A Cryptocurrency Dashboard

1.Introduction

Project Title: A Cryptocurrency Dashboard

Team Members:

1.Anushka Pathak – pathakanushka2005@gmail.com (Team leader)

2.Sakthipriya.K – sakthima460@gmail.com (Team member)

3.Meenaloshini.S – crazycinderella6@gmail.com (Team member)

4.Nusrath Sulthana.S [-mdafrarmdafrar9@gmail.com](mailto:mdafrarmdafrar9@gmail.com) (Team member)

2.Project Overview

Purpose:

The cryptocurrency dashboard, Cryptoverse, aims to offer investors a detailed analysis of historical price data. By providing a visually rich interface, users can track market movements, compare different cryptocurrencies, and gain insights into price fluctuations. This tool assists investors in making data-driven decisions and understanding the evolving nature of the crypto market.

Features:

Interactive charts displaying historical price data.

Search functionality for quick access to specific cryptocurrencies.

Comparison of different cryptocurrencies' performance over time.

Educational insights into market trends.

User-friendly navigation with a well-structured layout.

Cryptoverse enhances user experience by integrating multiple analytical features, enabling traders to perform comprehensive market research in one platform. The project is structured to provide real-time market updates, ensuring users are always informed about price changes and trends. By leveraging React.js and Redux, the dashboard offers a highly responsive and efficient interface.

The inclusion of a search feature allows users to quickly locate specific cryptocurrencies, while interactive charts facilitate deeper

market analysis. Users can view historical trends, analyze volatility, and track asset performance with just a few clicks. By focusing on accessibility and data visualization, this project aims to make cryptocurrency trading more approachable for new investors while offering advanced tools for seasoned traders.

3.Architecture

Component Structure:

The dashboard consists of multiple React components, including charts, navigation, search functionality, and currency details.

Each component is designed for modularity and reusability.

State Management:

Redux Toolkit is used to manage global state efficiently, particularly for fetching and storing cryptocurrency data from APIs.

This ensures that data retrieval is optimized and the UI remains responsive.

Routing:

React Router DOM is used for seamless navigation between different pages such as the homepage, cryptocurrency listings, and detailed currency insights.

Ensures dynamic content loading without unnecessary page refreshes.

The architecture follows a modular approach, ensuring scalability and maintainability. Each React component is designed for

reusability, allowing for easy integration of new features in future updates. The project also follows best practices for API management, ensuring optimal performance when handling large datasets.

By leveraging a component-based structure, the project enables seamless UI updates without affecting the overall application flow. The Redux store effectively handles state changes, ensuring smooth data fetching and rendering. Additionally, the routing system enables dynamic page transitions, improving user experience.
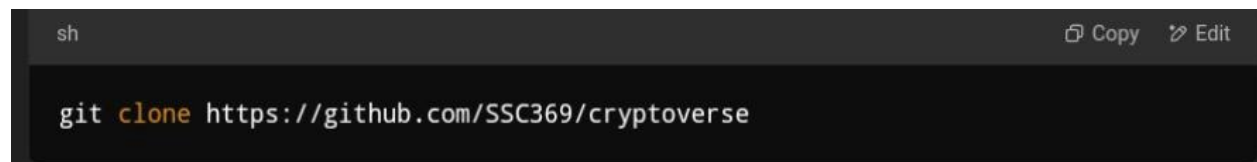
4.Setup Instructions

Prerequisites:

Node.js and npm installed on the system.

Basic knowledge of React.js and Redux.

