

Project Report Format

1. INTRODUCTION

1.1 Project Overview

This project is a modern **Expense Tracking Web App** designed to help users such as students, small business owners, and everyday individual to track, manage, and optimize their spending habits.

Built with a sleek and responsive frontend using React, Tailwind CSS, and Redux Toolkit, the app allows users to categorize their income and expenses, visualize financial summaries through interactive charts, and securely manage their profile information.

Key features include:

- Authentication (Login/Register)
- Transaction Management (Add, filter, view)
- Real-time Charts (Income vs Expense)
- Category Management (Custom types and names)
- Profile Updates & Password Change

The backend (built with Node.js and Express) handles user data securely with JWT-based authentication and connects to a database (MongoDB) to store transaction history and category data.

1.2 Purpose

The purpose of this project is to:

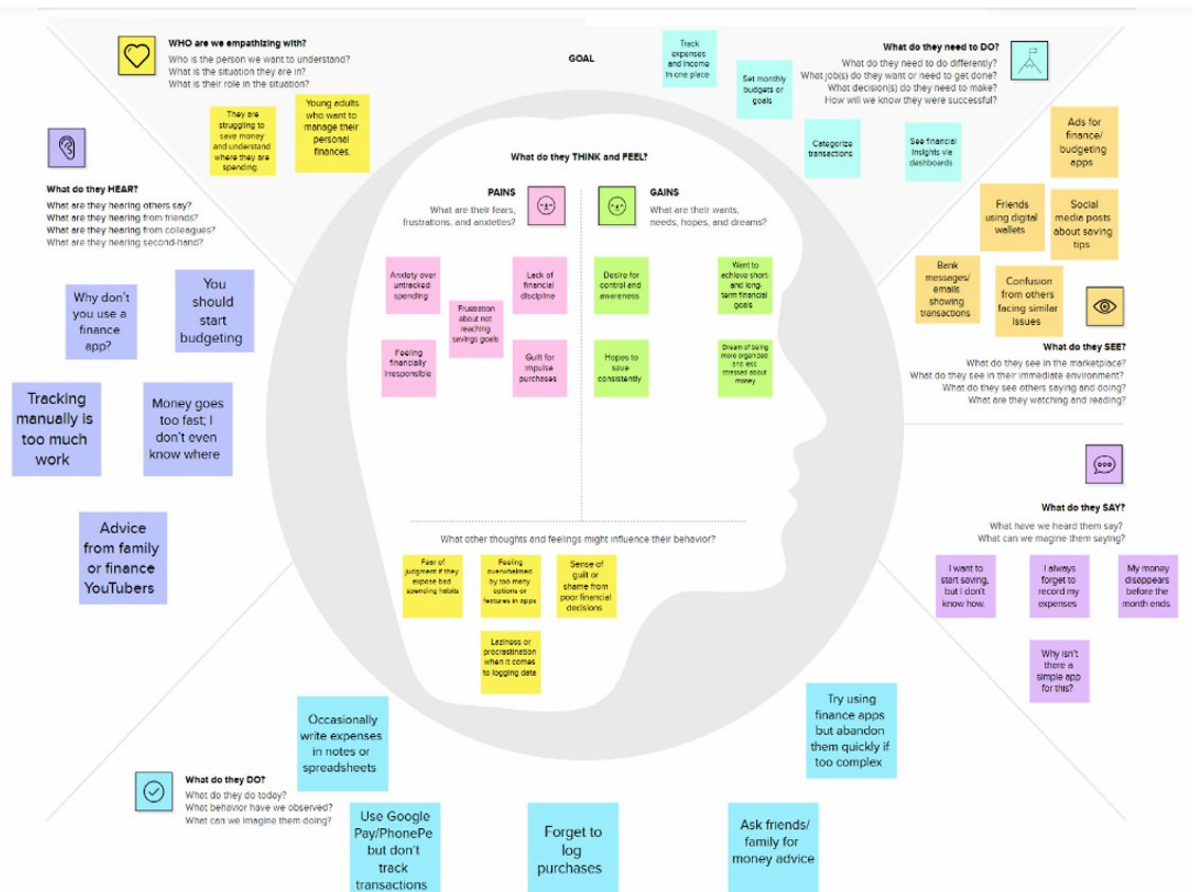
- Empower users with financial awareness by offering a simple and visual way to track where their money goes.
- Provide a centralized and secure platform for recording income and expenses in real-time.
- Replace manual spreadsheet tracking with an intelligent and intuitive dashboard.
- Offer a customizable, extensible base for future integrations (e.g., bank APIs, spending recommendations, multi-currency support).
- Serve as a learning and portfolio project demonstrating full-stack development skills, API integration, state management, and UI/UX best practices.

