**FULL STACK DEVELOPMENT PROJECT DOCUMENTATION**

* Introduction

App name : WalletWatch – The expense tracker app

Team Members:

* M.Madhuri Nagalakshmi
* Rohan Singh
* Shrinidhi N
* Project Overview

Purpose : The purpose of the WalletWatch App is to help users effectively manage their personal finances by providing tools to track income and expenses, categorize transactions, and generate insightful reports. Some key purposes include the following

1. **Financial Awareness:**
   * Gain a clear understanding of where money is being spent.
   * Identify spending patterns and habits.
2. **Budget Management:**
   * Set and monitor budgets for different categories.
   * Avoid overspending and manage finances efficiently.
3. **Expense Control:**
   * Track daily, weekly, and monthly expenses.
   * Cut down unnecessary expenditures by identifying areas of overspending.
4. **Income Tracking:**
   * Record and categorize various sources of income.
   * Monitor cash flow to ensure financial stability.
5. **Financial Planning:**
   * Plan for future expenses by analyzing past spending trends.
   * Save for specific goals such as vacations, emergencies, or large purchases.
6. **Convenience:**
   * Simplify the process of managing finances through a user-friendly interface.
   * Access financial data anytime, anywhere via mobile or web platforms.
7. **Report Generation:**
   * Generate detailed financial reports for better analysis.
   * Use visualizations such as charts and graphs to understand financial data.
8. **Data Security:**
   * Ensure that all financial data is securely stored and protected.
   * Provide backup and recovery options to prevent data loss.

By fulfilling these purposes, WalletWatch App empowers users to take control of their financial health, make informed financial decisions, and achieve their financial goals more effectively.

Key Features: Some key features of the WalletWatch app include

* **User Authentication:** Secure login and registration.
* **Dashboard:** Overview of financial status.
* **Expense Management:** Add, edit, and delete expenses.
* **Income Management:** Add, edit, and delete income entries.
* **Categories:** Categorize transactions for better organization.
* **Reports:** Generate monthly financial reports.
* **Notifications:** Set reminders for bill payments.
* Architecture

Frontend:  
The frontend is built using React, which provides a responsive and interactive user interface. It includes components like

* login
* registration
* dashboard
* transaction management
* report generation.

Backend:

The backend is developed with Node.js and Express.js, providing RESTful API services for user authentication, transaction management, and data retrieval. JWT (JSON Web Tokens) is used for secure authentication.

Database:  
The database is MongoDB, which stores user data, transactions, categories, and reports. Mongoose is used for schema definition and interaction with MongoDB.

* Setup instructions

Prequisites:

* Node.js 14.0+
* MongoDB 4.0+
* npm 6.0+

Installation

1. Repository Cloning .
2. Frontend setup
3. Backened setup

* Folder Structure:
* Running the Application

Frontend:

Code:

Cd frontend

Npm start

Backend:

Code:

Cd backend

Npm start

* API documentation

Authentication:

* POST /api/auth/register: Register a new user.
* POST /api/auth/login: User login.

Transactions:

* GET /api/transactions: Get all transactions.
* POST /api/transactions: Add a new transaction.
* PUT /api/transactions/:id: Update a transaction.
* DELETE /api/transactions/:id: Delete a transaction.

