



# TRACKIT

*A Minor project submitted for the partial fulfillment of the requirements  
for the degree of*

**BACHELOR OF COMPUTER APPLICATION**

*Submitted By*

Xxxxxxx Xxxxxxxxxxx

**Roll No: 15201XXXXXX Session : 2009-2010**

**Registration No.: 16152XXXXXXXX Of: XXXX-XX**



Techno India (Hooghly Campus)  
Dharampur, Shantiniketan, Near Khadina more  
G.T. Road, Chinsurah, Hooghly-712101  
Phone: 033-2680-2389

Website: [www.technoindiahooghly.org](http://www.technoindiahooghly.org) Email: [info@technoindiahooghly.org](mailto:info@technoindiahooghly.org)



## Techno India Hooghly - Campus

Dharampur, Shantiniketan, Near Khadinamore  
G. T. Road, Chinsurah, Hooghly - 712101  
Phone: (033) 2680-2389 / 6565  
E-mail: info@technoindiahooghly.org  
Website: www.technoindiahooghly.org

### ***Certificate***

*This is to certify that the project entitled “**TRACKIT**” submitted by **xxxxxx**  
**xxxxxx**, University Roll No: **999909999999**, as Assigned Project (minor)  
for the partial fulfillment of degree of Bachelor of Computer Application is  
worth of acceptance.*

---

**(Mr. Abhijit Ghosh)**

Asst. Professor

Techno India (Hooghly Campus)

---

**(Dr. Soumya Mukherjee)**

Principal

Techno India (Hooghly Campus)

---

**(Mr. Tapan Kumar Chakraborty)**

Head of the Department, BCA

Techno India (Hooghly Campus)

## **Acknowledgement**

It is our pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced our thinking, behaviour, and acts during the course of study.

We express sincere gratitude to our respected **Principal Sir** for providing us an opportunity to undergo this project at **TECHNO INDIA (HOOGHLY CAMPUS)**.

We are also thankful to **Abhijit Ghosh Sir** for their support, cooperation, and motivation provided to us during the training for constant inspiration, presence and blessings.

Lastly, we would like to thank the almighty and our parents for their moral support and our friends with whom we shared our day-to-day experience and received lots of suggestions that improved our quality of work.

**Name of the Students:**

**Signature of the Students:**

**Sakshi Pandey**

**Rohit Gupta**

**Anjali Sharma**

**Aryan Mishra**

**Sajal Chitlangia**

# Table of Contents

1. Abstract .....	6
2. Introduction.....	7
3. Objective and Scope of the Project.....	8
4. Theoretical Background.....	9
5. Problem Definition.....	10
6. Solution Approach .....	11
7. Feasibility Study.....	12-13
a. Why is feasibility study required?.....	12
b. Technical Feasibility .....	12
c. Economic Feasibility.....	12
d. Operational Feasibility.....	12
e. Social Feasibility .....	13
8. System Planning.....	14-17
a. Software Development Life Cycle (SDLC).....	14
b. Gantt Chart .....	16
c. Pert Chart.....	17
9. Software Requirements Specification (SRS) .....	18-20
a. Purpose .....	18
b. Scope .....	18
c. Product Perspective.....	18
d. Product Functions.....	18
e. Functional Requirements .....	19
f. Non-Functional Requirements .....	19
g. Software and Hardware Requirements .....	20
10. Software Design Specification .....	21-28
a. ER Diagram.....	21
b. Level 0 DFD.....	22
c. Level 1 DFD.....	23
d. Use Case Diagram.....	24
e. Activity Diagram.....	25
f. Sequence Diagram .....	26
g. Class Diagram .....	27
h. Database Schema .....	28

<b>11. Cost Analysis.....</b>	<b>29-30</b>
<b>12. Testing &amp; Analysis .....</b>	<b>31-41</b>
a. Unit Testing .....	31
b. Integration Testing.....	31
c. System Testing.....	31
d. Acceptance Testing.....	31
e. Recovery Testing.....	31
f. Functional Testing .....	31
g. Hardware/Software Testing.....	31
h. Security Testing .....	31
i. Unit Test Cases and Results.....	32
<b>13. User Manual .....</b>	<b>42-46</b>
<b>14. Conclusion, References and Limitations .....</b>	<b>47-48</b>
a. Conclusion.....	47
b. References .....	47
c. Limitations .....	48
d. Future Enhancements.....	48

## Abstract

**TrackIt** is a sophisticated and comprehensive financial management application crafted to streamline the tracking of both income and expenses for individuals and businesses alike. This cutting-edge tool is designed with user-friendliness in mind, offering an intuitive interface that simplifies the process of inputting and categorizing financial transactions. With **TrackIt**, users gain a clear and detailed overview of their financial health, facilitating more informed decision-making and enhanced financial stability.

**TrackIt** excels in providing real-time updates on account balances, ensuring that users always have the most current view of their financial standing. This feature is essential for maintaining an accurate and up-to-date understanding of one's finances. Alongside this, the app offers detailed reports on spending patterns. These reports delve into various aspects of financial behavior, helping users gain insight into their expenditure habits and making it easier to identify areas where adjustments can be made.

**TrackIt** offers real-time balance updates and detailed reports on spending patterns, helping users understand their financial behavior and make informed decisions. The app includes budget planning tools and customizable alerts to prevent overspending, while also managing daily, weekly, monthly, and yearly expenses with ease.

Users can categorize expenses, upload images of bills, add location details, and specify amounts. This data is securely stored, allowing users to view and sort expenses by different timeframes. The app also supports sharing bills and managing group expenses, and displays financial data through interactive graphs.

With features like income tracking, expense limits, and automatic calculations, **TrackIt** helps reduce manual work and provides a streamlined approach to financial management. By consolidating all financial information in one place, **TrackIt** empowers users to enhance their savings strategies and achieve financial stability efficiently.

## Introduction

In this project we propose a website known as “**TrackIt**” which will help to manage our income and expenses. It also acts as an indicator or reminder to remind the users of updating their expenses and paying their bills for the month. Due to some conflict or some other stress, we sometimes forget about the payments and bills that we have to pay. This site will help you to make a note for what are the things we have to pay for by the end of month and/or year like gas bills, phone bill, electricity bill, taxes and some other personal expenses. In this fast-moving world this mobile application and website will be very useful for people who have a family and especially for business persons. Budgeting is an integral part of the society. Budget Tracking involves recording and analyzing the incomes and expenses of a person or an organization over a particular period of time. Today, since we are living in a fast-paced society, many people are looking forward to efficient ways to budget their time and money. During recent years, some research has been carried out on household budget. It has been noted that in most cases, budget management is being done mentally and never being put on paper which makes Budget Tracking very difficult. Tracking income and expenses is essential for managing personal finances or running a business. Here’s an introduction to help you get started



# Objective and Scope of the Project

## Objective

The primary objective of the "TrackIt" project is to develop a comprehensive personal finance management tool. This tool aims to help users efficiently track their income, expenses, and budget, offering clear insights into their financial status. By doing so, the project seeks to empower users to make informed financial decisions, achieve their financial goals, and maintain overall financial health.

## Scope

The scope of the "TrackIt" project encompasses several key features and functionalities:

### 1. Income and Expense Tracking:

- **Detailed Recording:** Users can record all their income sources, such as salaries, freelance earnings, and investments, in a structured manner.
- **Expense Categorization:** Users can categorize expenses into predefined categories like bills, groceries, entertainment, and more, allowing for better analysis and control.

### 2. Budget Management:

- **Budget Creation:** Users can create monthly, quarterly, or annual budgets tailored to their financial goals and spending habits.
- **Monitoring and Adjustment:** The tool provides real-time updates on budget performance, allowing users to adjust their budgets as needed to stay on track.

### 3. Financial Reports:

- **Comprehensive Reports:** Users can generate detailed reports that summarize income, expenses, and budget performance over specific periods.
- **Visual Representations:** Graphs and charts provide visual insights into financial data, making it easier to understand spending patterns and financial trends.

### 4. Account Management:

- **Multiple Account Support:** Users can manage different types of accounts, such as savings, cash, and investment accounts, within the tool.
- **Account Addition and Deletion:** Users can add new accounts or delete old ones as their financial situation changes.

### 5. User-Friendly Interface:

- **Intuitive Design:** The interface is designed to be user-friendly, ensuring that users with varying levels of financial literacy can navigate and use the tool effectively.

# Theoretical Background

## HTML (Hyper Text Markup Language)

HTML provides the foundation for the "**TrackIt**" web application, structuring content and defining the layout. HTML elements are used to create forms for user input, organize financial data, and display charts and graphs. HTML5 semantic tags, like `<header>`, `<footer>`, `<section>`, and `<article>`, enhance accessibility and SEO, ensuring that the application is both user-friendly and search engine optimized.

## CSS (Cascading Style Sheets)

CSS is used to style the HTML content, enhancing the visual appeal and ensuring a responsive design. With CSS3, advanced features like transitions, animations, and media queries are utilized to create a dynamic and adaptive user interface. The "**TrackIt**" project uses CSS to maintain a consistent look and feel across different devices, ensuring a seamless user experience.

## JavaScript (JS)

JavaScript adds interactivity and dynamic behavior to the "**TrackIt**" application. It handles client-side scripting, enabling real-time updates, form validation, and dynamic content manipulation without requiring a page reload. JavaScript libraries and frameworks, such as jQuery or Chart.js, may be used to create interactive charts and graphs that provide visual insights into the user's financial data.

## PHP (Hypertext Preprocessor)

PHP is the server-side scripting language used in "**TrackIt**" to manage the application's backend. It processes user inputs, interacts with the database, and generates dynamic content. PHP scripts handle tasks such as user authentication, financial data storage and retrieval, and budget calculations. By embedding PHP within HTML, the application can deliver personalized and dynamic web pages based on user interactions.

## Git and GitHub

Version control is managed using Git, a distributed version control system that tracks changes in the source code. GitHub, a platform for hosting Git repositories, is used for collaborative development. It allows multiple developers to work on the "**TrackIt**" project simultaneously, manage changes, and resolve conflicts efficiently. GitHub also facilitates issue tracking, code reviews, and continuous integration, ensuring that the project maintains high quality and reliability.

## Integration of Technologies

In "**TrackIt**", HTML structures the content, CSS styles it, JavaScript adds interactivity, and PHP handles server-side logic and database interactions. Git and GitHub ensure efficient version control and collaboration. This combination of technologies creates a robust, user-friendly web application for comprehensive financial tracking, budget planning and data visualization.

## Problem Definition

Many individuals face challenges in managing their finances effectively due to a variety of factors. Here are some key issues that contribute to these difficulties:

1. **Lack of Basic Financial Literacy:** Many people do not have a strong foundation in financial literacy. This means they may not understand fundamental concepts such as budgeting, saving, investing, or the impact of interest rates on debt. Without this knowledge, it's challenging to make informed financial decisions, leading to mismanagement of funds. This lack of understanding can result in poor financial habits, making it difficult to achieve long-term financial stability and goals.
2. **Inadequate Tracking of Expenses:** Without proper tracking mechanisms, individuals can easily lose sight of where their money is going. This often leads to overspending, as people may not realize how much they are spending on non-essential items. Tools like expense tracking apps or maintaining a spending diary can help, but many people do not use these tools consistently.
3. **Difficulty in Creating and Sticking to a Budget:** Budgeting is a fundamental aspect of financial management, but many people struggle with it. Creating a realistic budget requires an understanding of income, expenses, and financial goals. Sticking to a budget requires discipline and often involves making difficult choices about spending. Without a budget, it's easy to spend more than one earns, leading to financial instability.
4. **Insufficient Savings for Emergencies and Future Goals:** A significant number of individuals do not save adequately for emergencies or future goals. Emergency savings are crucial to cover unexpected expenses such as medical emergencies or car repairs. Additionally, saving for long-term goals like buying a house, education, or retirement is essential for financial security. Many people, however, find it challenging to save regularly due to various financial pressures and priorities.
5. **Challenges in Managing and Paying Off Debts:** Debt management is another common challenge. Whether it's credit card debt, student loans, or mortgages, managing and paying off debts can be overwhelming without a clear plan. High-interest rates can quickly compound debt, making it even harder to pay off. Developing a debt repayment strategy, such as the snowball or avalanche method, can help, but it requires knowledge and discipline. Without a proper strategy, debt can spiral out of control, affecting credit scores and limiting future financial opportunities.

## Solution Approach

The "TrackIt" project aims to address these challenges by providing a robust tool that offers:

1. **Comprehensive Tracking:** Users can track all their income sources and expenses in one place, allowing for a detailed overview of their financial status. This holistic tracking helps identify areas where spending can be optimized.
2. **Budget Planning:** The tool helps users create realistic budgets based on their financial goals and track their progress over time. It also provides alerts for overspending, ensuring users stay within their planned budgets.
3. **Visual Insights:** Graphs and charts provide clear visual representations of financial data, making it easier to understand spending patterns. These insights empower users to make informed decisions about their finances quickly.
4. **Multiple Accounts Management:** Users can manage different accounts, including savings, cash, and investments, to monitor their overall financial health. This feature allows for easy tracking and comparison of various financial assets in one unified platform.
5. **User-Friendly Design:** The intuitive interface ensures users can easily navigate and use the tool without extensive financial knowledge. It includes guided prompts and tips to help users make the most out of their financial planning journey

# **Feasibility Study**

## **Why is Feasibility Study required?**

This comprehensive feasibility study assesses the viability and practicality of 'TrackIt', an online finance tracking website. Our in-depth analysis evaluates technical, economic, operational, and social aspects to determine whether the project is worth pursuing. We examine the project's potential benefits, risks, and challenges to ensure a well-informed decision-making process.

### **Technical Feasibility:**

Technical feasibility assesses a project's practicality from a technical perspective. It evaluates compatibility with existing systems, available tools, technical expertise, scalability, and performance. A technically feasible solution can be implemented, maintained, and supported within given constraints, ensuring a stable and efficient outcome, and minimizing technical risks and challenges.

