** GCUF

GOVERNMENT COLLEGE UNIVERSITY FAISALABAD

FINAL REPORT

ExpenseHub

(Expense Management System)

**Submitted By**

**Shiza Malik**

**2020-GCUF-059987**



**DEPARTMENT OF COMPUTER SCIENCE**

**Govt. College of Science Samanabad, Faisalabad**

**2024**

# DECLARATION

I hereby declare that the content of the project, **“ExpenseHub”** are product of my own research and no part has been copied from any published source (except the references, some standard mathematical or genetic models / equations / protocols etc.). I further declare that this work has not been submitted for award of any other diploma/degree. The university can take action if the above statement is found inaccurate at any stage.

**Signature**

**Shiza Malik**

**2020-GCUF-059987** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CERTIFICATE BY SUPERVISORY COMMITTEE**

We certify that the contents and the form of project submitted by **Shiza Malik Roll # 3215/459875**, have been found satisfactory and recommend it to be processed for evaluation by the External Examiner(s) for the award of degree.

**Supervisors:**

**Faiza Batool Signature** : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Internal Panel**

**Member 1: Name Signature** : \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Member 2: Name Signature** : \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Member 3: Name Signature** : \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Project Coordinator Head of Department**

**Principal**

# DEDICATION

***To My Beloved Parents***

&

***Teachers***

**ACKNOWLEDGEMENT**

All praise to ALMIGHTY ALLAH, the most merciful and the most compassionate and his Holy Prophet .MUHAMMAD'(Peace) be upon him) the most perfect and exalted among and even bon) on the surface of earth, who is, forever a torch of guidance and knowledge for the humanity as a whole.

The work presented in this manuscript was accomplished under the inspiring guidance, gorgeous assistance, constructive criticism and enlightened supervision of **Miss. Faiza Batool** Department of Computer Science in Govt. College of Science Samanabad, Faisalabad for his skillful guidance, constructive criticism, masterly advice, valuable suggestions and sympathetic behavior for the completion of this manuscript

I feel highly privileged to take this opportunity to express one heartiest gratitude and deep sense of indebt to our worthy supervisory committee, **Miss. Faiza Batool** of Computer Science Govt. College of Science Samanabad, Faisalabad under whose kind and scholastic guidance, keen interest and constant encouragement.

Words are very important to convey thoughts and thanks, the words are impossible to find thank our Father and whole family for their prayers and encouragement for us and for our work.

Finally, we are apologizing if we have caused anger of offence to anybody and the errors that remain in the manuscript are mine alone.

Shiza Malik

# Abstract

The ExpenseHub project is a comprehensive expense management platform designed to provide users with a practical solution for tracking and managing their daily expenses. Built with modern client-side technologies such as React.js, HTML, and CSS, along with PHP and sql on the server side, ExpenseHub allows users to efficiently record, categorize, and analyze their spending patterns. The primary goal of this project is to simplify personal finance management by offering an intuitive interface where users can input expenses, view detailed reports, and set budgeting goals.

Key features of ExpenseHub include a dynamic dashboard for real-time expense tracking, user authentication for secure access, and advanced categorization to organize expenses by type, date, and source. The platform also supports data visualization, enabling users to view spending trends through charts and graphs. Future enhancements may include multi-currency support, integration with bank APIs for automated transaction imports, and personalized financial insights. This project serves as a powerful learning tool for students to gain hands-on experience in building financial management systems while addressing real-world challenges in personal expense tracking.

This documentation provides a detailed analysis of the project’s architecture, functional and non-functional requirements, and potential future developments. By emphasizing practical application and user experience, the project aims to bridge theoretical knowledge with real-world e-commerce solutions.

Table of Contents

[DECLARATION ii](#_Toc179914305)

[DEDICATION iv](#_Toc179914306)

[ACKNOWLEDGEMENT v](#_Toc179914307)

[Abstract vi](#_Toc179914308)

[Chapter 1 1](#_Toc179914309)

[INTRODUCTION TO THE PROBLEM 1](#_Toc179914310)

[1.3 Background Study 4](#_Toc179914311)

[1.4 Purpose 6](#_Toc179914312)

[1.5 Scope 7](#_Toc179914313)

[1.6 Objective 7](#_Toc179914314)

[1.7 Intended Audience and Reading Suggestions 7](#_Toc179914315)

[1.8 Documentation Conventions 11](#_Toc179914316)

[Chapter 2 12](#_Toc179914317)

[SOFTWARE REQUIREMENT SPECIFICATION 12](#_Toc179914318)

[2.1 User Registration Requirements 12](#_Toc179914319)

[2.2 External Interface Requirements 13](#_Toc179914320)

[2.3 Other Non-Functional Requirements 22](#_Toc179914321)

[**2.4 Architecture Diagram** 27](#_Toc179914322)

[**2.5 Entity-Relationship Diagram (ERD)** 29](#_Toc179914323)

[**2.6 Data Flow Diagram (DFD)** 32](#_Toc179914324)

[**2.8 Object Diagram** 38](#_Toc179914325)

[**2.9 Sequence Diagram** 40](#_Toc179914326)

[**2.10 Activity Diagram** 41](#_Toc179914327)

[**2.11 Collaboration Diagram** 42](#_Toc179914328)

[**2.12 State Transition Diagram** 43](#_Toc179914329)

[Chapter 3 45](#_Toc179914330)

[ANALYSIS (USE CASE MODEL) 45](#_Toc179914331)

[**3.1 Identifying Actors and Use Cases using Textual Analysis** 45](#_Toc179914332)

[**3.2 Forming Use Case Diagram with Candidate and Use Cases** 46](#_Toc179914333)

[**3.3 Describe the Events Flow for Use Case** 47](#_Toc179914334)

[Chapter 4 49](#_Toc179914335)

[DESIGN 49](#_Toc179914336)

[4.1 Overall Description 49](#_Toc179914337)

[**4.1.1 Product Perspective** 49](#_Toc179914338)

[**4.1.4 Assumption and Dependencies** 50](#_Toc179914339)

[4.2 System Features 50](#_Toc179914340)

[**4.2.1 Log Expense Page** 50](#_Toc179914341)

[**4.2.2 Contact Us Page** 51](#_Toc179914342)

[**4.2.3 About Us Page** 51](#_Toc179914343)

[**4.2.4 Sign up** 51](#_Toc179914344)

[**4.2.5 Login** 52](#_Toc179914345)

[**4.2.8 Reset Password Flow Chart** 53](#_Toc179914346)

[**4.2.6 Edit Profile Flow Chart** 53](#_Toc179914347)

[**4.2.7 Search Expense Flow Chart** 54](#_Toc179914348)

[**4.2.8 Logout** 54](#_Toc179914349)

[**4.2.9 Report Generation** 55](#_Toc179914350)

[**4.2.10 Dashboard** 55](#_Toc179914351)

[**4.2.11 Add Expense** 55](#_Toc179914352)

[**4.2.12 Search Expense** 55](#_Toc179914353)

[**4.2.13 View Expense Details** 55](#_Toc179914354)

[**4.2.14 Update/Delete Expense** 56](#_Toc179914355)

[**4.2.15 Expense Graphs** 56](#_Toc179914356)

[Chapter 5 56](#_Toc179914357)

[IMPLEMENTATION 56](#_Toc179914358)

[**5.1 Component Diagram** 56](#_Toc179914359)

[**5.2 Deployment Diagram** 57](#_Toc179914360)

[**5.3 Database Architecture (1- Tier, 2-Tier, 3- Tier Architecture)** 58](#_Toc179914361)

[Chapter 6 61](#_Toc179914362)

[TESTING (SOFTWARE QUALITY ATTRIBUTES) 61](#_Toc179914363)

[**6.1 Test Case Specification** 61](#_Toc179914364)

[**6.2 Black Box Test Cases** 61](#_Toc179914365)

[**6.3 White Box Testing** 63](#_Toc179914366)

[Chapter 7 65](#_Toc179914367)

[TOOL & TECNOLOGIES 65](#_Toc179914368)

[**7.1 Programming Languages** 65](#_Toc179914369)

[**7.2 Operating Environment** 66](#_Toc179914370)

[**Appendix A: User documentation** 74](#_Toc179914371)

**LIST OF FIGURE**

|  |  |  |
| --- | --- | --- |
| **Figure no. No** | **Statement** | **Page No** |
| Figure no. 2.1 | MVC Diagram | 29 |
| Figure no. 2.2 | ER-Diagram | 32 |
| Figure no. 2.3 | DFD (L1) | 36 |
| Figure no. 2.4 | DFD (L2) | 36 |
| Figure no. 2.5 | Class Diagram | 38 |
| Figure no. 2.6 | Object Diagram | 39 |
| Figure no. 2.7 | Sequence Diagram | 41 |
| Figure no. 2.8 | Activity Diagram | 42 |
| Figure no. 2.9 | Collaboration Diagram | 43 |
| Figure no. 2.10 | State Transition Diagram | 44 |
| Figure no. 3.1 | ERD Site Flow | 46 |
| Figure no. 4.1 | Constraints and Implementation Constraints | 49 |
| Figure no. 4.2 | Login Flow Chart | 52 |
| Figure no. 4.3 | Reset Password Flow Chart | 53 |
| Figure no. 4.4 | Edit Profile Flow Chart | 53 |
| Figure no. 4.5 | Search Product Flow Chart | 54 |
| Figure no. 4.6 | Add Product Cart and Checkout Flow Chart | 56 |
| Figure no. 5.1 | Component Diagram | 57 |
| Figure no. 5.2 | Deployment Diagram | 58 |
| Figure no. 5.3 | Database Architecture Diagram | 60 |
| Figure no. 7.1 | VS Code | 66 |
| Figure no. 7.2 | Data Flow Diagram L0 | 69 |
| Figure no. 7.3 | Data Flow Diagram L1 | 71 |

# Chapter 1

# INTRODUCTION TO THE PROBLEM

**1.1 Introduction**

The comprehensive documentation of "ExpenseHub", an expense management application developed using HTML, CSS, JavaScript, jQuery, and SQL, provides a detailed understanding of the project, its objectives, scope, and functionality. This documentation is intended for students and developers interested in web development, application design, and personal finance management systems.

Managing personal finances effectively is essential for maintaining financial health, and ExpenseHub serves as a user-friendly platform for tracking and managing daily expenses. Built with core web technologies, HTML and CSS provide the structure and styling for a clean and responsive interface, while JavaScript and jQuery enhance interactivity, allowing users to dynamically interact with the application. The backend is powered by SQL, which manages the storage and retrieval of expense data, ensuring a reliable and efficient database structure for users to log, categorize, and review their financial records.

The primary goal of ExpenseHub is to simplify the process of tracking expenses. Users can easily register, log in, and access a dashboard that displays their spending summary. Key features of the platform include the ability to add new expenses, search for specific entries, view detailed expense reports, and update or delete expense records. The application also generates expense reports in a variety of formats, including visual graphs and charts, allowing users to gain insights into their spending patterns over time.

With its simple yet powerful design, ExpenseHub offers an intuitive and interactive experience, making it an ideal tool for individuals who want to better manage their finances. Additionally, for students learning web development, the project serves as an excellent example of integrating front-end technologies with a robust SQL-based backend, providing real-world experience in building dynamic and functional applications.

**1.2 Overview**

Effective expense management is an essential aspect of personal finance, helping individuals and businesses track their spending, manage budgets, and make informed financial decisions. The ExpenseHub application offers a comprehensive solution for managing day-to-day expenses by providing users with a robust platform to log, categorize, and analyze their financial transactions. The primary goal of this project is to develop a general-purpose expense management system where users can easily record their expenditures, view detailed reports, and gain insights into their spending patterns, all through a user-friendly interface. ExpenseHub enables users to track various expenses, ranging from personal purchases to business-related transactions, all in one place.

Built using HTML, CSS, JavaScript, jQuery, and SQL, ExpenseHub focuses on providing an intuitive experience where users can easily input and manage their expenses. The application offers key features such as user registration and login, allowing individuals to create secure profiles where their financial data is stored. Once logged in, users can access the dashboard, which offers an overview of their spending. They can add expenses, search for specific transactions, and view expense details to monitor where their money is going. Update and delete expense functionalities are also provided, ensuring that users can keep their records up to date and correct any mistakes in their transaction history.

ExpenseHub goes beyond simple expense logging by offering expense reports that give users a clear breakdown of their spending in various categories, such as groceries, transportation, or entertainment. The application also provides graphs and charts to help users visualize their spending patterns over time, making it easier to identify trends and make adjustments to their budgets. This feature is particularly helpful for users who want to gain a deeper understanding of their financial habits and set goals for future spending.

In addition to managing expenses, users can also update their profile information and securely log out once they have finished using the system. The application’s SQL-based backend ensures that all expense data is stored reliably, and users can access their historical spending data whenever they need it. By making use of jQuery, the interface is highly interactive, providing a smooth and responsive user experience when adding, editing, or viewing expenses.

The ExpenseHub system offers a solution to many of the challenges faced in traditional expense management. Without such a system, individuals may rely on manual processes like keeping receipts or recording expenses in spreadsheets, which can be time-consuming, prone to errors, and difficult to maintain. Manual tracking also lacks the ability to provide real-time insights or automated reports, which are essential for making informed financial decisions. The ExpenseHub platform addresses these shortcomings by offering a centralized, automated system for expense tracking that is accessible from any device with an internet connection.

In traditional methods of expense tracking, individuals may struggle to maintain accurate records, especially when dealing with multiple types of expenditures. ExpenseHub resolves this issue by allowing users to categorize expenses and view them in a structured format, enabling easier management and control over finances. Unlike spreadsheets or manual logs, which require users to create their own templates and systems for tracking expenses, ExpenseHub provides an out-of-the-box solution that streamlines the entire process.

Furthermore, users can generate various kinds of reports, including detailed breakdowns of their spending, and visualize their expenses through graphs and charts. These reports enable users to quickly assess their financial health, detect overspending in certain areas, and make adjustments to their budgets accordingly. The system's integration of multiple features like search, update, and delete, combined with its automated reporting functionality, ensures that users can manage their expenses efficiently, saving both time and effort.

ExpenseHub is particularly beneficial in today’s busy lifestyle, where managing personal or business finances can become overwhelming without the proper tools. By providing a centralized platform for logging and analyzing expenses, users can focus more on making sound financial decisions rather than spending time on tedious manual tasks. The convenience of accessing financial data anytime from any location further enhances the value of the system, making it a highly practical tool for individuals and businesses alike.

ExpenseHub serves as an effective and modern solution for personal and business expense management. It allows users to track their spending, generate reports, and make data-driven financial decisions, all through an easy-to-use interface. As a full-featured expense management platform, ExpenseHub eliminates many of the challenges associated with traditional expense tracking and empowers users to take control of their financial health. The application's ability to generate insightful reports and visualize data in real-time makes it a valuable tool for anyone looking to better manage their finances.

## 1.3 Background Study

Effective expense management is critical for maintaining financial stability, both on a personal and business level. Traditional methods of managing expenses involved manually tracking expenditures through paper receipts or basic spreadsheets, which proved to be time-consuming, prone to errors, and limited in their ability to provide comprehensive insights. As technology advanced, digital tools and applications emerged, making it easier for individuals and businesses to manage their finances. ExpenseHub is an example of such a tool, designed to offer a user-friendly platform for logging, categorizing, and analyzing expenses. This project aims to streamline the process of tracking daily expenditures and provide users with detailed reports and visual insights into their financial habits.

The concept of expense management has evolved significantly over the years. Initially, individuals would manually record their spending and often found it challenging to keep track of their budgets, especially when dealing with multiple categories such as groceries, rent, utilities, and personal expenses. The rise of digital tools, including expense management software and mobile apps, has revolutionized how people handle their finances. Today, users expect features that go beyond basic logging, such as real-time updates, data visualization through graphs and charts, and the ability to categorize expenses in an organized manner. ExpenseHub addresses these needs by offering a full suite of features, including add expense, search expense, and view detailed expense reports, all designed to simplify the expense-tracking process.

The growth of personal finance applications has been driven by several factors, including the widespread adoption of mobile devices and increased awareness of the importance of financial literacy. Many individuals now prefer to track their expenses digitally rather than relying on outdated manual methods. Expense management apps like ExpenseHub allow users to input their spending data quickly and efficiently, helping them monitor their financial behavior, identify patterns, and make informed decisions. ExpenseHub uses a combination of HTML, CSS, JavaScript, jQuery, and SQL to create a seamless and responsive platform that is accessible to users from any location with an internet connection. The platform’s ability to store and retrieve data from an SQL backend ensures that users' financial information is secure and easily accessible for future reference.

The rise of technology has not only simplified the process of recording expenses but has also introduced advanced features like expense reports and graphs. These tools provide users with visual representations of their spending habits, making it easier to understand where their money is going. ExpenseHub includes these features, allowing users to generate detailed reports and view graphical representations of their expenses by category, time period, or other criteria. This functionality helps users to identify areas where they may be overspending and make adjustments to improve their financial health. By offering these tools in a single platform, ExpenseHub enhances users' ability to manage their finances effectively.

In today's fast-paced world, individuals are increasingly seeking convenient solutions that allow them to manage their financial lives efficiently. ExpenseHub provides a platform where users can track their expenses in real-time, update or delete entries, and receive detailed feedback on their spending patterns. The platform also includes user registration, login, and profile management features, ensuring that each user’s data is secure and personalized. Users can access their financial data at any time, making it easy to keep track of their expenses, even on the go. The platform's design ensures that it is both user-friendly and robust, capable of handling a variety of user needs from personal expense tracking to business financial management.

One of the key challenges in traditional expense management systems is the lack of automation and the inability to generate detailed insights. Manual methods require individuals to spend considerable time reviewing their records and calculating totals, which can lead to inaccuracies. ExpenseHub resolves this issue by automating much of the process, including generating reports that break down spending into categories like food, transportation, and entertainment. This not only saves time but also provides users with accurate, up-to-date information about their financial standing.

The success of any expense management platform depends on its ability to provide a simple yet powerful interface while ensuring the security and integrity of users' data. ExpenseHub achieves this balance by incorporating secure login and data management features, while also offering users the ability to categorize, edit, and delete their expenses. The platform’s intuitive design, combined with its comprehensive set of features, makes it an ideal tool for individuals looking to gain control over their finances. Whether users are tracking personal expenses or managing a business budget, ExpenseHub provides the necessary tools to make informed financial decisions.

