A Project Report On

FinTrack : Income-Expense Tracker

Developed By

Vijay Manani

Submitted To

Geetanjali Collage of Computer Science & Commerce (BBA)

Saurashtra University,
Rajkot.
Academic Year : 2024-25

Guide Name

Acknowledgement

- I am pleased to submit my project titled "FinTrack: Income-Expense Tracker" to Saurashtra University, Rajkot as part of the requirements for the BCA degree in the Computer Applications branch.

- I extend my sincere gratitude to Prof. Brijesh Shah, Head of the Department, and all the esteemed faculty members of the Department of Computer Science for their invaluable guidance and unwavering support throughout this journey.

- I am also thankful to the distinguished authors of various books, research papers, and blogs that I referred to during the development of this project. Their works served as an essential foundation for my understanding and progress.

- A special note of appreciation goes to my parents, whose constant encouragement and support have been instrumental in not only the completion of this project but also in every endeavor I undertake in life. I am deeply grateful to my family for standing by me.

- Gratitude is a feeling that comes from the heart, and even the smallest timely help can make a big difference in life. For every piece of guidance and support I received, I extend my heartfelt thanks.

Preface

- This project report is a culmination of my efforts during the **6th semester of the Bachelor of Computer Applications (BCA) program**. It has been an enlightening experience that highlighted the difference between theoretical learning and its practical application. While theoretical knowledge lays the foundation, it is through hands-on project work that the true depth of understanding is gained.
- The project titled **FinTrack** is based on an **Income-Expense Tracker**, which I selected after careful study of the subject. This project gave me the opportunity to explore **software development in real-world scenarios** and implement the principles of software development that I have studied throughout the course.
- Working on this project has been both **challenging and rewarding**. I realized the importance of **practical training** in the field of software development, as it enables us to apply the concepts we have learned in class. The **computerized tracking system** developed in this project offers **efficiency, speed, and accuracy**, providing users with a streamlined way to manage their financial transactions.
- This project report provides an **overview of the entire development process**, from design to implementation. The report includes **various figures and technical details** that will allow anyone with a basic understanding of software development to follow the content easily. I believe this experience has **strengthened my skills** and prepared me for future endeavors in this field.
- This project was completed as part of the curriculum of the **6th semester of BCA at Geetanjali Group of Colleges**. I hope this report will serve as a useful reference for anyone interested in understanding the technical aspects of the project.

Index

Project Profile

FinTrack is a desktop-based Income-Expense Tracker designed to help users efficiently manage their financial transactions by categorizing income and expenses. Built using Python with a MySQL relational database, FinTrack offers a user-friendly interface that enables users to track, search, and analyze their financial data with ease. The system ensures accuracy, security, and accessibility, making personal finance management seamless and hassle-free.

- **Key-Features**

  - ✓ **Income Management:**
    Add, edit, and manage income records with details such as date, source, category, and amount. View total income with categorized breakdowns for better financial insights. Search transactions easily by date, category, or amount within the Income section.

  - ✓ **Expense Tracking:**
    Record daily expenses with relevant details, including category, payment method, and amount. Analyze spending patterns to improve financial decision-making. Search for expense transactions by date, category, or amount within the Expense section.

  - ✓ **Transaction Search Functionality:**
    Integrated search inside the Income and Expense pages. Find transactions quickly based on amount, date, or category, helping users analyze their financial history.

  - ✓ **Financial Reports & Insights:**
    Generate detailed income vs. expense reports for better budget planning.

- **Benefits**

  - **Better Financial Management –** Provides a structured and organized way to track income and expenses, helping users plan budgets effectively.
  - **Enhanced Accuracy & Efficiency –-** Eliminates the need for manual record-keeping, reducing errors and saving time in financial tracking.
  - **Improved Accessibility & Organization** – Centralized storage in a secure MySQL database, ensuring easy retrieval and data consistency.
  - **User-Friendly Experience** – Simple and intuitive interface, making it convenient for users to add, search, and analyze financial data effortlessly.

System Requirement

Note - To implement this project, the following system hardware and network are required.

➢ Hardware Requirements

Processor	Intel Core i3 or higher (or equivalent)
RAM	Minimum 4 GB

➢ Software Requirements

Operating System	Windows 7 or higher
Programming Language	Python
Database	SQLite
IDE	Visual Studio Code
Framework	Tkinter (for GUI)

Platform Specification

- **Development Framework:**

  The FinTrack: Income-Expense Tracker will be developed as a standalone desktop application using Tkinter, Python's built-in GUI framework. Tkinter provides a lightweight, responsive, and customizable environment for building intuitive and interactive desktop applications, ensuring a smooth user experience for financial tracking.

- **Programming Language:**
  The application will be coded in Python, a versatile and widely used programming language. Python's simplicity, strong community support, and extensive libraries make it an ideal choice for developing finance-related applications, ensuring scalability, maintainability, and efficient performance.

- **Database Management System (DBMS):**

  The system will use MySQL as the relational database management system (RDBMS). MySQL is known for its efficiency, reliability, and easy integration with Python through MySQL connectors. It provides a secure and structured way to store and manage financial transactions, ensuring fast and accurate data retrieval.

- **Desktop Application Framework:**

  The Tkinter library will be used as the primary UI framework for developing the application. Tkinter offers a simple yet powerful way to create user-friendly interfaces, allowing users to efficiently add, search, and analyze their financial transactions with ease.

- **Deployment Environment:**

  The FinTrack application will be deployed on Windows-based systems, ensuring broad compatibility. It can be installed and run locally on individual desktops, making it a secure and standalone solution for personal financial management.

Data Dictionary

- **Users Table**

Sr. No	Field Name	Data Type	Constraints
1	id	INTEGER	PRIMARY KEY, AUTOINCREMENT
2	name	TEXT	NOT NULL
3	email	TEXT	UNIQUE, NOT NULL
4	contact	TEXT	UNIQUE, NOT NULL
5	password	TEXT	NOT NULL

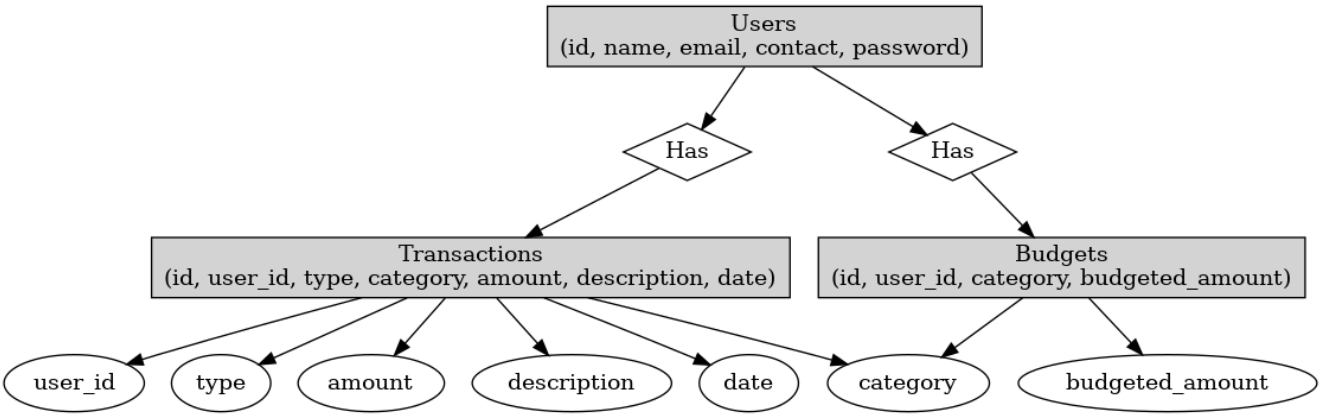
- **Budgets Table**

Sr. No	Field Name	Data Type	Constraints
1	id	INTEGER	PRIMARY KEY, AUTOINCREMENT
2	user_id	INTEGER	FOREIGN KEY REFERENCES users(id), NOT NULL
3	category	TEXT	NOT NULL
4	budgeted_amount	REAL	NOT NULL