- The feasibility to produce outputs in a given time.
- Response time under certain conditions.
- Ability to process a certain volume of the transaction at a particular speed.
- Facility to communicate data.

### **Economic Feasibility:**

Economic feasibility assesses a project's financial practicality, evaluating costs versus benefits. It considers initial investment, operating costs, revenue, return on investment (ROI), and break-even point. A project is economically feasible if it generates sufficient financial returns to justify investment, ensuring a positive impact on the organization's financial health.

- Cost of Hardware and Software.
- Cost of Software to be acquired to build and run the product is a onetime cost.
- Database is the major part of hardware and software cost.
- Cost saving decision is needed.

### **Operational Feasibility:**

Operational feasibility assesses a project's ability to be implemented and integrated into existing operations. It evaluates the impact on workflows, processes, and resources, considering factors such as staffing, training, and infrastructure. A project is operationally feasible if it can be successfully executed and sustained within the organization's existing framework.

- If there is sufficient support for the management from the user?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

## **Social Feasibility:**

Social feasibility assesses a project's acceptance and impact on stakeholders, including users, customers, and communities. It evaluates factors such as social norms, cultural values, and potential resistance to change. A project is socially feasible if it aligns with stakeholder needs and values, ensuring adoption and minimizing negative social consequences.

- **Team Dynamics:** Establish clear roles, respect, and open communication to foster a positive, productive, and inclusive team environment.
- **Shared Goals:** Unite under a common objective, motivating each other to ensure collective success and a sense of accomplishment.
- **Learning Environment:** Encourage experimentation, feedback, and growth, promoting continuous learning, improvement, and innovation.
- **Social Impact:** Assess the project's potential social impact, ensuring it aligns with societal values, needs, and expectations.

# System Planning

## Software Development Life Cycle (SDLC)

The project will follow the Iterative Waterfall Model, a variant of the traditional Waterfall model that incorporates iterative cycles to allow for refinement and improvements based on feedback and evolving requirements. This model is structured into sequential phases, with each phase providing a distinct focus while allowing for iterative development and testing. Here's how the iterative waterfall model will be applied:

### 1. Feasibility Study

- **Objective:** Evaluate the feasibility of the project in terms of technical, financial, and operational aspects before proceeding to the development phase.
- **Iteration Aspect:** Feasibility findings may be revisited and updated throughout the project to accommodate any changes in scope or constraints.

### 2. Requirements Gathering and Analysis

- **Objective:** Collect and analyze all the requirements for the project. This involves understanding user needs, business requirements, and any constraints.
- **Iteration Aspect:** Requirements may be revisited and refined through iterations to ensure accuracy and completeness.

### 3. System Design

- **Objective:** Develop the system architecture and design based on the gathered requirements. This includes both high-level and detailed design.
- **Iteration Aspect:** Design may be iteratively updated based on feedback and new insights gained during the development phase.

### 4. Implementation

- **Objective:** Code the system according to the design specifications. This phase involves developing the software components and integrating them.
- **Iteration Aspect:** Iterative cycles involve implementing features, then reviewing and revising them based on testing and feedback.

### 5. Testing

- **Objective:** Test the system to ensure that it meets the specified requirements and is free of defects. This phase includes unit testing, integration testing, and system testing.
- **Iteration Aspect:** Testing is performed iteratively, with each iteration focusing on different aspects of the system. Feedback from testing leads to revisions and improvements.

### 6. Deployment

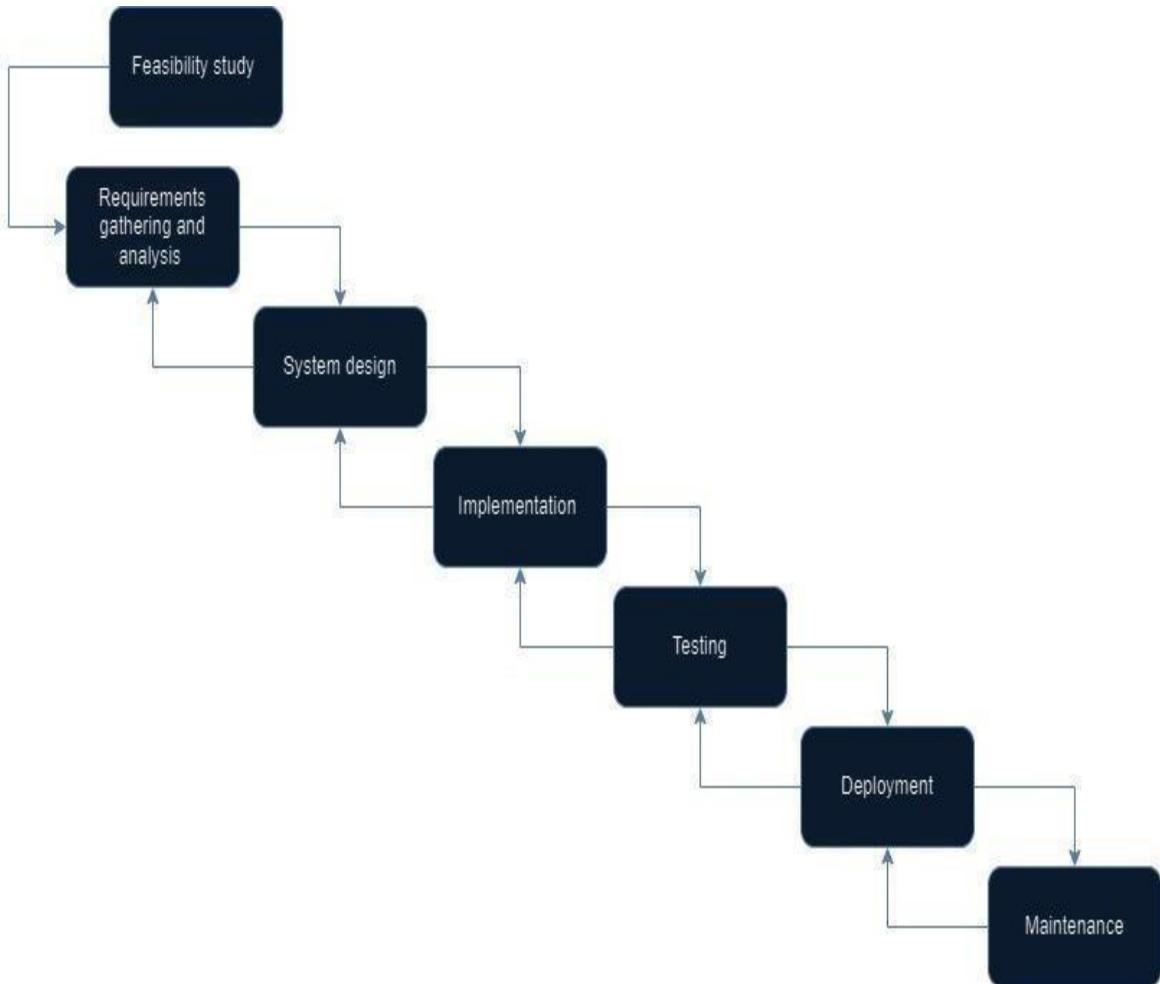
- **Objective:** Deploy the system to the production environment, making it available to end-users.
- **Iteration Aspect:** Deployment may occur in stages, with iterative releases to gather user feedback and make incremental improvements.

### 7. Maintenance

- **Objective:** Provide ongoing support and maintenance to address issues, perform updates, and ensure the system continues to operate effectively.
- **Iteration Aspect:** Maintenance includes iterative updates and enhancements based on user

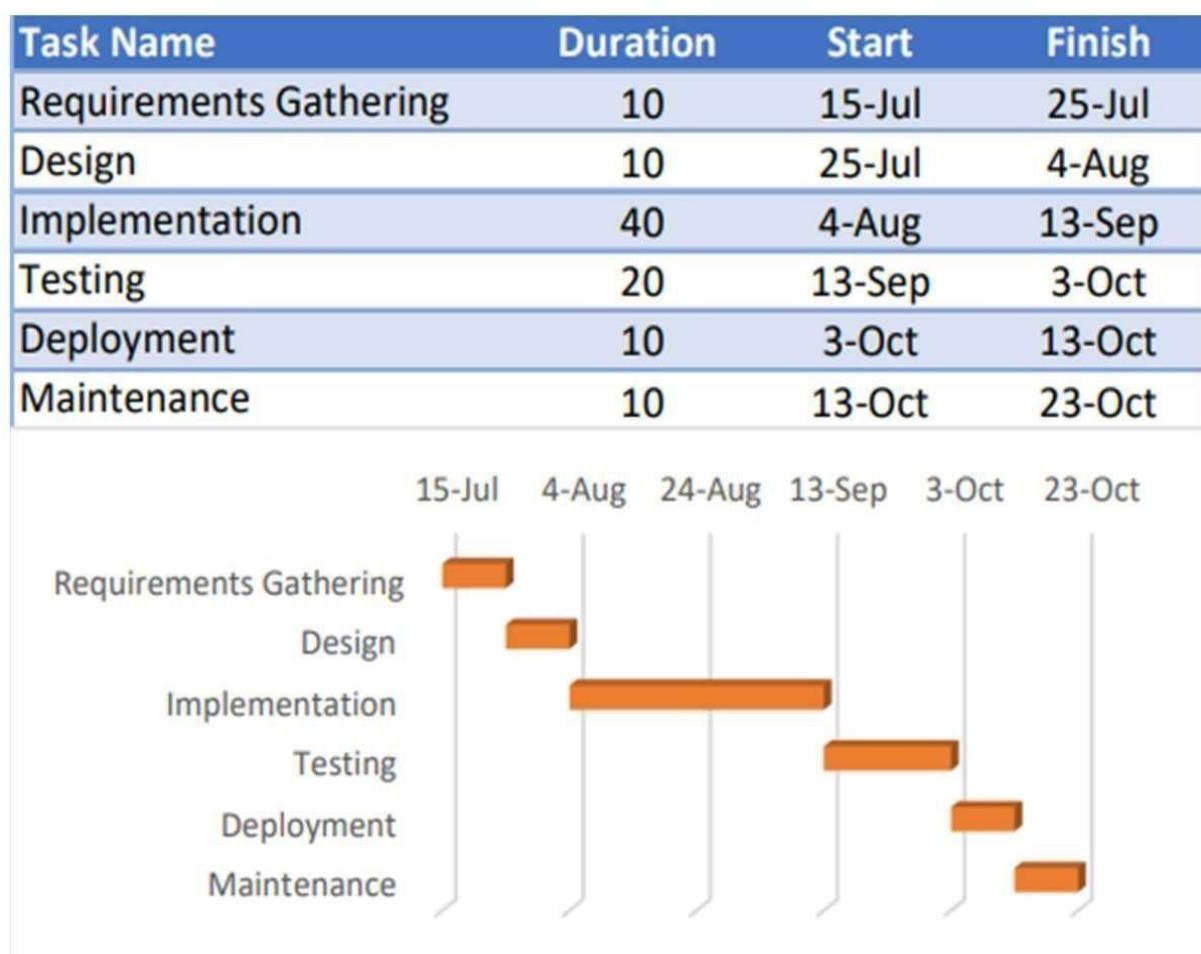
feedback and changing requirements.

The iterative nature of this model allows for flexibility and responsiveness to changes, making it suitable for projects where requirements may evolve or become clearer over time. Each iteration provides an opportunity to refine the project, ensuring better alignment with user needs and expectations



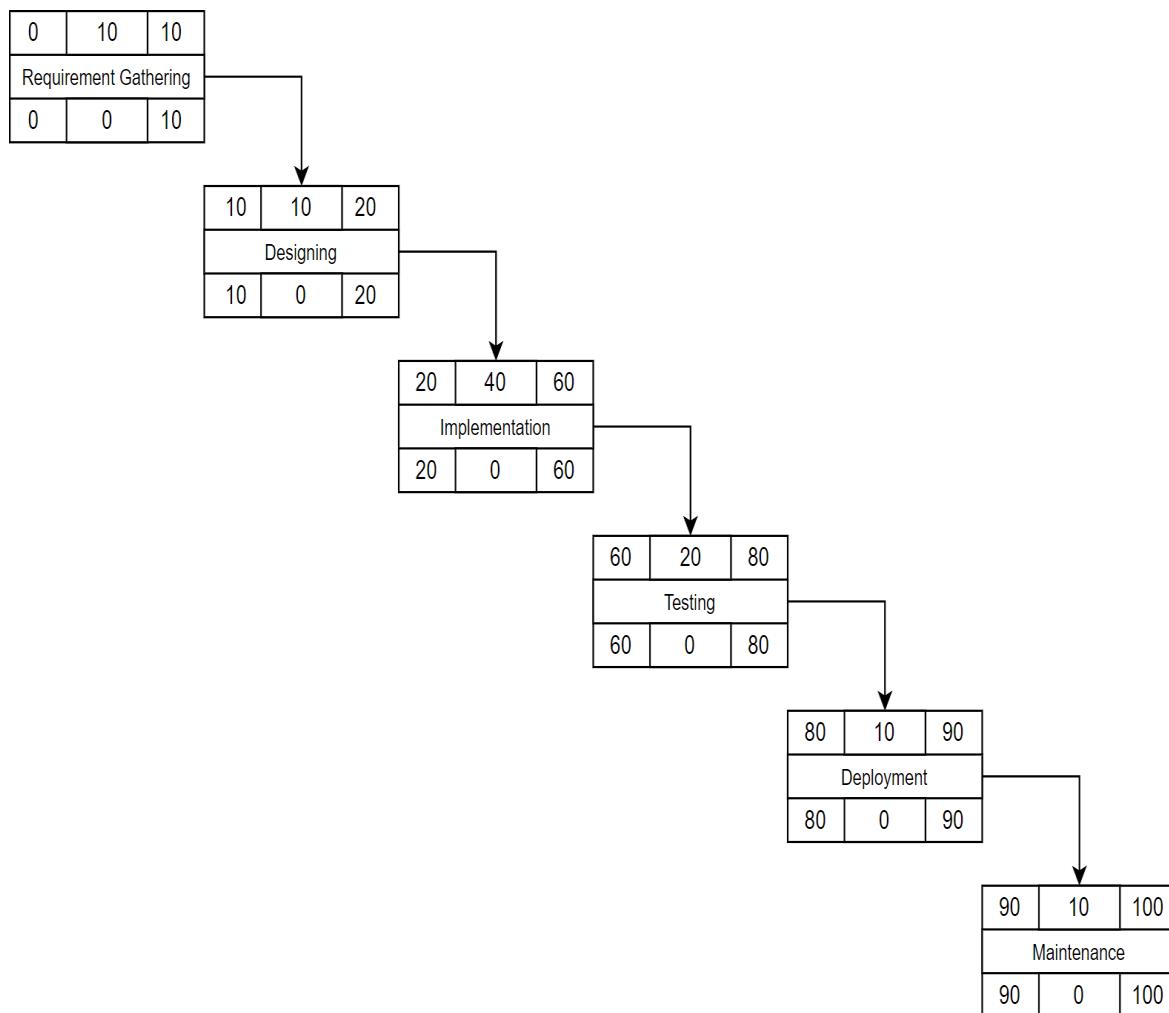
## Gantt Chart

A Gantt chart is a visual project management tool that provides a clear overview of a project's timeline. It represents tasks as horizontal bars, where the length of each bar corresponds to the task's duration and its position indicates the start and end dates. Gantt charts excel at displaying task dependencies, milestones, and resource allocation. This visual representation makes it easy to identify potential bottlenecks, track progress, and make adjustments to the project schedule. By providing a clear picture of the project timeline, Gantt charts facilitate effective communication among team members and stakeholders.



