

# University of Houston-Victoria

Department of Computer and Information Science

Group Project

## Expense Tracker Overview

Prepared for Partial Fulfillment of the Course

COSC 6342 Software Engineering Project Management

Spring 2024

By

Aradhana Sharma (2250955)

Kapil Suryawanshi (2039409)

Shayan Khan (2244554)

Suraj Odedra (2329291)

Advisor: Dr. Amjad Nusayr

Director of UG Computer Science & Computer Information Systems

Associate Professor of Computer Science

University of Houston Victoria

# Table of Contents

<b>I. Introduction.....</b>	<b>3</b>
<b>II. Objectives .....</b>	<b>3</b>
<b>III. Features.....</b>	<b>3</b>
<b>IV. Technology Stack .....</b>	<b>4</b>
<b>V. Source Code Summary .....</b>	<b>4</b>
<b>Program Units and Modules.....</b>	<b>4</b>
<b>1. Front End .....</b>	<b>4</b>
• React Components:.....	4
• Third-Party Modules: .....	5
<b>2. Back End .....</b>	<b>5</b>
• Routers:.....	5
• Controllers: .....	5
• Repositories: .....	5
• Middleware:.....	6
• Third-Party Modules.....	6
<b>3. Database .....</b>	<b>6</b>
<b>VI. Future Enhancements: .....</b>	<b>Error! Bookmark not defined.</b>
<b>VII. Conclusion .....</b>	<b>6</b>

# I. Introduction

Expense Tracker is a robust financial management system designed to empower users to effectively track, manage, and analyze their expenses and incomes. It provides a comprehensive set of features to streamline financial tasks, enhance budgeting practices, and promote better financial decision-making. This document offers an in-depth analysis of the source code structure and key components of the application.

## II. Objectives

- Offer users a centralized platform to monitor their financial activities.
- Facilitate the categorization and tracking of expenses and incomes.
- Provide insightful visualizations and reports for better financial understanding.
- Enable seamless collaboration and expense sharing among groups.
- Ensure data security, privacy, and user authentication.

## III. Features

User Management:

- Secure user registration and login functionality.
- Authentication and authorization mechanisms to protect user data.

Dashboard Interaction:

- Interactive dashboard with visual representations of expenses, incomes, and budgetary allocations.
- Overview of group transactions and individual financial summaries.

Category Management:

- Creation, editing, and deletion of customizable expense categories.
- Organization of expenses into structured categories for better tracking.

Budget Management:

- Setting and monitoring of personalized budgets for expense categories.
- Notifications and alerts for budget thresholds and overspending.

Transaction Management:

- Creation, editing, and deletion of individual transactions with detailed information.
- Transaction history and filtering options for better visibility.

Group Management:

- Establishment and management of groups for collaborative expense tracking.
- Group-specific transaction tracking and settlement functionalities.

Group Transaction Management:

- Creation of group transactions with multiple participants and expense splitting.
- Real-time updates on group transaction settlements and contributions.

## IV. Technology Stack

- Frontend: HTML5, CSS3, JavaScript, React.js
- Backend: Node.js, Express.js, MySQL.
- Authentication: JWT (JSON Web Tokens) for secure user authentication
- Visualization: recharts for interactive data visualization

## V. Source Code Summary

### Program Units and Modules

#### 1. Front End

- React Components:
  - BudgetForm: Manages budget-related operations, allowing users to create, update, and display budgets. It facilitates budget planning and tracking within the application.
  - CategoryForm: Handles category management, enabling users to create, edit, and delete expense categories. This component enhances expense organization and classification.
  - GroupForm: Manages group creation, editing, and deletion, facilitating collaborative expense tracking for shared expenses among group members.
  - GroupTransactionForm: Handles group transaction creation and management, allowing users to record and manage expenses associated with group activities or shared expenditures.
  - Reporting: Provides reporting and visualization functionality, including dashboards and charts. It offers insights into spending patterns, trends, and budget performance through graphical representations.
  - TransactionForm: Manages individual user transactions, including creation, editing, and deletion. This component serves as the primary interface for recording and managing personal expenses.
  - UserForm: Handles user account management, authentication, and authorization. It facilitates user registration, login, profile management, and access control within the application.
  - Table Components:
    - ExpenseTable: A custom component that displays expense data in a tabular format, providing functionalities for rendering, formatting, and managing the data.