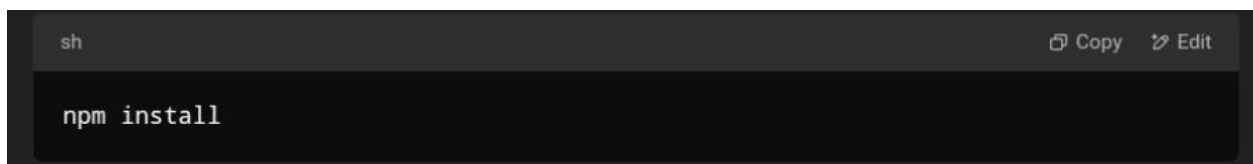Installation: Clone the repository using

```sh
git clone https://github.com/SSC369/cryptoverse
```
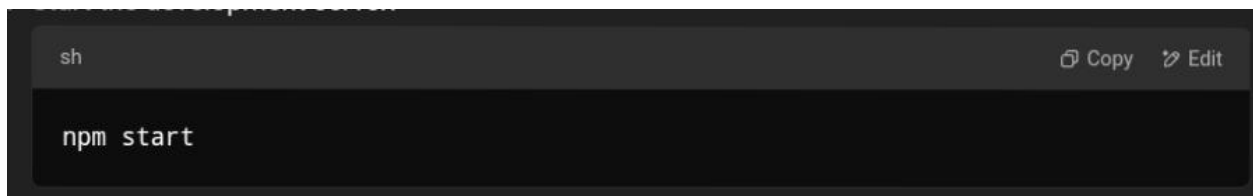
Navigate to the project folder:

```sh
cd crypto
```

Install dependencies:

```sh
npm install
```

Start the development server :

```sh
npm start
```

Open the browser and navigate to http://localhost:3000.

Ensuring that all dependencies are installed correctly is crucial for the smooth execution of the project. The system should have a stable internet connection to fetch real-time data. It is also recommended to use a modern web browser, such as Google Chrome or Mozilla Firefox, for the best user experience.

5.Folder Structure

Client:

Components/: Contains reusable UI elements.

Pages/: Includes main pages like Home, Cryptocurrency List, and Details.

Services/: Handles API interactions using Redux Toolkit.

Redux/: Manages application state using Redux.

Utilities:

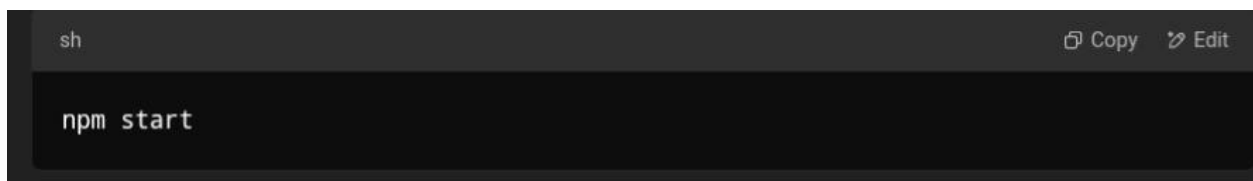Helpers.js: Contains utility functions for data formatting.

Hooks/: Includes custom React hooks for fetching data.

This structure ensures the project remains scalable, organized, and easy to maintain.

6.Running the Application

Start the frontend application using:

```sh
npm start
```

This will launch the development server, allowing users to access the dashboard via http://localhost:3000.

Before running the project, ensure all dependencies are installed and that you have an active internet connection.

7.Component Documentation

Key Components:

CryptoList: Displays a list of cryptocurrencies with search functionality.

CryptoDetails: Provides historical data, trends, and stats for selected currencies.

ChartComponent: Renders interactive charts using Chart.js.

Navbar: Handles navigation across different pages.

SearchBar: Allows users to search for specific cryptocurrencies.

Each of these components is structured to work efficiently with Redux state management, ensuring smooth data flow. Components use props to receive relevant data and update dynamically as users interact with the dashboard.

The ChartComponent is responsible for rendering historical price trends in an interactive way. It utilizes react-chartjs-2 to provide dynamic line charts, making it easier for users to visualize trends over time.

Additionally, the CryptoDetails component fetches real-time data for a selected cryptocurrency, providing crucial statistics like market cap, 24-hour volume, and price fluctuations. It ensures that users receive the most up-to-date market insights.

Reusable components like buttons, cards, and modals are used throughout the application to maintain UI consistency and reduce redundant code. The project follows a component-based architecture to improve scalability and maintainability.

8.State management

Global State Management:

The project uses Redux Toolkit to manage the application's global state efficiently.

API data, including cryptocurrency prices and details, are stored in the Redux store for optimal performance.

Redux middleware is used to handle asynchronous API requests seamlessly.

