PERSONAL EXPENSE TRACKER

A PROJECT REPORT

Submitted by

KEERTHIGA T (2303811710622052)

in partial fulfillment of requirements for the award of the course

EGB1221- DATABASE MANAGEMENT SYSTEMS

In

ELECTRONICS AND COMMUNICATION ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

JUNE - 2025

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on "**PERSONAL EXPENSE TRACKER"** is the bonafide work of **KEERTHIGA T (2303811710622052)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

SIGNATURE

Dr.S.SYEDAKBAR, M.E., Ph.D.,

HEAD OF THE DEPARTMENT

ASSISTANT PROFESSOR

Department of ECE

K.Ramakrishnan College of Technology
(Autonomous)

Samayapuram–621112.

SIGNATURE

Mrs.R.BHAVANI,M.E,MBA.,(Ph.D).,

SUPERVISOR

ASSISTANT PROFESSOR

Department of ECE

K.Ramakrishnan College of Technology
(Autonomous)

Samayapuram–621112.

Submitted for the viva-voce examination held on …………….

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

I declare that the project report on **“PERSONAL EXPENSE TRACKER”** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **“ANNA UNIVERSITY CHENNAI”** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfillment of the requirement of the award of the course **EGB1221 - DATABASE MANAGEMENT SYSTEMS.**

Signature

Keerthiga T

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and indebtedness to our institution, "**K. Ramakrishnan College of Technology (Autonomous)**", for providing us with the opportunity to do this project.

I extend our sincere acknowledgment and appreciation to the esteemed and honorable Chairman, **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director, **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding our project and offering an adequate duration to complete it.

I would like to thank **Dr. N. VASUDEVAN, M.TECH., Ph.D.,** Principal, who gave the opportunity to frame the project to full satisfaction.

I thank **Dr. S. SYEDAKBAR, M.E., Ph.D.,** Head of the Department of **ELECTRONICS AND COMMUNICATION ENGINEERING**, for providing her encouragement in pursuing this project.

I wish to convey our profound and heartfelt gratitude to our esteemed project guide **Mrs. R. BHAVANI, M.E., MBA., (PhD),** Department of **ELECTRONICS AND COMMUNICATION ENGINEERING,** for her incalculable suggestions, creativity, assistance and patience, which motivated us to carry out this project.

I render our sincere thanks to the Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards

MISSION OF THE INSTITUTION

- Be a center of excellence for technical education in emerging technologies by exceeding the needs of the industry and society.
- Be an institute with world class research facilities
- Be an institute nurturing talent and enhancing the competency of students to transform them as all-round personality respecting moral and ethical values

VISION OF DEPARTMENT

To create innovative and socially responsible Electronics and Communication Engineers with design skills and research focus to meet Societal and Industrial needs.

MISSION OF DEPARTMENT

M1: To provide high quality education and professional ethics to students through enhanced learning environment.

M2: To impart a creative environment towards centre of excellence in department with design skill and exposure for research.

M3: To nurture required employable skills of students to satisfy the industry and social needs with ethical and human values.

PROGRAM EDUCATIONAL OBJECTIVES

PEO1: Core Knowledge Development: Graduates will have enhanced engineering skills in the field of electronics, communication and interdisciplinary areas to serve the society with global standards.

PEO2: Professional development: Graduates will apply the technical knowledge for continuous up gradation of their professional skills to become an inimitable employee, researcher or entrepreneur.

PEO3: Analytical Thinking: Graduates will have analytic and thinking skills to provide the innovative solutions for industry and societal requirements.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: To analyze, design and develop solutions by applying foundational concepts of electronics and communication engineering.

PSO2: To apply design principles and best practices for developing quality products for scientific and business applications.

PSO3: To adapt to emerging information and communication technologies (ICT) to innovate ideas and solutions to existing/novel problems.

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences

3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and

need for sustainable development

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

The Personal Expense Tracker is a database-driven application designed to help individuals efficiently manage and monitor their daily financial activities. This system enables users to record, categorize, and analyze their income and expenses, thereby promoting better financial planning and budgeting. Built using fundamental principles of database management, the project incorporates structured data storage, relational tables, and SQL-based operations for data insertion, retrieval, update, and deletion. The database design ensures data integrity, security, and consistency while supporting features such as monthly reports, category-wise analysis, and historical tracking of expenses. This project highlights the practical implementation of DBMS concepts including normalization, query optimization, and relational schema design, providing a scalable and user-friendly solution for personal finance management.The Personal Expense Tracker is a digital solution designed to help individuals manage their daily finances efficiently. The system enables users to record income and expenses, categorize transactions, set monthly budgets, and visualize spending patterns through charts and summaries. By providing real-time insights and historical data analysis, the tracker promotes better financial awareness and planning. Developed using modern technologies, it emphasizes user-friendly interfaces, secure data handling, and cross-platform accessibility. This project aims to foster financial discipline and support informed decision-making for personal financial health.Users can register and securely log into the system, add income sources and expenses under various categories, and view summarized data through a centralized dashboard. The backend is developed using SQL for efficient data storage, retrieval, and management, ensuring accuracy and scalability. The application promotes better financial planning and awareness by providing real-time feedback and data visualization on user spending.This project demonstrates the practical implementation of database management systems, form handling, and CRUD operations. It also highlights the integration of frontend interfaces (using HTML) with SQL databases, offering a complete solution for day-to-day financial tracking and analysis.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD DATABASE APPLICATIONS FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
The **Personal Expense Tracker** is a web-based application designed to help users efficiently manage their daily finances. It allows users to register, log in securely, and record their income and expenses in a structured way. With features like income tracking, expense categorization, and budgeting, the system offers users a clear overview of their financial activities. The backend is powered by SQL for data storage and retrieval, ensuring data integrity and easy access. This project aims to promote better financial habits by offering a user-friendly and accessible platform for personal financial management. The Personal Expense Tracker is a digital solution designed to help individuals manage their daily finances efficiently. The system enables users to record income and expenses, categorize transactions, set monthly budgets, and visualize spending patterns through charts and summaries. By providing real-time insights and historical data analysis, the tracker promotes better financial awareness and planning. Developed using modern technologies, it emphasizes user-friendly interfaces, secure data handling, and cross-platform accessibility.	**PO1-3** **PO2-3** **PO3-3** **PO4-3** **PO5-3** **PO6-3** **PO7-3** **PO8-3** **PO9-3** **PO10-3** **PO11-3** **PO12-3**	**PSO3 -1** **PSO3 - 2** **PSO3 - 3**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

DB	-	Data Base
UI	-	User Interface
PHP	-	Personal Home Page
CSS	-	Cascading Style SheetS
HTML	-	HyperText Markup Language
SQL	-	Structured Query Language
CURD	-	Create, Read, Update, Delete
XAMPP	-	Cross-Platform Apache, MariaDB, PHP, Perl
APIs	-	Application Programming Interface

CHAPTER 1
INTRODUCTION

1.1Objective

The Personal Expense Tracker project aims to develop a secure, user-friendly, and efficient application for managing individual financial activities using core concepts of Database Management Systems (DBMS). The primary objective is to help users accurately track their income and expenses, categorize their spending, and maintain a clear overview of their financial status through interactive dashboards and reports. By allowing users to set monthly or category-wise budgets, the system also promotes better budgeting and financial discipline. The project emphasizes data integrity, user authentication, and secure access to personal financial records. Through well-structured relational databases and optimized queries, the system ensures reliable data storage, easy retrieval, and smooth performance. Overall, this project demonstrates the practical application of DBMS principles in solving real-world problems related to personal finance management.

1.2 Overview

The Personal Expense Tracker is a database-driven application designed to assist individuals in managing their personal finances effectively. It enables users to record daily income and expenses, categorize transactions, and monitor financial activity over time. The system features essential modules such as user authentication, a dynamic dashboard, income and expense management, and a budgeting tool that collectively provide a comprehensive view of the user's financial health. With a strong focus on data organization and integrity, the project leverages the principles of Database Management Systems (DBMS) to store, retrieve, and process financial data efficiently. The intuitive interface and analytical features, including visual reports and alerts for budget limits, make it easier for users to make informed financial decisions. This project serves as a practical example of how DBMS concepts can be applied to develop a real-world application that enhances personal money management and financial planning.

1.3 SQL and Database Concepts

Database Design (ER Diagram & Schema)

- ❖ Designing tables for entities like Users, Income, Expenses, Categories, and

Budgets.

- Establishing relationships using primary keys and foreign keys.

EXAMPLE :

```
CREATE TABLE Users (
   user_id INT PRIMARY KEY AUTO_INCREMENT,
   username VARCHAR(50),
   email VARCHAR(100),
   password_hash VARCHAR(255)
);
```

Data Normalization

- Applying normal forms (1NF, 2NF, 3NF) to avoid redundancy and ensure data consistency.
- For example, storing categories in a separate table instead of repeating category names in every transaction.

SQL CRUD Operations

- Create, Read, Update, Delete operations are used extensively in modules like income and expense management.