- ExpenseTableModel: A custom table model that manages the data displayed in the ExpenseTable component, handling retrieval and manipulation of expense data for the table.
- Third-Party Modules:
  - bootstrap: A popular CSS framework used for UI styling, ensuring consistency and responsiveness across different devices and screen sizes.
  - react-dom: Enables the rendering of React components on the Document Object Model (DOM), facilitating dynamic updates and interactive user interfaces.
  - recharts: A charting library for React that allows the creation of interactive and customizable charts and visualizations. It enhances the reporting capabilities of the application with rich data visualization features.

## 2. Back End

- Routers:
  - user.js: Manages user-related routes and endpoints, handling operations such as user registration, authentication, and profile management.
  - transaction.js: Handles transaction-related routes and endpoints, including CRUD operations for individual user transactions.
  - groupTransaction.js: Manages group transaction-related routes and endpoints, facilitating the recording and management of expenses within group activities.
  - category.js: Handles category-related routes and endpoints for managing expense categories within the application.
  - budget.js: Manages budget-related routes and endpoints, allowing users to create, update, and retrieve budget information.
- Controllers:
  - User: Implements user-related request handling, orchestrating interactions with the service layer for user management functionalities.
  - Transaction: Orchestrates transaction-related operations, delegating tasks to the service layer for CRUD operations on individual user transactions.
  - GroupTransaction: Manages group transaction-related operations, coordinating actions with the service layer to handle group expense recording and management.
  - Category: Handles category-related operations, including CRUD operations for managing expense categories.
  - Budget: Orchestrates budget-related operations, interacting with the service layer to manage budget creation, updates, and retrieval.
- Repositories:
  - User: Implements data access and manipulation methods for user entities, including CRUD operations and user-specific queries.
  - Transaction: Facilitates database interactions for transaction entities, providing methods for storing, retrieving, and updating individual user transactions.
  - Group: Handles database interactions for group entities, offering methods for managing group data such as creation, retrieval, updating, and deletion.

- GroupTransaction: Manages database operations for group transaction entities, including CRUD operations and queries related to group expenses.
- Category: Handles database interactions for expense category entities, offering methods for managing category data.
- Budget: Manages database operations for budget entities, including CRUD operations and budget-specific queries.
- Middleware:
  - authenticator.js: Implements middleware functions for authentication and authorization, ensuring secure access to protected routes and resources within the application.
- Third-Party Modules:
  - mysql2: A MySQL database driver for Node.js, providing connectivity and interaction capabilities with the MySQL database.
  - jsonwebtoken: Facilitates JSON Web Token (JWT)-based authentication mechanisms, enabling secure user authentication and authorization.
  - bcryptjs: A library for password hashing and encryption, enhancing the security of user credentials stored in the database.
  - express: A web application framework for Node.js, simplifying the creation of RESTful APIs and handling HTTP requests and responses.
  - joi: Provides input validation capabilities, ensuring the integrity and validity of user input data.
  - dotenv: Facilitates environment configuration management, allowing the application to load environment variables from a .env file.

### 3. Database

The Expense Tracker application utilizes a MySQL relational database to store and manage data related to user accounts, transactions, categories, budgets, and group transactions. The database schema is designed to establish relationships between entities and ensure data integrity through normalization and appropriate constraints.

## VI. Conclusion

Expense Tracker is a powerful tool designed to simplify financial management and promote financial well-being. By providing users with intuitive features and actionable insights, it aims to empower individuals and groups to make informed financial decisions and achieve their financial goals. The Expense Tracker source code embodies a structured and modular architecture, leveraging ReactJS for the front end, JavaScript for the back end, and MySQL for the database. Through a comprehensive set of program units, modules, and components, the application delivers an intuitive and feature-rich experience for expense tracking, budget management, and group

collaboration. With its extensible design and robust functionality, the Expense Tracker application empowers users to gain insights into their financial activities and make informed decisions for better financial health.