Local State Management:

Components like SearchBar use local state for managing user input dynamically.

UI-related states, such as loading indicators and modals, are managed locally within components.

The application follows best practices for Redux slice architecture, where each feature (such as cryptocurrencies and charts) has its own slice of the Redux store. This ensures modularity and better performance.

State changes are efficiently handled using Redux's useSelector and useDispatch hooks. These hooks ensure that components re-render only when necessary, preventing unnecessary API calls and improving efficiency.

By combining Redux Toolkit with React Query, the project ensures optimal state management, reducing API load while keeping the UI responsive.

9.User Interface

The UI is designed to be intuitive, modern, and visually appealing. It consists of several key sections that enhance user experience:

Homepage UI:

Displays global cryptocurrency statistics, including total market cap and exchanges.

Lists the top 10 cryptocurrencies with their current market prices.

Includes a navigation bar for quick access to other pages.

Cryptocurrency List Page:

Displays a comprehensive list of all tracked cryptocurrencies.

Features a search bar to quickly find specific cryptocurrencies.

Uses card-based UI to present cryptocurrency details in a structured format.

Cryptocurrency Details Page:

Provides interactive charts for analyzing historical price trends.

Displays real-time price data, including market cap, 24-hour volume, and price changes.

Includes links to official cryptocurrency websites and resources.

UI Enhancements:

Dark Mode Support: Users can toggle between light and dark themes for better readability.

Mobile Responsiveness: The UI is optimized for both desktop and mobile users.

Loading Indicators: Ensures a smooth user experience while fetching API data.

By focusing on usability and visual aesthetics, the UI enables both beginners and experienced traders to navigate the platform effortlessly.

10.Styling

CSS Frameworks/Libraries Used:

Ant Design: Provides pre-styled UI components like buttons, cards, and input fields.

Styled Components: Used for writing CSS within JavaScript, enabling dynamic styling.

React-ChartJS-2: For rendering interactive line and bar charts.

Styling Approach:

Global Styles: Maintained in a dedicated CSS file for consistent theming.

Component-Specific Styles: Each component has its own styled file to improve modularity.

Dark Mode Integration:

Users can switch between light and dark themes based on their preference.

Theme settings are stored in local storage, so the UI remains consistent on reload.

Animations & Transitions:

CSS transitions are used for smoother interactions.

Components like modals and dropdowns include fade-in effects for a polished UI experience.

The styling follows responsive design principles, ensuring that the dashboard adapts seamlessly to different screen sizes.

11.Testing

Testing Strategy:

The project follows a structured testing approach to ensure reliability and performance.

Unit Testing:

- Key components like CryptoList and CryptoDetails are tested using Jest and React Testing Library.
- Ensures that UI components render correctly and respond to props and state changes.

Integration Testing:

- API calls are tested using Mock Service Worker (MSW) to simulate real-world data fetching.
- Verifies that Redux state updates correctly when API responses change.

End-to-End (E2E) Testing:

- Automated tests are written using Cypress to simulate real user interactions.
- Ensures that users can search for cryptocurrencies, view charts, and navigate between pages.

Code Coverage:

- Coverage Reports: Jest generates code coverage reports to identify untested areas.
- Continuous Integration (CI): Tests are run automatically in CI/CD pipelines to prevent regression issues.

By following a robust testing strategy, the project ensures that the dashboard remains reliable, responsive, and free from major bugs.

12. Screenshot & Demo

To better understand the functionality and UI of the **Cryptoverse* cryptocurrency dashboard, below are the links to the project demo and UI screenshots.