- **Transactions Table**

Sr. No	Field Name	Data Type	Constraints
1	id	INTEGER	PRIMARY KEY, AUTOINCREMENT
2	user_id	INTEGER	FOREIGN KEY REFERENCES users(id), NOT NULL
3	type	TEXT	NOT NULL (Values: 'income' or 'expense')
4	category	TEXT	NOT NULL
5	amount	REAL	NOT NULL
6	description	TEXT	NULL
7	date	DATE	NOT NULL

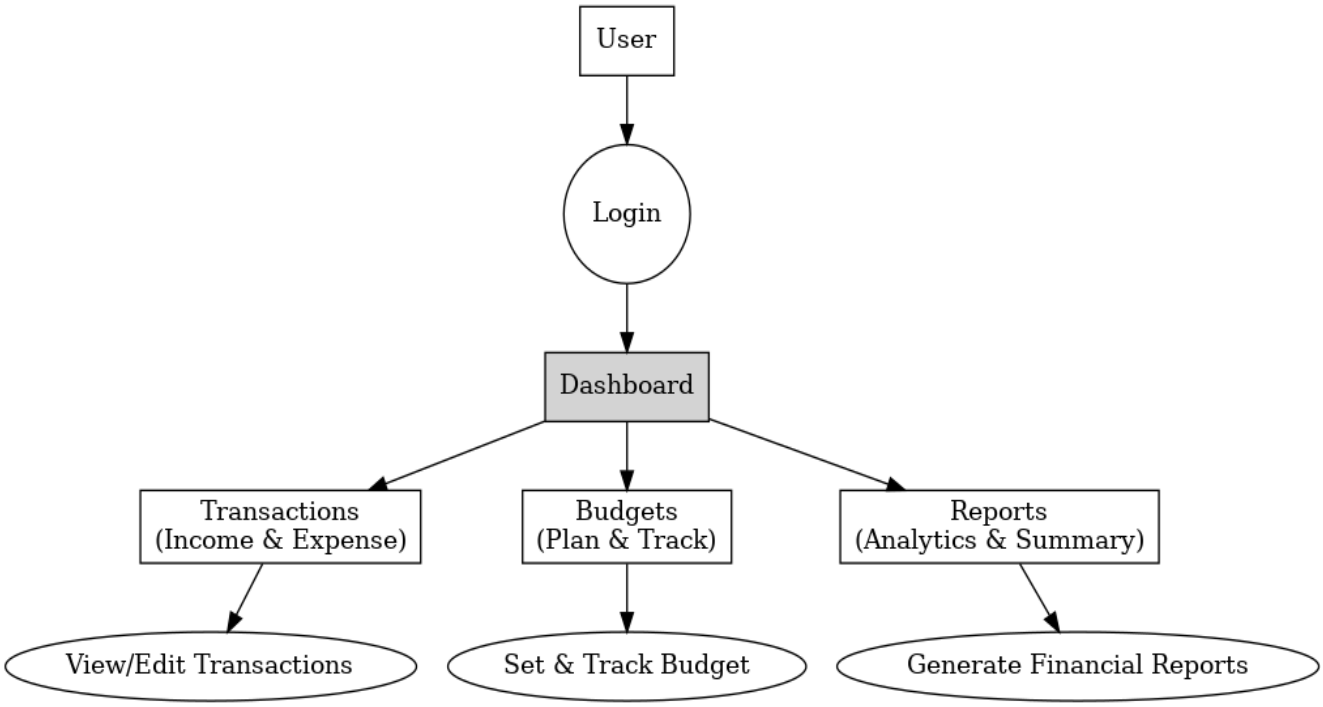
ER Diagrams



- The provided ER diagram outlines the structure of a financial tracking system with three main entities: Users, Transactions, and Budgets. The Users entity stores user details like id, name, email, contact, and password.
- The Transactions entity records financial activities with attributes such as id, user_id, type (income/expense), category, amount, description, and date.
- The Budgets entity manages user budgets with id, user_id, category, and budgeted_amount.
- The relationship Has connects users to their Transactions and Budgets, ensuring each user can have multiple transactions and budgets. This structure supports efficient tracking of finances and budgeting for users.

Data Flow Diagram

- **Context-Level DFD**



SDLC

- **Requirement Gathering and Analysis**

  Understand the core functionalities needed for the income-expense tracking system. Identify key features such as income and expense management, category-wise tracking, transaction search, budget management, user authentication, and report generation.

- **Feasibility Study**

  Evaluate the technical, operational, and financial feasibility of developing FinTrack using Python and SQLite. Ensure that the system can efficiently handle transactions and provide accurate financial insights.

- **System Design**

  Design the architecture, database schema, and user interface for FinTrack. Define the structure for storing transactions, user details, and budget information. Plan for intuitive navigation and an easy-to-use UI for financial tracking.

- **Development**

  Implement the core functionalities based on the design specifications. Develop the GUI using Python frameworks (such as Tkinter). Implement business logic for adding, updating, and deleting transactions. Integrate SQLite for securely storing transaction data with relevant constraints and relationships.

- **Testing**

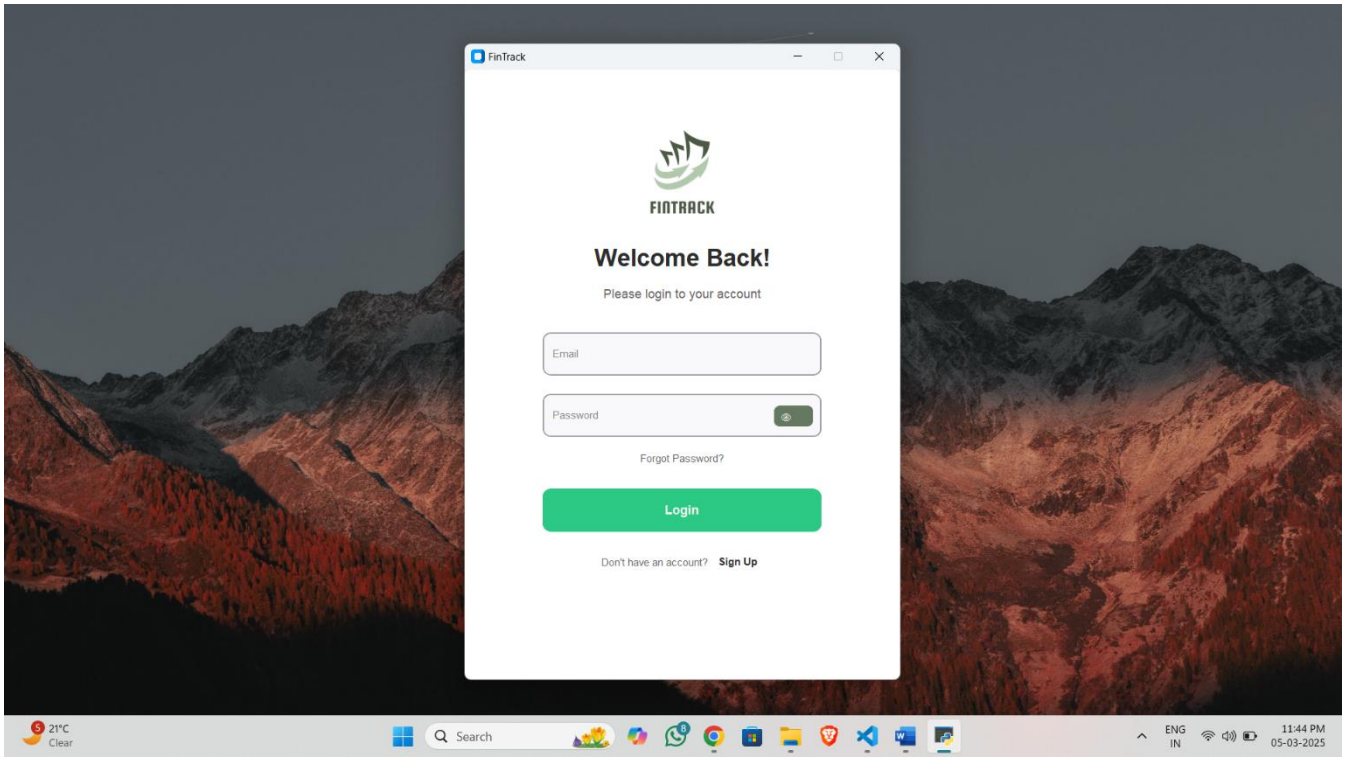
  Ensure that the system functions correctly, is secure, and meets the requirements.