## Pert Chart

A PERT (Program Evaluation and Review Technique) chart is a network diagram used to analyze and represent the tasks involved in completing a project. It focuses on identifying the critical path, which is the sequence of tasks that directly impact the project's overall duration. PERT charts use nodes to represent events or milestones and arrows to indicate tasks and their dependencies. By visualizing task relationships, PERT charts help project managers identify potential delays, assess the impact of uncertainties, and calculate the probability of project completion within a specific timeframe. This information is invaluable for making informed decisions, allocating resources efficiently, and managing project risks.



# Software Requirements Specification (SRS)

## Purpose

The purpose of this document is to describe the software requirements for the “**TrackIt**” finance tracking website. This website is designed to help users manage their income and expenses, provide reminders for bill payments, and offer insights into their financial status through detailed reports and visualizations. The intended audience includes end-users, developers, and project managers.

## Scope

The scope of the “**TrackIt**” project encompasses the development of a personal finance management tool that offers the following functionalities:

- Income and Expense Tracking
- Budget Management
- Financial Reports
- Account Management
- User-Friendly Interface

## Product Perspective

TrackIt is a standalone web application designed to help users manage their personal finances. It integrates functionalities such as income and expense tracking, budget management, financial reporting, and account management to provide a comprehensive personal finance management solution.

## Product Functions

- **Income and Expense Tracking:** Users can record and categorize their income and expenses.
- **Budget Management:** Users can create, monitor, and adjust budgets.
- **Financial Reports:** Users can generate detailed financial reports with visual representations.
- **Account Management:** Users can manage multiple accounts and their details.
- **User Authentication:** Users can register, log in, and manage their accounts securely.

## **Functional Requirements**

### **1. User Authentication and Authorization**

- Users must be able to register and log in to the website.
- User details must be encrypted during registration.
- There will be two types of users: Admin and User.

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

- Users can record all their income sources (e.g., salaries, freelance earnings, investments).
- Users can categorize expenses (e.g., bills, groceries, entertainment).

### **3. Budget Management**

- Users can create monthly, quarterly, or annual budgets.
- The system provides real-time updates on budget performance.
- Users can adjust their budgets as needed.

### **4. Financial Reports**

- Users can generate detailed reports summarizing income, expenses, and budget performance.
- The system provides visual representations (graphs and charts) of financial data.

### **5. Account Management**

- Users can manage multiple accounts (e.g., savings, cash, investment accounts).
- Users can add new accounts or delete old ones.

## **Non-Functional Requirements**

### **1. Usability**

- The interface must be intuitive and easy to navigate.
- The design should cater to users with varying levels of financial literacy.

### **2. Security**

- All user data must be encrypted.
- The system must adhere to data protection regulations.

## **Software and Hardware Requirements**

### **1. Software Requirements**

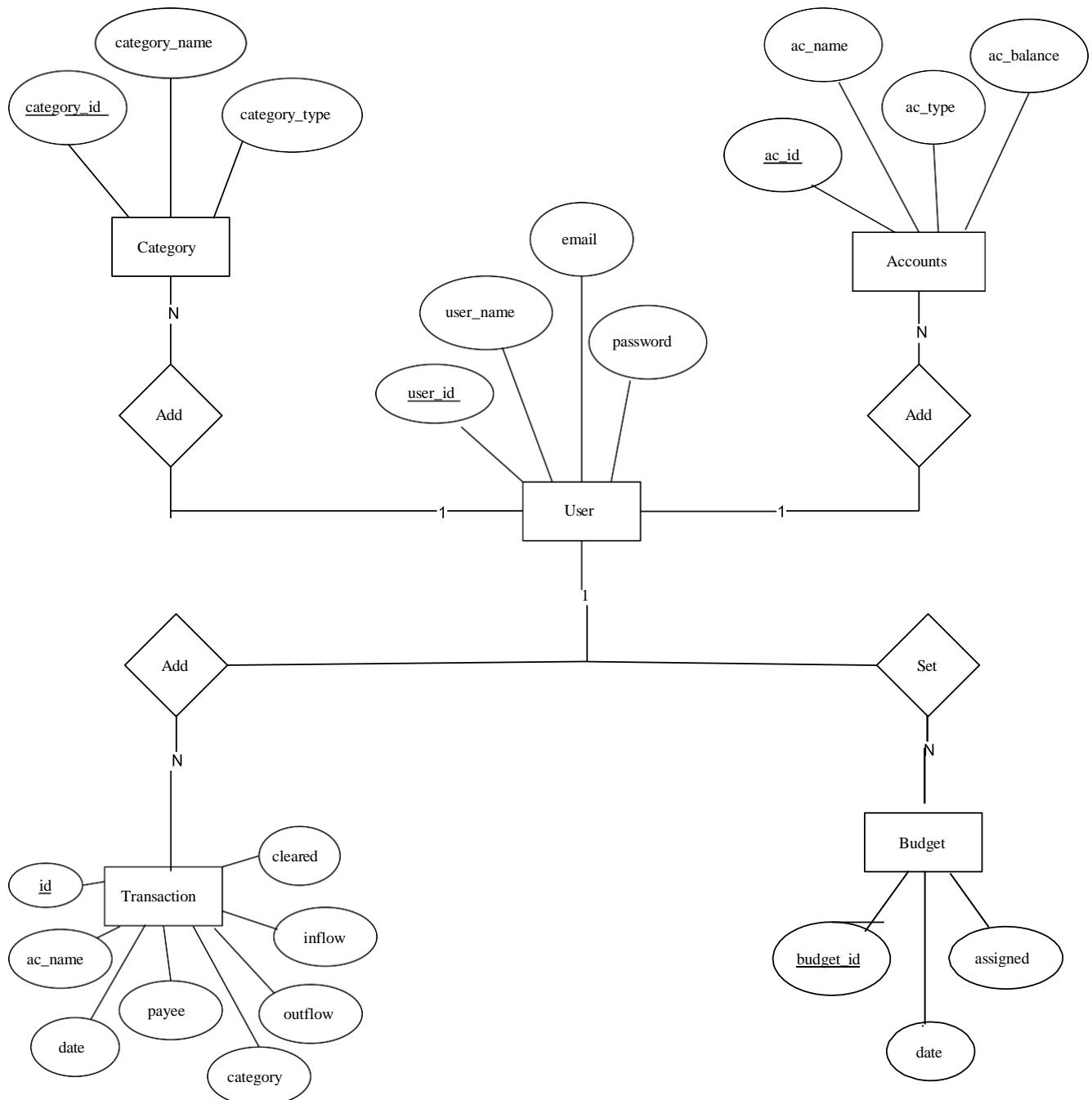
- **Operating System:** Windows, macOS, or Linux for development.
- **Web Server:** Apache.
- **Database:** MySQL.
- **Programming Language:** HTML, CSS, JavaScript, PHP.
- **Browser Compatibility:** Latest versions of Chrome, Firefox, Safari, and Edge.
- **Development Tools:** Visual Studio Code or any preferred IDE, Git for version control.

### **2. Hardware Requirements**

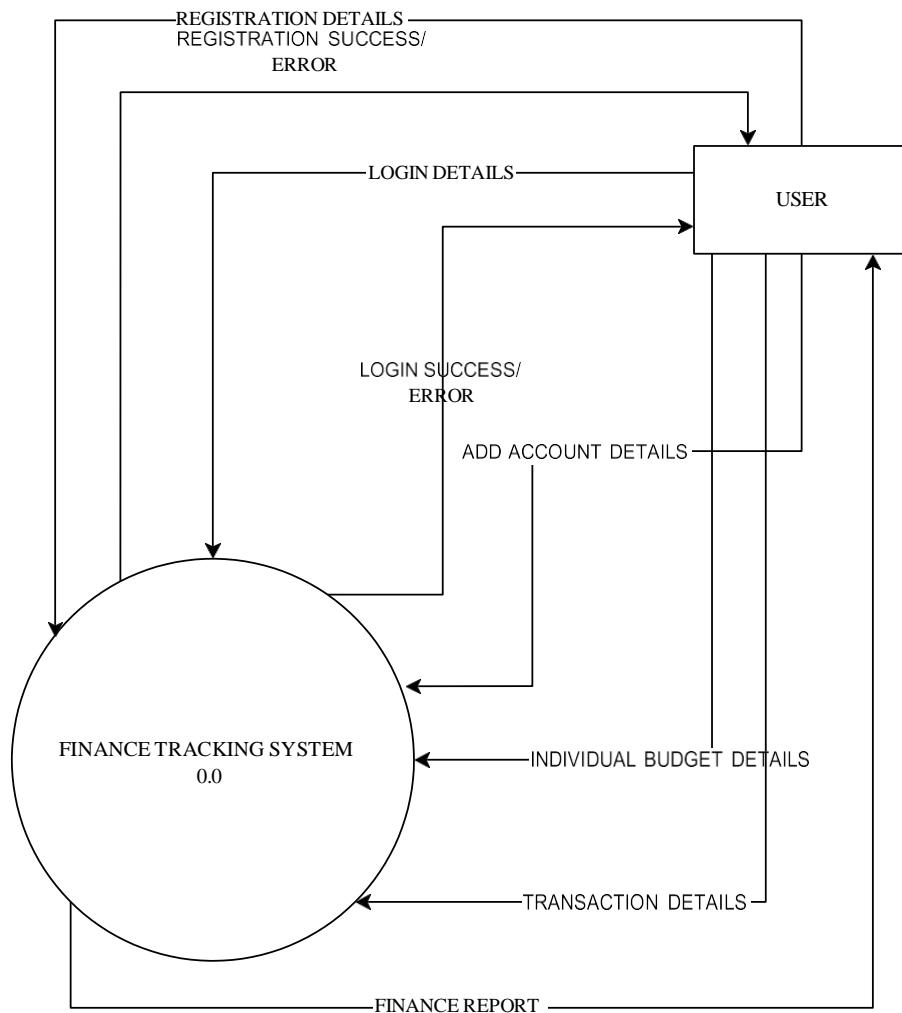
- **Processor:** Intel Core i3 or higher.
- **RAM:** 4GB or higher.
- **Storage:** 128GB or higher.

