# INTRODUCTION

**1.1 PROJECT TITLE**

Cryptocurrency Dashboard

## 1.2 TEAM MEMBERS

**NAME:** HARINI.S

**EMAIL ID**: pikazoo0423@gmail.com

**NAME:** ABITHA AFRIN .S.N

**EMAIL ID:** abithaafrin49@gmail.com

**NAME:** BHAKYA LAKSHMI.V

**EMAIL ID:** pradeepapradeepa451@gmail.com

**NAME:** JANANI BALA.B

**EMAIL ID:** jananibalajanu@gmail.com

# PROJECT OVERVIEW

## 2.1 PURPOSE

The Cryptocurrency Dashboard is a web-based application developed using React that provides real-time data on various cryptocurrencies. The application helps users track market trends, view detailed cryptocurrency information, and analyze historical price movements using interactive charts. It is designed for crypto enthusiasts, traders, and investors who require a comprehensive and user-friendly platform for monitoring cryptocurrency prices and trends.

## 2.2FEATURES

* **Real-time Market Data:** Fetches live cryptocurrency prices and market details from an API.
* **Detailed Cryptocurrency Information:** Displays details such as market cap, trading volume, circulating supply, and price changes over different time frames.
* **Historical Price Charts:** Interactive line charts for analyzing trends over days, weeks, or months.
* **Search & Filter Functionality:** Allows users to search for specific cryptocurrencies and filter based on various criteria.
* **User-friendly UI:** Clean and responsive design for desktop and mobile use.
* **Navigation System:** Seamless routing between different pages using React Router.

# ARCHITECTURE

## 3.1 COMPONENT STRUCTURE

The project follows a modular component-based architecture, ensuring scalability and maintainability. The major components are:

* **Home.jsx:** Displays an overview of top-performing cryptocurrencies.
* **Cryptocurrencies.jsx:** Lists all available cryptocurrencies along with key market data. ➢ **CryptoDetails.jsx:** Provides in-depth information about a selected cryptocurrency, including price charts.
* **LineChart.jsx:** Renders a dynamic line chart for tracking price movements.
* **Navbar.jsx:** Navigation bar for easy access to different sections of the application.
* **Loader.jsx:** Displays a loading animation while fetching data.

## 3.2 STATE MANAGEMENT

The application uses Redux Toolkit for managing global state, ensuring efficient data fetching and state synchronization across components. API data is stored in the Redux store and accessed by different components to maintain consistency and performance.

## 3.3 ROUTING

The application uses React Router for navigation between different pages. Key routes include:

* /: Home page
* /cryptocurrencies: Displays a list of all cryptocurrencies
* /crypto/:id: Provides detailed insights about a specific cryptocurrency

# SETUP INSTRUCTIONS

## 4.1 PREREQUISITES

**Ensure you have the following installed before running the application:**

* Node.js (latest stable version)
* npm (Node Package Manager) or yarn

**4.2 INSTALLATION**

**Follow these steps to set up and run the application:**

**1.Clone the repository:**

git clone <repository-url>

**2**.**Navigate to the project directory:**

cd cryptocurrency-dashboard

**3.Install dependencies:**

npm install

**4.Start the development server:**

npm run dev

**5.Open the application in your browser at:**

http://localhost:3000

# FOLDER STRUCTURE

cryptocurrency-dashboard/

├── public/

│ ├── \_redirects

│ ├── vite.svg

├── src/

│ ├── assets/

│ │ ├── cryptocurrency.png

│ ├── components/

│ │ ├── Home.jsx

│ │ ├── Cryptocurrencies.jsx

│ │ ├── CryptoDetails.jsx

│ │ ├── LineChart.jsx

│ │ ├── Navbar.jsx

│ │ ├── Loader.jsx

│ │ ├── index.js

│ ├── services/

│ │ ├── cryptoApi.js

│ ├── app/

│ │ ├── store.js

│ ├── index.css

│ ├── App.css

│ ├── main.jsx

│ ├── App.jsx ├── .gitignore

├── .eslintrc.cjs

├── index.html

├── package.json

├── package-lock.json

├── vite.config.js

├── README.md

# RUNNING THE APPLICATION

**To start the development server, use:**

npm run dev

# COMPONENT DOCUMENTATION

## 7.1 KEY COMPONENTS

* **Home.jsx:** Displays an overview of the cryptocurrency market.
* **Cryptocurrencies.jsx:** Fetches and displays a list of cryptocurrencies with their details.
* **CryptoDetails.jsx:** Provides a deep dive into individual cryptocurrency data.
* **LineChart.jsx:** Displays historical price trends in a chart format.
* **Navbar.jsx:** Enables smooth navigation between pages.