2. IDEATION PHASE

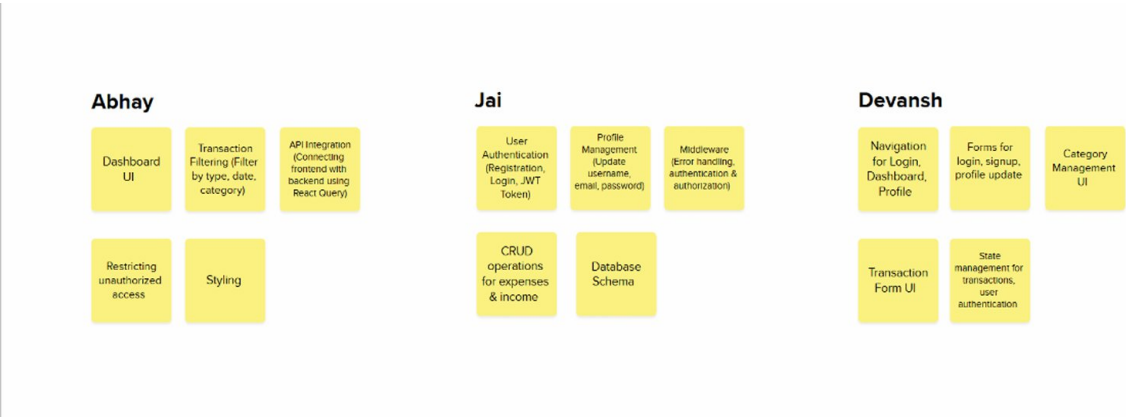
2.1 Problem Statement

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	I am a college student.	I'm trying to track my daily income and expenses to better manage my limited budget.	But I often forget to note down my transactions or lose track of spending.	Because no easy, centralized, and user-friendly system to remind me and show my spending clearly.	Which makes me feel stressed, disorganized, and unaware of my financial habits.
PS-2	I am a working professional living away from home.	I'm trying to save money and analyze where most of my salary is going	But it's difficult to categorize expenses or compare income vs. spending trends over time.	Because I don't have an intuitive dashboard or automated tracking tool.	Which makes me feel worried and unsure about my financial planning and savings goals.