# Software Design Specification

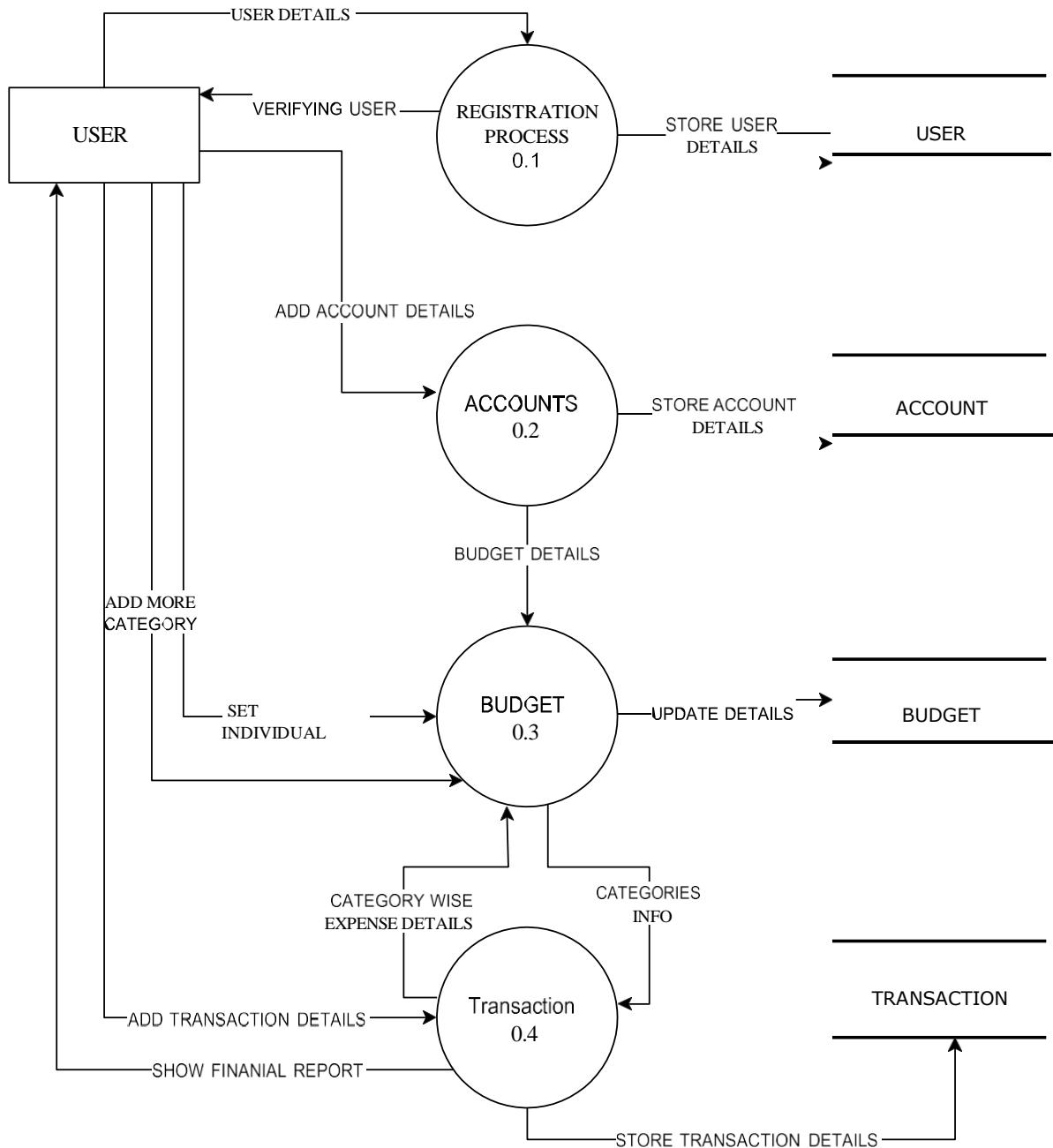
## ER Diagram



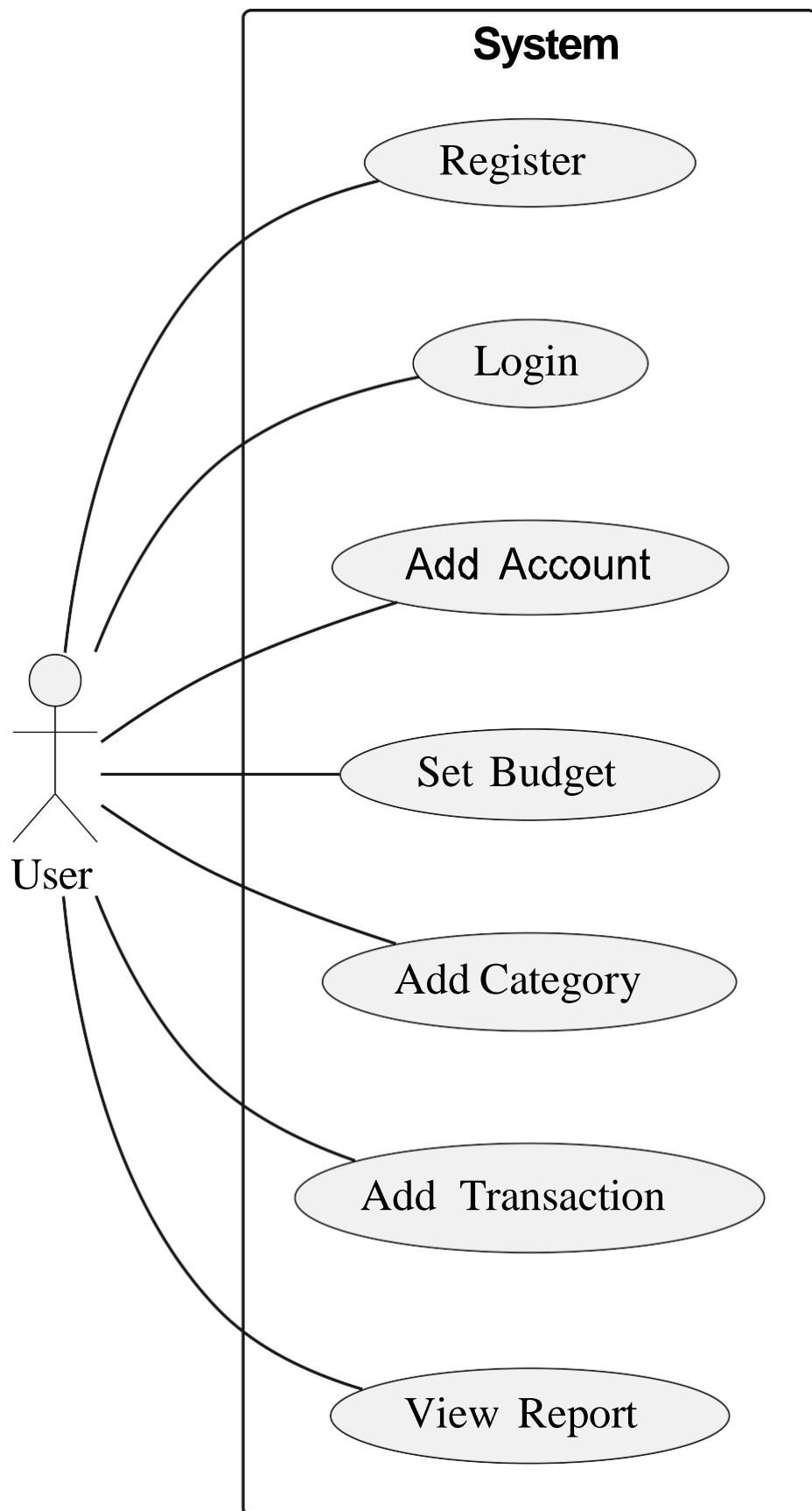
## Level 0 DFD



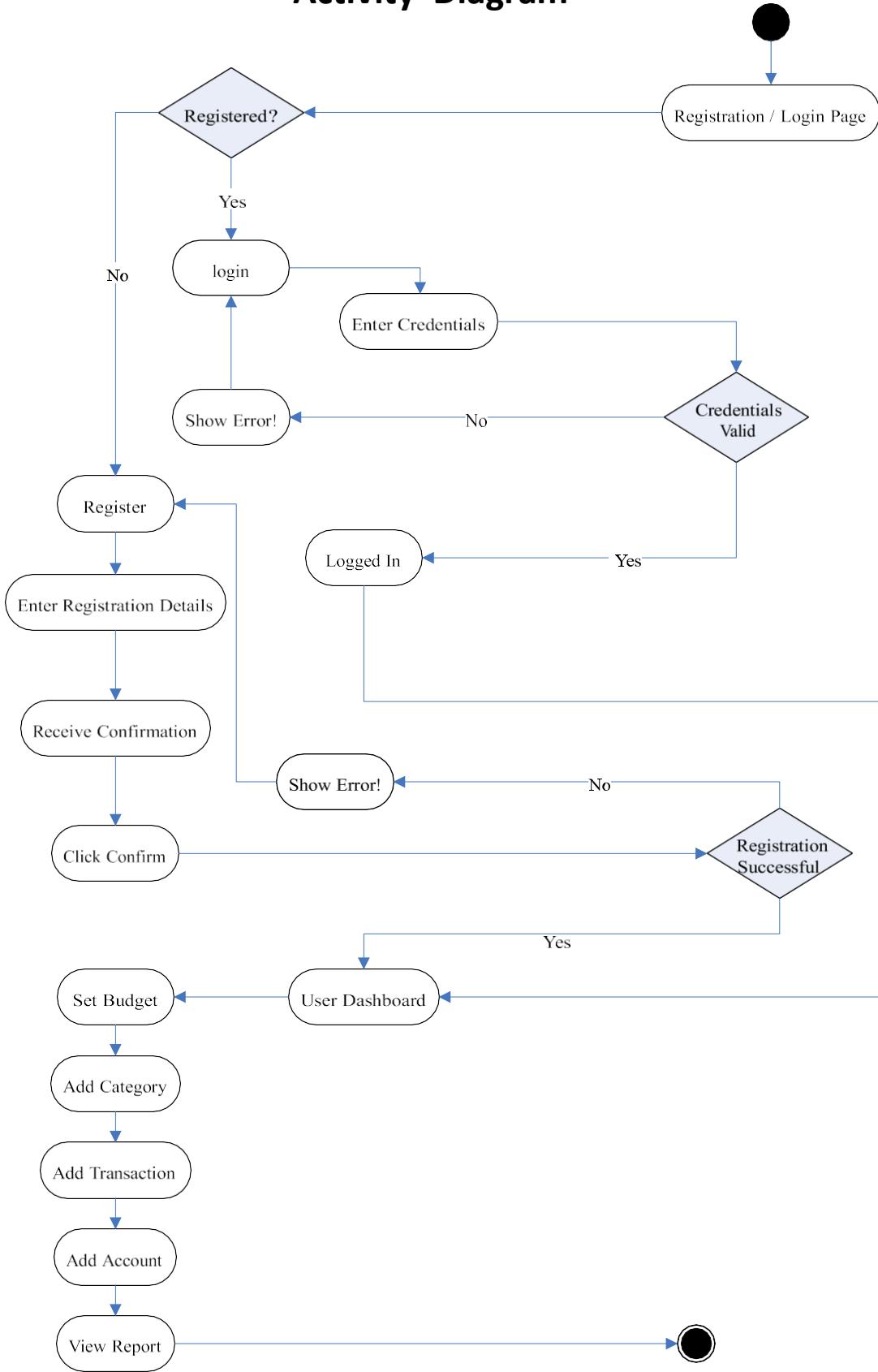
## Level 1 DFD



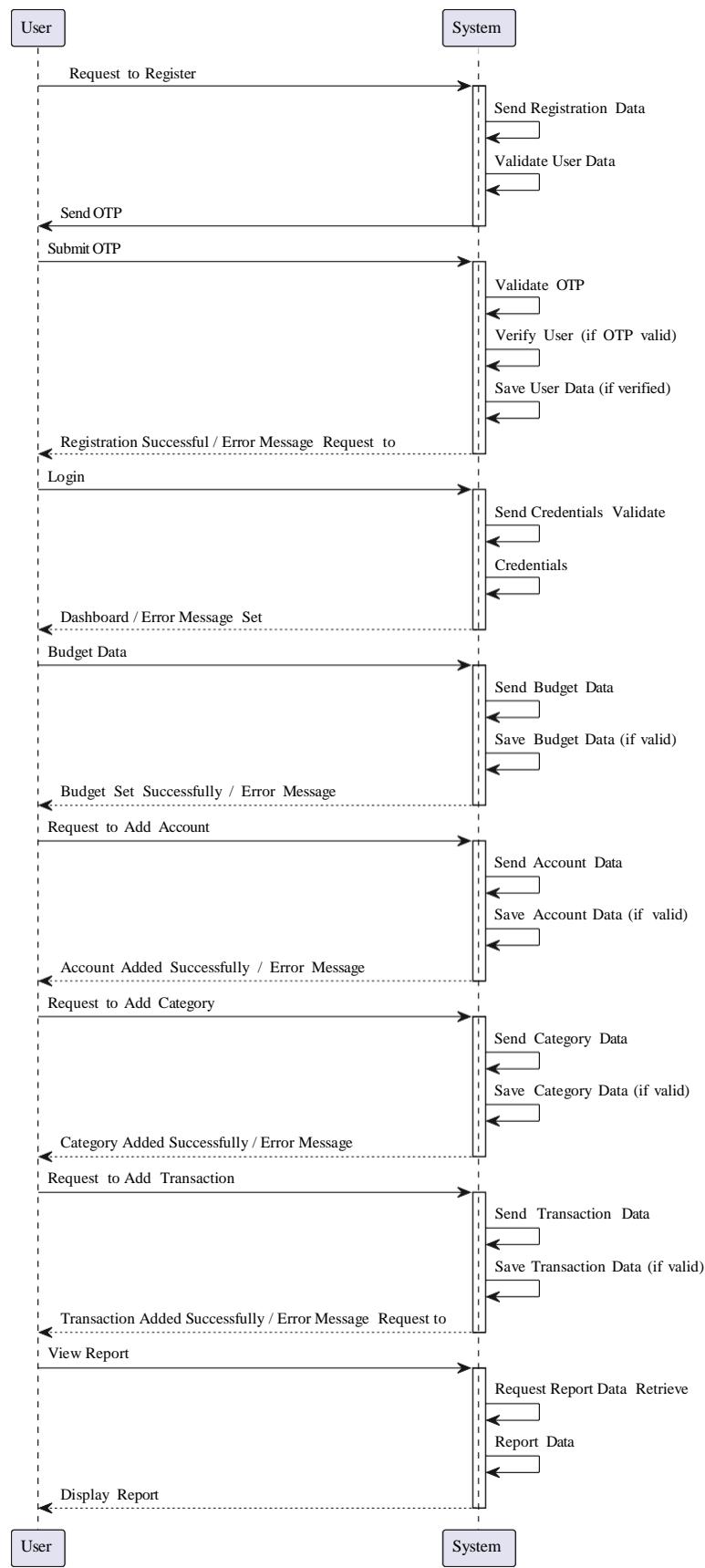
## Use Case Diagram



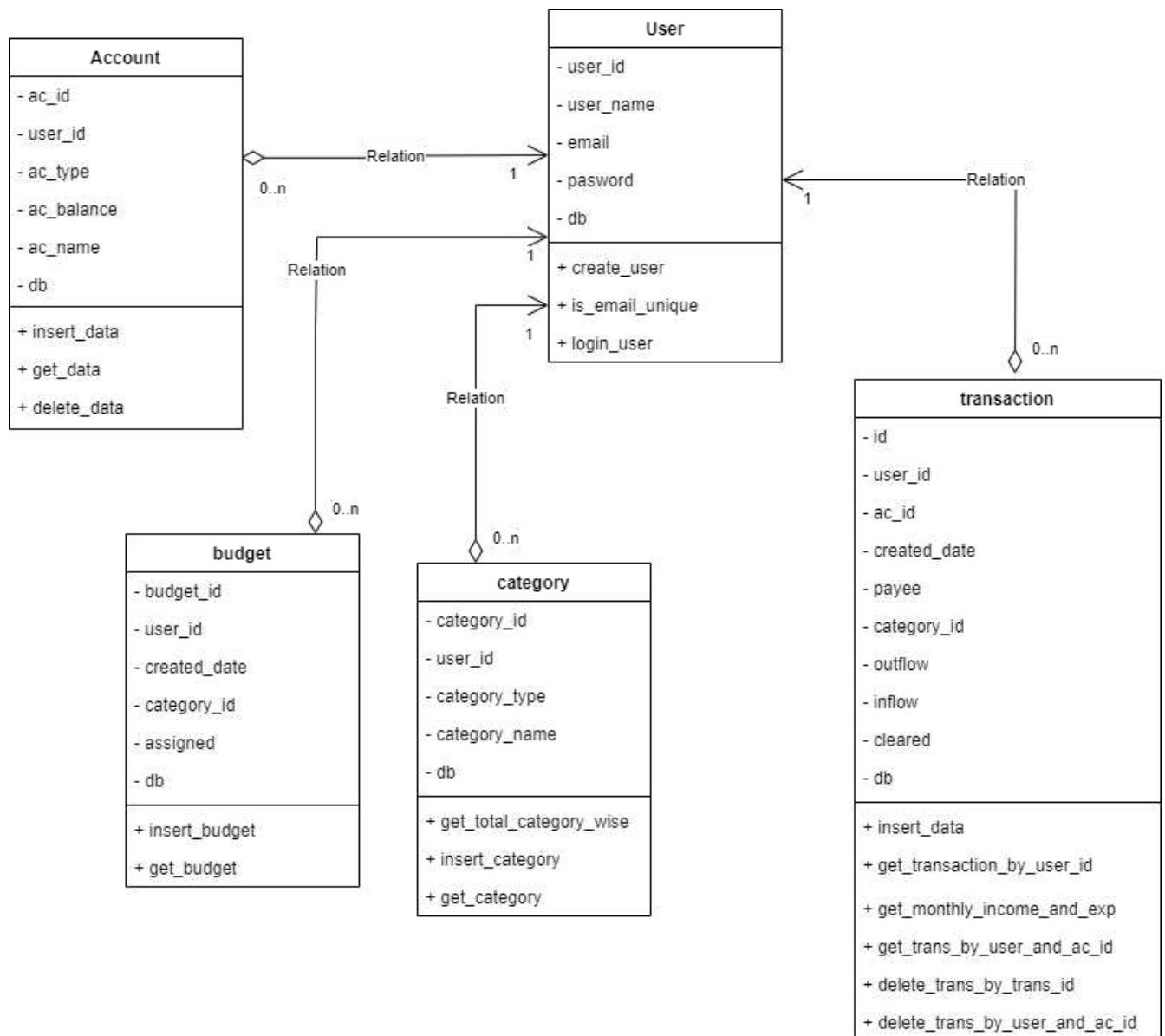
## Activity Diagram



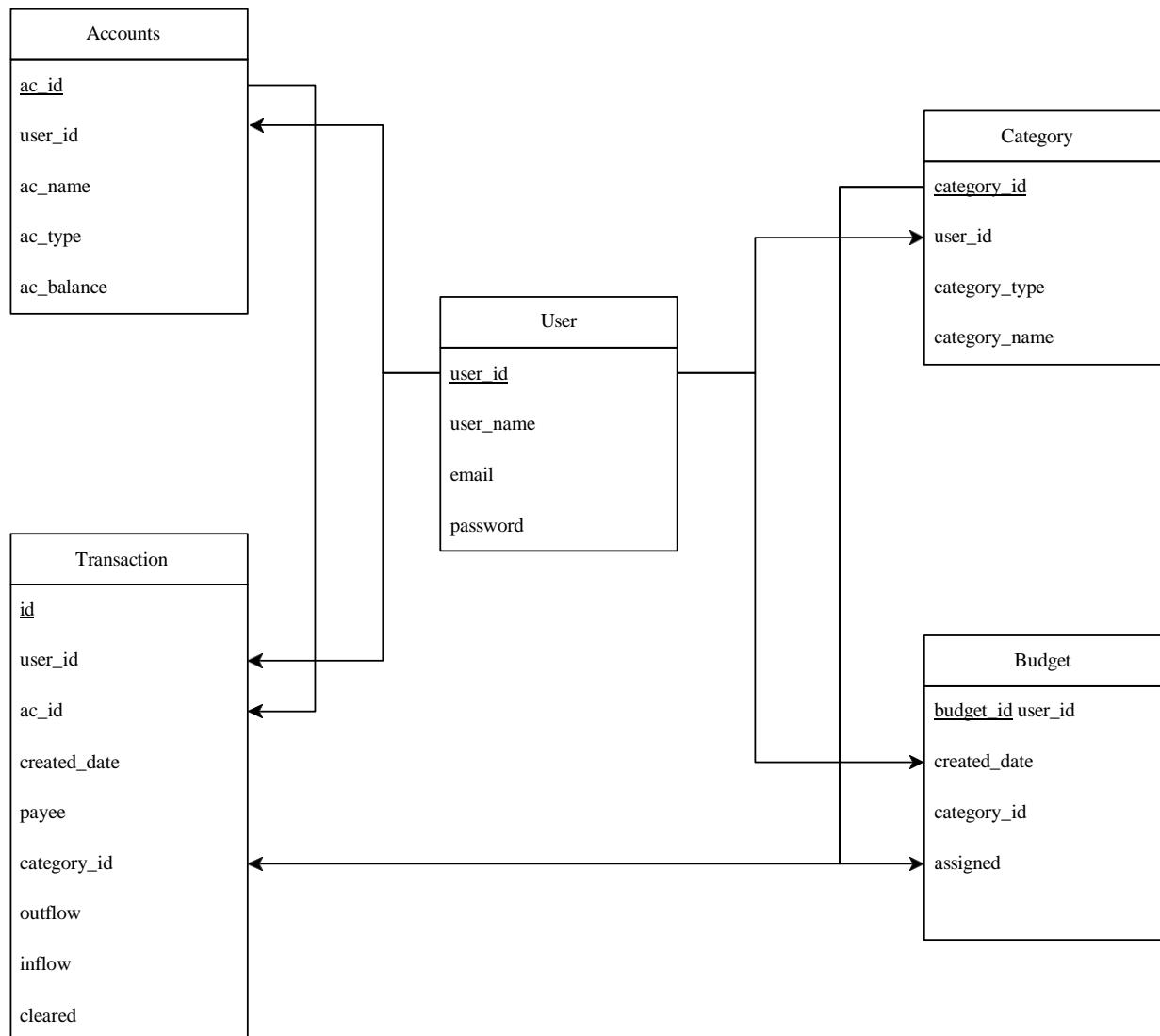
# Sequence Diagram



# Class Diagram



# Database Schema



## Cost Analysis

- Taking software project as organic type as team size is adequately small, the problem is well understood and has been solved in the past and also team members have nominal experience regarding the problem.
- a, b, c and d are constants for organic type system and the corresponding values are shown below:

Values			
a	b	c	d
2.4	1.05	2.5	0.38

- KLOC-Kilo Lines of Codes is the estimated size of the software product.
- EAF - Effort Adjustment Factor. The factors and the corresponding values for developing this project are shown below, considering the values are high for organic type system below:

Factors	Values
Software reliability(f1)	1.0
Application Database(f2)	1.0
Product complexity(f3)	1.0
Runtime Performance(f4)	1.0
Memory Constraints(f5)	1.0
Volatility of Virtual Machine(f6)	1.0
Turnaround time(f7)	1.0
Analyst capability(f8)	1.0
Application experience(f9)	1.13
S/w Engineer capability(f10)	1.0
Virtual machine experience(f11)	1.0
Programming language experience(f12)	1.07
Application of s/w engineering methods(f13)	1.0
Use of software tools(f14)	1.0
Required development schedule(f15)	1.0

## COCOMO 1

Here LOC = 2000

Therefore KLOC = 2000/1000 = 2

## **For Organic:**

**Effort:**

$$a \times (\text{KLOC})^b \text{ PM}$$

$$= 2.4 \times 2^{1.05} \text{ PM}$$

$$= 4.969271 \text{ PM}$$

**Development Time:**

$$c \times (\text{Effort})^d \text{ Months}$$

$$= 2.5 \times (4.969271)^{0.38} \text{ Months}$$

$$= 4.5976 \text{ Months}$$

$$\approx 5 \text{ Months}$$

## **COCOMO 2:**

**Effort Adjustment Factor (EAF):**

$$= f_1 \times f_2 \times f_3 \times f_4 \times f_5 \times f_6 \times f_7 \times f_8 \times f_9 \times f_{10} \times f_{11} \times f_{12} \times f_{13} \times f_{14} \times f_{15}$$

$$= 1.0 \times 1.13 \times 1.0 \times 1.07 \times 1.0 \times 1.0 \times 1.0$$

$$= 1.2091$$

**Effort:**

$$= 3.2 \times (\text{KLOC})^{1.05} \times \text{EAF PM}$$

$$= 3.2 \times (2)^{1.05} \times 1.2091 \text{ PM}$$

$$= 8.0111 \text{ PM}$$

