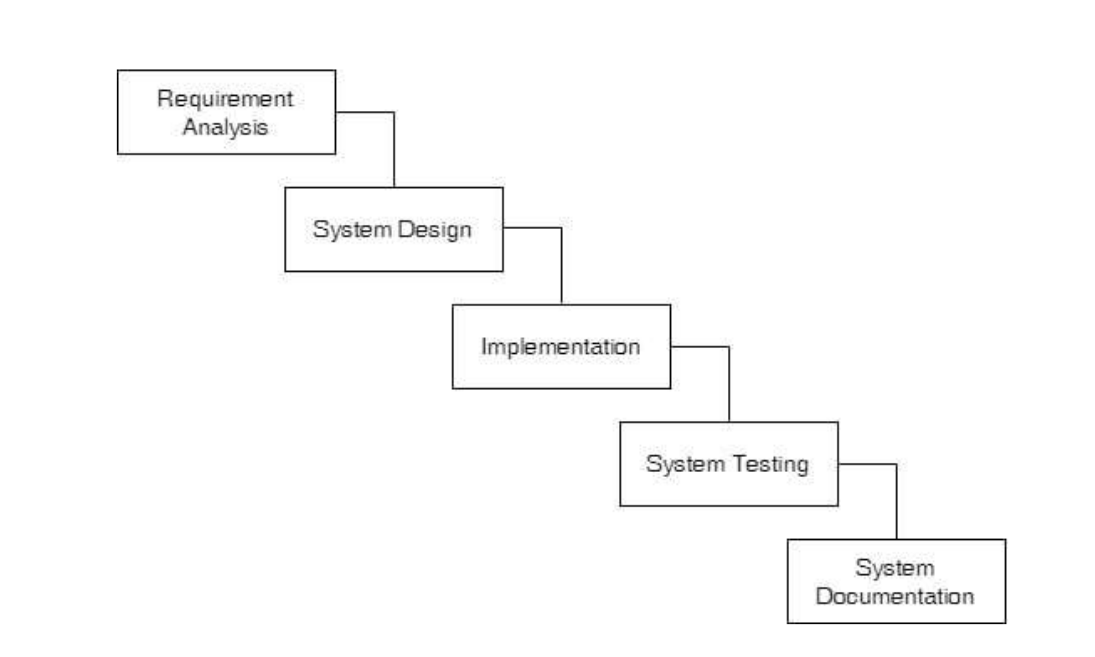
For the construction of this website, we have employed the waterfall model. This model was chosen because it allows us to focus on each component individually during its development, ensuring a structured and systematic approach. By following this model, the project can be divided into distinct and manageable phases, making it easier to track progress and address issues at each stage.

The PocketPlan website has been developed using the waterfall model. While the waterfall model is typically linear, we have incorporated feedback loops to allow for adjustments and improvements as needed. Given that only two developers are responsible for this project, the waterfall model aligns well with our requirements. The following stages were followed during the software development process:

* Requirement Analysis
* System Design
* Implementation
* System Testing
* System Documentation

The waterfall methodology was selected for several reasons. First, the project has a defined timeframe, and the waterfall model is well-suited for projects with clear deadlines. Second, the project involves simple CRUD (Create, Retrieve, Update, Delete) operations, which are predictable and straightforward to implement. The waterfall model is ideal for such projects, as the outcomes are known in advance, and the process flows in a logical, step-by-step manner.

Additionally, the project’s simplicity and the small team size made the waterfall model a practical choice. With a clear understanding of the project’s goals and a well-defined process, the waterfall methodology provided a structured framework that ensured efficiency and clarity throughout the development process.



**Figure 3. 1: Waterfall Software Development Model**

**3.1.1 Requirement Analysis**

**Functional Requirements**

Functional requirements define what the system should do to fulfill its intended purpose. PocketPlan includes the following key functionalities:

#### **1. User Authentication**

* Users can register and create an account.
* Users can log in securely using their credentials.
* Users can log out of the system.

#### **2. Income and Expense Tracking**

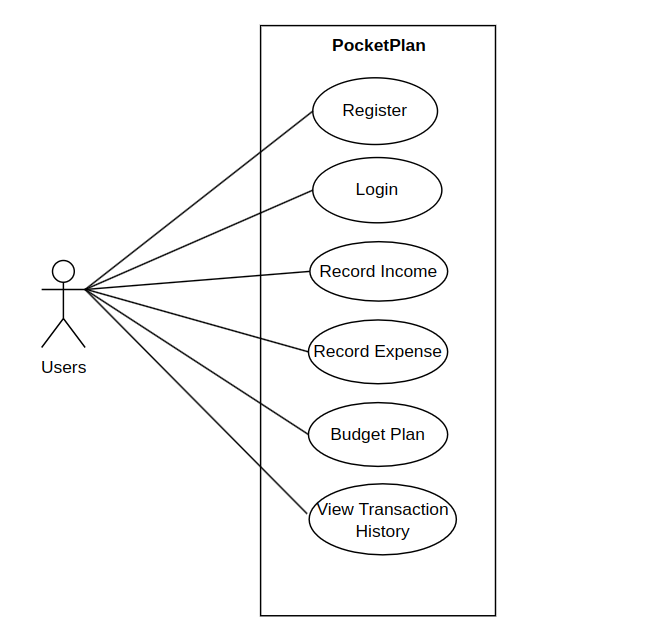
* Users can add, update, and delete income and expense records.
* Users can categorize transactions (e.g., salary, rent, groceries).

#### **3. Budget Management**

* Users can set monthly budgets for different categories (e.g., food, entertainment).
* Users can view their budget usage and remaining balance.

**4. Transaction History**

* Users can view a detailed history of their transactions.
* Users can filter transactions by date, category, or type (income/expense).



**Figure 3. 2: Use Case Diagram of PocketPlan**

**Non-Functional Requirements**

Non-functional requirements define how the system should perform, focusing on usability, security, performance, and availability.

#### **1. Usability**

* The system should have an intuitive and user-friendly interface.
* The system should be accessible to users with minimal technical knowledge.

**2. Performance**

* The system should respond to user actions within 2-3 seconds.
* The system should be able to handle up to 1,000 concurrent users without performance degradation.

**3. Security**

* User data should be encrypted and stored securely.
* The system should implement password hashing and secure authentication mechanisms.

#### **4. Availability**

* The system should be available 24/7 with minimal downtime.
* Regular backups should be performed to prevent data loss.

### 3.1.2 Feasibility Analysis

A feasibility analysis was conducted to assess the practicality of developing PocketPlan in terms of technical, operational, economic, and schedule feasibility.

#### **Technical Feasibility**

* The project is built using widely available and well-supported technologies such as PHP**,** MySQL, HTML, CSS, and JavaScript, making it technically feasible.
* The development team has the necessary skills and experience to implement and maintain the system effectively.

#### **Operational Feasibility**

* The system is designed to be user-friendly, ensuring easy adoption by users without requiring extensive training.
* The platform can be maintained with minimal resources, as most operations, such as recording transactions and managing budgets, are automated.

**Economic Feasibility**

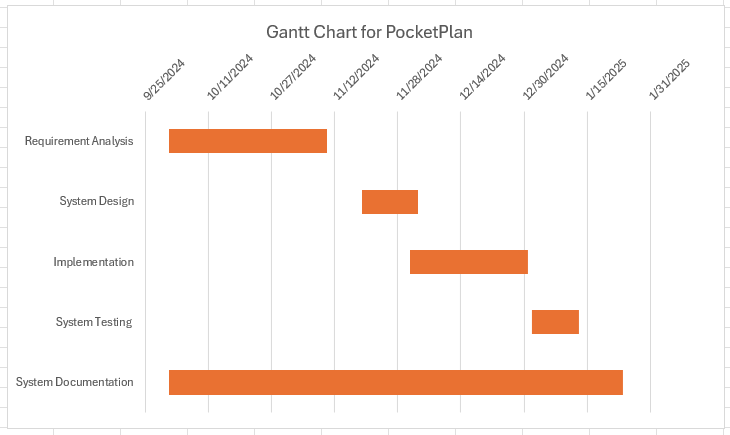
* The development costs are low, as the project relies on open-source tools and technologies.
* The system provides value to users by helping them track and manage their finances efficiently, making it economically viable.

**Schedule Feasibility**

* The project was completed within the allocated time frame using a structured waterfall model with clear milestones.
* The development plan ensured a smooth workflow, allowing each phase to be completed without major delays.