2.2 Empathy Map Canvas

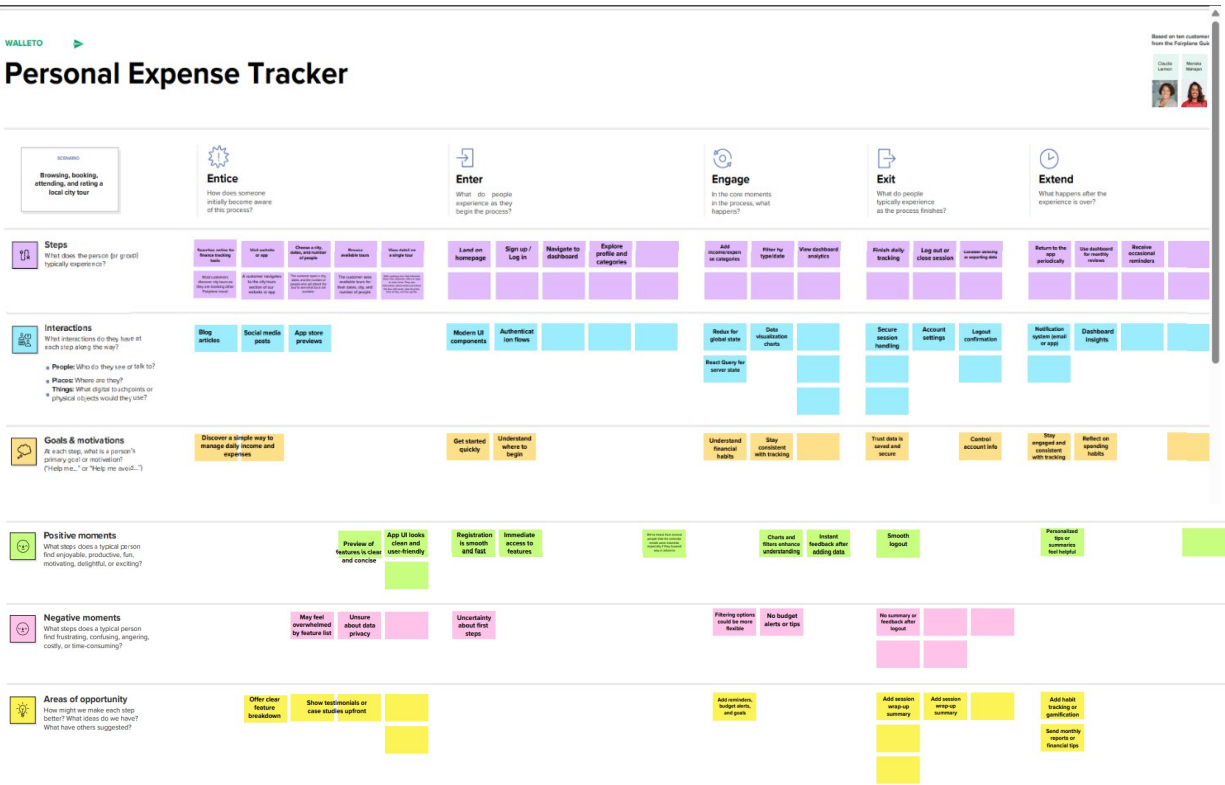


2.3 Brainstorming



3. REQUIREMENT ANALYSIS

3.1 Customer Journey map



3.2 Solution Requirement

Functional Requirements:

The following are the functional requirements of the proposed solution.

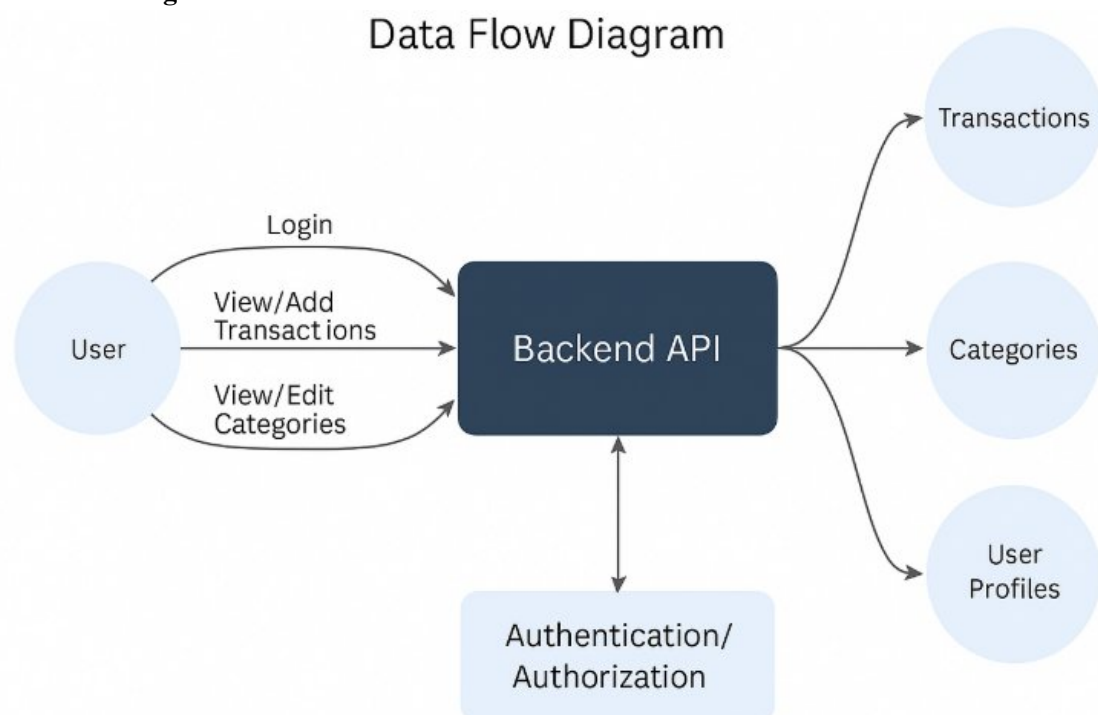
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Expense Management	Add Expense Edit/Delete Expense Categorize Expense View Monthly Summary
FR-4	Budget & Analytics	Set Monthly Budget Track Budget Usage Generate Graphs for Spending Trends