Project Demo :

A working demo of the project can be accessed through the following link:

[Project   Demo   Link](
https://drive.google.com/file/d/17QcxXV50ZszkUZD8iVeJCl52IJkPNyxO/view?usp=sharing)

User Interface Screenshots:

The application features a clean and modern UI with different sections for tracking cryptocurrencies. Below are screenshots of key components:
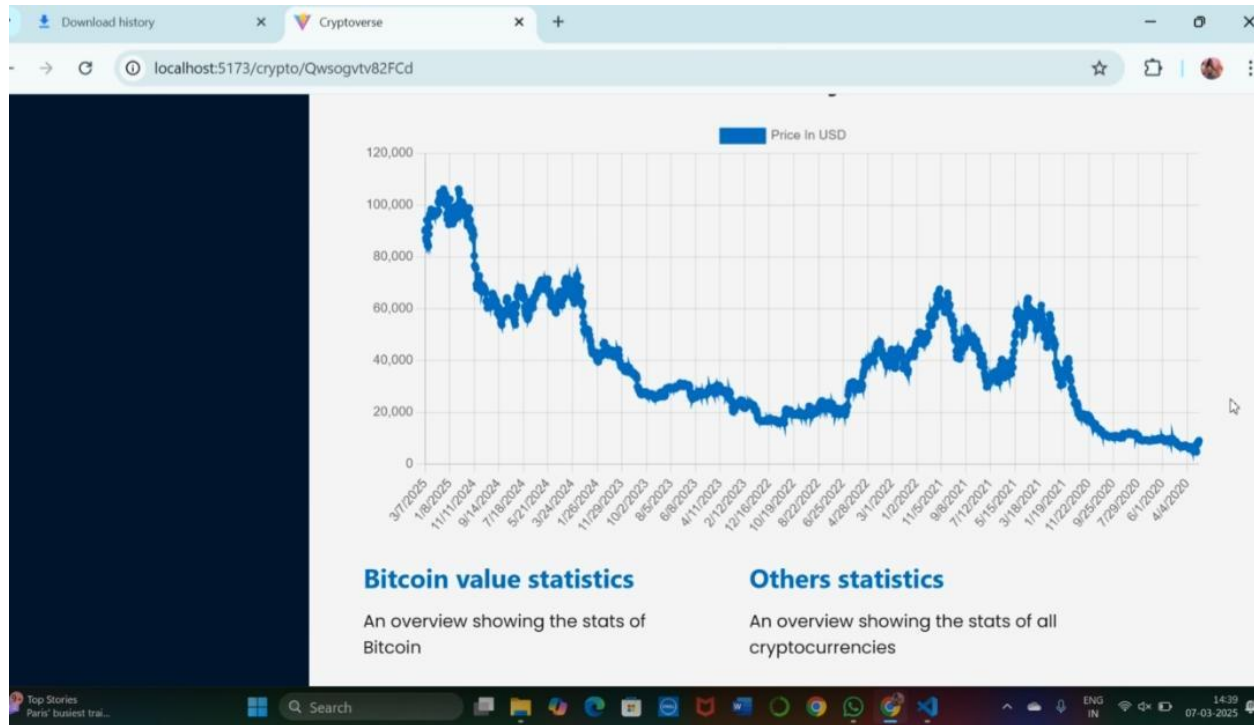
1. Home Page:

- Displays global cryptocurrency statistics, including total market cap, exchanges, and total volume.

- Showcases the top 10 cryptocurrencies with price trends and market fluctuations.