**Table 1: Scheduling of PocketPlan**

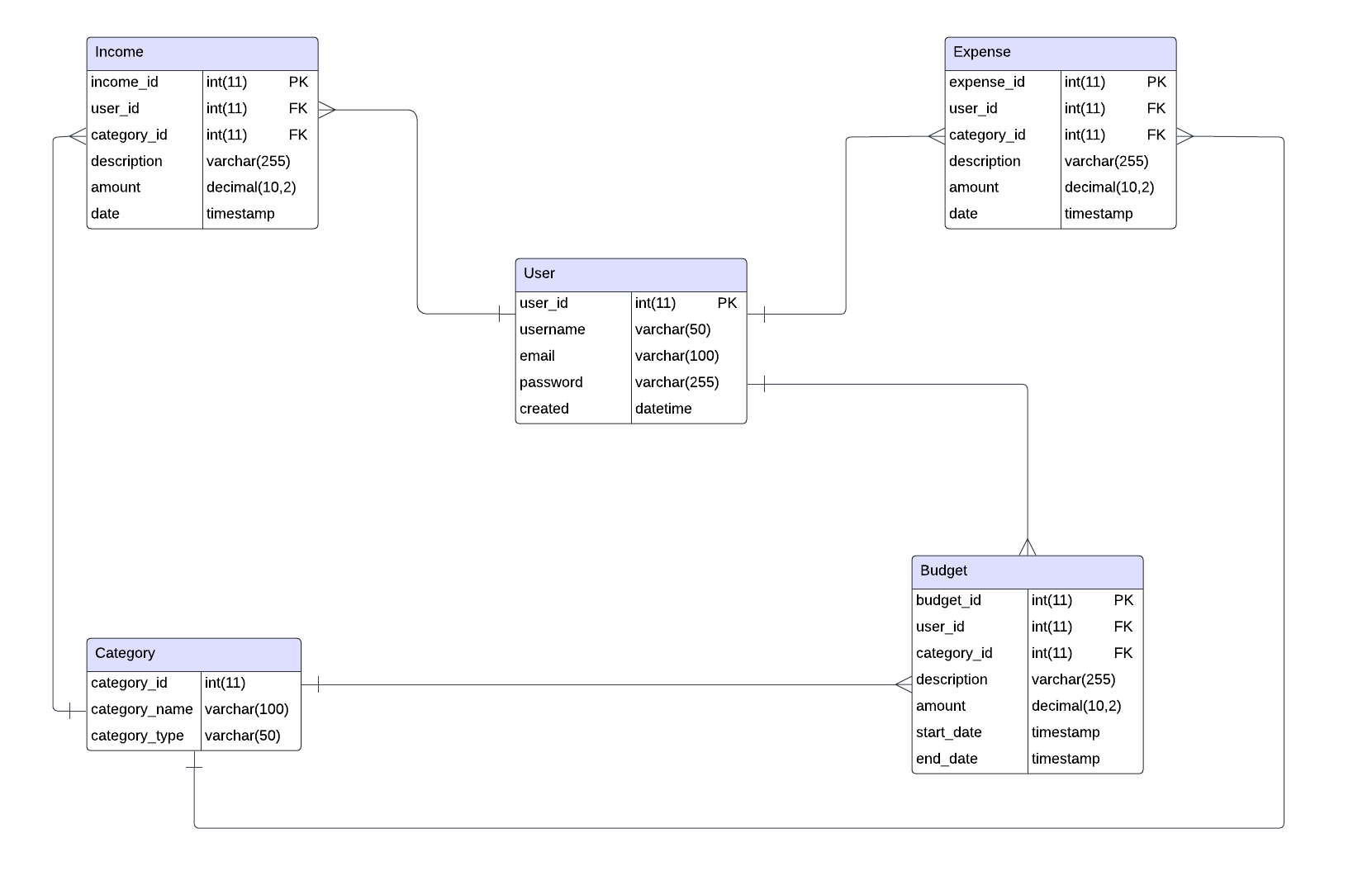
|  |  |  |
| --- | --- | --- |
| **WORK** | **Starting Date** | **Ending Date** |
| Requirement Gathering | 1-Oct-24 | 18-Nov-24 |
| System Design | 19-Nov-24 | 29-Nov-24 |
| Implementation | 01-Dec-24 | 30-Dec-24 |
| Integration and Testing | 1-Jan-25 | 13-Jan-25 |
| System Documentation | 1-Oct-24 | 25-Jan-25 |



**Figure 3. 3: Gantt Chart for PocketPlan**

The PocketPlan budget tracking app follows the Waterfall Model of SDLC for a structured development process. The project begins with Requirement Gathering (1st Oct–18th Nov 2024), followed by System Design (19th Nov–29th Nov 2024). Implementation runs from (1st Dec–30th Dec 2024), then Integration and Testing from 1st Jan–13th Jan 2025. Finally, System Documentation is completed between (1st Oct 2024–25th Jan 2025). This timeline ensures an organized and efficient development process.

### 3.1.3 Data Modeling (ER-Diagram)



**Figure 3. 4: ER-Diagram of PocketPlan**

The Entity-Relationship (ER) Diagram for PocketPlan represents the structure of the database and the relationships between different entities. It ensures that data is organized efficiently, allowing seamless storage, retrieval, and management of financial records.

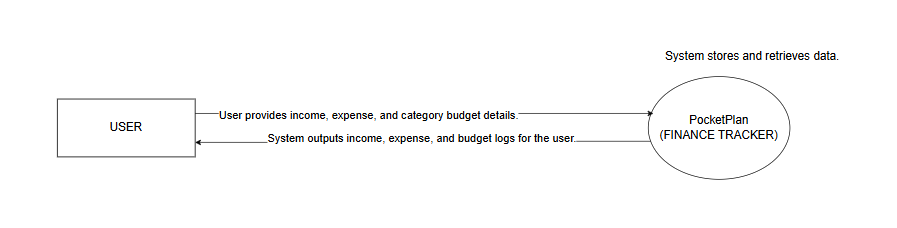
Entities and Relationships

1. **User**
   * Stores user-related information such as User ID, Username, Email, Password, and Created (date of account creation).
   * Acts as the central entity, linking to all financial data.
2. **Income**
   * Records income transactions with fields: Income ID, User ID, Category ID, Amount, Description, and Date.
   * User ID (FK) links income records to the respective user.
   * Category ID (FK) ensures each income entry belongs to a specific category.
3. **Expenses**
   * Stores expense records, including Expense ID, User ID, Category ID, Amount, Description, and Date.
   * User ID (FK) associates’ expenses with a specific user.
   * Category ID (FK) ensures proper categorization of expenses.
4. **Category**
   * Defines different categories of transactions with fields: Category ID, Category Name, and Category Type.
   * Connected to both Income and Expenses, ensuring transactions are categorized correctly.
5. **Budget**
   * Manages budget allocations with fields: Budget ID, User ID, Category ID, Amount, Start Date, and End Date.
   * User ID (FK) ensures each budget is user-specific.
   * Category ID (FK) links budget allocations to specific categories, ensuring structured financial planning.

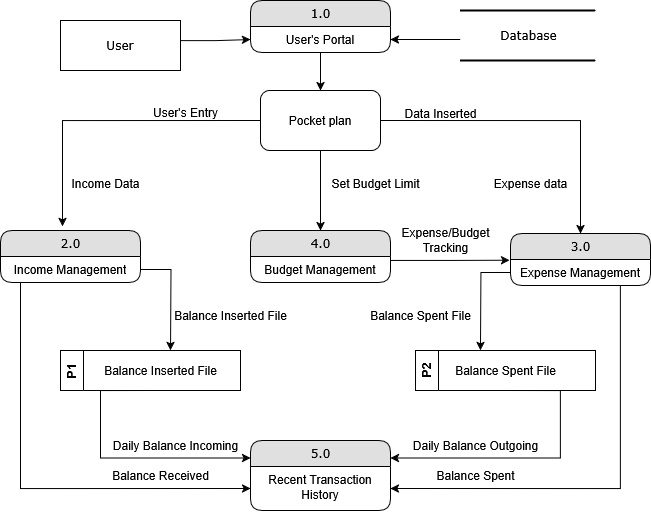
**Relationships in the ER Diagram**

* **One-to-Many Relationship between User and Transactions (Income & Expenses):**
  + A User can have multiple Income and Expense records, but each transaction belongs to only one User.
* **One-to-Many Relationship between User and Budget:**
  + A User can set multiple Budgets, but each budget entry is associated with a single User.
* **One-to-Many Relationship between Category and Transactions (Income & Expenses):**
  + Each Category can be used for multiple Income and Expense records, but each transaction belongs to a single Category.
* **One-to-Many Relationship between Category and Budget:**
  + Each Category can have multiple budgets assigned, ensuring proper financial planning.

### 3.1.4 Process Modeling (DFD)



**Figure 3. 5: Level 0 DFD of PocketPlan**



**Figure 3. 6: Level 1 DFD of PocketPlan**

The Data Flow Diagram (DFD) visually represents how data moves through the system.

* **Level 0 DFD:** Shows high-level interactions between users, the system, and the database.
* **Level 1 DFD:** Breaks down the processes into user registration, transaction recording, budget management, and feedback submission.

The DFD helps visualize the flow of information within the system, ensuring that all critical processes are well-defined and accounted for.

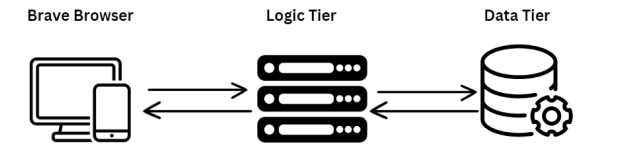
**3.2 System Design**

The System Design phase translates the requirements into a functional structure for implementation. This includes architectural design, database schema, user interface (UI/UX) design, and system interactions.

### 3.2.1 Architectural Design

PocketPlan follows a 3-tier architecture, ensuring modularity and scalability:

1. **Presentation Layer:**
   * Built using HTML, CSS, and JavaScript, this layer provides a responsive and user-friendly interface for interaction.
2. **Application Layer:**
   * Developed in PHP, this layer handles business logic such as user authentication, transaction processing, and budget calculations.
3. **Data Layer:**
   * A MySQL database manages all data related to users, transactions, and budgets.



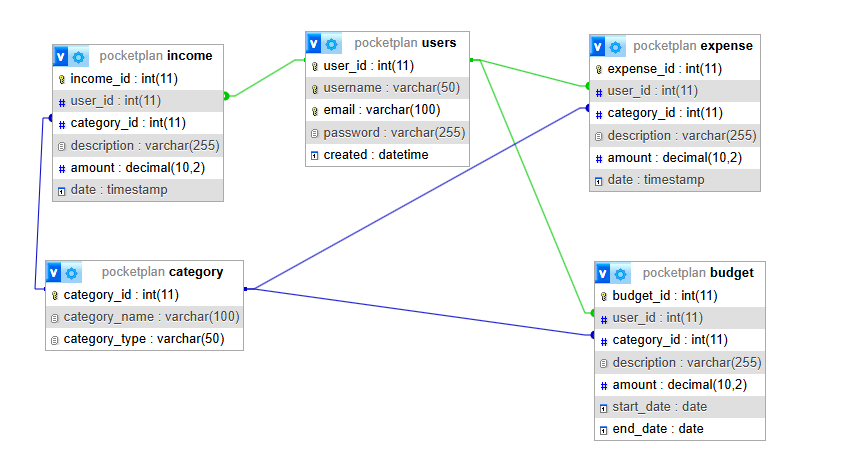
**Pocket Plan** pocketplan

**Figure 3. 7: 3-Tier Architecture of PocketPlan**

This structured approach ensures the system is efficient, scalable, and easy to maintain.

