**“EXPENSE TRACKER WEB APPLICATION”**



**A Final Project Report Submitted to the**

NARAYANI MODEL SECONDARY SCHOOL

FACULTY OF **DIPLOMA IN INFORMATION TECHNOLOGY**

COUNCIL FOR TECHNICAL EDUCATION AND VOCATIONAL TRANING

INSTITUTE OF TECHNICAL EDUCATION

Submitted By:

Bandana Gurung (reg no. DIT-0012-020, symbol no. 17001842)

Manisha Parajuli (reg no. DIT-00138-020, symbol no. 17001856)

In partial fulfillment of the requirement for the award of degree of

**DIPLOMA IN INFORMATION TECHNOLOGY**

Under the supervision of

**MR. Suraj Shrestha**

March , 2024

**NARAYANI MODEL SECONDARY SCHOOL**

**AFFILIATED BY CTEVT**

**DIPLOMA IN INFORMATION AND TECHNOLOGY**

**Certificate**

Certified that this is a bona fide record of the project work entitled

“**Expense Tracker Web Application”**

Done by the following student

**BANDANA GURUNG**

**MANISHA PARAJULI**

Of the 6th semester, Diploma in information and technology in the year 2080 in partial fulfillment of the requirements to the award of major in Diploma in information and technology of Narayani Model Secondary School affilliated by CTEVT.

Date:2080/11/19

................................. ............................

Mr. Ramsaran Duwadi Mr. Suraj Shrestha

(Principle) (Project supervisor)

.............................. ..............................

Mr. Deepak Tiwari Er. Devendra Kumar Yadav

(Head of Department) (External Teacher)

# Acknowledgements

I would like to thank everyone who helped us build the Expense Tracker Web Application so successfully.

We extend our thanks to project supervisor Suraj Shrestha, whose valuable insights and mentorship has been helpful in shaping the project. His constant encouragement and feedback have been really helpful in keeping us on track. This project would not have been possible without the support, guidance, directions and encouragement from teachers and our supervisor.

Bandana gurung

Manisha Parajuli

# Abstract

The Personal Expense Tracker Web Application is a user-friendly tool designed to help individuals track and manage their personal expenses. By registering and authenticating with their username and password, users can securely add income and expenses to the application. The dashboard provides an overview of the user’s income and expense transactions, as well as a summary of their total income and expenditure. The front end of the application is built using HTML and CSS, while the back end utilizes PHP and JavaScript. The server is powered by MYSQL, and the application is tested using unit testing, integration testing.

Table of Contents

[**Acknowledgements i**](#_Toc160905915)

[**Abstract ii**](#_Toc160905916)

[**INTRODUCTION 1**](#_Toc160905917)

[**1.1 Purpose 1**](#_Toc160905918)

[**1.2 Scope 1**](#_Toc160905919)

[**1.3 Intended audience 1**](#_Toc160905920)

[**1.4 Objectives 2**](#_Toc160905921)

[**1.5 Preliminary literature review 2**](#_Toc160905922)

[**1.6 Development methodology 3**](#_Toc160905923)

[**1.7 Document conventions 4**](#_Toc160905924)

[**SYSTEM ANALYSIS 5**](#_Toc160905925)

[**2.1 Functional requirements 5**](#_Toc160905926)

[**2.1.1 User registration and authentication 5**](#_Toc160905927)

[**2.1.2 Income add 5**](#_Toc160905928)

[**2.1.3 Expense add 5**](#_Toc160905929)

[**2.2 Non-functional requirements 5**](#_Toc160905930)

[**2.2.1 Performance 5**](#_Toc160905931)

[**2.2.2 Security 6**](#_Toc160905932)

[**HARDWARE AND SOFTWARE 7**](#_Toc160905933)

[**3.1 Hardware requirement 7**](#_Toc160905934)

[**3.2 Software requirements 7**](#_Toc160905935)

[**SYSTEM ARCHITECTURE 8**](#_Toc160905936)

[**4.1 login module 8**](#_Toc160905937)

[**4.2 Sign up module 9**](#_Toc160905938)

[**4.3 Dash Board 10**](#_Toc160905939)

[**SYSTEM DESIGN 11**](#_Toc160905940)

[**5.1 Data Flow Diagram (DFD) 11**](#_Toc160905941)

[**5.2 Entity Relationship Diagram 13**](#_Toc160905942)

[**5.3 Gantt chart 14**](#_Toc160905943)

[**5.4 Flowchart 15**](#_Toc160905944)

[**5.5 Database design 17**](#_Toc160905945)

[**5.6 System architecture design 19**](#_Toc160905946)

[**OVERALL DESCRIPTION 21**](#_Toc160905947)

[**6.1 Product perspective 21**](#_Toc160905948)

[**6.2 Product features 21**](#_Toc160905949)

[**6.3 User classes and characteristics 22**](#_Toc160905950)

[**6.4 Use case model 22**](#_Toc160905951)

[**TECHNOLOGY OVERVIEW 24**](#_Toc160905952)

[**7.1 Front end: HTML, CSS 24**](#_Toc160905953)

[**7.2 Backend: PHP, JavaScript 24**](#_Toc160905954)

[**7.3 Server: MYSQL 24**](#_Toc160905955)

[**7.4 XAMPP 24**](#_Toc160905956)

[**PROJECT DESCRIPTION 25**](#_Toc160905957)

[**8.1 Introduction 25**](#_Toc160905958)

[**8.2 Description of database tables 25**](#_Toc160905959)

[**8.3 Description of root directory content 27**](#_Toc160905960)

[**SCREENSHOTS WITH CODES 28**](#_Toc160905961)

[**TESTING 51**](#_Toc160905962)

[**9.1 Unit testing 51**](#_Toc160905963)

[**9.2 Integration testing 51**](#_Toc160905964)

[**9.3 System testing 51**](#_Toc160905965)

[**CONCLUSIONS 52**](#_Toc160905966)

**REFERENCE 53**

# LIST OF FIGURES

**Figure 1: level-0 DFD** 11

**Figure 2: level-1 DFD** 12

**Figure 3: ER Diagram** 13

**Figure 4: Flowchart** 15

**Figure 5: Database design** 17

**Figure 6: System architecture design** 21

**Figure 7: Use case model** 23

# INTRODUCTION

## **Purpose**

The primary purpose of this app is to provide users with a convenient and efficient way to keep track of their spending, understand their financial habits, and make informed decisions about budgeting and saving, this makes managing money easier. It helps users keep track of what they spend, set budgets, and understand where their money goes. By providing a simple and accessible platform, it aims to empower people to make smarter financial decisions and improve their overall financial well-being.

## **Scope**

The application offers a user-friendly interface for registering, authenticating and adding income and expense transactions. Users can also view their income and expenditure summary on the dashboard. The project’s scope includes a data flow diagram (DFD), entity relationship diagram (ERD), Gantt chart, Flowchart, and database design. The project’s root directory contains the necessary files for the web application, including HTML, CSS, PHP and JavaScript files. The application’s functionality is described in detail using a use case model, which represents the interactions between the user and the application

## **Intended audience**

The expense tracker app is for everyone who wants to keep track of their money easily. It's great for individuals and families, but also helps freelancers and small business owners to separate personal and work expenses. If you're trying to save money or just starting to manage your own finances, this app can help.

Basically, it's for anyone who wants to understand where their money goes and make smarter choices about spending.

## **Objectives**

Objectives of this project are

* Simplify Financial Management
* Provide insights into spending habits
* Effective Budgeting
* Data Analysis and Insights
* Real-Time Expense Tracking
* Security and Privacy
* Mobile Accessibility

## **Preliminary literature review**

Expense tracking has come a long way, from writing expenses in registers to using desktop apps like Quicken and Microsoft Money which was not so familiar with users. Later, QuickBooks became the preferred tool for small businesses, and apps like Personal Finance and Dollar Bird made it simpler to visualize expenses in chart or graphs with the calendar system. Recently, Mint became more well-known, and AI-powered apps like YNAB which was released in 2013 and Penny automated spending tracking.

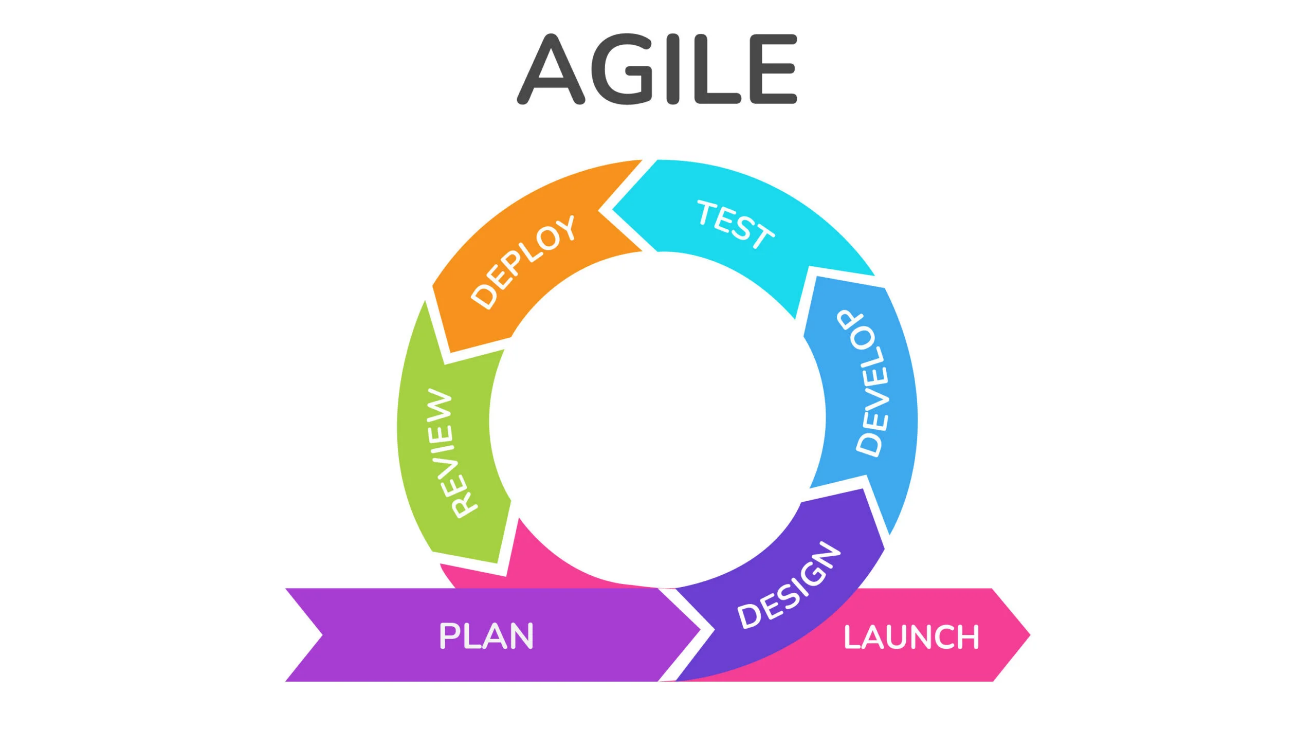
This application does not give any detailed information about alert message while our application will get that information via text message. Moreover, we will be working on simple, intuitive interface that feels familiars and easy to navigate than the other build application.Every feature of this project is designed to be user-friendly, from simple expense logging to eye-catching data visualizations for expense analysis. Our objective is to simplify and even enjoy the process of controlling spending, encouraging user familiarity and simplicity of use across the entire program.

* 1. **Development methodology**

For this project, we used agile methodology. It means we work in small, flexible steps rather than trying to do everything at once. Here’s how we approach it:

* Requirement gathering: Set goals and features, identify target users.

* Platform development:We build the online platform in stages, starting with the basic features and gradually adding more functionality based on requirements.
* Testing:we regularly tested the platform to ensure it works smoothly and meets the needs of users. This helps us identify any issues or areas that need improvement.
* Deployment: The platform is successfully deployed.



## **1.7 Document conventions**

Document conventions are an important part of technical documentation, helping to ensure that information is presented in a clear, concise and usable way. By following established conventions and guidelines, technical writers can create documentation that is easy to read, understand and use, improving the overall experience. Document convention used in our project are:

CSS : Cascading Style Sheet

HTML : Hyper Text Markup Language

PHP : Hypertext Preprocessor

DFD : Data Flow Diagram

ER : Entity Relationship

# SYSTEM ANALYSIS

## **2.1 Functional requirements**

Functional requirements are specific descriptions of what a software system must do to meet its intended purpose and objectives. They outline the system’s functionality, features and interactions with user or other systems. Here’s our projects functional requirements are:

### **2.1.1 User registration and authentication**

User can create their own accounts with unique usernames and secure passwords. Our login system ensures that only authorized user can access the platform, and we’ve even added a forgot password features for extra convenience.

### **2.1.2 Income add**

Users can input their income, such as salaries and payments. This feature provides a comprehensive view of their financial situation, enabling better budgeting and financial planning.

### **2.1.3 Expense add**

Users can input expense details, including the amount spent, expense description, which the app organizes for easy reference. The web app enables users to understand their financial habits better. Additionally, it facilitates budgeting by allowing users to set spending limits and monitor their progress.

## **2.2 Non-functional requirements**

Non-functional requirements describe characteristics of a software system that extend beyond its specific features and functions, such as performance, usability security or usability.

### **2.2.1 Performance**

Our web application is fast and works well. It loads pages quickly.

### **2.2.2 Security**

Password Encryption: user passwords are securely encrypted to keep them safe.

Authentication: we use secure mechanisms to authenticate users.

Access control: Role based access control is used to restrict user access to specific features.

