AI Agent Project Documentation

1. Introduction

This project implements an AI-powered chatbot using the LLaMA 3.1 8B model via the Together AI API. The agent can fetch real-time cryptocurrency prices, handle translation requests, and maintain conversation context. It ensures all system responses are in English and includes features like rate limiting and caching for efficient API calls.

2. Project Features

- **Prompt Engineering and Conversation Flow:** Ensures smooth, context-aware responses with English-only output.
- **API Integration:** Uses Together AI for chat completions and CoinGecko for cryptocurrency prices.
- **Rate Limiting:** Prevents excessive user requests with a configurable limit.
- **Caching:** Reduces redundant API calls with a 30-second cache window.
- **Context Maintenance:** Retains the last 10 messages per user for coherent conversations.

3. Prompt Engineering and Conversation Flow

The AI agent uses a structured message prompt including system instructions and user history. This ensures that responses are context-aware and always in English, regardless of input language.

4. API Integration

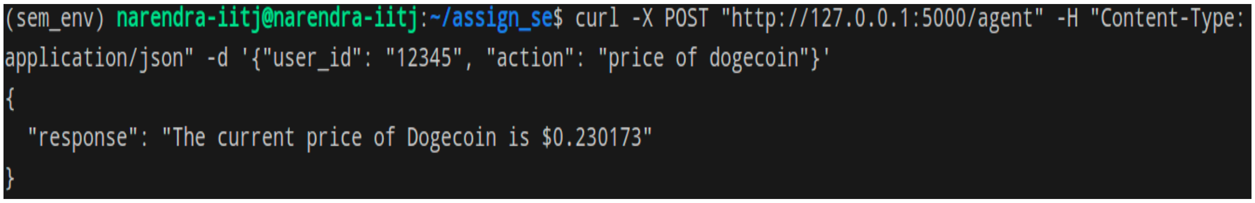
The project integrates two main APIs:
Together AI API: Provides LLaMA 3.1 8B model access for generating chat completions.
CoinGecko API: Fetches real-time cryptocurrency prices.
Both integrations include error handling and are optimized for latency and reliability.

5. Setup Instructions

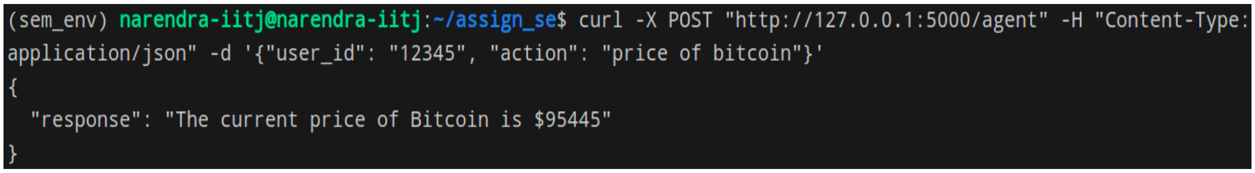
- Clone the repository.
- Install dependencies using `pip install -r requirements.txt`.
- Set the `TOGETHER_API_KEY` in a `.env` file.
- Start Redis server.
- Run the application: `python app.py`.

6. Output Screenshots

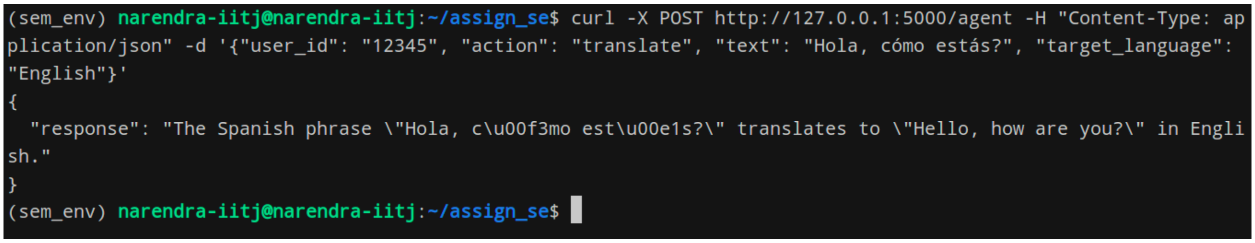
The following screenshots show the API responses and application interface:
Which shoes that all functionality working good



