Project Documentation: MERN Stack Application

MERN Stack Application Documentation

Project Overview

This project is a MERN stack application implementing a product transaction system. It includes backend APIs to manage and retrieve transaction data, a frontend interface to interact with the data, and uses MongoDB for data storage.

The application supports:

- Database initialization from a third-party API.
- APIs for search, pagination, and filtering transactions by month.
- Additional APIs for statistics, bar charts, and pie charts.
- Frontend implementation to display data and visualizations.

Backend Packages

1. Express: Used to create a RESTful API server.
2. Mongoose: A MongoDB ODM to interact with the database using models.
3. Axios: Used to fetch data from the third-party API for database initialization.
4. Dotenv: For environment variable management (e.g., MongoDB connection string).
5. Cors: Middleware to allow cross-origin requests.
6. Nodemon (dev dependency): Automatically restarts the server during development.

Project Documentation: MERN Stack Application

Frontend Packages

1. React: Library for building the user interface.

2. Axios: For making HTTP requests to the backend APIs.

3. Chart.js: For creating bar and pie charts.

4. React-Chartjs-2: React wrapper for Chart.js.

5. React-Bootstrap & Bootstrap: For styling the frontend interface.

Backend Implementation

- Entry Point: `server.js`

- MongoDB Model: `Transaction.js`

- API Routes: `routes/api.js`

1. **Database Initialization API**:

   Fetches data from the third-party API and seeds the MongoDB database.

2. **Transaction Listing API**:

   - Supports search, pagination, and filtering by month.

   - Search matches on title, description, and price.

   - Pagination uses `page` and `perPage` query parameters.

3. **Statistics API**:

   - Returns total sales, sold items, and unsold items for a given month.

4. **Bar Chart API**:

   - Returns item counts by price range for a given month.

5. **Pie Chart API**:

   - Returns item counts grouped by categories for a given month.

Frontend Implementation

- Main Component: `App.js`

- Subcomponents:

  1. `TransactionsTable.js`: Displays paginated transaction data with search functionality.

  2. `Statistics.js`: Shows statistics for the selected month.

  3. `BarChart.js`: Displays a bar chart of item counts by price range.

  4. `PieChart.js`: Displays a pie chart of item counts by category.

Key Features:

- Search and Pagination: Seamlessly integrated into the `TransactionsTable` component.

- Visualizations: Bar and pie charts dynamically update based on the selected month.

- Dropdown: Month selection dropdown updates all components.