# HARDWARE AND SOFTWARE

## **3.1 Hardware requirement**

* A computer system with a modern processor and at least 4GB of RAM
* A compatible operating system such as Windows, macOS, or Linux

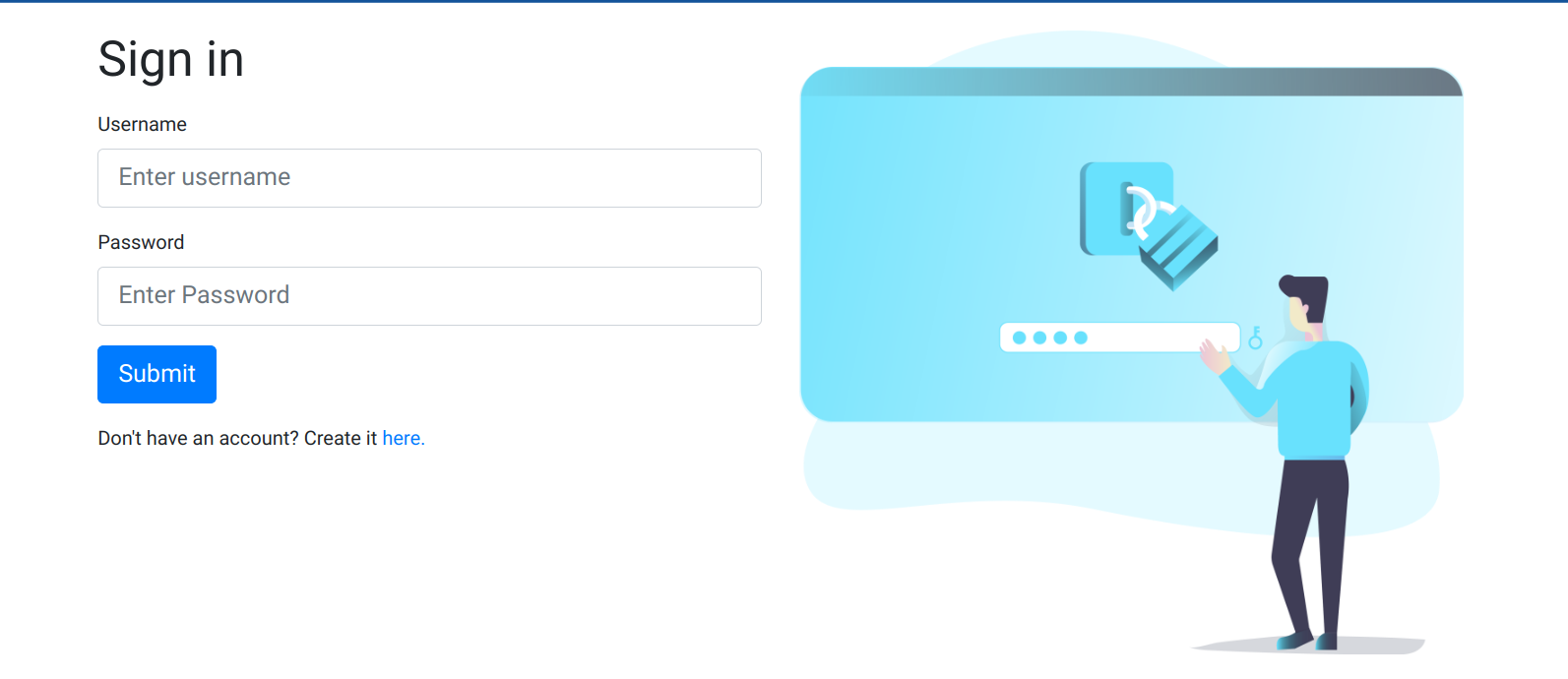
## **3.2 Software requirements**

* A modern web browser such as Google Chrome, Mozilla Firefox
* A reliable internet connection
* Optionally, a text editor or IDE for contributing to the project.

# SYSTEM ARCHITECTURE

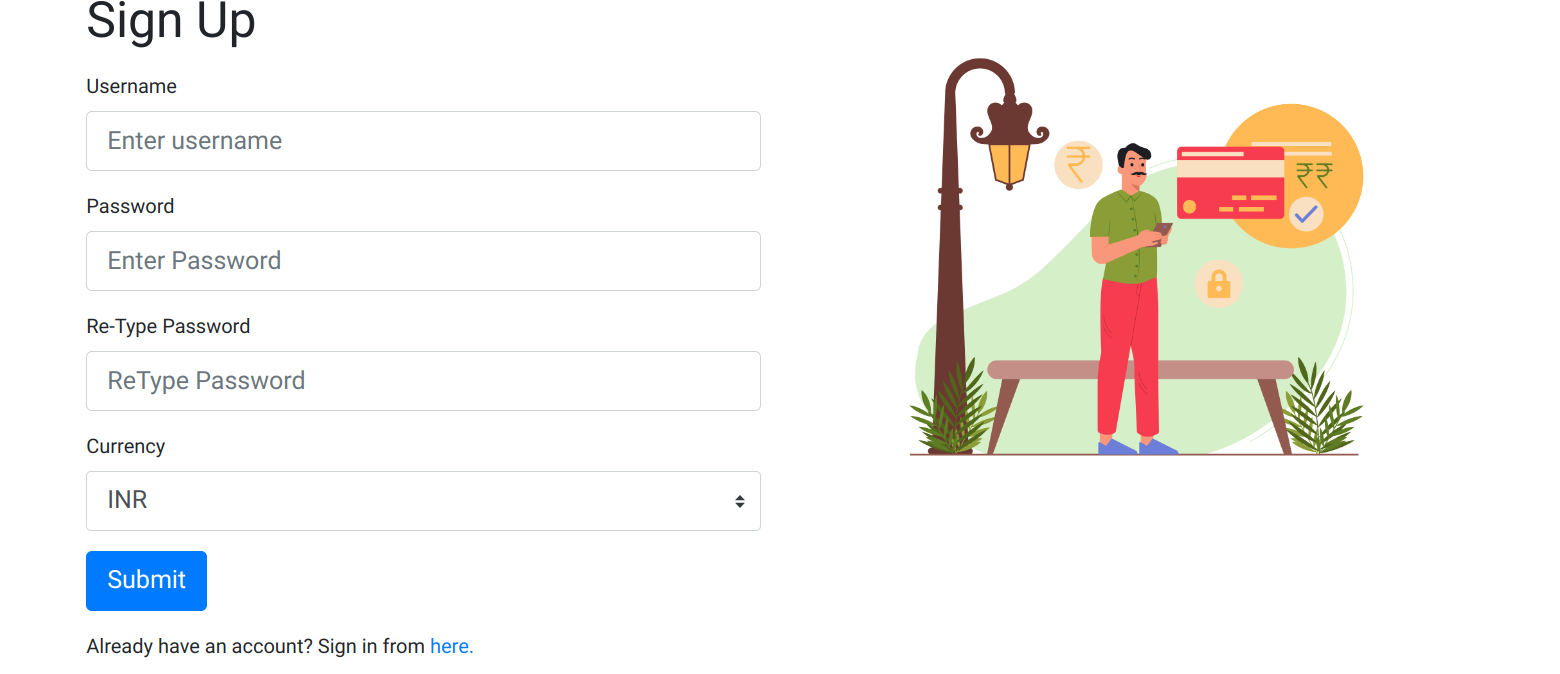
## **4.1 login module**

Users create accounts with unique credentials (username and password) for secure access. It authenticates identities and ensures data privacy, offering features like password recovery and for added security.



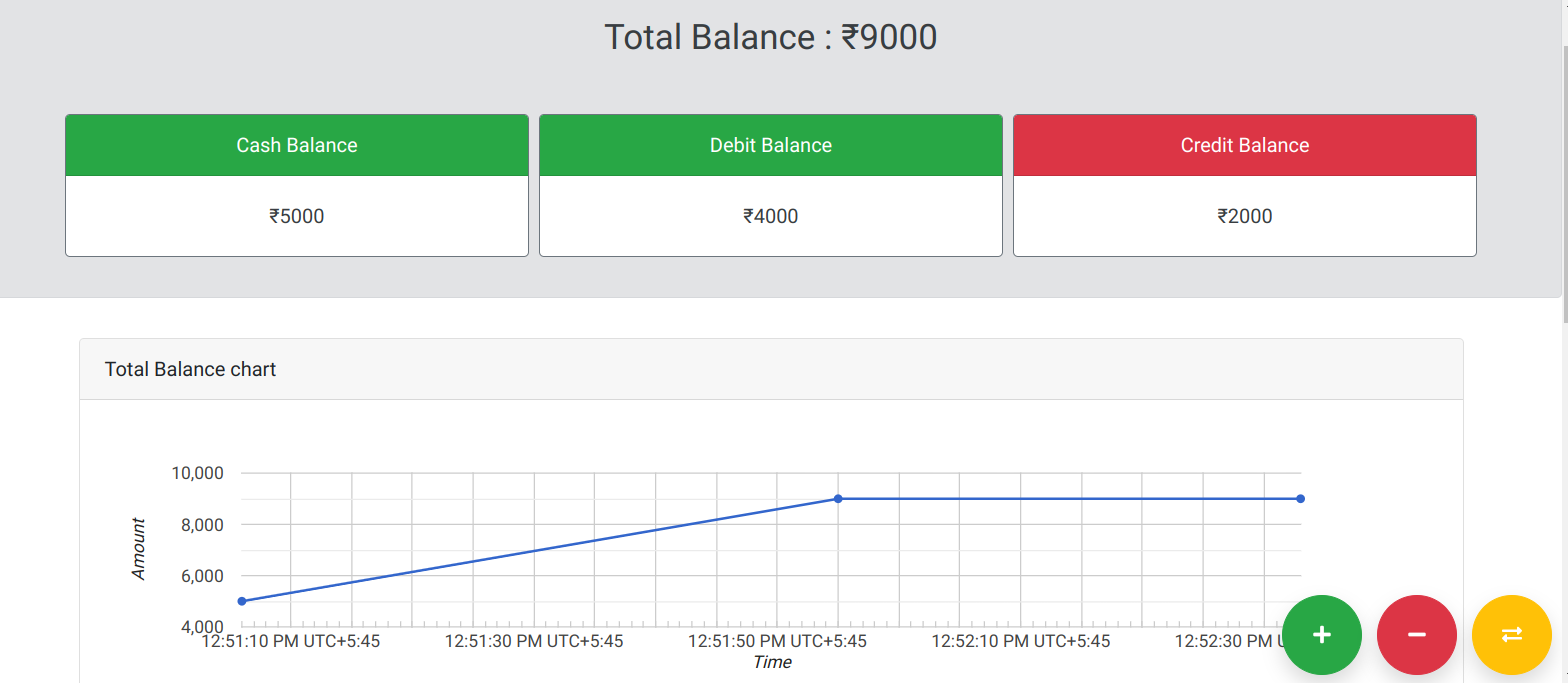
**4.2 Sign up module**

Users input basic info (email, username, password) for account creation. It verifies email addresses and enforces password strength, offering optional features. After registration, users gain access to expense tracking features.



## **4.3 Dash Board**

The dashboard in an expense tracker app provides users with a centralized overview of their financial activities. It displays key metrics such as total expenses, income, allowing users to quickly assess their financial health.



# SYSTEM DESIGN

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

A Data Flow Diagram (DFD) is a graphical tool that represents the flow of data in a system or process. It shows the inputs, outputs, and transformation of each entity, but does not include control flow or decisions rules.

Level-0 DFD of our system

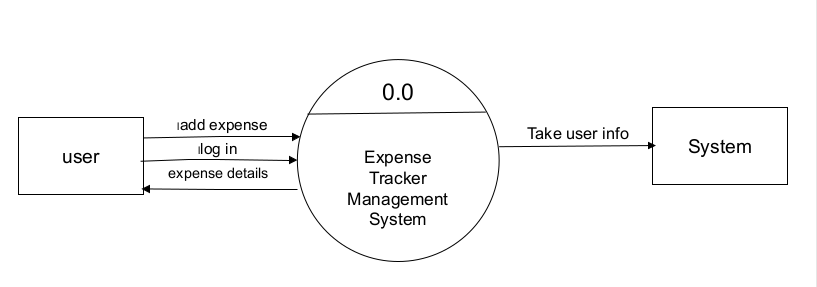


Fig1: level-0 DFD

In this DFD, the Expense tracker Management is the process, and the user’s information and expense details are the data flows. The user is the external entity that interacts with the system. The system takes the user’s information as input, processes it to create expense data.

Level-1

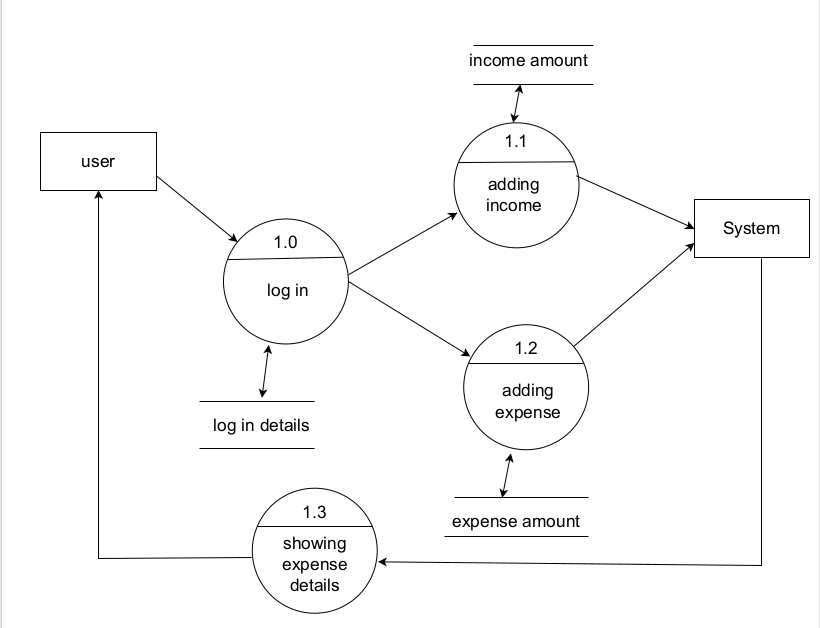


Fig 2: level-1 DFD

In this DFD, system has process for login, expense and income and expense details. The login process takes the user’s login details as input and produces a logged-in user. The income and expense details process take the user’s income and expense details as input and produces expense data. The use is the external entity that interact with the system. After all the process system then displays the expense details and store them for future use.