- Includes a navigation bar for seamless movement across different sections.
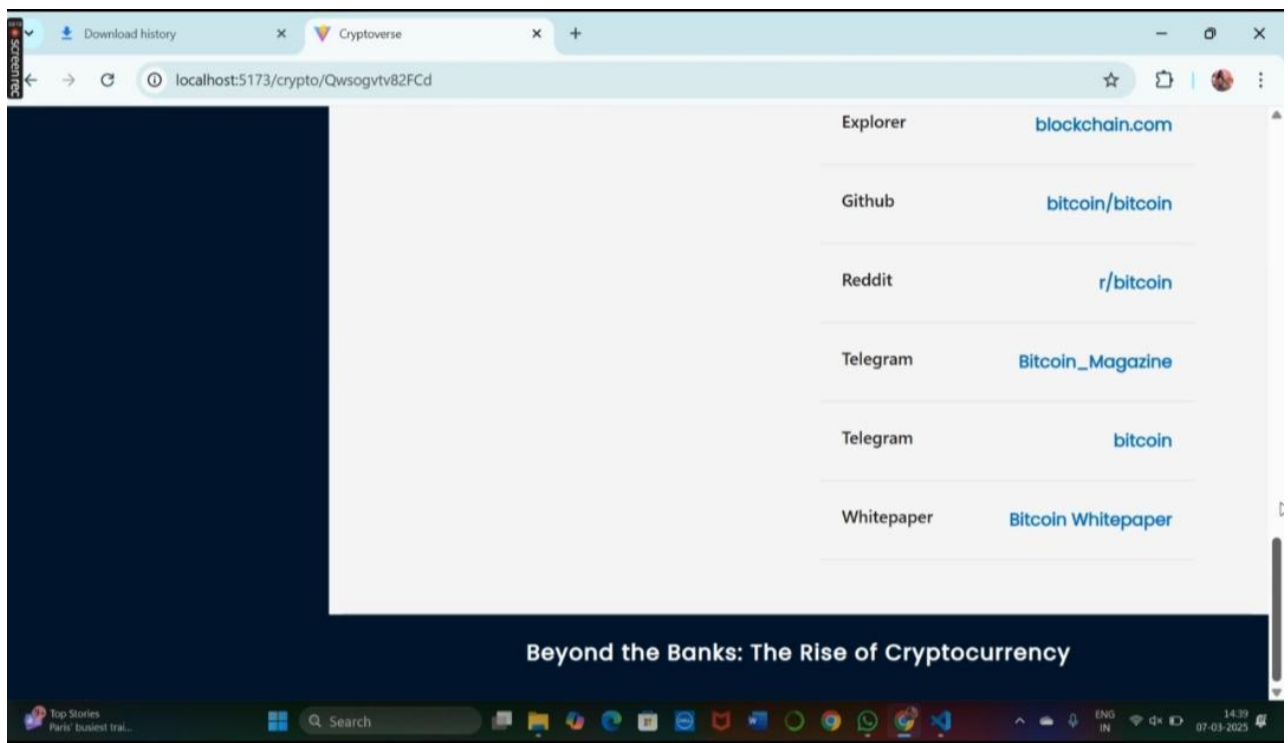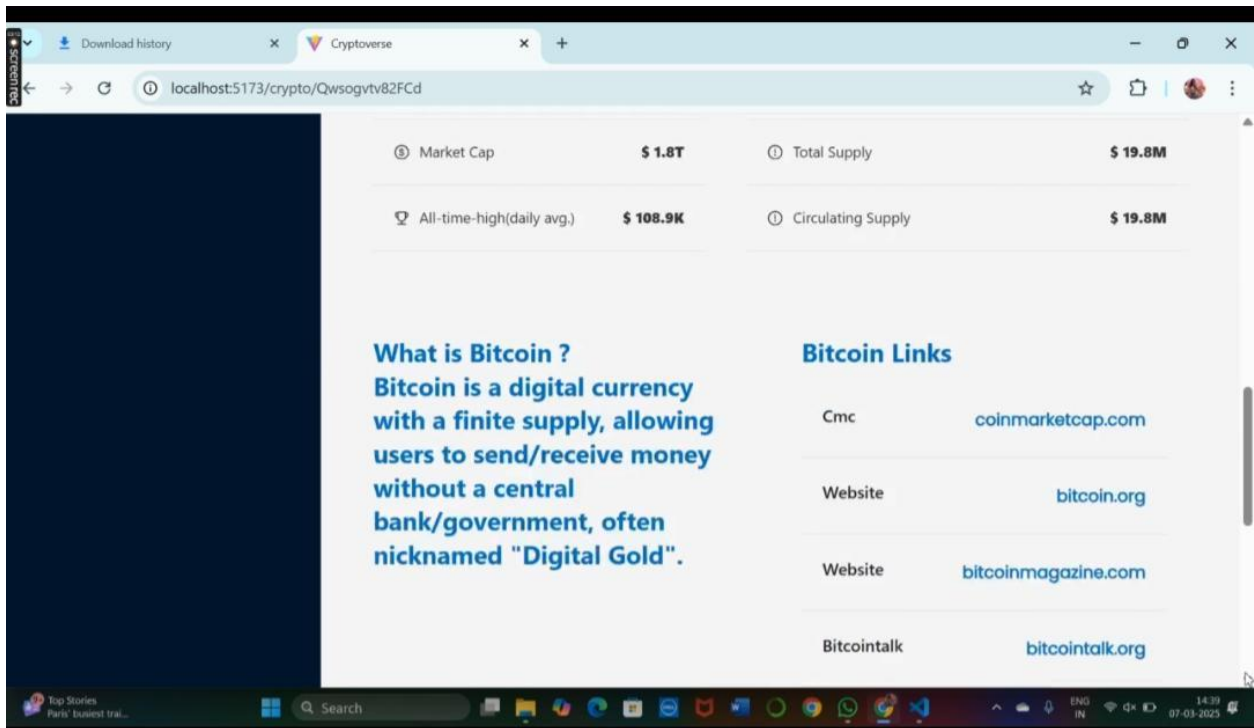