### 3.2.2 Database Schema

The PocketPlan database consists of the following key tables:

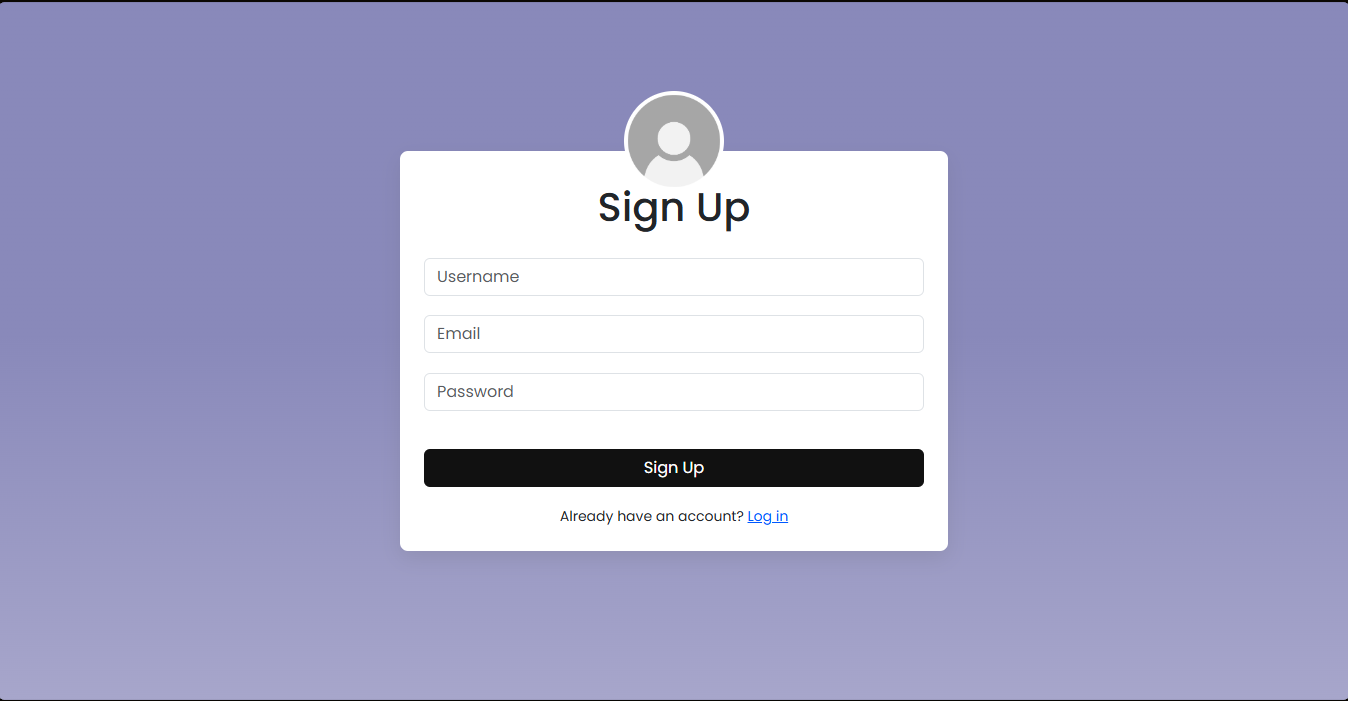


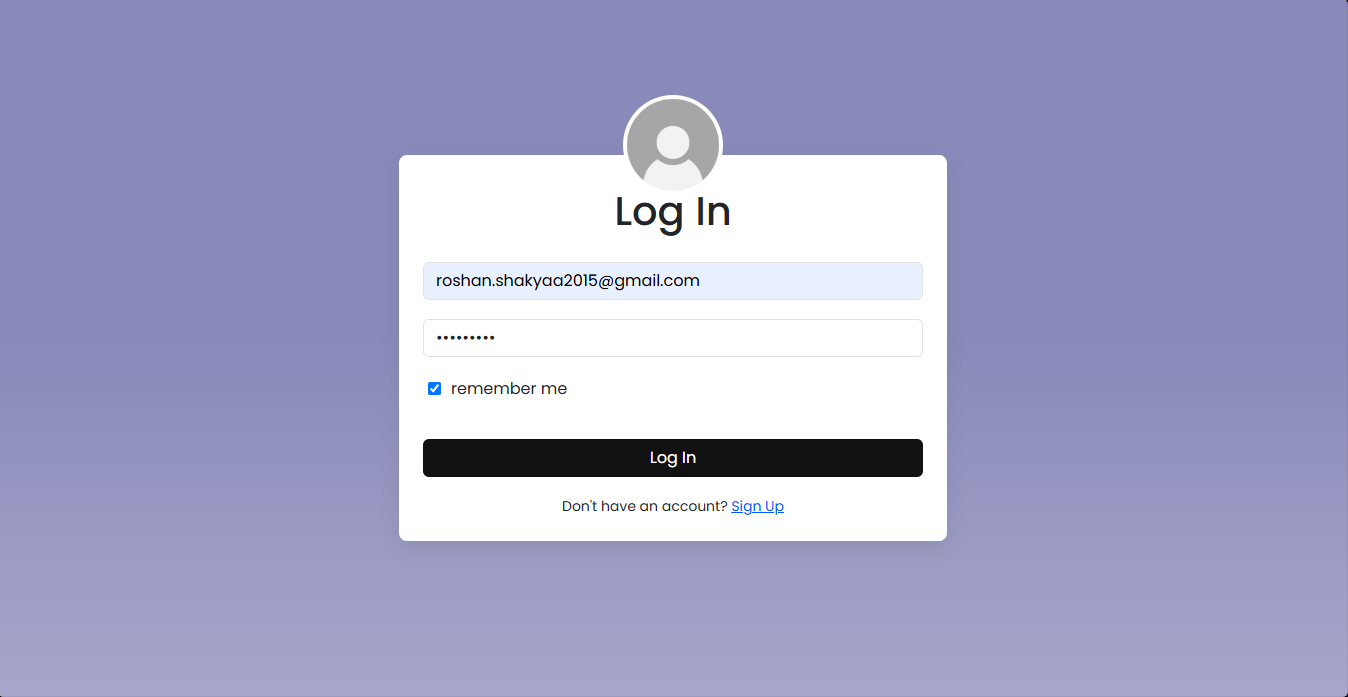
**Figure 3. 8: Database Schema of PocketPlan**

* **Users** – Stores user details including username, email, password, and account creation date.
* **Income** – Records user income transactions with category, amount, description, and date.
* **Expense** – Stores user expense transactions categorized with amount, description, and date.
* **Category** – Defines different categories for income and expenses with name and type.
* **Budget** – Manages user budgets with assigned category, amount, description, and duration

### 3.2.3 Interface Design (UI/UX)

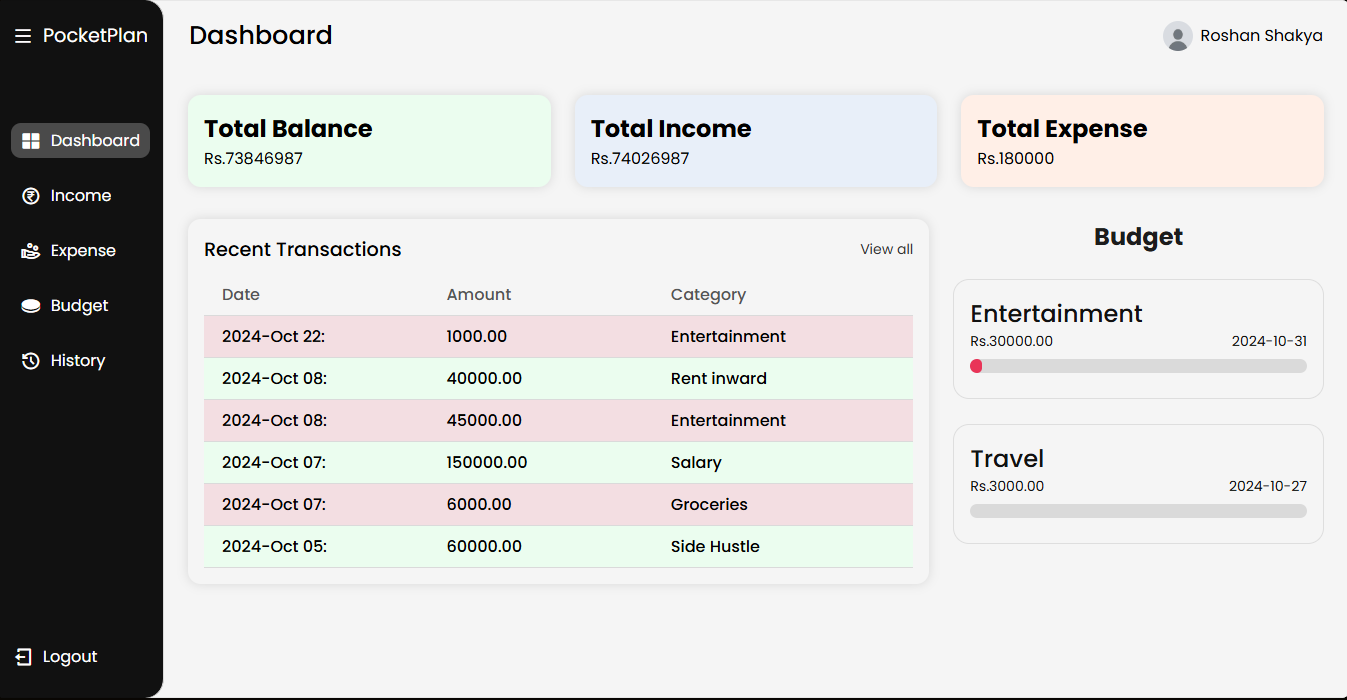
The PocketPlan user interface is designed for simplicity and ease of use.