- **Deployment**

  Deploy FinTrack as a standalone desktop application. Ensure it is optimized for local database storage, enabling smooth performance for users managing their financial records.

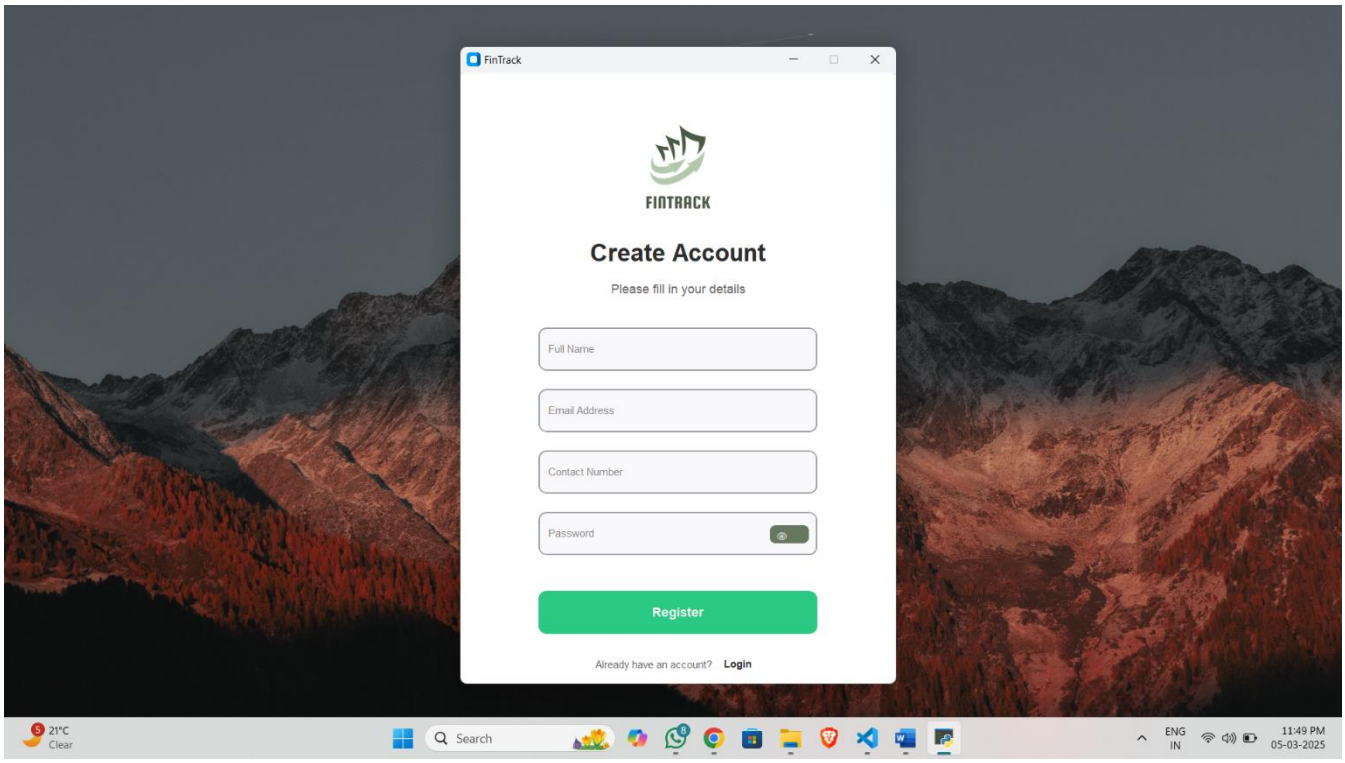
Screen Shots

- **Login Page**



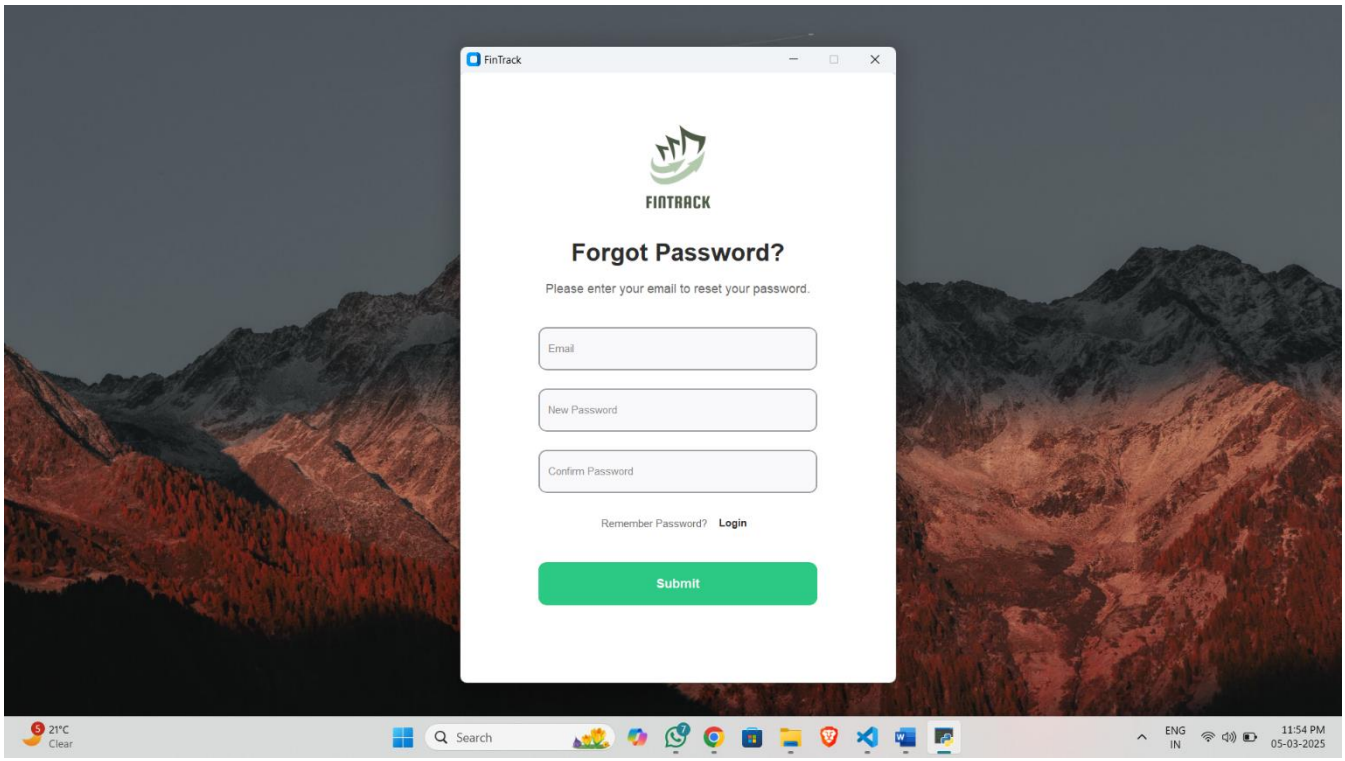
- **User can log into the software by entering their Email and password.**
- **If the entered credentials are correct, the user will successfully gain access to the system.**
- **User has the option to view the entered password by selecting the "Show Password" checkbox.**

