CRYPTOVERSE: A Cryptocurrency Dashboard

1. Introduction:

Project Title: CRYPTOVERSE: A Cryptocurrency Dashboard

Team Members:

- Jayasri K (Team Leader)
- Danusa R
- Vishali M
- Poojithha B
- Yuvasri S B

2. Project Overview:

- Purpose:

Cryptoverse is a comprehensive cryptocurrency dashboard that provides investors with deep insights into market trends through historical price analysis spanning five years. With interactive charts, advanced search functionality, and seamless navigation, it enables users to track top-performing assets and make informed investment decisions.

- Key Features:
  - ➢ Historical Data Analysis: Access detailed price trends for various cryptocurrencies.
  - ➢ Interactive Charts & Visualizations: Gain insights through visually engaging graphs and analytical tools.
  - ➢ Advanced Search & Filtering: Quickly find and compare cryptocurrencies based on performance metrics.
  - ➢ User-Friendly Dashboard: Intuitive interface for smooth navigation and data exploration.
  - ➢ Educational Insights: Learn about cryptocurrency market dynamics and investment strategies

3. Architecture:

Component Structure:

Cryptoverse follows a React + Redux architecture, with APIs providing real-time cryptocurrency and news data. The architecture consists of the following key layers:

- Frontend (React): The UI is built using React.js with Ant Design for styling.
- State Management (Redux Toolkit): Handles API data and user interactions efficiently.
- Data Fetching (Axios & RapidAPI): APIs provide live cryptocurrency data and news.

- Routing (React Router): Allows seamless navigation between pages.
- Visualization (Chart.js & React-Chartjs-2): Displays historical price trends and insights.

4. Setup Instructions:

Prerequisites:

- Node.js
- Visual Studio Code
- Git

Installation

1. Clone the repository:
2. Navigate to the project directory
3. Install dependencies
4. Set up environment variables
5. Start the application
6. Access the application: **http://localhost:3000**
7. git clone: https://github.com/jai1910/cryptoverse_dashboard

5. Folder Structure:

- cryptoverse
  - ❖ public   # Static assets
    - ➢ favicon.ico   # App favicon
    - ➢ index.html   # Main HTML file
    - ➢ manifest.json   # Metadata for PWA
    - ➢ robots.txt   # SEO and web crawlers
  - ❖ src   # Main source code
    - ➢ assets   # Images, icons, and static resources
    - ➢ components   # Reusable UI components
    - ➢ Navbar.js   # Navigation bar
    - ➢ CryptoCard.js # Card component for cryptocurrency
    - ➢ NewsCard.js   # Card component for news
    - ➢ LineChart.js   # Chart for price trends
    - ➢ Loader.js   # Loading spinner
    - ➢ index.js   # Component exports
  - ❖ pages   # Main pages
    - ➢ Homepage.js   # Dashboard with market stats
    - ➢ Cryptocurrencies.js # List of cryptocurrencies
    - ➢ CryptoDetails.js # Individual crypto details
    - ➢ News.js   # Cryptocurrency news
    - ➢ index.js   # Page exports
  - ❖ redux   # Redux store and slices
    - ➢ store.js   # Redux store configuration
    - ➢ cryptoSlice.js # Cryptocurrency API state
    - ➢ newsSlice.js   # News API state
    - ➢ index.js   # Redux exports
  - ❖ utils   # Utility functions
    - ➢ fetchData.js   # API request helper

- formatData.js # Data formatting functions
- index.js   # Utility exports
- App.js   # Main app component
- index.js   # Entry point
- store.js   # Redux store setup
- routes.js   # React Router configuration
- ❖ node_modules   # Dependencies (auto-generated)
  - .env   # Environment variables
  - .gitignore   # Git ignore file
  - package.json   # Project dependencies and scripts
  - package-lock.json   # Exact dependency versions
  - README.md   # Project documentation
  - .eslintrc.js   # ESLint configuration

6. Running the Application:
To run the application locally, use the command:
npm start

7. Component Documentation:

Main Components

- Navbar.jsx: Navigation bar for switching between different sections of the application.
- Homepage.jsx: Displays an overview of the cryptocurrency market, including top-performing coins and global statistics.
- Cryptocurrencies.jsx: Lists all available cryptocurrencies with filtering and sorting options.
- CryptoDetails.jsx: Provides in-depth details of a selected cryptocurrency, including historical price charts and market data.
- News.jsx: Fetches and displays the latest cryptocurrency-related news articles.

Reusable Components

- CryptoCard.jsx: Dynamically renders individual cryptocurrency details in a card format.
- NewsCard.jsx: Displays news articles with images, sources, and short descriptions.
- LineChart.jsx: Visualizes historical price trends of a selected cryptocurrency using Chart.js.
- Loader.jsx: Displays a loading spinner during API data fetching

8. State Management:

- Global State (Redux Toolkit)
  - The application uses Redux Toolkit for managing global state.
  - Cryptocurrency and news data are stored in Redux slices for efficient state management.
  - API calls are handled asynchronously using Redux Thunks.
- Local State (React Hooks)
  - useState: Manages component-specific states like search input and UI toggles.
  - useEffect: Fetches API data and updates components dynamically when dependencies change.