ExpenseHub reflects the growing need for digital solutions that can efficiently manage personal and business expenses. The project leverages modern technologies to create a user-friendly and secure platform that simplifies the process of tracking expenses, generating reports, and visualizing financial data. With the rise of mobile devices and the increasing importance of financial literacy, tools like ExpenseHub play a critical role in helping individuals and businesses maintain control over their finances. The platform’s ability to automate much of the expense-tracking process and provide insightful reports makes it an invaluable tool for users seeking to improve their financial health.

## 1.4 Purpose

The The primary purpose of "ExpenseHub" is to provide users with an efficient tool for managing and tracking their daily expenses, while also serving as an educational resource for students interested in web development. By leveraging technologies such as HTML, CSS, JavaScript, jQuery, and SQL, students and developers alike can gain practical experience in building interactive and data-driven web applications.

**Key objectives of ExpenseHub include:**

* Automation of expense tracking: Simplify the process of recording and managing daily expenses.
* Expense categorization: Allow users to categorize and track different types of expenditures.
* Detailed reporting and visualization: Generate and display insightful reports, graphs, and charts to help users understand their spending patterns.
* Consistent data updates: Ensure that all expense data is updated in real-time for accurate financial management.
* Security and privacy: Provide a secure platform with authorized user access through proper email and password authentication.
* Efficient database management: Handle expense data effectively using SQL to store, retrieve, and update user records.

## 1.5 Scope

The "ExpenseHub" project has a wide-ranging scope, offering a comprehensive platform for users to manage and track their daily expenses anytime, from anywhere. Available 24x7, ExpenseHub allows users to log, categorize, and analyze their expenses with ease, providing valuable insights into their spending habits. In the future, the system could be expanded to integrate with bank APIs, enabling automated imports of financial transactions, and offering multi-currency support for international users. Additionally, features such as advanced budgeting tools and personalized financial insights could be added, enhancing the overall user experience and helping individuals maintain better control over their finances.

## 1.6 Objective

The motivation behind developing the "ExpenseHub" application arises from the need for a practical and efficient solution to manage daily expenses, along with a desire to create a streamlined, user-friendly platform for personal finance management. The project aims to demonstrate the effective use of HTML, CSS, JavaScript, jQuery, and SQL in building an interactive and functional web application. Beyond its utility, ExpenseHub serves as an educational tool, offering students and developers valuable insights into designing and implementing real-world financial tracking systems. The project also emphasizes the importance of organizing financial data, generating reports, and visualizing expenses, making it an essential resource for improving financial literacy and development skills.

## 1.7 Intended Audience and Reading Suggestions

This documentation is primarily intended for computer science students, web developers, and anyone interested in developing web-based financial management systems. If you are new to web development or expense management, it is recommended to start from the beginning of this document and proceed sequentially to gain a comprehensive understanding of the system's design and functionality. For experienced developers or users familiar with similar systems, you can skip to specific sections that align with your interests or needs, such as the implementation of expense tracking, report generation, or the use of SQL for database management.

|  |  |
| --- | --- |
| **Agency Goals** | **Project Objectives** |
| A well-defined goal ensures that everyone on the team is aligned and focused on the primary objectives, creating a clear path toward success. In the case of ExpenseHub, establishing clear goals helps the team prioritize tasks, allocate resources effectively, and maintain a shared understanding of what needs to be achieved. These principles are essential not only for the development process but also for the long-term success of the platform. By defining specific goals, such as improving user experience or enhancing the accuracy of financial reporting, the team can focus on delivering a high-quality expense management application that meets user needs. Clear goals also allow for better collaboration, ensuring that each team member understands their role in achieving the project's objectives and contributing to its overall success. | An objective is a measurable milestone that needs to be accomplished to achieve a broader goal. In the context of ExpenseHub, objectives are the specific actions that must be taken to meet the overall goal of improving users' ability to manage and track their expenses effectively. For example, if the goal is to enhance users’ financial awareness, one of the objectives might be to increase the use of the expense report generation feature by 25%. Achieving this objective would indicate that users are engaging more with the application and gaining better insights into their spending patterns. |
| When setting goals and objectives for ExpenseHub or any expense management platform, it is essential to differentiate between outputs and outcomes. Outputs are the tangible deliverables that are produced as part of the project, while outcomes are the results or the impact of achieving those deliverables. In simple terms, outputs represent the tasks you need to complete to meet your goals, while outcomes demonstrate the value or changes these tasks bring to users.  For example, one output for ExpenseHub might be to develop and implement a new expense report feature. This output is a concrete deliverable that fulfills the goal of providing users with detailed insights into their spending. The outcome, on the other hand, is the impact this feature has—users gain a better understanding of their financial habits, leading to improved budgeting and spending control.  Another example could be implementing an automated expense categorization system. While the output is the successful development of this feature, the outcome is the increased efficiency and ease of use for users as they no longer need to manually categorize each expense, thus saving time and reducing errors. Understanding this distinction ensures that both the deliverables (outputs) and the results (outcomes) are aligned with the broader goals of the project. | Setting clear and measurable goals for ExpenseHub is essential for guiding the development process and ensuring that the project stays aligned with its vision. These goals also help the team remain focused, enabling easy tracking of progress and allowing for necessary adjustments along the way. By establishing specific, actionable objectives, the project can advance steadily toward its overall goal, without losing sight of the details that contribute to the larger picture. |
| **Output-based deliverables and products**  Outputs are tangible deliverables produced as a direct result of the project’s goals and objectives. In the context of ExpenseHub, outputs represent the concrete tasks and features that are developed to achieve the system's primary objectives. For example, an output-based deliverable might be the creation of the expense tracking interface, allowing users to log their daily expenditures easily. Another output could be the report generation feature, which enables users to generate detailed summaries of their expenses over a specific period.  These outputs are crucial because they are the building blocks needed to meet broader project objectives. For instance, implementing a new user-friendly dashboard or automated expense categorization system would be another output-based deliverable. These features, once completed, become functional products within the platform, directly contributing to the overall goals of ExpenseHub.  At the same time, while outputs are the actual products and services delivered, outcome-based results measure the impact of those outputs. For example, if the goal is to enhance financial awareness among users, the outcome of the new reporting feature would be an increase in user engagement with financial insights, ultimately leading to better financial decision-making. This distinction helps focus on not just building deliverables, but also ensuring they produce meaningful results for users. | There are many different ways of setting goals and objectives for ExpenseHub, but the most effective way to measure success is by using the SMART framework. SMART stands for Specific, Measurable, Achievable, Relevant, and Time-bound, and helps to ensure that each goal is clearly defined and actionable.  When setting goals for ExpenseHub, it’s crucial to be as specific as possible. For example, instead of a vague goal like "improve user experience," a more specific objective would be "reduce the time it takes for users to log their expenses by 15%." Clearly defining the outcome and the target makes it easier to track progress and understand what you’re trying to achieve.  To gauge success, it's important to establish clear metrics. For ExpenseHub, relevant metrics could include user engagement (e.g., the number of expenses logged per user each month) or feature usage (e.g., the percentage of users generating monthly expense reports). Additionally, tracking the average time users spend on logging expenses before and after making improvements can indicate whether the changes are effective. Tools like in-app analytics or Google Analytics can help monitor these metrics and offer data insights to measure progress.  By applying the SMART framework, ExpenseHub can set clear and actionable goals, define how success will be measured, and ensure that objectives are realistic, relevant to the platform's purpose, and achievable within a set timeframe. This structured approach allows for continuous progress and adaptation based on measurable outcomes. |

## 1.8 Documentation Conventions

To support users in navigating and utilizing the application effectively, "ExpenseHub" provides comprehensive user documentation, including step-by-step guides and FAQs.

# Chapter 2

# SOFTWARE REQUIREMENT SPECIFICATION

The analysis model is a crucial phase in the software development process, as it helps in understanding the system's requirements, interactions, and relationships between different components. In this chapter, we present various analysis diagrams that offer a comprehensive view of the "ExpenseHub" web app.

# 2.1 User Registration Requirements

* Build Build value around registration: Before users sign up for ExpenseHub, it is important to explain the benefits of creating an account. This could include access to personalized expense tracking, detailed reports, and the ability to analyze financial habits easily.
* Eliminate unnecessary fields: Only collect essential information such as the user's email and password during registration. Any additional details (like phone number or address) should be made optional to streamline the sign-up process and make it user-friendly.
* Group fields logically: For a smooth registration process, fields should be grouped logically. If additional information is required (e.g., preferences for financial tracking), these can be gathered after registration. The form should be broken down into steps, if needed, to avoid overwhelming the user with too many fields at once.
* Clearly state security and privacy policies: Users should feel confident that their financial data is secure when using ExpenseHub. Include a clear link to the platform’s privacy policy, outlining how their data will be protected.
* Easy password recovery: Users should be able to easily recover their password if forgotten. The registration and login forms should both include a clear "Forgot Password" option to help users reset their credentials.
* Captcha option for security: To prevent bots from creating fake accounts, include a captcha during the registration process, with an option for users to refresh the captcha if it becomes unreadable.
* Email verification: Before users can fully access their ExpenseHub accounts, require email verification to confirm the authenticity of their registration and enhance security.
* User Login System: To access their personalized expense dashboard, users must log in with their registered email and password. This login system ensures that only authenticated users can manage their financial data.
* Admin Panel: ExpenseHub will require an admin panel for system administrators to manage user data and ensure smooth operation of the platform. The admin panel should be secure, allowing only authorized personnel (e.g., two designated admins) to access it. Administrators should be able to manage user accounts, review user activity, and handle data related to the platform, ensuring security and user privacy.

# 2.2 External Interface Requirements

TThe success of ExpenseHub relies heavily on its user-friendly interface, designed to simplify the process of managing daily expenses. The system prioritizes ease of use, ensuring that users of all technical levels can navigate the platform effortlessly. The design is clean, intuitive, and focused on enhancing user experience through logical layouts and accessible features.

**Homepage:**

The homepage of ExpenseHub acts as the gateway for users to quickly understand the platform's purpose. It welcomes users with a simple and visually appealing layout, explaining the key features of the app such as expense tracking, report generation, and visual insights into spending patterns. Clear navigation options direct users to sign up, log in, or explore more about how the app works. The homepage is designed to make first-time users feel comfortable and encourage them to engage with the platform.

**Registration and Login Page:**

The registration and login process is straightforward, requiring only essential details such as email and password. For ease of use, the system keeps the required fields to a minimum. Returning users can log in securely, and if they forget their password, a quick recovery option is available. The login page features helpful, user-friendly error messages to assist users in troubleshooting issues such as incorrect password entry.

**Dashboard:**

The dashboard is the heart of ExpenseHub, where users can see a summary of their financial activities at a glance. It presents an overview of recent expenses, categorized by type (e.g., groceries, utilities, transport), and displays total spending, monthly comparisons, and any upcoming budget alerts. With an intuitive layout, the dashboard offers easy access to features like adding new expenses, viewing expense reports, or generating expense graphs. Users can customize the view to focus on specific categories or timeframes, making financial tracking more personalized and effective.

**Add Expense:**

The "Add Expense" interface is designed for simplicity and speed, allowing users to quickly input expenses with minimal clicks. Users can input details such as expense category, amount, date, and optional notes. Frequently used categories are suggested to help speed up the process. The interface is clear, ensuring users can easily log expenses without confusion.

**Expense Listings and Search:**

The expense listings section provides a comprehensive view of all expenses entered by the user. Each entry includes key details such as the amount spent, category, date, and description. Users can sort and filter expenses by different criteria such as date, amount, or category. The search function allows users to find specific expenses quickly, making it easy to review past spending and make necessary updates.

**Expense Report:**

The expense report interface offers a detailed breakdown of the user's financial data. Users can generate reports for specific timeframes (e.g., weekly, monthly, or custom ranges) and by categories (e.g., housing, groceries, entertainment). These reports can be exported in various formats (PDF, CSV), allowing users to analyze their spending trends. The design emphasizes simplicity, with clear labels and visual summaries that highlight spending habits and areas for improvement.

**Expense Graphs:**

The Expense Graphs feature provides a visual representation of the user’s spending patterns through bar graphs, pie charts, and line graphs. Users can select specific categories or timeframes to visualize, helping them gain insights into where their money is going. This graphical view of expenses makes it easier for users to identify trends, compare monthly or weekly spending, and adjust their financial habits accordingly.

**Profile Management:**

In the user profile section, users can manage their personal information, including updating their email address, password, and preferred currency. Users can also configure settings such as notification preferences for budgeting alerts or expense reminders. Additionally, the profile area allows users to manage saved payment methods and update billing addresses, ensuring that all personal and financial information is kept current.

**Budgeting Tools:**

The Budgeting Tools interface enables users to set financial goals and limits for specific categories. Users can set monthly budgets for items like groceries or entertainment, and ExpenseHub will track progress against these goals. The dashboard will alert users if they are nearing their spending limits, helping them stay on track financially.

**Notifications and Alerts:**

ExpenseHub includes a notifications system that informs users about important financial updates. For example, users can receive alerts if they are about to exceed their set budget or reminders to log expenses at regular intervals. These notifications can be managed in the profile settings, ensuring users stay informed without feeling overwhelmed by too many alerts.

**Admin Panel:**

The admin panel is designed for system administrators to manage user accounts and oversee the functionality of the platform. This secure interface allows authorized administrators to review user activity, manage user data, and handle platform operations such as maintenance, feature updates, and security checks. The admin panel also provides insights into user trends, helping the team improve the overall platform.

**Data Security and Privacy:**

Given the sensitive nature of personal financial data, ExpenseHub emphasizes robust security measures. Users can rest assured that their data is encrypted, and the platform includes a clear privacy policy accessible from both the registration page and the user profile. Passwords are securely stored, and two-factor authentication can be enabled to add an extra layer of security.

The user interfaces in ExpenseHub are designed to provide an intuitive and seamless experience, ensuring that both novice users and experienced budgeters can navigate the platform easily. The emphasis on clear, logical layouts, coupled with accessible features like expense tracking and report generation, helps foster user confidence and satisfaction. This ensures ExpenseHub is not only functional but also user-friendly, encouraging long-term engagement.

**Terms & Conditions**

* **Privacy Policy**: The privacy policy, outlining how ExpenseHub uses and protects user information. This policy details the platform's commitment to ensuring the privacy and security of user data.
* **Prohibitions:** Users must not misuse ExpenseHub. Prohibited activities include committing or encouraging any criminal offenses, transmitting or distributing malware, viruses, worms, or any harmful code. Breaching these terms will be considered a criminal offense and may result in legal action.
* **Intellectual Property, Software, and Content:** All intellectual property rights, including software, content, and graphical elements made available to users through ExpenseHub, remain the property of [ExpenseHub URL]. Unauthorized use or reproduction of these materials is strictly prohibited.
* **Terms of Service:** By registering on ExpenseHub, users agree to the terms and conditions outlined here. All features, services, and functionalities offered by the platform are subject to availability and confirmation, and ExpenseHub reserves the right to update or modify these terms as necessary.
* **Contract:** Once a user registers or submits financial data on ExpenseHub, they will receive an acknowledgment via email confirming receipt of the input.
* **Accuracy:** While ExpenseHub strives to ensure that all features, descriptions, and services offered are accurate, errors may occasionally occur.
* **Payment Authorization:** When using premium features, ExpenseHub will perform standard authorization checks on the user's payment methods to ensure sufficient funds are available to complete the transaction.
* **Complaints:** ExpenseHub has a complaints-handling procedure to resolve any issues or disputes that arise promptly. Users can submit complaints or comments about the system, and ExpenseHub will work to address these concerns.
* **Entire Agreement:** These Terms of Service constitute the entire agreement between the user and ExpenseHub and supersede any previous agreements. Any waiver of any provision of the Terms will only be valid if provided in writing and signed by an authorized representative of ExpenseHub.

**Hardware Interface Requirements for ExpenseHub:**

The ExpenseHub system operates on standard hardware components that ensure reliability and accessibility across various platforms. The hardware infrastructure is designed to support seamless expense management for users, whether accessing the system from desktop or mobile devices.

**User Devices:** ExpenseHub is accessible on a range of devices, including desktop computers, laptops, smartphones, and tablets. The platform employs a responsive design, ensuring that the user interface adapts smoothly to different screen sizes and resolutions, allowing users to manage their finances on any device conveniently.

**Servers:** ExpenseHub relies on high-performance servers to host the application and manage user data. These servers are optimized for efficiency, offering the necessary processing power, storage capacity, and memory to handle concurrent user requests without compromising speed. The server infrastructure ensures minimal downtime and quick response times, providing users with a smooth and reliable experience.

**Network Infrastructure:** A stable network infrastructure is essential to ensure uninterrupted access to ExpenseHub. The platform employs a robust network setup with redundancy and load balancing to minimize disruptions. Reliable network connectivity guarantees that users can consistently access and use the platform to manage their expenses without delay.

**Data Storage Solutions:** ExpenseHub uses secure and scalable data storage solutions, whether cloud-based or on-premises databases, to securely store user profiles, transaction records, and expense data. These solutions offer data redundancy, regular backups, and high availability to ensure that user data is protected against loss and can be accessed whenever needed. Data security is a top priority, ensuring compliance with industry-standard protocols to protect sensitive financial information.

The hardware infrastructure supporting ExpenseHub ensures optimal system performance and reliability, offering users a seamless experience across devices. By leveraging modern hardware and network technologies, ExpenseHub can efficiently manage high volumes of financial data and provide a secure, user-friendly platform for personal expense management.

**Specification**

**Analysis**

The objectives of the ExpenseHub project were carefully analyzed to ensure that the final product maximizes its usefulness for potential users. This analysis involved multiple rounds of internal and external feedback to align the platform’s features with user needs. The development team conducted thorough research on existing expense management applications and identified key trends, focusing on areas for improvement. This led to the discovery of several unique features and enhancements for ExpenseHub. The primary target audience includes individuals looking for a simplified yet powerful tool to manage their daily expenses, create reports, and track spending patterns. Following the research and analysis, a deeper understanding of the expense management field was established, guiding the development of a solution tailored to user needs.

**Design**

The design of ExpenseHub began from scratch, ensuring a smooth and user-friendly experience. Every page was carefully laid out to provide intuitive navigation and an efficient workflow. The design ensures that users can easily add, view, and categorize expenses with minimal effort. The dashboard was given particular focus to provide users with a comprehensive view of their financial status, allowing them to track monthly expenses, see spending trends, and manage budgets. Additionally, responsive design principles were employed to ensure ExpenseHub functions seamlessly across devices, including desktops, tablets, and smartphones. Visual cues like graphs and charts were incorporated to provide users with easily digestible insights into their spending habits.