## **5.2 Entity Relationship Diagram**

An Entity Relationship (ER) diagram is a graphical representation of data and the relationships between different types of data. It consists of entities, attributes, and relationships. Entities are objects or concepts that can be identified and described. Relationship are the connections between entities. Attributes are the characteristics or properties of entities.

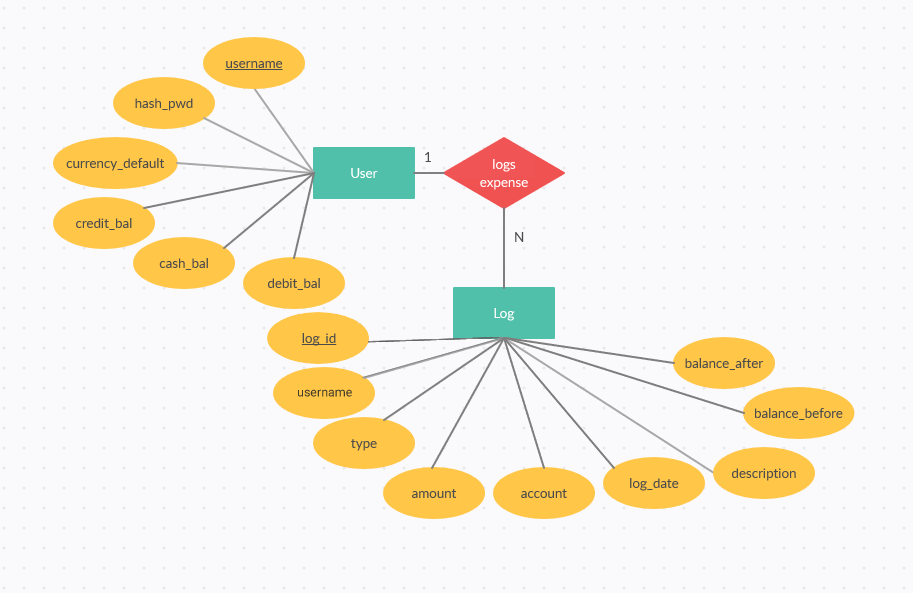


Fig 3: ER diagram

The given ER diagram represents a database schema for our expense tracker system which allows user to track their account balances. It consists of two entities: user, log.

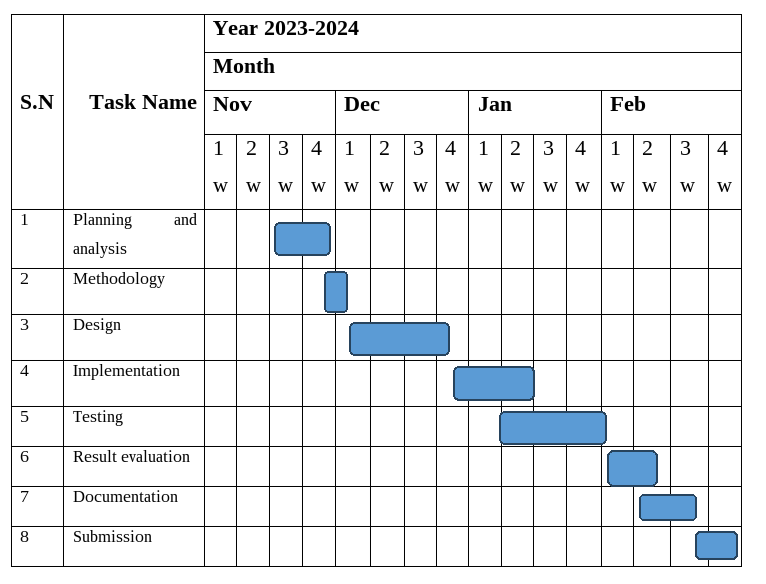
The user entity has six attributes: hash\_pwd, currency\_default, credit\_bal and cash\_bal,username debit\_bal. These attributes represent the user’s encrypted password, the default currency and balances for their credit and cash accounts.

The log entity has nine attibutes: log\_id, username, types, amount, account, log\_date, description, balance\_before and balance\_after. These attributes represent the unique identifier for each log entry, the username of the user who created the log entry, the type of transaction (either expense or deposit), and the balance before and after the transaction.

## **5.3 Gantt chart**

A Gantt chart is a graphical representation of a project schedule, showing the start and end dates, duration, and dependencies of individuals tasks. It is a popular tool used in project management to plan, track and manage projects.

It can be used to illustrate the schedule and timeline of the project, making it easier to understand the sequence and duration of tasks. It can help project managers and stakeholders to visualize the project’s progress, identify potential delays or risks, and make informed decisions.



February is the last month to be completed our project.

## **5.4 Flowchart**

A flowchart is a graphical representation of a process or workflow, showing the steps, decisions, and connections between them. It is a popular tool used in process analysis, design, and documentation to visualize and communicate the steps involved in a process.

A flowchart typically consists of a series of symbols, such as ovals, rectangles, diamonds, and arrows, that represents the steps, decisions, and connections in the process. The oval symbol is used to represent the start and end points of the process, the rectangle symbol is used to represent the process steps, and the diamond symbol is used to represent decisions. The arrows represent the connections between the symbols, showing the flow of the process.

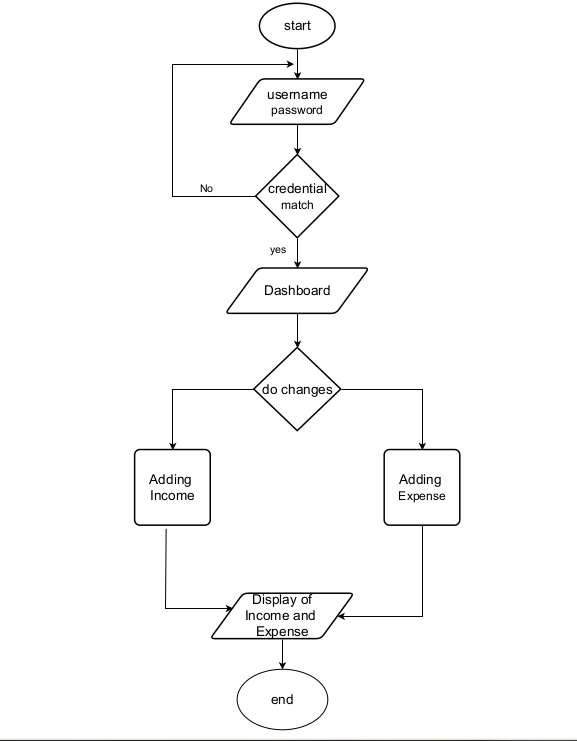


Figure 4: Flowchart

The flowchart shows the steps involved in logging into our application and adding, viewing and managing personal expenses and incomes. The user enters their username and password, and system checks if the credentials match. If they do, the user is taken to the dashboard, where they can add new expenses or incomes, view their expense and income history and manage their personal finances.

## **5.5 Database design**

Database design is the process of creating a structured format for storing, organizing, and managing data in a database. It involves defining the structure of the database, including the tables, fields, relationships, and constraints, to ensure the data is consistent, accurate, and easily accessible. A well-designed database can help improve the efficiency and performance of data-driven applications and make it easier to manage and analyse data. Database design typically follows a set of best practices and methodologies to ensure the database is scalable, secure, and maintainable over time.

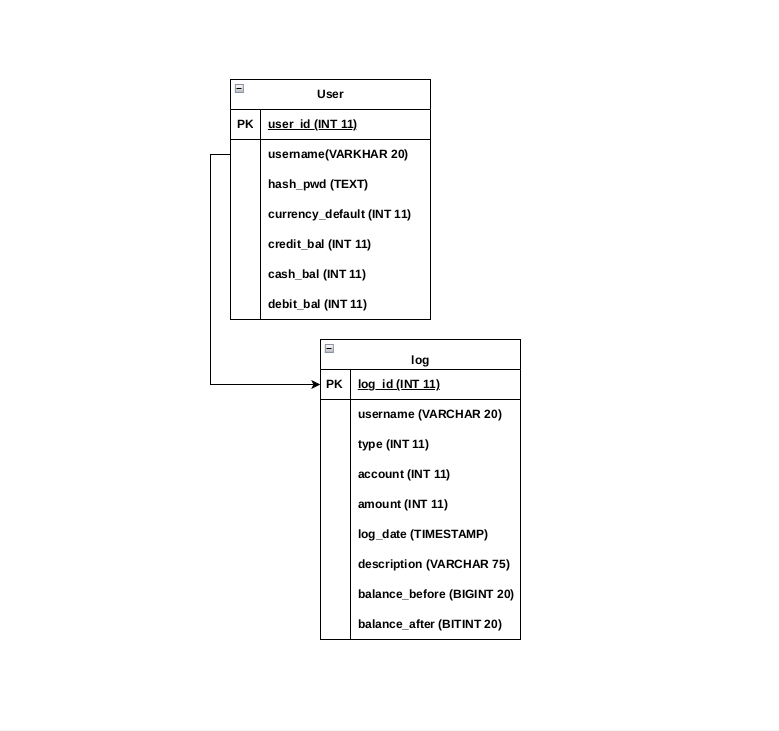


Figure 5: Database design

Our database schema includes two tables: user and log. The user tables store information about users, including their user ID, username, hashed password, default currency, and account balances for credit, cash, and debit accounts. The log table record activities related to the user’s username, the type of activity, the account type, the amount of the activity, the timestamp, a description, and the balance before and after the activity. The log table is linked to the user tables through the username field.

Key elements of database design

1. Tables: We have organized our data into different tables, each representing a specific type of information. For example, we have tables log and user.
2. Fields: Within each table, we define fields to store individual pieces of data. For instance, in the User table, we have fields like "Username," "hash\_pwd", "cash-bal", “credit”, “debit”.
3. Normalization: We have normalized our database to minimize redundancy and improve data integrity. The user and log tables are in at least the third normal form (3NF), as they do not t contain any transitive dependencies or partial dependencies. Each attribute (field) in the tables is atomic and functionally dependent on the primary key (user id) in the case of the user table and primary key (log\_id), and username field is a foreign key that links to the user tables in case of log table.
4. Data validation: we have implemented data validation rules to ensure that the information entered into the database is accurate and consistent. For instance, we validate email addresse to ensure they follow a proper format.

## **5.6 System architecture design**

A system architecture diagram is a visual representation of the components and relationships in a software system. It provides a high-level overview of the system's structure, including the main components, their interactions, and their dependencies. A system architecture diagram can help developers and stakeholders understand the system's overall design and architecture, as well as identify potential areas for improvement or optimization.

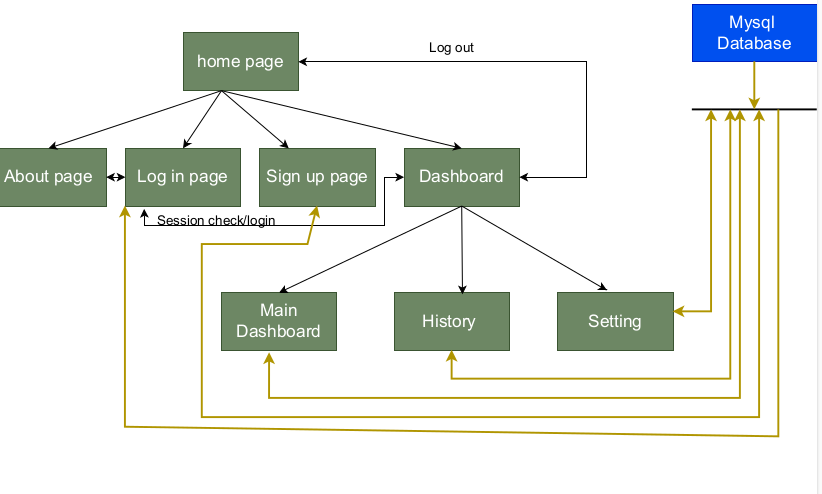


Figure 6: system architecture design

The system architecture design of our system includes following components:

Home page: this is the initial page users see when they access the application. It contains introductory information.

* About page: this page provides information about the application. It contain introductory information.
* Log in page and sign-up page: these pages are for user authentication. The log-in page is for existing user to access their accounts, while the sign-up page is for new users to create an account.
* Dashboard: this is the main interfaces for users after they log-in. It might contain various features like data visualization, settings etc.
* Session check/login: this function is for checking if a user is currently logged in. If not, it redirects the user to the log-in page.
* History: this page allows user to view their past activities or actions within the application.
* Setting: this page allows users to customize their preferences for the application.
* MySQL and database: these are backend components for data storage and retrieval. MySQL is a specific types of database management system used in this system.

In summary, this system architecture design visualizes component structure of our system with user authentication, a main user interface (dashboard), data management and various functionalities for user convenience.

# OVERALL DESCRIPTION

## **6.1 Product perspective**

Our personal expense tracker web application offers users a simplified solution for managing their finances. With intuitive sign-in and sign-up pages, users can quickly create accounts and access their financial data securely. The dashboard provides a clear overview of their income, expenses, empowering users to make informed financial decisions. By focusing on simplicity and accessibility, our product aims to help users achieve better financial control and peace of mind in managing their expenses.

## **6.2 Product features**

1. User Registration and Authentication:

* + Secure user registration with username, password, and email
  + User login with username and password
  + User authentication and access control

1. Income and Expense Transactions:
   * Add income and expense transactions with amount and description
   * Edit and delete transactions
2. Income and Expenditure Summary:
   * Display total income, total expenditure, and net balance
   * Visual representation of income and expenditure.
3. Secure Data Storage and Management:
   * Secure user registration and login process
   * Encrypted data storage in the database
   * Secure data transmission between the server and the client
4. Compatibility and Browser Support:
   * Compatible with modern web browsers like Google Chrome.

