

# Project Documentation

## Introduction

**Project Title:** CryptoVerse

**Team Leader :**

- **Srivalasalan M S** – 22g347srivalasalan@gmail.com

**Team Members:**

- **Reshma A** - 22g348reshma@gmail.com
- **Poovizhi** - 22g350boopalan@gmail.com
- **Boopalan** - Mohanpooandko468@gmail.com
- **Krishna Kumar** - rajsuriyakrishna@gmail.com

## Project Overview

### Purpose

CryptoVerse is a web application designed to provide users with real-time information on various cryptocurrencies, including their current prices, market trends, and detailed statistics. The platform aims to offer both novice and experienced investors a comprehensive tool to track and analyze digital currencies.

### Features

- Real-time cryptocurrency price tracking
- Detailed statistics and market trends
- Search functionality for specific cryptocurrencies
- Responsive design for optimal viewing on various devices
- Interactive charts for price visualization

## Architecture

### Component Structure

CryptoVerse\_NaanMudhalvan/

```
| — public/           # Public assets
| | — _redirects       # Redirect settings
| | — cryptocurrency.png # Cryptocurrency image
| | — index.html       # Main HTML file
```

```
|
|
| — src/                # Main source code
|   | — app/            # Global state management
|   |   | — store.js    # Redux or Context API store
|   |
|   | — components/     # React components
|   |   | — CryptoDetails.jsx
|   |   | — Cryptocurrencies.jsx
|   |   | — Exchanges.jsx
|   |   | — Homepage.jsx
|   |   | — LineChart.jsx
|   |   | — Loader.jsx
|   |   | — Navbar.jsx
|   |   | — News.jsx
|   |   | — index.js
|   |
|   | — images/         # Image assets
|   |   | — cryptocurrency.png
|   |
|   | — services/       # API services
|   |   | — cryptoApi.js # API calls for cryptocurrency data
|   |   | — cryptoNewsApi.js # API calls for cryptocurrency news
|   |
|   | — App.css         # Global styles
|   | — App.js          # Main App component
```

```
| |—— index.js      # Entry point
|
|—— .env.example    # Example environment variables
|—— .eslintrc.js    # ESLint configuration
|—— .gitignore      # Git ignored files
|—— README.md       # Documentation
|—— package-lock.json # Dependency lock file
|—— package.json     # Project dependencies.
```

## State Management

The application utilizes **Redux Toolkit** for efficient state management, allowing centralized and predictable state updates. It includes slices for:

- Cryptocurrencies
- News
- Price history

## Routing

**React Router** is employed to handle navigation between different pages, ensuring a seamless user experience.

## Setup Instructions

### Prerequisites

- Node.js (version 14.x or higher)
- npm package manager

### Installation

1. Clone the repository:
2. `git clone https://github.com/Srivalasalan/CryptoVerse_NaanMudhalvan.git`
3. Navigate to the project directory:
4. `cd CryptoVerse_NaanMudhalvan`
5. Install dependencies:
6. `npm install`

# Folder Structure

## Client

- **src/**: Contains the source code of the React application.
  - **components/**: Reusable UI components (e.g., Navbar, CryptoCard).
  - **pages/**: Page components corresponding to different routes (e.g., Homepage, Cryptocurrencies, CryptoDetails, News).
  - **app/**: Contains the main App component and routing logic.
  - **services/**: API service functions for fetching data from external sources.
  - **app/store.js**: Redux store configuration.

## Running the Application

To start the application locally:

```
npm start
```

This will launch the application at `http://localhost:3000`.

## Component Documentation

### Key Components

- **Navbar**: Renders navigation links to different sections of the application.
- **Homepage**: Displays top cryptocurrencies and latest market news.
- **Cryptocurrencies**: Lists all cryptocurrencies with pagination support.
- **CryptoDetails**: Provides detailed information about a selected cryptocurrency, including charts and statistics.
- **News**: Shows the latest news articles related to cryptocurrencies.
- **LineChart**: Visualizes cryptocurrency price trends.
- **Loader**: Displays a loading indicator when fetching data.

### Reusable Components

- **CryptoCard**: Displays brief information about a cryptocurrency, used in lists and search results.
- **Button**: Custom-styled button used across the application.
- **SearchBar**: Enables searching for cryptocurrencies.

## State Management

### Global State

Managed using **Redux Toolkit**, encompassing:

- **Cryptocurrencies**: State slice for storing cryptocurrency data.
- **News**: State slice for storing news articles.

## Local State

Handled within components using the `useState` hook for UI-specific states, such as form inputs and modal visibility.

## User Interface

Screenshots or GIFs showcasing different UI features, such as pages, forms, or interactions.

## Styling

### CSS Frameworks/Libraries

- The application uses **Ant Design** for consistent and responsive UI components.

### Theming

- Custom theming is applied using Ant Design's theming capabilities to align with the application's branding.

## Testing

### Testing Strategy

- The project employs **Jest** and **React Testing Library** for unit and integration testing of components and Redux slices.

### Code Coverage

- Code coverage is monitored using Jest's built-in coverage tools, aiming for comprehensive test coverage across all modules.

## Screenshots or Demo

### Live Demo

<https://drive.google.com/drive/folders/10whrma5VDJhR625YxMH3XTQFIuj7oYIN?usp=sharing>

### Screenshots

<https://drive.google.com/drive/folders/1Myg0eV7-RCUkCQWRvFNJLOZoVmJsRTVF?usp=sharing>

## Known Issues

- The Exchange and News options are not working properly.
- API Rate Limits: Some cryptocurrency APIs impose rate limits, which might cause occasional delays or failures in fetching data.
- Incomplete Data Display: Certain cryptocurrencies may not have all the required details, leading to missing information in the CryptoDetails page.
- Pagination Bug: In the Cryptocurrencies list, sometimes pagination does not update properly when switching between filters.
- Mobile Responsiveness: Some UI elements may not be fully optimized for smaller screens, causing layout inconsistencies.
- Loading Performance: The app might take longer to fetch and display data when dealing with large datasets or slower networks.

## **Future Enhancements**

- Dark mode toggle feature
- Real-time WebSocket price updates
- Improved UI animations
- Additional charting options
- User authentication for personalized watchlists