- **Sign Up Page**



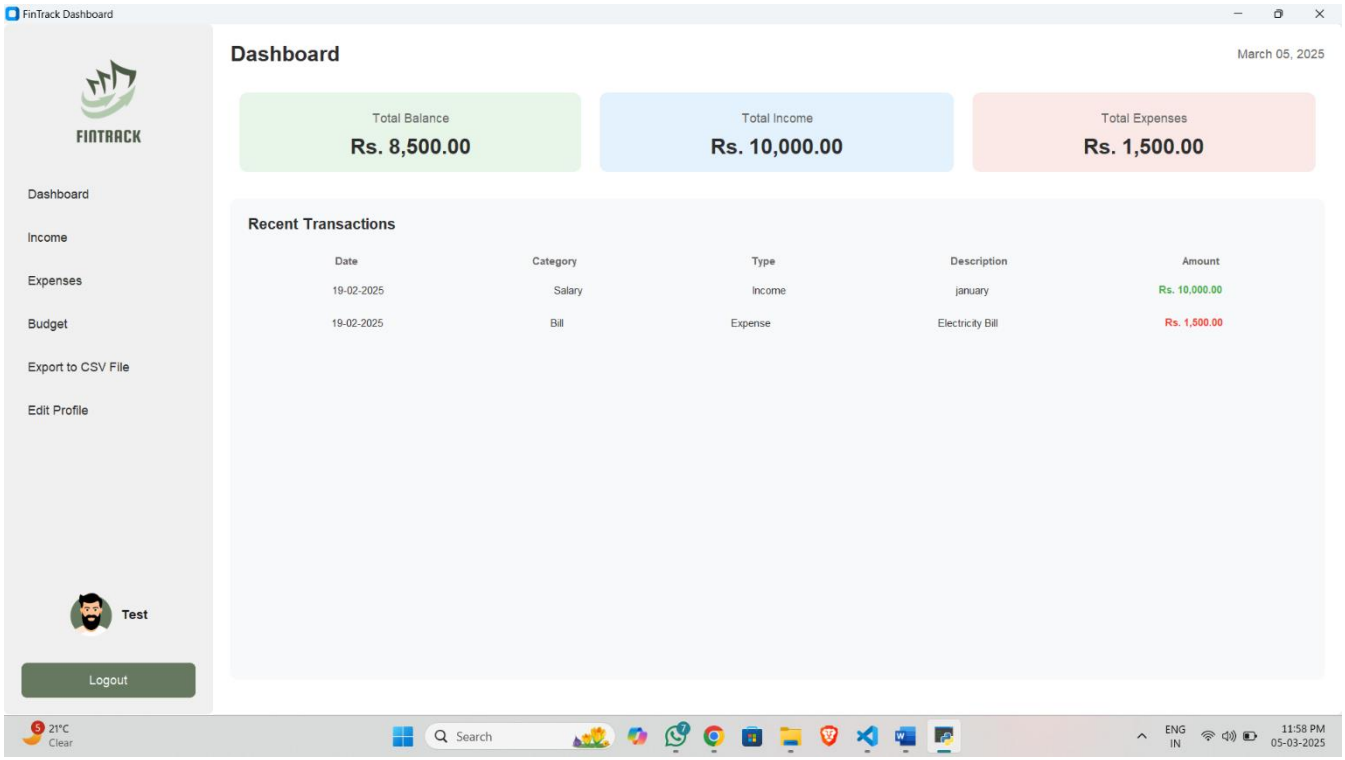
- **If the user is logging in for the first time, they must create an account first. After registration, they can log in using their email and password.**

- **Forget Password Page**



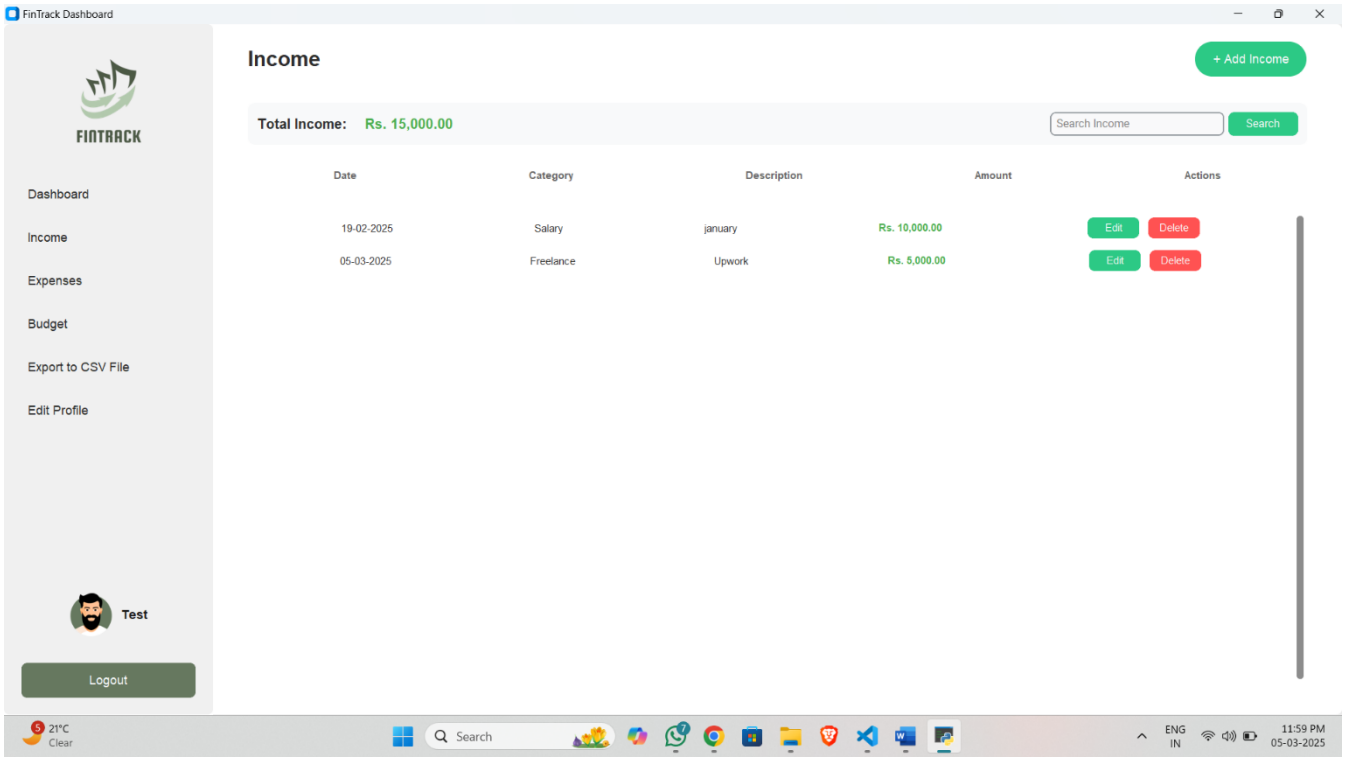
- **If the user forgets their password, they can reset it by selecting the "Forgot Password" option and following the instructions to create a new password.**

