# Introduction

* Project Title: Cryptocurrency Tracker ● Team Members:
* [Member Name :PRASANNA KUMAR SK] - Role Team lead
* [Member Name 2:BALAJI D] - Role Team member
* [Member Name 3:JESWIN PRAISON S] - Role Team member
* [Member Name 4:KAMALESHWARAN ] - Role Team member
* [Member Name 5:SYED NIZAMUDDIN ] - Role Team member

# Project Overview

* Purpose:
* The purpose of this project is to create a user-friendly application that allows users to track cryptocurrency prices and trends in real-time.
* Features:
* Real-time price updates
* Historical data visualization
* User authentication
* Portfolio management

# Architecture

* Component Structure:
* The application is structured into several major components including Home, CryptoDetails, Cryptocurrencies, and LineChart.
* State Management:
* The project uses Redux for global state management to handle the application state efficiently.
* Routing:
* React Router is used for navigation between different pages of the application.

# Setup Instructions

● Prerequisites:

1. Node.js
2. npm (Node Package Manager) ● Installation:
3. Clone the repository: git clone https://github.com/Bycks/cryptocurrency.git
4. Navigate to the project directory: cd cryptocurrency
5. Install dependencies: npm install
6. Configure environment variables as needed.

# Folder Structure

* Client:
* Contains the React application organized into folders such as components, pages, and assets.
* Utilities:
* Includes helper functions and custom hooks used throughout the project.

# Running the Application

* Commands:
* To start the frontend server locally, run: npm start in the client directory.

# Component Documentation

* Key Components:
* Home: Displays the main dashboard with cryptocurrency prices.
* CryptoDetails: Shows detailed information about a selected cryptocurrency.
* Cryptocurrencies: Lists all available cryptocurrencies.
* LineChart: Visualizes historical price data.
* Reusable Components:
* Loader: Displays a loading spinner while data is being fetched.
* Navbar: Navigation bar for the application.

# State Management

* Global State:
* Managed using Redux, allowing for a centralized store for all application state.
* Local State:
* Handled within individual components using React's useState hook.

# User Interface

* Screenshot:
* Include screenshots or GIFs showcasing different UI features, such as the homepage, cryptocurrency details page, and charts.

# Styling

* CSS Frameworks/Libraries:
* The project uses CSS modules for styling components.
* Theming:
* Custom theming is implemented to provide a consistent look and feel across the application.

# Testing

* Testing Strategy:
* The project employs unit testing using Jest and React Testing Library to ensure component functionality.
* Code Coverage:
* Code coverage is monitored using Jest's built-in coverage tool.

# Screenshots or Demo

* Demo Link:
* Provide a link to a live demo or screenshots showcasing the application’s features and design.

# Known Issues

* Documented Issues:
* List any known bugs or issues that users or developers should be aware of.

# Future Enhancements

* Potential Features:
* Integration of additional cryptocurrencies. ● Enhanced data visualization options. ● User notifications for price changes