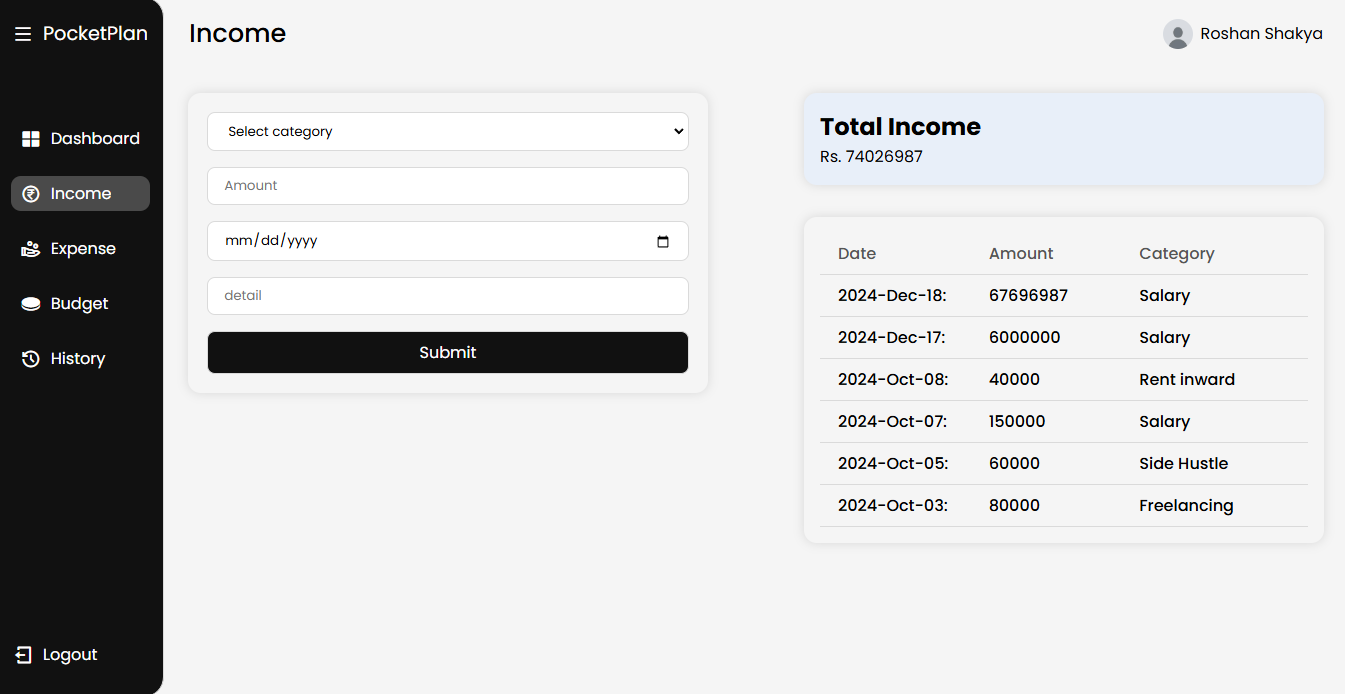
**Figure 3. 9: Auth Pages of PocketPlan**

These pages allow users to sign up for a new account or log in to an existing one securely. It ensures only authorized users can access their financial data.



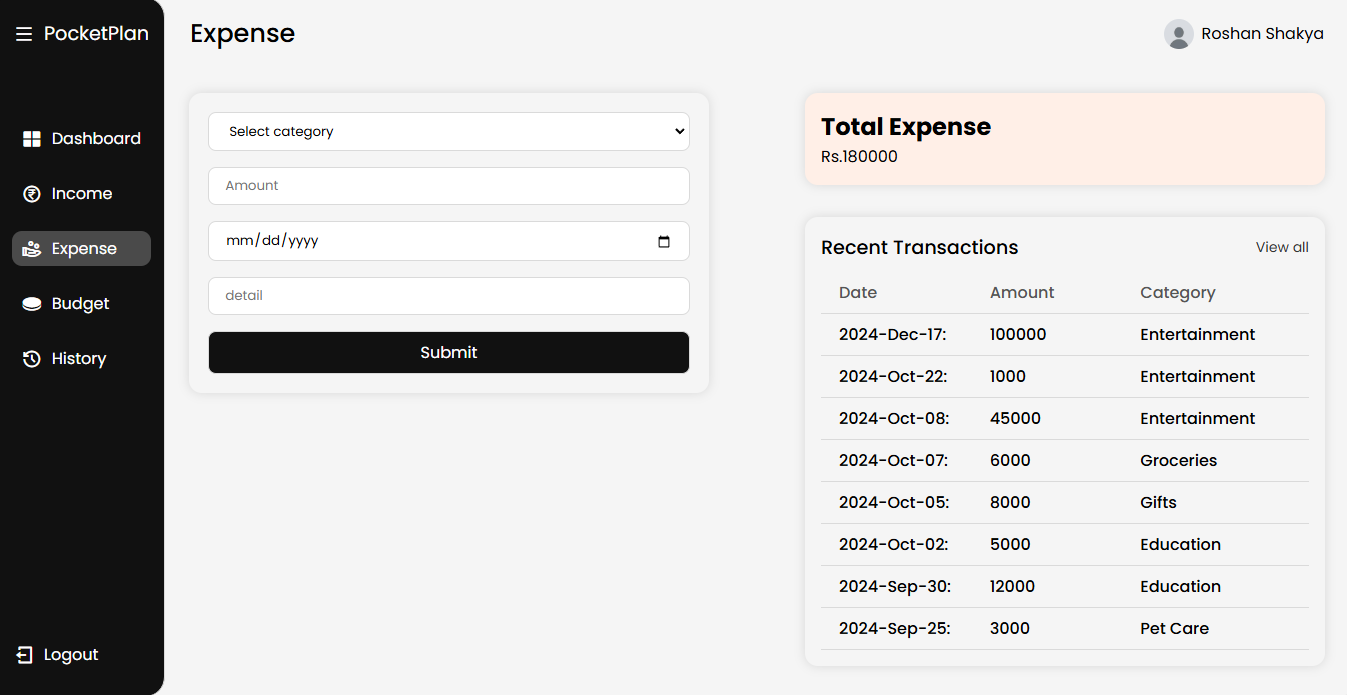
**Figure 3. 10: Dashboard Page of PocketPlan**

The dashboard provides an overview of the user's financial activities, displaying total income, expenses, transaction history and budget status in a visually organized manner. It helps users track their financial health at a glance. It’s the first thing they see when they log in.



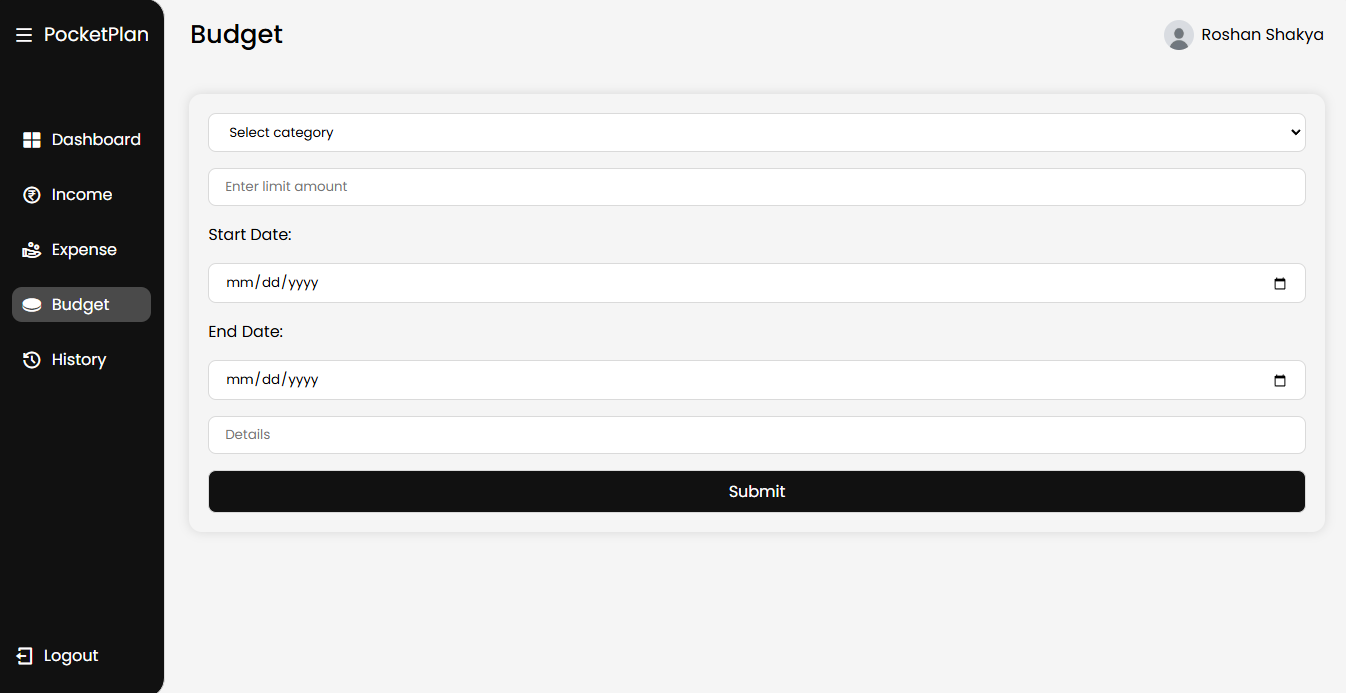
**Figure 3. 11: Income Page of PocketPlan**

This page allows users to add income records by specifying the amount,date of transaction, category, and description. It ensures users can keep track of their earnings efficiently. Also shows users total income till yet and history of only their income transactions.