## **Testing & Analysis**

Unit testing focuses on verifying individual components or modules of the software. Each function or method is tested in isolation to ensure that it works as expected. Unit testing is typically done early in the development process and often automated using tools like JUnit (for Java), PHP Unit (for PHP), or Jest (for JavaScript).

### **Integration Testing**

Integration testing validates the interaction between different modules or components of the software. After the individual units have been tested, integration testing ensures that these components work together as expected. It helps identify issues related to the interaction, such as data flow problems or interface mismatches, ensuring the system's cohesion.

### **System Testing**

System testing evaluates the complete and fully integrated system to verify that it meets the specified requirements. It tests the entire system in a real-world environment, focusing on end-to-end functionality, performance, security, and usability. System testing includes various subtypes, such as performance and security testing.

### **Acceptance Testing**

Also known as User Acceptance Testing (UAT), acceptance testing is conducted to ensure that the system satisfies business requirements and is ready for deployment. This is typically done by the end users or stakeholders to validate that the software behaves as expected and meets their needs. It's the final testing phase before release and can include both alpha and beta testing.

### **Recovery Testing**

Recovery testing assesses the system's ability to recover from failures, such as crashes, hardware malfunctions, or network failures. It checks how well the system can restore itself after encountering an issue and how quickly it returns to normal operation. This type of testing ensures that the software is resilient and fault-tolerant.

### **Functional Testing**

Functional testing focuses on verifying that the software's functionalities operate according to the specified requirements. This type of testing checks whether each feature of the system performs as expected, covering input/output, user commands, data manipulation, and interactions with external systems. It ensures that the system functions in line with user expectations.

### **Hardware/Software Testing**

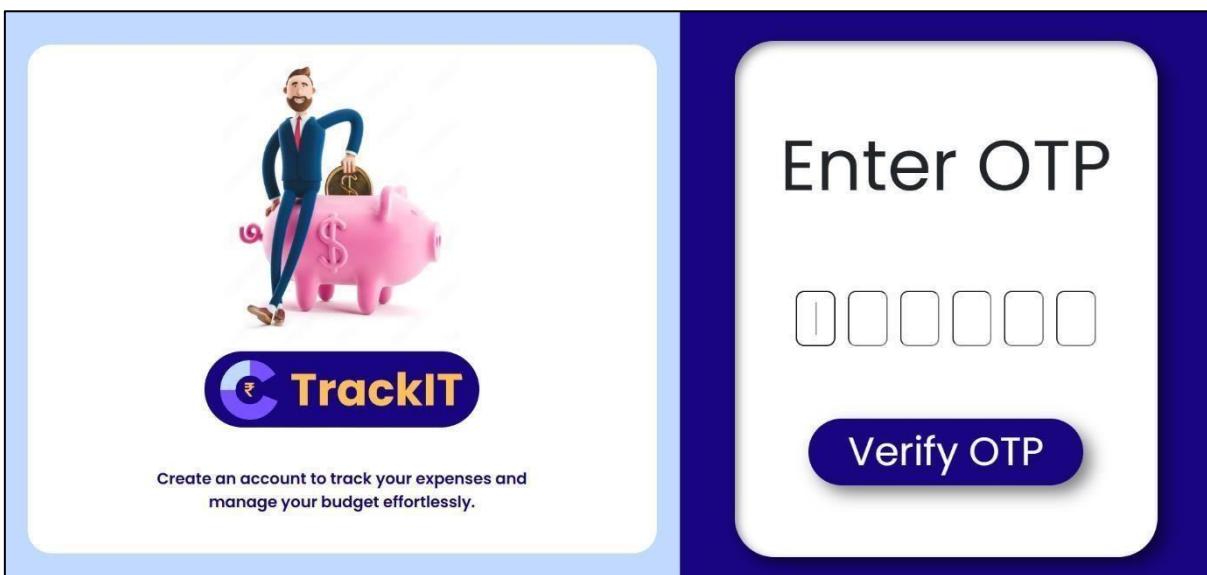
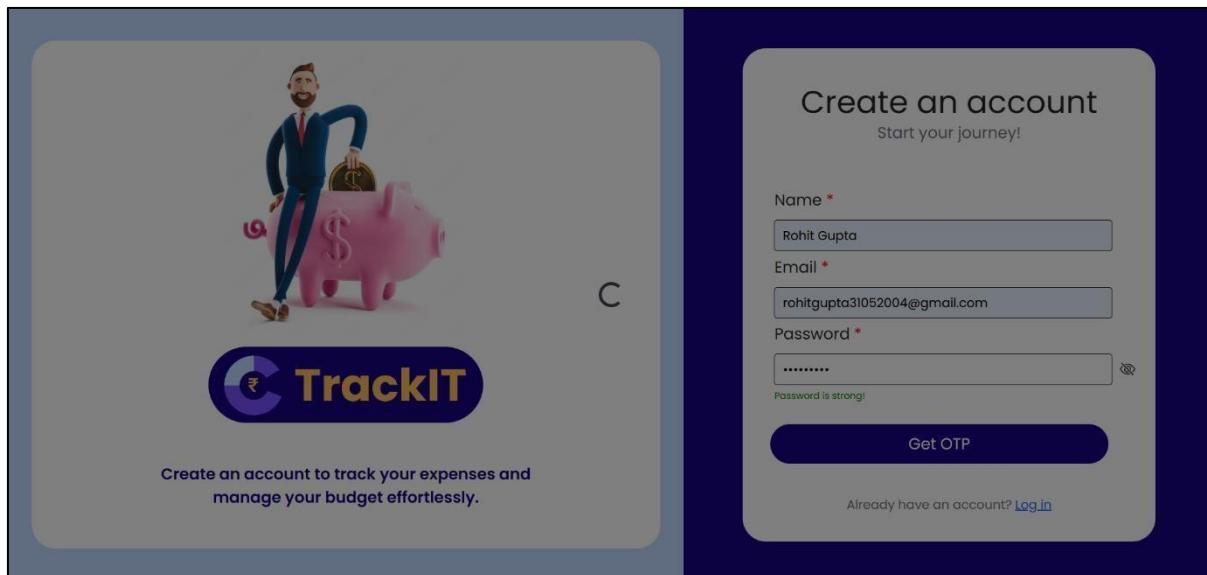
Hardware/software testing ensures that the software is compatible with the hardware on which it runs. It tests the interactions between hardware components and software systems, verifying that they work together without errors. This type of testing is crucial for systems with specific hardware dependencies, such as embedded systems, IoT devices, or specialized hardware setups.