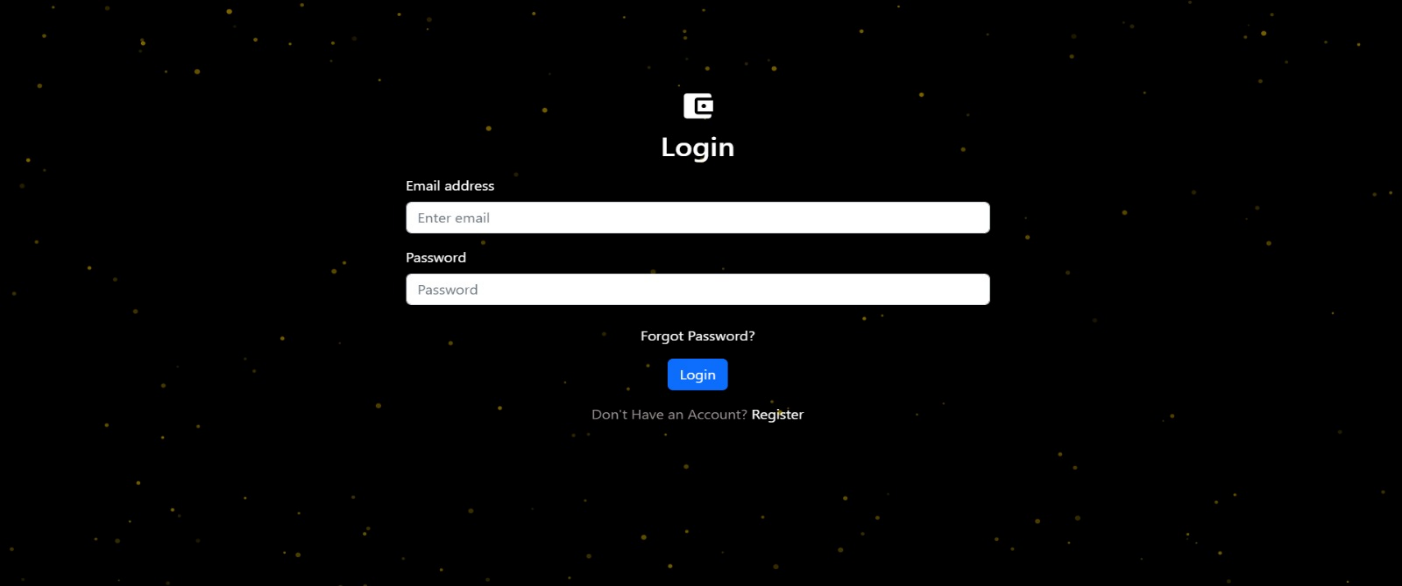
Categories:

* GET /api/categories: Get all categories.
* POST /api/categories: Add a new category.
* Authentication

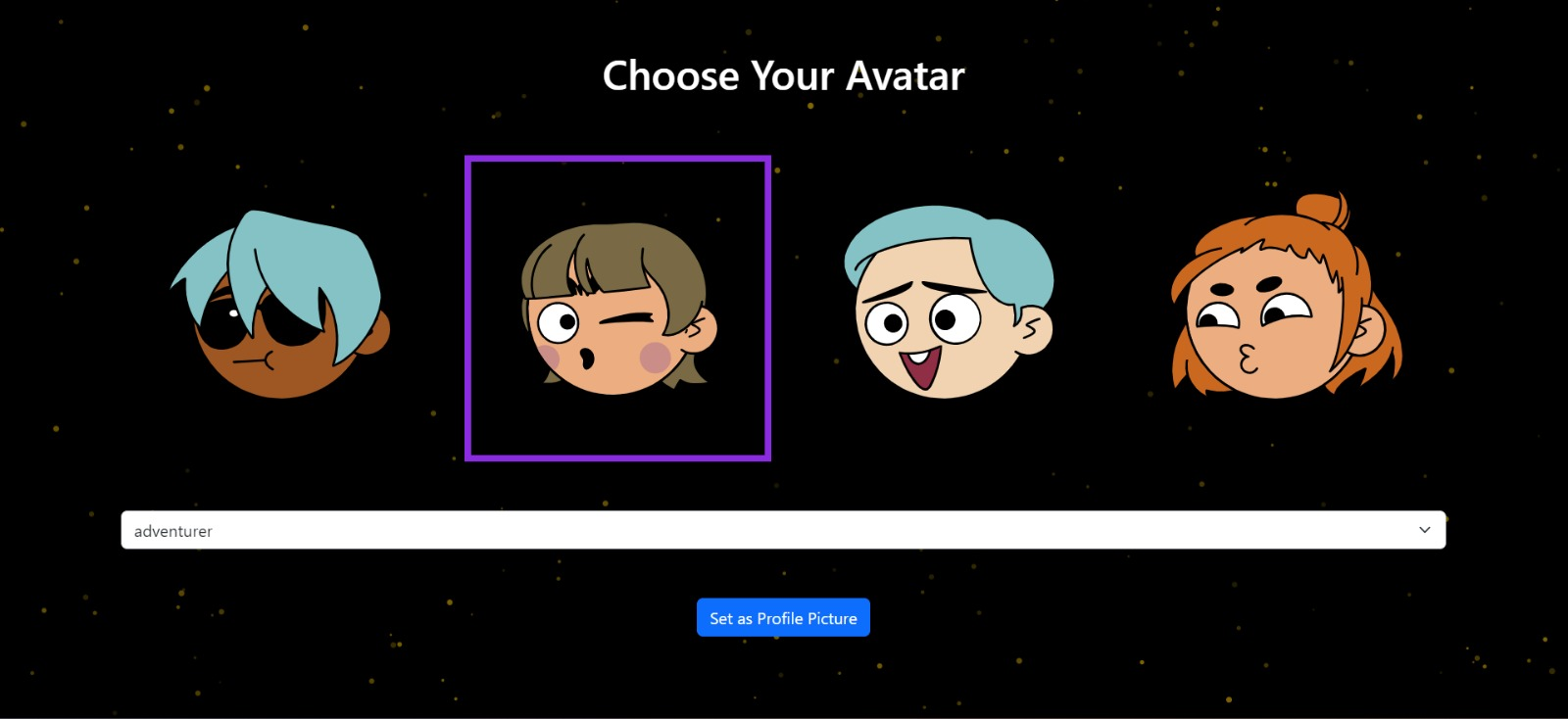
Authentication is handled using JSON Web Tokens. Upon successful login, a token is issued to the user, which must be included in the Authorization header of subsequent requests to access protected routes. Token validation and user sessions are managed server-side.