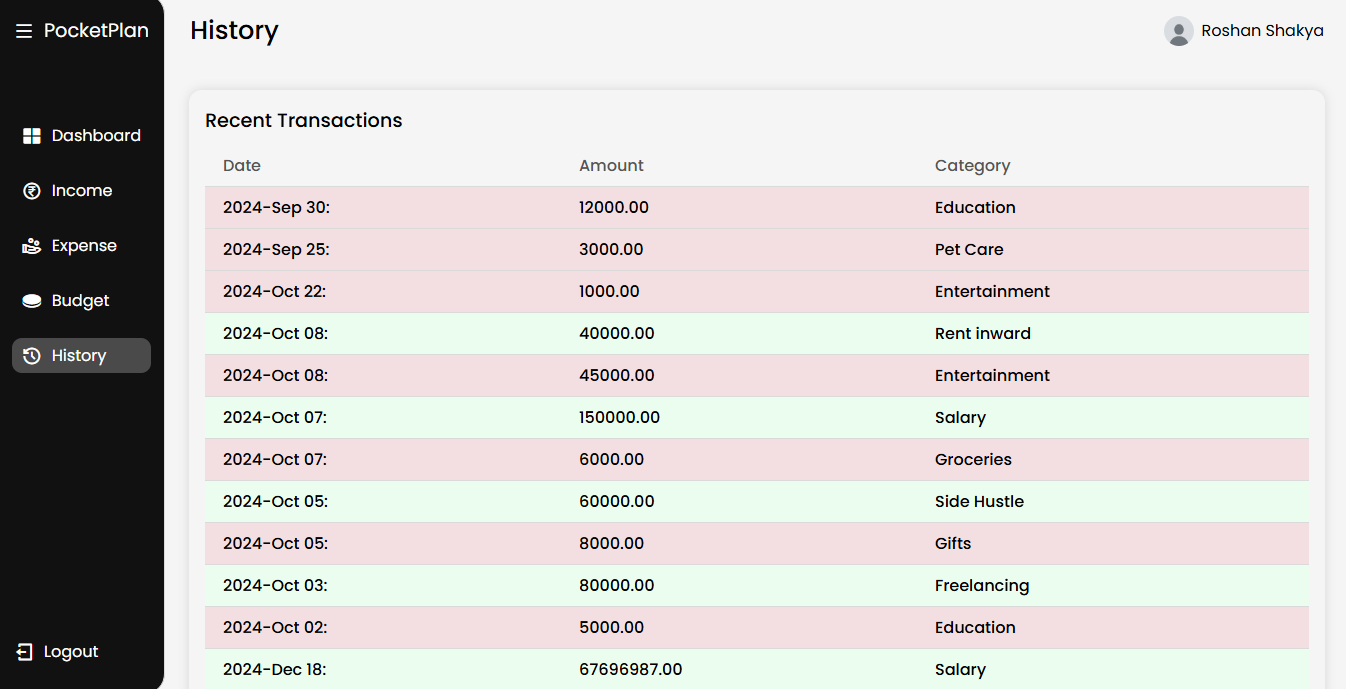
**Figure 3. 12: Expense Page of PocketPlan**

Users can record their expenses on this page, categorizing their spending and adding descriptions. It helps in monitoring where money is being spent. Like in the income page it also shows your total expenses till yet and expense history.



**Figure 3. 13: Budget Page of PocketPlan**

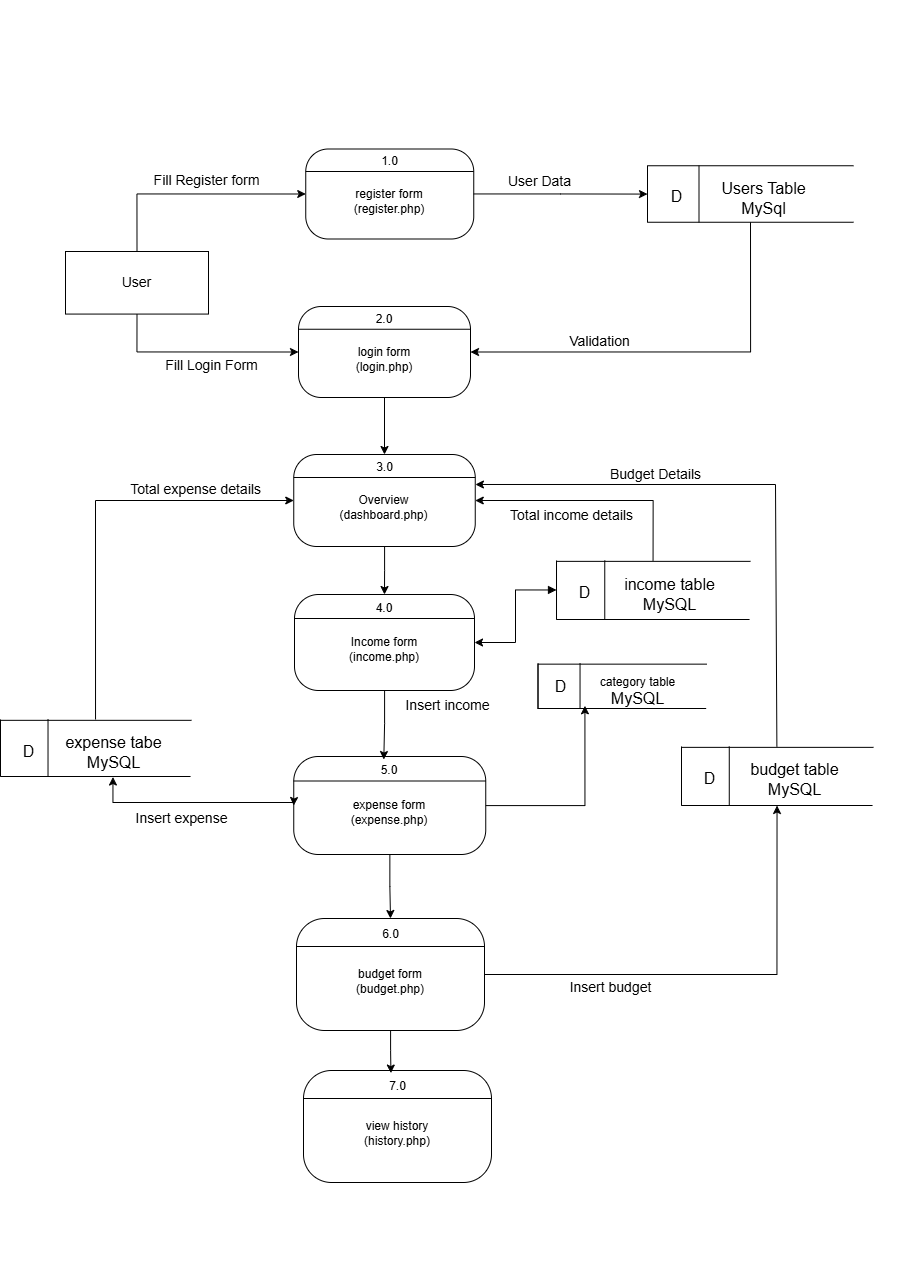
Users can set budgets for different categories and monitor their spending limits here. It helps in maintaining financial discipline and avoiding overspending.



**Figure 3. 14: History Page of PocketPlan**

This page enables users to manage their financial records by adding, editing, or deleting transactions. It provides a centralized place for tracking income and expenses.

### 3.2.3 Physical DFD



**Figure 3. 15: Physical DFD of PocketPlan**

The Physical DFD for PocketPlan shows how users manage finances. They register (register.php), log in (login.php), and access the dashboard (dashboard.php) to view income, expenses, and budgets. Users can add income (income.php), log expenses (expense.php), and set budgets (budget.php), with data stored in MySQL tables. They can also track transaction history (history.php). The system is entirely user-driven, with no admin role.

# CHAPTER 4: IMPLEMENTATION AND TESTING

## 4.1 IMPLEMENTATION

The implementation phase focused on converting the system design into a fully functional web application. This involved frontend and backend development, database management, and integration to ensure a seamless user experience.

### 4.1.1 Tools Used

The development of PocketPlan utilized the following technologies:

#### **Frontend Development**

* **HTML5:** Used to structure web pages.
* **CSS3:** Styled the application for a clean and responsive layout.
* **JavaScript:** Added interactivity to enhance the user experience.
* **Bootstrap:** Provided pre-designed components for a modern UI.

#### **Backend Development**

* **PHP:** Handled server-side logic, including authentication and financial data processing.
* **MySQL:** Used for database management, storing user accounts, transactions, and budget data.

#### **Development Environment**

* **XAMPP:** Provided a local server environment for running PHP and managing the MySQL database.
* **Visual Studio Code:** Used as the primary code editor for writing and debugging the application.

#### **Version Control**

* **Git & GitHub:** Used to track changes and manage project versions.

### 4.1.2 Implementation Details of Modules

The PocketPlan system consists of several modules, each serving a specific function. Below is an overview of their implementation:

#### **User Authentication Module**

* Users can sign up with a username, email and password.
* Passwords are securely stored using PHP's password\_hash() function to keep account safe.
* Sessions helps to manage logins only allowing the users to access data.

#### **Income and Expense Management Module**

* Users can add, edit, and delete income and expenses entries.
* Each entries includes amount, category and timestamps for better tracking.
* All transactions are stored in a database and displayed in a user interface.

#### **Budget Management Module**

* Users can set and manage budgets for different categories.
* The system calculates the remaining budget based on expenses.
* Users receive real-time updates on their spending.

#### **Transaction History Module**

* A complete history of all transactions are displayed.
* Transactions can be viewed and edited according to user’s requirement.

## 4.2 Testing

### 4.2.1 Test Case for Unit Testing

**1. Register page test case**

**Table 2: Register Page Test Case**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| T.C. No | Test Scenario | Test Data | Expected Output | Actual Output | Result |
| 1.1 | User Enters an invalid email | Email: wakawak24.com  Password: 12345 | Invalid Email | Success | fail |
| 1.2 | Password less than 8 character | Email: papaji24@gmail.com Password: 12345 | Password must contain at least 8 character | As expected | pass |
| 1.3 | Missing form fields,  Either one of them | Username:  Email: [someone@gmail.com](mailto:someone@gmail.com)  Password: p@ssword | All fields are required | As expected | pass |
| 1.4 | Enter valid username, email, password, | Username: Roshan Shakya  Email: [roshan.shakyaa2015@gmail.com](mailto:roshan.shakyaa2015@gmail.com)  Password: admin@123 | Redirected to dashboard | As expected | pass |

**2. Login page test case**

**Table 3: Login Page Test Case**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| T.C. No | Test Scenario | Test Data | Expected Output | Actual Output | Result |
| 2.1 | User Enters an invalid email | Email: roshan2003@gml.com  Password: admin@123 | Invalid Email | As expected | pass |
| 2.2 | User Enters an invalid email and valid Password | Email: papaji24@gmail.com Password: 12345 | Invalid username or password | As expected | pass |
| 2.3 | Missing form fields,  Either one of them | Username:  Email: [someone@gmail.com](mailto:someone@gmail.com)  Password: p@ssword | All fields are required | As expected | pass |
| 2.4 | Valid username or password | Username: Roshan Shakya  Email: [roshan.shakyaa2015@gmail.com](mailto:roshan.shakyaa2015@gmail.com)  Password: admin@123 | Redirected to dashboard | As expected | pass |

**3. Income / Expense Page test Case**

**Table 4: Income / Expense Page Test Case**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.N | Test Scenario | Test Data | Expected Output | Actual Output | Result |
| 3.1 | User enter amount in negative | Amount: -1000 | Amount must be greater than zero | Form submitted | fail |
| 3.2 | User enter amount in negative | Amount: -1000 | Amount must be greater than zero | As expected, | pass |
| 3.3 | User submits without any inputs | Category:  Amount:  Date:  Detail: | Please enter a number | As expected, | pass |
| 2.4 | User doesn’t select any category | Category: | Please select a category | As expected, | pass |

# CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATIONS

## 5.1. Lesson Learnt / Outcome

Upon completing the project, PocketPlan users will be able to manage their budget on a daily, weekly, or monthly basis, tailored to their preferences. This revolutionary budget tracker will not only monitor their expenses but also provide insights into potential savings and areas for expense reduction. With its easy-to-use interface, PocketPlan will cater to a wide range of users, from children and young adults to senior citizens, ensuring that budgeting is more effective, accessible, and convenient for everyone.

