**Cryptoverse: Frontend Documentation**

**1. Introduction**

**Project Title:** Cryptoverse

**Team Members:**

* Project Lead - NAVEEN B(naveenbalaraman21122003@gmail.com)
* Member - SATHISH KUMAR K(sathish7122004@gmail.com)
* Member - SARAVANA KUMAR J(sl9380447@gmail.com)
* Member - HARIKRISHNAN R(harikrishnan46237@gmail.com)

**2. Project Overview**

**Purpose:**

Cryptoverse is a cryptocurrency dashboard that provides real-time data, market trends, and insights for digital assets. The platform aggregates data from multiple APIs to deliver an interactive experience for users, including price tracking, historical data visualization, and portfolio management.

**Features:**

* **Real-Time Cryptocurrency Prices** – Fetches live price updates.
* **Market Trends & Analysis** – Displays historical price trends and key insights.
* **Portfolio Management** – Users can track their crypto holdings.
* **News Aggregation** – Curated crypto news from multiple sources.
* **Currency Converter** – Convert between different cryptocurrencies and fiat.
* **Dark Mode & Custom Themes** – Customizable UI for better readability.

**3. Architecture**

**Component Structure:**

* **App.js** – Root component managing routes and global state.
* **Layout.js** – Handles header, sidebar, and page structure.
* **Dashboard.js** – Displays market data and portfolio insights.
* **CryptoList.js** – Lists cryptocurrencies with their prices and percentage changes.
* **CryptoDetails.js** – Provides a detailed view of a selected cryptocurrency.
* **NewsFeed.js** – Displays the latest cryptocurrency-related news.
* **Portfolio.js** – Allows users to track their crypto assets.
* **SearchBar.js** – Enables real-time search for cryptocurrencies.

**State Management:**

* **Redux Toolkit** for global state management:
* cryptoSlice.js – Manages cryptocurrency data.
* userPreferencesSlice.js – Stores user settings
* portfolioSlice.js – Tracks portfolio holdings.
* **useState & useContext** for UI-related states (e.g., dark mode toggle).

**Routing (React Router):**

* / – Home/Dashboard
* /cryptos – Cryptocurrency List
* /crypto/:id – Cryptocurrency Details
* /portfolio – User Portfolio
* /news – Crypto News Feed
* \* – 404 Not Found

**4. Setup Instructions**

**Prerequisites:**

* Node.js (v16 or later)
* npm or yarn
* Git
* Code Editor (VS Code recommended)

**Installation Guide:**

# Clone the repository

git clone https://github.com/your-username/cryptoverse.git

cd cryptoverse

# Install dependencies

npm install # or yarn install

# Set up environment variables

cp .env.example .env

Configure .env file:

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

REACT\_APP\_API\_BASE\_URL=https://api.coingecko.com/api/v3

Run the development server:

npm start # or yarn start

**5. Folder Structure**

Cryptoverse/

├── public/ # Static assets

├── src/

│ ├── components/ # Reusable UI components

│ ├── pages/ # Page components

│ ├── redux/ # Redux slices and store

│ ├── utils/ # Helper functions and API calls

│ ├── styles/ # Global styles and themes

│ ├── App.js # Main app component

│ ├── index.js # App entry point

├── .env # Environment variables

├── package.json # Project dependencies

└── README.md # Project documentation

**6. Running the Application**

npm start # or yarn start

**7. Component Documentation**

**Key Components:**

* **CryptoCard.js** – Displays an individual cryptocurrency.
* **CryptoList.js** – Fetches and renders a list of cryptocurrencies.
* **SearchBar.js** – Handles search functionality.
* **Portfolio.js** – Manages and displays user’s crypto holdings.
* **NewsFeed.js** – Shows latest crypto news.

**Reusable Components:**

* **Button.js** – Customizable button component.
* **Loader.js** – Displays a loading spinner.
* **ErrorBoundary.js** – Catches and displays errors.

**8. State Management**

**Global State (Redux Toolkit):**

* Centralized store for cryptocurrency data, user preferences, and portfolio tracking.
* Utilizes createSlice() for modular state management.