- **Dashboard Page**



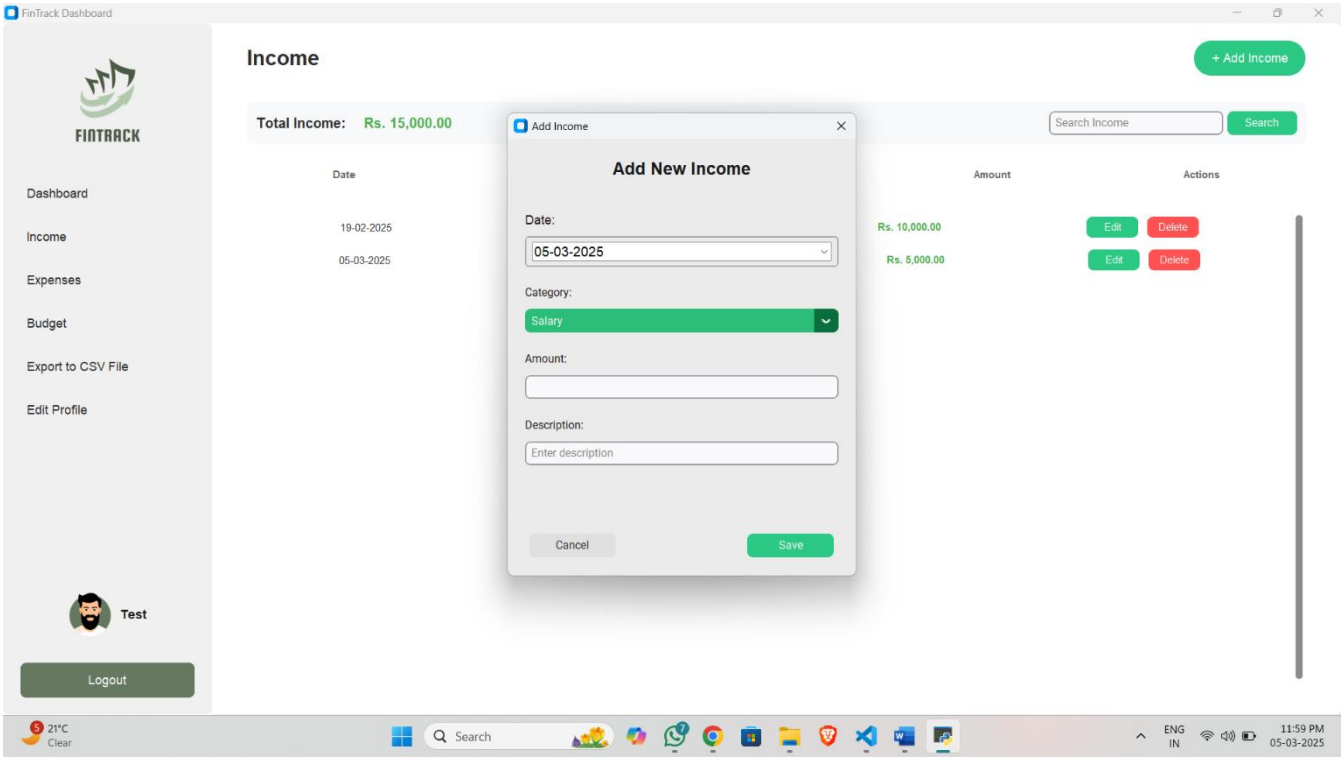
- **In Dashboard Overview:** The dashboard provides a summary of the user's financial status, displaying Total Balance, Total Income, and Total Expenses in separate sections for clear visibility.
- **Recent Transactions:** Users can view a list of their latest transactions, including Date, Category, Type (Income/Expense), Description, and Amount, with income amounts shown in green and expenses in red.
- **Navigation Panel:** The left sidebar allows users to navigate between different sections, including Income, Expenses, Budget, Edit Profile and , Export to CSV File allows to user convert data into csv file.
- **User Profile & Logout:** The bottom-left section displays the logged-in user's profile information, with an option to log out securely.

- **Income Page**



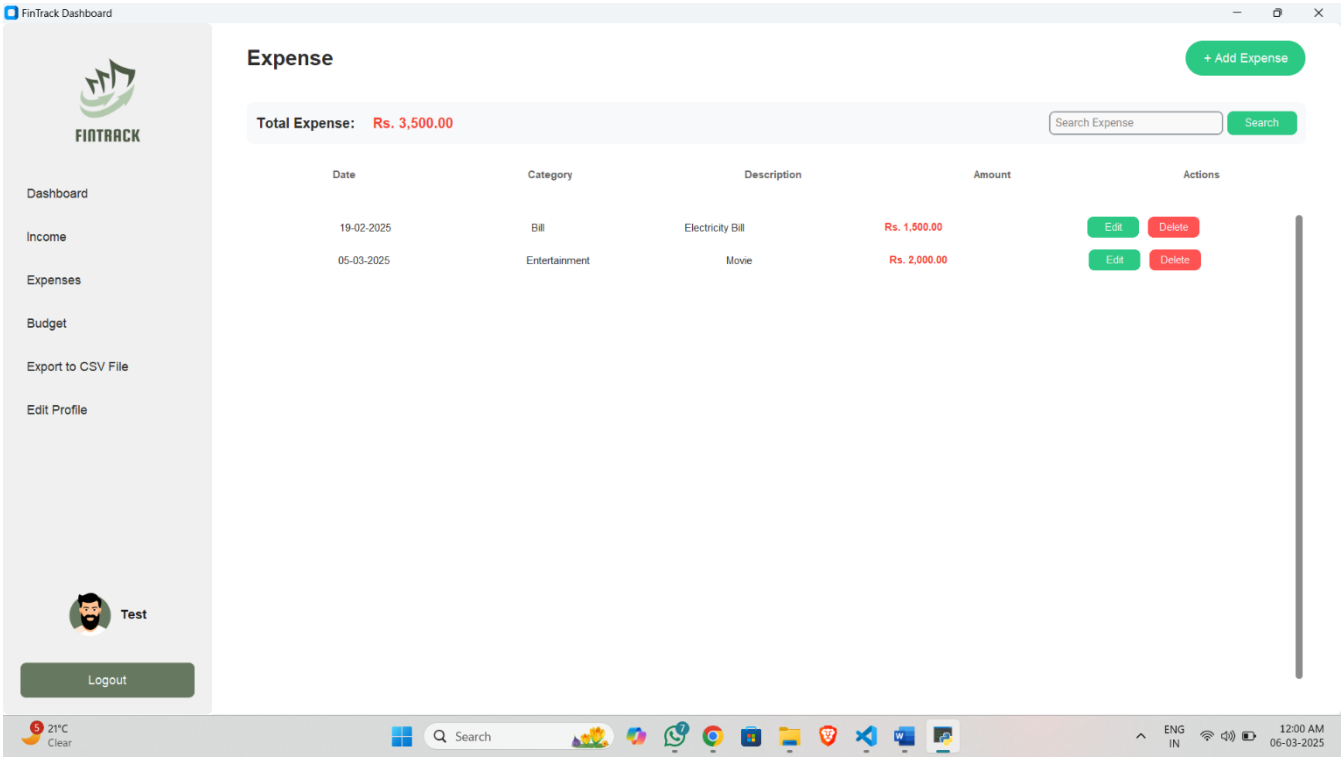
- **In the Income Page, users can manage their income records efficiently. The total income amount is displayed at the top, summarizing all recorded transactions. Users can add new income by clicking the "+ Add Income" button, where they can enter details such as date, category, description, and amount.**
- **A search functionality allows users to filter income records based on keywords, including category, description, and amount.**
- **Users can edit or delete existing income entries using the respective "Edit" and "Delete" buttons under the Actions column.**

