**Cryptoverse Dashboard Documentation**

**1. Introduction**

**Project Title: Cryptoverse Dashboard**

**Team Members:**

* **Amith babu** – Team Leader
* **Akash** – Team member
* **Deepak** - Team member
* **Jeeva prakash**- Team member
* **Kamesh** - Team member

**2. Project Overview**

**Purpose**

The **Cryptoverse Dashboard** is a frontend application that provides real-time data visualization and tracking of cryptocurrency trends. It aims to deliver an intuitive interface for users to analyze crypto markets effectively.

**Features**

* Real-time cryptocurrency prices and trends.
* Interactive charts and data visualization.
* Custom watchlists and favorites.
* Authentication and user profiles.
* Dark and light theme support.

**3. Architecture**

**Component Structure**

* **App Component**: Root component managing global state and layout.
* **Navbar Component**: Handles navigation between sections.
* **Dashboard Component**: Displays real-time crypto market insights.
* **Chart Component**: Visualizes price trends using libraries like Recharts or Chart.js.
* **UserProfile Component**: Manages user authentication and settings.

**State Management**

* Uses **Redux Toolkit** for managing global state.
* Local state handled via React's **useState** and **useEffect** hooks.

**Routing**

* Utilizes **React Router** for page navigation:
  + / - Homepage
  + /dashboard - Crypto analytics
  + /profile - User settings
  + /favorites - Saved cryptocurrencies

**4. Setup Instructions**

**Prerequisites**

* **Node.js** (v16+)
* **npm** or **yarn**

**Installation**

1. Clone the repository:

git clone https://github.com/your-repo/cryptoverse-dashboard.git

1. Navigate to the project directory:

cd cryptoverse-dashboard

1. Install dependencies:

npm install

1. Configure environment variables in .env file:

REACT\_APP\_API\_KEY=your\_api\_key\_here

**5. Folder Structure**

cryptoverse-dashboard/

│── public/

│── src/

│ ├── components/

│ ├── pages/

│ ├── assets/

│ ├── store/

│ ├── utils/

│ ├── App.js

│ ├── index.js

│── .env

│── package.json

**6. Running the Application**

To start the frontend server:

npm start

**7. Component Documentation**

**Key Components**

* **CryptoList**: Displays a list of cryptocurrencies with real-time price updates.
* **PriceChart**: Interactive chart for tracking price trends.
* **SearchBar**: Allows users to search for cryptocurrencies.

**Reusable Components**

* **Button**: Customizable button component.
* **Card**: Styled container for displaying data.

**8. State Management**

**Global State**

* **Redux Toolkit** stores user preferences and API responses.
* Middleware such as **Redux Thunk** handles API requests.

**Local State**

* Components use **useState** for UI-based interactions (e.g., modals, dropdowns).

**9. User Interface**

* Interactive charts using **Chart.js**.
* Responsive design supporting mobile and desktop views.

**10. Styling**

**CSS Frameworks/Libraries**

* **Tailwind CSS** for styling components.
* **Styled-Components** for dynamic theming.

**Theming**

* Supports both **light** and **dark** modes using context-based theming.