Non-functional Requirements:

The following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The system must provide a user-friendly interface for easy navigation.
NFR-2	Security	The system must implement secure login, encrypted data storage, and token-based authentication.
NFR-3	Reliability	The application should perform consistently and without failure under expected usage.
NFR-4	Performance	The app must handle up to 100 concurrent users with minimal response time (< 2 seconds).
NFR-5	Availability	The system should be accessible 99.9% of the time, excluding maintenance.
NFR-6	Scalability	The system should support future growth in users, features, and data volume without performance degradation

3.3 Data Flow Diagram



3.4 Technology Stack

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / React JS / Tailwind CSS

2.	Application Logic-1	Logic for adding, editing, and deleting expense entries	Node.js / Express.js
3.	Application Logic-2	Logic for managing user authentication (login, register)	JWT, bcrypt, Express.js
4.	Application Logic-3	Logic for category-wise filtering, search, and monthly limits	Node.js / Express.js
5.	Database	stores expense records, user info categories	MongoDB
6.	Cloud Database	Database Service on Cloud	MongoDB Atlas
7.	File Storage	Storing profile pictures or attachments (if any)	Cloudinary / Firebase Storage
8.	External API-1	Currency exchange rates (if supporting multi-currency)	ExchangeRate-API
9.	External API-2	Optional: Reminder API or email alert integration	SendGrid API

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Utilized open-source frameworks for both frontend and backend	React.js, Node.js, Express.js, MongoDB
2.	Security Implementations	Implemented user authentication and data protection mechanisms	JWT, bcrypt, HTTPS, SHA-256, OWASP guidelines
3.	Scalable Architecture	Modular codebase with clear separation of concerns; RESTful API design	MERN stack
4.	Availability	Deployed on reliable cloud services with automatic redeployments and versioning support	Vercel / Render / GitHub Actions
5.	Performance	Optimized data fetch using pagination, implemented caching at the client side	Browser Cache, Axios, Lazy Loading in React