- **Income Page > Add Income**



- **In the Add Income Page, users can enter new income records by providing details such as date, category, amount, and description.**
- **The page provides two action buttons: "Save" to add the income entry to the system and "Cancel" to discard the input and close the window.**

- **Expense Page**

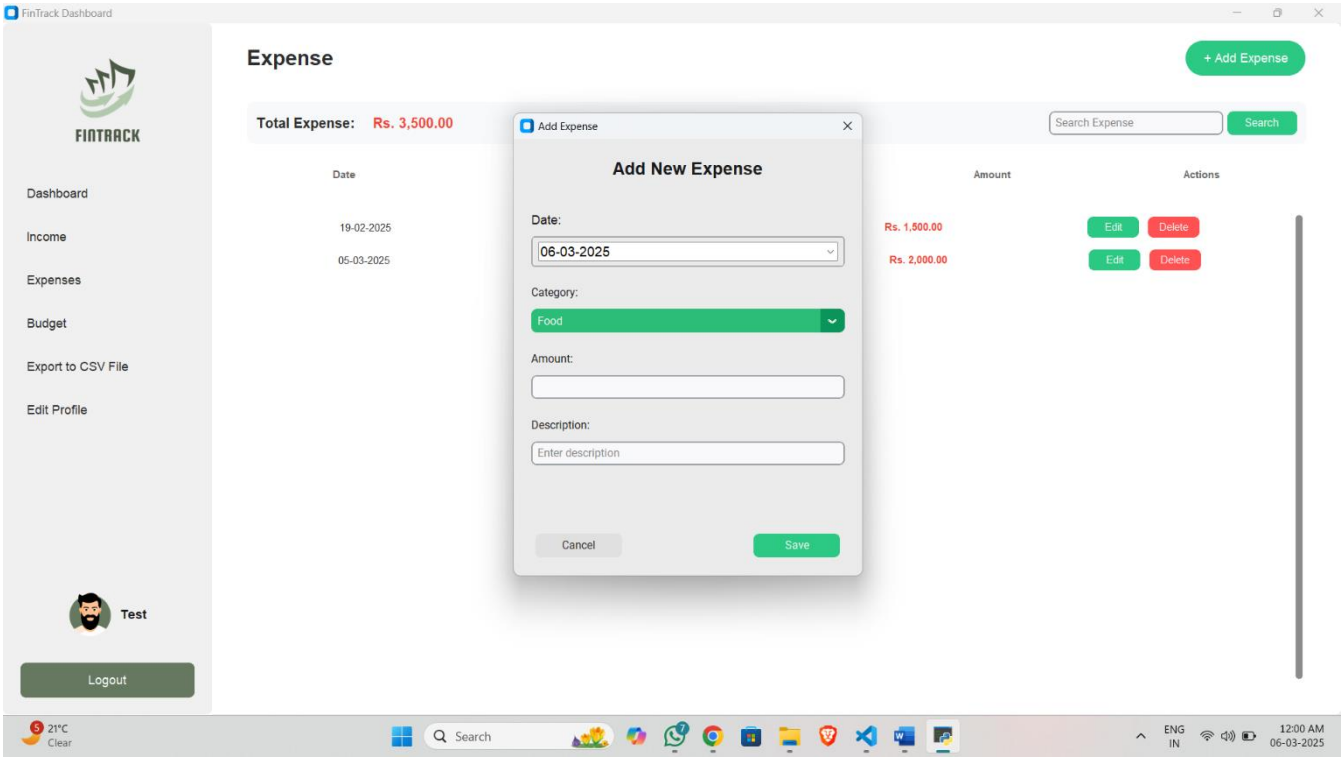


- **In the Expense Page, users can manage their expense records efficiently. The total expense amount is displayed at the top, summarizing all recorded transactions. Users can add new expenses by clicking the "+ Add Expense" button, where they can enter details such as date, category, description, and amount.**

- **A search functionality allows users to filter expense records based on keywords, including category, description, and amount.**

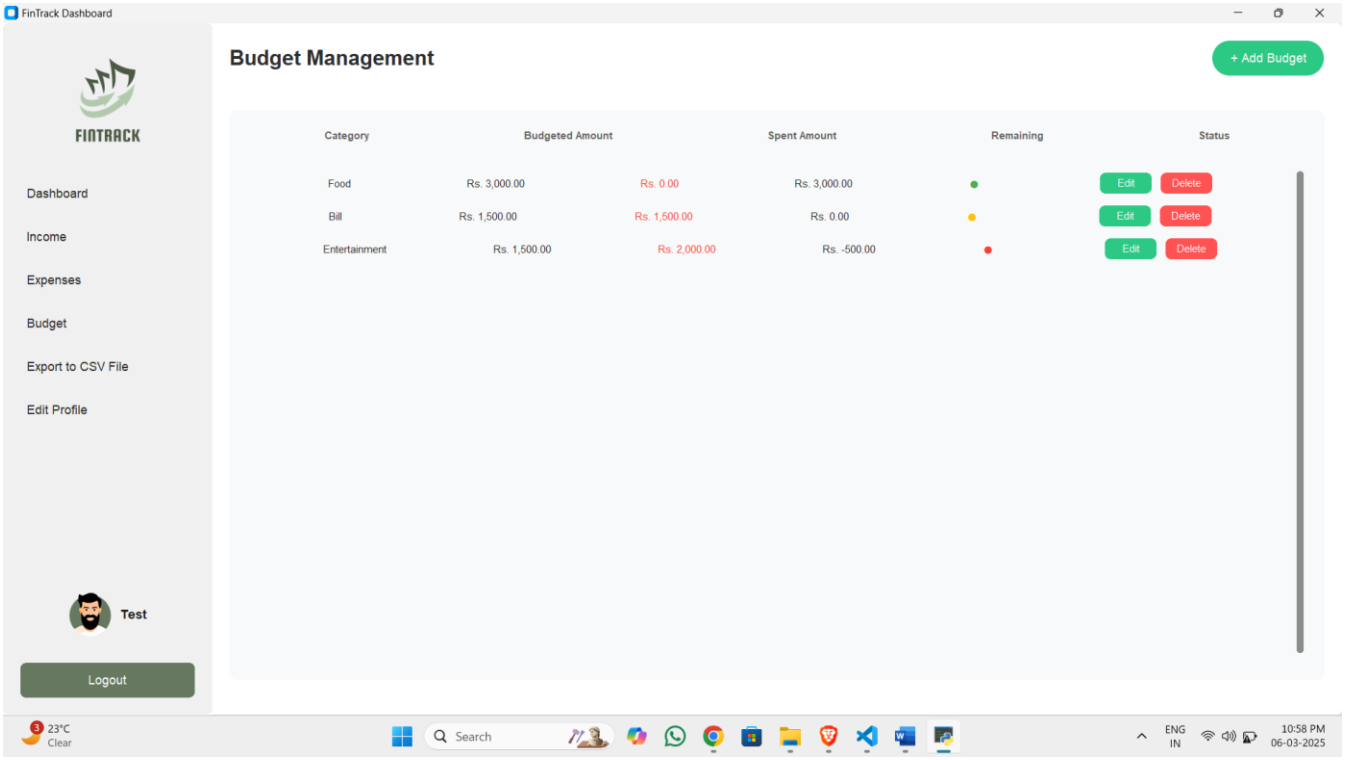
- **Users can edit or delete existing expense entries using the respective "Edit" and "Delete" buttons under the Actions column.**