**Verification Plan**

The verification process is a crucial part of the ExpenseHub development cycle, where newly built features are rigorously tested to ensure functionality aligns with expectations. Each feature, such as adding expenses, generating reports, and expense categorization, is tested by both developers and test users to identify bugs and ensure a smooth user experience. Feedback from early testers and peers is incorporated to improve functionality and user experience. The verification plan ensures that all components of ExpenseHub work correctly and that users can rely on the platform for accurate financial tracking and reporting.

**Software Interface Requirements**

ExpenseHub leverages modern software technologies to create a seamless and secure user experience, ensuring that all components work cohesively to support smooth expense management.

**Operating Systems:**

ExpenseHub is designed to run efficiently on various operating systems, including Windows, macOS, Linux, and mobile platforms like iOS and Android. This ensures flexibility, allowing users to manage their finances from any device.

**Web Browsers:**

The platform is optimized for all popular web browsers, including Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge. Ensuring cross-browser compatibility allows a wider audience to access ExpenseHub without facing technical limitations.

**Database Management System:**

ExpenseHub uses an SQL-based database to manage and store user profiles, expense records, and financial reports. The SQL database provides the necessary reliability and scalability for handling large volumes of data securely and efficiently.

**Server-Side Technologies:**

The server side of ExpenseHub is powered by Node.js and Express.js, offering a fast, event-driven architecture that ensures efficient handling of multiple user requests. This combination facilitates smooth communication between the client and server, enabling quick data retrieval and updates.

**Client-Side Technologies:**

ExpenseHub uses HTML, CSS, JavaScript, and jQuery on the frontend to create an interactive and responsive user interface. These technologies allow for dynamic content updates and intuitive navigation, providing a seamless experience for users managing their expenses.

**Payment Gateways and APIs:**

If ExpenseHub offers premium features, it can integrate with secure payment gateways like PayPal or Stripe. These services would handle payment processing for any in-app purchases, ensuring secure and reliable transactions.

**Communication Interface Requirements**

Effective communication interfaces are vital to ensure that ExpenseHub can provide timely updates and smooth interactions between the system and its users.

**HTTP/HTTPS:**

Communication between user devices and the ExpenseHub server is managed via HTTP/HTTPS protocols. HTTPS ensures that all data transmissions, especially sensitive information such as user login credentials and payment details, are encrypted and secure.

**Email Services:**

ExpenseHub incorporates email services to communicate with users effectively. Features like password recovery, expense reminders, and reporting notifications are sent to users’ registered email addresses, ensuring they stay informed about their financial activity and account status.

**Real-Time Notifications:**

ExpenseHub plans to offer real-time notifications through push notifications or email alerts, informing users of budget alerts, expense logs, or upcoming payments. These notifications keep users updated on their financial status without needing to log in frequently.

**Hardware Interface Requirements**

To support the performance and accessibility of ExpenseHub, the platform uses several essential hardware interfaces:

**User Devices:**

ExpenseHub is accessible across a variety of devices, including desktops, laptops, smartphones, and tablets. The platform's responsive design ensures a consistent user experience, whether users are managing their expenses on a large screen or a mobile device.

**Servers:**

The platform relies on robust servers that provide adequate processing power and storage to manage large amounts of expense data efficiently. The servers support multiple concurrent users, ensuring fast response times and minimal downtime.

**Network Infrastructure:**

A strong network infrastructure is vital to support the real-time functionality of ExpenseHub. Reliable network connections allow users to securely log in, update expenses, and retrieve reports without interruption. Redundant networks and load balancing ensure that the platform remains accessible, even under heavy usage.

**Data Storage Solutions:**

ExpenseHub employs secure cloud-based storage solutions or on-premises databases to store user data and expense records. The data storage solutions are designed to offer scalability and redundancy, protecting against data loss and ensuring availability in case of unexpected failures.

# 2.3 Other Non-Functional Requirements

NonNon-functional requirements are crucial to the overall success of ExpenseHub, as they define system properties such as performance, security, and scalability. These aspects ensure that the system operates efficiently, securely, and reliably while offering a smooth user experience. Below are the key non-functional requirements for ExpenseHub.

**2.3.1 Performance Requirements**

Performance is critical to ensuring that ExpenseHub provides a seamless experience for users managing their expenses. These requirements focus on response times, throughput, and scalability.

**Response Times:**

ExpenseHub is designed to offer fast response times for all user interactions, including logging expenses, generating reports, and viewing expense summaries. Loading times for the dashboard, expense lists, and graphs should be minimal, ensuring users can quickly access their data without unnecessary delays. The goal is to keep response times under 2 seconds for most interactions to maintain user engagement and satisfaction.

**Throughput:**

The platform is built to handle multiple users accessing the system concurrently, especially during peak usage times, such as the end of the month when many users might review their expenses. ExpenseHub must be able to process numerous expense entries and generate reports simultaneously without any degradation in performance.

**Scalability:**

As the user base grows, ExpenseHub must scale effectively to accommodate increased demand. The system should be capable of handling large amounts of data and increased transaction volumes without performance bottlenecks. Elastic computing resources and horizontal scaling strategies will be employed to ensure the platform remains efficient as it grows.

**Safety Requirements:**

Safety requirements are essential to protect user data and maintain the security and integrity of the ExpenseHub platform. These include data security, payment security, and user authentication.

**Data Security:**

ExpenseHub employs strong encryption techniques to protect sensitive user data, including financial information and personal details. All data transmissions between the user and the platform are secured using SSL (Secure Socket Layer) and TLS (Transport Layer Security) protocols to prevent unauthorized access and ensure the confidentiality of user data.

**User Authentication:**

The platform uses secure authentication mechanisms to protect user accounts. Passwords are hashed and salted before being stored in the database to protect against breaches. Additionally, multi-factor authentication (MFA) can be implemented to add an extra layer of security, reducing the likelihood of unauthorized access.

**Data Privacy:**

ExpenseHub complies with data protection regulations, ensuring that user data is collected, stored, and processed securely. Privacy policies are made transparent to users, detailing how their information is handled and protected.

**Reliability Requirements**

Reliability is crucial for the trustworthiness of ExpenseHub. The platform must remain available to users, providing consistent access to their financial data and functionalities.

**Uptime:**

ExpenseHub should have a high level of availability, aiming for 99.9% uptime, ensuring users can access the platform at all times. Regular backups and redundancy measures will be implemented to minimize downtime and ensure data is recoverable in case of system failures.

**Error Handling:**

The system must include robust error-handling mechanisms to ensure that any disruptions (e.g., failed expense entries or incomplete report generation) are properly logged, and users are notified with clear error messages. These mechanisms should ensure that errors do not compromise data integrity or user experience.

**Maintainability Requirements**

The platform must be easy to maintain and update as new features are added or bugs are fixed.

**Modular Design:**

ExpenseHub is designed with a modular architecture, allowing for easy updates, enhancements, and maintenance without affecting the entire system. New features can be integrated seamlessly, and any part of the system can be updated independently of others.

**Documentation:**

Comprehensive documentation is provided for developers to ensure smooth transitions between team members and to facilitate any future updates or fixes.

**Usability Requirements**

User experience is a priority for ExpenseHub, and the platform is designed to be accessible to a wide range of users, from tech-savvy individuals to those with limited technical knowledge.

**Intuitive Interface:**

The user interface is simple and intuitive, ensuring users can log expenses, generate reports, and track their financial data with ease. Clear visual cues, clean navigation, and minimalistic design contribute to a smooth user experience.

**Mobile Accessibility:**

The platform is fully responsive and optimized for mobile devices, ensuring that users can access their expense data from smartphones and tablets just as easily as they can on desktop computers.

**Security Requirements**

Security is a top priority in ExpenseHub, given its role in managing sensitive user financial data. The platform incorporates a comprehensive set of security measures to protect user information, ensure secure financial tracking, and maintain the integrity of the system. These security features focus on authentication, data protection, and safeguarding system integrity, ensuring users can manage their expenses with confidence.

**User Authentication and Authorization:**

ExpenseHub implements secure user authentication mechanisms to verify user identities during login and registration. Secure password storage practices, such as salting and hashing, protect user credentials from unauthorized access. The system uses role-based access control to ensure users have appropriate permissions based on their roles, preventing unauthorized actions and restricting access to sensitive areas of the platform.

**Data Encryption and Privacy:**

Sensitive information, such as user profiles, expense data, and payment details (for premium features), is encrypted using strong encryption algorithms. ExpenseHub complies with data privacy regulations, ensuring user data is protected from unauthorized access. Users are given control over their personal data, with options to manage privacy settings within the platform.

**Protection against Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF):**

The platform includes robust input validation and sanitization measures to prevent XSS attacks that could inject malicious scripts into the application. Additionally, CSRF tokens are used to safeguard against cross-site request forgery, ensuring that requests are validated and only legitimate actions are processed.

**Secure Session Management:**

ExpenseHub uses secure session management techniques, such as HTTP-only and secure cookies, to protect user sessions from hijacking. Session timeouts automatically log out inactive users, enhancing security and preventing unauthorized access through abandoned sessions.

**Preventing SQL Injection:**

The application uses parameterized queries and prepared statements to guard against SQL injection attacks, ensuring that malicious SQL code cannot be executed on the database. This approach helps to maintain data integrity and security, especially when handling user-generated inputs.

**Regular Security Audits and Penetration Testing:**

To identify potential vulnerabilities, ExpenseHub undergoes regular security audits and penetration testing. This process ensures that any weaknesses in the system are promptly identified and resolved, maintaining a secure environment for all users.

**Error Handling and Logging:**

ExpenseHub implements secure and detailed error handling mechanisms. Errors are logged in a way that prevents exposure of sensitive information. These logs help identify security issues and assist in debugging while ensuring that user data is not compromised.

**Secure File Uploads:**

If ExpenseHub offers a feature for file uploads (such as for receipts or financial documents), strict validation and restrictions are applied to prevent the upload of malicious files. Uploaded files are stored in secure locations, and access controls are enforced to ensure only authorized users can access them.

**Security Patches and Updates:**

The ExpenseHub development team consistently applies security patches and updates for all components, including frameworks and libraries. This proactive approach ensures that the platform remains protected against newly discovered vulnerabilities.

**Compliance with Industry Standards:**

ExpenseHub adheres to industry best practices and security standards, such as General Data Protection Regulation (GDPR) for data privacy, ensuring the secure management of user data and financial records. If handling payments, the platform would also comply with Payment Card Industry Data Security Standard (PCI DSS) to safeguard payment information.

By incorporating these security requirements, ExpenseHub ensures a safe and reliable platform for users to manage their financial data. These comprehensive measures demonstrate ExpenseHub's commitment to user privacy, data security, and system integrity, fostering trust and confidence among its users.

### **2.4 Architecture Diagram**

The technologies used for building ExpenseHub include HTML, CSS, JavaScript, jQuery, and SQL. This combination of technologies forms the core of the ExpenseHub architecture, ensuring a seamless user experience for tracking and managing expenses.

The architectural workflow of ExpenseHub is structured as follows:

**Client-Side (Frontend):**

The user interacts with the HTML, CSS, JavaScript, and jQuery components on the frontend. These technologies handle the presentation and user interaction aspects, ensuring a dynamic and responsive user interface where users can log, update, and view their expenses.

**Server-Side (Backend):**

The server-side logic is managed through Node.js (or similar backend technologies), which processes the user's requests. When a user interacts with the platform, such as submitting an expense or requesting a report, the request is sent to the server.

**SQL Database:**

ExpenseHub uses an SQL database to store and manage all expense-related data, user profiles, and financial records. The server receives the user's request and interacts with the SQL database to either retrieve or store data. For example, when a user adds a new expense, the data is securely inserted into the SQL database.

**Response Handling:**

Once the SQL database has processed the data, it returns the required information (such as the updated expense list or report) to the backend. The server then sends this data back to the JavaScript components on the frontend.

**Client-Side Display:** The frontend, utilizing JavaScript and jQuery, dynamically updates the user interface to reflect the new data, such as displaying updated expenses or generating visual reports (like graphs or charts) based on user input.

The underlying architecture of ExpenseHub follows a client-server model, with clear separation between the frontend (client-side), backend (server-side), and database. The system does not directly use MVC (Model-View-Controller) but implements similar concepts by separating concerns: HTML/CSS/JS handle the view, SQL manages the data model, and the server functions as the controller, orchestrating the communication between the user interface and the database.

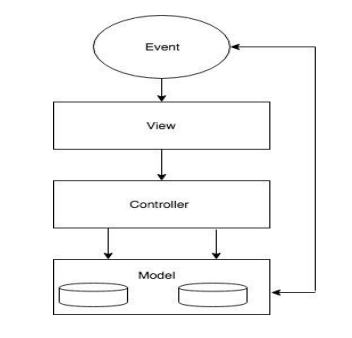


Figure 2.1 MVC Diagram

• **Model**: Model manages application data and it responds to the requests made from views, also listens to the instructions from controller.

• **View**: View is the portion which we see in the web application. View is responsible to display the data to user.

• **Controller**: Controller is very important because it controls the interactions between Model and Views. Controller responds to user input and performs interactions on the models. This also validates input.

### **2.5 Entity-Relationship Diagram (ERD)**

The Entity-Relationship Diagram (ERD) for ExpenseHub visually represents the data model and relationships between various entities within the system. This diagram plays a crucial role in outlining how data is structured and connected in ExpenseHub to support its expense management features.

**Key Entities in the ERD include:**

**User:**

**Attributes:** User ID, Name, Email, Password, Profile Picture, and Account Type (Standard or Admin).

**Relationships:**

One-to-Many relationship with Expense (A user can log multiple expenses).

One-to-Many relationship with Category (A user can create multiple categories).

**Expense:**

**Attributes:** Expense ID, Amount, Date, Description, Category ID, User ID.

**Relationships:**

Many-to-One relationship with User (Multiple expenses can be logged by one user).

Many-to-One relationship with Category (Each expense is categorized).

**Category:**

**Attributes**: Category ID, Name, User ID.

**Relationships:**

One-to-Many relationship with Expense (A category can have multiple expenses).

Many-to-One relationship with User (Each category is assigned to a specific user).

**Report:**

**Attributes**: Report ID, Date Generated, User ID, Report Data (Total Expenses, Monthly Breakdown, Graphical Data).

**Relationships:**

Many-to-One relationship with User (A user can generate multiple reports).

**Relationship Overview:**

**User**-**Expense**: A one-to-many relationship, as a single user can have multiple expenses logged over time.

**Expense-Category:** A many-to-one relationship, as each expense belongs to a specific category, and each category can group multiple expenses (e.g., groceries, utilities).

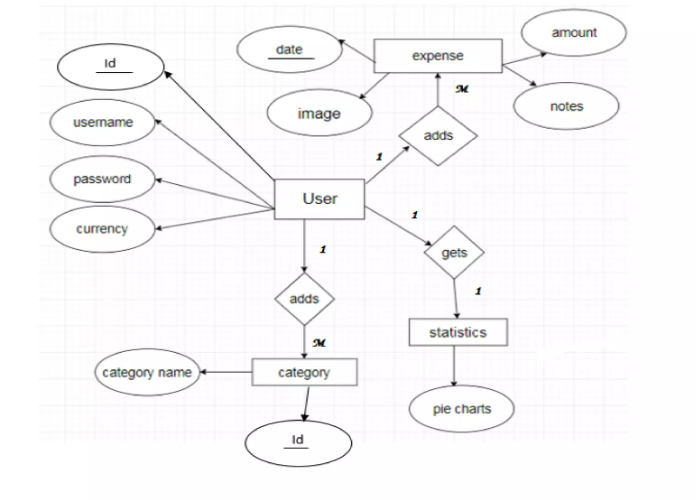
**User-Category:** A one-to-many relationship, as each user can create their own set of categories to organize their expenses.

**User-Report:** A one-to-many relationship, as a user can generate multiple reports, summarizing their expenses over time.

**ERD Summary:**

In ExpenseHub, the ERD defines how core entities like User, Expense, Category, and Report are connected, ensuring that users can efficiently log, categorize, and analyze their expenses. The diagram highlights the relational structure, ensuring that expenses can be organized and tracked based on categories, while reports can be generated to provide insights into financial habits. This data model plays a critical role in managing user data, supporting features like budget tracking and expense reports.

By establishing these clear relationships between entities, ExpenseHub provides a structured and scalable system for personal finance management, making it easy for users to organize their financial data and draw meaningful insights.



**Figure 2.2 ER-Diagram**

### **2.6 Data Flow Diagram (DFD)**

A Data Flow Diagram (DFD) is a graphical representation of how data flows through a system. For ExpenseHub, the DFD illustrates the movement of data between users, processes, and data stores, providing a clear overview of how the platform manages expense tracking and related activities.

**Level 0 DFD:**

The Level 0 DFD provides an overview of the main processes and data stores in ExpenseHub, along with the interaction between external entities and the system.

**1. External Entities:**

**User:** Represents the end-users of ExpenseHub who log expenses, view reports, and track spending.

**Administrator:** Represents the admin users who manage the system, monitor usage, and handle any administrative functions, such as user account management.

**2. Processes:**

**Log Expense:** Process where users input new expense data, such as amount, category, and description.

**View Expenses:** Process where users can retrieve and view their recorded expenses.

**Generate Report:** Process that allows users to generate a financial report based on their logged expenses.

**Manage Categories:** Process where users can add, update, or delete expense categories for better organization.

**Admin Operations:** Process where administrators can manage user data and system performance.

**3. Data Stores:**

**User Database:** Stores user data, including login credentials, profile details, and preferences.

**Expense Database:** Stores all expense records, including the amount, category, date, and descriptions logged by users.

**Report Database:** Stores generated reports, including monthly summaries and breakdowns by category.

**4. Data Flow:**

**Log Expense:** The data flow from the "Log Expense" process to the "Expense Database" shows that the system stores expense details entered by users.

**View Expenses:** The data flow from the "View Expenses" process retrieves expense data from the "Expense Database" to display it to the user.

**Generate Report:** The data flow from the "Generate Report" process to the "Report Database" shows how report data is generated and stored for future reference.

**Level 1 DFD:**

The Level 1 DFD breaks down the main processes from the Level 0 DFD into sub-processes, providing more detail on the specific operations within ExpenseHub.

