**Naan Mudhalvan- Front end Development and Database Administration**

**VALLIAMMAL COLLEGE FOR WOMEN**

**(College Code: 1363)**

**Department of Computer Science**

**Project title: Cryptoverse : A Cryptocurrency Dashboard**

**NM ID: SWTID1741250110150019, Team-150019**

**Frontend Development with React.js**

***Project Documentation for Cryptoverse***

**1.** **Introduction**

• **Project** **Title: Cryptoverse**

• **Team** **Members**:

**Shalini M** (**Team** **Leader**)

**Sandhiya J**

**Sarumi K**

**Sharmila R**

[Email Id: [m.shalini22cs073@gmail.com](mailto:m.shalini22cs073@gmail.com) ]

[Email Id: [sandhiyauma09082004@gmail.com](mailto:sandhiyauma09082004@gmail.com) ]

[Email Id: [sarumi22cs072@gmail.com](mailto:sarumi22cs072@gmail.com) ]

[Email Id: [sharmilar22cs074@gmail.com ]](mailto:sharmilar22cs074@gmail.com%20])

**2.** **Project** **Overview**

• **Purpose**:

CryptoVerse is a digital platform that provides real-time updates, insights, and information about cryptocurrencies. Users can track market trends, view historical data, and explore various digital assets.

• **Features**:

o Cryptocurrency Market Overview.

o Live Price Updates.

o User Authentication.

o Trending Coins and Market Insights.

**3.** **Architecture**

• **Component** **Structure**:

The application is built using React.js with a component-based architecture. Major components include:

**o** **Header**: Contains the navigation bar and search bar.

**o Sidebar**: Displays trending cryptocurrencies and navigation links.

**o Home Page**: Shows Market Trends, Most Popular Cryptos.

**o Search Page**: Allows users to search for specific cryptocurrencies and tokens.

• **State** **Management**:

The application uses **Redux** for global state management. The Redux store manages user authentication, market data and search results.

• **Routing**:

The application uses **React** **Router** for navigation. Routes include:

o /: Home page

o /search: Search page

o /login: User login page

**4.** **Setup** **Instructions**

• **Prerequisites**:

o Node.js (v16 or higher)

o npm (v8 or higher)

o Git

• **Installation**:

1. Clone the repository: git clone : <https://github.com/comp22cs073/Cryptoverse_Shalini>

2. Install dependencies: npm install

3. Configure environment variables: Create a .env file in the client directory and add the necessary variables (e.g., API keys).

4. Start the development server: npm start

**5.** **Folder** **Structure**

• **Client**:

o **src/components:** # Reusable components (Header, Marketlist, etc.)

o **src/pages:** # Page components (HomePage, SearchPage, etc.) o **src/assets:** # Images, icons, and other static files

o **src/redux:** # Redux store, actions, and reducers o **src/utils:** # Utility functions and helpers

o **App.js:** # Main application component o **index.js:** # Entry point

• **Utilities**:

o **api.js**: Handles API requests to the backend.

o **auth.js**: Manages user authentication and token storage.

**6.** **Running** **the** **Application**

**Frontend**:

o To start the frontend server, run the following command in the client directory: npm start

o npm install

o The application will be available at http://localhost:3000

**7.** **Component** **Documentation**

• **Key** **Components**:

o **Header**: Displays the navigation bar and search bar.

▪ Props: onSearch (function to handle search queries).

o MarketList: Displays a list of cryptocurrencies with prices.

▪ Props: cryptos (array containing market details), onSelect (function to handle selection).

• **Reusable** **Components**:

o **Button**: A customizable button component.

▪ Props: text, onClick, disabled.

o **Input**: A reusable input field for forms and search.

▪ Props: type, placeholder, value, onChange.

**8.** **State** **Management**

• **Global** **State**:

The Redux store manages the following global states:

o **user:** Current authenticated user.

o **marketdata:** Live cryptocurrency market data and trends.

o **searchResults:** Results from the search functionality.