## 5.2. Conclusion

Upon successful completion of the PocketPlan project, users can now continue to track their finance more effectively than they ever could. The project has effectively met all its objectives, adhering to specifications while also enhancing certain features where necessary. Challenges were encountered, particularly in the backend development, to ensure predictable application responses.

The choice of PHP for this project was based on its simplicity, ease of use, robust data handling capabilities, and widespread global usage. PHP's open-source nature and platform independence made it an ideal choice for developing a versatile system.

With the project's completion, users can easily track their expenses, set personalized budgets, and receive insights on how to save money more effectively. The app caters to all age groups, offering tailored features that make budgeting accessible, efficient, and simple for everyone.

## 5.3. Future Recommendation

In the context of PocketPlan, there is considerable potential for future enhancements that could significantly boost usability, elevate the user experience, and improve the platform's overall portability. The current version serves as a strong foundation for what has the potential to evolve into a more expansive and advanced system. Although these future developments may require additional time and resources, they remain both realistic and achievable. Addition buyout features are:

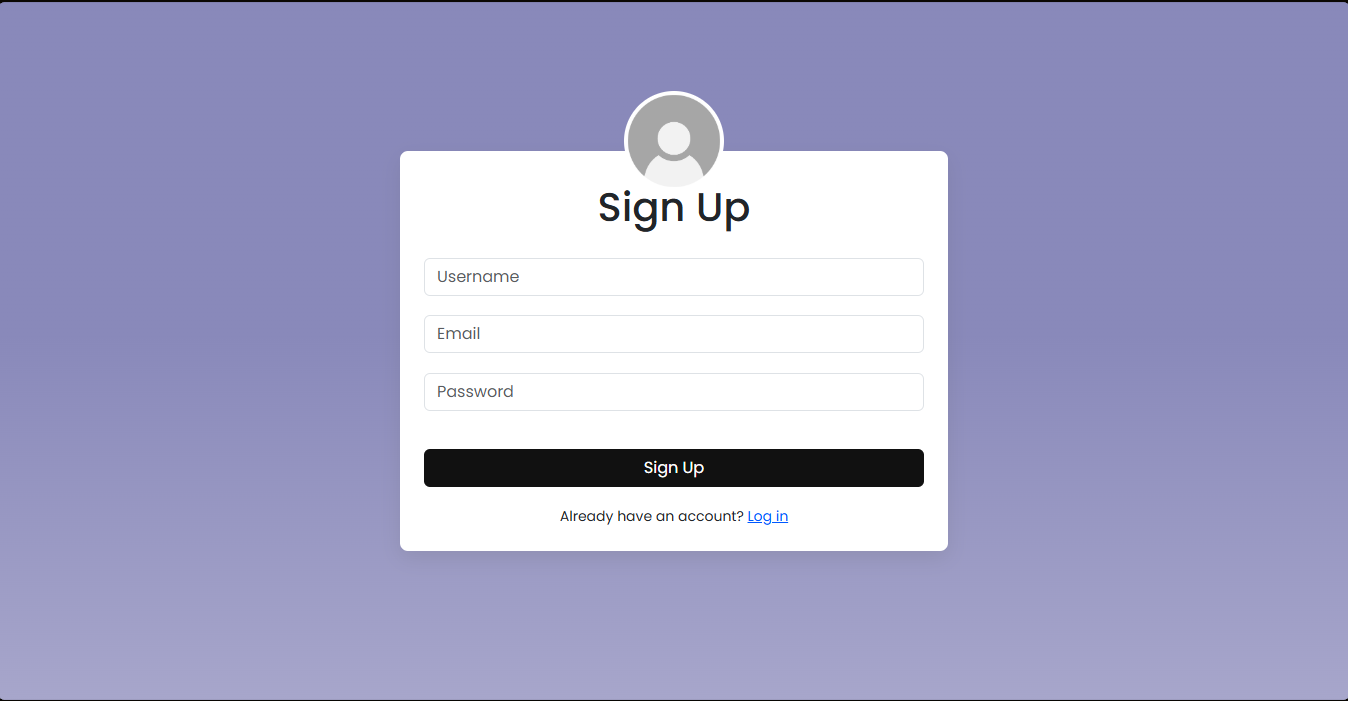
* **AI-Integration:** Integrate AI for users, such as personalized saving tips, investment advice, or budgeting strategies based on spending habits and goals.
* **Multi-Currency Support:** Enable users to track and manage expenses in different currencies, making the app useful for people who travel frequently or live abroad.
* **Bill Payment Integration:** Add features to schedule and pay bills directly through the app, along with automatic reminders for due payments to avoid late fees.
* **Goal-Based Savings Plans:** Allow users to create specific savings plans for particular goals (e.g., vacation, home purchase) and track progress toward achieving them.
* **Add Graphs and Chart:** Adding graphs and chart regarding the daily financial status will be great upgrade for the user to visually analyze and track their daily budgeting.

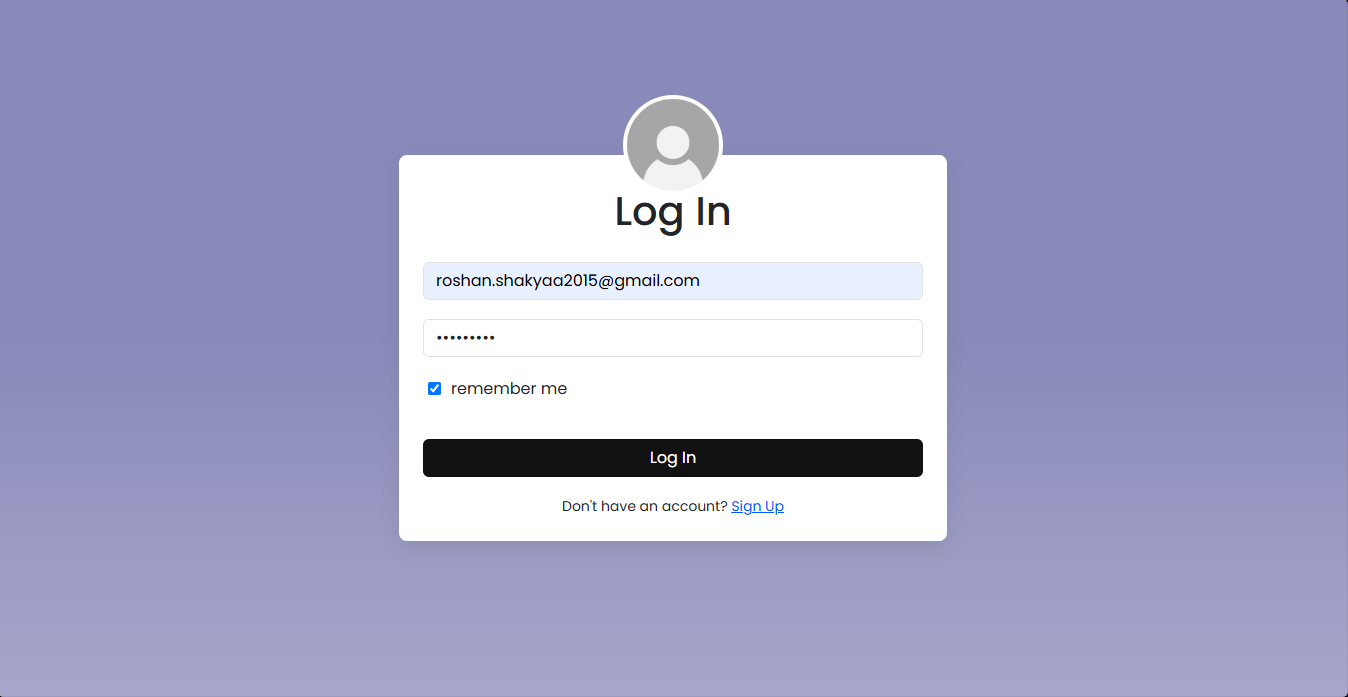
# REFERENCES

|  |  |
| --- | --- |
| [1] | "PHP Manual," PHP Group, 1993. [Online]. Available: https://www.php.net/. |
| [2] | "Remix Icon," Remix Design, 2019. [Online]. Available: https://remixicon.com/. |
| [3] | "Mint," Intuit, [Online]. Available: https://www.mint.com. |
| [4] | "Pinterest," Pinterest, Inc., 2010. [Online]. Available: https://www.pinterest.com. |
| [5] | "W3schools," Refsnes Data, 1998. [Online]. Available: https://www.w3schools.com/. |