.
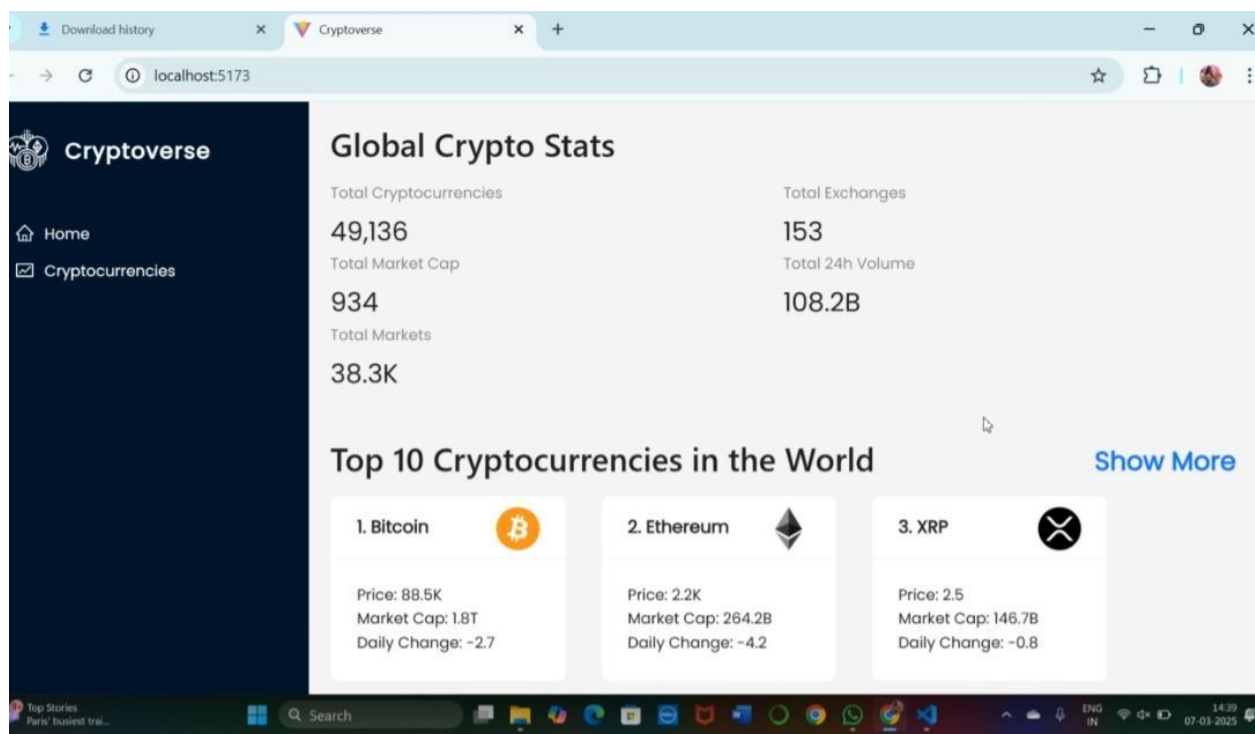
2. Cryptocurrencies Page:
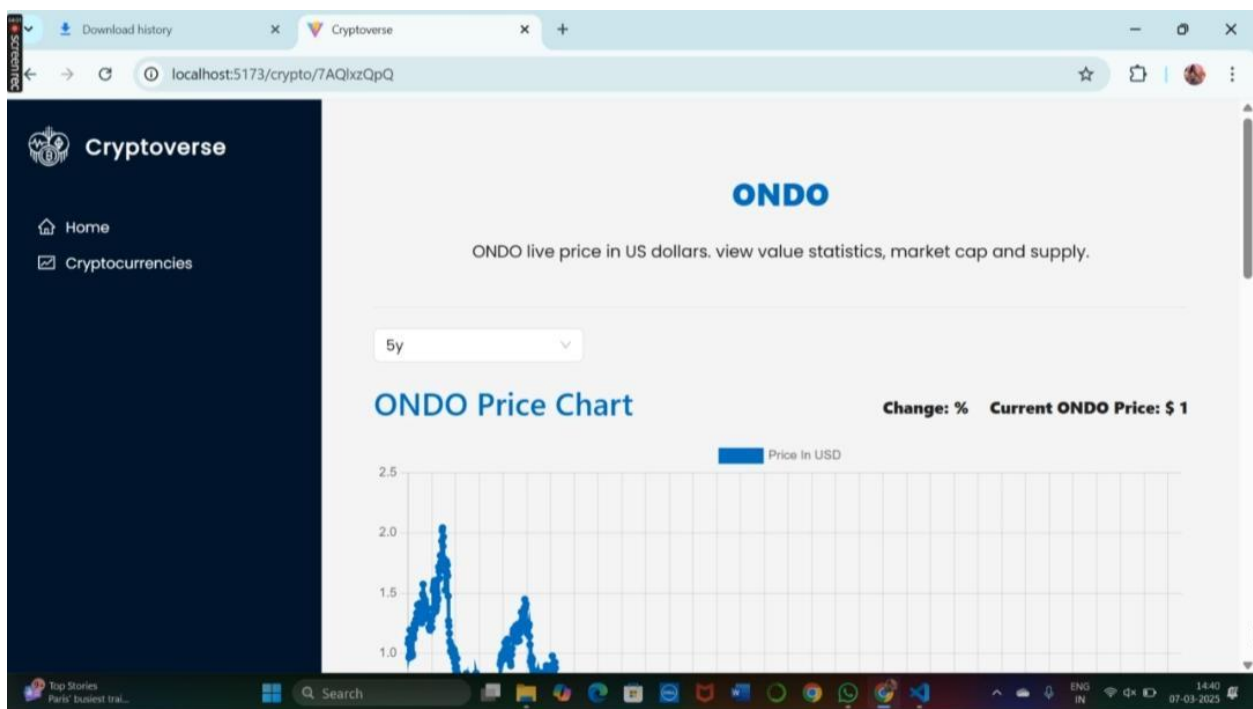