Basic AI Chatbot Prototype for E-commerce

Objective :

The objective of this project is to develop a basic AI chatbot prototype that can assist users with common queries related to e-commerce. The chatbot handles basic greetings, farewells, and answers at least five frequently asked questions (FAQs) related to e-commerce. It also provides fallback responses for unrecognized queries.

Approach :

Frontend Development :

Created a simple and user-friendly chat interface using HTML and CSS.
Integrated JavaScript to handle user inputs and send them to the backend.

Backend Development :

Developed a Flask application to serve the frontend and handle user inputs.
Utilized spaCy for basic Natural Language Processing (NLP) to understand user queries.
Employed FuzzyWuzzy to match user queries with predefined FAQs stored in a JSON file.

NLP and FAQ Matching :

Loaded FAQs from a JSON file containing questions and corresponding answers.
Used spaCy to preprocess user inputs.
Implemented FuzzyWuzzy's token sort ratio to find the best matching FAQ based on string similarity.
Set a similarity threshold to determine if the chatbot can answer the query or provide a fallback response.

Technologies Used :

HTML/CSS : For creating the chat interface.
JavaScript : For handling user interactions and sending data to the backend.
Flask : As the web framework for handling HTTP requests and serving the application.

spaCy : For basic NLP tasks like tokenization.
FuzzyWuzzy : For string similarity matching to find the best matching FAQ.

Challenges :

String Similarity Threshold : Determining an appropriate threshold for string similarity to balance between accurate responses and fallback responses.
NLP Processing : Handling variations in user inputs effectively to ensure accurate FAQ matching.
User Interface : Designing a responsive and intuitive chat interface that enhances user experience.

Code Documentation

Flask application (app.py) :

```
from flask import Flask, request, jsonify, render_template
import spacy
import json
from fuzzywuzzy import fuzz

# Initialize the Flask application
app = Flask(__name__)

# Load spaCy's English model for NLP tasks
nlp = spacy.load("en_core_web_sm")

# Load FAQs data from a JSON file
with open('Ecommerce_FAQ_Chatbot_dataset.json') as f:
    faq_data = json.load(f)["questions"]

def get_response(user_input):
    """
    Process the user input and return an appropriate response.
    """
    doc = nlp(user_input)
    # Check for basic greetings and farewells
    for token in doc:
        if token.text.lower() in ["hello", "hi", "hey"]:
            return "Hello! How can I help you today?"
        if token.text.lower() in ["goodbye", "bye"]:
            return "Goodbye! Have a great day!"

    # Initialize variables for finding the best matching FAQ
    max_similarity = 0
    best_match = None

    # Loop through each FAQ entry to find the best match
    for entry in faq_data:
        question = entry["question"]
        similarity = fuzz.token_sort_ratio(user_input.lower(), question.lower())
        if similarity > max_similarity:
            max_similarity = similarity
            best_match = entry

    # Check if the best match meets the similarity threshold
    if max_similarity > 50:  # Adjust the threshold as needed
        response = best_match["answer"]
    else:
        response = "I'm sorry, I don't understand that question. Can you please rephrase?"

    return response + " Need any help?"
@app.route('/')
def index():
    """
    Render the main chat interface.
    """
    return render_template('index.html')

@app.route('/chatbot', methods=['POST'])
def chatbot():
    """
    Handle user messages and return chatbot responses.
    """
    user_input = request.json.get("message")
    response = get_response(user_input)
    return jsonify({"response": response})

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

HTML and JavaScript Code :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>E-commerce Chatbot</title>
    <style>
        body {
```

```
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;
}
.chat-container {
    background-color: white;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0,0,0,0.1);
    width: 400px;
    max-width: 100%;
    padding: 20px;
}
.chat-box {
    border: 1px solid #ddd;
    border-radius: 5px;
    padding: 10px;
    height: 300px;
    overflow-y: auto;
    margin-bottom: 10px;
}
.message {
    margin: 10px 0;
    padding: 10px;
    border-radius: 5px;
}
.bot-message {
    background-color: #f1f1f1;
}
.user-message {
    background-color: #007bff;
    color: white;
    text-align: right;
}
input[type="text"] {
    width: calc(100% - 80px);
    padding: 10px;
```

```
            border: 1px solid #ddd;
            border-radius: 5px;
        }
        button {
            width: 60px;
            padding: 10px;
            border: none;
            background-color: #007bff;
            color: white;
            border-radius: 5px;
            cursor: pointer;
        }
    </style>
</head>
<body>
    <div class="chat-container">
        <div class="chat-box" id="chat-box">
            <div class="message bot-message">Hello! How can I help you today?</div>
        </div>
        <input type="text" id="user-input" placeholder="Type your message here...">
        <button id="send-button">Send</button>
    </div>
    <script>
        document.getElementById('send-button').addEventListener('click', () => {
            sendMessage();
        });

        document.getElementById('user-input').addEventListener('keypress', (event) => {
            if (event.key === 'Enter') {
                sendMessage();
            }
        });

        function sendMessage() {
            const userInput = document.getElementById('user-input').value;
            if (userInput) {
                addMessage(userInput, 'user-message');
                fetch('/chatbot', {
                    method: 'POST',
                    headers: {
```

```
                    'Content-Type': 'application/json'
                },
                body: JSON.stringify({ message: userInput })
            })
            .then(response => response.json())
            .then(data => {
                addMessage(data.response, 'bot-message');
            })
            .catch(error => console.error('Error:', error));
            document.getElementById('user-input').value = '';
        }
    }

    function addMessage(message, className) {
        const chatBox = document.getElementById('chat-box');
        const messageElement = document.createElement('div');
        messageElement.className = `message ${className}`;
        messageElement.textContent = message;
        chatBox.appendChild(messageElement);
        chatBox.scrollTop = chatBox.scrollHeight;
    }
  </script>
</body>
</html>
```

Instructions to Run the Chatbot

1) Install the required libraries :

Bash :

```
pip install flask spacy fuzzywuzzy
python -m spacy download en_core_web_sm
```

1. **Load the `Ecommerce_FAQ_Chatbot_dataset.json` file with the provided structure to the project folder**
2. **Save the HTML code in a file named `templates/index.html`.**
3. **Save the Python script in a file named `app.py`.**

2) Run the Flask app:

Bash :

```
python app.py
```

4. **Open a web browser and go to `http://127.0.0.1:5000/` to interact with the chatbot.**

Conclusion

This project demonstrates the integration of basic AI concepts into a SaaS model through the development of an e-commerce chatbot. The chatbot effectively handles greetings, farewells, and FAQs, providing a solid foundation for more advanced AI applications.