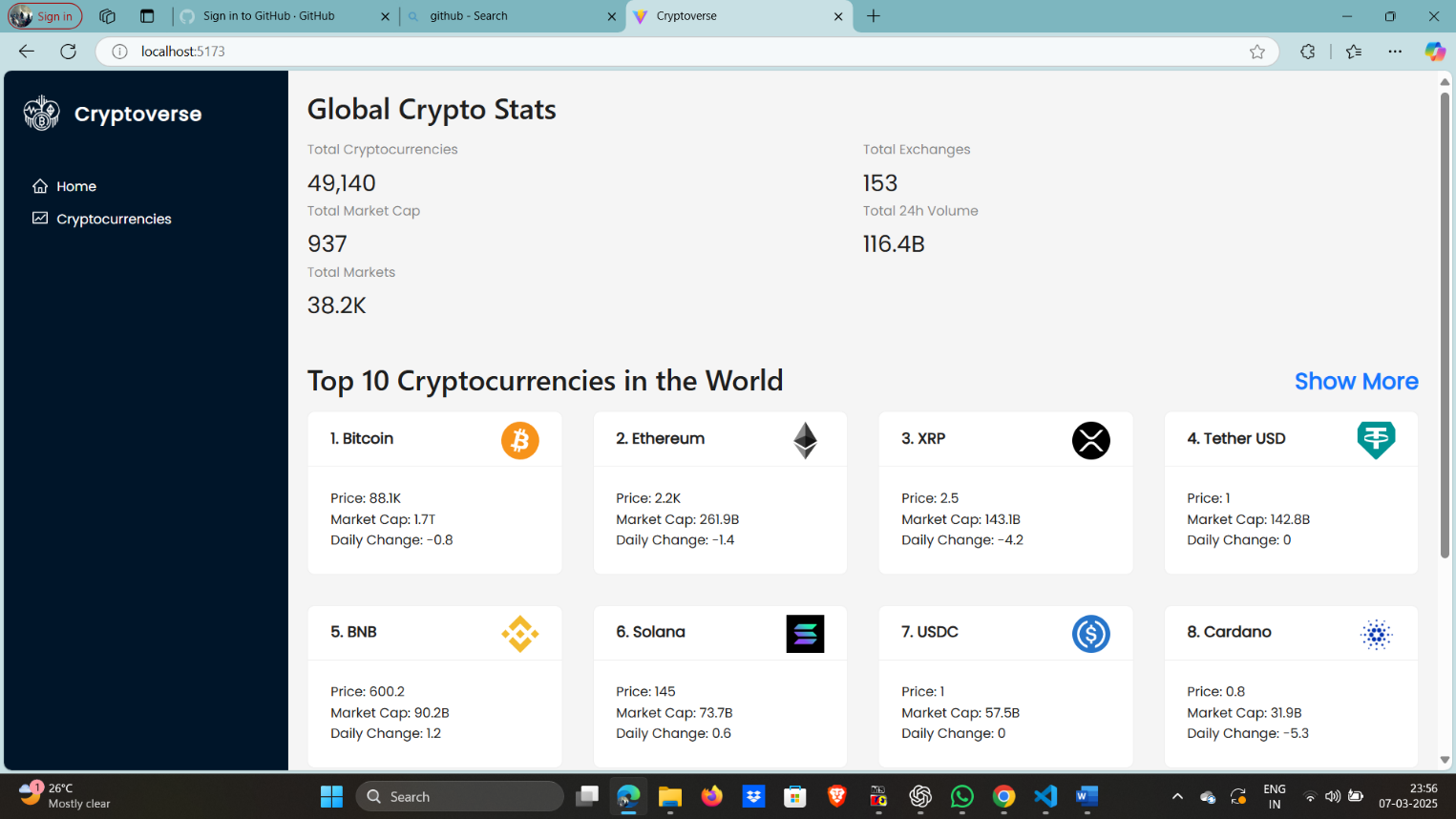
**Local State:**

* **useState** for UI interactions like toggles and modals.