4. PROJECT DESIGN

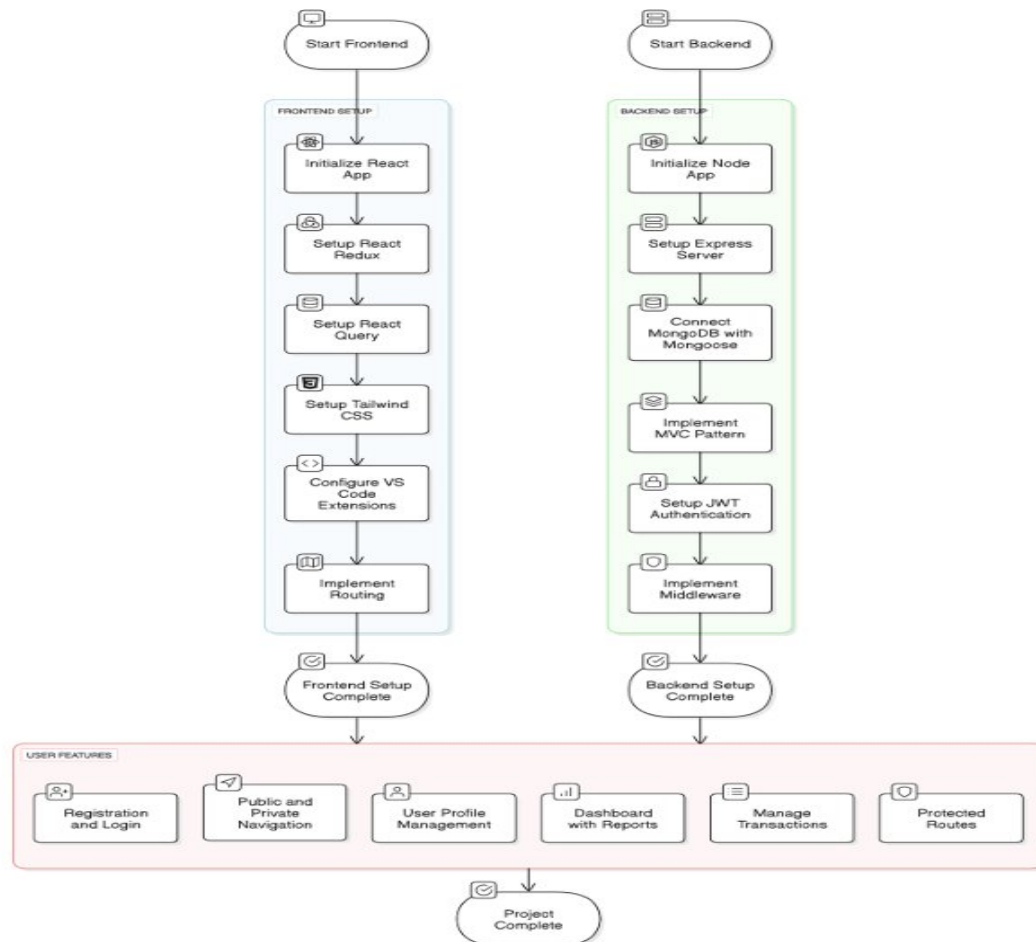
4.1 Problem Solution Fit

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) <ul style="list-style-type: none"> College students Working professionals Freelancers and small business owners 	CS	6. CUSTOMER CONSTRAINTS <ul style="list-style-type: none"> Low technical skills Limited budget (prefer free or freemium apps) Poor network connectivity in some areas Use of only mobile or browser-based tools 	CC	5. AVAILABLE SOLUTIONS <ul style="list-style-type: none"> Excel/Google Sheets (manual, error-prone) Mobile finance apps (limited features in free tier) Accounting software (complex for personal use) 	AS	Explore AS, differentiate
Focus on J&P, tap into BE, understand RC	2. JOBS-TO-BE-DONE / PROBLEMS <ul style="list-style-type: none"> Track income and expenses regularly Categorize transactions for clarity Understand monthly spending behavior Visualize savings/investments for budgeting 	J&P	9. PROBLEM ROOT CAUSE <ul style="list-style-type: none"> Lack of simple, accessible, and secure personal finance tools People don't track expenses until it becomes urgent Existing tools are either too complex or too limited 	RC	7. BEHAVIOUR <ul style="list-style-type: none"> Keeps receipts or notes transactions manually Checks bank SMS/email statements Occasionally logs expenses in mobile apps Looks for "quick budget fix" YouTube videos 	BE	Focus on J&P, tap into BE, understand RC
Identify strong TR & EM	3. TRIGGERS <ul style="list-style-type: none"> Sudden realization of overspending Personal goals like saving for a trip or loan repayment Recommendations from peers or influencers on finance tracking Awareness from news/articles on managing personal finances 	TR	10. YOUR SOLUTION <ul style="list-style-type: none"> A MERN-based web app that: <ul style="list-style-type: none"> Enables secure sign-up and login Allows categorized transaction entries Offers real-time filtering and dashboards Uses React Query + Redux for performance Styled with Tailwind CSS for ease and accessibility Protects routes and data using JWT and auth middleware 	SL	8. CHANNELS of BEHAVIOUR <ul style="list-style-type: none"> 8.1 ONLINE: <ul style="list-style-type: none"> Watches YouTube tutorials on budgeting Searches for "best budget tracking tools" Uses browser-based tools (Chrome, mobile web) 8.2 OFFLINE: <ul style="list-style-type: none"> Discusses money habits with friends/family Jobs expenses in diary/notebook Uses printed bank statements for reconciliation 	CH	Extract online & offline CH of BE
	4. EMOTIONS: BEFORE / AFTER <ul style="list-style-type: none"> Before: Anxious, confused, disorganized After: Confident, in control, financially aware 	EM					

4.2 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Managing personal finances manually is difficult and inefficient. Users struggle to keep track of income, expenses, and budgeting, which may lead to poor financial decisions and lack of awareness about their financial health.
2.	Idea / Solution description	This project proposes a full-stack web application using the MERN stack (MongoDB, Express.js, React.js, Node.js). It allows users to register, log in, and manage their financial transactions. It supports categorization, filtering, insightful reports via a dashboard, and secure user data handling using JWT.
3.	Novelty / Uniqueness	The use of React Redux for global state and React Query for server state adds performance efficiency. Tailwind CSS provides a modern, responsive UI. The app also includes protected routes, authenticated dashboards, and a visually engaging analytics display for personal finance tracking.
4.	Social Impact / Customer Satisfaction	Helps individuals and small businesses improve their financial literacy, planning, and savings by offering a simple and interactive tool to monitor expenses and income. Provides peace of mind with secure access, intuitive UI, and a reliable platform for financial tracking.
5.	Business Model (Revenue Model)	Freemium model: basic expense tracking features are free. Premium plans can offer multi-device sync, export to Excel/PDF, personalized insights, ad-free experience, and budget alerts. Additional monetization options include targeted financial product advertisements.
6.	Scalability of the Solution	The app can be scaled to support multiple users using cloud-based MongoDB (e.g., Atlas), containerized backend with Docker, and horizontal scaling with load balancers. The architecture supports mobile extension, PWA development, and enterprise-level financial modules for small businesses.