EXAMPLE :

```
INSERT INTO Expenses (user_id, amount, category_id, expense_date,
description)
VALUES (1, 500, 2, '2025-05-12', 'Groceries');
SELECT * FROM Expenses WHERE user_id = 1;
UPDATE Expenses SET amount = 600 WHERE expense_id = 5;
DELETE FROM Expenses WHERE expense_id = 5;
```

Joins

- Using INNER JOIN, LEFT JOIN, etc., to combine data from multiple related tables (e.g., linking expenses with categories).

EXAMPLE :

```
SELECT e.amount, c.category_name
FROM Expenses e
JOIN Categories c ON e.category_id = c.category_id
WHERE e.user_id = 1;
```

Aggregate Functions & Grouping

- Using SUM(), COUNT(), AVG(), GROUP BY, and HAVING to analyze financial data (e.g., total spending per category).

  EXAMPLE :

```
SELECT c.category_name, SUM(e.amount) AS total_spent

FROM Expenses e

JOIN Categories c ON e.category_id = c.category_id

WHERE e.user_id = 1

GROUP BY c.category_name;
```

Constraints

- Enforcing NOT NULL, UNIQUE, CHECK, and FOREIGN KEY constraints to ensure data validity and consistency.

Views

- Creating SQL views for simplified access to frequently used complex queries, like monthly summaries.

Indexing

- Creating indexes on frequently queried columns (e.g., user_id, expense_date) to improve query performance.

Transactions

- Ensuring data reliability using SQL(Structured Query Language) transactions during sensitive operations like bulk uploads or deletions.

Stored Procedures

- Automating common tasks like monthly report generation or sending budget alerts.
- A Stored Procedure is a precompiled collection of one or more SQL statements stored in the database. It can be executed repeatedly by the application or user without rewriting the SQL logic.

CHAPTER 2
PROJECT METHODOLOGY

2.1Proposed Work

The proposed work for the **Personal Expense Tracker** project involves designing and developing a database-driven application that allows users to effectively manage and analyze their personal finances. The project will begin with the creation of a well-structured relational database(DB) using normalization techniques to define tables such as users, income, expenses, categories, and budgets. The system will include a secure user authentication module to ensure that financial data is protected and accessible only by authorized individuals. Once authenticated, users will be directed to a personalized dashboard that presents a visual summary of their financial activities, including total income, total expenses, and balance, along with category-wise spending charts.

The application will allow users to enter and manage both income and expense records, with the ability to categorize transactions, filter them by date or type, and view detailed histories. Additionally, a budgeting module will be implemented to enable users to set monthly or category-based spending limits and receive alerts when those limits are exceeded. SQL(Structured Query Language) will be used for data manipulation and reporting, including CRUD(Create, Read, Update, Delete) operations, joins, aggregate functions, and query optimization for better performance. The system will also support the financial summaries to help users make informed decisions. Overall, the proposed work aims to deliver a reliable, user-friendly, and secure platform that leverages database management concepts to assist users in gaining better control over their financial habits.

In addition to core functionalities, the proposed system will focus on enhancing usability and scalability to accommodate a wide range of users with varying financial tracking needs. The application will feature intuitive data entry forms, dynamic filtering options, and interactive charts to ensure a smooth user experience. From a technical perspective, best practices in database design will be applied, including the use of foreign keys for referential integrity, indexing for faster query execution, and stored procedures or triggers where appropriate to automate recurring operations like generating monthly summaries or updating budget status. Future enhancements may include data export options (e.g., PDF or Excel reports), multi-currency support, and mobile-friendly interfaces. Overall, this project not only demonstrates the practical application of database management system concepts but also delivers a valuable tool that can help users build better financial awareness and discipline through organized, data-driven insights.

2.2 Block Diagram

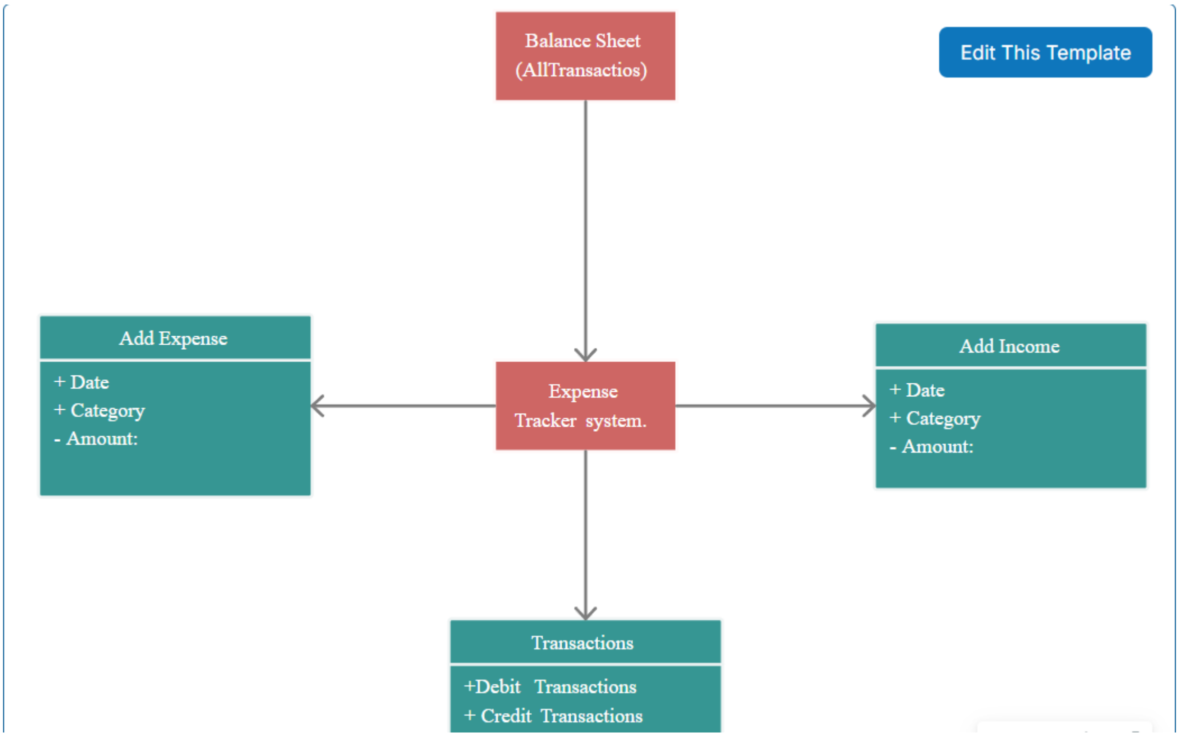


Fig 2.2.1 Block Diagram

The block diagram of the Personal Expense Tracker system represents the overall structure and workflow of the application in a clear and organized manner. At the top of the diagram is the User Authentication Module, which is responsible for verifying user credentials and allowing access to the system. Once a user logs in successfully, they are directed to the Dashboard Module, which serves as the central control panel. From the dashboard, users can navigate to three key modules: the Income Management Module, the Expense Management Module, and the Budgeting Module. Each of these modules handles specific tasks—recording income, tracking expenses, and setting or monitoring budgets, respectively. At the bottom of the diagram, all modules are connected to a centralized Database, which stores all user data securely and supports the entire system's operations.

The block diagram of the Personal Expense Tracker System visually represents how various components of the system interact to help users manage their finances. At the center of the system is the Expense Tracker System, which acts as the core processor for all income and expense entries. Users can input income details through the Add Income module, where they specify the date, category (such as salary or gift), and amount received. Similarly, users log their expenses using the Add Expense module by providing the date, expense category (like rent or groceries), and amount spent.

Once entered, the data flows into the Transactions module, which stores and classifies each record as either a Credit Transaction (income) or a Debit Transaction (expense). This organized logging is essential for tracking financial activities over time. All the information from the transactions module is then utilized by the Balance Sheet (All Transactions) component. This section compiles the entire financial record of the user to present a clear picture of total income, total expenses, and the resulting balance.

In essence, the system streamlines financial data entry and automatically organizes and summarizes it to help users maintain a healthy financial overview. The modular design ensures each function—whether it’s adding income, tracking expenses, or viewing balances—is easy to access and interact with.

CHAPTER 3
MODULE DESCRIPTION

3.1 User Authentication Module

This module is responsible for managing user access to the system. It allows new users to register by creating an account with a username, email, and password, and enables existing users to log in securely. It ensures that each user interface(UI) is private and protected using secure password storage methods (like hashing). Only authenticated users can access their financial data. This module may also include features like password recovery and session management.

The User Authentication Module is responsible for managing secure access to the application. It allows users to register by providing their credentials such as username, email, and password. Once registered, users can log in to access their personal dashboard and financial data. This module ensures that only authorized users can view and manage their expense records. Passwords are typically encrypted using hashing techniques to enhance security. Additionally, it handles session management to keep users logged in securely until they choose to log out. This module forms the foundation of user-level data protection and personalized access.

3.2 Dashboard Module

Once a user logs in, they are taken to the Dashboard, which acts as the main interface. It displays a summary of key financial information such as total income, total expenses, balance, and graphical reports (e.g., pie charts or bar graphs showing expense categories). It provides an overview of financial activity and serves as a gateway to other modules like income, expenses, and budgeting**.**

The Dashboard Module acts as the central hub of the application, presenting the user with a summarized view of their financial activities. It displays an overview of total income, total expenses, and the current balance. Users can quickly review recent transactions and access shortcuts to important features such as adding income or logging an expense. Visual aids such as charts and graphs are often included to give a clear picture of spending patterns. This module is essential for helping users understand their financial status at a glance and provides an intuitive interface to navigate other parts of the tracker.