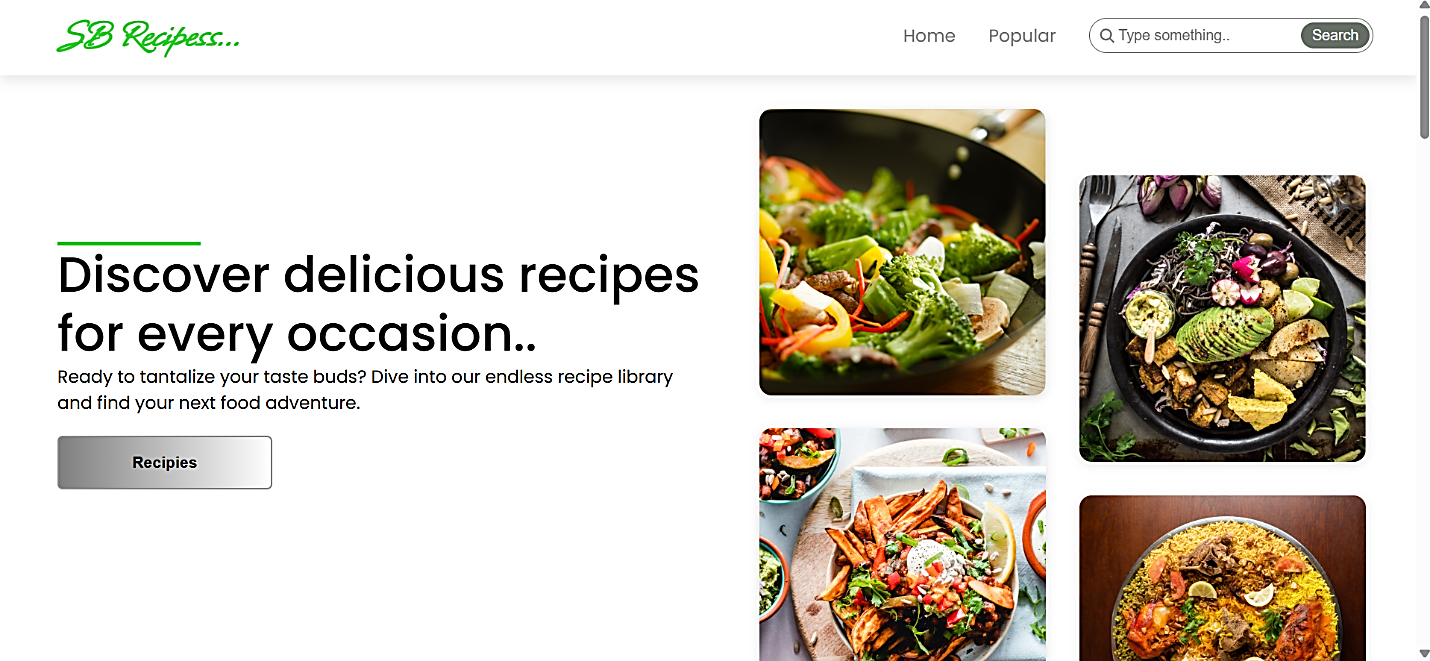
• **Local** **State**:

Local state is managed using React's useState hook within components. For example, the SearchPage component manages the search query input locally.