## **6.3 User classes and characteristics**

The Expense Tracker web application has two main user classes: registered users and unregistered users.

1. Registered Users:
   * Have a valid username, password, and email
   * Can authenticate and access the application's features
   * Can add, edit, and delete income and expense
2. Unregistered Users:
   * Can view the application's features.
   * Can access the homepage and about page and sign up page
   * Can register for a new account using their username, password, and email

## **6.4 Use case model**

A use case model is a visual representation of the interactions between users and a system. It includes actors, use cases, relationship, and flow of events. Actors are individuals or systems that interact with system, while use cases describe specific actions or tasks that can be performed. Relationships illustrate the interactions between actors and use cases, and the flow of events outlines the sequence of steps in each use case. Use case models help define functional requirements, clarify user needs, and ensure that the system is designed to meet user goals, as well as identify gaps, potential errors and areas for additional testing.

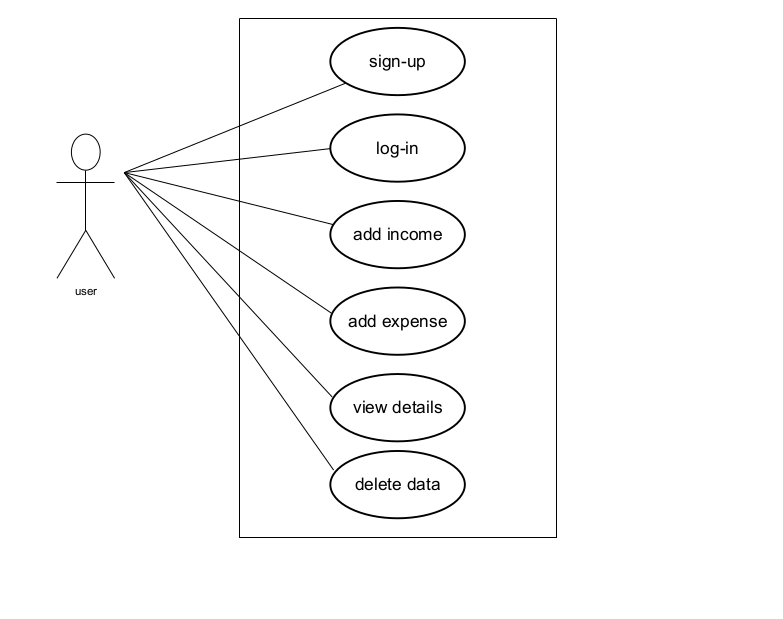


Figure 7 : use case model

This figure shows a simple use case model for a personal finance management system, which includes following cases:

* User sign-up: the process of creating a new user account in the system.
* User log-in: the process of logging into an existing user account.
* Add income: the process of adding new income record to the user’s account.
* Add expense: the process of adding new expense record to the user’s account.
* View details: the process of viewing the user’s account details, including income and expense records.
* Delete data: the process of deleting data from the user’s account.

In summary, this use case model describes our system that allows users to sign up, log in, add income and expense records, view account details, and delete data. The use case model focuses on the functional requirements of the system and the interactions between the user and the system, providing a clear and concise description of the system's capabilities.

# TECHNOLOGY OVERVIEW

## **7.1 Front end: HTML, CSS**

HTML (Hypertext Markup Language): Used to create the structure and content of web pages.

CSS (Cascading Style Sheets): Used for designing and formatting the appearance of web pages, ensuring a visually appealing and user-friendly interface.

## **7.2 Backend: PHP, JavaScript**

PHP (Hypertext Preprocessor): It is a server-side scripting language that handles data processing, server communication, and business logic.

JavaScript: It is used for client-side interactivity, enhancing user experience through features like dynamic content and form validation.

## **7.3 Server: MYSQL**

MySQL is a relational database management system (RDBMS) used to store, retrieve, and manage data efficiently and securely.

## **7.4 XAMPP**

XAMPP is a software package that includes Apache (a web server), MySQL, PHP, and Perl, creating a development environment for building and testing web applications locally before deploying them to a live server.

# PROJECT DESCRIPTION

## **8.1 Introduction**

This documentation is about simple personal expense tracker web application. The application will allow users to add and view their expenses.

## **8.2 Description of database tables**

Table 1: user table

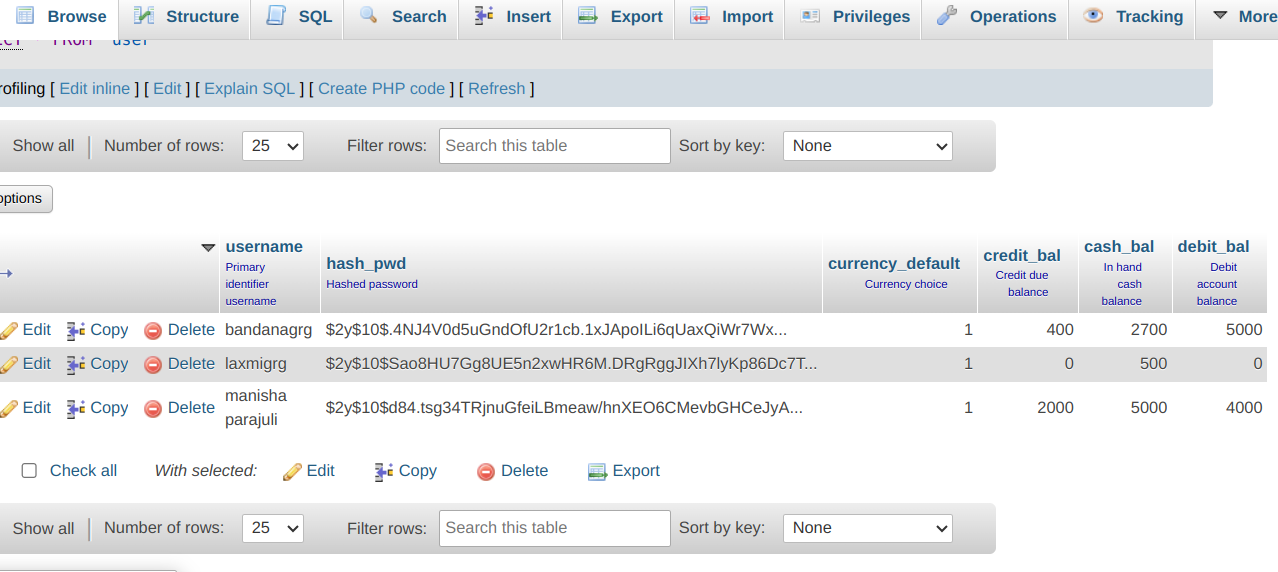
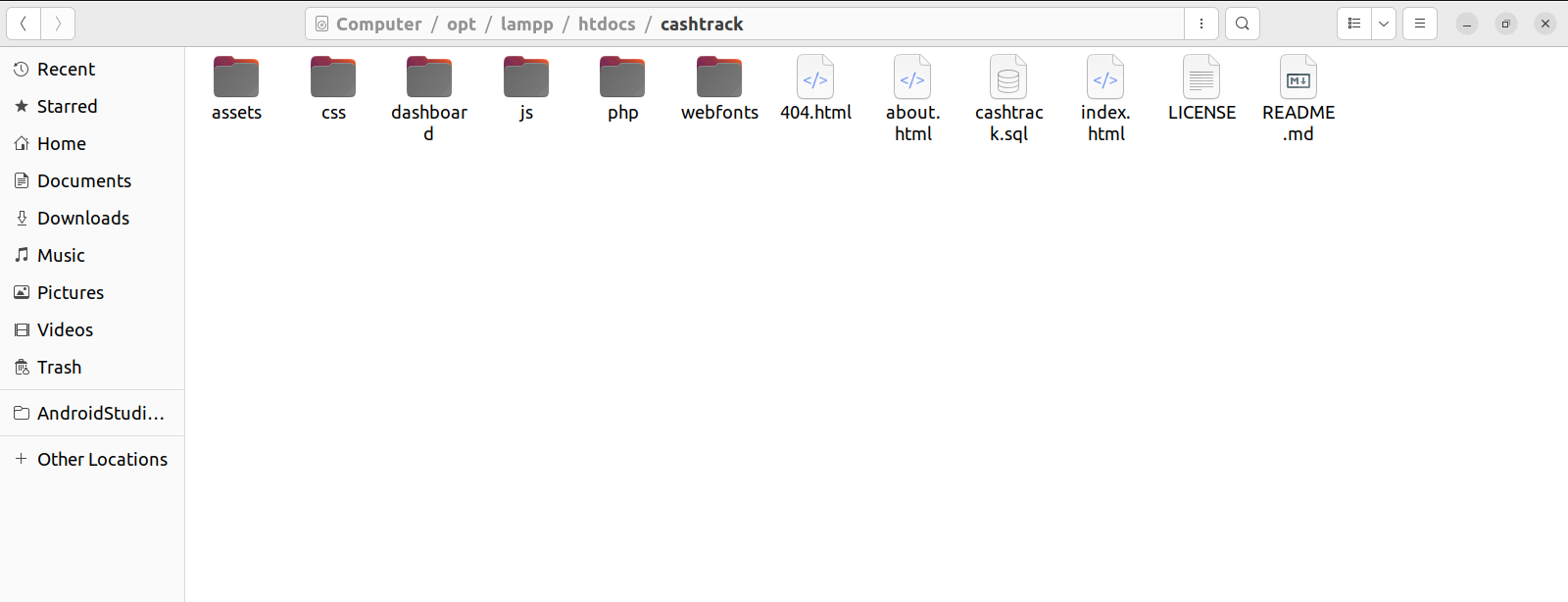


Table 2: log table



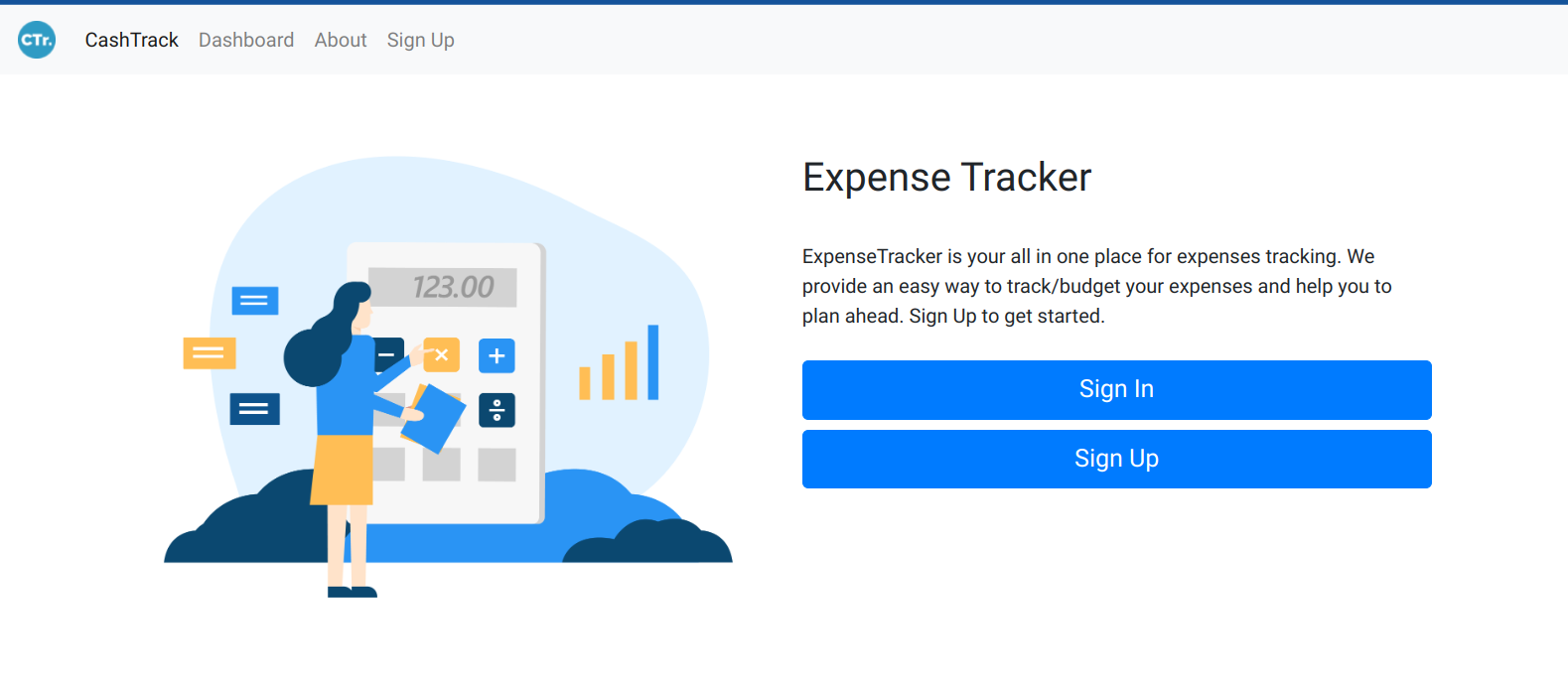
## **8.3 Description of root directory content**