* User Interface

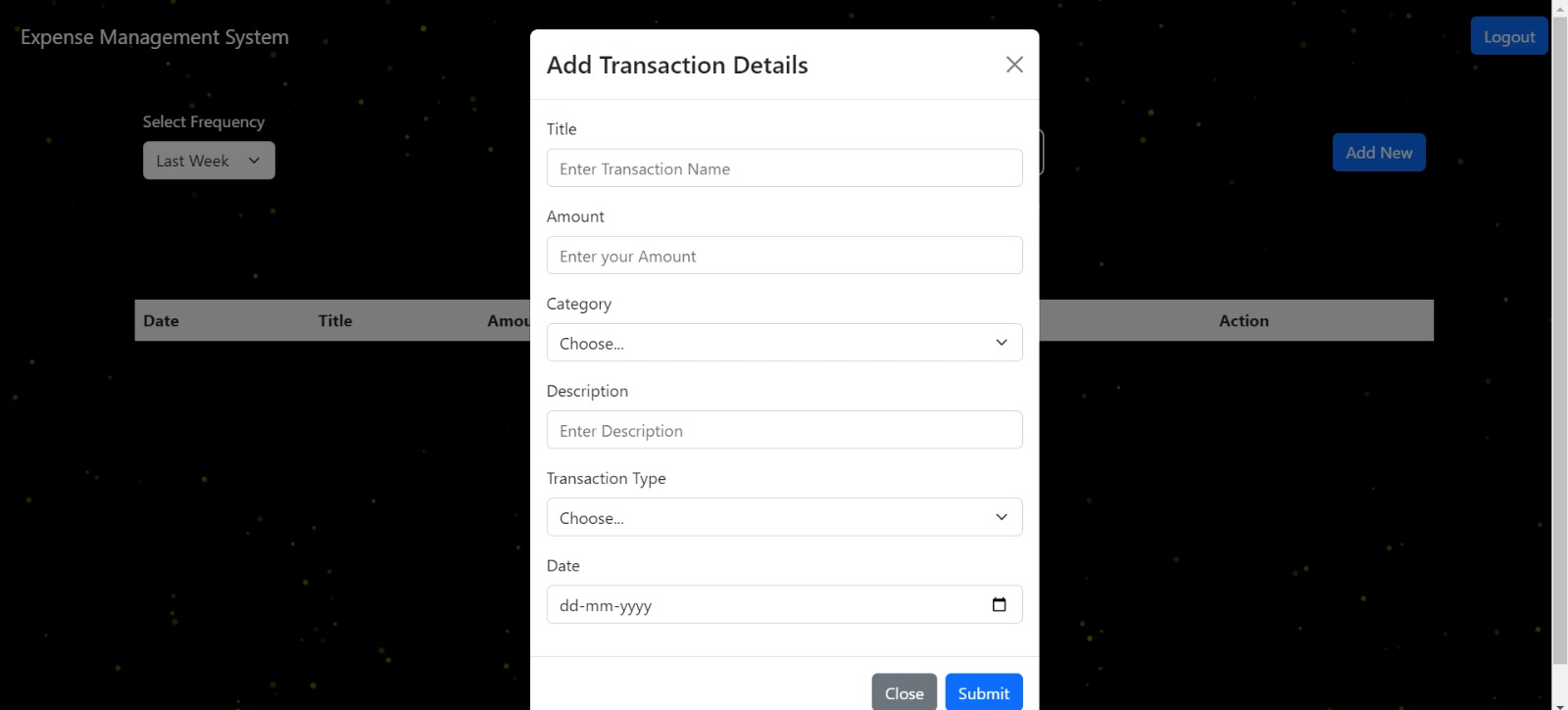
LOGIN PAGE



PROFILE PICTURE SELECTION

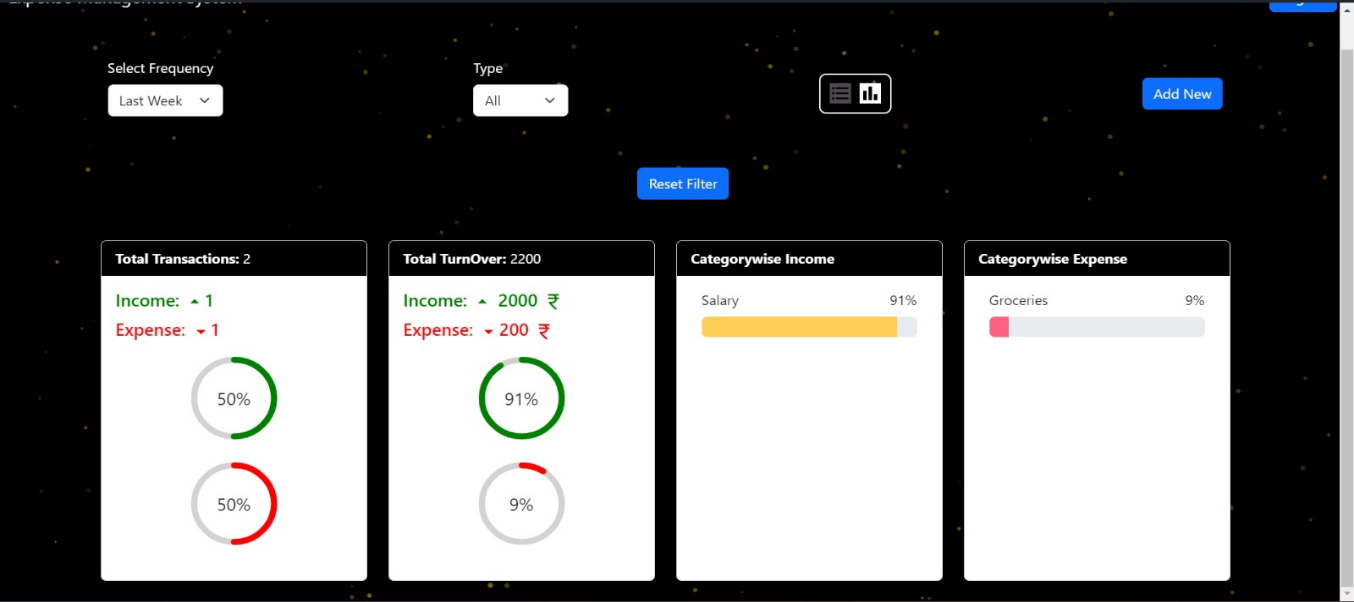


ADDING TRANSACTIONS

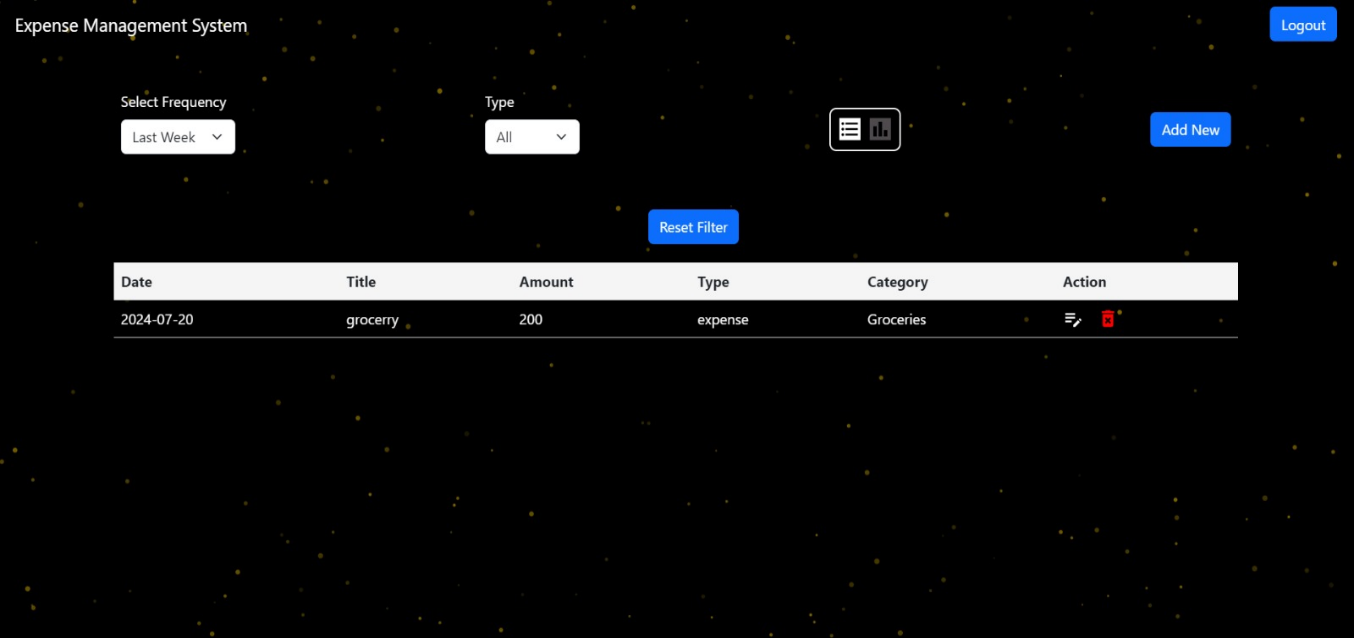


Testing

CHART VIEW

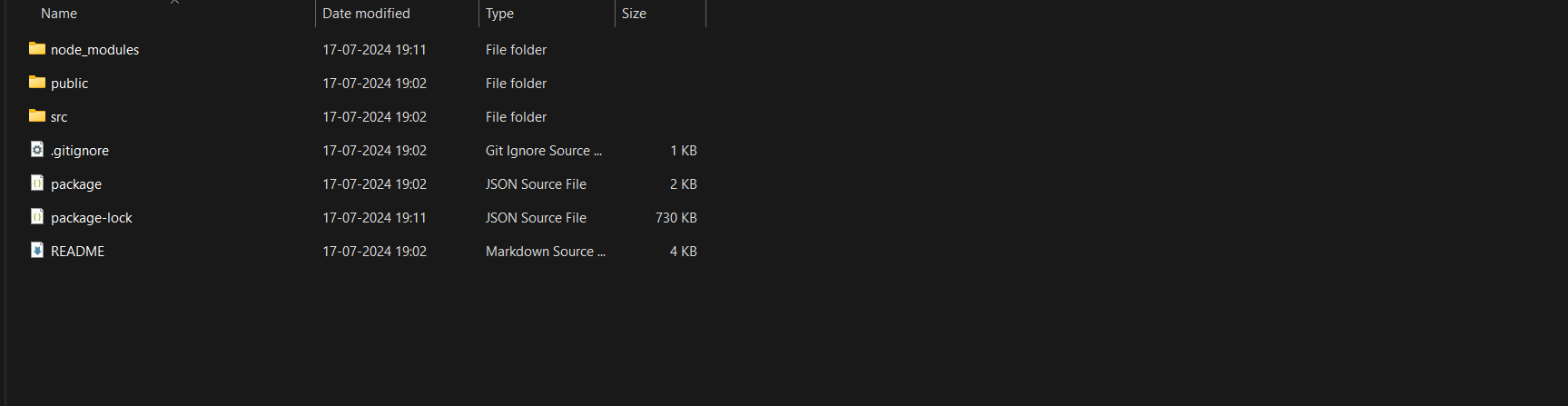


LIST VIEW



* ELEMENTS

FRONTEND



BACKEND



* Known Issues
* Occasionally, the login page may not redirect correctly after successful authentication.
* Loading can take more time than usual sometimes because of large datasets.
* Future Enhancements
* Implementing a mobile application using React Native.
* Adding a feature for budget planning and tracking.
* Integrating with third-party financial services for automated transaction imports.
* Track expenses and income in different currencies for users with international financial activities.