3.3 Income Management Module

This module allows users to record and manage their income entries. Users can input different sources of income such as salary, freelance work, or business earnings, along with the amount, date, and a description. The data can be viewed, updated, or deleted as needed. This module helps users track all money coming in and contributes to accurate balance calculations shown in the dashboard. The Income Management Module allows users to log and monitor all their income sources efficiently. Users can add new income entries by specifying the amount, date, income category (such as salary, freelance, or interest), and any additional notes. All previous income records can be viewed, updated, or deleted through this module. By keeping a detailed log of income, users can track their earning trends over time and ensure that their financial planning starts from accurate earnings data. This module plays a crucial role in ensuring accurate calculations for budgeting and balance display.

3.4 Expense Management Module

The expense module helps users log and categorize their spending. Each expense entry includes details like amount, date, category (e.g., food, transport, entertainment), and optional notes. Users can view, filter, update, or delete expense records in XAMPP(Cross-Platform Apache, MariaDB, PHP, Perl). The system uses this enabling users to understand their spending behavior over time.

The Expense Management Module is one of the core components of the application, designed to let users track their daily, weekly, or monthly expenses. Users can input expense details including amount, date, category (like groceries, rent, transport), and a description. It also provides options to filter and search for expenses based on time or category, and to edit or delete entries when necessary. This module ensures that users can maintain a detailed and organized history of where their money is being spent, helping them identify areas for potential savings and avoid overspending.

3.5 Budgeting Module

This module allows users to set monthly or category-wise budgets to control their spending. For example, a user might set a ₹5000 limit for food expenses per month. The system monitors the user's expenses and provides alerts or notifications if they exceed their budget. This helps promote better financial discipline by encouraging users to stick to their planned spending limits.

The Budgeting Module enables users to plan their finances more effectively by setting a monthly budget limit. Users can define how much they intend to spend in a given month, and the system compares actual expenses with the set budget in real-time. It provides warnings or visual alerts when the spending nears or exceeds the budget. Additionally, users can review past budget history to evaluate their financial discipline and make informed decisions for future planning. This module encourages better money management and helps users align their spending habits with their financial goals.The Budgeting Module can also provide category-wise budgeting, allowing users to allocate specific amounts to different expense categories such as food, utilities, transportation, or entertainment. This feature gives users more detailed control over their finances and helps in analyzing which areas often exceed planned spending. The module may also include visual representations such as budget vs. actual bar graphs or pie charts to easily spot variances. Over time, these insights help users adjust their budgeting strategies to improve savings and build financial discipline, making this module a valuable tool for long-term money management.

CHAPTER 4

CONCLUSION AND FUTURE ENHANCEMENT

The Personal Expense Tracker project effectively demonstrates how database management systems can be applied to solve real-life financial management problems. By integrating core modules such as user authentication, income and expense tracking, budgeting, and an interactive dashboard, the system offers users a practical and user-friendly solution for managing their personal finances. The use of a structured relational database ensures secure, organized, and efficient data handling, while SQL queries facilitate accurate retrieval and analysis of financial information. Through features like category-wise tracking, budget alerts, and visual summaries, users gain better insights into their spending habits and are encouraged to make informed financial decisions. Overall, this project not only showcases technical skills in database design and application development but also delivers a valuable tool that promotes financial awareness and discipline in daily life.

While the current version of the Personal Expense Tracker provides a solid foundation for managing personal finances, several future enhancements can be introduced to improve its functionality, usability, and adaptability. One major enhancement could be the integration of mobile app support, allowing users to track their expenses on the go. Adding bank account and digital wallet integration would enable automatic transaction imports, reducing manual data entry. Another useful feature would be the inclusion of AI-based spending predictions and suggestions, helping users forecast expenses based on past trends and suggesting areas where savings can be made.

Moreover, implementing multi-user support or shared family/group accounts could allow households to manage budgets together. The system can also benefit from data visualization upgrades, offering more interactive charts, comparisons, and custom reports. Providing data export options (PDF, Excel, etc.) and cloud backup would enhance data portability and safety. Lastly, incorporating multi-currency support and language localization would make the tracker suitable for international users. These enhancements would not only improve the overall user experience but also make the application more robust, intelligent, and scalable for broader use.

The Personal Expense Tracker project represents a practical and efficient solution for managing personal financial records in an increasingly complex and digitally driven world. It addresses a fundamental need of individuals to gain better visibility into their financial habits by offering a centralized platform to track income, categorize expenses, monitor spending patterns, and generate summaries or balance sheets. Built using a combination of HTML, CSS, and JavaScript for the front-end and PHP with MySQL for the back-end, this project showcases the seamless integration of client-side interactivity and server-side logic. It allows users to input their daily financial transactions in a categorized and structured format, ensuring easy data retrieval and meaningful analysis. From a technical perspective, the project demonstrates proficiency in database design, including the use of normalized tables for users, expenses, income, and transaction categories, along with stored procedures that automate common tasks like inserting or summarizing data. The core functionality—adding income, logging expenses, displaying balance sheets, and managing transaction records—ensures that users remain aware of their financial status at all times, promoting disciplined and informed financial behavior.

Beyond its utility as a budgeting tool, the Personal Expense Tracker also holds educational value by familiarizing students and beginner developers with web technologies and database interaction patterns. It promotes good software development practices such as modular design, use of reusable components, secure input handling, and clear database schema management. The interface, though basic, ensures ease of use through intuitive forms and logical layout of features. The ability to filter transactions by date or category empowers users to perform their own spending audits without requiring any external help or financial literacy training. Moreover, the graphical user interface can be further enhanced with JavaScript charting libraries like Chart.js or Google Charts to present visual insights such as monthly spending patterns, savings growth, or category-wise distribution. These improvements would not only make the system more interactive but also aid in better comprehension of complex financial data. Additionally, the back-end could be optimized using object-oriented PHP and APIs to improve scalability and facilitate integration with third-party services like bank feeds, digital wallets, or accounting software.

As for future enhancements, there are numerous opportunities to elevate this project into a full-fledged personal finance management platform. One of the major additions could be the implementation of a secure user authentication system with password encryption, role management (admin/user), and session tracking to ensure personalized and private financial data management. Another potential upgrade would involve enabling budget planning and alerts, where users can set spending limits for each category and receive notifications if they exceed those limits. This would serve as a proactive approach to personal financial discipline. Additionally, support for multi-currency transactions, recurring entries for fixed monthly bills, and automatic data backups could enhance the overall usability and robustness of the application. Incorporating a progressive web app (PWA) approach would make the platform accessible even in low-network conditions, and a mobile version—developed using technologies like React Native or Flutter—would cater to users who prefer on-the-go finance tracking. For tech-savvy users or enterprises, advanced features like data export to Excel/PDF, transaction tagging, and API endpoints for integration with external systems could also be introduced. Finally, deploying the application on a live cloud server with SSL support, and building an admin dashboard for monitoring overall system usage and logs, would transition this project from a personal utility to a scalable solution.

In conclusion, the Personal Expense Tracker is not just a project that reflects proficiency in web development and database systems, but it also carries real-world applicability, with the potential to be transformed into a powerful financial assistant. Its simplicity ensures accessibility for users of all backgrounds, while its extendibility opens doors for continuous development, innovation, and commercialization. With the rapid shift toward digital financial solutions, such a project, when enhanced with advanced features and professional polish, can contribute meaningfully to personal finance management practices in today’s tech-savvy world.