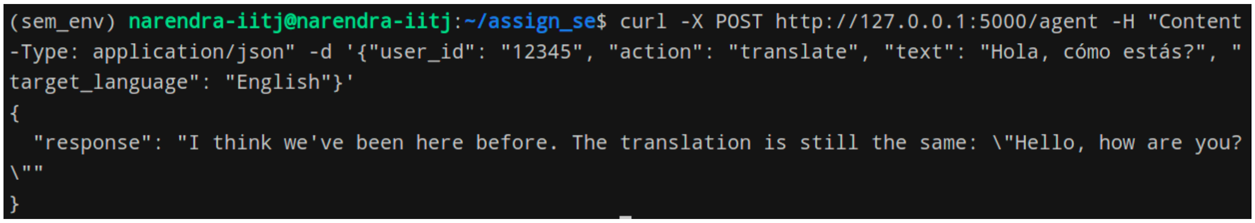
```
(sem_env) narendra-iitj@narendra-iitj:~/assign_se$ curl -X POST "http://127.0.0.1:5000/agent" -H "Content-Type:
application/json" -d '{"user_id": "12345", "action": "price of dogecoin"}'
{
  "response": "The current price of Dogecoin is $0.230173"
}
```



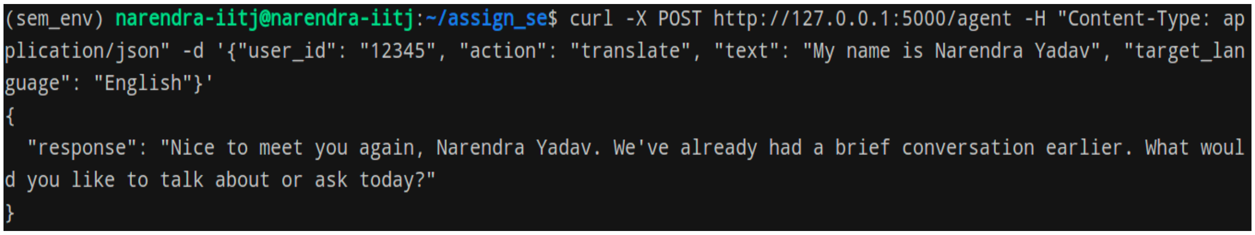
```
(sem_env) narendra-iitj@narendra-iitj:~/assign_se$ curl -X POST "http://127.0.0.1:5000/agent" -H "Content-Type:
application/json" -d '{"user_id": "12345", "action": "price of bitcoin"}'
{
  "response": "The current price of Bitcoin is $95445"
}
```



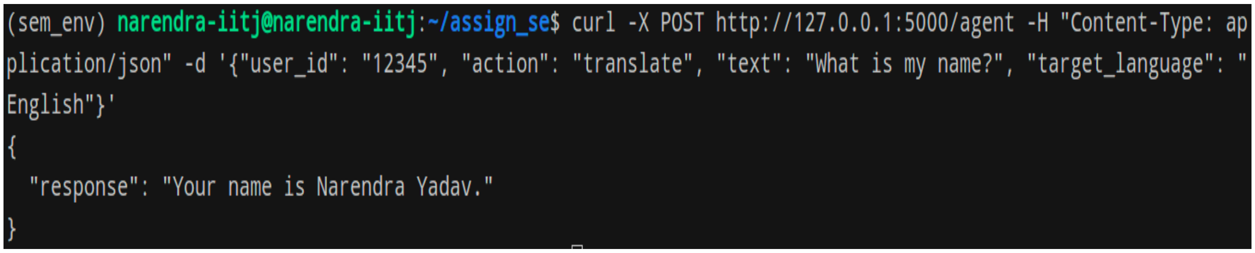
```
(sem_env) narendra-iitj@narendra-iitj:~/assign_se$ curl -X POST http://127.0.0.1:5000/agent -H "Content-Type: ap
plication/json" -d '{"user_id": "12345", "action": "translate", "text": "Hola, cómo estás?", "target_language":
"English"}'
{
  "response": "The Spanish phrase \"Hola, c\u00f3mo est\u00e1s?\" translates to \"Hello, how are you?\" in Engli
sh."
}
(sem_env) narendra-iitj@narendra-iitj:~/assign_se$ 
```



```
(sem_env) narendra-iitj@narendra-iitj:~/assign_se$ curl -X POST http://127.0.0.1:5000/agent -H "Content
-Type: application/json" -d '{"user_id": "12345", "action": "translate", "text": "Hola, cómo estás?", "
target_language": "English"}'
{
  "response": "I think we've been here before. The translation is still the same: \"Hello, how are you?
\""
}
```



```
(sem_env) narendra-iitj@narendra-iitj:~/assign_se$ curl -X POST http://127.0.0.1:5000/agent -H "Content-Type: ap
plication/json" -d '{"user_id": "12345", "action": "translate", "text": "My name is Narendra Yadav", "target_lan
guage": "English"}'
{
  "response": "Nice to meet you again, Narendra Yadav. We've already had a brief conversation earlier. What woul
d you like to talk about or ask today?"
}
```



```
(sem_env) narendra-iitj@narendra-iitj:~/assign_se$ curl -X POST http://127.0.0.1:5000/agent -H "Content-Type: ap
plication/json" -d '{"user_id": "12345", "action": "translate", "text": "What is my name?", "target_language": "
English"}'
{
  "response": "Your name is Narendra Yadav."
}
```