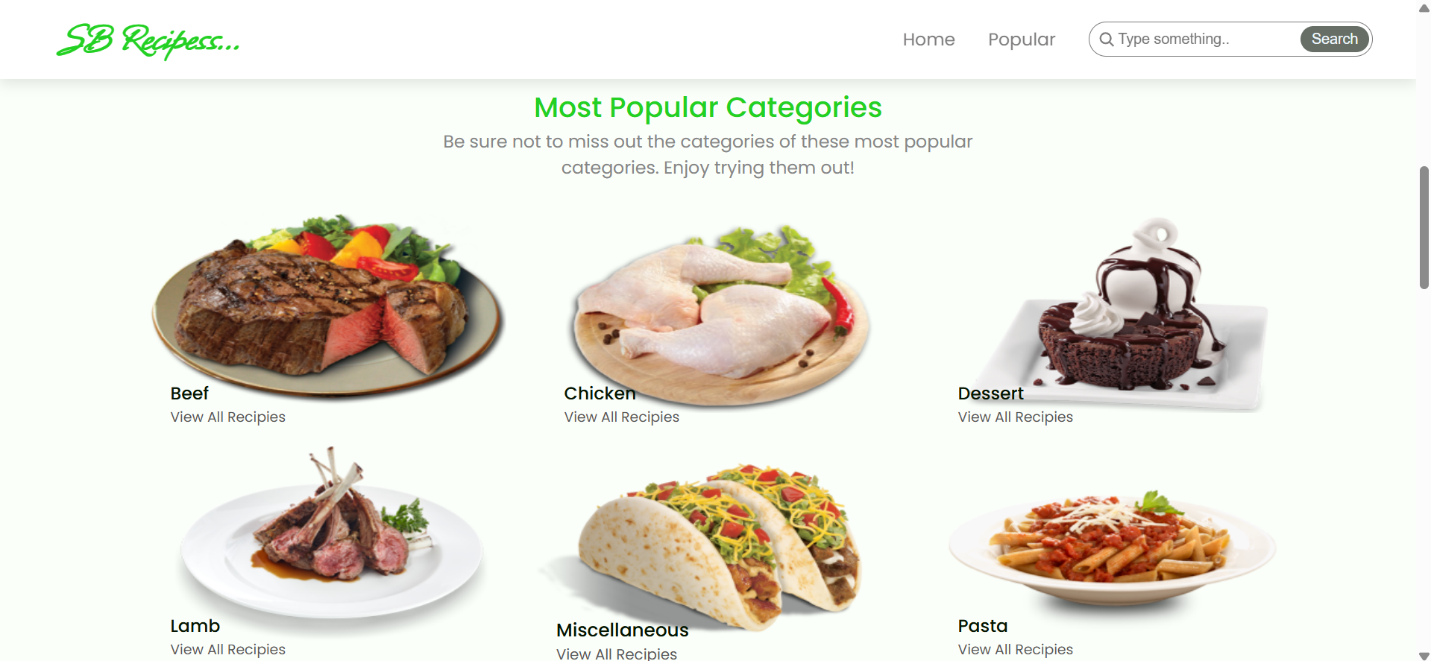
**9.** **User** **Interface**

• **Screenshots**

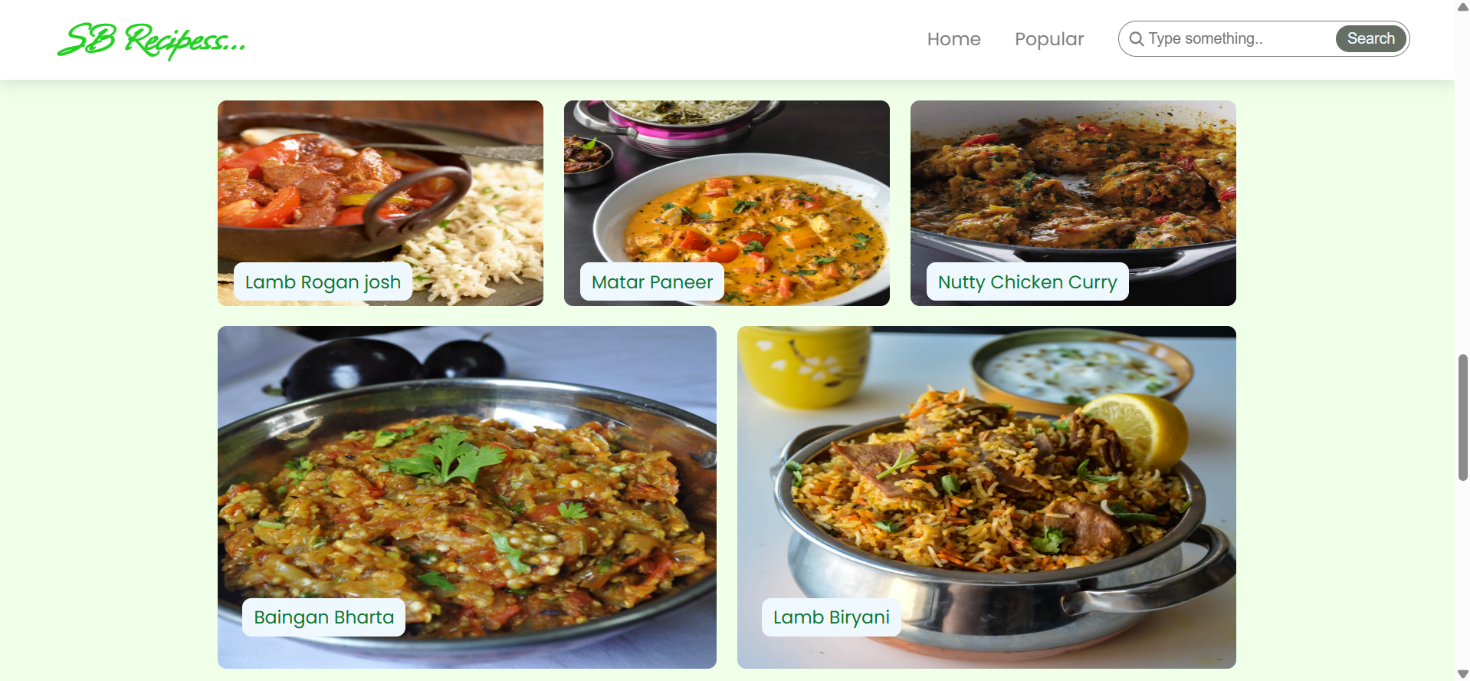
o **Home** **Page:** Displays trending cryptocurrencies and market trends.



o **popular cryptocurrencies :** Highlights the most traded and valuable coins.



o **Market Insights:** Displays market fluctuations and historical data charts



**10.** **Styling**

• **CSS** **Frameworks/Libraries**:

The application uses **Styled-Components** for styling. This allows for modular and scoped CSS within components.

• **Theming**:

A custom theme is implemented using Styled-Components, with support for light and dark modes.

**11.** **Testing**

• **Testing** **Strategy**:

o **Unit** **Testing:** Using **Jest** and **React** **Testing** **Library**.

o **Integration** **Testing**: Ensures that components work together as expected.

o **End-to-End** **Testing:** **Cypress** is used for end-to-end testing of user flows.

• **Code** **Coverage**:

o Code coverage is monitored using Jest’s built in coverage tool. The current coverage is 85%.

**12.** **Demo**

• **Demo** **Link:**

<https://drive.google.com/file/d/1tP28Jpqky8IvxwWZs83iND1WhqB7Z_Ay/view?usp=drive_link>

**13.** **Known** **Issues**

• **Issue** **1**: The search functionality is slow with large datasets.

**14.** **Future** **Enhancements**

• **Future** **Features**:

o Add support for user profiles and portfolio sharing.

o Implement a recommendation engine for personalized cryptocurrency insights.

o Add animations and transitions for a smoother user experience.

This documentation provides a comprehensive overview of the **crytoverse** project, including its architecture, setup instructions, and future plans.