CHAPTER 5

5.1 APPENDIX A (Project Source Code)

```
/ Database Connection - DBConnection.java
import java.sql.*;

public class DBConnection {
    private static final String URL = "jdbc:mysql://localhost:3306/expense_tracker";
    private static final String USER = "root";
    private static final String PASSWORD = "your_password";

    public static Connection getConnection() throws SQLException {
        return DriverManager.getConnection(URL, USER, PASSWORD);
    }
}

// User Authentication - AuthService.java
public class AuthService {

    public boolean login(String username, String password) {
        try (Connection conn = DBConnection.getConnection()) {
            String query = "SELECT * FROM users WHERE username = ? AND password = ?";
            PreparedStatement stmt = conn.prepareStatement(query);
            stmt.setString(1, username);
            stmt.setString(2, password);
            ResultSet rs = stmt.executeQuery();
            return rs.next();
        } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}
```

```
  public boolean register(String username, String password) {
        try (Connection conn = DBConnection.getConnection()) {
            String query = "INSERT INTO users(username, password) VALUES(?, ?)";
            PreparedStatement stmt = conn.prepareStatement(query);
            stmt.setString(1, username);
            stmt.setString(2, password);
            stmt.executeUpdate();
            return true;
        } catch (SQLIntegrityConstraintViolationException e) {
            System.out.println("User already exists!");
            return false;
        } catch (Exception e) {
            e.printStackTrace();
            return false;
        }
    }
}

// Dashboard - Dashboard.java
public class Dashboard {
    public static void showDashboard(int userId) {
        try (Connection conn = DBConnection.getConnection()) {
            Statement stmt = conn.createStatement();

            ResultSet incomeRs = stmt.executeQuery("SELECT SUM(amount) FROM income
WHERE user_id=" + userId);
            incomeRs.next();
            double totalIncome = incomeRs.getDouble(1);

            ResultSet expenseRs = stmt.executeQuery("SELECT SUM(amount) FROM expenses
WHERE user_id=" + userId);
```

```
            expenseRs.next();
            double totalExpense = expenseRs.getDouble(1);

            System.out.println("Total Income: " + totalIncome);
            System.out.println("Total Expenses: " + totalExpense);
            System.out.println("Balance: " + (totalIncome - totalExpense));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

// Income Management - IncomeService.java
public class IncomeService {
    public void addIncome(int userId, String source, double amount, String date) {
        try (Connection conn = DBConnection.getConnection()) {
            String query = "INSERT INTO income(user_id, source, amount, date) VALUES(?, ?, ?,
?)";
            PreparedStatement stmt = conn.prepareStatement(query);
            stmt.setInt(1, userId);
            stmt.setString(2, source);
            stmt.setDouble(3, amount);
            stmt.setDate(4, Date.valueOf(date));
            stmt.executeUpdate();
            System.out.println("Income added successfully!");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
// Expense Management - ExpenseService.java
public class ExpenseService {
    public void addExpense(int userId, String category, double amount, String date) {
        try (Connection conn = DBConnection.getConnection()) {
            String query = "INSERT INTO expenses(user_id, category, amount, date) VALUES(?, ?,
?, ?)";
            PreparedStatement stmt = conn.prepareStatement(query);
            stmt.setInt(1, userId);
            stmt.setString(2, category);
            stmt.setDouble(3, amount);
            stmt.setDate(4, Date.valueOf(date));
            stmt.executeUpdate();
            System.out.println("Expense recorded successfully!");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

// Budget Management - BudgetService.java
public class BudgetService {
    public void setBudget(int userId, String category, double limit, String month) {
        try (Connection conn = DBConnection.getConnection()) {
            String query = "INSERT INTO budgets(user_id, category, limit_amount, month)
VALUES (?, ?, ?, ?)";
            PreparedStatement stmt = conn.prepareStatement(query);
            stmt.setInt(1, userId);
            stmt.setString(2, category);
            stmt.setDouble(3, limit);
            stmt.setString(4, month);
            stmt.executeUpdate();
            System.out.println("Budget set successfully.");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

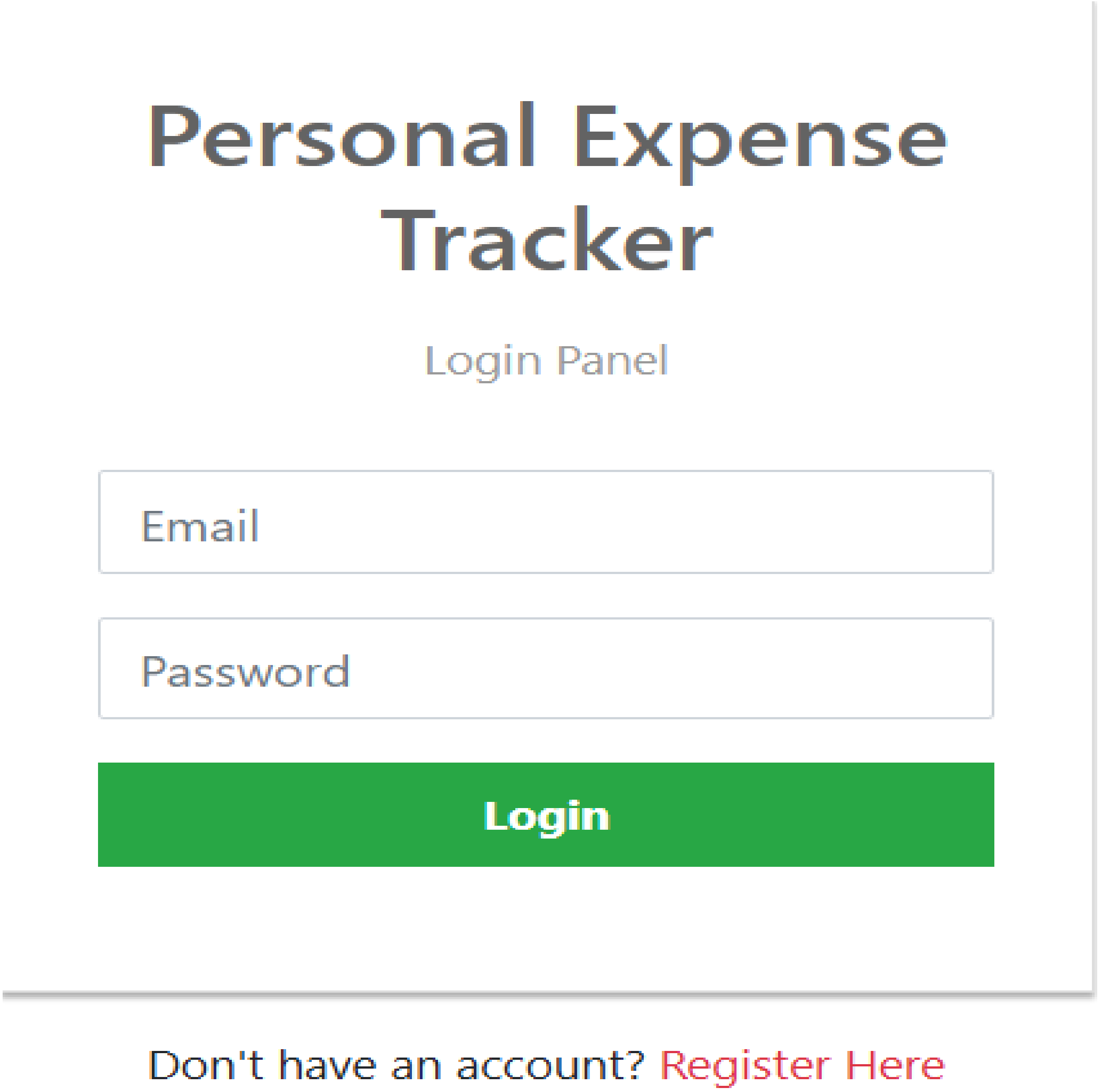
```
// Login Form - LoginForm.java
import javax.swing.*;
        import java.awt.event.*;

public class LoginForm {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Login");
        JLabel userLabel = new JLabel("Username:");
        JLabel passLabel = new JLabel("Password:");
        JTextField userText = new JTextField();
        JPasswordField passText = new JPasswordField();
        JButton loginButton = new JButton("Login");

        userLabel.setBounds(30, 30, 80, 30);
        passLabel.setBounds(30, 70, 80, 30);
        userText.setBounds(120, 30, 150, 30);
        passText.setBounds(120, 70, 150, 30);
        loginButton.setBounds(120, 110, 100, 30);

        frame.add(userLabel);
        frame.add(passLabel);
        frame.add(userText);
        frame.add(passText);
        frame.add(loginButton);
        frame.setSize(350, 200);
        frame.setLayout(null);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        AuthService auth = new AuthService();
        loginButton.addActionListener(e -> {
            String user = userText.getText();
            String pass = String.valueOf(passText.getPassword());
            if (auth.login(user, pass)) {
```

```java
                JOptionPane.showMessageDialog(frame, "Login Successful");
                frame.dispose();
                Dashboard.showDashboard(1); // Replace with real user ID logic
            } else {
                JOptionPane.showMessageDialog(frame, "Login Failed");
            }
        });
    }
}
```

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Personal Expense Tracker</title>
  <link rel="stylesheet" href="style.css" />
</head>
<body>
  <div id="auth-container">
    <h2>Login</h2>
    <input type="text" id="login-username" placeholder="Username" />
    <input type="password" id="login-password" placeholder="Password" />
    <button onclick="login()">Login</button>
    <p>or</p>
    <input type="text" id="register-username" placeholder="New Username" />
    <input type="password" id="register-password" placeholder="New Password" />
    <button onclick="register()">Register</button>
  </div>

  <div id="app-container" style="display:none">
```