**1. Log Expense:**

**Authenticate User:** Sub-process that verifies the user's credentials before allowing them to log an expense.

**Add Expense:** Sub-process where users enter expense details (amount, category, date, and description).

**Validate Input:** Sub-process that checks whether the input is valid and correctly formatted.

**Store Expense:** Sub-process that saves the expense record to the Expense Database.

**2. View Expenses:**

**Retrieve Expenses:** Sub-process that fetches all expenses from the Expense Database for the logged-in user.

**Filter Expenses:** Sub-process where users can filter expenses by date, category, or amount.

**Display Expenses:** Sub-process that presents the filtered expenses to the user.

**3. Generate Report:**

Select Time Period: Sub-process where users select the time period for the report (e.g., monthly or custom range).

**Calculate Totals:** Sub-process that calculates total expenses and categorizes them for the report.

**Create Graphs:** Sub-process that generates visual graphs based on the expense data.

**Store Report:** Sub-process that saves the generated report in the Report Database.

**4. Manage Categories:**

**Add Category:** Sub-process that allows users to create new categories for organizing expenses.

**Update Category:** Sub-process where users can edit the name or details of existing categories.

**Delete Category:** Sub-process that enables users to remove categories, ensuring that any associated expenses are updated accordingly.

**5. Admin Operations:**

View User Data: Sub-process where administrators can retrieve and monitor user activity and expense data.

Manage User Accounts: Sub-process that allows administrators to update or deactivate user accounts.

System Maintenance: Sub-process for performing system maintenance tasks, ensuring the smooth operation of ExpenseHub.

**Data Flow Breakdown:**

**Log Expense:** Users enter expense data, which is stored in the Expense Database after input validation.

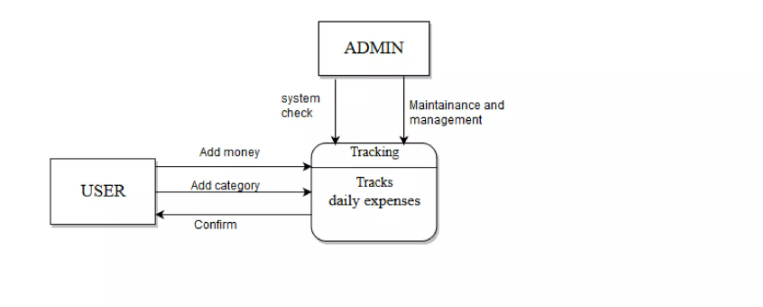
**View Expenses:** Users can retrieve their expenses from the Expense Database for review and filtering.

**Generate Report:** The system compiles expense data into reports, stores the report in the Report Database, and provides visualizations for easy understanding.

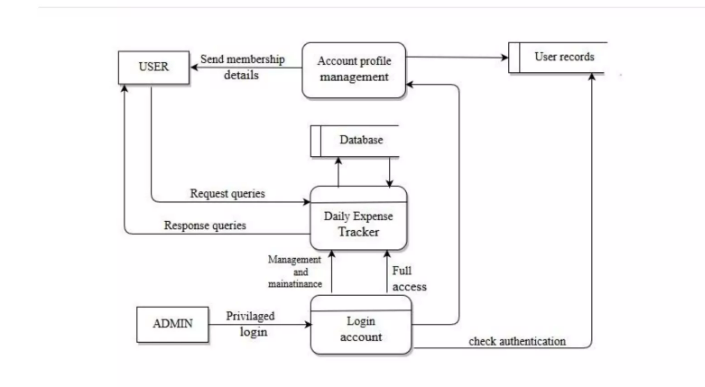
**Manage Categories:** Users can customize their expense categories, with all changes reflected in the Expense Database.

**Admin Operations:** Administrators manage the system and user accounts, ensuring smooth operation and user satisfaction.

.



**Figure 2.3 DFD (L1)**



**Figure 2.4 DFD (L2)**

**2.7 Class Diagram**:

The Class Diagram provides a static view of the ExpenseHub system, illustrating the main classes, attributes, and relationships between objects. This diagram helps define how data and functions are structured in the application.

The main classes in the ExpenseHub system include:

**User:**

Attributes: User ID, Name, Email, Password, Profile Picture, Account Type (Standard or Admin).

Methods: Register, Login, Update Profile, Delete Account.

Relationships: Each user can have multiple expense entries and categories, and can generate reports.

**Expense:**

Attributes: Expense ID, Amount, Date, Description, Category ID, User ID.

Methods: Add Expense, Edit Expense, Delete Expense, View Expense.

Relationships: Each expense belongs to a specific category and user.

**Category:**

Attributes: Category ID, Name, User ID.

Methods: Add Category, Update Category, Delete Category.

Relationships: A user can create multiple categories, and each expense is assigned to one category.

**Report:**

Attributes: Report ID, Date Generated, Total Expenses, Monthly Breakdown, Graphical Data.

Methods: Generate Report, View Report, Export Report.

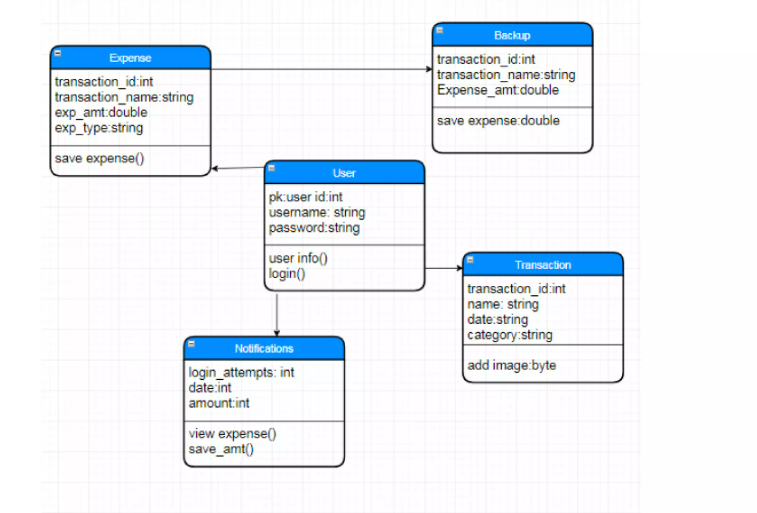
Relationships: Each user can generate multiple reports based on their logged expenses.

**Administrator:**

Attributes: Admin ID, Name, Email, Password.

Methods: Manage User Accounts, View User Data, Perform System Maintenance.

Relationships: The administrator manages user data and oversees the system’s performance.



**Figure 2.5 Class Diagram**

The relationships between classes are represented by associations, such as the relationship between User and Expense (one-to-many), User and Report(one-to-one), and Expense and Category (many-to-one).

### **2.8 Object Diagram**

The Object Diagram provides a snapshot of the system at a specific moment in time, showing objects and their relationships as instances of the classes defined in the Class Diagram.

Example Object Diagram:

**User (object):**

Name: Jane Smith

Email: jane.smith@example.com

**Expense (object):**

Expense ID: 789

Amount: $45

Date: 2024-10-15

Description: Grocery shopping

Category: Groceries

**Category (object):**

Category Name: Groceries

User: Jane Smith

**Report (object):**

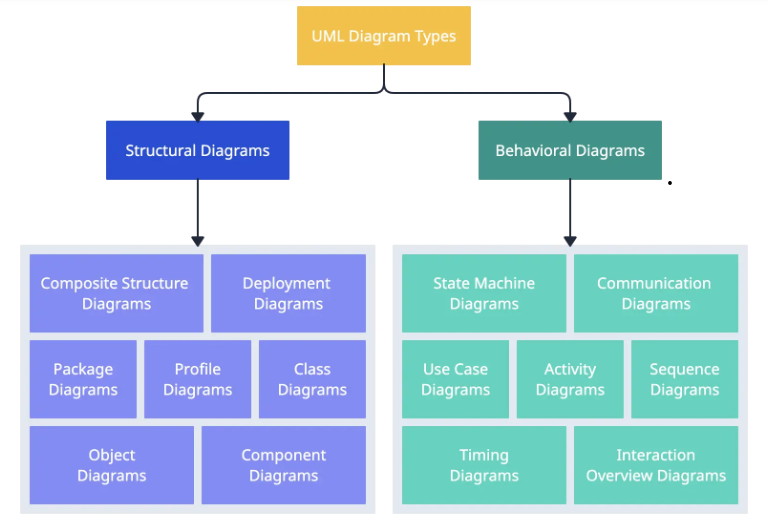
Report ID: 5678

User: Jane Smith

Date Generated: 2024-10-15

Total Expenses: $600

Breakdown: Groceries - $300, Utilities - $150, Entertainment - $150



**Figure 2.6 Object Diagram**

### **2.9 Sequence Diagram**

The Sequence Diagram illustrates the interactions between objects and components over time, showcasing the flow of messages and actions during specific use cases.

**Example Sequence Diagram (Logging an Expense and Generating a Report):**

User logs into the system.

User -> System: Provide login credentials (email, password).

System -> User: Validate login and provide confirmation of successful login.

**User logs a new expense.**

User -> System: Enter details for the expense (amount, category, date, description).

System -> Expense Database: Save the new expense to the database.

Expense Database -> System: Confirm that the expense has been saved.

System -> User: Notify user that the expense was successfully logged.

**User requests to generate a report.**

User -> System: Request to generate a report for the current month.

System -> Expense Database: Fetch all expense records for the selected time period from the database.

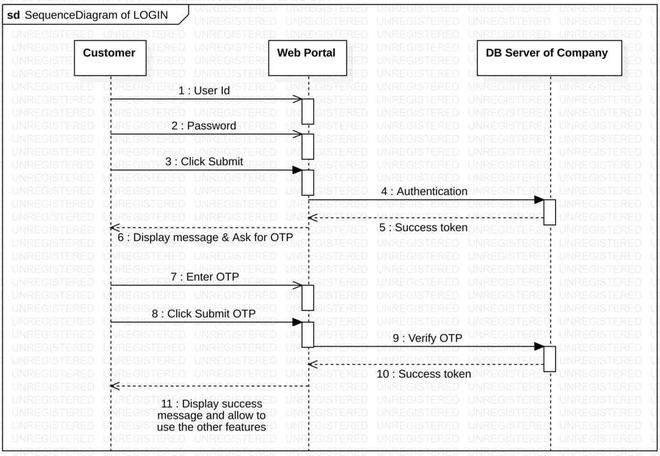
Expense Database -> System: Return the expense data.

System: Calculate total expenses and prepare report.

System -> Report Database: Save the generated report for future reference.

Report Database -> System: Confirm the report has been saved.

System -> User: Display the report to the user, including graphs and totals, and offer export options.

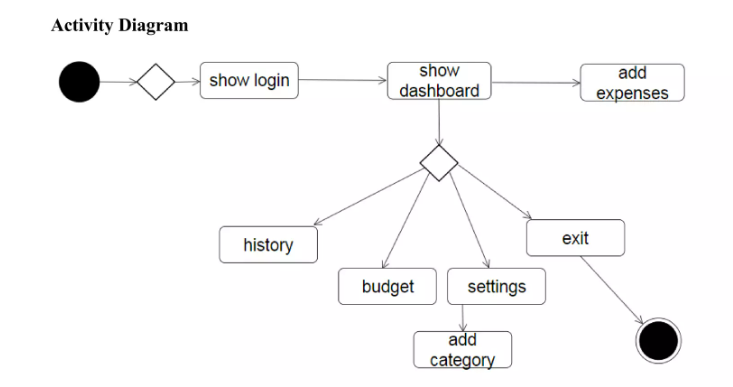


**Figure 2.7 Sequence Diagram**

### **2.10 Activity Diagram**

**User starts the process to log an expense.**

* The system prompts the user to enter expense details (amount, category, date, description).
* User provides the required details.
* The system verifies the input and saves the expense to the database.
* The logging process is complete.

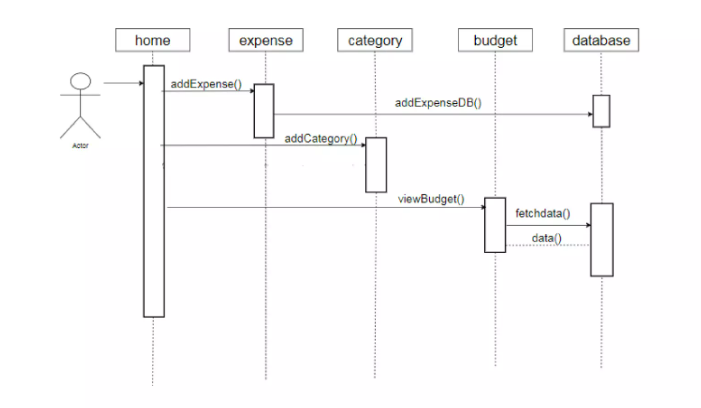


**Figure 2.8 Activity Diagram**

### **2.11 Collaboration Diagram**

**Example Collaboration Diagram (Log Expense):**

* User enters expense details and submits them to the system.
* The system receives the expense data and sends it to the Expense Database.
* The Expense Database stores the new expense record and confirms the update.
* The system sends a confirmation message back to the user that the expense has been logged.
* The user’s expense list is updated to reflect the new entry.



**Figure 2.9 Collaboration Diagram**

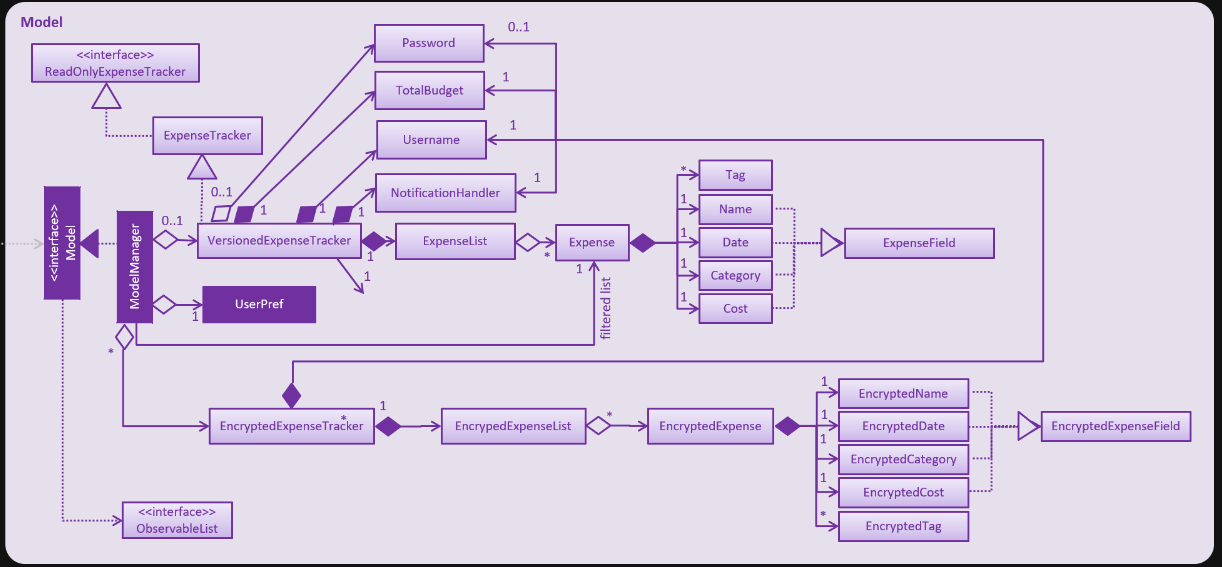
### **2.12 State Transition Diagram**

**Example State Transition Diagram (Expense Status):**

**Initial State:** Expense Created

**Events:** Expense Logged, Expense Edited, Expense Deleted, Expense Report Generated

The system transitions the expense record through different states based on user actions, such as logging, editing, deleting an expense, or generating a report that includes the expense data.



**Figure 2.10 State Transition Diagram**

# Chapter 3

# ANALYSIS (USE CASE MODEL)

### **3.1 Identifying Actors and Use Cases using Textual Analysis**

In this phase of the analysis, we identify the key actors and use cases for the ExpenseHub application through textual analysis. Actors represent the different roles or users that interact with the system, while use cases represent the various functionalities or actions that the system can perform.

**Key Actors:**

**User**: A registered user who logs into ExpenseHub using their credentials. They have access to core functionalities such as logging expenses, viewing and generating reports, and managing their categories.

**Administrator**: A privileged user responsible for managing the ExpenseHub platform. They have access to administrative functionalities such as managing user accounts and maintaining system operations.

**Use Cases:**

**Log Expense:** Users can log new expenses by entering details such as amount, category, date, and description.

**View Expenses:** Users can view their list of expenses, filtered by date, category, or other criteria, allowing them to track their spending.

**Search Expenses:** Users can search for specific expenses using keywords, date ranges, or categories to quickly find specific entries.

**Generate Report:** Users can generate detailed reports based on their expenses, including total spending and breakdowns by category.

**View Report:** After generating a report, users can view a summary of their expenses and visual representations like charts or graphs.

**Manage Categories:** Users can create, update, or delete expense categories to organize their expenses more effectively.

**Register Account:** A new user can create an account by providing required details, such as name, email, and password.

**Login:** Users can log into their accounts using their registered email and password, granting them access to their personal expense tracking and management features.

**Update Profile:** Users can update their personal details, including email, password, and account settings.

**Manage Users (Administrator):** The administrator can manage user accounts, including viewing user activity, updating user information, or deactivating accounts.

**System Maintenance (Administrator):** The administrator can perform system maintenance tasks, ensuring smooth operation of ExpenseHub by managing backups, updates, and other administrative operations.

### **3.2 Forming Use Case Diagram with Candidate and Use Cases**

In this section, we provide a comprehensive overview of the functional modules and requirements of the "ExpenseHub" system.

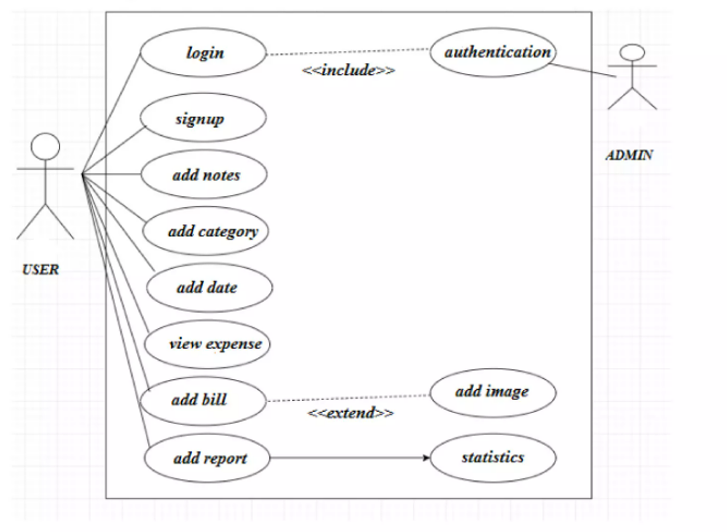


Figure 3.1 ERD Site Flow

### **3.3 Describe the Events Flow for Use Case**

**Browse Products:**

**Log Expense:**

1. The user selects the "Log Expense" option from the main menu.
2. The system presents a form to enter expense details (amount, category, date, description).
3. The user fills in the required details and submits the form.
4. The system validates the input and saves the expense in the database.
5. The use case ends.