### **Security Testing**

Security testing identifies vulnerabilities, risks, and potential threats in the system. It ensures that the application is protected from attacks like SQL injections, cross-site scripting (XSS), and unauthorized access. Security testing includes penetration testing, risk assessments, and testing for data encryption, authentication, and access controls to prevent breaches.

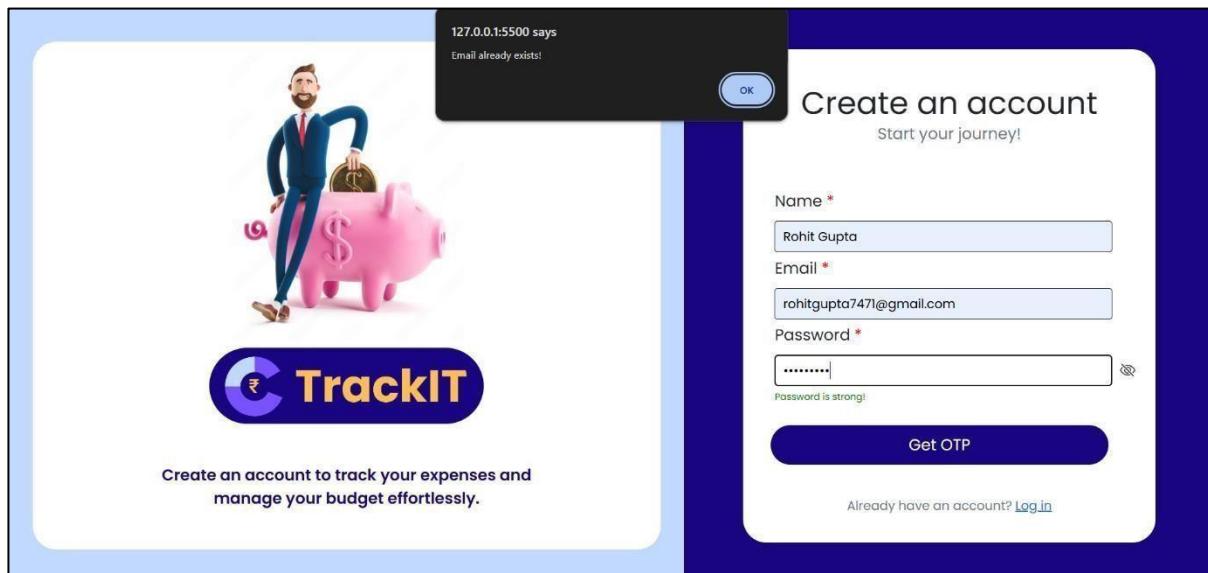
## Test Case 1: Successful Signup

<b>Test Name</b>	Unit Testing
<b>Test Description</b>	Verify that a user can successfully create an account and proceed to OTP verification.
<b>Objective</b>	Ensure that the user can complete the signup process.
<b>Input</b>	Enter valid username, email, and password and OTP.
<b>Expected Result</b>	The user is redirected to the OTP verification page. An OTP is sent to the registered email.
<b>Output</b>	Passed



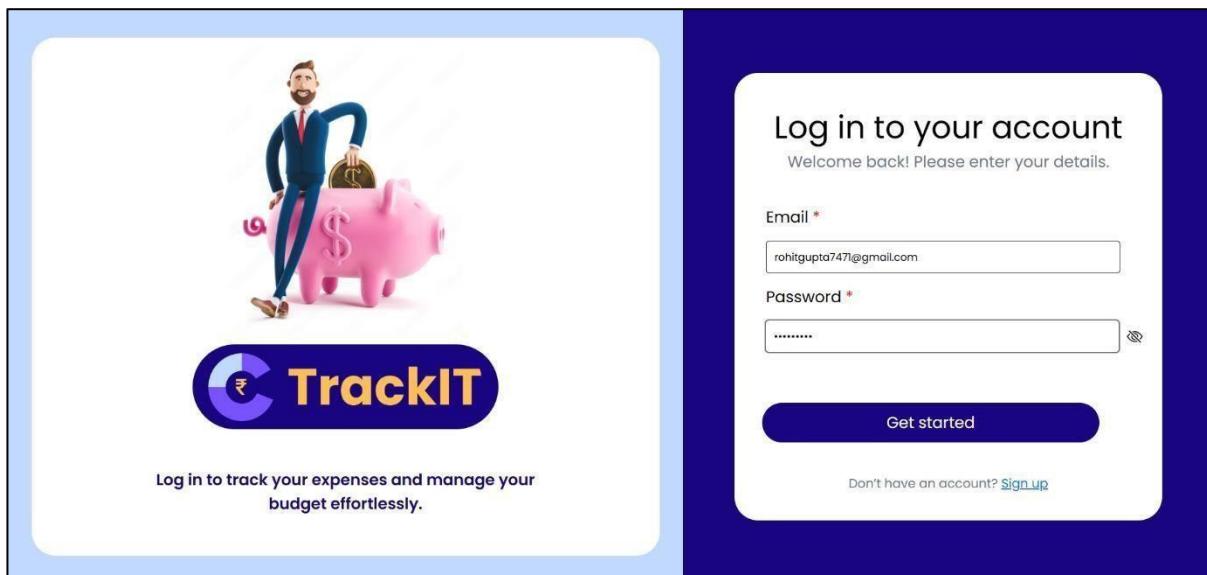
## Test Case 2: Signup with Existing Email

<b>Test Name</b>	Unit Testing
<b>Test Description</b>	Verify that signup fails when using an email that is already registered.
<b>Objective</b>	Ensure that the system prevents duplicate email registration.
<b>Input</b>	Enter an email that is already registered.
<b>Expected Result</b>	The system displays an error message indicating that the email is already in use.
<b>Output</b>	Passed



## Test Case 3: Successful Login

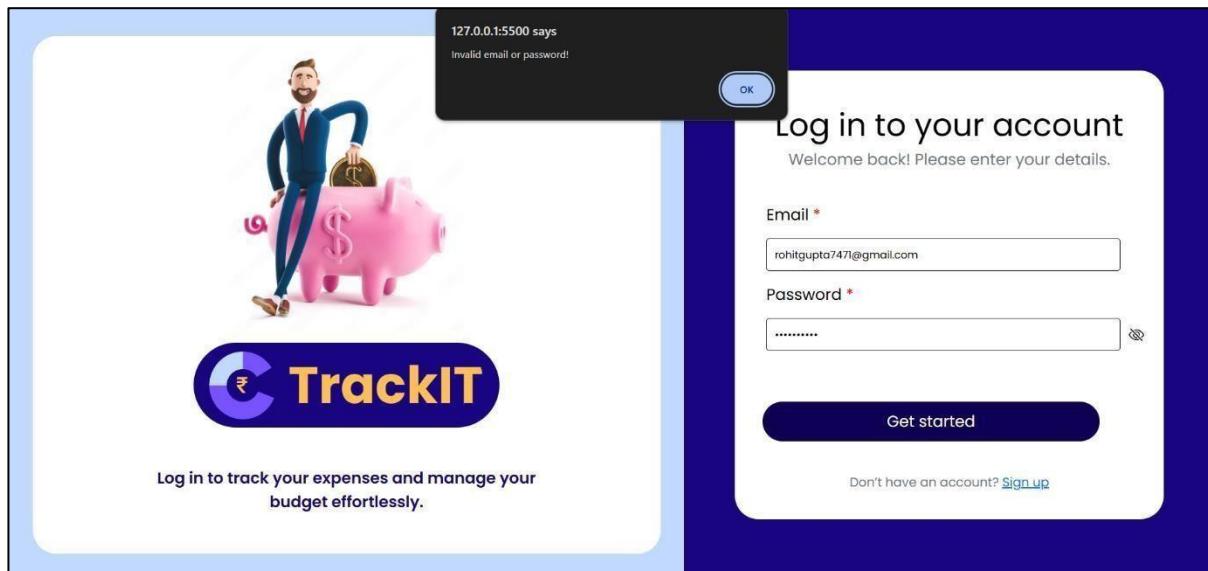
<b>Test Name</b>	Unit Testing
<b>Test Description</b>	Verify that a user can successfully log in with valid credentials.
<b>Objective</b>	Ensure that the user can log in with a valid username and password.
<b>Input</b>	Enter a valid username and password.
<b>Expected Result</b>	The user is logged in successfully and redirected to the dashboard.
<b>Output</b>	Passed



Category	Assigned	Activity	Available
Bills	₹19,500	₹500	₹19,000
Electricity	15000	₹0.00	₹15,000
Water	2000	₹0.00	₹2,000
Rent	1000	₹0.00	₹1,000
Internet	1500	₹500.00	₹1,000
Phone	0	₹0.00	₹0
Needs	₹2,000	₹800	₹1,200
Groceries	2000	₹0.00	₹2,000
Transport	0	₹0.00	₹0

## Test Case 4: Login with Invalid Credentials

<b>Test Name</b>	Unit Testing
<b>Test Description</b>	Verify that login fails with invalid credentials.
<b>Objective</b>	Ensure that incorrect login attempts are handled properly.
<b>Input</b>	Enter an invalid username or password.
<b>Expected Result</b>	The system displays an error message indicating invalid credentials.
<b>Output</b>	Passed



## Test Case 5: Add Account

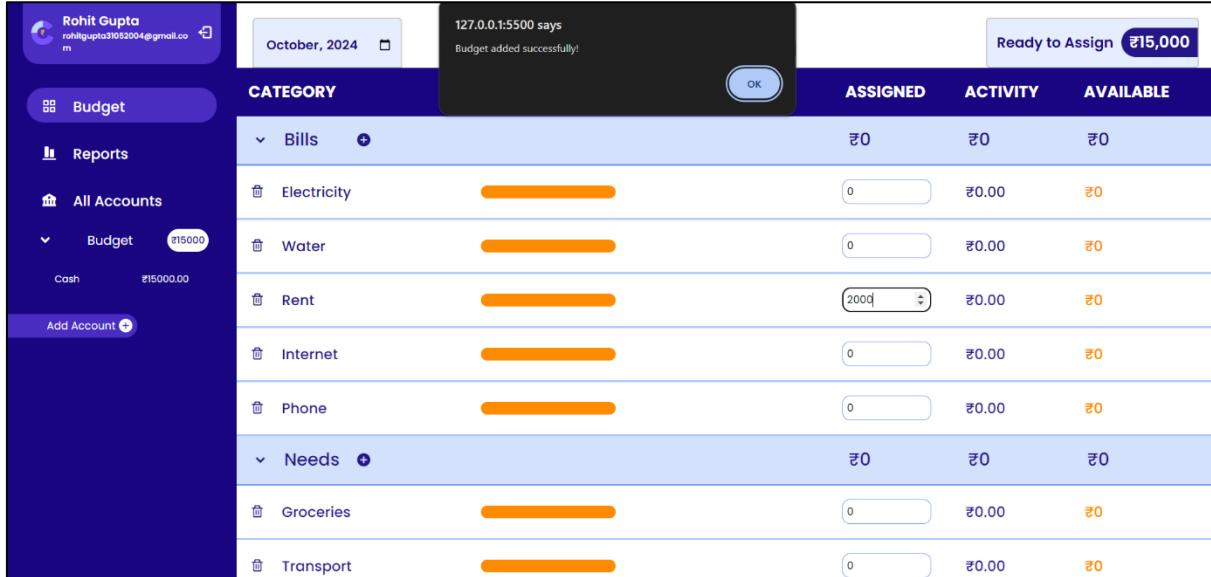
<b>Test Name</b>	Unit Testing
<b>Test Description</b>	Verify that a user can successfully add a new account.
<b>Objective</b>	Ensure that users can add accounts to manage finances.
<b>Input</b>	Enter account name, account type and initial amount and press save.
<b>Expected Result</b>	The account is added successfully, and a confirmation message is displayed.
<b>Output</b>	Passed

Screenshot of a budgeting application showing the 'Add Account' dialog box. The dialog asks for a name (Cash), account type (Cash), and current balance (₹15000). A confirmation message "Account added successfully!" is displayed at the top.

Screenshot of the application showing the 'Cash' account details. It displays cleared balance, uncleared balance, and working balance all as ₹0. There is a red 'Delete Account' button in the top right corner.