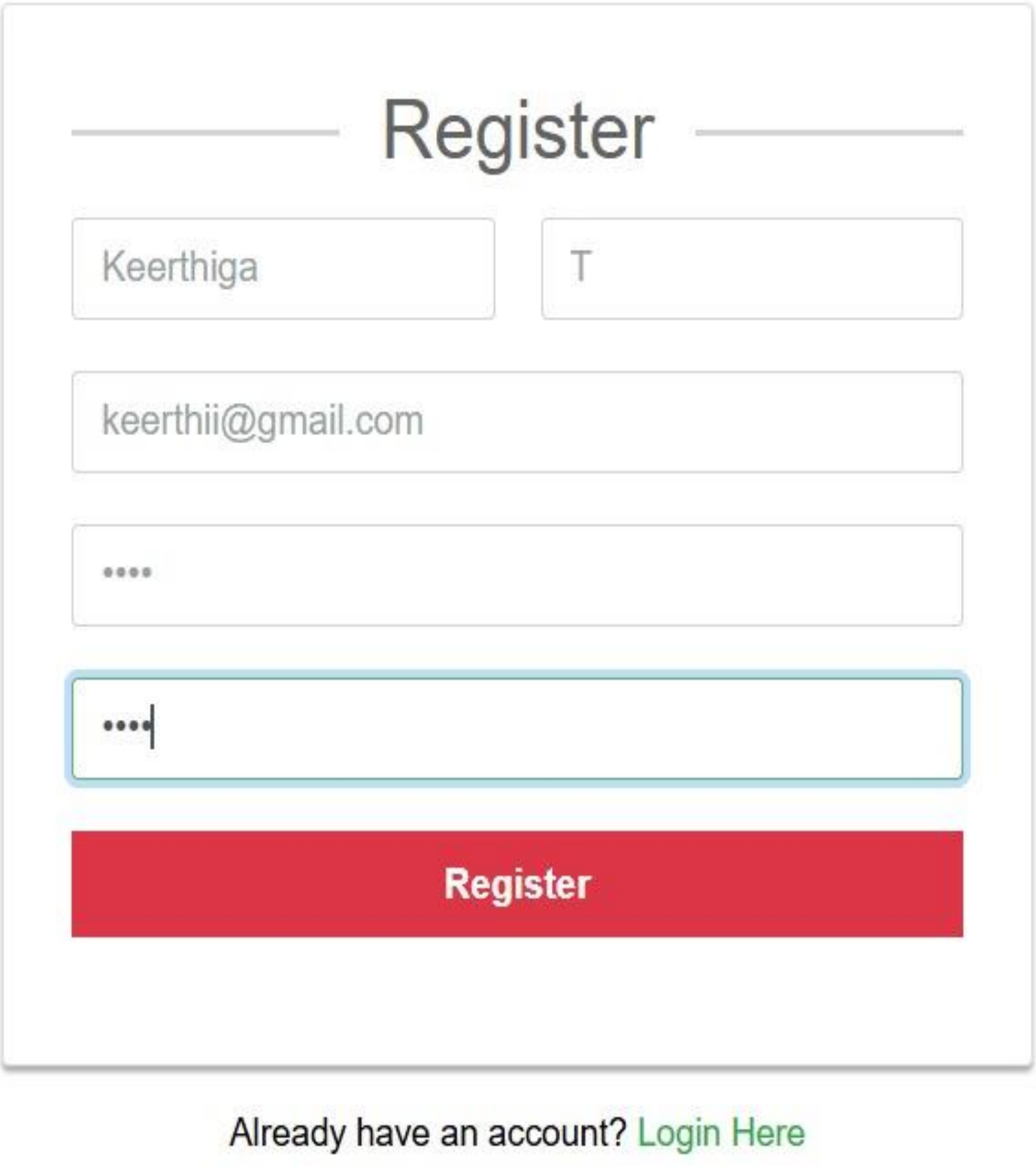
```
    <h2>Dashboard</h2>
    <p>Welcome, <span id="user-display"></span></p>
    <button onclick="logout()">Logout</button>

    <div class="dashboard-section">
      <h3>Balance: <span id="balance">$0.00</span></h3>
      <h4>Income: <span id="income">$0.00</span> | Expense: <span
id="expense">$0.00</span></h4>
    </div>

    <div class="dashboard-section">
      <h3>Add Income</h3>
      <input type="text" id="income-desc" placeholder="Description" />
      <input type="number" id="income-amount" placeholder="Amount" />
      <button onclick="addIncome()">Add Income</button>

      <h3>Add Expense</h3>
      <input type="text" id="expense-desc" placeholder="Description" />
      <input type="number" id="expense-amount" placeholder="Amount" />
      <button onclick="addExpense()">Add Expense</button>
    </div>

    <div class="dashboard-section">
      <h3>Budget Limit</h3>
      <input type="number" id="budget-limit" placeholder="Set Monthly Limit" />
      <button onclick="setBudget()">Set Budget</button>
      <p>Budget: $<span id="budget-display">0</span> | Remaining: $<span id="remaining-
budget">0</span></p>
    </div>
    <div class="dashboard-section">
      <h3>Transaction History</h3>
      <ul id="transaction-history"></ul>
    </div>
  </div>

  <script src="script.js"></script>
</body>
</html>
```