**View Expenses:**

1. The user selects the "View Expenses" option from the main menu.
2. The system retrieves the user's expenses from the database.
3. The system displays the list of expenses, allowing the user to filter or sort them.
4. The use case ends.

**Generate Report:**

1. The user selects the "Generate Report" option from the main menu.
2. The system prompts the user to select a date range or category for the report.
3. The user confirms the selection and submits the request.
4. The system retrieves the relevant expenses and generates a report, including totals and visual graphs.
5. The system displays the report to the user.
6. The use case ends.

**Register Account:**

1. The guest selects the "Register" option from the main menu.
2. The system presents a registration form.
3. The guest fills in the required details (name, email, password) and submits the form.
4. The system validates the input and creates a new user account.
5. The use case ends.

**Login:**

1. The user selects the "Login" option from the main menu.
2. The system presents a login form.
3. The user enters their registered email and password and submits the form.
4. The system verifies the credentials and grants access to the user's account.
5. The use case ends.

**Update Profile:**

1. The user selects the "Profile" option from the main menu.
2. The system displays the user's profile information.
3. The user updates their personal information, such as email or password.
4. The user saves the changes.
5. The system updates the profile in the database.
6. The use case ends.

**Manage Users (Administrator):**

1. The administrator selects the "Manage Users" option from the admin panel.
2. The system retrieves a list of all users from the database.
3. The administrator can view user activity, update user details, or deactivate accounts.
4. The system saves any changes made by the administrator.
5. The use case ends.

By outlining the event flows for the identified use cases in ExpenseHub, we gain a comprehensive understanding of how users and administrators interact with the system, ensuring smooth functionality and an intuitive experience.

# Chapter 4

# DESIGN

# 4.1 Overall Description

ExpenseHub is a web-based expense management system developed using HTML, CSS, JavaScript, and SQL. It aims to provide users with a simple yet powerful platform for tracking daily expenses and generating financial reports. The project is designed to offer hands-on experience in building a functional, data-driven web application, allowing users to manage personal finances effectively. As digital tools for personal finance management continue to grow, ExpenseHub contributes to this evolving domain by offering a user-friendly solution for budgeting and expense tracking.

### **4.1.1 Product Perspective**

ExpenseHub operates in the context of personal finance management, allowing users to log, categorize, and track their daily expenses. Users can create an account, log their expenses, generate reports, and manage expense categories. The administrator manages system operations, ensuring that users have a smooth experience.

**4.1.2 Product Features**

**Log Expenses:** Users can log expenses by entering details such as amount, category, and date.

**View and Search Expenses:** Users can view their expense history and search for specific entries.

**Generate Reports:** Users can generate reports with visual charts to analyze their spending patterns.

**Manage Categories:** Users can create, update, and delete categories to better organize their expenses.

### **4.1.4 Assumption and Dependencies**

The successful functioning of the ExpenseHub application relies on the following dependencies:

**Technology Stack:** HTML, CSS, JavaScript, jQuery, and SQL form the core of the application's development, ensuring a responsive and functional user experience.

**Internet Infrastructure:** A stable and reliable internet connection is essential for smooth expense logging, data retrieval, and report generation.

**User Engagement:** The application's effectiveness depends on attracting a consistent user base that regularly uses the platform for managing expenses.

**Security Protocols:** Robust security measures, such as encryption and secure data storage, are critical to protecting user data and maintaining trust in the platform.

## 4.2 System Features

### **4.2.1 Log Expense Page**

Welcome to the Log Expense page! Here, you can easily log your daily expenses. Simply enter the amount, category, date, and description of your expense in the provided fields. Once the information is entered, click Submit to add the expense to your account. Your logged expenses will be stored in the system and can be viewed at any time for tracking and analysis. This feature helps you stay organized and maintain control over your finances.

### **4.2.2 Contact Us Page**

Both registered users and visitors can contact the ExpenseHub team through the Contact Us page. Users simply need to fill out the form with their name, email, subject, and the message they want to send. Upon submitting the form, the system delivers the message directly to the administrators, who will respond promptly to user inquiries..

### **4.2.3 About Us Page**

The About Us page provides important information about the ExpenseHub platform. It explains how ExpenseHub works, its mission, and the benefits of using the system for managing daily expenses. The page also includes details about the team behind the platform and contact information. This page serves as a valuable resource for users looking to understand the platform's features and functionalities.

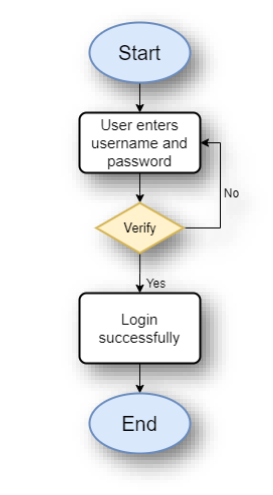
### **4.2.4 Sign up**

The Sign Up page makes account creation easy. Users need to fill out their name, email address, and create a secure password. By signing up, users gain access to personalized expense tracking, report generation, and category management. Once registered, users can log in anytime to track their financial activities and take advantage of the full range of features that ExpenseHub offers.

### **4.2.5 Login**

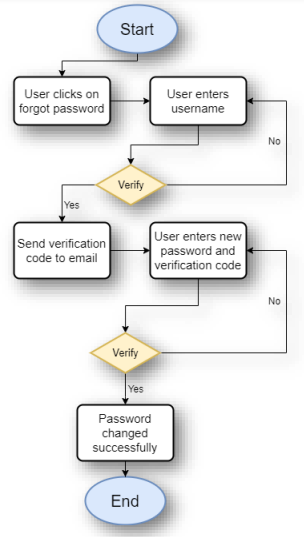
Once registered, users can log in to ExpenseHub using their email and password. The login process serves as a secure authentication step, ensuring only authorized users can access their account. Upon successful login, users can manage their expenses, view reports, and organize their spending categories. The system's secure login mechanism ensures that personal and financial data remains protected from unauthorized access, offering a safe and reliable platform for expense management.

The flow of a user logging in. It was a simple login flow that followed the general step. User could only log in with username and password but not email. Methods were called to verify the user’s input and display whether or not he or she login successfully.

****

**Figure 4.2 Login Flow Chart**

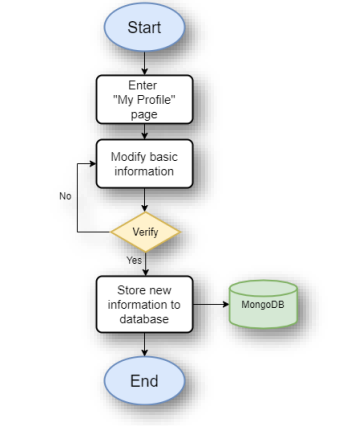
### **4.2.8 Reset Password Flow Chart**

****

**Figure 4.3 Reset Password Flow Chart**

The picture above showed the steps of resetting the password of a user. The user was required to enter his or her username so that a function would be called to check if the username was existed or not.

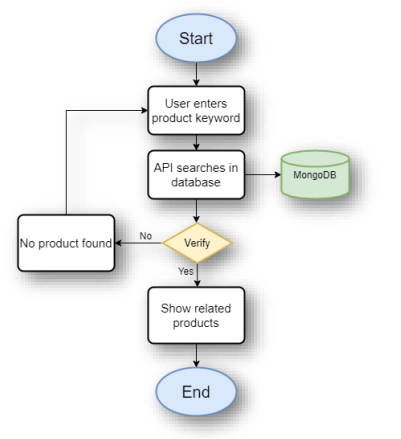
### **4.2.6 Edit Profile Flow Chart**

****

**Figure 4.4 Edit Profile Flow Chart**

The Figure no. above showed the steps of editing the profile information of a user. After the user clicked into the “My Profile” page, he or she could modify some basic information such as profile picture and display name of his or her account. After verification, newly modified data would be stored in the online database.

### **4.2.7 Search Expense Flow Chart**

****

**Figure 4.5 Search Expense Flow Chart**

The picture above depicted the flow of searching for a product on this website. When the user enters a searching term in the search bar provided, an API method would be called to perform searching in the online database for matching product names.

### **4.2.8 Logout**

After managing their expenses or generating reports, users have the option to log out of the ExpenseHub platform. Logging out ensures the security of their financial data and personal information by preventing unauthorized access to their account. This feature promotes safe usage of the platform, encouraging users to log out after each session, particularly when using shared or public devices. By logging out, users maintain full control over their account and protect their sensitive financial details.

### **4.2.9 Report Generation**

In ExpenseHub, users can generate detailed expense reports based on their logged expenses over a selected time period. These reports provide a summary of spending across different categories and timeframes, helping users analyze their financial habits. After generating the report, users can view a breakdown of their expenses through visual charts and graphs, simplifying the process of tracking spending patterns. A copy of the report can be exported for personal records or future reference, and the system stores reports within the database for easy access at any time. This feature enhances financial awareness and supports better budgeting and decision-making.

### **4.2.10 Dashboard**

The ExpenseHub dashboard provides a centralized view for both users and administrators to manage their data efficiently. For regular users, the dashboard offers an overview of logged expenses, spending summaries, and category breakdowns through easy-to-read visual charts. Users can also access options to log new expenses, generate reports, and manage categories directly from the dashboard.

### **4.2.11 Add Expense**

On the Add Expense page, users can easily log new expenses by entering details such as amount, category, date, and an optional description. Once submitted, the expense is saved to the system and available for future reference. This feature helps users maintain an organized and up-to-date record of their financial transactions.

### **4.2.12 Search Expense**

The Search Expense feature allows users to quickly find specific expenses by entering keywords, selecting date ranges, or filtering by category. This feature is especially useful for reviewing past expenses or finding entries for detailed analysis. Once the search is completed, relevant results are displayed for easy access.

### **4.2.13 View Expense Details**

The View Expense Details feature provides users with a detailed breakdown of any selected expense. Users can view the amount, category, date, and description of each transaction. This helps users better understand and manage their expenses, enabling accurate tracking of their financial records.

### **4.2.14 Update/Delete Expense**

Users can modify or remove their expenses through the Update/Delete Expense feature. By selecting an expense, users can update any details such as the amount, category, or description. Alternatively, if an entry is no longer needed, users can delete the expense from their records. This feature ensures flexibility and control over the user’s financial data.

### **4.2.15 Expense Graphs**

The Expense Graphs feature visualizes user expenses in the form of bar charts, pie charts, and line graphs. These visualizations help users easily identify spending trends and track their progress over time. Graphs can be generated based on categories, date ranges, or total spending, offering an intuitive way to analyze financial.

# Chapter 5

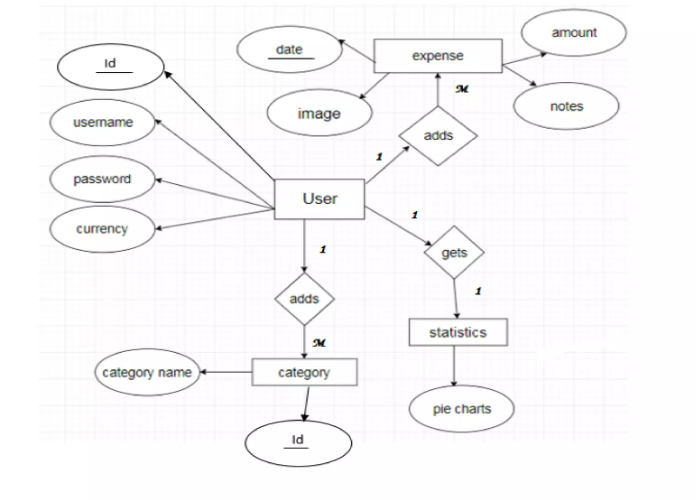
# IMPLEMENTATION

The ExpenseHub application is implemented using a stack that includes HTML, CSS, JavaScript, jQuery, and SQL. These technologies were chosen for their simplicity, efficiency in handling dynamic content, and ability to provide a responsive user interface while maintaining an organized backend for data storage. The implementation process is broken down into several key components, each addressing essential aspects of the application such as database design, frontend development, and interaction with server-side components.

### **5.1 Component Diagram**

A Component Diagram is a structural diagram that illustrates the organization and dependencies of the various components in the system. It provides a high-level overview of the architecture and shows how different modules work together to achieve the desired functionality.

In the context of the ExpenseHub project, the Component Diagram would outline the major software components and their relationships. These components include the frontend (implemented with HTML, CSS, JavaScript, and jQuery) and the backend (supported by Node.js and SQL). The diagram would also depict how these components interact through well-defined APIs to perform key functions such as expense logging, report generation, and data management.

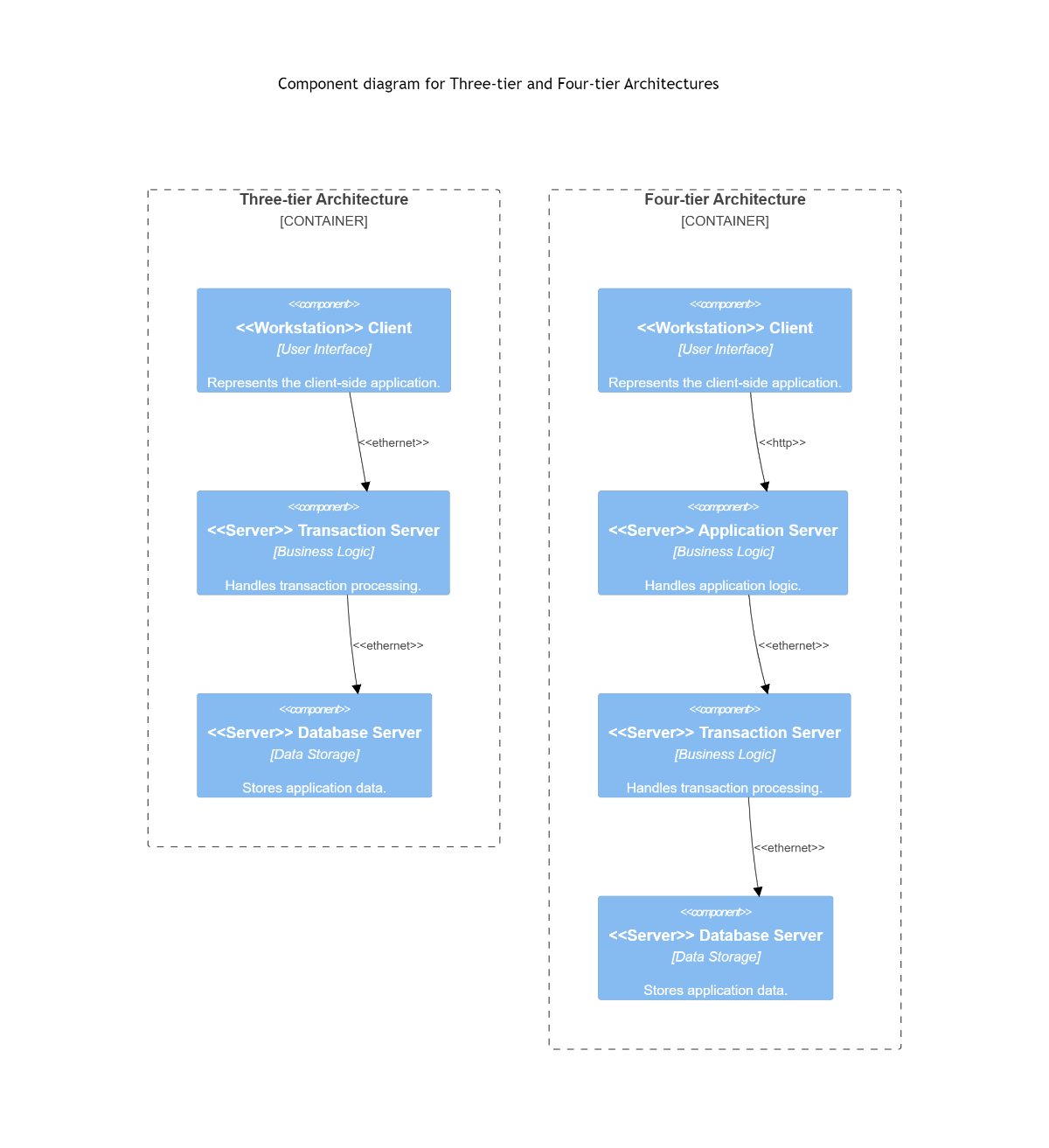
****

**Figure 5.1 Component Diagram**

### **5.2 Deployment Diagram**

A Deployment Diagram models the physical deployment of software components on hardware nodes, illustrating how the system is structured across different devices or servers. It provides an overview of how the software components are distributed and configured within the hardware infrastructure.

In the ExpenseHub project, the Deployment Diagram would depict how the frontend (built with HTML, CSS, JavaScript, and jQuery) and backend (powered by Node.js and connected to an SQL Database) are deployed on servers or cloud-based platforms. The diagram would also illustrate the relationships between these components and the hardware they run on, as well as the communication channels between the frontend and backend.



**Figure 5.2 Deployment Diagram**

### **5.3 Database Architecture (1- Tier, 2-Tier, 3- Tier Architecture)**

Database Architecture refers to how data is structured, accessed, and managed within an application. It determines how the system interacts with the database and organizes the flow of data between the user, application, and storage layers.

In the case of ExpenseHub, the application follows a 3-Tier Architecture, which separates the system into three distinct layers:

**Presentation Layer (Frontend):**

The Presentation Layer in ExpenseHub is handled by the HTML, CSS, JavaScript, and jQuery components. This layer is responsible for rendering the user interface, handling user interactions, and displaying data from the system. Users can log expenses, view reports, and manage their categories through this layer.

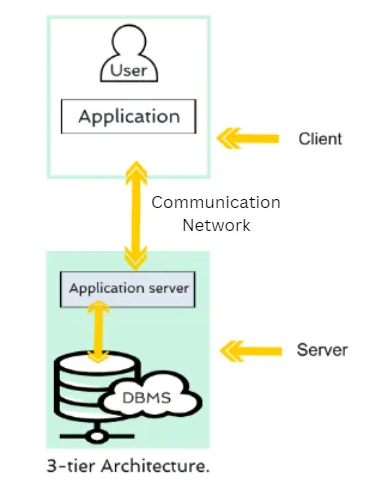
**Application Layer (Backend):**

The Application Layer is powered by Node.js and Express.js. This layer contains the business logic and processes all requests from the Presentation Layer. It interacts with the database, processes user requests such as logging expenses or generating reports, and serves the required data back to the frontend. This layer acts as the mediator between the user interface and the database.