**11. Testing**

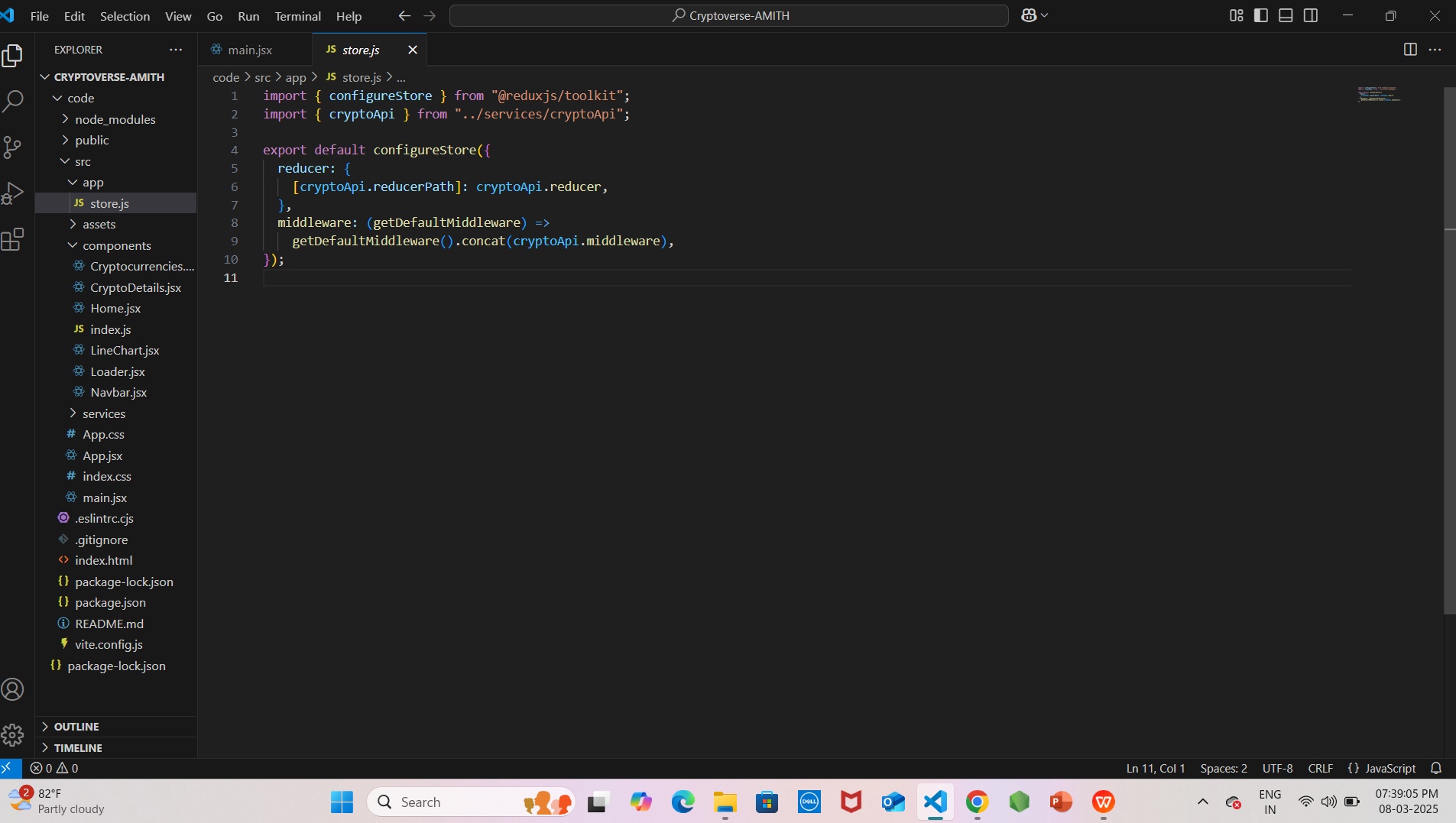
**Testing Strategy**

* **Unit Testing**: Jest & React Testing Library for component tests.
* **Integration Testing**: Testing API data rendering.
* **End-to-End Testing**: Cypress for user interaction testing.

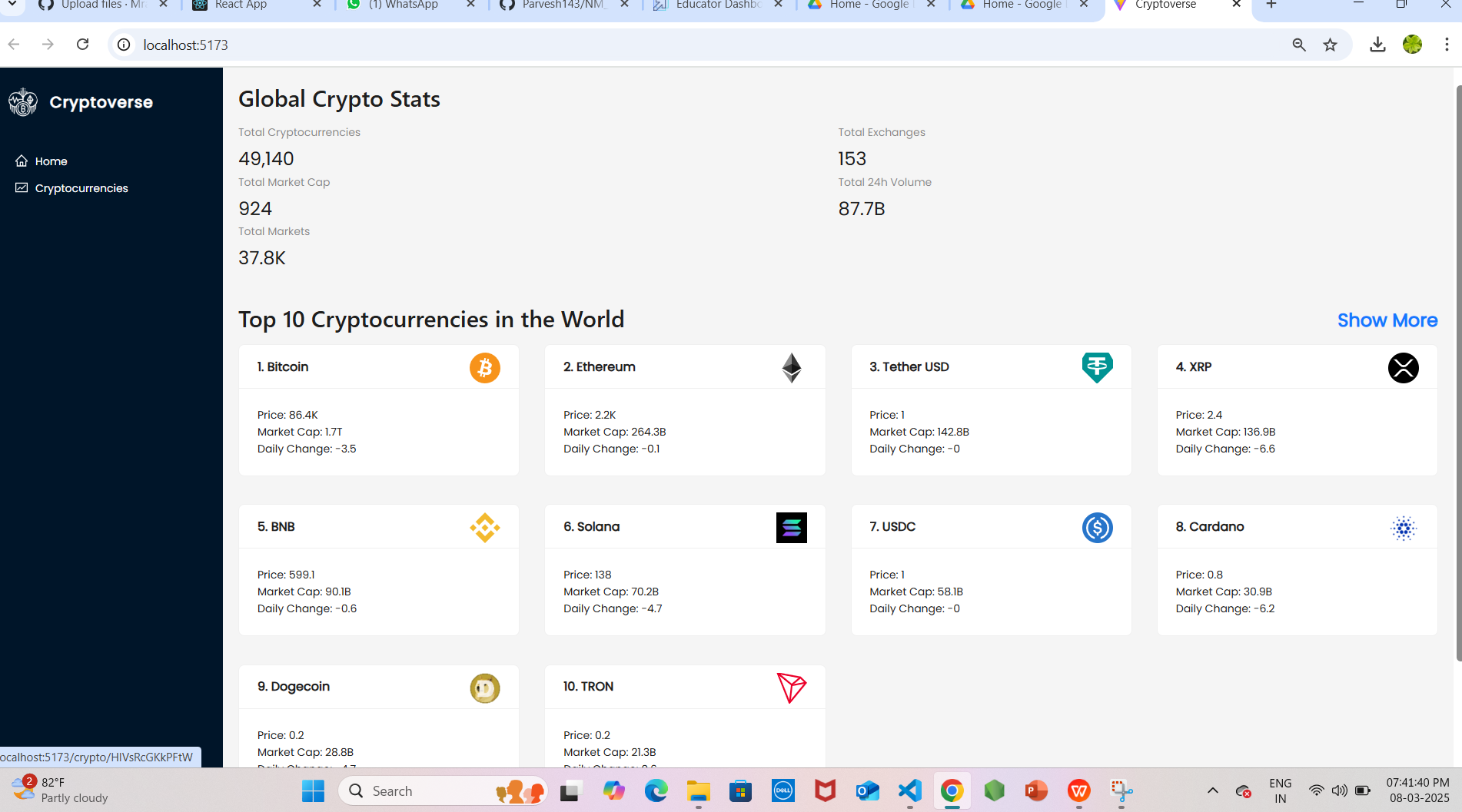
**Code Coverage**

* Jest configured for test coverage reports.

1. **Screenshots or Demo**



**Example of code overview**

****

**13. Known Issues**

* API rate limits might affect real-time updates.
* UI responsiveness improvements in progress.

**14. Future Enhancements**

* Integration with more crypto exchanges.
* AI-driven price predictions.
* Improved accessibility features.