## Test Case 6: Add Budget

<b>Test Name</b>	Unit Testing
<b>Test Description</b>	Verify that a user can successfully set a budget for a specific category.
<b>Objective</b>	Ensure that users can add and set budgets for their expense categories.
<b>Input</b>	Select the budget category and enter the budget amount.
<b>Expected Result</b>	The budget is saved successfully, and a confirmation message is displayed. The budget should now be visible in the budget summary.
<b>Output</b>	Passed



## Test Case 7: Add Transaction

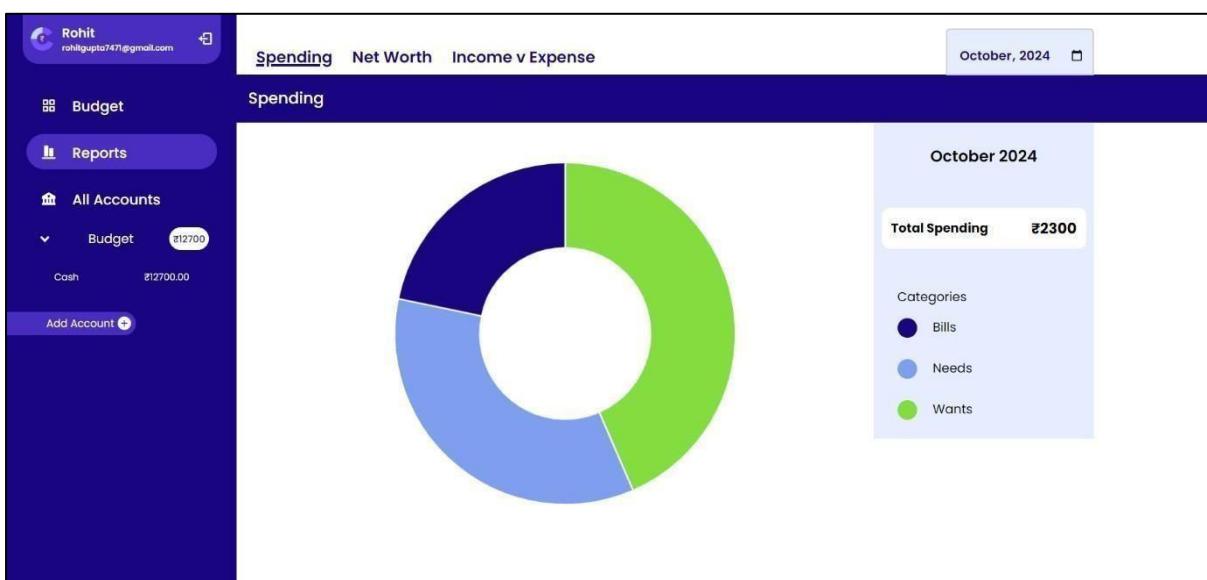
<b>Test Name</b>	Unit Testing
<b>Test Description</b>	Verify that a user can successfully add a transaction.
<b>Objective</b>	Ensure that users can add income and expense transactions accurately.
<b>Input</b>	Select account and enter transaction details. i.e., date, amount, type (inflow/outflow), date, category, cleared status.
<b>Expected Result</b>	The transaction is saved successfully, and the balance updates accordingly
<b>Output</b>	Passed

The screenshot shows the 'All Accounts' screen of a budgeting app. A modal dialog box is displayed in the center, showing the message "127.0.0.1:5500 says Transaction added successfully!" with an "OK" button. Below the modal, there is a transaction entry form. The form fields include: Account (Cash), Date (17-10-2024), Payee (Recharge), Category (Internet), Outflow (₹500), Inflow (Inflow checked), and Cleared (unchecked). On the left side of the screen, the sidebar shows the user profile (Rohit Gupta, rohitgupta31052004@gmail.com), Budget section with a balance of ₹15000.00, and a summary table for Cash showing a balance of ₹15000.00.

The screenshot shows the 'All Accounts' screen of the app after the transaction has been added. The transaction details are now visible in the table below the form. The table shows one row with the following data: Account (Cash), Date (2024-10-17), Payee (Recharge), Category (Internet), Outflow (₹500.00), Inflow (cleared checked), and Cleared (checked). The sidebar on the left remains the same, displaying the user profile and budget information.

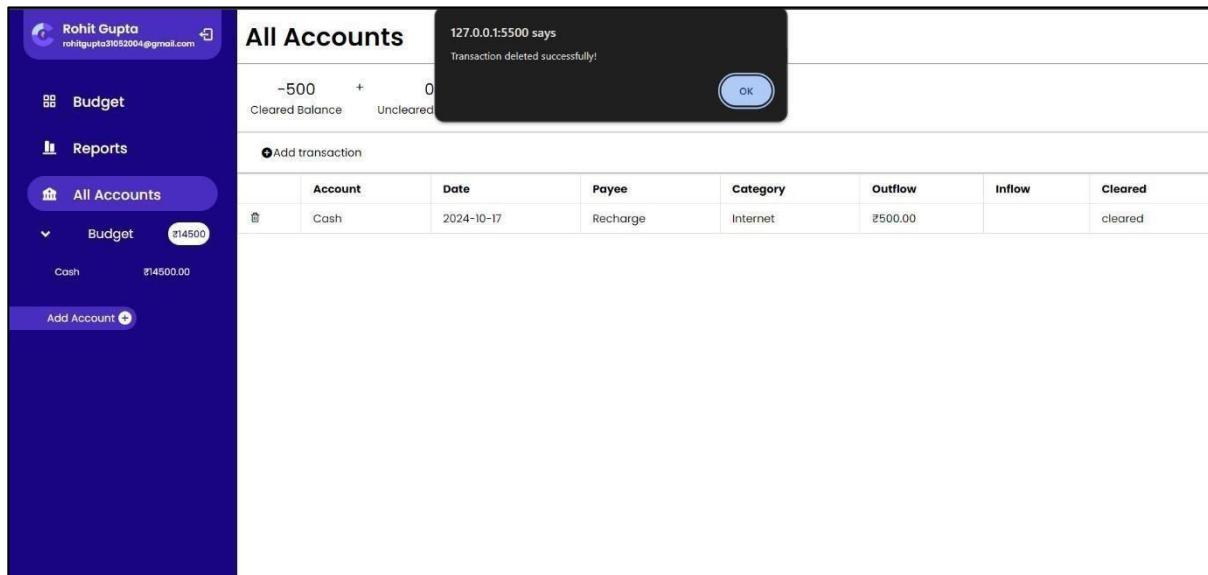
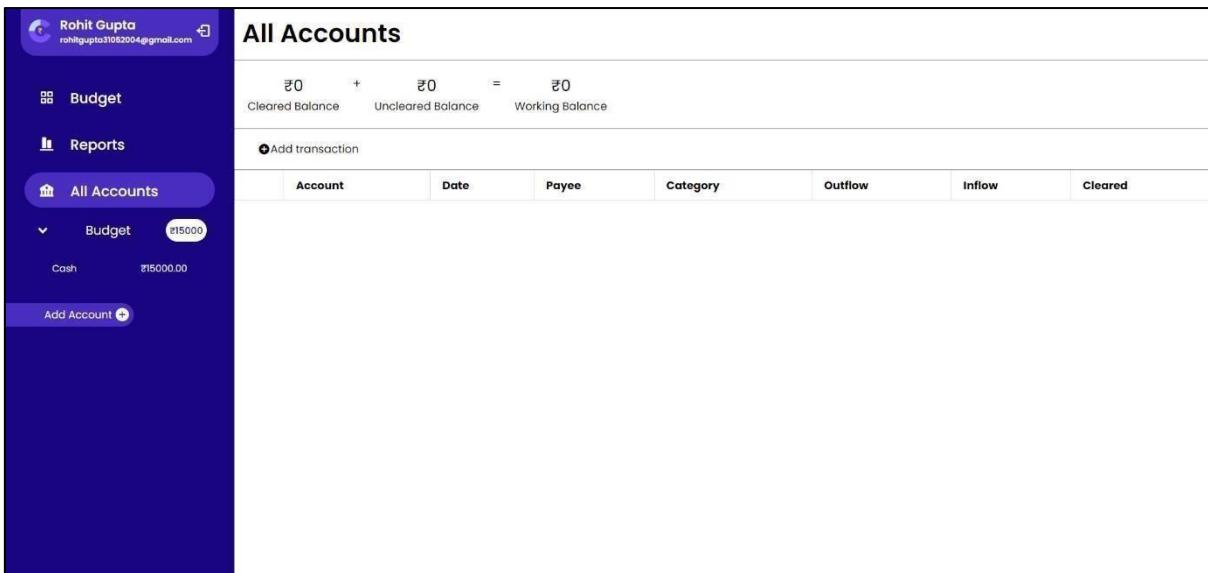
## Test Case 8: View Category-wise Spending and Report

<b>Test Name</b>	Unit Testing
<b>Test Description</b>	Verify that the user can view category-wise spending data and generate monthly financial report.
<b>Objective</b>	Ensure that users can view their spending based on different categories and monthly report.
<b>Input</b>	Select a date range.
<b>Expected Result</b>	Category-wise spending data and monthly report is displayed in a chart and table format respectively.
<b>Output</b>	Passed



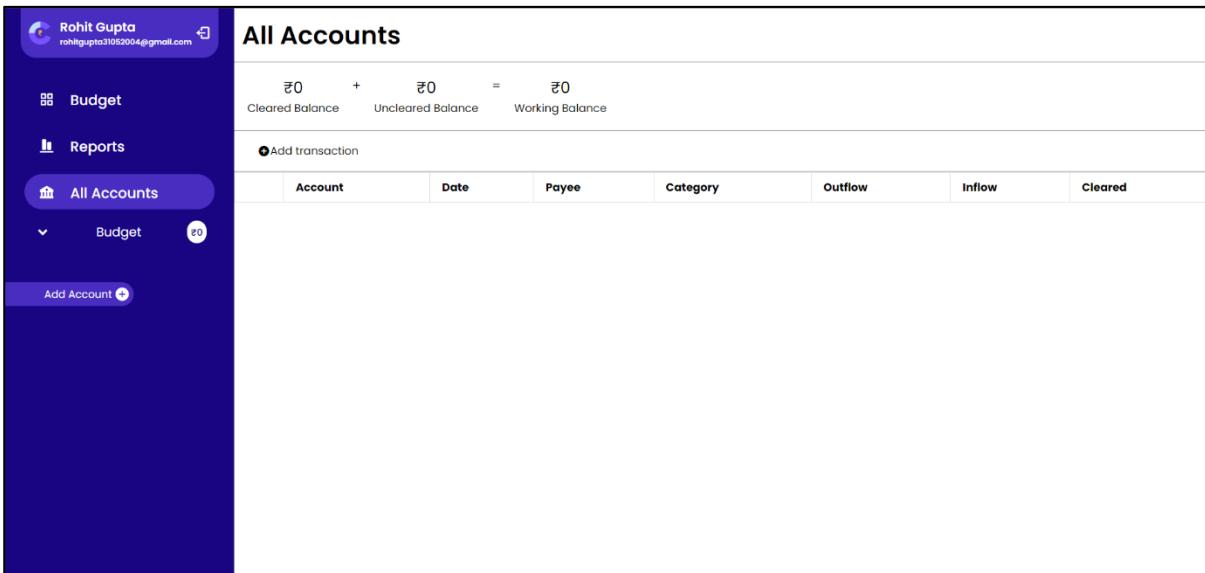
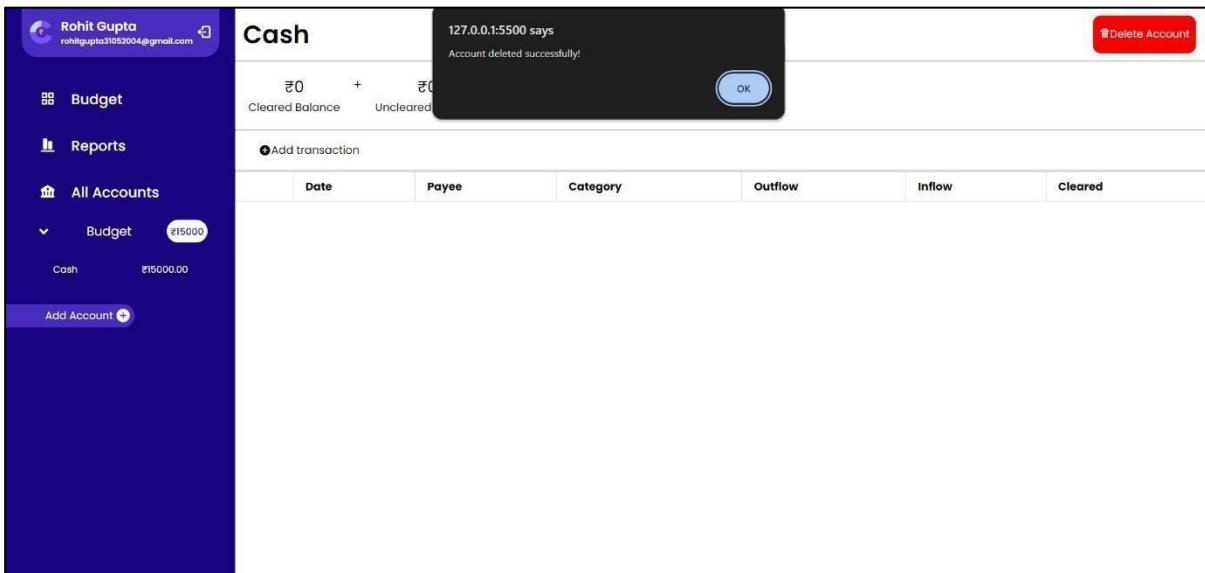
## Test Case 9: Delete Transaction

<b>Test Name</b>	Unit Testing
<b>Test Description</b>	Verify that a user can delete a transaction.
<b>Objective</b>	Ensure that users can remove transactions.
<b>Input</b>	Click the delete button next to the transaction you want to delete.
<b>Expected Result</b>	The transaction is deleted successfully, and the balance updates accordingly.
<b>Output</b>	Passed



## Test Case 10: Delete Account

<b>Test Name</b>	Unit Testing
<b>Test Description</b>	Verify that a user can successfully delete an account.
<b>Objective</b>	Ensure that users can delete an account from their account list.
<b>Input</b>	Select the account to delete from the account list, click the delete button next to it, and confirm the deletion action when prompted.
<b>Expected Result</b>	The account is deleted successfully, a confirmation message is displayed, and the account is no longer visible in the account list.
<b>Output</b>	Passed



# User Manual

Welcome to the *TrackIt* User Manual! This guide will walk you through how to use this finance tracker to efficiently manage your finances, track expenses, set budgets, and generate reports.

