

Name: Jordan Brown

Class: SDET101

Section: 102

Teacher: Professor Iaboni

## **Expense TrackerPro - Final Report**

A Personal Finance Management System

### **Abstract**

Expense TrackerPro is a comprehensive full-stack personal finance application created to assist users in tracking their income and expenses, managing budgets, and visualizing their spending patterns. This system includes secure user authentication, transaction management, budget tracking, bill reminders, and advanced data visualization tools. The frontend of the application is built using React and TypeScript, while the backend is powered by Express.js and PostgreSQL. Expense TrackerPro offers an intuitive interface for financial management while ensuring strong data protection and security.

### **Introduction**

#### **Problem Statement**

A significant number of individuals struggle with financial visibility, which can lead to overspending, missed payments, and challenges in meeting savings goals. Many existing financial tools do not address these issues effectively. They often lack intuitive interfaces, fail to provide actionable insights, and do not integrate budgeting features with transaction tracking capabilities.

### **Objectives**

The goal of this project was to create a unified platform where users can monitor all financial activities. The system was designed to include visual tools to help analyze spending behavior, implement proactive budget management functions that alert users when they are nearing their limits, and ensure data security through proper authentication mechanisms.

### **Implementation Details**

Expense TrackerPro was developed using a modern technology stack. The frontend was built with React, TypeScript, Tailwind CSS, and Chart.js to ensure responsive and visually engaging components. On the backend, Express.js and Node.js handle the server-side logic, while PostgreSQL serves as the primary database, using Drizzle ORM for smooth data interaction. Authentication is managed using JWT tokens.

with password hashing for security. Data export capabilities are provided through PDFKit for generating PDFs and a CSV parser for exporting data in CSV format.

### Core Features

The application includes several key features to support personal financial management. User authentication is handled securely, allowing users to register and log in through a streamlined process. JWT token validation protects all API endpoints to ensure that user data remains secure.

Transaction management allows users to perform full CRUD (Create, Read, Update, Delete) operations on their income and expense entries. Transactions are organized by category to simplify tracking and reporting.

The dashboard offers analytics that display the user's monthly balance, including income and expenses, through interactive charts. It includes a six-month trend chart and a category breakdown represented in pie charts, providing users with clear insight into their financial behavior.

The budgeting system visually tracks user progress, such as showing that 30% of a housing budget has been used. Alerts notify users when they are approaching or exceeding their budget limits.

The bills manager helps users keep track of recurring payments. It sends timely reminders, such as notifying users when their rent is due within the next day. Users can select the frequency of bills (monthly or weekly) and mark them as paid when appropriate.

### Additional Features

The application includes data export capabilities, allowing users to generate downloadable CSV or PDF reports of their financial data. Its responsive design ensures a smooth experience on both desktop and mobile devices. For new users, helpful guidance appears when there are no transactions or data present, helping them get started.

### Challenges and Solutions

During development, several challenges were encountered. One issue involved ensuring real-time updates of budget data. This was resolved by implementing WebSocket connections to provide live updates. Another challenge was rendering complex charts with dynamic data. This was addressed by using Chart.js, which allowed for flexible and responsive chart

LoginRegister

Welcome back

Sign in to your account to continue

Username

johndoe

Password

\*\*\*\*\*

👁

Sign In

LoginRegister

Create an account

Sign up to start tracking your finances

Username

johndoe

Full Name (Optional)

John Doe

Email (Optional)

john@example.com

Password

\*\*\*\*\*

👁

Confirm Password

\*\*\*\*\*

👁

Create Account

Add New Transaction

Enter the details of your transaction below.

Transaction Type

Expense

Income

Amount

\$ 0

Category

Select a category

Date

mm/dd/yyyy

📅

Description

e.g., Grocery shopping

Notes (Optional)

Add any additional details

Add Transaction

displays. Lastly, handling dates across different time zones proved to be complex. The team resolved this by standardizing date handling using the Day.js library with UTC formatting.

## Screenshots and Functionality

The registration page (Fig. 1) features a clean and simple form, including an optional email field and password confirmation validation. The login interface (Fig. 2) uses a minimalist design to emphasize ease of use.

The transaction form (Fig. 3) allows users to toggle between income and expense types. It includes a category dropdown and a date picker to organize entries efficiently.

The main dashboard (Fig. 4) highlights key financial metrics, such as income and expenses. It includes interactive charts, a transaction list, and a section showing upcoming bills.

The bills management section (Fig. 5) enables users to select billing frequency and provides an "already paid" toggle for added flexibility.

## Conclusion and Future Enhancements

Throughout the development of Expense TrackerPro, several important lessons were learned. The importance of real-time data synchronization became clear, as it greatly enhances the user experience. Visual feedback proved highly valuable in helping users understand their financial situation. Moreover, securing financial data was recognized as a top priority, leading to the implementation of robust authentication measures.

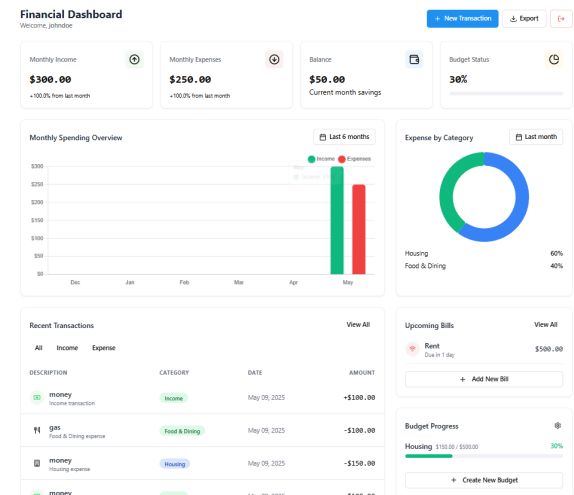
Looking ahead, several enhancements are planned. These include developing a mobile app version for both iOS and Android platforms, integrating with banks to import transactions automatically, implementing AI-driven suggestions to help users analyze and improve their spending habits, and enabling multi-user support to allow shared household budgeting.

## References

Chart.js Documentation - <https://www.chartjs.org/>

Drizzle ORM - <https://orm.drizzle.team/>

React Hook Form - <https://react-hook-form.com/>



The 'Add New Bill' form is designed to capture all necessary details for a new bill or recurring payment. It includes fields for the Bill Name (with a placeholder example), Amount, Category (a dropdown menu), Due Date (with a calendar icon), and Frequency (a dropdown menu). A 'How often this bill recurs.' label is placed below the frequency dropdown. At the bottom, there is an 'Already paid' checkbox with a sub-label 'Mark this bill as already paid for the current period.' The form concludes with 'Cancel' and 'Add Bill' buttons.