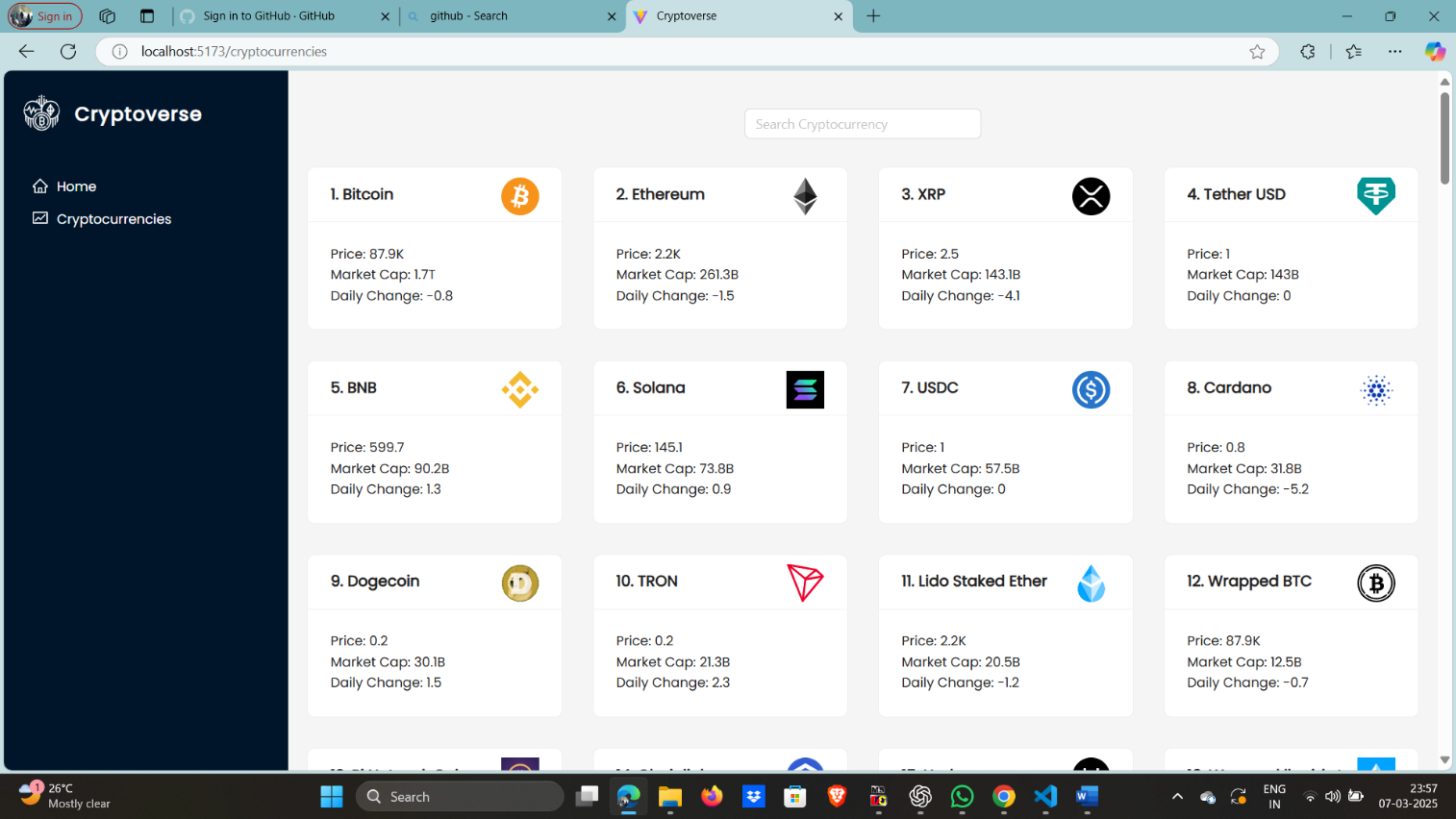
**9. User Interface**

* Screenshots or GIFs showcasing

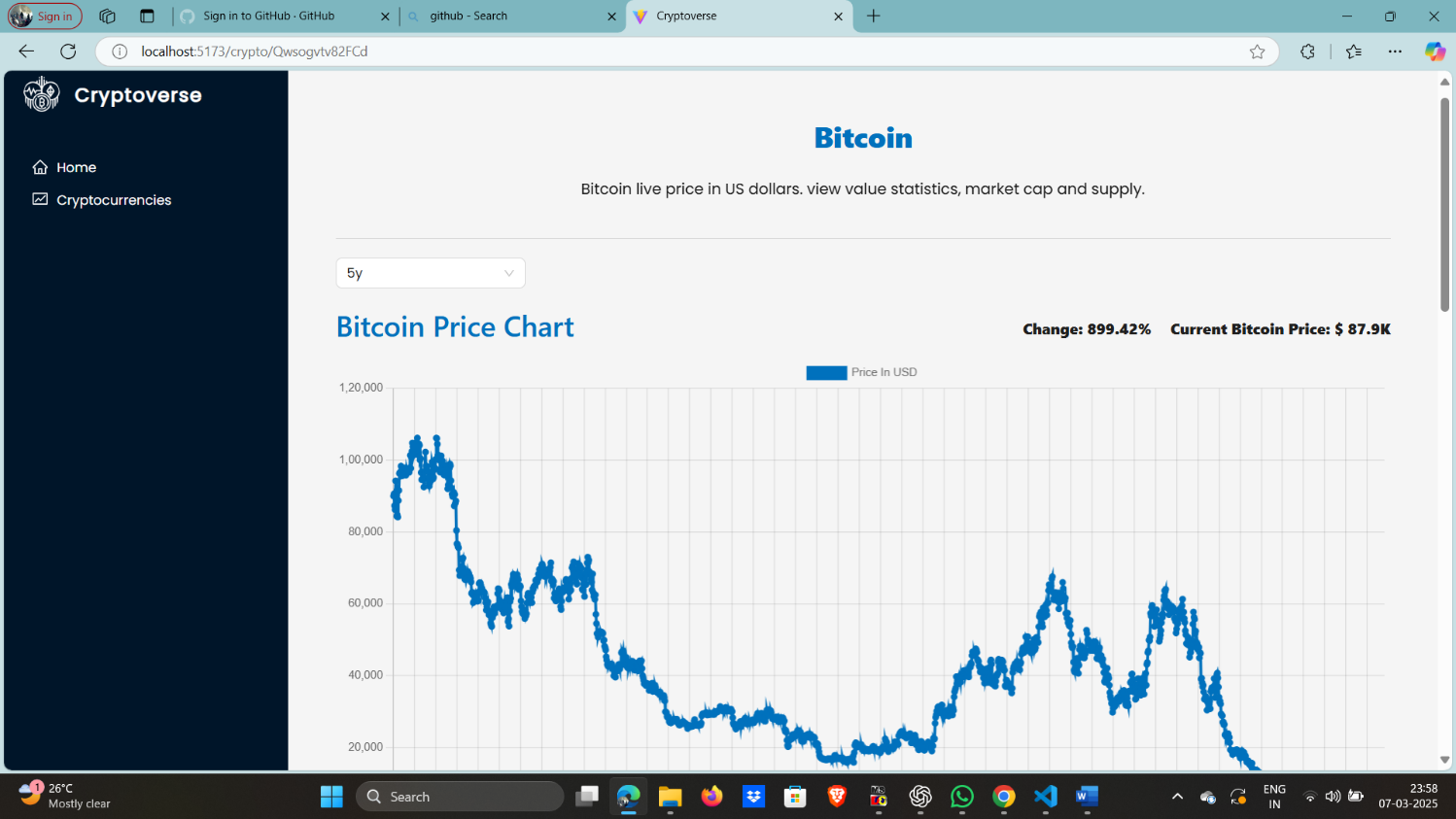
* **Dashboard**



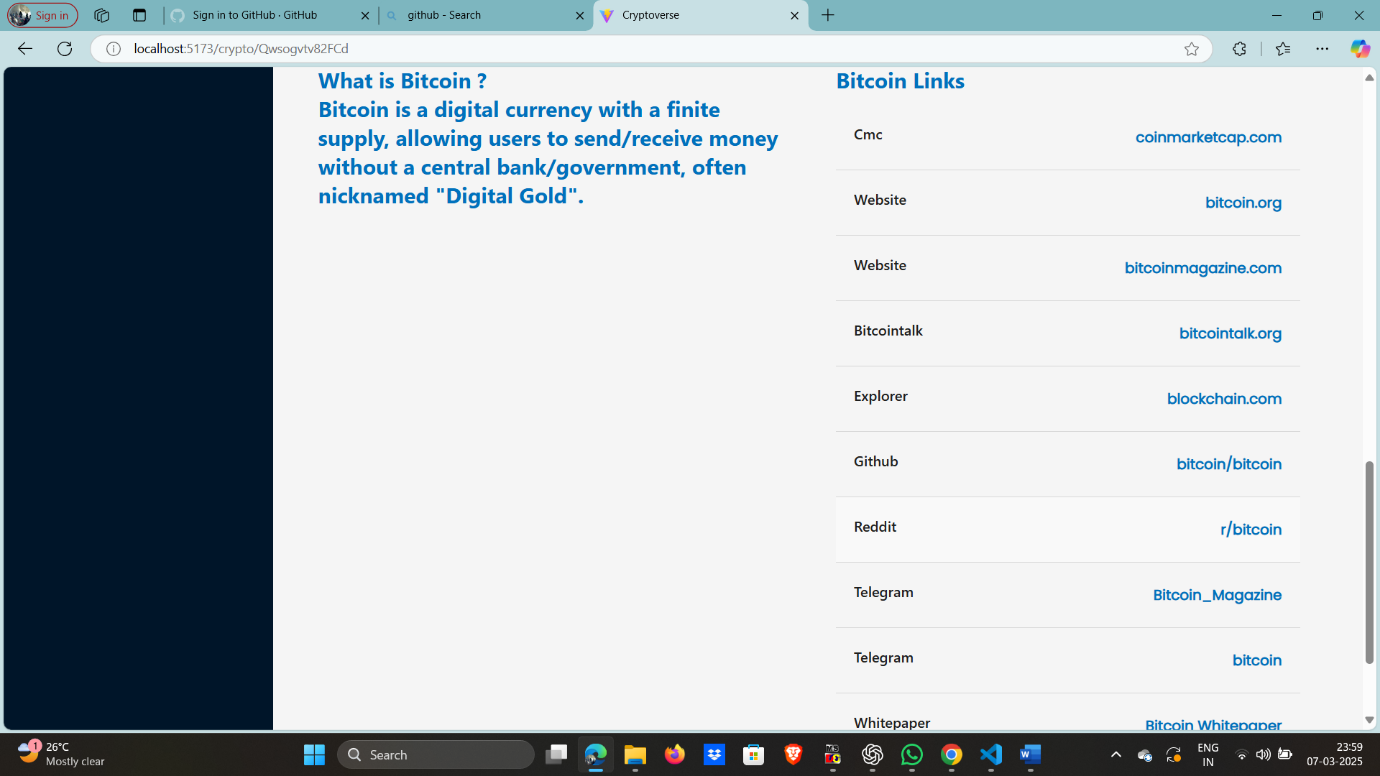
* **Crypto Details Page**



* **Portfolio Page**



* **News Section**



**10. Styling**

**CSS Frameworks/Libraries:**

* Tailwind CSS for modern, responsive design.
* Styled-Components for dynamic styling.

**Theming:**

* Dark mode toggle with persisted user settings.
* Customizable themes with predefined color palettes.

**11. Testing**

**Testing Strategy:**

* **Jest & React Testing Library** for unit and integration tests.
* **Cypress** for end-to-end (E2E) testing.

**Code Coverage:**

npm test # or yarn test

**12. Known Issues**

* Some API endpoints have rate limits.
* Improve responsiveness for smaller screens.
* Optimize search performance for large datasets.

**13. Future Enhancements**

* Implement user authentication (OAuth/Firebase/Auth0).
* Add advanced analytics and AI-driven market predictions.
* Expand fiat currency support and multi-language options.
* Improve accessibility with screen reader support.