**Project: Conversational AI Chatbot**

**Objective:** Create a prompt framework for developing a conversational AI chatbot that can engage users on various topics, provide information, and offer recommendations. This project focuses on crafting conversational prompts that maintain context and coherence.

**Expected Outcome:** A set of prompts that enable the chatbot to have meaningful conversations, demonstrating students’ ability to guide AI in conversational contexts.

**FrontEnd Code :**

import { useState } from 'react';

import axios from 'axios';

function Chatbot() {

const [messages, setMessages] = useState([]);

const [input, setInput] = useState('');

const sendMessage = async () => {

if (!input.trim()) return;

const newMessages = [...messages, { role: 'user', text: input }];

setMessages(newMessages);

setInput('');

try {

const res = await axios.post('http://localhost:5000/api/chat', {

message: input,

history: newMessages.map(m => ({ role: m.role, text: m.text }))

});

const botReply = res.data.reply;

console.log("🤖 Bot says:", botReply);

setMessages([...newMessages, { role: 'bot', text: botReply }]);

} catch (err) {

console.error("❌ Error from backend:", err);

setMessages([...newMessages, { role: 'bot', text: 'Something went wrong!' }]);

}

};

return (

<div style={{ padding: 20 }}>

<h2>💬 AI Chatbot</h2>

<div style={{ marginBottom: 10 }}>

{messages.map((msg, i) => (

<p key={i} style={{ textAlign: msg.role === 'user' ? 'right' : 'left' }}>

<b>{msg.role === 'user' ? 'You' : 'Bot'}:</b> {msg.text}

</p>

))}

</div>

<div style = {display :flex}>

<input

value={input} placeholder= “Type your message here”

onChange={(e) => setInput(e.target.value)}

onKeyDown={(e) => e.key === 'Enter' && sendMessage()}

style={{ width: '70%', padding: 8 }}

/>

<button onClick={sendMessage} style={{ padding: '8px 16px', marginLeft: 8 color : blue}}>

Send

</button>

</div>

</div>

);

}

export default Chatbot;

**BackEnd Code ;**

from flask import Flask, request, jsonify

from flask\_cors import CORS

from chatbot\_model import generate\_reply

from db import conversations

app = Flask(\_\_name\_\_)

CORS(app)

@app.route('/api/chat', methods=['POST'])

def chat():

user\_msg = request.json.get("message")

history = request.json.get("history", [])

if not user\_msg:

return jsonify({"reply": "No message received."}), 400

try:

reply = generate\_reply(user\_msg, history)

conversations.insert\_one({"user": user\_msg, "bot": reply})

return jsonify({"reply": reply})

except Exception as e:

print("❌ BACKEND ERROR:", str(e))

return jsonify({"reply": "Something went wrong!"}), 500

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

**Chatbot\_model.py :**

import requests

def generate\_reply(user\_msg, history=[]):

full\_prompt = ""

for msg in history:

full\_prompt += f"{msg['role'].capitalize()}: {msg['text']}\n"

full\_prompt += f"User: {user\_msg}\nBot:"

res = requests.post(

"http://localhost:11434/api/generate",

json={"model": "gemma:2b", "prompt": full\_prompt, "stream": False}

)

return res.json()["response"]

**db.py:**

from pymongo import MongoClient

client = MongoClient("mongodb://localhost:27017/")

db = client.chatbotDB

conversations = db.conversations

**Output:**

