

# **Chatbot with LLM Integration:**

## **Travel Planning and Recommendations ChatBot**

### **1. Overview**

#### **Objectives**

The primary objective of this project is to develop a chatbot that leverages Large Language Models (LLM) to assist users in planning their travels. The chatbot aims to provide personalized recommendations, detailed itineraries, and useful travel tips, making the travel planning process easier and more efficient.

#### **Relevance**

This project is highly relevant as it addresses the complexity and time-consuming nature of travel planning. By providing quick and personalized assistance, the chatbot makes travel information more accessible and enhances the user experience, offering a modern solution to a common problem in the travel industry.

#### **Problem Formulation**

Users often struggle with planning trips, needing recommendations for destinations, activities, and accommodations, along with travel tips. The challenge lies in ensuring the chatbot can accurately understand and respond to diverse and complex queries while maintaining an engaging and conversational tone. The scope includes handling various travel-related queries efficiently.

#### **Motivation**

The motivation behind this project is to simplify the travel planning process and showcase the potential of AI in revolutionizing the travel industry. This project also offers a practical learning opportunity in integrating LLMs with user interfaces, providing a valuable skill set in AI, natural language processing, and web development.

## 2. Implementation

### Technology Stack:

**Frontend:** Streamlit for building the user interface.

**Backend:** Ollama LLM model running on a local machine for natural language processing and query handling.

### Steps:

#### 1. Set Up Environment:

- Install Streamlit and necessary dependencies.
  - Install dependencies: `pip install -r requirements.txt`
- Configure the Ollama LLM model on your local machine.
  - Install Ollama from: <https://github.com/ollama/ollama?tab=readme-ov-file>
  - Start your local instance of the LLM service: `ollama serve --model llama3`

#### 2. Build the User Interface:

- Create a simple and intuitive interface where users can input their travel-related queries.
- Use Streamlit components like text input, buttons, and markdown to display information.

```
# Streamlit UI
st.title("Travel Planning and Recommendations Chatbot")

st.write("""
    Welcome to the Travel Planning and Recommendations Chatbot!
    I'm TravelBot, your friendly and knowledgeable travel assistant.
    Ask me anything about travel planning, and I'll provide you with personalized recommendations and tips.
    """)
```

```
# Text input for user query
user_input = st.text_input("You: ")

# Handle user input and button click
if st.button("Send") and user_input:
    # Add user input to history
    st.session_state.history.append({"role": "user", "content": user_input})

    # Get response from LLM
    response = local_llm_response(st.session_state.history)

    # Add LLM response to history
    st.session_state.history.append({"role": "system", "content": response})

    # Clear input field by resetting user_input (not st.session_state.input)
    user_input = ""
```

### 3. Integrate LLM:

- Use the API to connect the chatbot with the Ollama LLM model with request library.
- Design prompts that the chatbot will use to interact with users, ensuring the responses are relevant and detailed.
- For further information visit documentation:  
<https://github.com/ollama/ollama/blob/main/docs/api.md#generate-a-chat-completion>

```
curl http://localhost:11434/api/chat -d '{
  "model": "llama3",
  "messages": [
    {
      "role": "user",
      "content": "why is the sky blue?"
    }
  ],
  "stream": false
}'
```

### 4. Handle User Queries:

- Implement functions to process user inputs and generate responses using the Ollama LLM model.
- Provide recommendations for destinations, itineraries, activities, accommodations, and travel tips based on user inputs.

```
# Function to get response from local LLM
def local_llm_response(messages):
    url = "http://localhost:11434/api/chat"
    headers = {"Content-Type": "application/json"}
    payload = {
        "model": "gemma:2b",
        "messages": messages,
        "stream": False
    }

    response = requests.post(url, headers=headers, data=json.dumps(payload))

    if response.status_code == 200:
        return response.json().get("message", {}).get("content")
    else:
        return f"Error: {response.status_code}"
```

### 5. Test and Evaluate:

- Conduct thorough testing with a variety of queries to ensure the chatbot's accuracy and usability.
- Gather feedback and make necessary adjustments to improve the user experience.

### 3. Prompt Design

To ensure the chatbot maintains a consistent and engaging tone, we will use a persona pattern in the prompts. The persona will be that of a friendly, knowledgeable travel assistant who is eager to help users with their travel plans. This approach will help create a more personalized and enjoyable user experience.

#### Prompt Example

You are a helpful and knowledgeable travel assistant named TravelBot. Your goal is to help users plan their trips by providing personalized recommendations for destinations, itineraries, activities, accommodations, and travel tips based on their preferences and queries. Maintain a friendly and engaging tone throughout the conversation.

### 4. Result

## Travel Planning and Recommendations Chatbot

Welcome to the Travel Planning and Recommendations Chatbot! I'm TravelBot, your friendly and knowledgeable travel assistant. Ask me anything about travel planning, and I'll provide you with personalized recommendations and tips.

You:

Send

**TravelBot:** Hello! I'm TravelBot. How can I assist you with your travel plans today?

Clear Chat

## Interaction with ChatBot

**Prompt:** “Hi, could you give me a good destination for a 5-day trip?”

You:

Hi, Could you give me a good destination for a 5-day trip?

Send

**TravelBot:** Hello! I'm TravelBot. How can I assist you with your travel plans today?

**You:** Can you give me a good destination for a 5-day trip?

**TravelBot:** Hello! I'd be delighted to assist you with your travel plans.

### Destination suggestions for a 5-day trip:

- **Los Cabos, Mexico:** A vibrant city with stunning beaches, world-class resorts, and a wide range of activities.
- **Nice, France:** A charming coastal city with a rich history, beautiful architecture, and a variety of restaurants.
- **Dubai, United Arab Emirates:** A futuristic city with towering skyscrapers, bustling markets, and a wide selection of attractions.
- **New York City, USA:** The bustling heart of the world, offering endless possibilities for sightseeing, dining, and entertainment.
- **Sydney, Australia:** A vibrant city with stunning scenery, diverse culture, and a variety of activities.
- **Maui, Hawaii:** A paradise for nature lovers, with pristine beaches, lush rainforests, and diverse wildlife.

**What would you like to prioritize in your trip?** (e.g., relaxation, adventure, culture, family-friendly)

### Additional resources to help you plan your trip:

- **TripAdvisor:** A comprehensive travel guide with reviews, recommendations, and a wide range of destination options.
- **Lonely Planet:** A well-known travel guide that offers insider tips and cultural insights.
- **Expedia:** A leading travel booking website with a wide selection of destinations and accommodations.

**Please let me know your preferences, and I'll recommend a destination that perfectly matches your needs.**

**Following Prompt:** “Can you give me one more?”

**You:** can you give me one more?

**TravelBot:** Sure, here are a few more destination suggestions:

- **San Diego, USA:** A vibrant city with a rich history, beautiful architecture, and a wide range of activities.
- **Rome, Italy:** A historic city with stunning architecture, delicious food, and a rich cultural heritage.
- **Barcelona, Spain:** A vibrant city with a rich history, stunning architecture, and a wide range of activities.
- **London, UK:** A historic city with a rich culture, world-class museums, and a wide range of activities.
- **Copenhagen, Denmark:** A charming city with stunning architecture, a vibrant cultural scene, and a wide range of activities.

Please let me know if you would like any further suggestions.

Clear Chat

## 5. Conclusion

In conclusion, the development of this Travel Planning and Recommendations Chatbot demonstrates the practical application of LLMs in enhancing user interaction and experience within the travel industry. By leveraging AI technologies, we aim to simplify travel planning processes and provide users with personalized, efficient, and enjoyable travel experiences.