# 1.Introduction:

# • Project title:

# Cryptoverse : A Cryptocurrency Dashboard (Dashboard)

# • Team Members:

# S.Vinoth - Team Leader

# Email - vinoths637414@gmail.com

# A.Venkatesan - Speaker about the project

# Email - venkadesan2749524@gmail.com

# N.Vinithkumar – Document Creator

# Email - [vinith442005@yahoo.com](mailto:vinith442005@yahoo.com)

# Yuvansridhar – Taking notes of the project

# Email - yuvansridhar795@gmail.com

# 2.Project Overview:

# • Purpose:

# The Cryptocurrency Dashboard is a web application that provides real-time data on various cryptocurrencies. It allows users to view prices, trends, and historical data through an interactive interface.

# • Features of Cryptoverse:

# 1. Live Cryptocurrency Price Tracking:

# \* Uses an API (such as CoinGecko or CoinMarketCap) to fetch live cryptocurrency prices.

# \*Data updates at regular intervals using useEffect and setInterval in React.

# \*Displayed in CryptoCard.js on the dashboard.

# 2. Historical Price Data with Interactive Charts

# \*Chart.js and react-chartjs-2 are used for dynamic line charts.

# \*Historical price data fetched via API and visualized in Chart.js component.

# \*Users can hover over charts to see price trends over time.

# 3. Customizable Timeframes for Analysis

# \*Users can select different timeframes (e.g., 1 day, 7 days, 1 month, 1 year, 5 years).

# \*Implemented using a Select dropdown in CryptoDetails.js.

# \*API requests update based on the selected timeframe.

# 4. Search and Filter Functionality

# \*A search bar in Cryptocurrencies.js lets users find specific coins.

# \*Uses useState and filter() to dynamically search cryptocurrencies.

# \*Filters by market cap, price, or percentage change.

# 3. Architecture

# • Component Structure:

# \*App.js – Root component

# \*Navbar.js – Navigation bar

# \*Dashboard.js – Main page displaying cryptocurrency data

# \*CryptoCard.js – Individual cryptocurrency information

# \*Chart.js – Graph visualization of price trends

# \*SearchBar.js – Search functionality

# \*ThemeToggle.js – Dark/light mode switch

# v.jpeg

# • Routing:

# 1. React Router-Based Navigation – The dashboard uses React Router DOM for seamless navigation between pages, such as the home page, cryptocurrency listings, and individual crypto details.

# 2. API Routing with Redux Toolkit – The app integrates Redux Toolkit's RTK Query to fetch cryptocurrency data through predefined API endpoints like getCryptos, getCryptoDetails, and getCryptoHistory.

# 3. State Management with Redux – The Redux store manages application state efficiently, ensuring smooth data flow and reducing unnecessary re-fetching of API data.

# 4. Data Visualization with Chart.js – Line charts powered by react-chartjs-2 allow users to view historical price trends, making data analysis more interactive and insightful.

# 5. Search & Filtering Functionality – The search feature helps users quickly find specific cryptocurrencies, enhancing the user experience and enabling better data exploration.

# •Routing Structure:

# / – Homepage with crypto statistics.

# /cryptocurrencies – List of all available cryptocurrencies.

# /crypto/:id – Detailed cryptocurrency price chart & data.

# 4.Setup Instructions

# •Prerequisites:

# Node.js (LTS version) - Download Here

# React.js

# Git for version control

# •Installation Steps:

# Clone the repository:

# git clone <https://github.com/SSC369/cryptoverse.git>

# Navigate to the project directory:

# cd cryptoverse

# 3. Install dependencies:

# npm install

# 4.Start the development server:

# npm run dev # (For Vite)

# npm start # (For Create-React-App)

# 5. Open http://localhost:3000 in your browser.

# 5.Folder Structure:

# /cryptoverse

# │── /src

# │ │── /components

# │ │ │── Navbar.js

# │ │ │── CryptoList.js

# │ │ │── CryptoDetails.js

# │ │ │── LineChart.js

# │ │── /pages

# │ │ │── Home.js

# │ │ │── CryptoPage.js

# │ │── /redux

# │ │ │── cryptoApi.js

# │ │ │── store.js

# │ │── /assets

# │── package.json

# │── README.md

# 6.Running the Application

# \*To start the project locally, use:

# npm run dev

# 7.Component Documentation

# •Key Components:

# \*CryptoList – Displays available cryptocurrencies.

# \*CryptoDetails – Shows historical data & price charts.

# \*LineChart – Graphical representation of price trends.

# \*SearchFeature – Allows filtering & searching.

# •Reusable Components:

# \*Card – Displays cryptocurrency stats.

# \*Button – Custom button components.

# 8.State Management

# •Global State Management:

# \*Uses Redux Toolkit for API integration.

# \*Implements cryptoApi to fetch data from cryptocurrency APIs.

# •Local State Management:

# \*React useState & useEffect for local component handling.

# \*Redux Middleware for API calls.

# 9.User Interface & Styling

# •UI Features:

# \*Dark mode support.

# \*Interactive UI with Ant Design.

# \*Smooth navigation & responsive layout.

# 10.Styling

# CSS Frameworks:

# Ant Design

# Styled Components / Tailwind CSS

# 11.Testing

# Testing Strategy:

# Jest & React Testing Library for unit tests.

# Cypress for end-to-end testing.

# Ensures API data rendering & UI consistency.

# Code Coverage:

# Tests for core functionality of API calls & UI rendering.

# 12.Screenshots or Demo

# Live Demo:

# Video link: <https://drive.google.com/file/d/1W8-DMSXNbdLMhfNVW1Yc2h-N0chDeNkc/view?usp=drivesdk>

# Gitub link: https://github.com/VenkatA-Cs/Cryptoversse--SWTID1741257816149389

# Thankyou!

# 

# 

# 

# 