AI Chatbot for Car Rental Application

Project Overview

This project focuses on developing an AI-powered chatbot for a car rental application. The chatbot is designed to assist users with queries related to car availability, booking, pricing, and general customer support. The chatbot aims to enhance the user experience by providing seamless and efficient interaction.

Objectives

1. **User Convenience:** Provide a quick and intuitive way for users to interact with the car rental service.
2. **Automated Assistance:** Minimize the need for manual intervention by automating responses to frequently asked questions.
3. **Scalability:** Create a modular and extensible chatbot to support future enhancements.

Features

- **Car Availability Inquiry:** Check the availability of specific car models.
- **Booking Assistance:** Facilitate car booking based on user-specified dates and locations.
- **Pricing Information:** Provide pricing details, including daily rates and total costs for specified durations.
- **General Customer Support:** Handle queries about rental policies, cancellation, and more.

Technologies Used

- Programming Language: Python
- Libraries:
  - spaCy for Natural Language Processing (NLP)
  - NLTK for text preprocessing
- **Framework:** Flask (for web application deployment)
- **Database:** SQL (for storing booking information)
- **Hosting Platform:** Xampp (for deployment)

Development Process

Phase 1: Requirement Analysis

- Identify key user intents:
  - Car availability
  - Booking
  - Pricing
  - General support
- Gather sample user queries to design conversational flows.

Phase 2: Setting Up the Environment

1. Install Python and necessary libraries:

   PIP INSTALL FLASK SPACY NLTK

2. Set up a Git repository for version control.

Phase 3: Designing the Chatbot Logic

- Create a dictionary of available cars with details like availability and pricing.
- Implement functions for key tasks:
  - Car Availability Check: Verify if a specific car is available.
  - Booking: Calculate the total price and confirm the booking.

Example Code Snippet:

```
cars = {
   "Sedan": {"price_per_day": 50, "available": True},
   "SUV": {"price_per_day": 80, "available": True},
   "Hatchback": {"price_per_day": 40, "available": False},
}

def check_availability(car_type):
   car = cars.get(car_type)
   if car:
      if car["available"]:
         return f"{car_type} is available at ${car['price_per_day']} per day."
      else:
         return f"Sorry, {car_type} is currently unavailable."
   return f"We don't have {car_type} in our fleet."
```

Phase 4: Adding NLP Capabilities

- Use spaCy for intent recognition and entity extraction (e.g., car type, dates).
- Train custom NLP models if required.

Phase 5: Creating the Web Application

- Use Flask to build a simple API for chatbot interaction.

Example Flask App Code:

```
from flask import Flask, request, jsonify

app = Flask(__name__)
```

```
@app.route("/chat", methods=["POST"])
def chat():
    user_input = request.json.get("message")
    response = chatbot_response(user_input)
    return jsonify({"response": response})

if __name__ == "__main__":
    app.run(debug=True)
```

Phase 6: Database Integration

- Use SQL to store booking details and user interactions.
- Example Schema:

```
CREATE TABLE bookings (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    car_type TEXT,
    start_date TEXT,
    end_date TEXT,
    total_price REAL
);
```

Usage Guide

1. Interact with the chatbot via the web interface or API.
2. Example Queries:
    - "Is a sedan available?"
    - "Book an SUV from 2025-01-15 to 2025-01-20."

Future Enhancements

1. Voice Integration: Add support for speech-to-text and text-to-speech.
2. Multilingual Support: Extend the chatbot to handle multiple languages.
3. Advanced Analytics: Track user interactions for insights and improvements.

Conclusion

The AI chatbot for the car rental application simplifies user interactions and automates key tasks, reducing the need for manual support. With its modular design and scalability, it is well-suited for future enhancements and broader deployment.