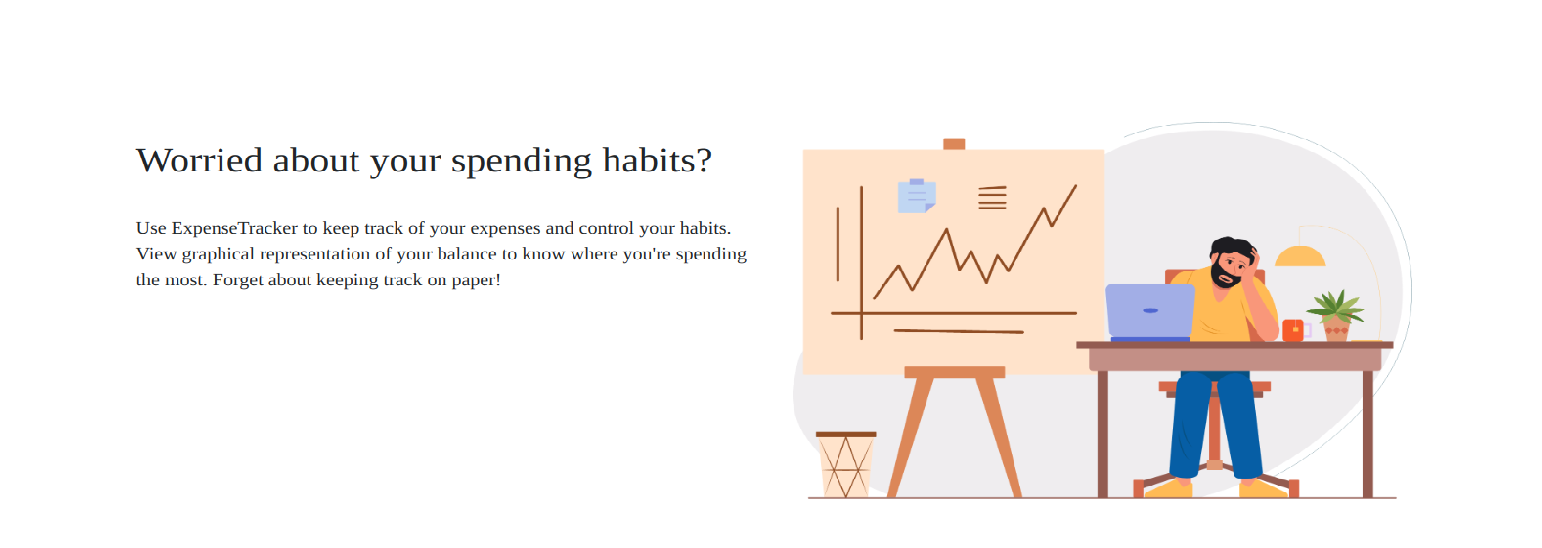
The directory structure of the project is as follows:



# SCREENSHOTS WITH CODES

**Homepage**

****



<!DOCTYPE html>

<html lang="en">

<head>

<body>

<!--Nav bar-->

<header>

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<a class="navbar-brand" href="">

<img src="./assets/favicon/android-icon-36x36.png" width="30" height="30" class="d-inline-block align-top" alt="Logo">

</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNavAltMarkup">

<div class="navbar-nav">

<a class="nav-item nav-link active" href="">CashTrack <span class="sr-only">(current)</span></a>

<a class="nav-item nav-link" href="./dashboard/">Dashboard</a>

<a class="nav-item nav-link" href="./about.html">About</a>

<a class="nav-item nav-link" href="/cashtrack/php/pages/signup.php">Sign Up</a>

</div>

</div>

</nav>

</header>

<!--Main page-->

<main>

<div>

<!--First-->

<div class="container p-5">

<div class="row">

<div class="col-md-6 col-sm-12">

<img src="./assets/illustrations/asset-1.png" class="image" alt="banner">

</div>

<div class="col-md-6 col-sm-12">

<p>

<h2>Expense Tracker</h2><br>

ExpenseTracker is your all in one place for expenses tracking. We provide an easy way to track/budget your expenses and help you to plan ahead. Sign Up to get started.

</p>

<div class="row">

<div class="col-12 ">

<a href="/cashtrack/php/pages/login.php" class="btn btn-primary mt-2 btn-lg btn-block">Sign In</a>

</div>

<div class="col-12">

<a href="/cashtrack/php/pages/signup.php" class="btn btn-primary mt-2 btn-lg btn-block">Sign Up</a>

</div>

</div>

</div>

</div>

</div>

</div>

<!--Second-->

<div class="container p-5">

<div class="row">

<div class="col-md-6 col-sm-12">

<p>

<h2>Worried about your spending habits?</h2> <br>

Use ExpenseTracker to keep track of your expenses and control your habits. View graphical representation of your balance to know where you're spending the most. Forget about keeping track on paper!

</p>

</div>

<div class="col-md-6 col-sm-12">

<img src="./assets/illustrations/asset-6.png" class="image" alt="banner">

</div>

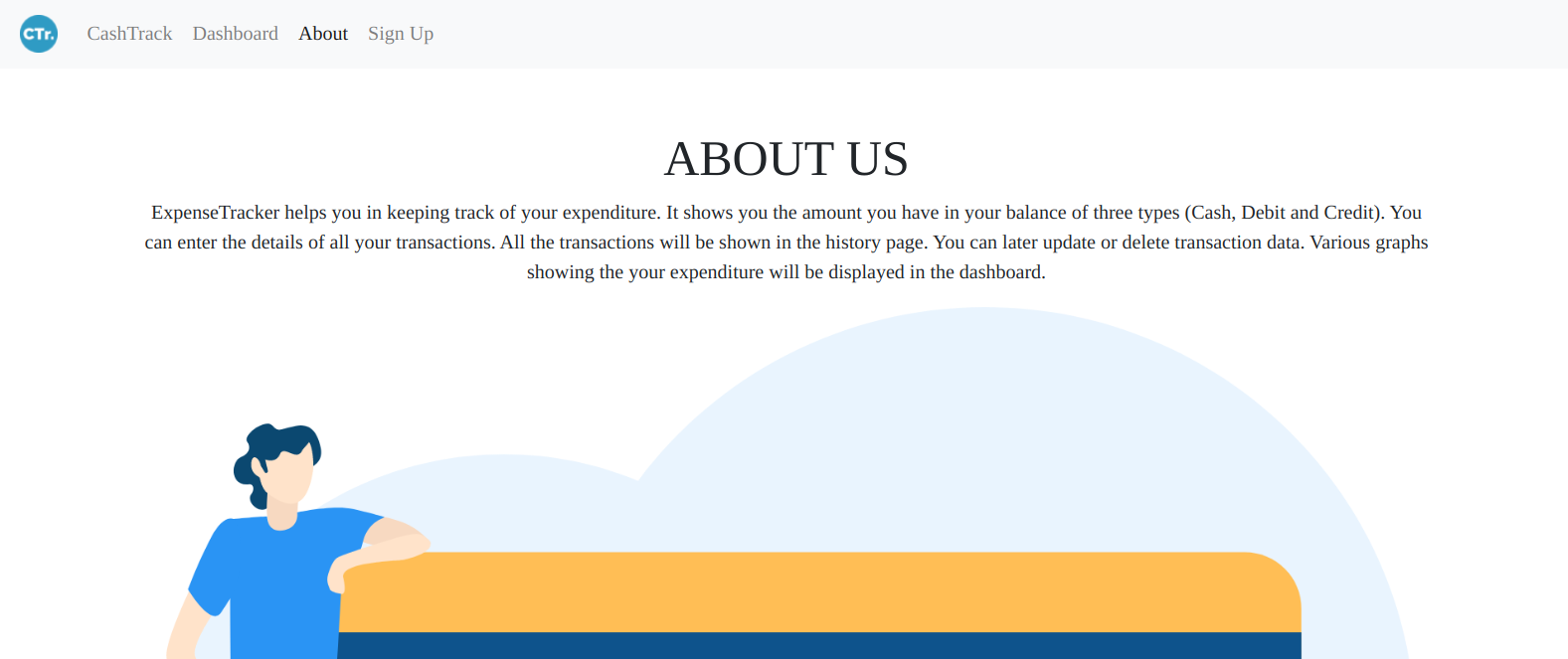
</div>

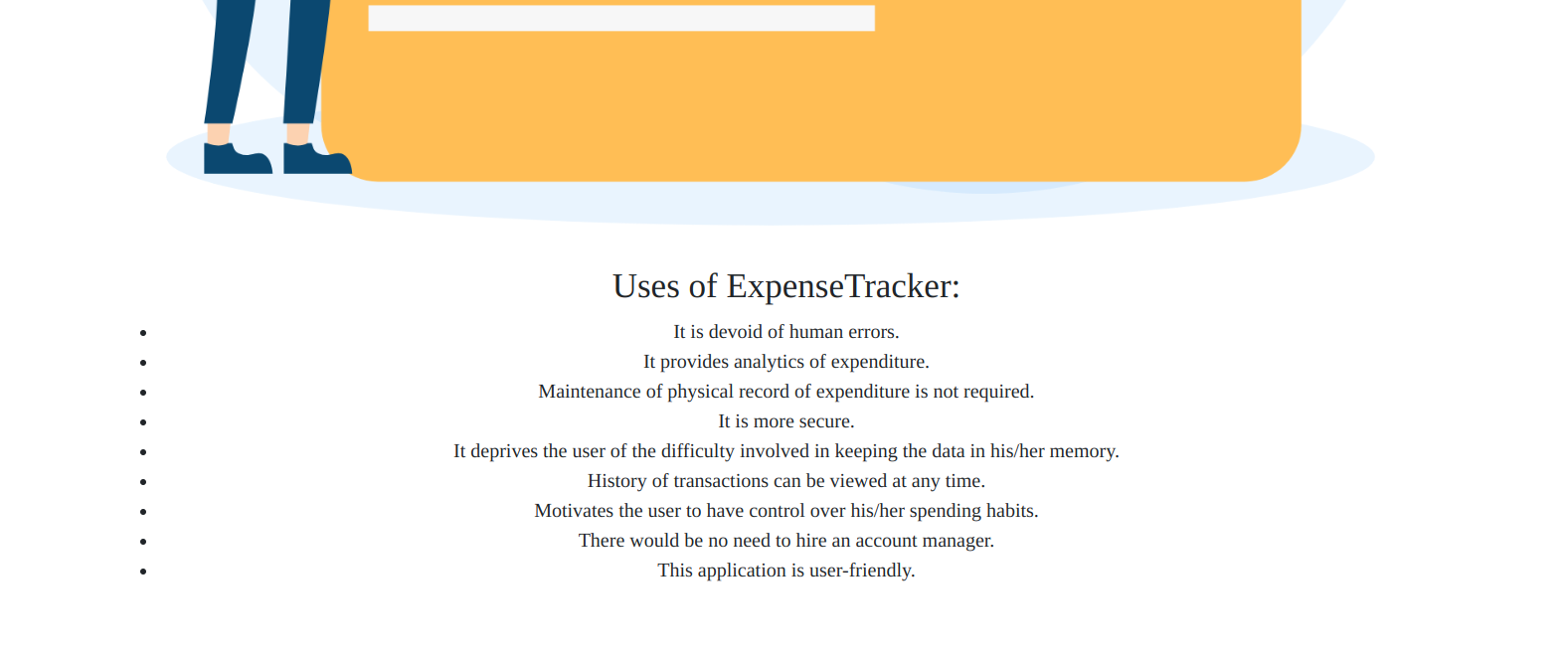
</div>

</body>

</html>

**About page**





<!DOCTYPE html>

<html lang="en">

<head>

<link rel="manifest" href="./assets/favicon/manifest.json">

<meta name="msapplication-TileColor" content="#ffffff">

<meta name="msapplication-TileImage" content="./assets/favicon/ms-icon-144x144.png">

<meta name="theme-color" content="#ffffff">

<!--Other-->

<title> About</title>

<!--CSS-->

<link rel="stylesheet" href="./css/main.css">

<link rel="stylesheet" href="./css/bootstrap.min.css">

<link rel="stylesheet" href="./css/fontawesome.all.min.css">

</head>

<body>

<!--Nav bar-->

<header>

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<a class="navbar-brand" href="./index.html">

<img src="./assets/favicon/android-icon-36x36.png" width="30" height="30" class="d-inline-block align-top" alt="Logo">

</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup"

aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNavAltMarkup">

<div class="navbar-nav">

<a class="nav-item nav-link" href="./index.html">CashTrack</a>

<a class="nav-item nav-link" href="./dashboard/">Dashboard</a>

<a class="nav-item nav-link active" href="">About <span class="sr-only">(current)</span></a>

<a class="nav-item nav-link" href="/cashtrack/php/pages/signup.php">Sign Up</a>

</div>

</div>

</nav>

</header>

<main>

<div class="container p-5">

<div class="text-center">

<h1>ABOUT US</h1>

ExpenseTracker helps you in keeping track of your expenditure.

It shows you the amount you have in your balance of three types (Cash, Debit and Credit).

You can enter the details of all your transactions.

All the transactions will be shown in the history page.

You can later update or delete transaction data.

Various graphs showing the your expenditure will be displayed in the dashboard.