- **Expense Page > Add Expense**



- **In the Add Expense Page, users can enter new expense records by providing details such as date, category, amount, and description.**
- **The page provides two action buttons: "Save" to add the income entry to the system and "Cancel" to discard the input and close the window.**

- **Budget Page**

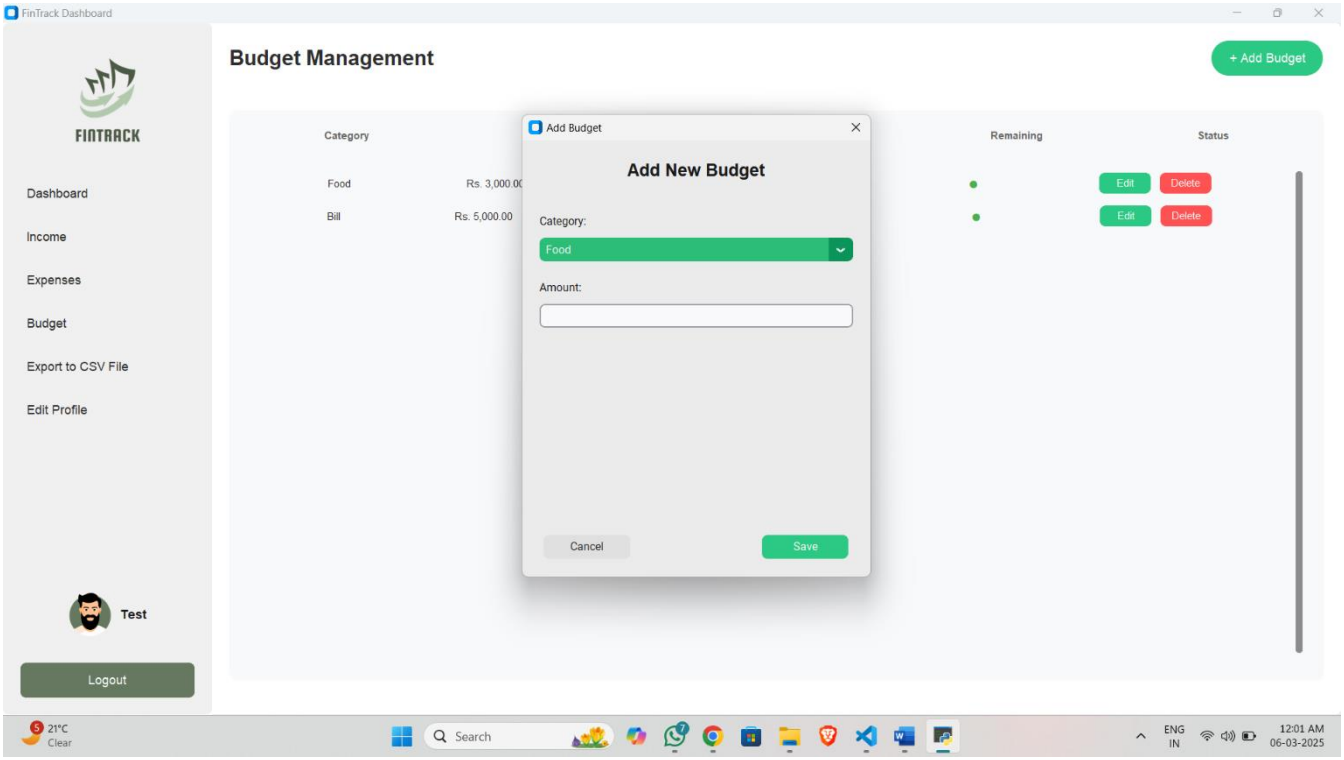


- **In the Budget Management Page, users can track and manage their budgets for different expense categories. The page displays a summary of budgeted amounts, spent amounts, and remaining balances for each category.**

- **A The Budget Management page includes colored status indicators: green for budgets within the limit, yellow for those nearing the limit, and red for overspent budgets.**

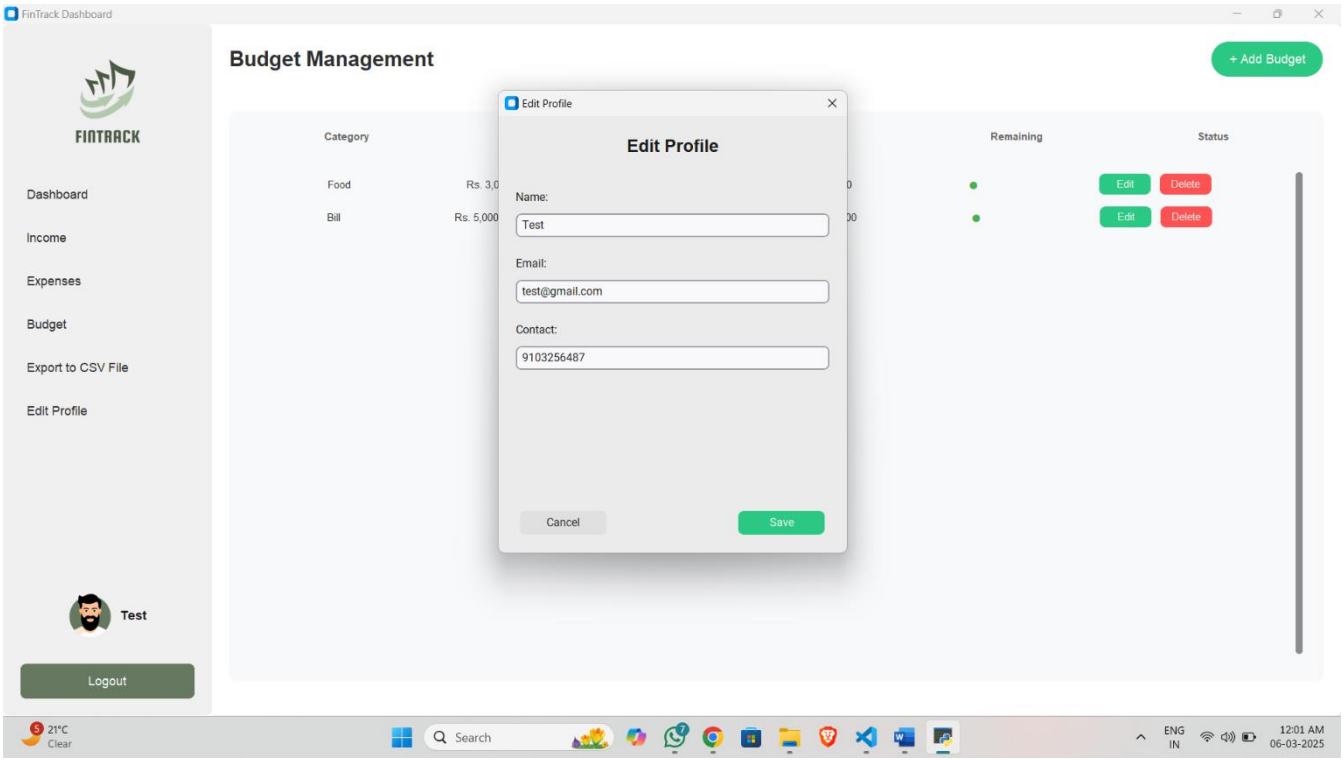
- **Users have the option to edit or delete budget records using the respective "Edit" and "Delete" buttons under the Status column.**