4.3 Solution Architecture



5. PROJECT PLANNING & SCHEDULING


5.1 Project Planning

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my name, email, and password.	2	High	Jai
Sprint-1	Registration	USN-2	As a user, I will receive a confirmation email after successful registration.	1	High	Abhay
Sprint-1	Login	USN-3	As a user, I can log into the application using my registered email and password.	2	High	Devansh
Sprint-2	Expense Input	USN-4	As a user, I can add a new expense by entering the category, amount, and date.	2	High	Jai
Sprint-2	Expense Input	USN-5	As a user, I can categorize my expenses (e.g., food, travel, utilities, etc.).	2	Medium	Abhay
Sprint-2	Dashboard	USN-6	As a user, I can view a dashboard showing my total expenses and a pie chart of categories.	2	High	Devansh
Sprint-3	Budget Setting	USN-7	As a user, I can set a monthly budget and get alerts when I'm about to exceed it.	2	Medium	Jai
Sprint-3	Report Generation	USN-8	As a user, I can generate monthly reports of my expenses in PDF format.	2	Medium	Abhay

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	3 Days	25 Mar 2025	27 Mar 2025	20	27 Mar 2025
Sprint-2	20	3 Days	28 Mar 2025	30 Mar 2025	20	30 Mar 2025
Sprint-3	20	3 Days	01 Apr 2025	03 Apr 2025	20	03 Apr 2025
Sprint-4	20	3 Days	4 Apr 2025	06 Apr 2025	20	06 Apr 2025

6. RESULTS

6.1 Output Screenshots

 **Walleto**

[Add Transaction](#) [Add Category](#) [Categories](#) [Profile](#) [Dashboard](#)

[Logout](#)

Add New Category

Fill in the details below

Type

Select transaction type

Name

Enter category name

Add Category

Categories

Manage your transaction categories

[+ Add New](#)

<div></div> salary <div>Income</div>	<div></div> <div></div>
<div></div> waterpark <div>Expense</div>	<div></div> <div></div>
<div></div> household <div>Expense</div>	<div></div> <div></div>

Add New Transaction

Track your income and expenses with ease

 Transaction Type

Select transaction type

 Amount

\$ 0.00

 Category

Select a category

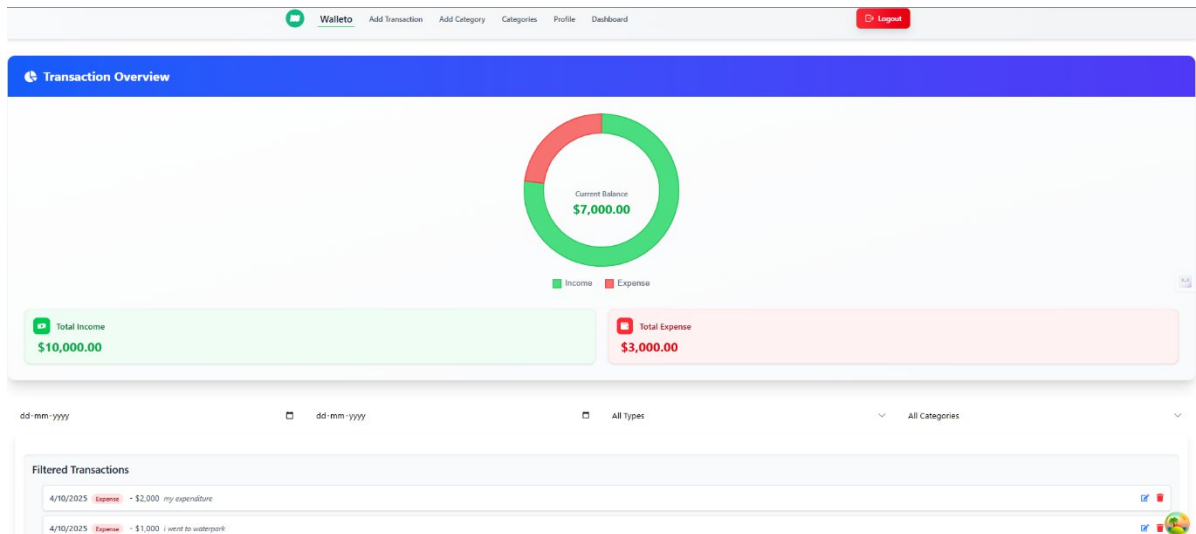
 Date

dd-mm-yyyy

 Description (Optional)