</div>

<div class="p-3">

<img src="./assets/illustrations/asset-5.png" alt="About Image" class="image">

</div>

<div class="p-3 text-center">

<h3>Uses of ExpenseTracker:</h3>

<ul>

<li>It is devoid of human errors.</li>

<li>It provides analytics of expenditure.</li>

<li>Maintenance of physical record of expenditure is not required.</li>

<li>It is more secure.</li>

<li>It deprives the user of the difficulty involved in keeping the data in his/her memory.</li>

<li>History of transactions can be viewed at any time.</li>

<li>Motivates the user to have control over his/her spending habits.</li>

<li>There would be no need to hire an account manager.</li>

<li>This application is user-friendly.</li>

</ul>

</div>

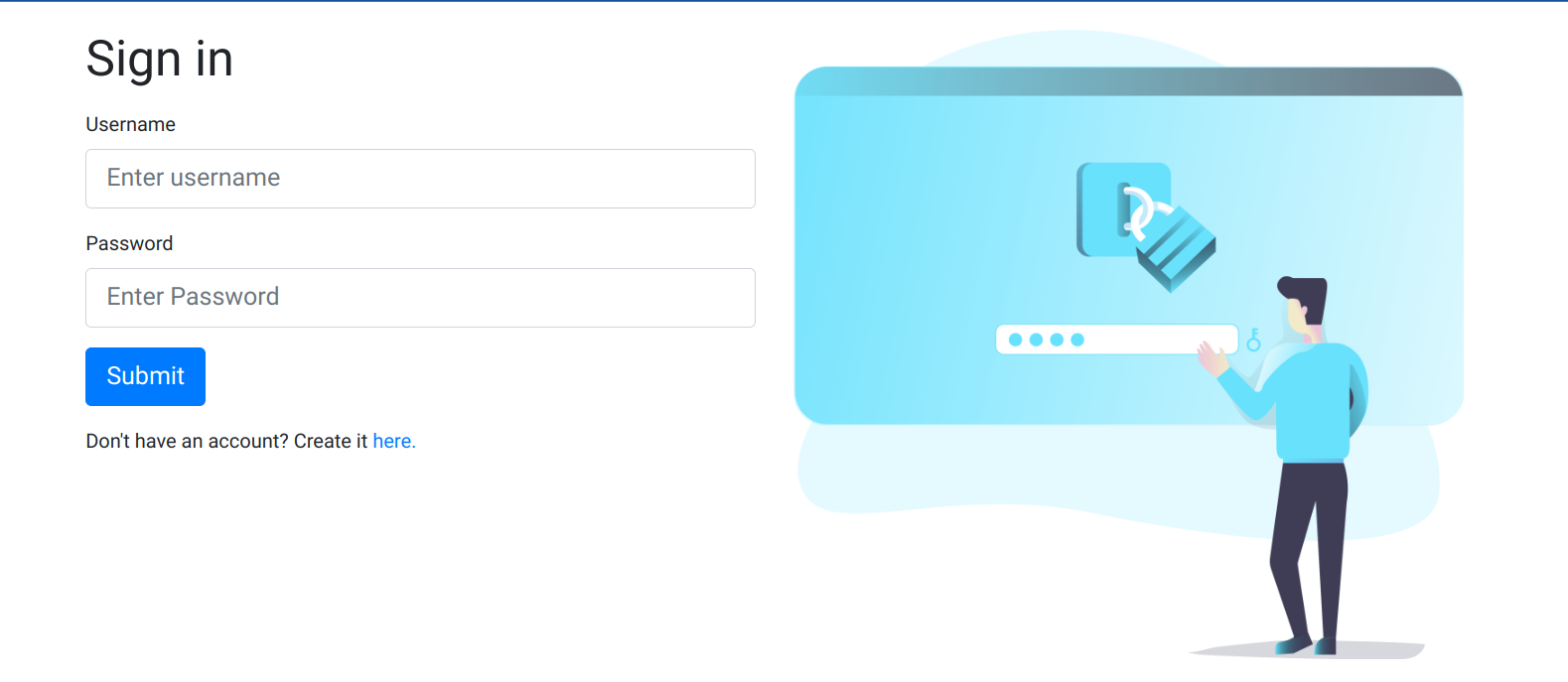
</div>

</main>

</body>

</html>

**Log in page**

****

<!DOCTYPE html>

<html lang="en">

<head>

<!--Meta Tags-->

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!--Favicons-->

<link rel="apple-touch-icon" sizes="57x57" href="../../assets/favicon/apple-icon-57x57.png">

<link rel="apple-touch-icon" sizes="60x60" href="../../assets/favicon/apple-icon-60x60.png">

<link rel="apple-touch-icon" sizes="72x72" href="../../assets/favicon/apple-icon-72x72.png">

<link rel="apple-touch-icon" sizes="76x76" href="../../assets/favicon/apple-icon-76x76.png">

<link rel="apple-touch-icon" sizes="114x114" href="../../assets/favicon/apple-icon-114x114.png">

<link rel="apple-touch-icon" sizes="120x120" href="../../assets/favicon/apple-icon-120x120.png">

<link rel="apple-touch-icon" sizes="144x144" href="../../assets/favicon/apple-icon-144x144.png">

<link rel="apple-touch-icon" sizes="152x152" href="../../assets/favicon/apple-icon-152x152.png">

<link rel="apple-touch-icon" sizes="180x180" href="../../assets/favicon/apple-icon-180x180.png">

<link rel="icon" type="image/png" sizes="192x192" href="../../assets/favicon/android-icon-192x192.png">

<link rel="icon" type="image/png" sizes="32x32" href="../../assets/favicon/favicon-32x32.png">

<link rel="icon" type="image/png" sizes="96x96" href="../../assets/favicon/favicon-96x96.png">

<link rel="icon" type="image/png" sizes="16x16" href="../../assets/favicon/favicon-16x16.png">

<link rel="manifest" href="../../assets/favicon/manifest.json">

<meta name="msapplication-TileColor" content="#ffffff">

<meta name="msapplication-TileImage" content="../../assets/favicon/ms-icon-144x144.png">

<meta name="theme-color" content="#ffffff">

<!--Other-->

<title>Sign in into CashTrack!</title>

<!--CSS-->

<link rel="stylesheet" href="../../css/main.css">

<link rel="stylesheet" href="../../css/bootstrap.min.css">

<link rel="stylesheet" href="../../css/fontawesome.all.min.css">

</head>

<body>

<main>

<div class="center-page container">

<div class="row">

<div class="col-sm-12 col-md-6">

<h1 class="mb-3 mt-3">Sign in</h1>

<form action="login.php" method="POST" onsubmit="return validateLoginForm()">

<div class="form-group">

<label for="username">Username</label>

<input type="text" class="form-control form-control-lg" id="username" name="username" placeholder="Enter username" required>

</div>

<div class="form-group">

<label for="password">Password</label>

<input type="password" class="form-control form-control-lg" id="password" name="password" placeholder="Enter Password" required>

</div>

<button type="submit" name="submit" class="btn btn-primary btn-lg">Submit</button>

</form>

<p class="pt-3">

Don't have an account? Create it

<a href="./signup.php">here.</a>

</p>

</div>

<div class="col-sm-12 col-md-6">

<img src="../../assets/illustrations/asset-2.png" class="image mb-3 mt-3" alt="banner">

</div>

</div>

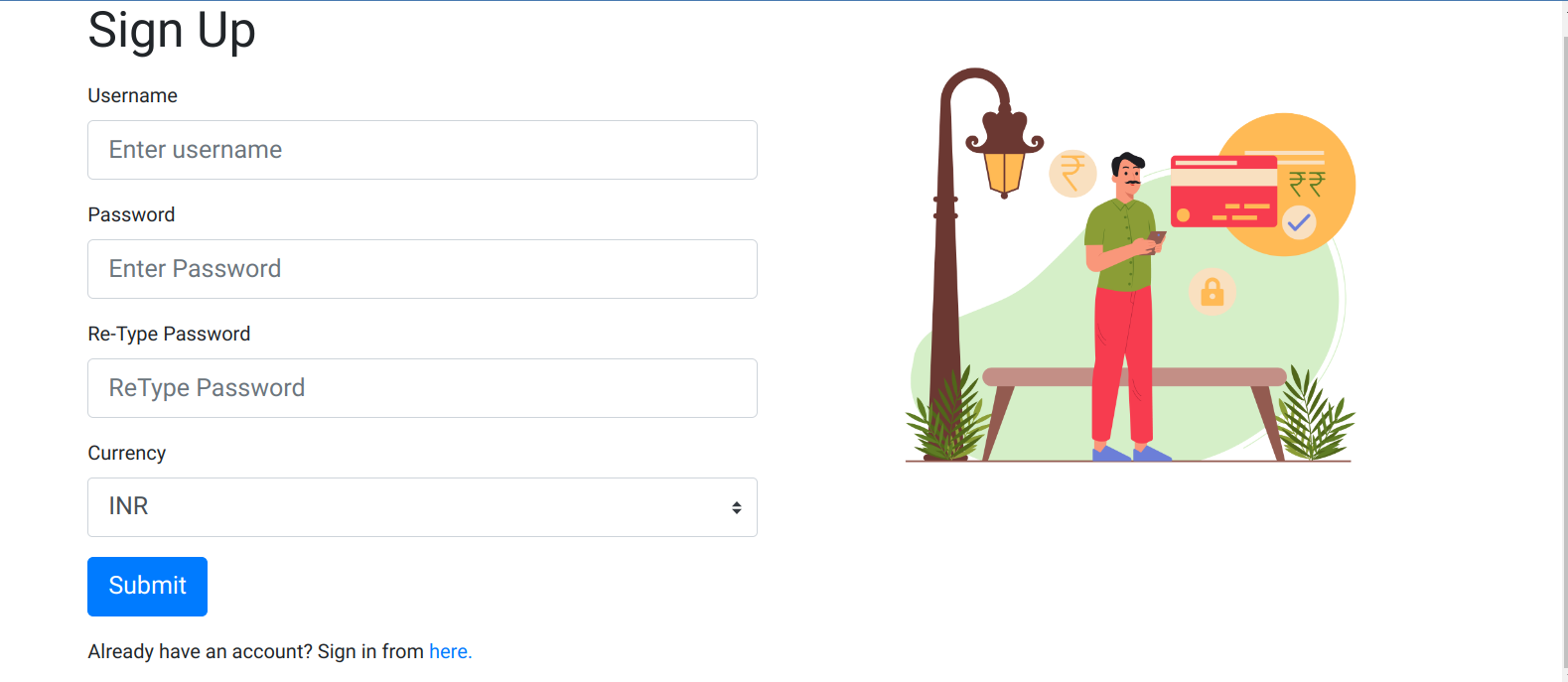
</div>

</main>

</body>

</html>

**Sign-up page**



<!DOCTYPE html>

<html lang="en">

<head>

<meta name="msapplication-TileColor" content="#ffffff">

<meta name="msapplication-TileImage" content="../../assets/favicon/ms-icon-144x144.png">

<meta name="theme-color" content="#ffffff">

<!--Other-->

<title>Sign up to CashTrack!</title>

<!--CSS-->

<link rel="stylesheet" href="../../css/main.css">

<link rel="stylesheet" href="../../css/bootstrap.min.css">

<link rel="stylesheet" href="../../css/fontawesome.all.min.css">

</head>

<body>

<main>

<div class="center-page container">

<div class="row">

<div class="col-sm-12 col-md-6">

<h1 class="mb-3 mt-3">Sign Up</h1>

<form action="signup.php" method="POST" onsubmit="return validateSignUpForm()">

<div class="form-group">

<label for="username">Username</label>

<input type="text" class="form-control form-control-lg" id="username" name="username" placeholder="Enter username" required>

</div>

<div class="form-group">

<label for="password">Password</label>

<input type="password" class="form-control form-control-lg" id="password" name="password" placeholder="Enter Password" required>

</div>

<div class="form-group">

<label for="password">Re-Type Password</label>

<input type="password" class="form-control form-control-lg" id="repassword" name="repassword" placeholder="ReType Password" required>

</div>

<div class="form-group">

<label for="currency">Currency</label>

<select class="custom-select custom-select-lg" name="currency" id="currency">

<option value="1" selected>INR</option>

<option value="2">USD</option>

<option value="3">GBP</option>

<option value="4">EUR</option>

</select>

</div>

<button type="submit" name="submit" class="btn btn-primary btn-lg">Submit</button>

</form>

<p class="pt-3">

Already have an account? Sign in from

<a href="./login.php">here.</a>

</p>

</div>

<div class="col-sm-12 col-md-6">

<img src="../../assets/illustrations/asset-4.png" class="image mb-3 mt-3" alt="banner">

</div>

</div>

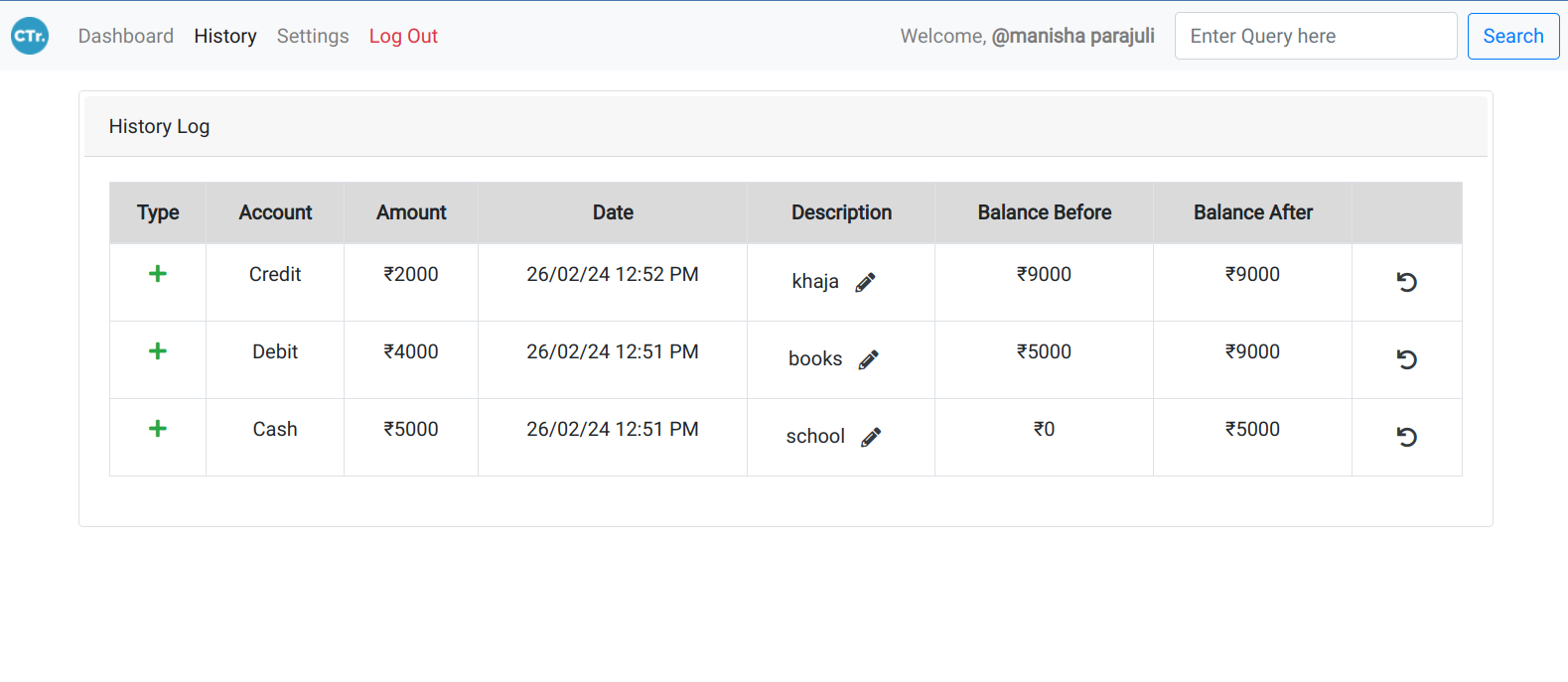
</div>

</main>

</body>

</html>

**History page**

****

<!DOCTYPE html>

<html lang="en">

<head>

</head>

<body>

<!-- Header for settings, history-->

<header>

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<a class="navbar-brand" href="http://localhost/cashtrack/index.html">