- **Budget Page > Add Budget**



- **The "Add New Budget" allows users new budget. The form allows them to select a category from a dropdown and enter the budgeted amount.**
- **It includes "Save" and "Cancel" buttons for confirming or discarding the entry.**

- **Edit Profile Page**



- **The "Edit Profile" popup window in the FinTrack Dashboard allows users to update their personal information.**

- **It contains three input fields for Name, Email, and Contact, which are pre-filled with existing user data. Below these fields, there are two buttons: "Cancel" to discard changes and "Save" to update the profile details.**

FUTURE REQUIREMENTS OF PROJECT

Anticipating future requirements for the **FinTrack** project is crucial to ensuring its long-term effectiveness and adaptability. Future requirements may include:

- **Mobile Application Support**:

  Develop a mobile application for Android and iOS to provide users with on-the-go access to their financial records, including income tracking, expense management, and budget insights.

- **Advanced Search and Filters:**

  Enhance the search functionality by allowing users to filter transactions by date range, category, description, and amount, making it easier to track specific transactions efficiently.

- **AI-Based Expense Analysis:**

  Integrate AI-driven insights to predict future expenses, detect spending patterns, and offer personalized budgeting recommendations to help users manage their finances better.

- **Graphical Reports and Dashboard Enhancements:**

  Introduce interactive visualizations such as bar charts, pie charts, and line graphs to help users better understand their financial trends over time. Provide monthly, quarterly, and yearly reports for deeper financial insights.

- **Integration with Bank Accounts and UPI:**

  Enable integration with bank accounts, UPI, and e-wallets to automate transaction tracking and categorize expenses automatically, reducing manual data entry.

- **Multi-Currency Support:**

  Support multiple currencies to allow users to track income and expenses in different regions, with real-time currency conversion based on exchange rates.

LIMITATION OF PROJECT

- **No Built-in Web Support**:

  Currently, FinTrack is a desktop-based application, which limits accessibility from different devices. Users cannot access their financial data through a web browser unless a web version is developed.

- **Limited Scalability**:

  As the number of transactions increases, database performance may slow down if not optimized properly. Handling a large volume of income and expense records may require future enhancements, such as indexing, caching, or database sharding.

- **Lack of Cloud Synchronization**:

  Since FinTrack is currently designed as an offline system, users cannot sync their data across multiple devices. If a user switches devices, they need to manually transfer their financial records, which can be inconvenient.

- **Security Concerns**:

  Since the application operates locally, data security is dependent on the user's device. There is no built-in encryption or cloud-based backup, which could pose risks if the system is compromised or the device is lost.

- **No Direct Bank Integration**:

  FinTrack does not currently support automatic bank transaction imports or UPI integration. Users must manually input their income and expenses, which could be time-consuming.

Test Cases

NO.	Test Case	Description	Precondition	Expected Result	Result
1.	User Login with Correct Credentials	User logs into the system by entering the correct email and password.	User credentials (email and password) exist in the database.	User is successfully logged in and redirected to the dashboard.	Pass
2.	User Login with Incorrect Credentials	User enters incorrect email or password.	User credentials (email and password) do not match any records in the database.	An error message appears indicating invalid login credentials.	Pass
3.	Add Income Record	User adds a new income entry using the "Add Income" button.	User is logged in and on the income page.	The new income record is successfully added and displayed in the income list.	Pass
4.	Edit Income Record	User edits an existing income record.	An income record exists in the database.	The income record is updated and reflected in the income list.	Pass
5.	Delete Income Record	User deletes an existing income record.	An income record exists in the database.	The income record is removed from the income list.	Pass
6.	Add Expense Record	User adds a new expense entry using the "Add Expense" button.	User is logged in and on the expense page.	The new expense record is successfully added and displayed in the expense list.	Pass
7.	Edit Expense Record	User edits an existing expense record.	An expense record exists in the database.	The expense record is updated and reflected in the expense list.	Pass
8.	Delete Expense Record	User deletes an existing expense record.	An expense record exists in the database.	The expense record is removed from the expense list.	Pass
9.	Add Budget	User adds a new budget using the "Add Budget" button.	User is logged in and on the budget page.	The new budget is successfully added and displayed in the budget list.	Pass

10.	Edit Budget	User edits an existing budget.	A budget exists in the database.	The budget is updated and reflected in the budget list.	Pass
11.	Delete Budget	User deletes an existing budget.	A budget exists in the database.	The budget is removed from the budget list.	Pass
12.	Export Transactions to CSV	User exports transactions to a CSV file.	At least one transaction exists in the database.	A CSV file containing the transactions is successfully saved to the specified location.	Pass
13.	User Registration with Valid Details	User registers with valid name, email, contact, and password.	No user with the same email or contact exists in the database.	User is successfully registered and redirected to the login page.	Pass
14.	User Registration with Existing Email	User attempts to register with an email that already exists.	A user with the same email exists in the database.	An error message appears indicating that the email already exists.	Pass
15.	User Registration with Existing Contact	User attempts to register with a contact number that already exists.	A user with the same contact number exists in the database.	An error message appears indicating that the contact number already exists.	Pass
16.	Reset Password with Valid Email	User resets password using a valid email.	A user with the provided email exists in the database.	Password is successfully reset, and the user is redirected to the login page.	Pass
17.	Reset Password with Invalid Email	User attempts to reset password using an invalid email.	No user with the provided email exists in the database.	An error message appears indicating that the email is not found.	Pass
18.	Add Income with Invalid Date Format	User attempts to add an income record with an invalid date format.	User is logged in and on the income page.	An error message appears indicating an invalid date format.	Pass
19.	Add Expense with Negative Amount	User attempts to add an expense record with a negative amount.	User is logged in and on the expense page.	An error message appears indicating that the amount must be greater than zero.	Pass

BIBLIOGRAPHY

- **UI Colors**

  https://colorhunt.co/palette/

- **Icons**

  https://www.flaticon.com/

- **Logo**

  https://www.canva.com/