9. User Interface:

- The application is styled using Ant Design (antd) for a modern and consistent UI.
- CSS Modules ensure component-specific styling without conflicts.
- Chart.js & React-Chartjs-2 are used for interactive and visually appealing cryptocurrency trend charts.
- The UI is fully responsive, ensuring smooth functionality on both desktop and mobile devices.
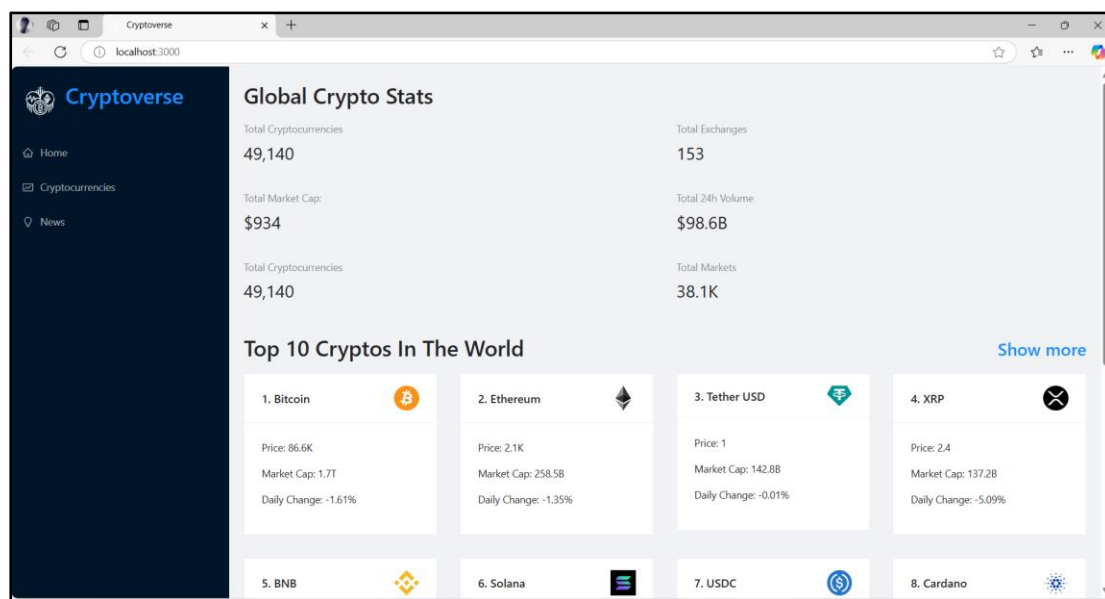
10. Styling:

- CSS Framework: Ant Design (antd) for a modern UI.
- Custom Styles: CSS Modules for scoped styling.
- Responsive Design: Fully optimized for all devices.
- Chart Styling: Custom-styled Chart.js for data visualization.
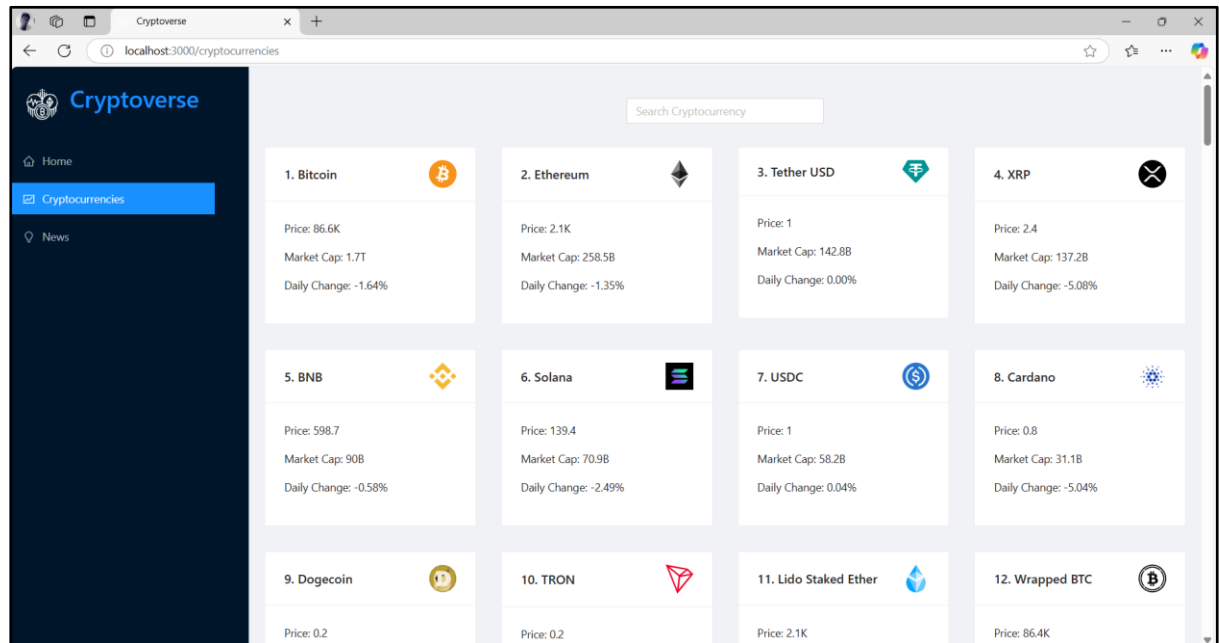
11. Testing:
- Testing Strategy:
  - Unit Testing: Uses Jest and React Testing Library to test individual components.
  - Integration Testing: Ensures Redux state management and API interactions work correctly.
  - End-to-End Testing (Optional): Can be implemented using Cypress for user flow testing.
- Code Coverage:
  - Jest Coverage Reports help track test coverage.
  - Focus on key components like CryptoCard, NewsCard, and LineChart.
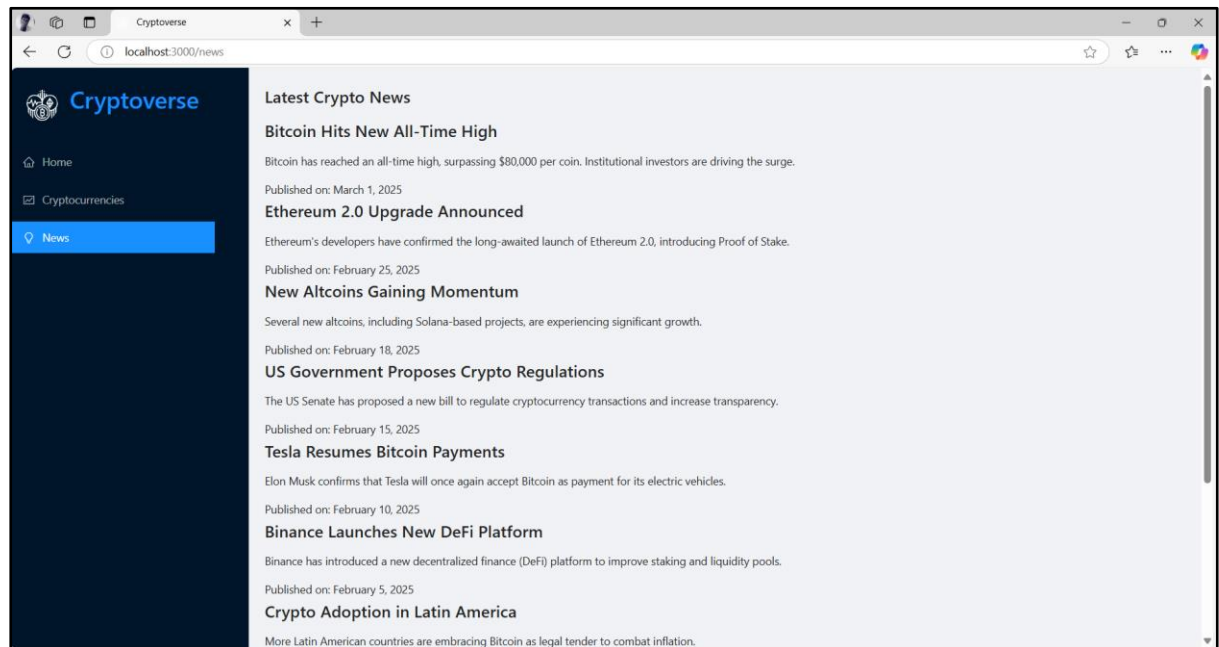
12. Screenshots or Demo:

- Home page:

- Cryptoverse page:

- News page:

13. Known Issues:

- API Rate Limits: RapidAPI may impose rate limits, causing temporary unavailability of data.

- Data Delay: Cryptocurrency price updates might have slight delays due to API response times.
- Mobile Responsiveness: Some charts may not display perfectly on smaller screens.
- Browser Compatibility: Certain UI components may behave differently in older browser versions.

14. Future Enhancements:

- Real-Time Data Updates: Implement WebSockets for live cryptocurrency price tracking.
- Dark Mode: Add a theme switcher for light/dark mode.
- Portfolio Tracker: Allow users to track their crypto holdings and gains/losses.
- Enhanced Charts: Introduce interactive charts with multiple timeframes and indicators.
- Mobile Optimization: Improve UI/UX for better performance on mobile devices.

Submitted By:

Jayasri K, Vishali M, Danusa R, Yuvasri SB, Poojithha B

Date: 07.03.2025