**BUILD AN AI CHATBOT ON YOUR CUSTOM DATA TO GET REALTIME CRYPTO MARKET DATA**

**D. L. C. Lakshitha**

**BSc (Hons) Software Engineering (Reading)**

**Humen Resources Management and Administration (DIP)**

**Advance Software Engineering and AI (Reading)**

Table of Contents

[INTRODUCTION 3](#_Toc197587873)

[TECHNOLOGIES 5](#_Toc197587874)

[Core MERN stacks 5](#_Toc197587875)

[Styling and UI tools 5](#_Toc197587876)

[State management and data fetching 5](#_Toc197587877)

[Authentication 5](#_Toc197587878)

[Development Tools 6](#_Toc197587879)

[UI Components and Effects 6](#_Toc197587880)

[Backend Helpers 6](#_Toc197587881)

[FEATURES 7](#_Toc197587882)

[User-Side Features (Customer View) 7](#_Toc197587883)

[Admin-Side Features 7](#_Toc197587884)

[LOGIC (WHAT HAPPEN) 7](#_Toc197587885)

[Frontend (What the user sees) 7](#_Toc197587886)

[Backend (What happens behind the scenes) 7](#_Toc197587887)

[Communicate Each Stage 7](#_Toc197587888)

[links 8](#_Toc197587889)

[ Reference Video: (1.04.20) 8](#_Toc197587890)

[ YT Video Summarize: 8](#_Toc197587891)

[ GitHub Repo 8](#_Toc197587892)

[ Code GPT 8](#_Toc197587893)

# INTRODUCTION

This project entails the development of an **AI-powered chatbot** to fetch **real-time cryptocurrency market data** using the **Gemini AI** and **CoinGecko API**. The chatbot is built with a focus on providing users with live data on cryptocurrencies such as **Bitcoin**, **Ethereum**, and more, using dynamic and responsive technologies.

The **frontend** of the chatbot is developed using **React.js** for interactive and responsive user experiences. On the **backend**, **Node.js** and **Express.js** provide the server-side logic for handling user requests and fetching cryptocurrency data from the APIs. **Gemini AI** is used for processing user queries, while **CoinGecko API** is employed to pull real-time data regarding various cryptocurrency metrics.

Key features include:

* **User-Side features** such as querying cryptocurrency prices, market cap, supply, and more, with a simple user interface.
* **Backend features** leveraging API integrations, where real-time data is fetched and processed in response to user queries.
* **Modern tools** including **Redux Toolkit** for state management, **Tailwind CSS** for responsive design, and **JWT** for secure user authentication.

The application aims to provide a seamless experience for users querying real-time data while keeping the interactions simple and engaging.

# TECHNOLOGIES

## Core MERN stacks

|  |  |  |
| --- | --- | --- |
| A green circle with a white leaf  AI-generated content may be incorrect. | MongoDB | NoSQL database for storing data like books, users, and orders. |
| A grey circle with white letters  AI-generated content may be incorrect. | **Express.js** | **Backend web application framework for Node.js used to build APIs.** |
| A blue atom symbol with a black background  AI-generated content may be incorrect. | **React.js** | **Frontend library for building user interfaces.** |
| A black circle with white text and green hexagon  AI-generated content may be incorrect. | **Node.js** | **JavaScript runtime for building server-side logic.** |

## Styling and UI tools

|  |  |  |
| --- | --- | --- |
| Tailwind Shades - Visual Studio Marketplace | Tailwind CSS | Installed and configured for custom styling. |

## State management and data fetching

|  |  |  |
| --- | --- | --- |
| A purple logo with dots  AI-generated content may be incorrect. | RTK & RTK Query | Mentioned as essential tools for managing and fetching app state, used throughout. |

## Authentication

|  |  |  |
| --- | --- | --- |
| A colorful arrows in a circle  AI-generated content may be incorrect. | JWT | Clearly implemented for both user login and admin dashboard access. |

## Development Tools

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

## UI Components and Effects

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

## Backend Helpers

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

# FEATURES

## User-Side Features (Customer View)

## Admin-Side Features

# LOGIC (WHAT HAPPEN)

## Frontend (What the user sees)

## Backend (What happens behind the scenes)

## Communicate Each Stage

# links

* Reference Video: (1.04.20)

## YT Video Summarize:

## GitHub Repo

## Code GPT