5.2 APPENDIX B (Screenshots)



Don't have an account? Register Here

Fig 5.2.1 User Login page



Register

Keerthiga

T

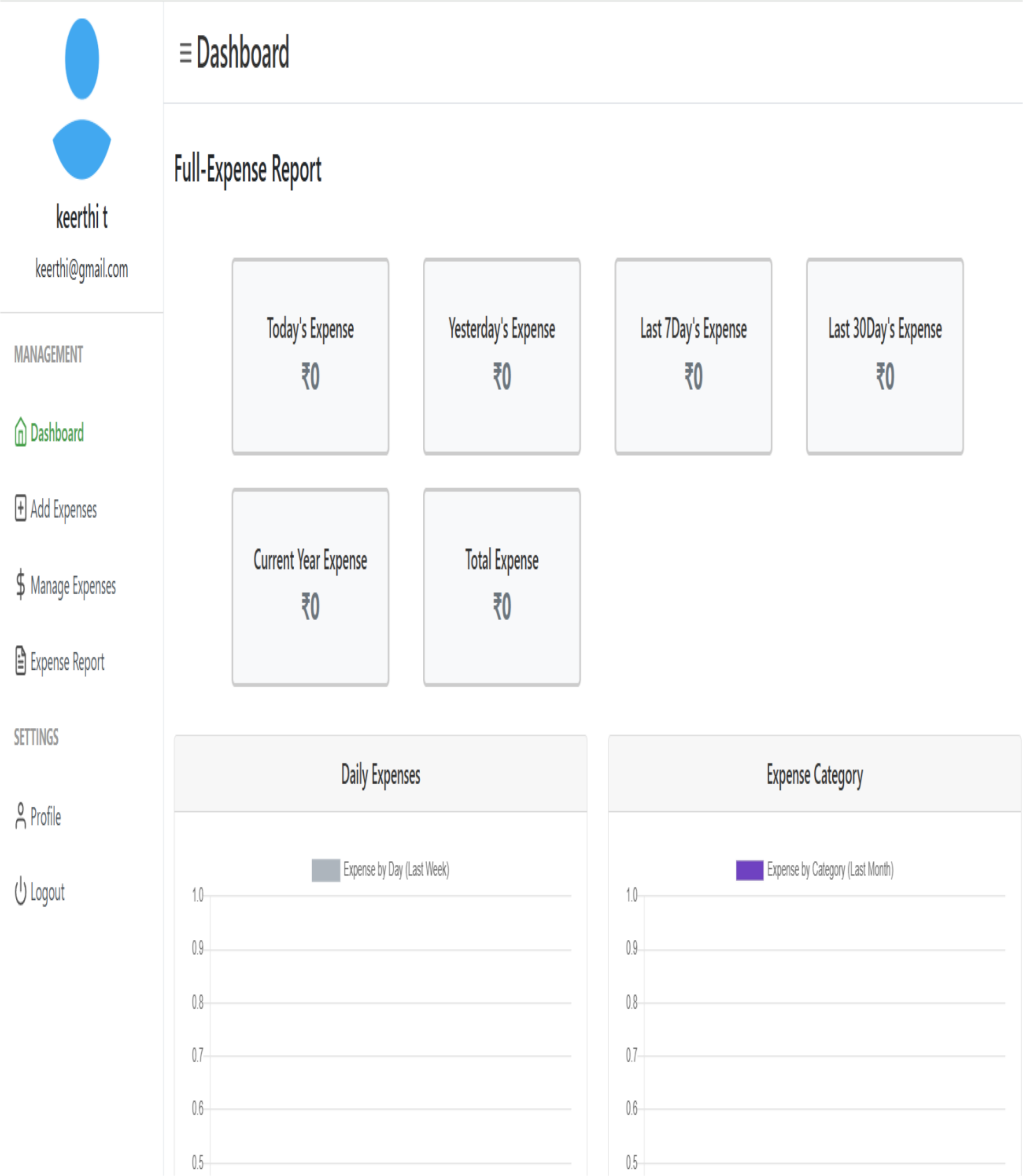
keerthii@gmail.com

••••

••••

Already have an account? Login Here

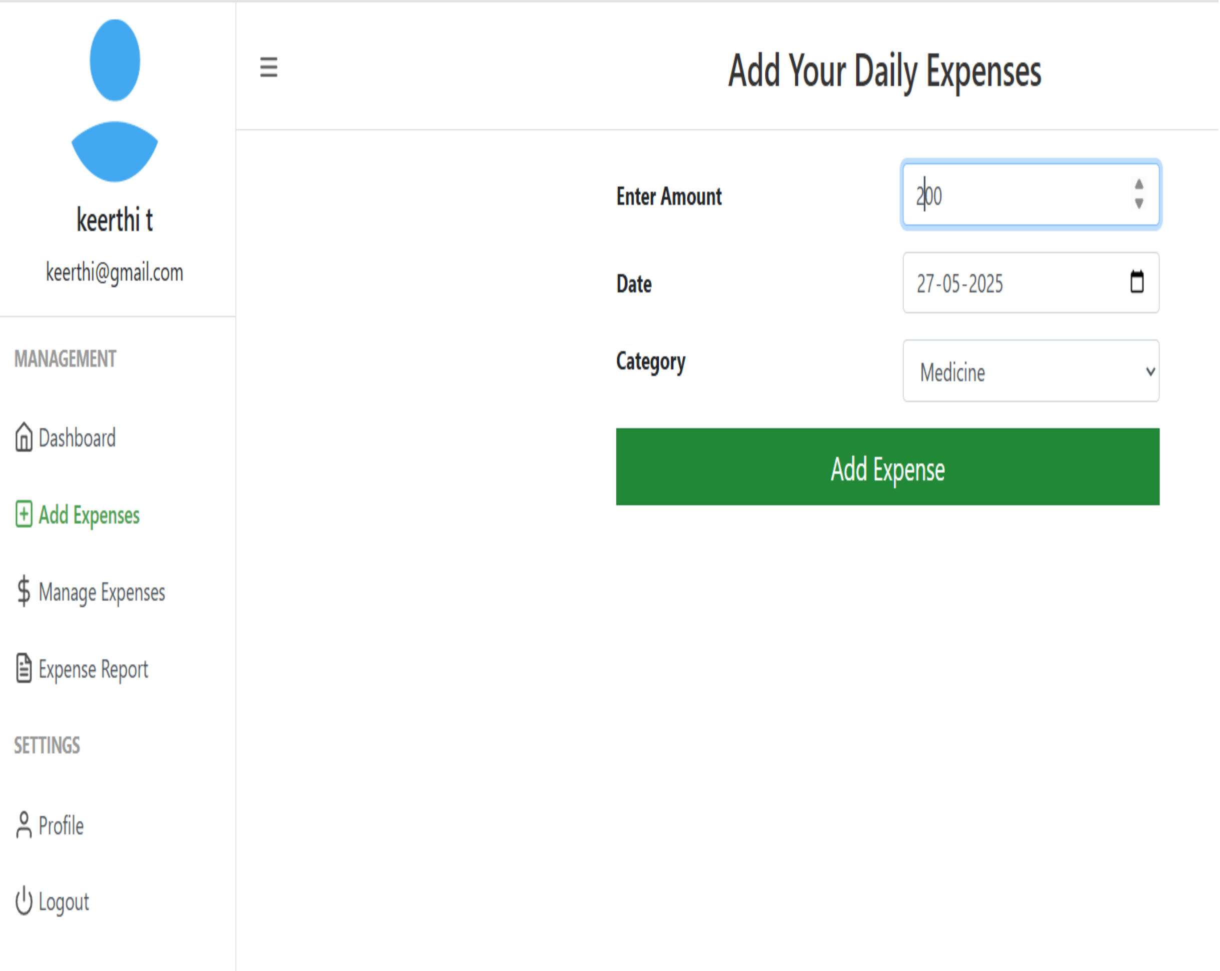
Fig 5.2.2 Registration page



Full-Expense Report

Today's Expense	₹0
Yesterday's Expense	₹0
Last 7Day's Expense	₹0
Last 30Day's Expense	₹0
Current Year Expense	₹0
Total Expense	₹0

Fig 5.2.3 User Dashboard page



keerthi t

keerthi@gmail.com

MANAGEMENT

- Dashboard
- Add Expenses
- Manage Expenses
- Expense Report

SETTINGS

- Profile
- Logout

Add Your Daily Expenses

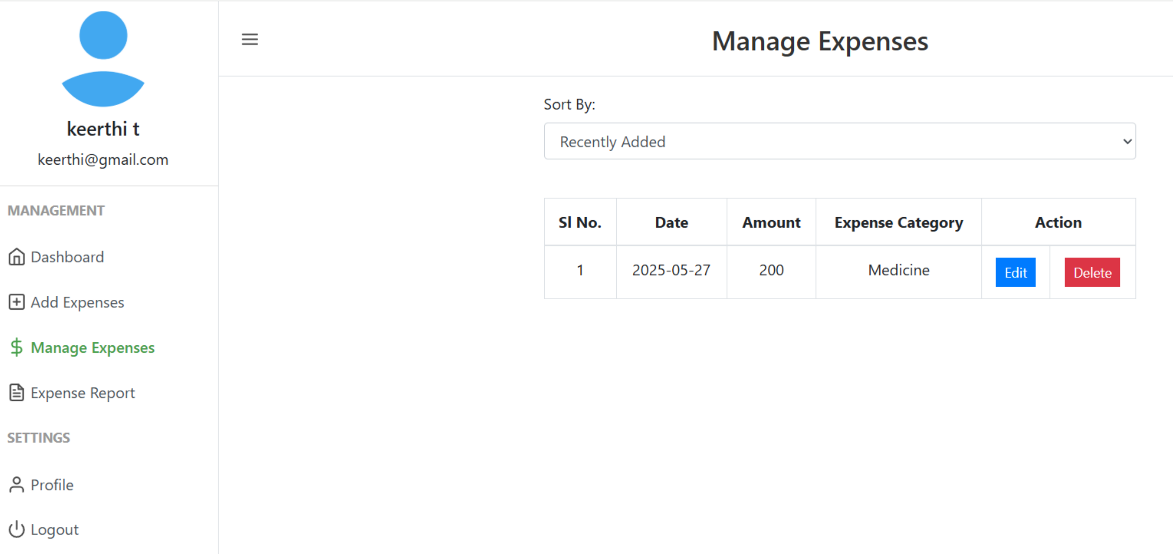
Enter Amount: 200

Date: 27-05-2025

Category: Medicine

Add Expense

Fig 5.2.4 Income Managemet page



keerthi t

keerthi@gmail.com

MANAGEMENT

- Dashboard
- Add Expenses
- Manage Expenses
- Expense Report

SETTINGS

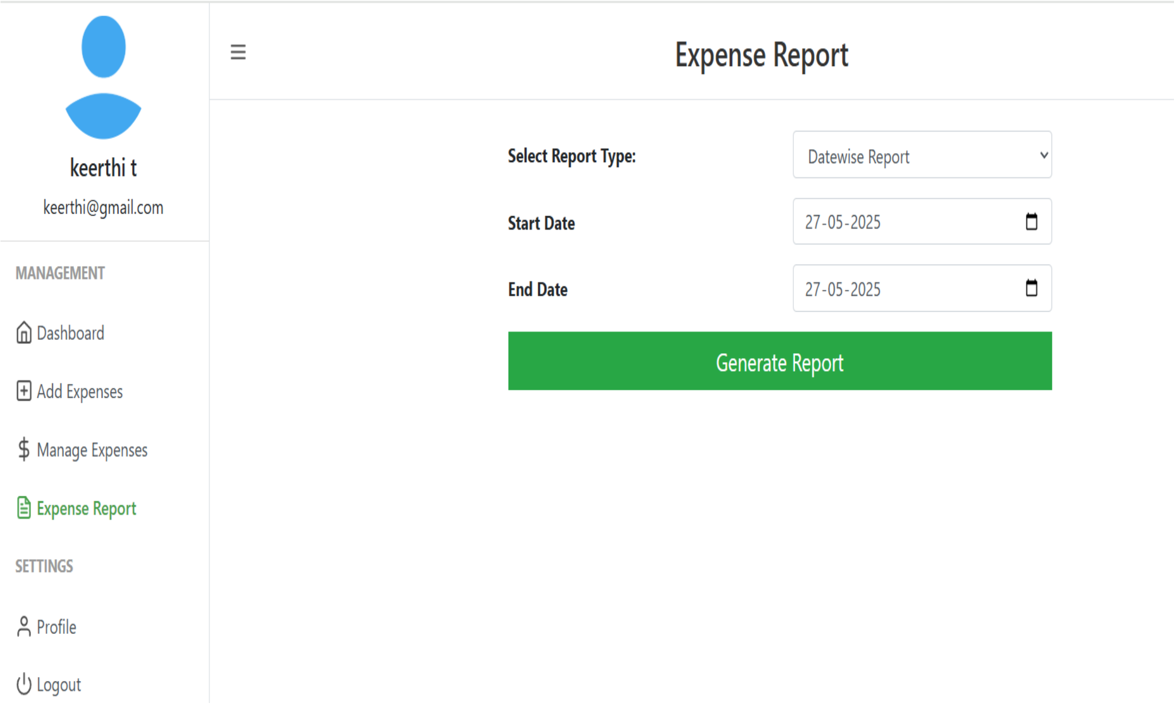
- Profile
- Logout

Manage Expenses

Sort By: Recently Added

Sl No.	Date	Amount	Expense Category	Action	
1	2025-05-27	200	Medicine	Edit	Delete