Add notes about this transaction

[+ Add Transaction](#)



7. ADVANTAGES & DISADVANTAGES

Advantages

1. User-Friendly Interface

The frontend is designed with modern UI/UX principles using Tailwind CSS, ensuring intuitive navigation and responsiveness across devices.

2. Efficient Expense Management

Users can easily categorize, filter, and visualize their income and expenses, helping them make informed financial decisions.

3. Real-Time Data Handling with React Query

State synchronization and data fetching are efficiently managed using React Query, providing real-time updates and cache handling.

4. Secure Authentication

JWT-based authentication ensures that user data is protected, with secure routes guarded via custom AuthRoute.

5. Scalable Architecture

The project uses modular folder structures, API abstraction, and Redux Toolkit for future scalability and maintainability.

6. Dynamic Visualizations

Doughnut charts and visual analytics help users quickly understand spending patterns, making financial insights visually appealing.

7. Reusable Component-Based Design

Reusable UI components like `AlertMessage`, `PrivateNavbar`, and `TransactionForm` enhance consistency and reduce code duplication.

Disadvantages

1. Initial Load Delay on Token Validation

Due to localStorage-based token initialization, users might see a brief "Token expired" message before Redux state syncs on app start.

2. No Offline Support

The app currently lacks offline capabilities or service workers, so it requires continuous internet access.

3. Single User Role

The system assumes all users are regular users (no admin/moderator role segregation), which may limit access control in larger systems.

4. No Push Notifications or Real-Time Sync

Users are not notified in real-time about changes; the experience is not reactive beyond user-triggered actions.

8. CONCLUSION

The **Walleto Expense Tracker** project demonstrates the successful integration of modern web development technologies to build a responsive, user-centric, and secure platform for managing personal or small business finances. By combining a robust backend built with Node.js and Express, and a dynamic frontend built with React, Redux Toolkit, React Query, and Tailwind CSS, the application provides users with intuitive tools to record transactions, categorize spending, and visualize financial insights in real time. Through this project, we addressed common pain points such as lack of financial visibility, poor budgeting discipline, and data fragmentation. The solution not only simplifies expense tracking but also empowers users to make smarter financial decisions through analytics and forecasting features. While the system has room for enhancement—such as integrating real-time sync, multi-role support, or mobile responsiveness—it lays a solid foundation for scalable, maintainable, and production-ready full-stack development. This project also strengthened practical knowledge in frontend-backend integration, authentication, state management, and clean UI design. Overall, **Walleto** serves as a practical and impactful financial tool for individuals and small businesses, and showcases the power of modern JavaScript stacks in solving real-world problems.

9. FUTURE SCOPE

The Walleto Expense Tracker project, while functionally complete, presents several promising opportunities for future enhancement and expansion. One of the key areas of growth lies in the integration of AI-powered financial insights, such as predictive budgeting based on historical data and personalized saving suggestions using machine learning models. Another potential enhancement is the development of a mobile application using frameworks like React Native or Flutter, allowing users to manage their expenses seamlessly on the go. Additionally, features like multi-user support, collaborative budgeting for families or teams, and real-time notifications for abnormal spending patterns can significantly improve user engagement. Integrating third-party APIs for automatic transaction syncing from banks or digital wallets will further streamline user experience. From a business standpoint, incorporating a subscription model for premium analytics or exportable reports (CSV, PDF) could provide monetization opportunities. With these future developments, Walleto can evolve from a personal expense tracker into a comprehensive, smart financial management platform.

10. APPENDIX

Source Code:

<https://github.com/JV456/SmartBridge-Fullstack-Development-MongoDB-Externship>

GitHub:

<https://github.com/JV456/SmartBridge-Fullstack-Development-MongoDB-Externship>

Project Demo Link:

<https://drive.google.com/file/d/1ZctKXaxWpr8tbgME6fkG-oOLOcnDXZwY/view?usp=sharing>