<img src="../../assets/favicon/android-icon-36x36.png" width="30" height="30" class="d-inline-block align-top" alt="Logo">

</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNavAltMarkup">

<div class="navbar-nav mr-auto">

<a class="nav-item nav-link" href="../">Dashboard</a>

<a class="nav-item nav-link active" href=".">History <span class="sr-only">(current)</span></a>

<a class="nav-item nav-link" href="../settings">Settings</a>

<a class="nav-item nav-link text-danger" href="../../php/auth/logout.php">Log Out</a>

</div>

<span class="navbar-text ml-auto mr-3">

Welcome,

<span class="font-weight-bold">

<?php echo '@'.$\_SESSION['username']; ?>

</span>

</span>

</div>

</nav>

</header>

<main class="p-3">

<div class="card container p-1">

<div class="card-header">History Log</div>

<div class="card-body">

<div class="table-responsive">

<table class="table table-bordered table-hover text-center">

<thead>

<tr class="table-active">

<th scope="col">Type</th>

<th scope="col">Account</th>

<th scope="col">Amount</th>

<th scope="col">Date</th>

<th scope="col">Description</th>

<th scope="col">Balance Before</th>

<th scope="col">Balance After</th>

<th scope="col"></th>

</tr>

</thead>

<tbody>

<?php

$username = $\_SESSION['username'];

$query = get\_user\_currency($conn,$username);

$result = mysqli\_query($conn,$query);

if(!$result) {

show\_alert("Database error!");

} else {

$row = mysqli\_fetch\_assoc($result);

switch ($row['currency\_default']) {

case 1:

$currency = "₹";

break;

case 2:

$currency = "$";

break;

case 3:

$currency = "£";

break;

default:

$currency = "€";

}

}

if(isset($\_GET['query'])){

$search\_query = $\_GET['query'];

$query = get\_log\_by\_query($conn,$username,$search\_query);

$result = mysqli\_query($conn,$query);

render\_from\_data($conn,$username,$currency,$result);

} else {

$query = get\_all\_logs($conn,$username);

$result = mysqli\_query($conn,$query);

render\_from\_data($conn,$username,$currency,$result);

}

function render\_from\_data($conn,$username,$currency,$result) {

if(mysqli\_num\_rows($result)==0) {

echo '<tr><td colspan="8" class="text-center">No history available.</td></tr>';

} else {

while($row = mysqli\_fetch\_array($result, MYSQLI\_ASSOC)){

$log\_id = $row['log\_id'];

$type = $row['type'];

$account = $row['account'];

$amount = $row['amount'];

$mysqlDate = $row['log\_date'];

$phpdate = strtotime( $mysqlDate );

$date = date('d/m/y g:i A', $phpdate );

$desc = $row['description'];

$balance\_before = $row['balance\_before'];

$balance\_after = $row['balance\_after'];

echo '<tr>';

echo '<td>';

if($type == 1) {

echo '<i class="fa fa-plus text-success"></i>';

} else if($type == 2){

echo '<i class="fa fa-minus text-danger"></i>';

} else {

echo '<i class="fa fa-exchange-alt text-warning"></i>';

}

echo '</td>';

echo '<td>';

switch($account) {

case 1 :

echo 'Cash';

break;

case 2 :

echo 'Debit';

break;

case 3 :

echo 'Credit';

break;

case 12 :

echo 'Cash to Debit';

break;

case 21 :

echo 'Debit to Cash';

break;

case 13 :

echo 'Cash to Credit';

break;

case 23 :

echo 'Debit to Credit';

break;

}

echo '</td>';

echo '<td>';

echo $currency.$amount;

echo '</td>';

echo '<td>';

echo $date;

echo '</td>';

echo '<td>';

//Update description button form

echo '<form action="../../php/routines/update\_description.php" onsubmit="return getUpdatedDescription('.$log\_id.')" method="post">';

echo $desc;

echo '<input type="hidden" value="'.$desc.'" name="currentDesc" id="currentDesc'.$log\_id.'">';

echo '<input type="hidden" value="'.$log\_id.'" name="logId" id="logId'.$log\_id.'">';

echo '<button type="submit" class="btn"><i class="fa fa-pencil-alt text-dark"></i></button>';

echo '</form>';

echo '</td>';

echo '<td>';

echo $currency.$balance\_before;

echo '</td>';

echo '<td>';

echo $currency.$balance\_after;

echo '</td>';

echo '<td>';

//Undo transaction button form

echo '<form action="../../php/routines/undo\_log.php" onsubmit="return undoTransactionConfirm()" method="post">';

echo '<button type="submit" class="btn"><i class="fa fa-undo text-dark"></i></button>';

echo '<input type="hidden" value="'.$mysqlDate.'" name="logDate" id="logDate">';

echo '</form>';

echo '</td>';

echo '</tr>';

}

}

}

function show\_alert($error) {

echo '<script>';

echo 'alert("Error : '.$error.'");';

echo '</script>';

}

mysqli\_close($conn);

?>

</tbody>

</table>

</div>

</div>

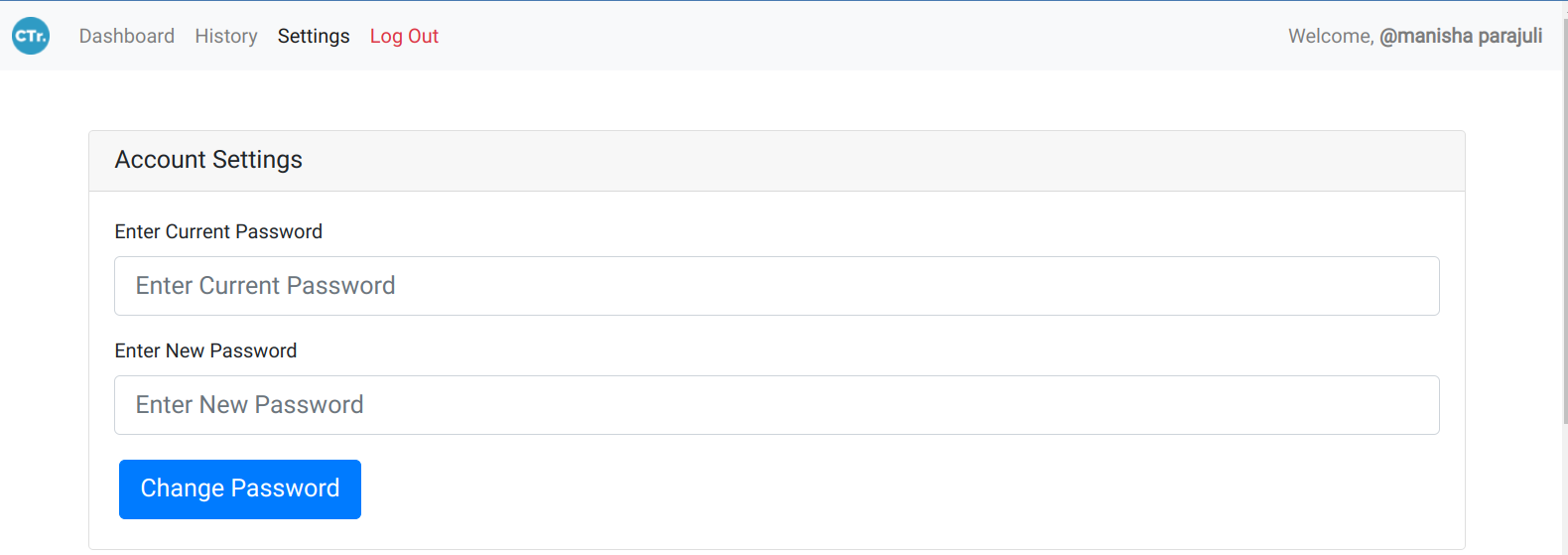
</div>

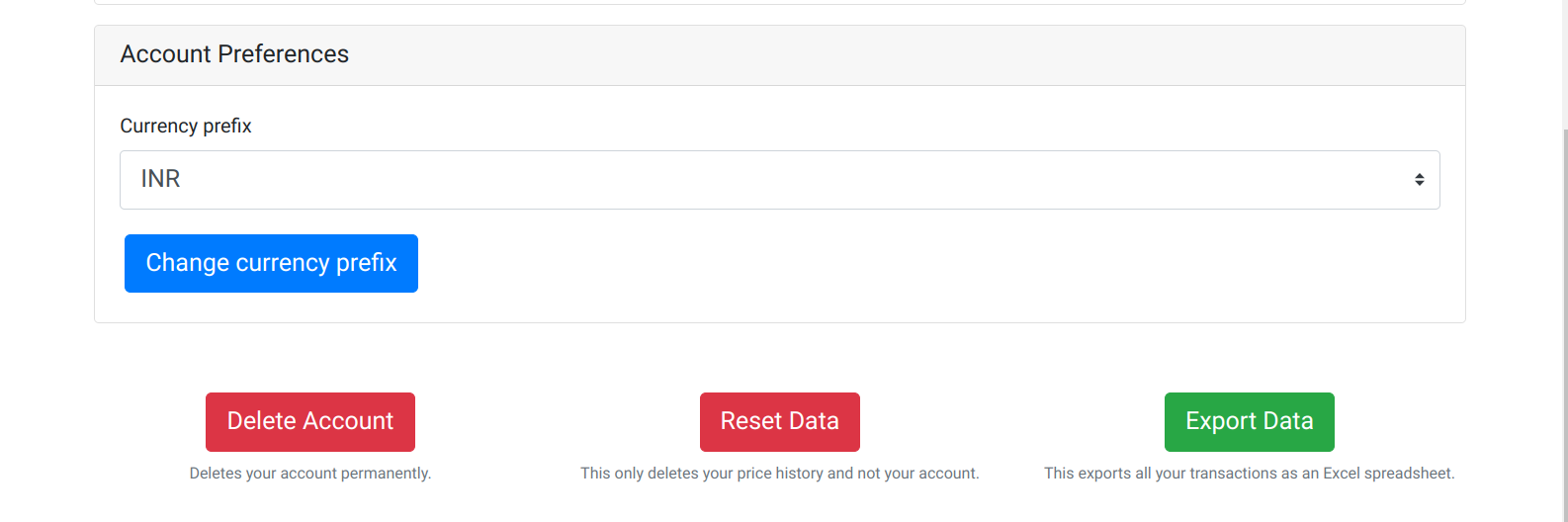
</main>

</body>

</html>

**Setting page**

****

****

<!DOCTYPE html>

<html lang="en">

<head>

</head>

<body>

<!-- Header for settings, history-->

<header>

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<a class="navbar-brand" href="http://localhost/cashtrack/index.html">

<img src="../../assets/favicon/android-icon-36x36.png" width="30" height="30" class="d-inline-block align-top" alt="Logo">

</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNavAltMarkup">

<div class="navbar-nav mr-auto">

<a class="nav-item nav-link" href="../">Dashboard</a>

<a class="nav-item nav-link" href="../history">History</a>

<a class="nav-item nav-link active" href=".">Settings <span class="sr-only">(current)</span></a>

<a class="nav-item nav-link text-danger" href="../../php/auth/logout.php">Log Out</a>

</div>

<span class="navbar-text">

Welcome,

<span class="font-weight-bold">

<?php echo '@'.$\_SESSION['username']; ?>

</span>

</span>

</div>

</nav>

</header>

<main>

<div class="container pt-5">

<div class="card">

<h5 class="card-header">Account Settings</h5>

<div class="card-body">

<form action="../../php/auth/password\_change.php" onsubmit="return validateAccountSettingsForm()" method="post">

<div class="form-group">

<label for="cpassword">Enter Current Password</label>

<input type="password" class="form-control form-control-lg" id="cpassword" name="cpassword" placeholder="Enter Current Password" required>

</div>

<div class="form-group">

<label for="cpassword">Enter New Password</label>

<input type="password" class="form-control form-control-lg" id="npassword" name="npassword" placeholder="Enter New Password" required>

</div>

<button type="submit" class="btn btn-primary btn-lg m-1">Change Password</button>

</form>

</div>

</div>

</div>

<div class="container pt-3">

<div class="card">

<h5 class="card-header">Account Preferences</h5>

<div class="card-body">

<form action="../../php/routines/update\_currency.php" method="post">

<div class="form-group">

<label for="currency">Currency prefix</label>

<select class="custom-select custom-select-lg" name="currency" id="currency">

<?php

switch ($currencyChoice) {

case 1:

echo '<option value="1" selected>INR</option><option value="2">USD</option><option value="3">GBP</option><option value="4">EUR</option>';

break;

case 2:

echo '<option value="1">INR</option><option value="2" selected>USD</option><option value="3">GBP</option><option value="4">EUR</option>';

break;

case 3:

echo '<option value="1">INR</option><option value="2">USD</option><option value="3" selected>GBP</option><option value="4">EUR</option>';

break;

default:

echo '<option value="1">INR</option><option value="2">USD</option><option value="3">GBP</option><option value="4" selected>EUR</option>';

break;

