Project Documentation: Wine Chatbot

1. Overall Approach

In this section, describe the overall approach you used to build the chatbot:

- **Objective**: Explain the purpose of the chatbot, which is to answer customer queries related to wines and services provided.
- **Design**: Outline how you designed the chatbot to handle various user inputs, integrate with the OpenAI API for responses, and provide relevant answers based on a predefined corpus of FAQs.
- **Implementation**: Detail the steps taken to implement the chatbot, including setting up the development environment, coding, and testing.

2. Frameworks/Libraries/Tools Used

List and describe the frameworks, libraries, and tools you used for the project:

- **Flask**:
  - **Purpose**: Used as the web framework to create the server and handle HTTP requests.
  - **Where Used**: The Flask application manages routes for interacting with the chatbot and serves the frontend.
- **OpenAI API**:
  - **Purpose**: Provides the natural language processing capabilities to generate responses based on user input.
  - **Where Used**: Integrated into the Flask app to process user queries and return appropriate responses.
- **HTML/CSS/JavaScript**:
  - **Purpose**: Used for building the user interface of the chatbot.
  - **Where Used**: HTML is used to structure the webpage, CSS for styling, and JavaScript for handling form interactions and AJAX requests.
- **Python**:
  - **Purpose**: Programming language used to write the backend logic of the chatbot.
  - **Where Used**: Python is used in the Flask app for handling requests and interacting with the OpenAI API.
- **GitHub**:
  - **Purpose**: Used for version control and collaboration.
  - **Where Used**: The project code is hosted on GitHub, allowing for version tracking and sharing with others.

3. Problems Faced and Solutions

Describe the challenges you encountered and how you resolved them:

- **Problem 1: API Version Compatibility**:
  - **Issue**: The OpenAI API had deprecated methods that caused issues with the initial implementation.
  - **Solution**: Updated the API integration to use the latest version and modified the code to align with the new API methods.
- **Problem 2: Large Cookie Size Warning**:
  - **Issue**: The application faced warnings about the cookie size being too large.
  - **Solution**: Reduced the amount of data stored in cookies and used session storage or a database to manage state.
- **Problem 3: Handling Corpus-Based Responses**:
  - **Issue**: The chatbot was not consistently answering questions from the corpus.
  - **Solution**: Enhanced the logic for checking if a user's question matches the corpus and improved the response handling mechanism.

4. Future Scope

Discuss potential improvements and future features for the chatbot:

- **Enhanced Natural Language Understanding**:
  - **Feature**: Incorporate advanced NLP techniques to better understand and respond to more complex or varied user queries.
- **Integration with External Data Sources**:
  - **Feature**: Connect the chatbot with databases or APIs to provide real-time information about inventory, pricing, or events.
- **Multi-Language Support**:
  - **Feature**: Add support for multiple languages to cater to a broader audience.
- **Personalization**:
  - **Feature**: Implement user profiles and personalized responses based on user preferences or history.
- **Voice Interaction**:
  - **Feature**: Integrate speech-to-text and text-to-speech functionalities to allow voice interactions with the chatbot.
- **Improved UI/UX**:
  - **Feature**: Enhance the user interface and experience by adding features like interactive elements, animations, or a more intuitive layout.

Appendix

Include any additional information such as:

- **Code Snippets**: Relevant parts of the code.

```python
import openai
import os
import string
from flask import Flask, request, jsonify, render_template

app = Flask(__name__)

openai.api_key = os.getenv("OPENAI_API_KEY")

# Function to normalize text by removing punctuation and extra spaces
def normalize_text(text):
    text = text.translate(str.maketrans('', '', string.punctuation))
    text = ' '.join(text.split())
    return text.lower()

# Expanded corpus with more questions and answers
corpus = {
    normalize_text("What is your best red wine?"): "Our best red wine is the 2018 Merlot. It's a full-bodied wine with notes of blackberry anc
    normalize_text("Tell me more about the 2018 Merlot."): "The 2018 Merlot is aged for 18 months in French oak barrels, giving it a smooth fi
    normalize_text("What are your hours of operation?"): "We are open from 10 AM to 6 PM, Monday through Saturday.",
    normalize_text("Do you offer wine tasting events?"): "Yes, we offer wine tasting events every Friday and Saturday from 2 PM to 5 PM. You c
    normalize_text("How can I book a wine tasting event?"): "You can book a wine tasting event by visiting our website and filling out the boc
    normalize_text("What is the price range of your wines?"): "Our wines range from $20 to $150 per bottle, depending on the variety and vinta
    normalize_text("Do you offer any discounts?"): "Yes, we offer a 10% discount on purchases of 12 bottles or more. We also have seasonal prc
    normalize_text("How can I join the wine club?"): "You can join our wine club by signing up on our website. Membership includes exclusive c
    normalize_text("Can I order wine online?"): "Yes, you can order wine directly from our website. We offer shipping to most states in the US
    normalize_text("What are your shipping policies?"): "We offer standard and expedited shipping options. Standard shipping usually takes 5-7
    normalize_text("Do you ship internationally?"): "Currently, we only ship within the United States. We hope to offer international shipping
    normalize_text("What is your return policy?"): "If you are not satisfied with your purchase, you can return the unopened bottles within 30
    normalize_text("Do you offer gift cards?"): "Yes, we offer gift cards in various denominations. They can be purchased on our website and a
    normalize_text("What are your most popular wines?"): "Our most popular wines include the 2018 Merlot, the 2020 Chardonnay, and the 2019 Ca
    normalize_text("Tell me more about the 2020 Chardonnay."): "The 2020 Chardonnay is a crisp and refreshing wine with notes of green apple a
    normalize_text("Tell me more about the 2019 Cabernet Sauvignon."): "The 2019 Cabernet Sauvignon is a robust wine with flavors of dark cher
    normalize_text("Do you offer tours of your vineyard?"): "Yes, we offer guided tours of our vineyard. Tours are available by appointment ar
    normalize_text("How can I book a vineyard tour?"): "You can book a vineyard tour by visiting our website and filling out the tour booking
    normalize_text("Where are you located?"): "We are located at 123 Vine Street, Wine Country, CA 98765.",
    normalize_text("Do you have a physical store?"): "Yes, we have a tasting room and store at our vineyard where you can sample and purchase
```



```python
    normalize_text("What are the benefits of joining your wine club?"): "Members of our wine club receive exclusive discounts, early access to
    normalize_text("Can I customize my wine club shipments?"): "Yes, you can customize your wine club shipments by selecting your preferred wi
    normalize_text("Do you have a loyalty program?"): "Yes, we have a loyalty program where you earn points for every purchase. Points can be
    normalize_text("What types of wines do you produce?"): "We produce a variety of wines including red, white, rosé, and sparkling wines. Our
    normalize_text("Do you have any vegan wines?"): "Yes, we offer a selection of vegan wines that are made without any animal-derived fining
}

def generate_response(user_message, context):
    print(f"User message: {user_message}")
    normalized_message = normalize_text(user_message)
    print(f"Normalized message: {normalized_message}")

    # Find the best matching response
    response = "Please contact the business directly for more information."
    for key in corpus:
        if key in normalized_message:
            response = corpus[key]
            break

    print(f"Response: {response}")
    return response

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/chat', methods=['POST'])
def chat():
    user_message = request.json['message']
    context = request.json.get('context', [])
    bot_response = generate_response(user_message, context)
    return jsonify({"response": bot_response})

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


- **References**: Links to documentation or resources used.
- **Screenshots**: Images of the application in use.

Wine Chatbot

You: Do you have any organic wines?
Bot: Yes, we offer a selection of organic wines that are made from grapes grown without synthetic pesticides or fertilizers. Look for the organic label on our website or in our store.
You: Can I customize my wine club shipments?
Bot: Yes, you can customize your wine club shipments by selecting your preferred wines from our current offerings. You can also choose the frequency of your shipments.

Type your message here | Send