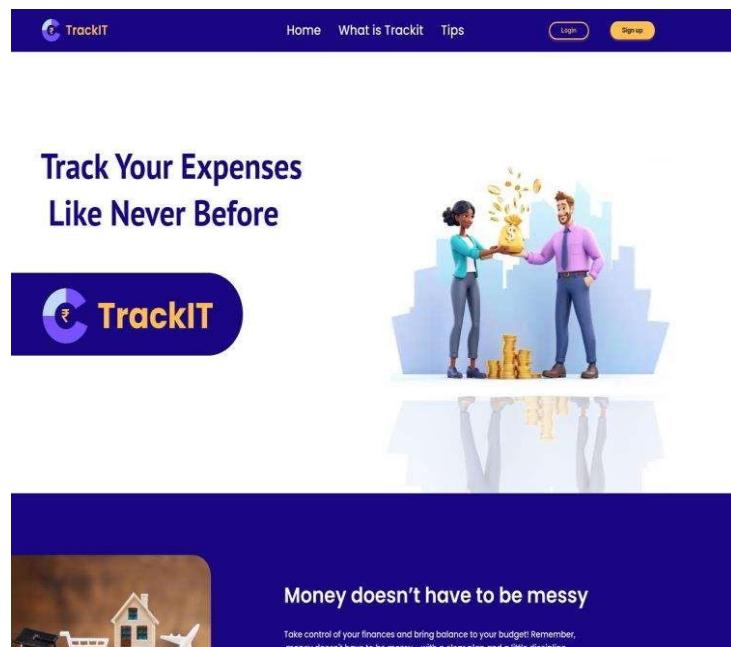
Ensure your device meets the following requirements to use the Finance Tracker:

- **Operating System:** Windows 10/11, macOS 10.15 or later, Android 8.0 or later, iOS 13.0 or later
- **Browser:** Chrome, Edge
- **Internet Connection:** Required for syncing data across devices

## \*\*\*Getting Started\*\*\*

### *Creating an Account*

1. Open the *TrackIt* website.
2. Click on the "Sign Up" button

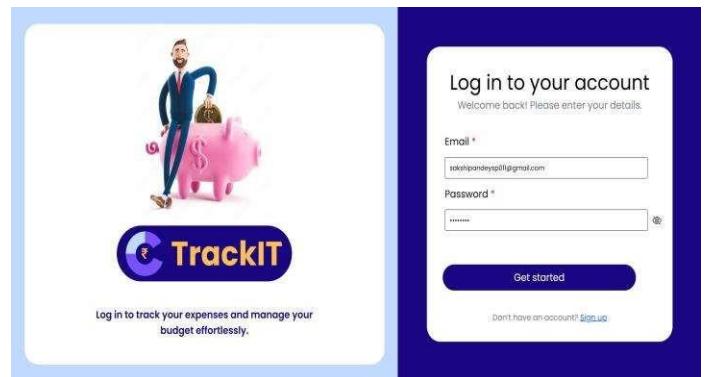


3. Enter your email and create a password.
4. Click "Create Account." You will receive a confirmation email to verify your account.



## Logging In

- 1 After creating an account, return to the login screen.
- 2 Enter your email and password.
- 3 Click "Log In" to access your *TrackIt* dashboard.



## Dashboard Overview

Upon logging in, the dashboard displays the budget page as the default view.

### ➤ Budget Section

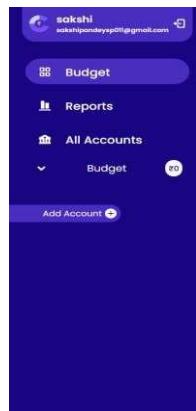
The **Budget page** gives a summary of

- Budget allocations for the current month
- Overview of different spending categories like “Bills”, “Needs” and “Wants” each containing specific items
- Each category includes three key columns:
  1. **Assigned:** The amount of money you've planned or assigned for that specific category.
  2. **Activity:** This shows any spending that has occurred within the month for that category.
  3. **Available:** The remaining amount of money left in that category for the month.

CATEGORY	ASSIGNED	ACTIVITY	AVAILABLE
Bills	₹0	₹0	₹0
Electricity	0	₹0.00	₹0
Water	0	₹0.00	₹0
Rent	0	₹0.00	₹0
Internet	0	₹0.00	₹0
Phone	0	₹0.00	₹0
Needs	₹0	₹0	₹0
Groceries	0	₹0.00	₹0
Transport	0	₹0.00	₹0

The sidebar contains:

- **Budget:** Main section for tracking and managing your budget.
- **Reports:** Likely for generating financial reports.
- **All Accounts:** Shows all linked accounts.
- **Add Account:** Option to link a new account.
- Profile information is displayed at the top.



## ➤ Accounts Section

Click **Add Account** to create a personal account, add income, and track expenses.

Add Account

Give it a name

What type of account are you adding?

Select account type...

What is your current account balance?

Save
Cancel

Add Account

Give it a name

What type of account are you adding?

Select account type...

Select account type...

Savings  
Current  
Cash  
Credit Card  
Loan  
Interest  
Other

October, 2024

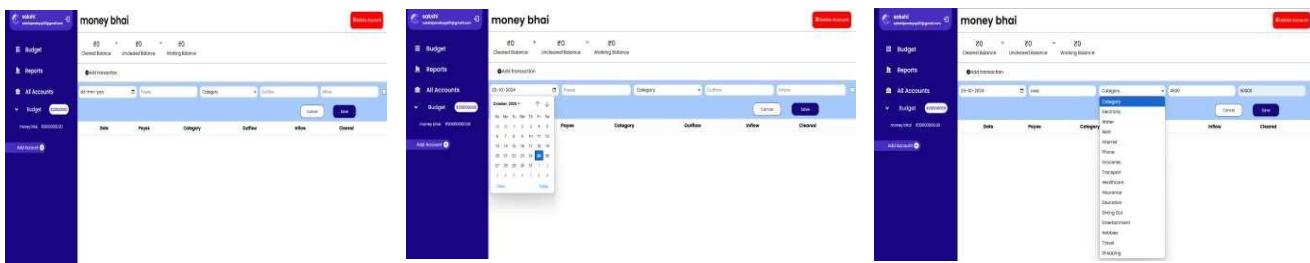
This page says  
Account added successfully!

Ready to Assign ₹0

CATEGORY	ASSIGNED	ACTIVITY	AVAILABLE
Bills	₹0	₹0	₹0
Electricity	0	₹0.00	₹0
Water	0	₹0.00	₹0
Rent	0	₹0.00	₹0
Internet	0	₹0.00	₹0
Phone	0	₹0.00	₹0
Needs	₹0	₹0	₹0
Groceries	0	₹0.00	₹0
Transport	0	₹0.00	₹0

Your account will be displayed in the sidebar, with the account balance shown under the budget arrow. Click on it to view your account page.

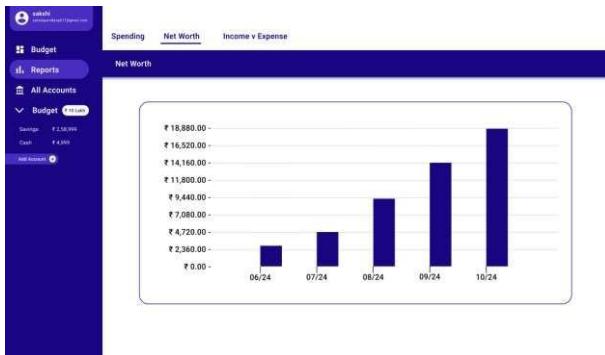
- **Record Transactions:** Users can log financial transactions by entering details like the date, payee, category (e.g., food, rent), and whether the money is an **outflow** (expense) or **inflow** (income). This helps track where money is going and coming from.
- **Monitor Balances:** The page shows three key balances:
  1. **Cleared Balance:** Funds that have been processed and are available.
  2. **Uncleared Balance:** Funds that have not yet been processed.
  3. **Working Balance:** The total available balance, combining cleared and uncleared amounts. This gives users a clear view of their current financial status.



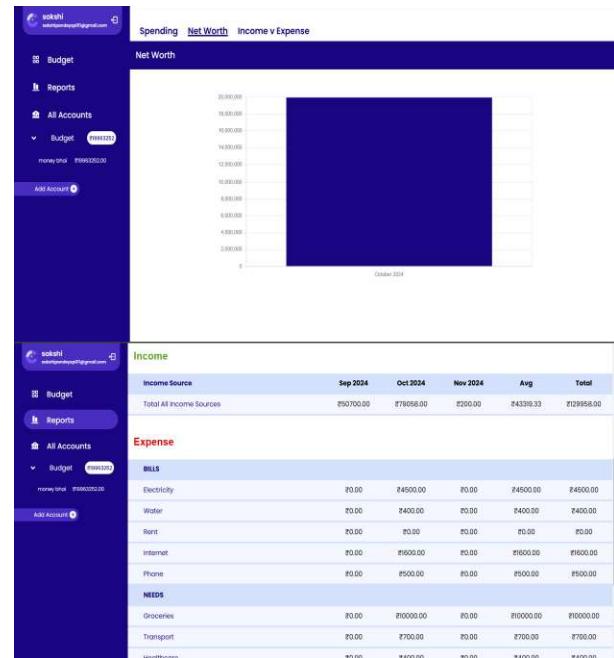
## ➤ *Reports Section*

To create an effective finance tracker, users need to log a sufficient number of transactions across key spending categories—such as bills, needs, and wants. This data entry will allow the website to generate accurate and insightful graphical representations in the reports section, helping users visualize and analyze their spending patterns.





The net worth page displays a bar chart of the last five months. If no data is available for previous months, only the current month's data is shown.



## Troubleshooting

### Common Issues and Solutions:

- Can't log in:** Ensure you are using the correct email and password.
- Transactions not saving:** Check your internet connection or refresh the page.

# Conclusion, References and Limitations

## Conclusion

The Personal Finance Tracker project provides users with a dynamic and accessible platform to manage their finances with ease and precision. By offering tools for users to log daily transactions, set specific budgets, and categorize their spending, the application empowers individuals to better understand and control their financial habits. Through its clear interface and organized structure, the Personal Finance Tracker facilitates a detailed view of a user's spending across various categories, ensuring that users can monitor both cleared and uncleared balances, ultimately leading to a comprehensive view of their working capital.

This platform stands out as an essential tool for individuals looking to enhance their financial awareness. It not only helps users adhere to their budgets but also enables them to identify spending patterns that may require adjustment, contributing to better financial decision-making over time.

Designed with user experience in mind, the Personal Finance Tracker combines functionality with ease of use, aiming to make financial management accessible to users of all backgrounds.

Overall, the Personal Finance Tracker project highlights the importance of personal financial management in today's economic landscape. With further development, including enhancements to customization options and data security, the project has the potential to become an invaluable tool for personal finance management, catering to a wide array of user needs and preferences.

## References

1. **W3Schools. (n.d.). PHP MySQL Database.**  
Available at: [https://www.w3schools.com/php/php\\_mysql\\_intro.asp](https://www.w3schools.com/php/php_mysql_intro.asp)
2. **Mozilla Developer Network. (n.d.). JavaScript Documentation.** Available at:  
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>
3. **Stack Overflow. (n.d.). Community Forums for Coding Queries and Solutions.** Available at:  
<https://stackoverflow.com/>
4. **PHP Manual. (n.d.). PHP Documentation and Coding Standards.** Available at:  
<https://www.php.net/manual/en/>

## Limitations

1. **Performance with Large Data Sets:** The current implementation may face performance challenges with large amounts of transaction data, as there is limited optimization for high data volume scenarios.
2. **Cross-Browser Compatibility:** Some UI components might display differently across browsers, particularly on older versions, impacting the overall user experience.
3. **Data Privacy Concerns:** While basic security measures are in place, additional work is needed to secure sensitive data, especially in cases where TrackIt may be deployed in a production environment.
4. **Mobile Responsiveness:** Although the application is designed to be responsive, certain layout elements may not display optimally on smaller screens, affecting usability on mobile devices.

## Future Enhancements

1. **Data Visualization Enhancements:** New charting options and advanced data visualizations will be integrated to provide users with deeper insights into their spending patterns. Interactive graphs, trend analysis, and monthly/annual breakdowns will make it easier to analyze financial behavior over time.
2. **Mobile Application Development:** A dedicated mobile application for Android and iOS platforms is in development. The mobile app will bring the full functionality of the web platform to users on the go, with an optimized layout for smaller screens and push notifications for transaction reminders and budget updates.
3. **Advanced Security Measures:** To address data privacy concerns, future versions will include robust security features, such as encryption, two-factor authentication, and optional biometric login. These additions aim to safeguard user information and enhance overall data security.
4. **Cross-Platform Syncing:** With future updates, users will have the ability to sync data seamlessly across multiple devices, ensuring that their financial information is accessible and up-to-date wherever they go.
5. **Automated Insights and Budget Suggestions:** The tracker will incorporate machine learning algorithms to analyze spending data and provide users with personalized recommendations for saving. Automated insights will help users identify areas to cut costs and offer suggestions for staying within budget.
6. **Integration with Financial Accounts:** Future updates may allow users to link the Personal Finance Tracker to their bank accounts and credit cards, enabling automatic transaction logging and more accurate spending tracking. This integration will reduce manual entry and improve ease of use.