}

?>

</select>

</div>

<button type="submit" class="btn btn-primary btn-lg m-1">Change currency prefix</button>

</form>

</div>

</div>

</div>

<div class="container py-5">

<div class="row mb-auto mt-auto">

<div class=" col-sm-12 col-md-4 text-center p-1">

<button class="btn btn-danger btn-lg m-1" onclick="confirmDeleteAccount()">Delete Account</button><br>

<small class="text-muted">Deletes your account permanently.</small>

</div>

<div class="col-sm-12 col-md-4 text-center p-1">

<button class="btn btn-danger btn-lg m-1" onclick="confirmResetData()">Reset Data</button><br>

<small class="text-muted">This only deletes your price history and not your account.</small>

</div>

<div class="col-sm-12 col-md-4 text-center p-1">

<button class="btn btn-success btn-lg m-1" onclick="exportAsExcel()">Export Data</button><br>

<small class="text-muted">This exports all your transactions as an Excel spreadsheet.</small>

</div>

</div>

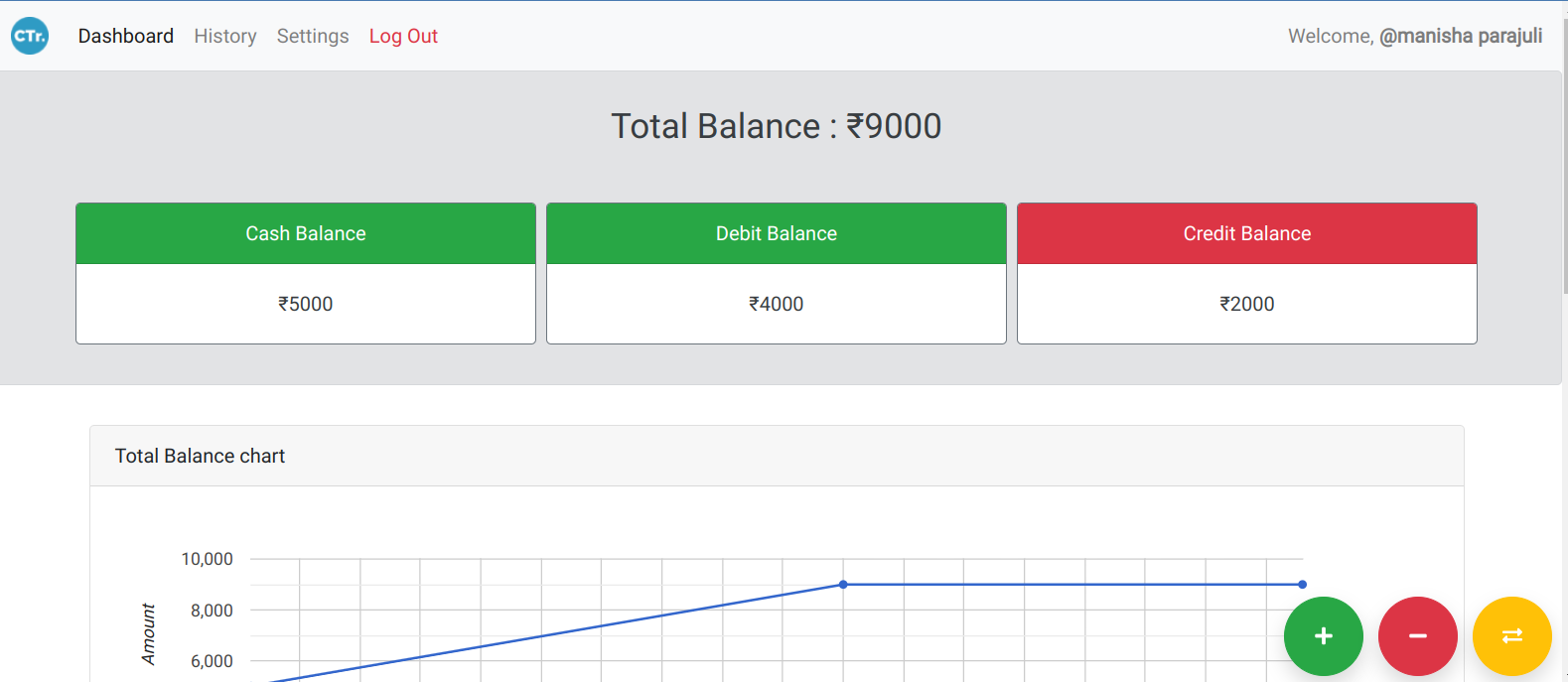
</div>

</main>

</body>

</html>

**Dashboard**



<html lang="en">

<body>

<!-- Header for settings, history-->

<header>

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<a class="navbar-brand" href="http://localhost/cashtrack/index.html">

<img src="../assets/favicon/android-icon-36x36.png" width="30" height="30" class="d-inline-block align-top" alt="Logo">

</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNavAltMarkup">

<div class="navbar-nav mr-auto">

<a class="nav-item nav-link active" href=".">Dashboard <span class="sr-only">(current)</span></a>

<a class="nav-item nav-link" href="./history">History</a>

<a class="nav-item nav-link" href="./settings">Settings</a>

<a class="nav-item nav-link text-danger" href="../php/auth/logout.php">Log Out</a>

</div>

<span class="navbar-text">

Welcome,

<span class="font-weight-bold">

<?php echo '@'.$\_SESSION['username']; ?>

</span>

</span>

</div>

</nav>

</header>

<!--Main page-->

<main>

<div class="alert alert-secondary">

<div class="text-center p-3 h3">

Total Balance :

</div>

<!-- Setting balances with currency-->

<div class="container p-3">

<div class="row">

<div class="col-lg-4 col-md-6 col-sm-12 p-1">

<div class="card text-center border-secondary">

<div class="card-header text-white bg-success">Cash Balance</div>

<div class="card-body">

<?php echo $currency.$cash\_bal; ?>

</div>

</div>

</div>

<div class="col-lg-4 col-md-6 col-sm-12 p-1">

<div class="card text-center border-secondary">

<div class="card-header text-white bg-success">Debit Balance</div>

<div class="card-body">

<?php echo $currency.$debit\_bal; ?>

</div>

</div>

</div>

<div class="col-lg-4 col-md-6 col-sm-12 p-1">

<div class="card text-center border-secondary">

<div class="card-header text-white bg-danger">Credit Balance</div>

<div class="card-body">

<?php echo $currency.$credit\_bal; ?>

</div>

</div>

</div>

</div>

</div>

</div>

<div class="container p-3">

<div class="card">

<div class="card-header">Total Balance chart</div>

<div class="card-body">

<div id="div\_balance\_line\_chart"></div>

</div>

</div>

</div>

<div class="container">

<div class="row">

<div class="col-sm-12 col-md-6 p-3">

<div class="card">

<div class="card-header">Expenditure</div>

<div class="card-body">

<div id="div\_expenditure\_pie\_chart"></div>

</div>

</div>

</div>

<div class="col-sm-12 col-md-6 p-3">

<div class="card">

<div class="card-header">Balance Breakdown</div>

<div class="card-body">

<div id="div\_balance\_breakdown"></div>

</div>

</div>

</div>

</div>

</div>

<div class="container p-3">

<div class="card">

<div class="card-header">Common Descriptions</div>

<div class="card-body">

<table class="table table-bordered text-center text-capitalize">

<thead>

<tr class="table-primary">

<th scope="col">Description</th>

<th scope="col">Count</th>

</tr>

</thead>

</table>

</div>

</div>

</div>

</main>

<!--Footer holding add and subtract buttons-->

<div class="p-1" style="position:fixed;right:0;bottom:0;">

<button type="button" class="btn btn-success shadow rounded-circle m-1" style="width:64px;height:64px;" data-toggle="modal" data-target="#addMoney">

<i class="fa fa-plus"></i>

</button>

<button type="button" class="btn btn-danger shadow rounded-circle m-1" style="width:64px;height:64px;" data-toggle="modal" data-target="#subMoney">

<i class="fa fa-minus"></i>

</button>

<button type="button" class="btn btn-warning shadow rounded-circle m-1" style="width:64px;height:64px;" data-toggle="modal" data-target="#transferMoney">

<i class="fa fa-exchange-alt text-white"></i>

</button>

</div>

<!--Modal dialogs for adding and subtracting money and transfer money-->

<div class="modal fade" id="addMoney" tabindex="-1" role="dialog" aria-labelledby="addMoneyCenterTitle" aria-hidden="true">

<div class="modal-dialog modal-dialog-centered" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="addMoneyLongTitle">Add to balance</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

<form action="../php/routines/add\_money.php" method="post" onsubmit="return validateAddAmount()" id="addMoneyForm">

<div class="form-group">

<label for="addAmount">Amount</label>

<input type="number" class="form-control form-control-lg" id="addAmount" name="amount" placeholder="Amount in Transaction" required>

</div>

<div class="form-group">

<label for="addDesc">Description</label>

<input type="text" class="form-control form-control-lg" id="addDesc" name="desc" placeholder="Description" required>

</div>

<input type="hidden" value="<?php echo $cash\_bal+$debit\_bal; ?>" name="balance">

<div class="form-group">

<label for="addAccount">Account</label>

<select class="custom-select custom-select-lg" name="account" id="addAccount">

<option value="1" selected>Cash</option>

<option value="2">Debit</option>

<option value="3">Credit</option>

</select>

</div>

<div class="add-credit-alert collapse alert alert-warning">

Warning : Adding to credit means that you're adding to your credit card due.

</div>

</form>

</div>

<div class="modal-footer">

<button type="submit" form="addMoneyForm" class="btn btn-primary px-3">Add</button>

</div>

</div>

</div>

</div>

<div class="modal fade" id="subMoney" tabindex="-1" role="dialog" aria-labelledby="subMoneyCenterTitle" aria-hidden="true">

<div class="modal-dialog modal-dialog-centered" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="subMoneyLongTitle">Subtract from balance</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

<form action="../php/routines/sub\_money.php" method="post" onsubmit="return validateSubAmount()" id="subMoneyForm">

<div class="form-group">

<label for="subAmount">Amount</label>

<input type="number" class="form-control form-control-lg" id="subAmount" name="amount" placeholder="Amount in Transaction" required>

</div>

<div class="form-group">

<label for="subDesc">Description</label>

<input type="text" class="form-control form-control-lg" id="subDesc" name="desc" placeholder="Description" required>

</div>

<input type="hidden" value="<?php echo $cash\_bal+$debit\_bal; ?>" name="balance">

<div class="form-group">

<label for="account">Account</label>

<select class="custom-select custom-select-lg" name="account" id="subAccount">

<option value="1" selected>Cash</option>

<option value="2">Debit</option>

<option value="3">Credit</option>

</select>

</div>

<div class="sub-credit-alert collapse alert alert-warning">

Warning : Subtracting from credit means that you've paid for your credit card due.

</div>

</form>

</div>

<div class="modal-footer">

<button type="submit" form="subMoneyForm" class="btn btn-primary px-3">Subtract</button>

</div>

</div>

</div>

</div>

<div class="modal fade" id="transferMoney" tabindex="-1" role="dialog" aria-labelledby="transferMoneyCenterTitle" aria-hidden="true">

<div class="modal-dialog modal-dialog-centered" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="transferMoneyLongTitle">Transfer balance</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

<form action="../php/routines/transfer\_money.php" method="post" onsubmit="return validateTransferAmount()" id="transferMoneyForm">

<div class="form-group">

<label for="transferAmount">Amount</label>

<input type="number" class="form-control form-control-lg" id="transferAmount" name="amount" placeholder="Amount in Transaction" required>

</div>

<div class="form-group">

<label for="transferDesc">Description</label>

<input type="text" class="form-control form-control-lg" id="transferDesc" name="desc" placeholder="Description" required>

</div>

<input type="hidden" value="<?php echo $cash\_bal+$debit\_bal; ?>" name="balance">

<div class="form-group">

<label for="transferFAccount">From Account</label>

<select class="custom-select custom-select-lg" name="faccount" id="transferFAccount">

<option value="1" selected>Cash</option>

<option value="2">Debit</option>

</select>

</div>

<div class="form-group">

<label for="transferTAccount">To Account</label>

<select class="custom-select custom-select-lg" name="taccount" id="transferTAccount">

<option value="1" selected>Cash</option>

<option value="2">Debit</option>

<option value="3">Credit</option>

</select>

</div>

<div class="transfer-credit-alert collapse alert alert-warning">

Warning : Transferring to credit means that you're paying back the debt.

</div>

</form>

</div>

<div class="modal-footer">

<button type="submit" form="transferMoneyForm" class="btn btn-primary px-3">Transfer</button>

</div>

</div>

</div>

</div>

</body>

</html>

# TESTING

Testing is crucial for ensuring the quality and reliability of products. It provides information on the different levels of testing that our project undergoes. They include:

## **9.1 Unit testing**

Unit testing is the process of testing individual units or components of a software application in isolation. A unit is the smallest testable part of the software, such as a function, method, or class.

## **9.2 Integration testing**

Integration testing is the process of testing the interactions and interfaces between different units or components of a software application. It focuses on ensuring that these components work together seamlessly. It ensures that data flows correctly between components and that they cooperate as expected.

**9.3 System testing**

System testing is the process of evaluating a complete, integrated software system to ensure it meets specified requirements and functions as expected in its intended environment.

# CONCLUSIONS

An expense tracker web application can provide numerous benefits for organizations, including improved efficiency, accuracy, compliance, cost savings, and employee satisfaction. By implementing such a system, companies can streamline their expense reporting process, reduce errors, and maintain compliance with financial regulations and standards.

In conclusions, this project succeeded in creating a user-friendly, reliable system that meets all its goals, performs well, is secure and can grow in the future. We documented everything carefully and good teamwork helped a lot.

# REFERENCE

1. <https://www.itilite.com/blog/expense-tracking-system/>

2. <https://fincent.com/glossary/expense-tracking>

3. <https://www.indeed.com/career-advice/career-development/project-expense-tracking>