**Data Layer (SQL Database):**

The Data Layer is represented by the SQL Database, where all data related to user accounts, expense records, categories, and generated reports are stored. The Application Layer communicates with the SQL Database to retrieve, update, or store data. This centralized storage ensures data integrity and efficient data management.

By adopting a 3-Tier Architecture, ExpenseHub ensures that data management is efficient, secure, and well-organized, while the business logic and user interface remain flexible and easy to maintain.



**Figure 5.3 Database Architecture Diagram**

# Chapter 6

# TESTING (SOFTWARE QUALITY ATTRIBUTES)

### **6.1 Test Case Specification**

Test Case Specification for ExpenseHub involves creating detailed test cases to describe specific inputs, actions, and expected outcomes to validate various functionalities. These test cases are designed to ensure that ExpenseHub meets its quality requirements, including functionality, usability, and security.

### **6.2 Black Box Test Cases**

Black Box Testing in ExpenseHub focuses on testing the application's functionality without any knowledge of its internal code structure. The goal is to verify that the system behaves as expected based on its requirements and specifications.

**6.2.1 BVA (Boundary Value Analysis)**

Boundary Value Analysis involves testing the values at the boundaries of input domains. The objective is to ensure that ExpenseHub handles edge cases correctly.

Example Test Case:

Verify if the system correctly handles the minimum and maximum values for expenses. Test cases may include logging an expense of $0, $10,000, and testing invalid amounts like negative values or excessively high numbers.

**6.2.2 Equivalence Class Partitioning**

Equivalence Class Partitioning groups similar inputs into classes to reduce the number of test cases while ensuring full test coverage.

Example Test Case:

Verify if ExpenseHub handles valid and invalid expense entries. Test cases include adding valid expenses (correct amount, category, date) and invalid entries (missing amount, invalid date format).

**6.2.3 State Transition Testing**

State Transition Testing ensures that the application transitions between different states correctly.

Example Test Case:

Verify if the system transitions correctly between the Add Expense state and the View Expenses state after submitting a new expense. Test cases include testing state transitions when an expense is successfully added or when an error occurs during input.

**6.2.4 Decision Table Testing**

Decision Table Testing validates combinations of inputs and their corresponding outcomes.

Example Test Case:

Verify if ExpenseHub generates correct reports based on different filters (e.g., date range, category). Test cases may include combinations of different filters to ensure the report generation functionality works as expected.

**6.2.5 Graph-Based Testing**

Graph-Based Testing models the application's behavior as a graph, and test cases are designed based on the graph's paths.

**Example Test Case:**

Verify if the system correctly handles different user navigation paths, such as logging in, adding expenses, and generating reports. Test cases include navigating through various paths and verifying that the system behaves as expected.

These Black Box Test Cases ensure that ExpenseHub functions correctly in terms of usability, security, and performance, covering a wide range of scenarios for effective quality assurance.

### **6.3 White Box Testing**

White Box Testing is a testing technique that allows the tester to access the internal code and structure of the software. In ExpenseHub, White Box Testing is used to ensure that all code paths are covered and function as expected. The following White Box Testing techniques are employed:

**6.3.1 Statement Coverage**

Statement Coverage measures the percentage of executable statements in the code that are executed during testing. This technique ensures that every line of code in ExpenseHub is tested at least once to verify that it behaves as intended.

Example: Test every line in the expense logging functionality to ensure that the expense data is correctly processed and stored in the database.

**6.3.2 Branch Coverage**

Branch Coverage measures the percentage of decision points (e.g., if-else conditions) in the code that are executed during testing. This technique ensures that all conditional branches in ExpenseHub are tested to verify that the system handles different decision outcomes correctly.

Example: Test different decision points in the Report Generation module, ensuring that each possible branch, such as valid date range and invalid date range, is covered.

**6.3.3 Path Coverage**

Path Coverage measures the percentage of unique paths through the code that are executed during testing. It ensures that all possible paths, from start to finish, are tested in ExpenseHub.

Example: Test all possible execution paths in the User Authentication module, including successful logins, failed logins, and password recovery flows.

By combining Black Box and White Box Testing techniques, ExpenseHub ensures comprehensive test coverage, helping identify potential defects and delivering a high-quality, reliable, and efficient application to its users.

# Chapter 7

# TOOL & TECNOLOGIES

The development and implementation of the "ExpenseHub" web app require a variety of tools and technologies to ensure efficiency, scalability, and security. In this chapter, we explore the programming languages, development frameworks, and operating systems employed in building and operating the system.

### **7.1 Programming Languages**

The development of ExpenseHub involved a variety of tools and technologies to ensure a responsive and secure system.

**HTML / JSX:** Used for structuring the UI and creating dynamic interface components.

**CSS:** Applied for styling the application, ensuring responsiveness and a user-friendly design.

**JavaScript (JS):** Handles dynamic operations, validating inputs, and updating the user interface.

**PHP:** Powers the backend, managing server-side logic, user authentication, and data handling.

**MySQL:** The database used to store and manage user data, including expenses and reports.

**Tools**

**VS Code:** IDE for coding and debugging the application.

**Browser Inspection Tool:** Used for testing and design tweaks in real time.

**Apache Tomcat:** Server for running backend services.

**XAMPP**: Local development environment for hosting the app during development.

**Heroku**: Platform for deploying the application online.

**GitHub** Desktop: Version control system for managing code updates and collaboration.

### **7.2 Operating Environment**

The "ExpenseHub" platform is developed on Windows-based operating systems using XAMPP for local development. However, the system is designed to be deployed on both Windows and Linux environments, with Linux being the preferred choice for web servers due to its stability, security, and wide support for PHP-based applications. The backend logic is executed on a server capable of running PHP, interacting with MySQL, and handling HTTP requests through Apache.

Kickstarting a project with proper development tools is very important. It provides developers with effective features to create, debug, maintain, and support the application throughout the development process. In this project, the software used includes Visual Studio Code 2019 as the primary coding tool, alongside XAMPP for local development testing.



**Figure 7.1 VS code**

**Minimum Hardware Requirements:**

Operating System: Windows 7 SP1 (with the latest updates) or later, Linux

Processor: 1.8 GHz or faster

Memory: 2 GB RAM (minimum)

Storage: 800 MB to 2 GB of available disk space

**7.3 Deployment Process**

The deployment of the ExpenseHub project on Heroku is a streamlined and efficient process, leveraging the platform’s scalability and performance features. Heroku provides a simple, yet powerful environment for hosting web applications like ExpenseHub, ensuring that it can handle various traffic loads without degrading performance.

Before deploying the application, both the frontend (UI/UX) and backend (PHP and MySQL operations) undergo optimization to ensure a production-ready build. This includes minifying assets, reducing file sizes, and ensuring database queries are efficient.

**Heroku:**Version Control Integration: The ExpenseHub project is managed using Git, allowing seamless integration with Heroku for continuous deployment. Whenever changes are pushed to the GitHub repository, Heroku automatically builds and deploys the latest version of the project.

**Continuous Deployment:** Heroku’s continuous deployment feature ensures that the latest codebase is always live. New commits trigger an automatic build process, ensuring that any updates to the ExpenseHub application are immediately available to users.

**Build and Optimization:** During the deployment process, Heroku generates optimized builds of the backend. The assets are minified and configured for faster loading times for end-users.

**Scalability and Performance:** Heroku’s environment automatically scales based on demand, ensuring that ExpenseHub can handle fluctuations in traffic without performance degradation. This serverless architecture ensures that resources are efficiently managed, allowing for smooth user experiences, even under heavy load.

**Global Content Delivery Network (CDN):** Heroku leverages a CDN to deliver static content faster by serving them from the nearest server location to the end-user. This significantly reduces latency, making the user experience smoother, especially for large file loads or during the generation of expense reports.

**Security:** Security is a top priority for Heroku, providing HTTPS encryption by default for all deployments. This ensures that user data, including login credentials and expense details, is securely transmitted between the client and the server.

**7.4 Monitoring and Analytics**

Heroku provides integrated monitoring and analytics tools that allow developers to track the performance of the ExpenseHub application in real time. Through these tools, insights into traffic, user behavior, and any potential errors can be analyzed, ensuring continuous improvement of the application. Performance logs are monitored to identify bottlenecks, and error tracking helps in quickly resolving issues as they arise.

**Table 7.1 Use Cases**

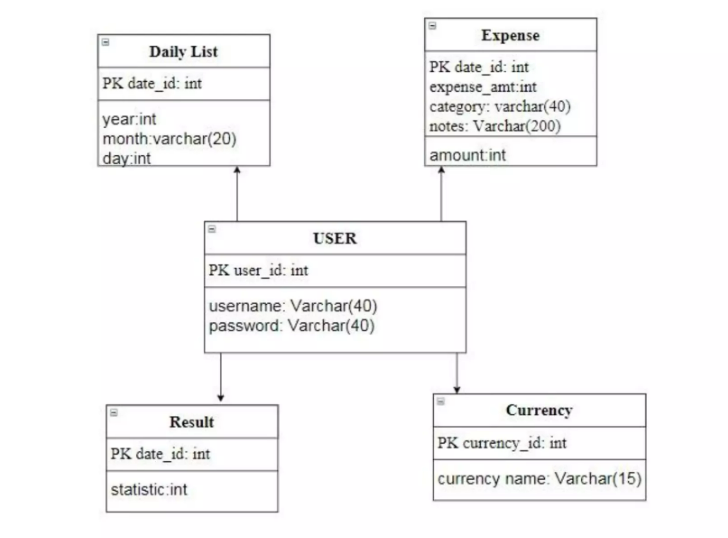
|  |  |  |
| --- | --- | --- |
| **Use Case** | **Actor** | **Goal** |
| Register as a User | User | To create an account on the platform. |
| Login | User | To log in to their account on the platform. |
| Dashboard | User | To browse the Dashboard on the platform. |
| Add Expense | User | To add a expense. |
| Manage Expense | User | To update or delete expense. |
| Generate Report | User | To generate expense report. |
| Update Profile | User | To manage User account settings. |
| Contact Customer Support | User | To contact customer support for help with any issues they may have. |

**Level 0 Data Flow Diagram**

At Level 0, the "ExpenseHub" system is viewed as a single process that interacts with the external entities (users). Users can submit expense details, request reports, and manage their profiles. Data storage involves interactions with a MySQL database that houses the user, expense, and report data.

**External Entities:** The external entities interacting with the system include the user (who adds, views, and manages expenses) and the system database (which stores user data).

**Data Stores:** The MySQL database stores all the relevant data, including user profiles, expense records, and reports.



**Fig 7.2 Data Storage Diagram L0**

In this representation, the "Data Storage (DBMS)" entity is shown separately, indicating that it houses various databases such as the "User Database," "Expense Database," and "Report Database." Each entity is connected with arrows depicting the flow of information or processes between them. The system interacts with these databases to store user profiles, track expenses, and generate reports.

**Level 1:**

At Level 1, the system's main processes are broken down into detailed subprocesses, providing a more comprehensive view of how ExpenseHub handles different operations.

**Receive Expense:** The process receives expense details from the user. The system validates the input (amount, category, date) and stores it in the database.

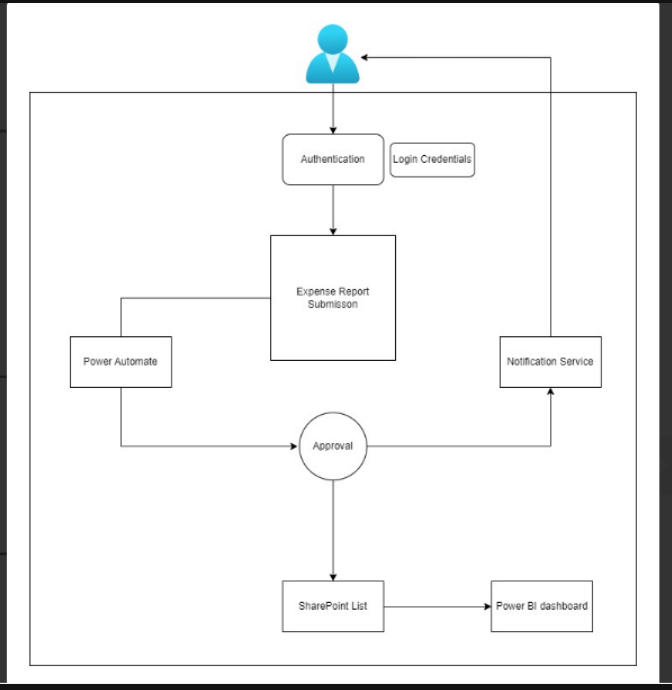
**Validate Expense:** The system ensures that the submitted expense is valid (e.g., positive amounts, proper date formats).

**Store Expense:** Once validated, the expense is stored in the MySQL database under the user’s profile.

**Generate Report:** The system can generate expense reports based on user inputs such as date ranges, categories, or specific timeframes. Reports can be viewed as lists or visualized through graphs.

**Manage Profile:** Users can update their profile details, such as username, password, and email, which are stored in the database.

**Logout:** The system logs the user out and terminates the session to prevent unauthorized access.



**Fig 7.3 Data Flow Diagram L1**

**Security and Performance Considerations**

Security and performance are critical aspects of the ExpenseHub implementation to ensure that user data is safe and the system runs smoothly.

**User Authentication:** Implement a secure user authentication system using hashed passwords and secure session management. PHP's built-in session handling functions are used to manage user sessions securely.

**Input Validation:** Ensure all user inputs are validated both on the client side (using JavaScript) and the server side (using PHP) to prevent SQL injection, XSS attacks, and other vulnerabilities.

**Database Security:** Protect the database by enforcing proper user roles and permissions. Limit database access to the necessary operations, and use prepared statements to prevent SQL injection.

Performance Optimization: Optimize SQL queries for performance, especially for large datasets when generating reports or displaying expenses. Implement pagination or filtering techniques to improve user experience during data retrieval.

**System Testing**

After development, extensive testing is performed to ensure the system functions correctly.

**Unit Testing:** Test individual modules and components to ensure they work as expected. For example, test the expense validation logic, the database query functions, and the login/logout process.

**Integration Testing:** Ensure that all components of the system (frontend, backend, database) work together without issues.

**User Acceptance Testing (UAT):** Test the system with real users to gather feedback on functionality, usability, and performance.

**Deployment**

Once development and testing are complete, the ExpenseHub system is deployed to a production environment. Deployment involves moving the application from the local XAMPP environment to a live server. This can be done using platforms like Heroku or any LAMP stack-based hosting provider. The system undergoes final performance testing on the live environment to ensure it can handle real user traffic efficiently.

**Frontend Development**

Develop the user interface (UI) components of the website, focusing on creating visually appealing and responsive designs. Implement the UI elements using HTML, CSS, and JavaScript, ensuring compatibility across different web browsers and devices.

**Backend Development**

Build the backend components that handle the business logic and data processing of the website. Develop server-side functionalities, such as user authentication, database interactions, and API integrations. Use appropriate programming languages and frameworks for backend development, ensuring robustness, security, and scalability.

**Database Setup**

Set up the database system and create the necessary database tables and relationships based on the database design. Implement database queries and interactions to store and retrieve data efficiently.

### **Appendix A: User documentation**

The ExpenseHub User Documentation is a comprehensive guide designed to help end-users effectively navigate and utilize the platform for managing their personal finances. It provides step-by-step instructions, illustrations, and explanations to ensure users can maximize the application's features and manage their expenses efficiently. The User Documentation includes the following sections:

**1. Introduction**

**Overview:** This section introduces the ExpenseHub application, offering a brief description of its purpose and key objectives. The primary goal of ExpenseHub is to help users track and manage their personal expenses in an organized manner, ensuring financial clarity and control.

**Benefits:** Highlights the advantages of using the platform, such as easy expense tracking, report generation, and visual expense graphs that give users better insights into their financial habits.

**Target Audience:** The documentation is aimed at end-users, such as individuals, freelancers, and professionals who wish to monitor and manage their personal or business expenses with ease.

**2. Getting Started**

**Accessing ExpenseHub:** Guides users through the process of accessing the ExpenseHub platform from various devices, including computers, smartphones, and tablets. Users can access the platform via any modern web browser.

**Creating a New Account:** Provides detailed, step-by-step instructions for new users on how to create an account. This includes:

* Filling out personal details like name, email, and password.
* Setting up a secure password for future logins.
* Verifying the account through email, if required.
* Logging In: Explains the login process for existing users. Users can log in by entering their registered email and password.

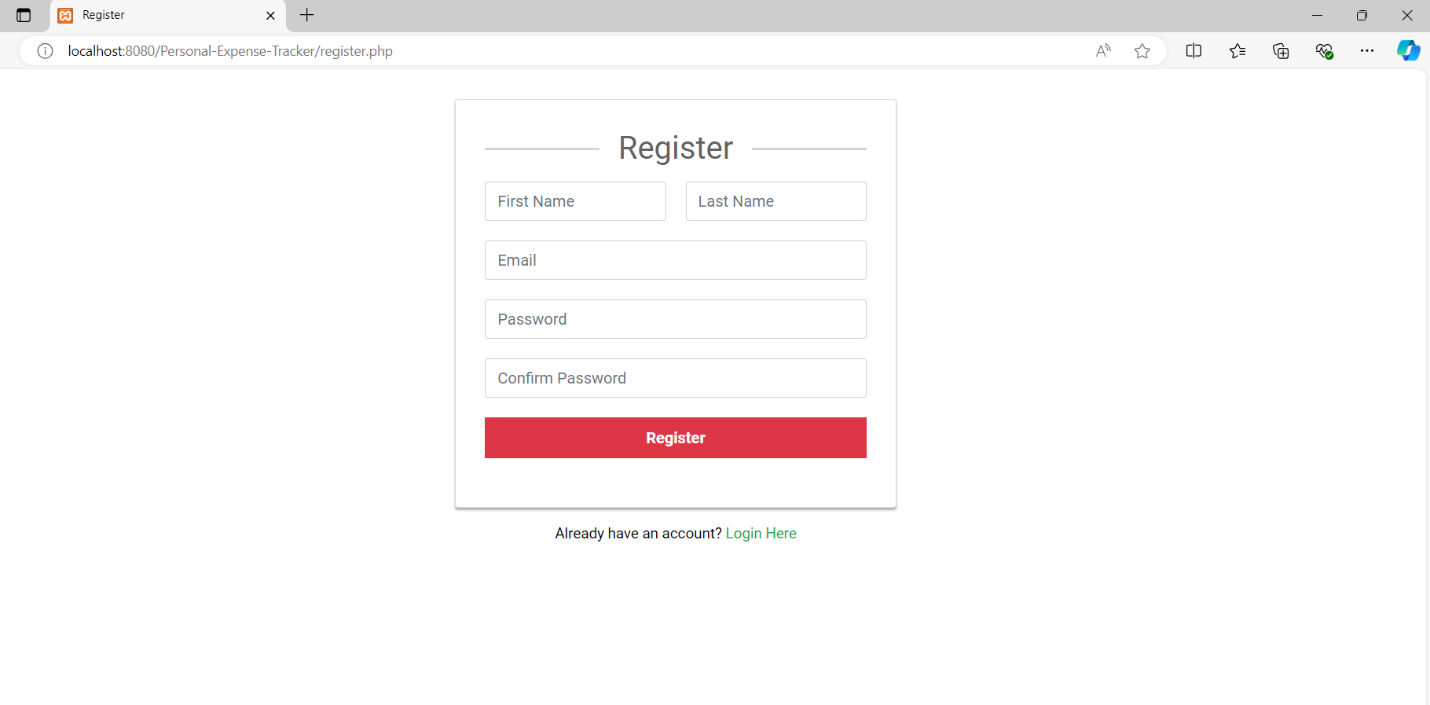
**3. User Interface**

Overview of the User Interface: Offers a detailed tour of the application's user interface (UI) to help users familiarize themselves with the design and layout of the platform.

