**Full Stack Development with MERN**

**Project Documentation**

**1. Introduction**

* **Project Title:** Expense Tracker Web Application
* **Team Leader :** Harsh Anand Dhumal
* **Team member :** Hariom Yadav
* **Team member :** Yash Detani
* **Team member :** Mohammad Umar Ali Khan

**2. Project Overview**

**Purpose**

To help users effectively manage their personal finances by tracking income and expenses, visualizing trends, and setting financial goals.

**Features**

* User authentication using JWT
* Add, update, delete income and expense entries
* Pie, bar, and line charts for visualization
* Date-wise filtering and category breakdowns
* Profile image upload and emoji-based categorization
* Responsive design for all devices

**3. Architecture**

**Frontend**

* **Library:** React.js with TailwindCSS
* **Features:** Routing, context-based state management, responsive components

**Backend**

* **Platform:** Node.js with Express.js
* **Structure:** REST API architecture with secure route handling and middleware

**Database**

* **Type:** NoSQL
* **Database:** MongoDB Atlas
* **Details:** Mongoose schema for Users, Transactions (income/expenses)

**4. Setup Instructions**

**Prerequisites**

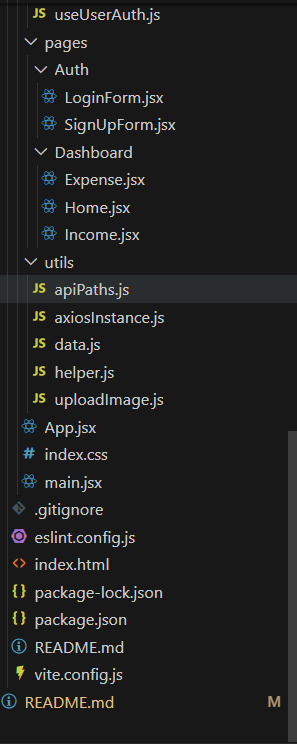
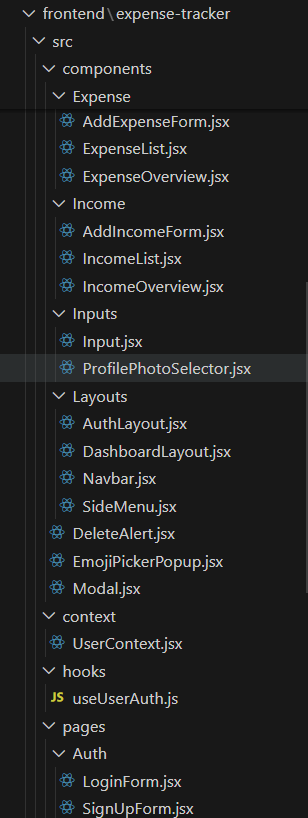
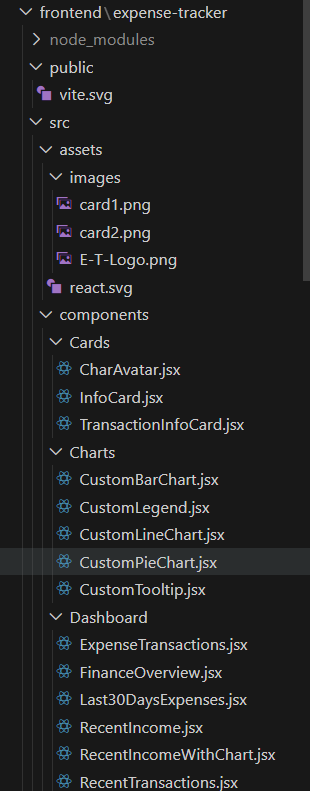
* Node.js
* MongoDB Atlas Account
* npm

**Installation**

1. Clone the repository
2. Navigate to the root and run npm install for both client and server folders
3. Add .env file in server with:
   * MONGO\_URI=your\_mongo\_db\_uri
   * JWT\_SECRET=your\_jwt\_secret
4. Run npm start in both folders

**5. Folder Structure**

**frontend**



**Backend**

A screenshot of a computer program

AI-generated content may be incorrect.

**6. Running the Application**

**Frontend:**

cd frontend/expense-tracker

npm run dev

**Backend:**

cd backend

npm run dev

**7. API Documentation**

**Auth Routes**

* POST /api/v1/auth/register
* POST /api/v1/auth/login

**Transaction Routes (Protected)**

* POST /api/v1/transactions/income
* POST /api/v1/transactions/expense
* GET /api/v1/transactions – all entries
* DELETE /api/v1/transactions/:id

All requests require a Bearer token after login.

**8. Authentication**

**Method: JWT (JSON Web Tokens)**

* On login/register, token is generated and stored in localStorage
* Protected routes verify the token via middleware
* User context is updated based on the token’s payload

**9. User Interface**

* Clean and modern interface with TailwindCSS
* Mobile responsive layout
* Interactive pie, bar, and line charts for income/expense breakdown
* upload/remove functionality

**10. Testing**

**Strategy**

* Manual testing across pages
* Input validations for email, amount, and date
* Token expiry handling

**11. Screenshots or Demo**

* [https://www.youtube.com/watch?v=BgJ273dJtBM&feature=youtu.be]

**12. Known Issues**

* No dark mode yet
* No recurring transaction logic
* User cannot edit existing entries (only delete)

**13. Future Enhancements**

* Add multi-user budgets and goal progress tracking
* Implement recurring transactions and reminders
* Export data as CSV or PDF
* Integrate UPI/banking APIs for live tracking
* Add dark/light theme toggle

**End of Documentation**