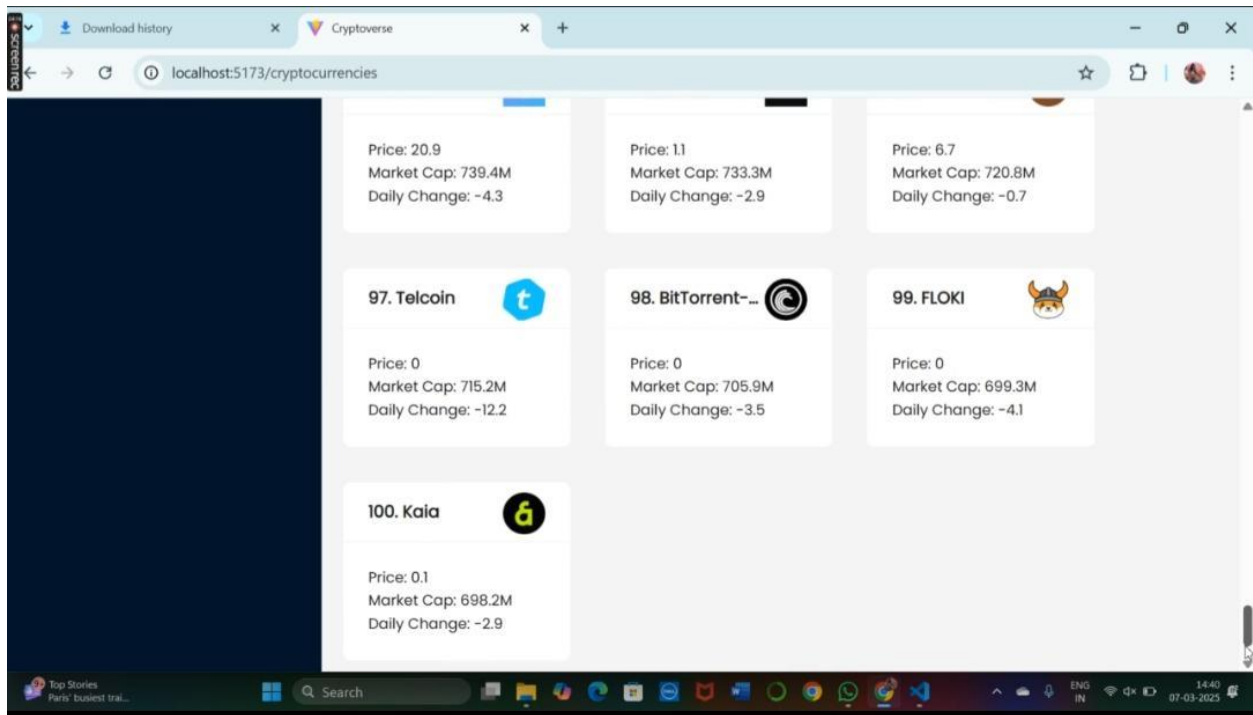
3. Cryptocurrency Details Page:

- Shows an interactive line chart representing historical price fluctuations.

- Provides real-time statistics, including market cap, 24-hour volume, and rank.

- Includes links to official cryptocurrency websites and market resources.

| Market Cap | $ 1.8T | Total Supply | $ 19.8M |
| All-time-high(daily avg.) | $ 108.9K | Circulating Supply | $ 19.8M |

What is Bitcoin ?
Bitcoin is a digital currency with a finite supply, allowing users to send/receive money without a central bank/government, often nicknamed "Digital Gold".

Bitcoin Links

Cmc	coinmarketcap.com
Website	bitcoin.org
Website	bitcoinmagazine.com
Bitcointalk	bitcointalk.org

Explorer	blockchain.com
Github	bitcoin/bitcoin
Reddit	r/bitcoin
Telegram	Bitcoin_Magazine
Telegram	bitcoin
Whitepaper	Bitcoin Whitepaper

Beyond the Banks: The Rise of Cryptocurrency

These screenshots provide an overview of the application's user experience and core functionalities. The project demo link allows users to interact with the dashboard in real time and explore cryptocurrency market trends dynamically.

13.Known Issues

While the application is functional and stable, there are some known issues that need attention:

- **API Rate Limit Issues:**

The cryptocurrency API has request limits, which may cause occasional data retrieval failures.

Implementing caching mechanisms can help reduce excessive API requests.

- **Chart Rendering Bugs:**

In some cases, charts may fail to render correctly on low-resolution or mobile screens.

Optimizing Chart.js configurations can improve performance and responsiveness.

- **Performance Optimization:**

The dashboard may experience slow rendering when loading a large number of cryptocurrencies.

Implementing pagination and lazy loading will enhance efficiency

- **Mobile Compatibility:**

While the app is mostly responsive, certain elements may not be fully optimized for small-screen devices.

Further CSS refinements are needed to improve mobile UI/UX.

Future updates will address these issues to enhance the platform's stability and user experience.

14. Future Enhancements

- Potential improvements include: Adding real-time price updates for a more dynamic experience.
- Enhancing search and filter functionality for better usability.
- Implementing additional chart types such as candlestick graphs.
- Providing AI-driven insights and predictions on cryptocurrency trends.
- Developing a mobile-friendly version for better accessibility.
- Introducing community-driven insights, allowing users to share opinions on market trends.
- Expanding API integrations for a broader dataset.
- Enhancing security features to prevent unauthorized data access.
- By continuously improving the platform, Cryptoverse aims to remain a valuable tool for cryptocurrency investors.