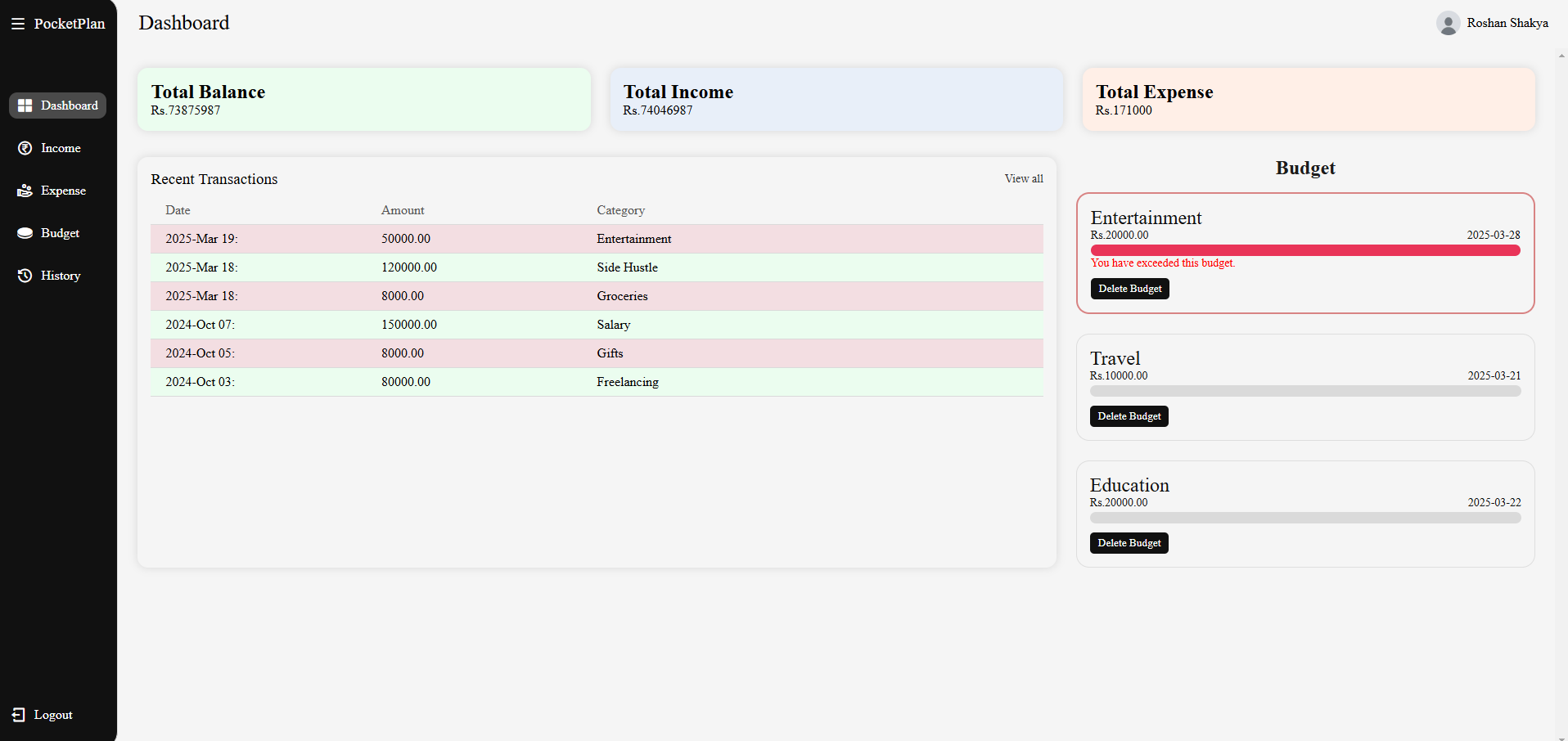
# APPENDICES

**Auth pages**

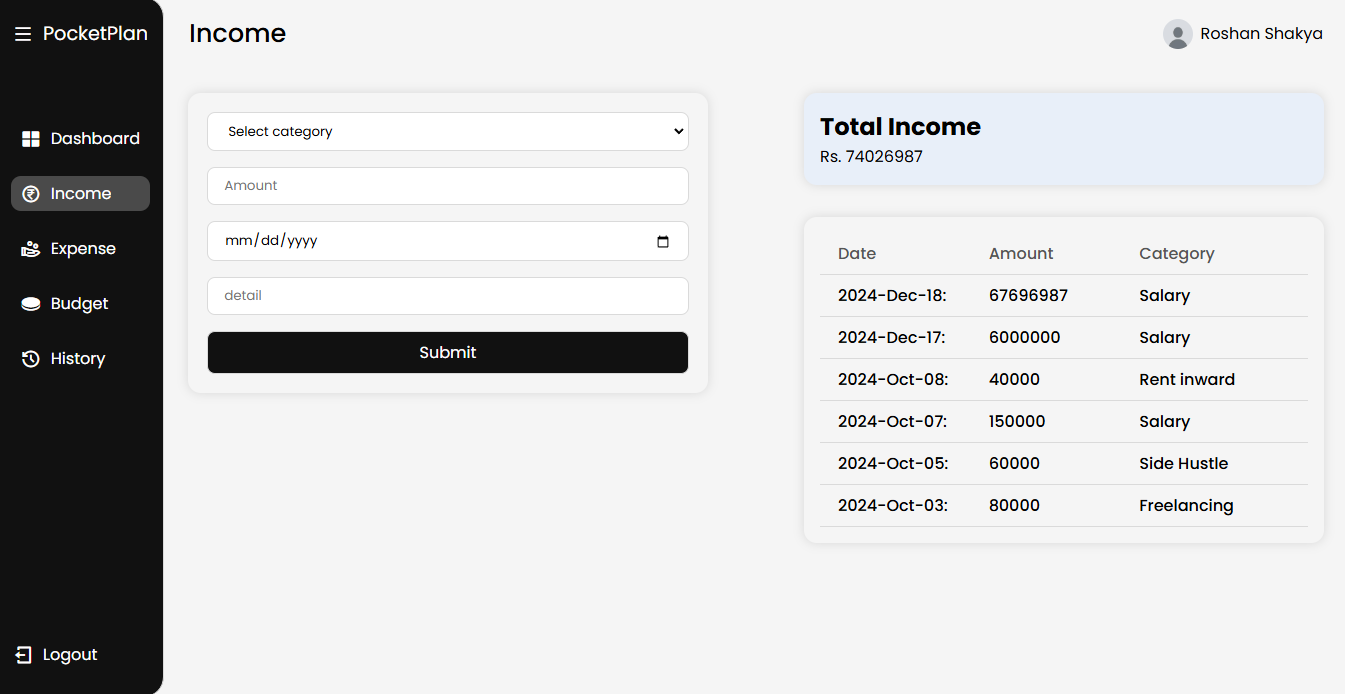




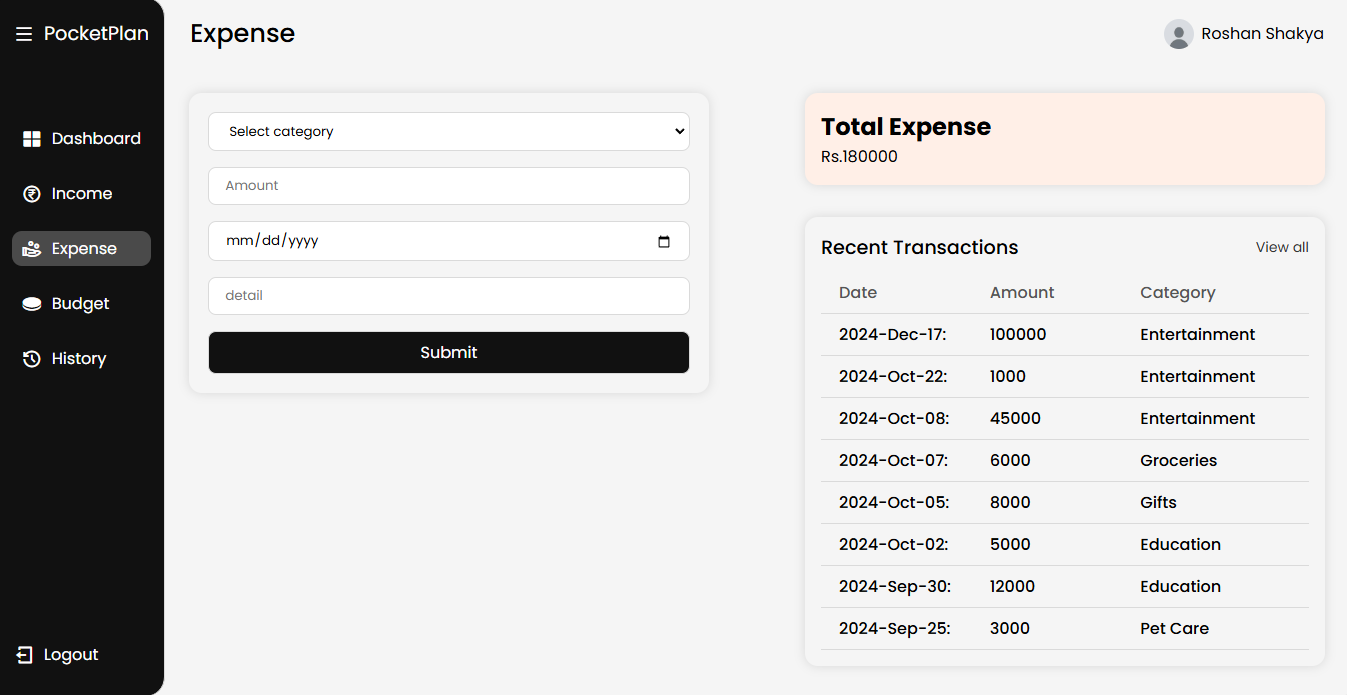
**Dashboard Page**

****

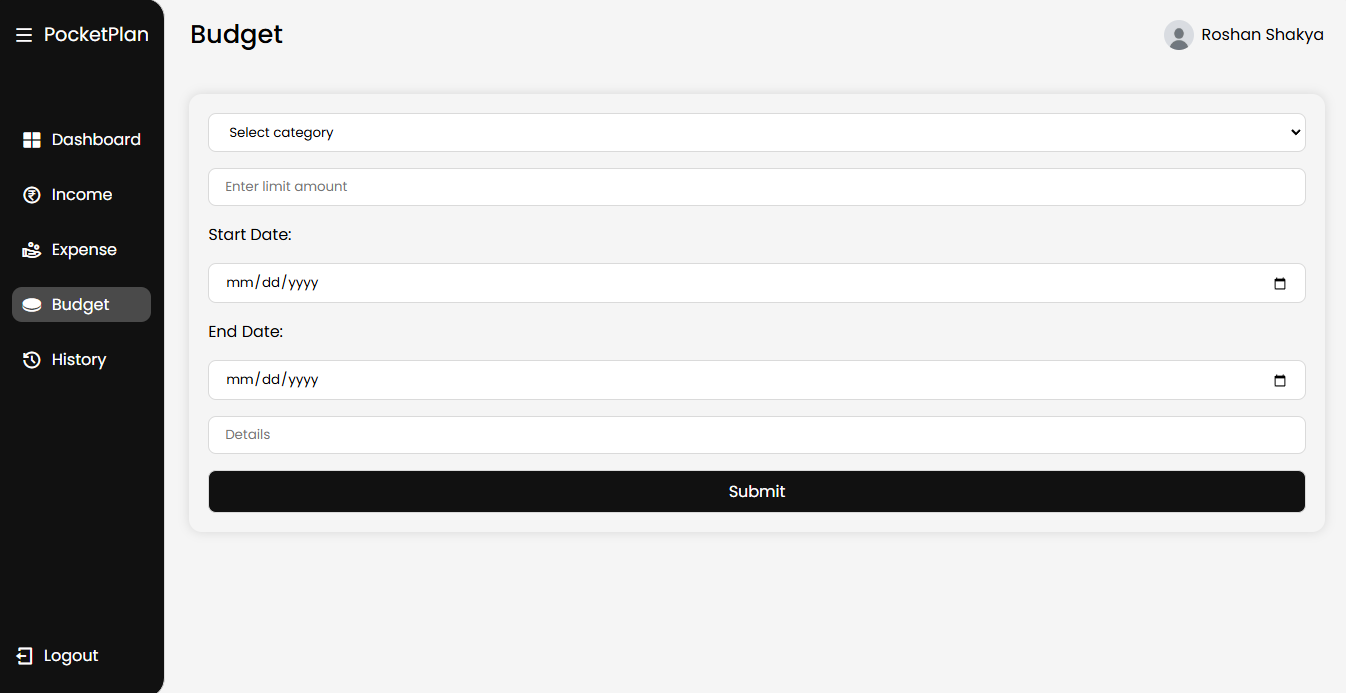
**Income Page**



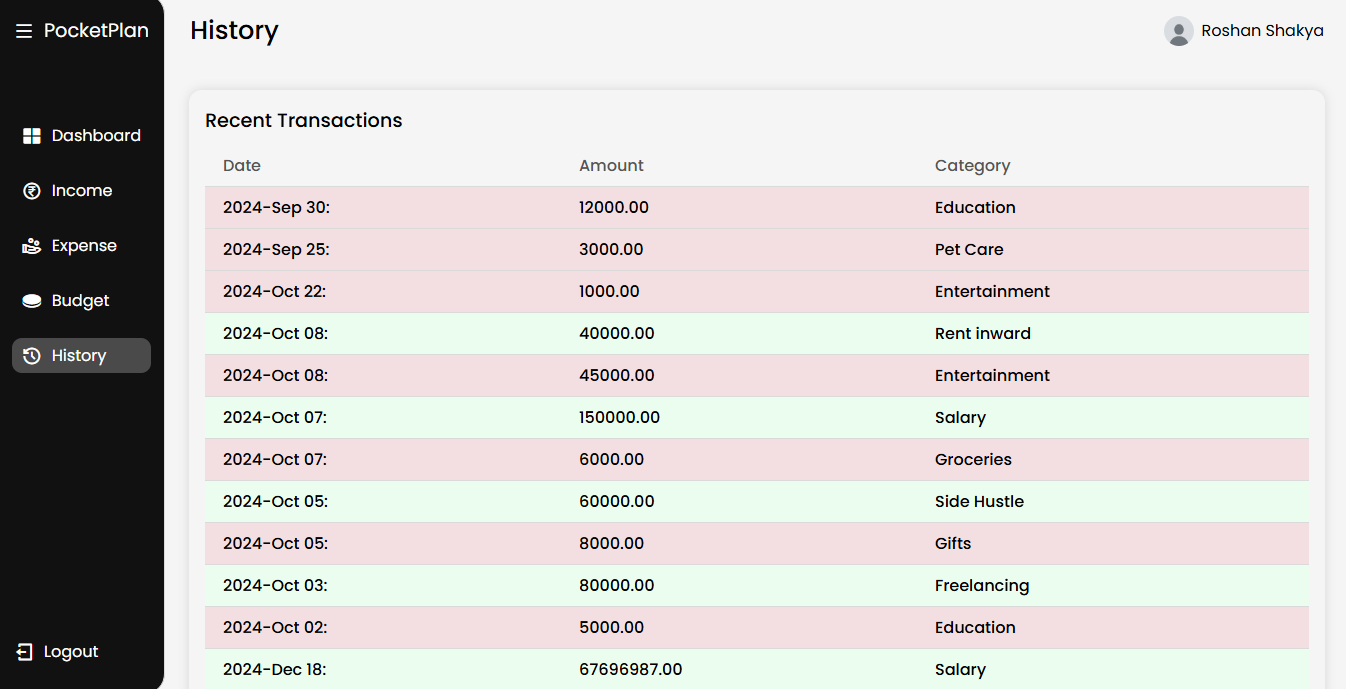
**Expense Page**



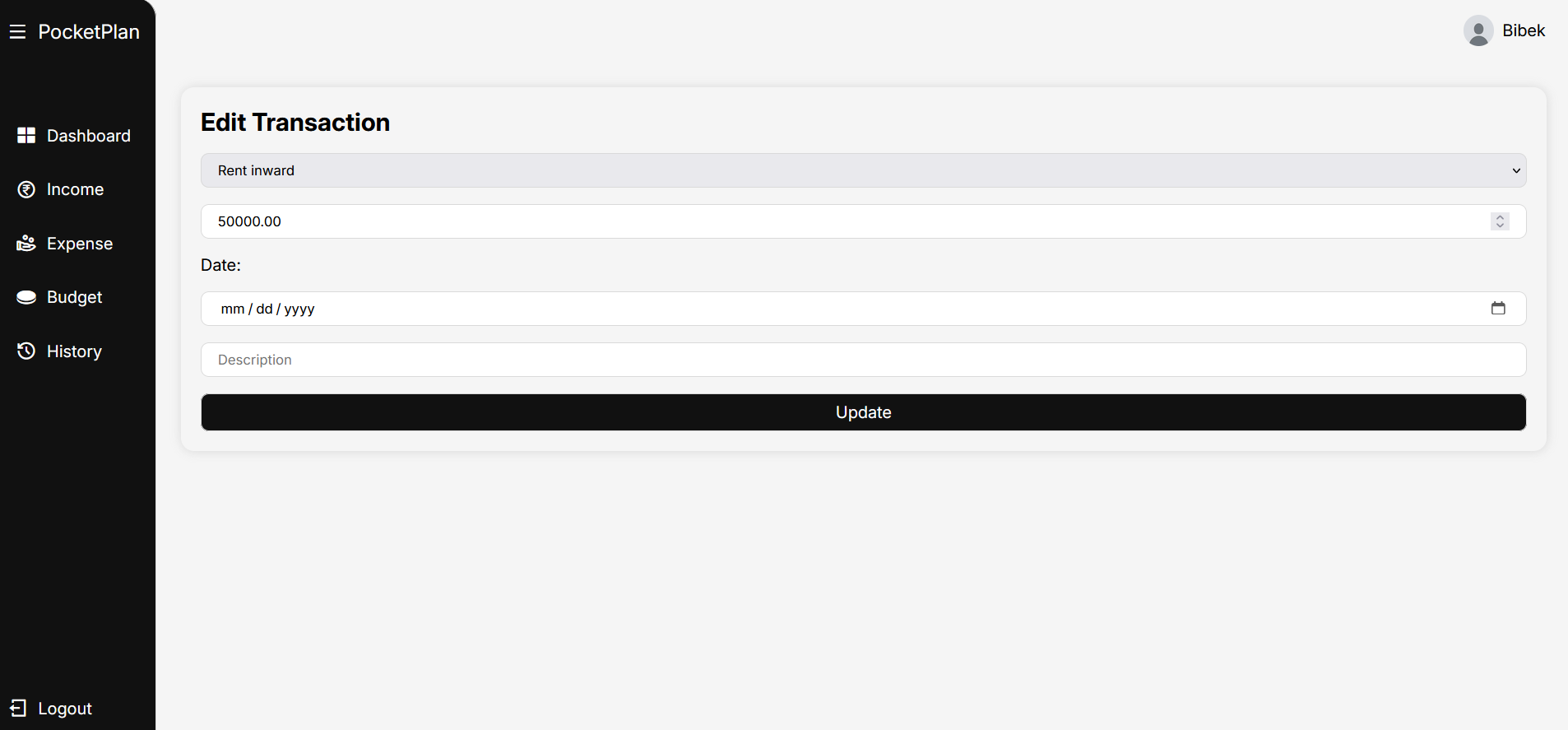
**Budget Page**



**History Page**



**Edit Page**

****

**PHP program for Delete Transaction**

<?php

include "./connect.php";

include "./auth/verify.php";

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    $type = $\_POST['type'];

    $amount = $\_POST['amount'];

    $user\_id = $\_POST['user\_id'];

    $description = $\_POST['description'];

    if ($type === "income") {

        $sql = "DELETE FROM income WHERE user\_id = ? AND amount = ? AND description = ? LIMIT 1";

    } else {

        $sql = "DELETE FROM expense WHERE user\_id = ? AND amount = ? AND description = ? LIMIT 1";

    }

    $stmt = $conn->prepare($sql);

    $stmt->bind\_param("ids", $user\_id, $amount, $description);

    if ($stmt->execute()) {

        header("Location: ./pages/dashboard.php"); // Redirect to dashboard after deletion

        exit();

    } else {

        echo "Error deleting transaction: " . $conn->error;

    }

    $stmt->close();

    $conn->close();

}

?>