**7.2 REUSABLE COMPONENTS**

* **Loader.jsx:** Used for showing a loading animation when fetching data from the API.

# STATE MANAGEMENT

## 8.1 GLOBAL STATE

* Redux Toolkit manages global state and stores API responses for efficient rendering.
* Data is fetched using cryptoApi.js, which interacts with the cryptocurrency API.

**8.2 LOCAL STATE**

Local state is used within components for handling UI elements like search inputs and filters.

# USER INTERFACE

* Fully responsive for both desktop and mobile devices.
* Designed with clean UI components to enhance user experience.
* Includes real-time data updates for better engagement.

# STYLING

**10.1 CSS Frameworks/Libraries**

* Uses CSS and Tailwind CSS for styling.

**10.2 Theming**

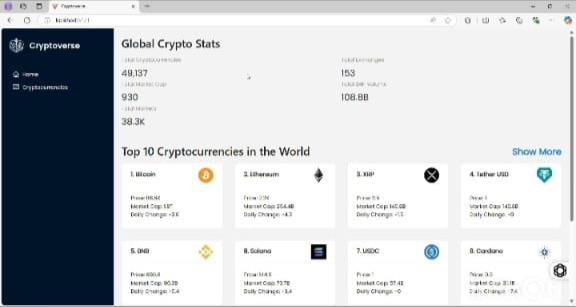
* Custom color themes can be modified in index.css.

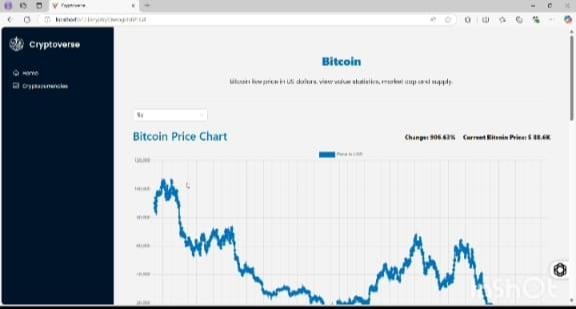
# TESTING

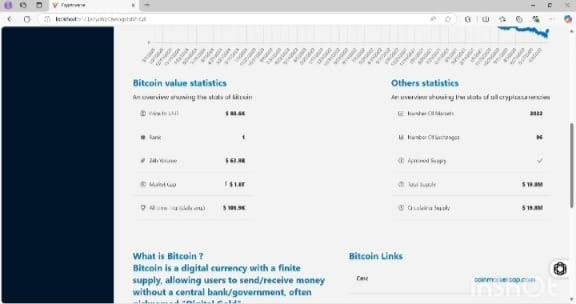
**11.1 Testing Strategy**

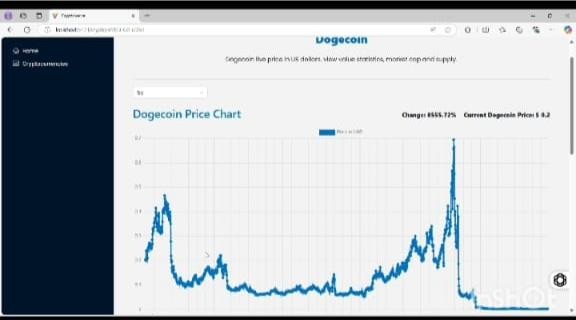
* Unit tests can be implemented using Jest and React Testing Library.
* API calls should be mocked for testing data retrieval and state updates.

# SCREENSHOTS









# KNOWN ISSUES

* API Rate Limits: Some API services may impose rate limits, causing delays in data updates.
* Incomplete Data: Some cryptocurrencies may not have full historical price data available.

# FUTURE ENHANCEMENTS

* **User Authentication:** Allow users to create accounts and track their favorite cryptocurrencies.
* **Dark Mode:** Implement a dark mode option for better accessibility.
* **Portfolio Tracking:** Enable users to create and monitor a virtual cryptocurrency portfolio.