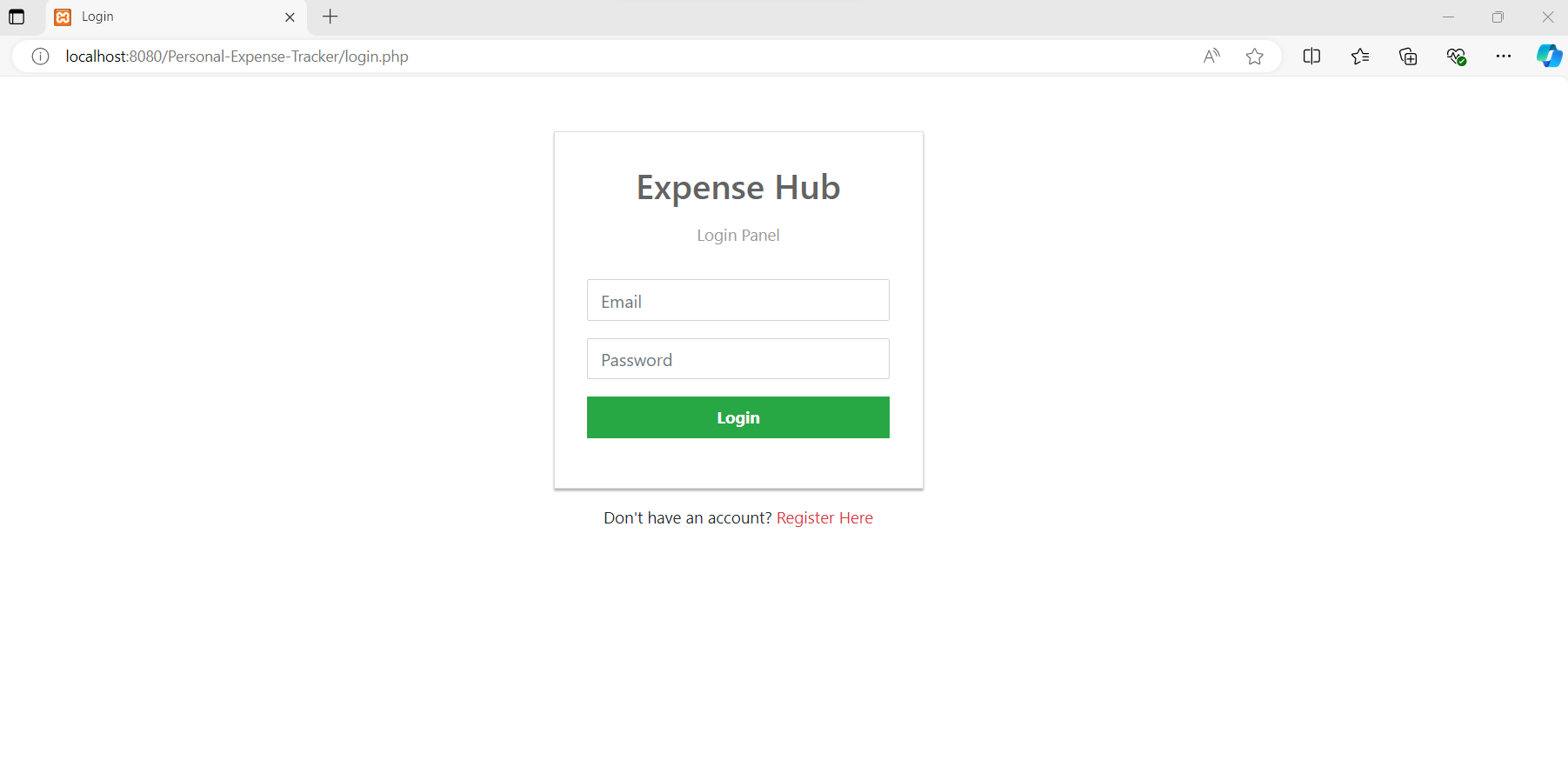
**Visual Aids and Screenshots:** Includes visual aids and screenshots to enhance understanding. This section demonstrates how to navigate the Dashboard, where users can view a summary of their expenses, total balances, and quick links to key actions like adding expenses or generating reports.

**Registration**

Allows new users to create an account by providing basic details like name, email, and password. Once registered, users can securely log in to manage their expenses.

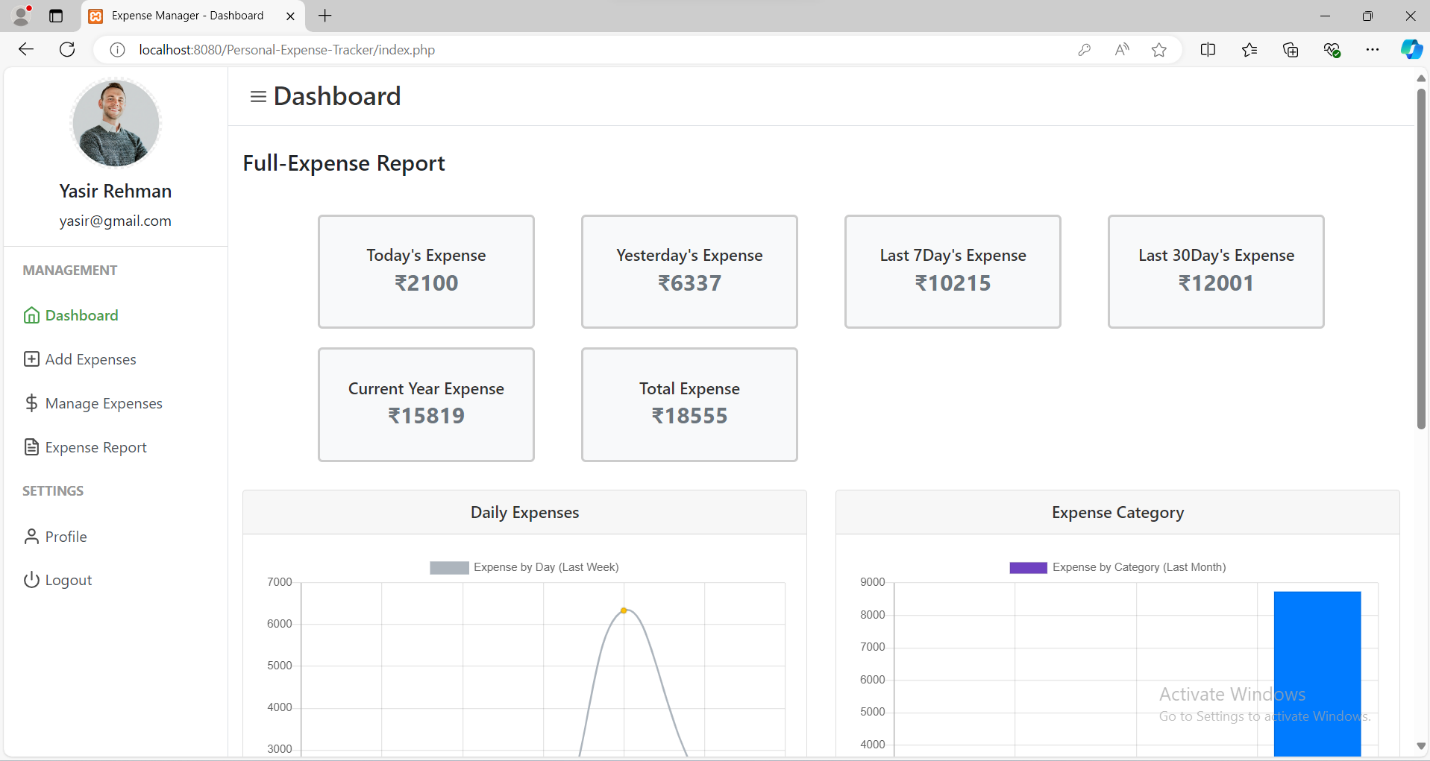
 **Login**

Users can log into their accounts using their registered email and password. It includes a password recovery option for users who forget their credentials.



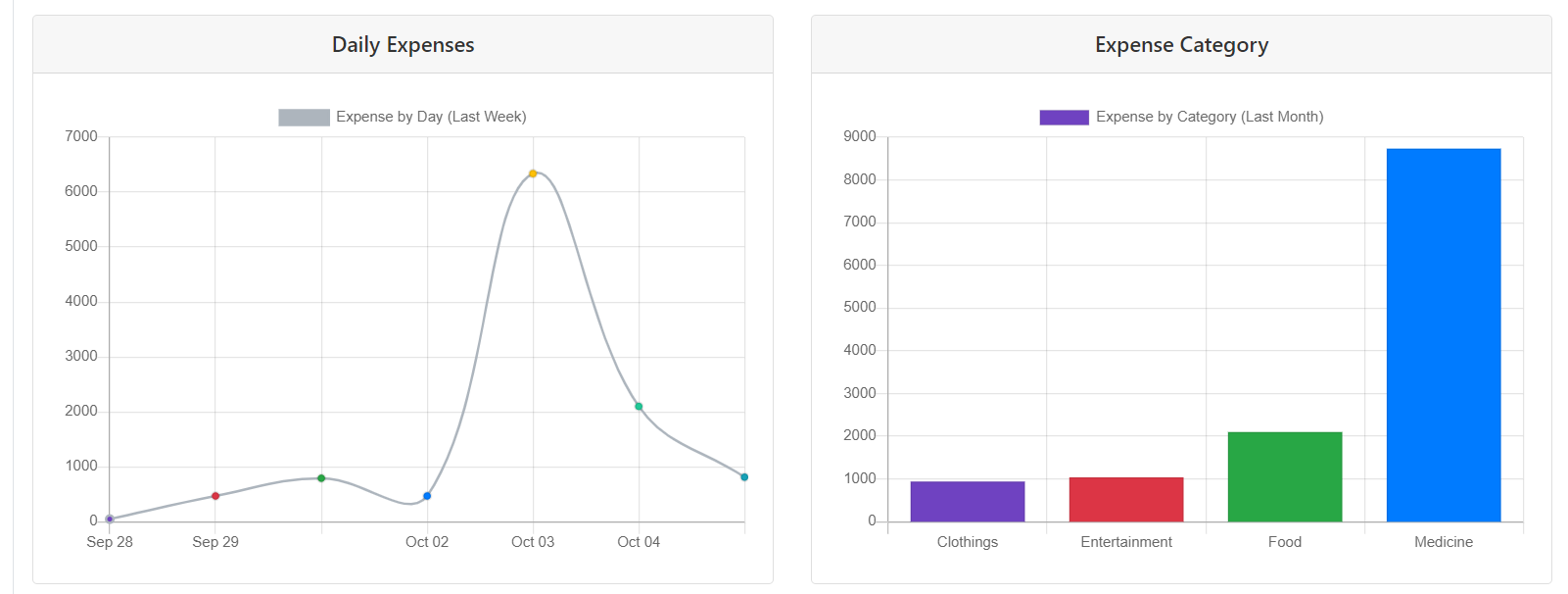
**Dashboard**

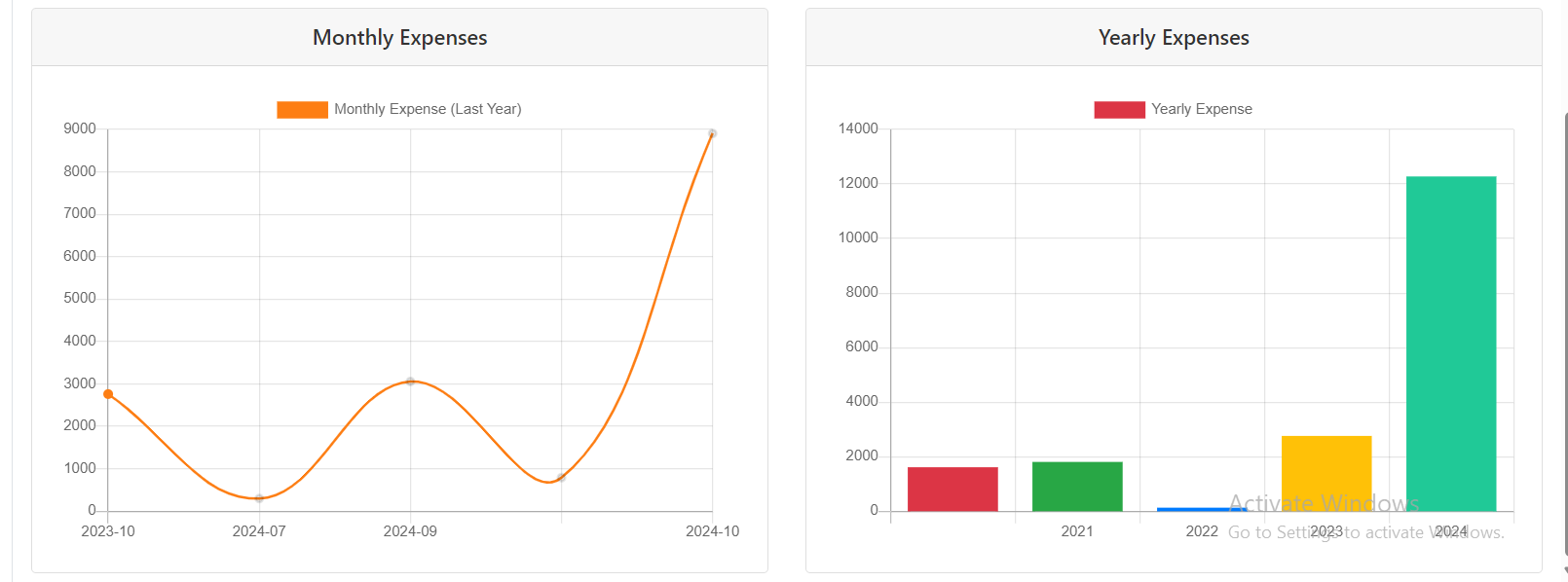
The dashboard provides a quick overview of recent expenses, account balance, and summary of spending categories, offering users insights into their financial activity.



**Dashboard Visualizations- Charts**

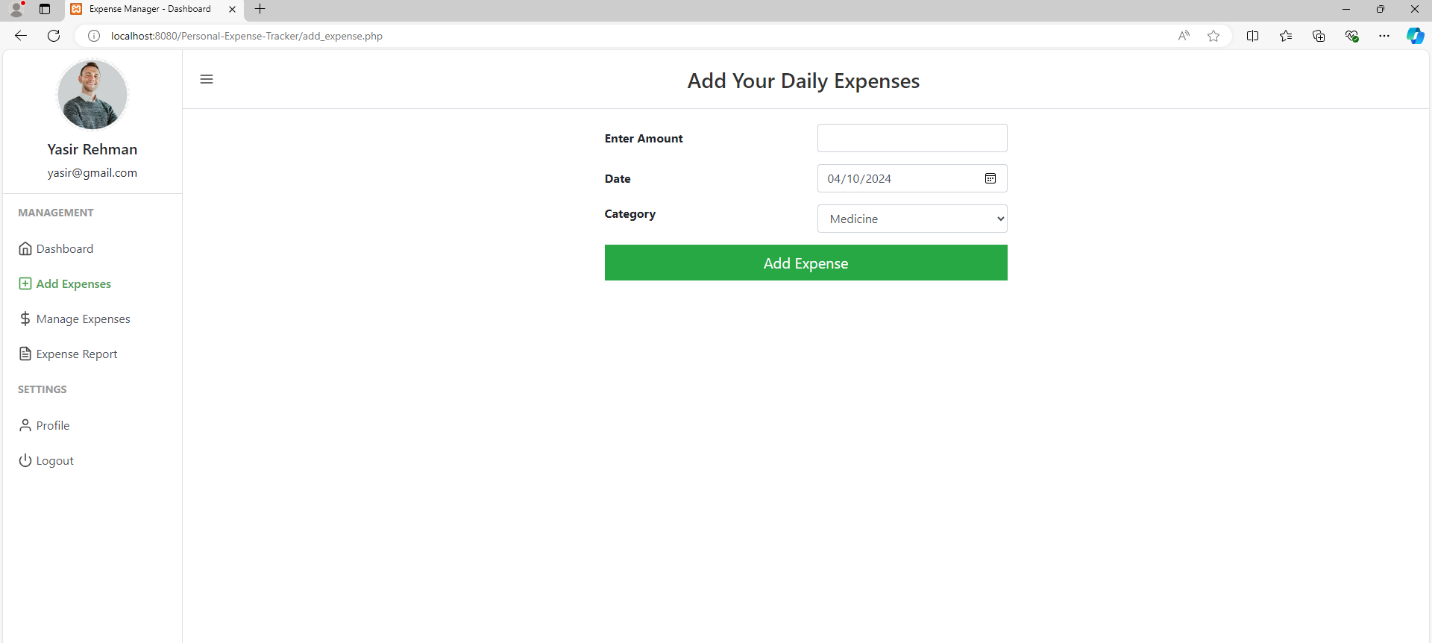
Displays visual representations of expenses through charts and graphs. This feature helps users analyze their spending patterns and identify areas where they can save money.