Fig 5.2.5 Expense Management page



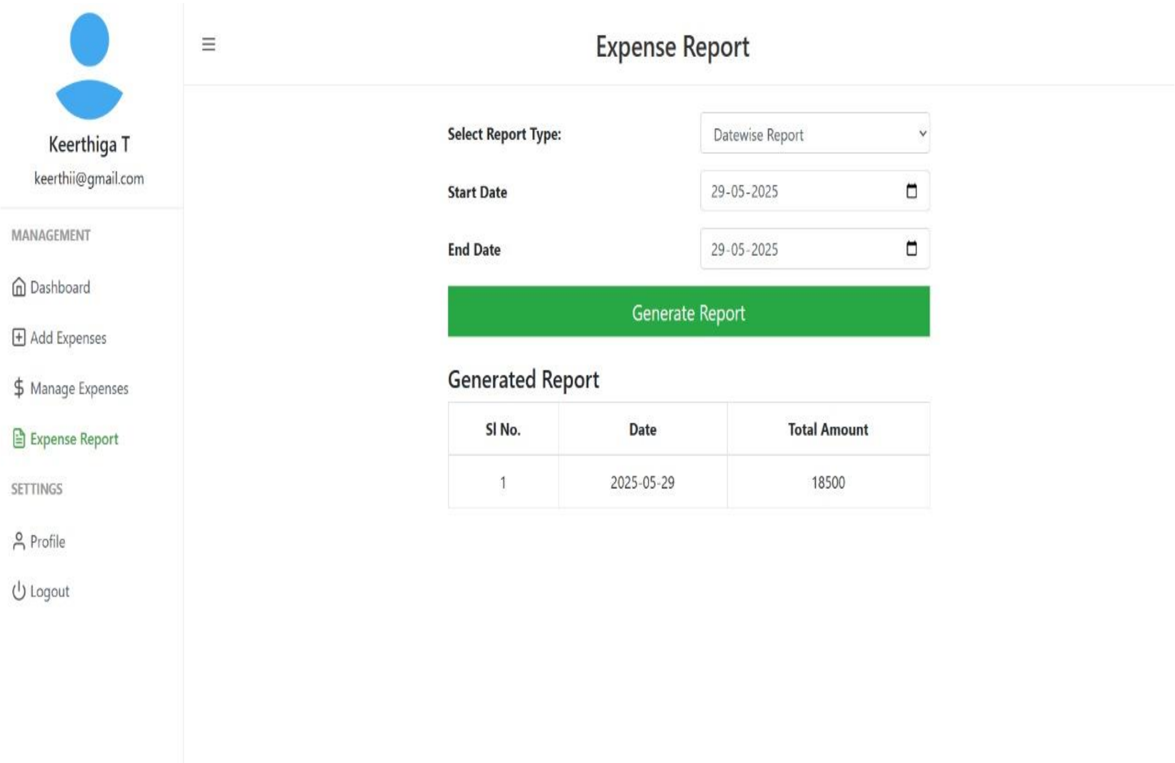
Expense Report

Select Report Type: Datewise Report

Start Date 27-05-2025

End Date 27-05-2025

Fig 5.2.6 Report Generation page



Expense Report

Select Report Type: Datewise Report

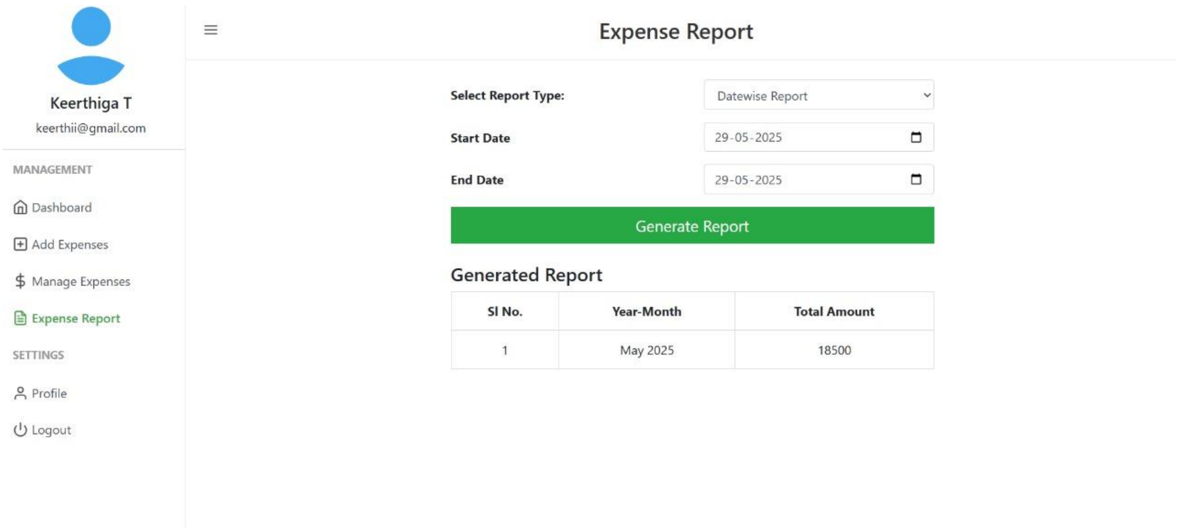
Start Date 29-05-2025

End Date 29-05-2025

Generated Report

Sl No.	Date	Total Amount
1	2025-05-29	18500

Fig 5.2.7 Datewise Report



Expense Report

Select Report Type: Datewise Report

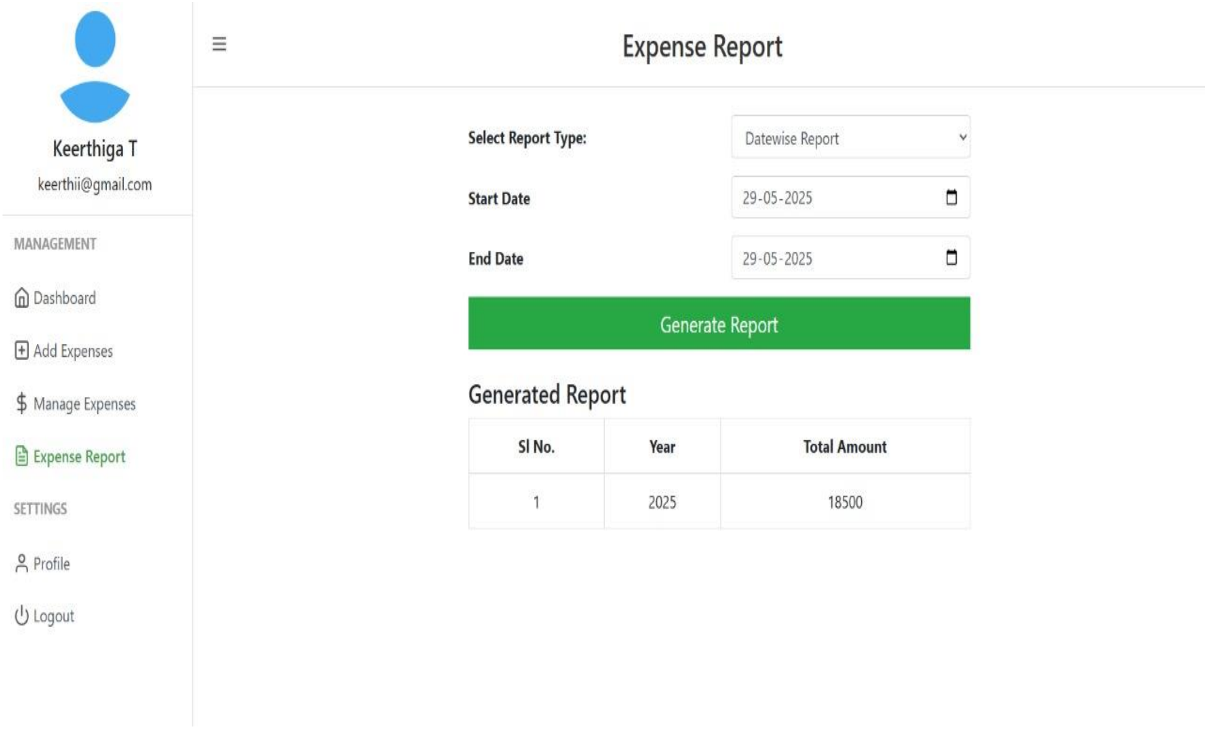
Start Date 29-05-2025

End Date 29-05-2025

Generated Report

Sl No.	Year-Month	Total Amount
1	May 2025	18500

Fig 5.2.8 Monthwise Report



Expense Report

Select Report Type: Datewise Report

Start Date 29-05-2025

End Date 29-05-2025

Generated Report

Sl No.	Year	Total Amount
1	2025	18500

Fig 5.2.9 Yearwise Report

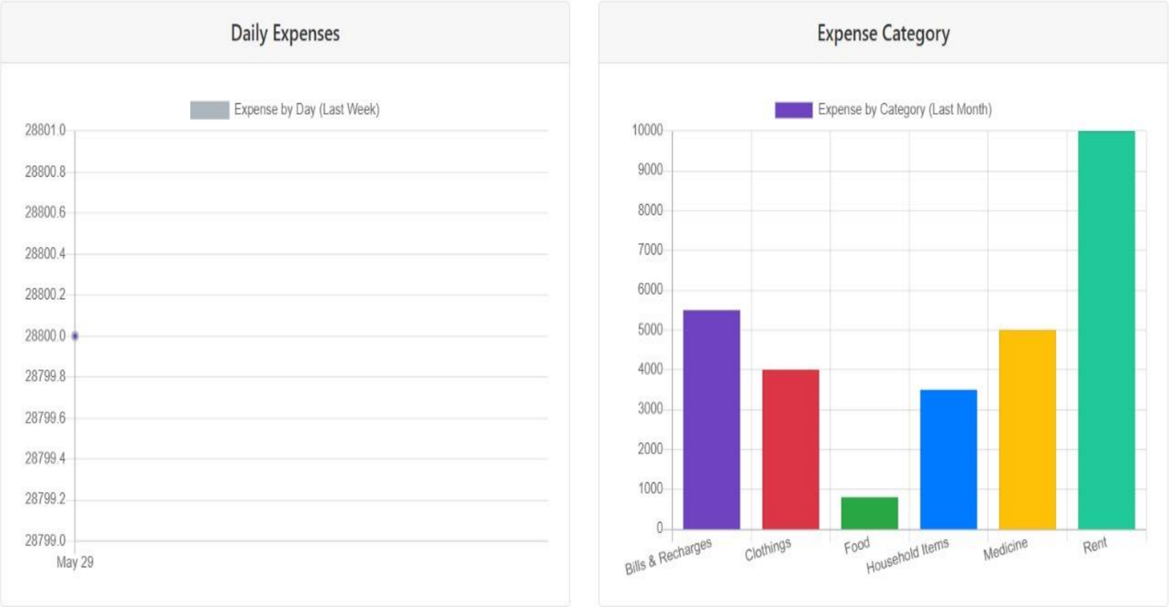


Fig 5.2.10 Categorywise Expense Chart

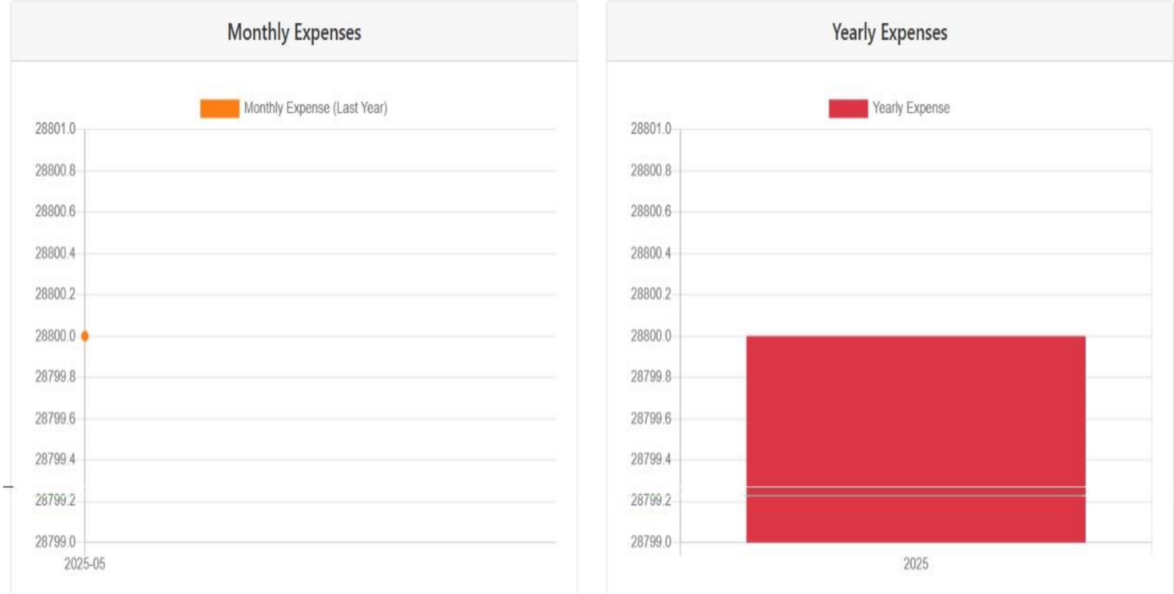
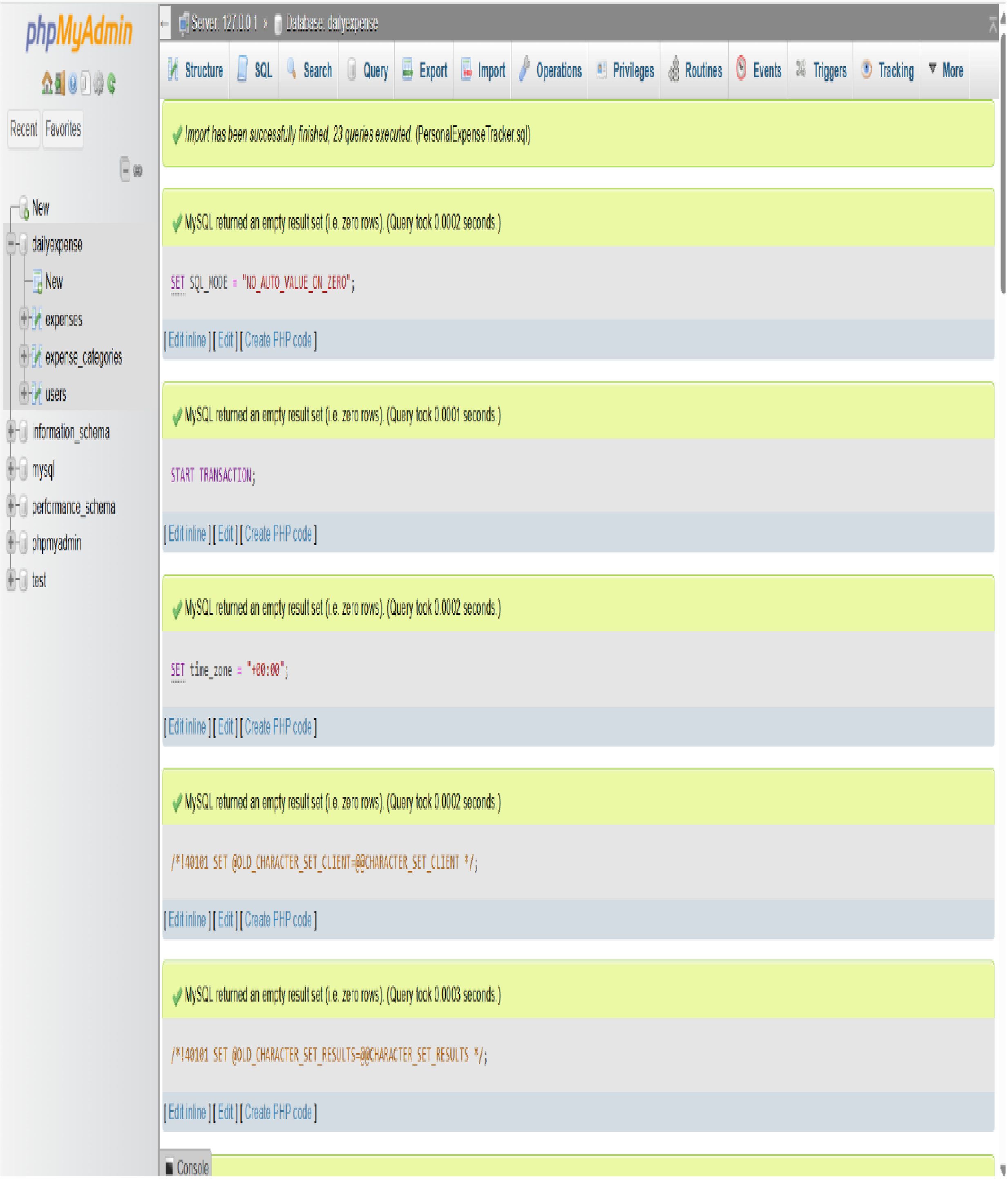


Fig 5.2.11 Yearly Expense Chart



✓ Import has been successfully finished, 23 queries executed. (PersonalExpenseTracker.sql)

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
```

[Edit inline] [Edit] [Create PHP code]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0001 seconds.)

```
START TRANSACTION;
```

[Edit inline] [Edit] [Create PHP code]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)

```
SET time_zone = "+00:00";
```

[Edit inline] [Edit] [Create PHP code]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
```

[Edit inline] [Edit] [Create PHP code]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)

```
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
```

[Edit inline] [Edit] [Create PHP code]

Fig 5.2.12 SQL Import Success

CHAPTER 6

REFERENCES:

- "Java: The Complete Reference" by Herbert Schildt.
- "Database System Concepts" by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan.
- "Head First Java" by Kathy Sierra & Bert Bates.
- W3Schools SQL Tutorial
- https://www.youtube.com/watch?v=QwrIqj2uI54
- https://www.youtube.com/watch?v=MSlVHE2yDWE
- XAMPP by Apache Friends. "Easy Apache + MySQL + PHP environment." [Online]. Available: https://www.apachefriends.org/
- GeeksforGeeks. "Web Development Tutorials & Projects." [Online]. Available: https://www.geeksforgeeks.org/