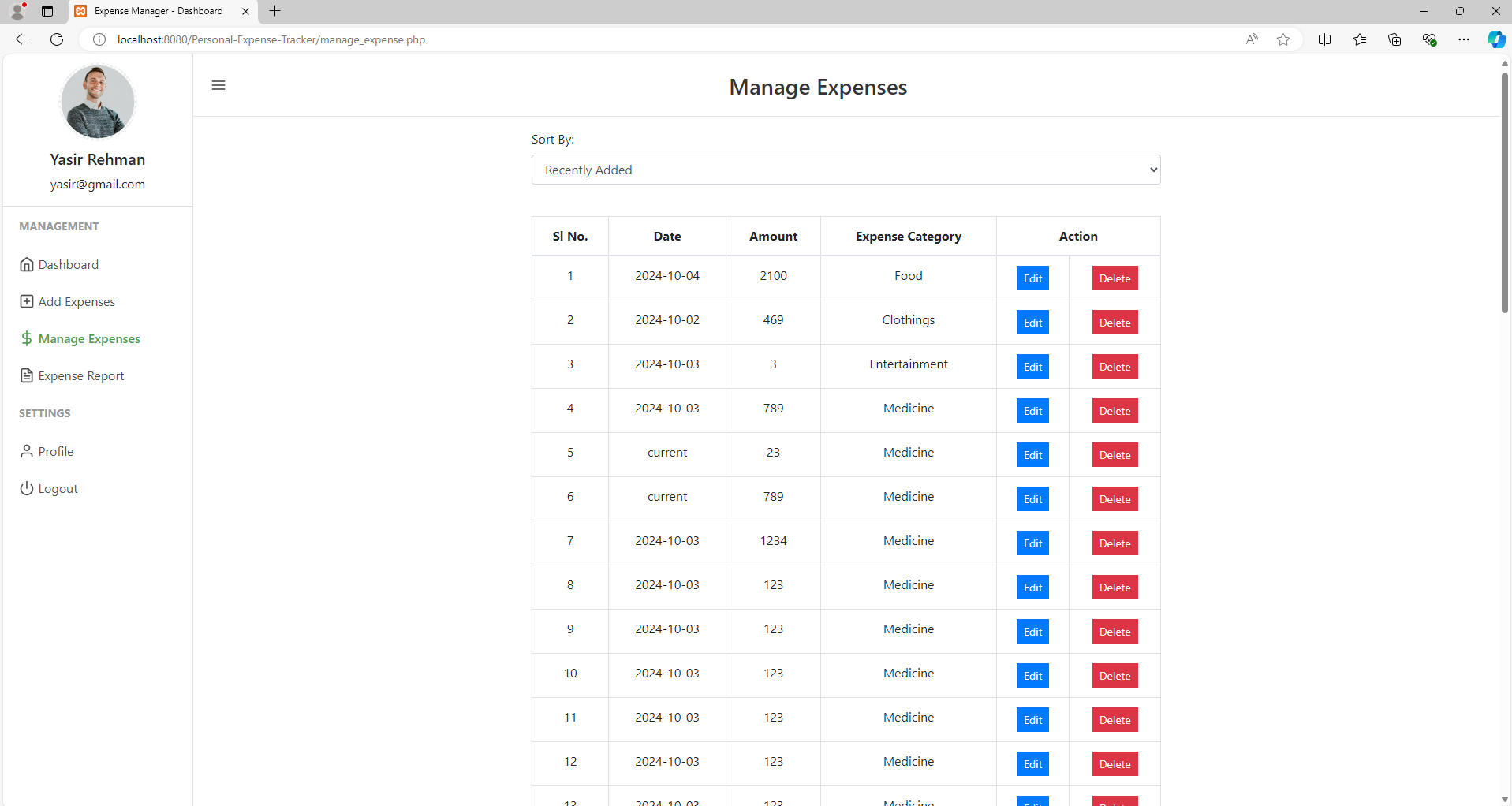
**Add Expense**

Users can add new expenses by filling out details such as the amount, date, category, and description. This feature helps users keep a record of their daily expenses.



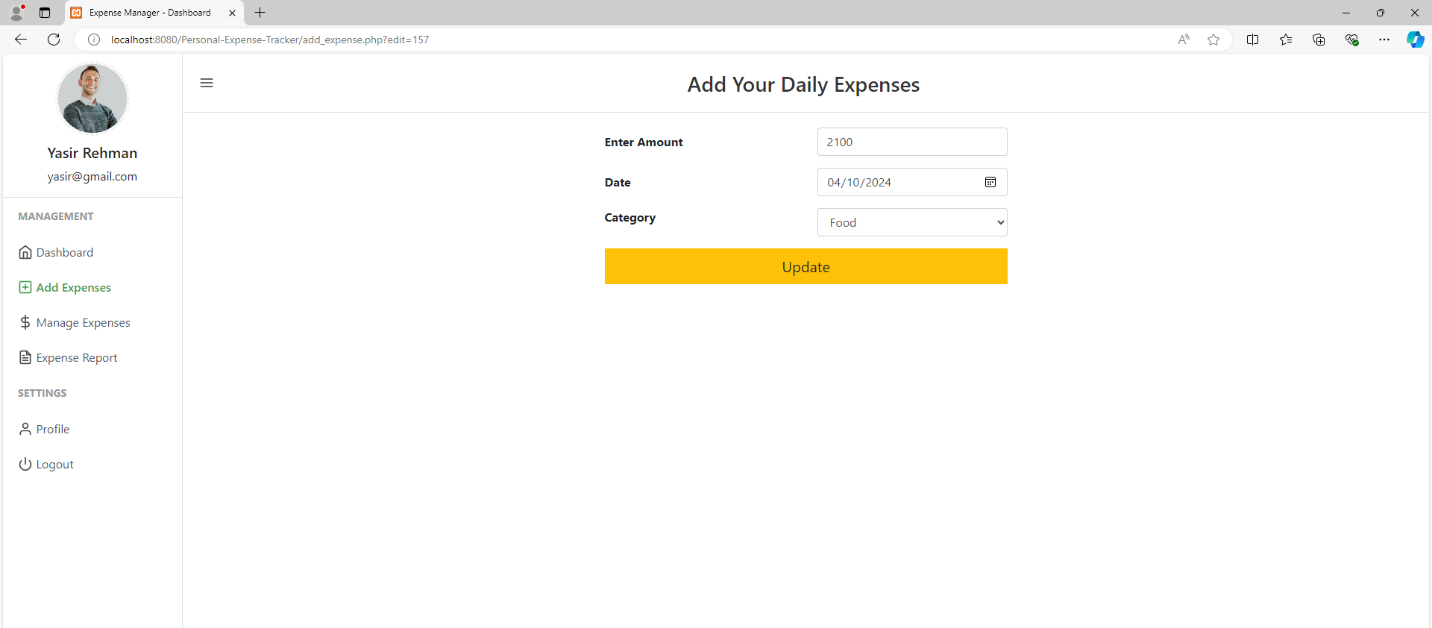
**Manage Expense**

Users can update or delete their existing expenses. This feature ensures flexibility in managing expenses and maintaining accurate records.



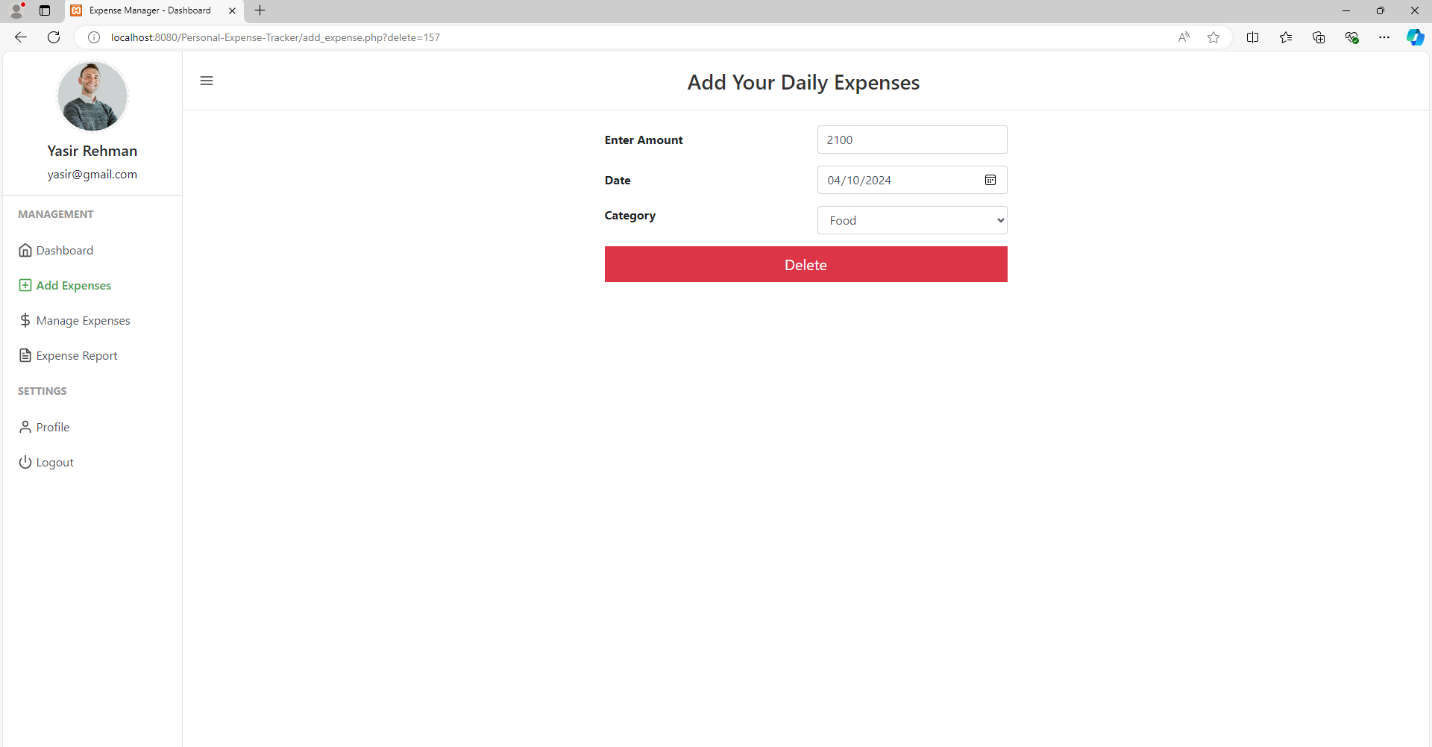
**Update Expense**

Users can update their existing expenses. This feature ensures flexibility in managing expenses and maintaining accurate records.



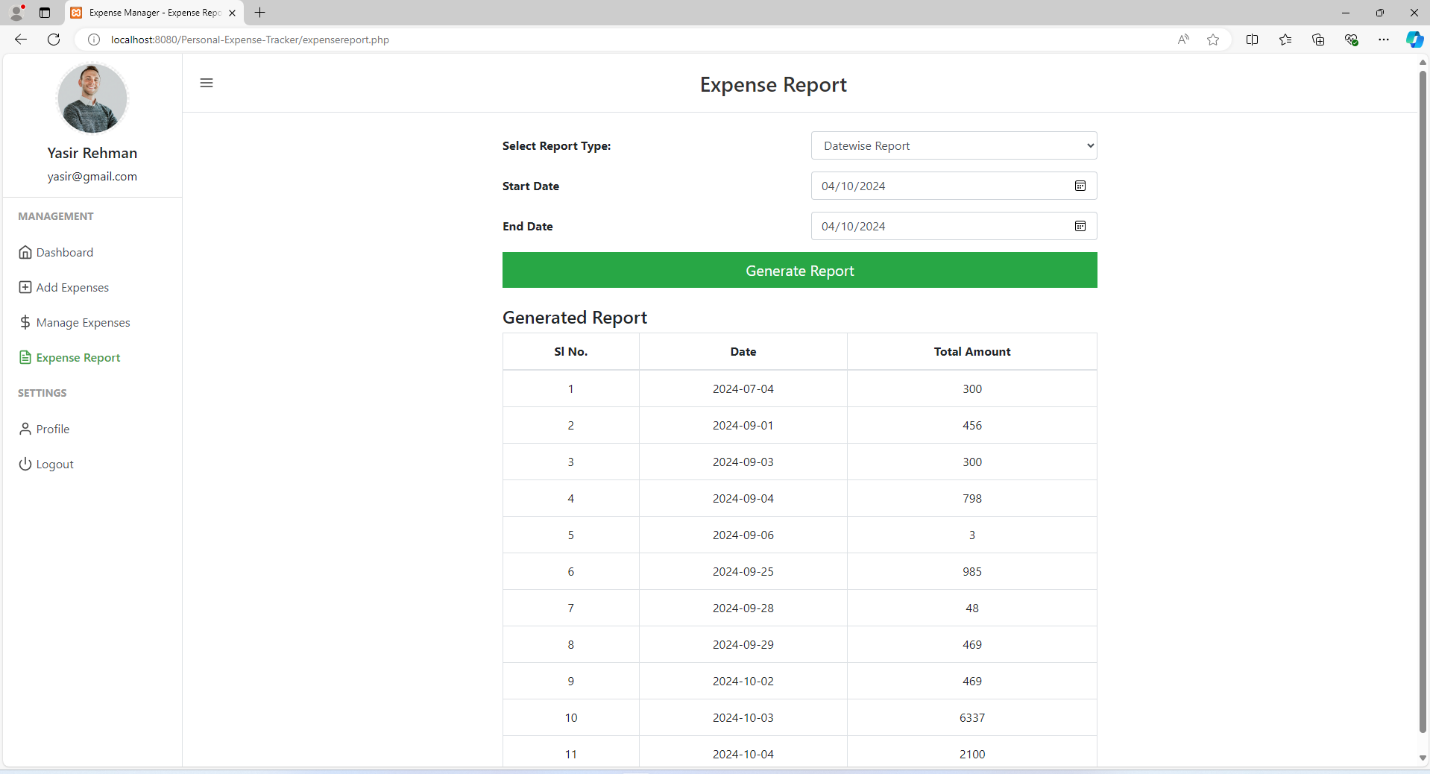
**Delete Expense**

Users can delete their existing expenses. This feature ensures flexibility in managing expenses and maintaining accurate records.



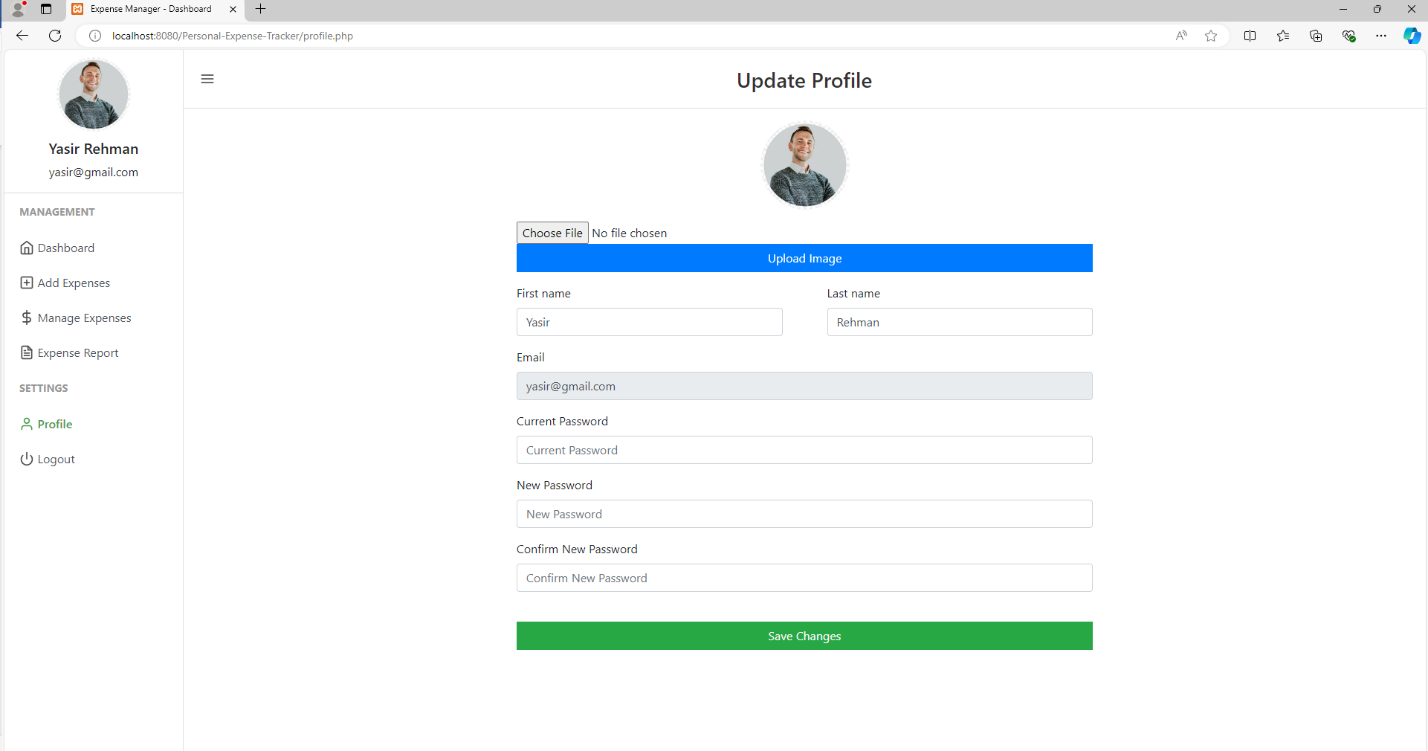
**Expense Report**

Generates detailed reports of user expenses based on selected criteria such as date range or category. Users can download or view these reports to track their financial history.



**Update Profile**

Allows users to update their personal details such as name, email, or password. This ensures that users can maintain their profile information and keep it up to date.



**14. Conclusion**

* Concludes the User Documentation with a summary of the key takeaways and benefits of using the " ExpenseHub " web application.
* Encourages users to explore the platform confidently and assures them of the support available for a smooth shopping journey.

The " ExpenseHub" User Documentation serves as a user-friendly and valuable resource, designed to empower users with knowledge and confidence in using the platform. Its detailed and comprehensive nature ensures that users can make informed decisions, navigate the application with ease, and enjoy a seamless, convenient, and enjoyable experience with managing his expenses.