
CAPSTONE PROJECT

CAFE CONVO - CHATBOT

Presented By:

1. Hamritha V-Shiv Nadar University Chennai-B.Tech CSE(IoT)

OUTLINE

- **Problem Statement** (Should not include solution)
- **Proposed System/Solution**
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment**
- **Result**
- **Conclusion**
- **Future Scope**
- **References**

PROBLEM STATEMENT

In a regular cafe, the service can be very hefty and inconvenient since it is done manually, hence making customers wait for long and delay, resulting in dissatisfaction. There will be queues for customers waiting to place orders or ask about menus, and the employees inside the cafe will most probably not be able to handle all these things comprehensively at one time. This results in bad service for and coordination in the cafe, which leads to lost business opportunities. Besides, the cafe staff would have no access or time to take the customer through personalized recommendations included in menus. This amounts to a lot of missed upselling and cross-selling opportunities.

PROPOSED SOLUTION

The proposed system aims to enhance customer service and provide personalized menu recommendations for cafe customers using a rule-based approach. The chatbot will respond to queries based on a predefined dataset, offering relevant information and recommendations based on the data provided.

- **Data Collection:**
 - Gather menu information and store it in a database.
 - Collect customer feedback and store it in a database for future improvements.
- **Data Preprocessing:**
 - Clean and preprocess the collected data to handle missing values, outliers, and inconsistencies.
- **Response System:**
 - Implement a system where the chatbot matches customer queries to predefined responses in the dataset.
 - Use the dataset to provide tailored menu suggestions based on customer preferences and available menu items.
- **Deployment:**
 - Develop an intuitive chatbot interface for automated customer service and personalized menu recommendations.
 - Deploy the chatbot on a scalable and reliable platform, ensuring efficient response times and accessibility for users.
- **Evaluation:**
 - Measure the chatbot's effectiveness using metrics such as response accuracy, customer satisfaction, and the completeness of the dataset. Refine the chatbot based on feedback and performance metrics, updating the dataset as needed to enhance response quality.
- **Result:**
 - A Flask-based web application that provides automated customer service through a chatbot interface, integrating with a database to retrieve menu information and provide accurate responses to customer queries.

SYSTEM APPROACH

■ System requirements

- Operating System: Any platform that supports Python (Windows, macOS, Linux)
- Python Version: Python 3.x
- Web Framework: Flask
- Database: CSV files (for storing menu information and customer feedback)

■ Library required to build the model

- Flask: for building the web application
- Pandas (pd): for data manipulation and loading CSV files
- jsonify: for returning JSON responses from the Flask API

ALGORITHM & DEPLOYMENT

- **Algorithm Selection:**

- A simple dictionary-based lookup algorithm is chosen for this chatbot system. This algorithm is suitable for this problem because it allows for quick and efficient retrieval of responses based on user input.

- **Data Input:**

- The input features used by the algorithm are the user queries, which are matched against a predefined dictionary of responses

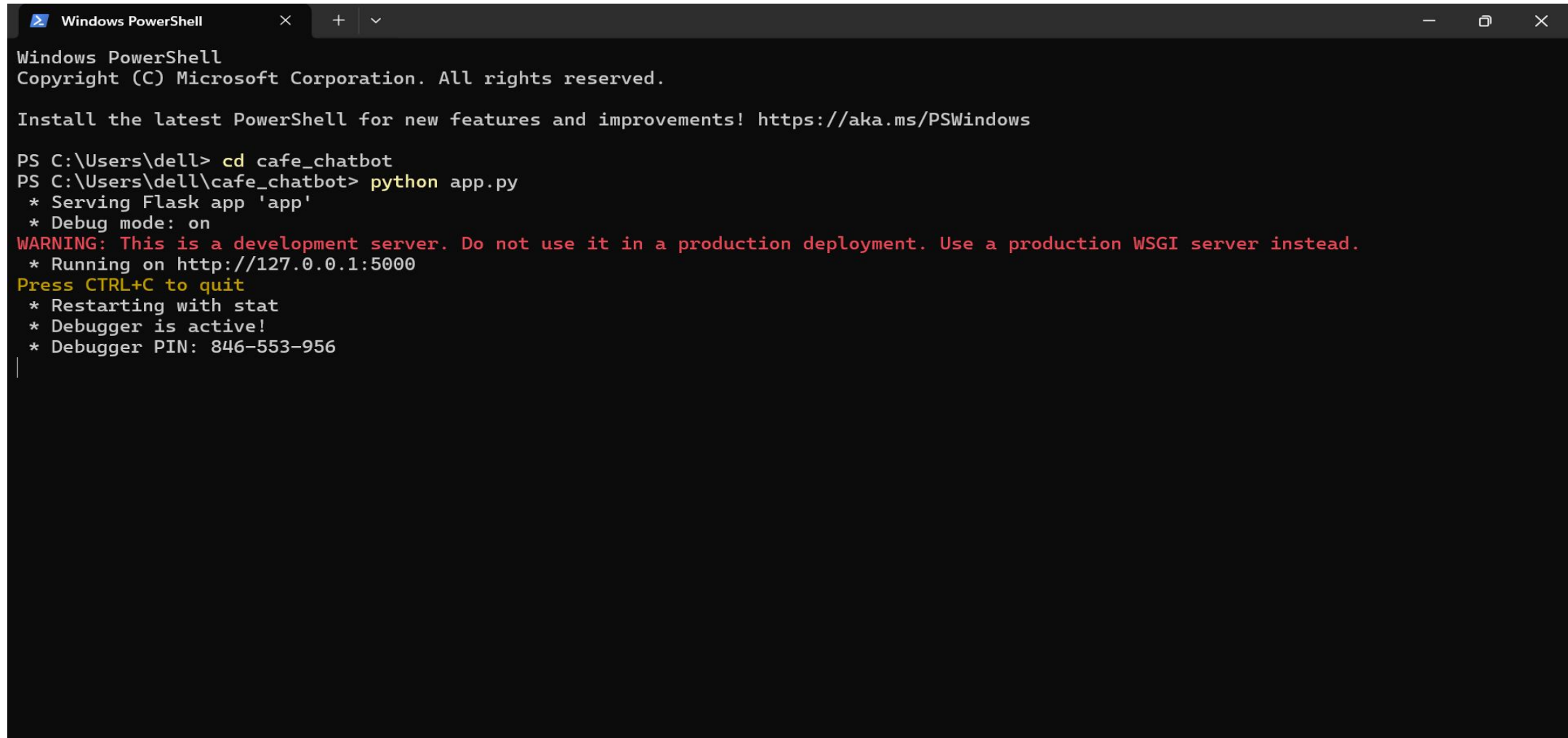
- **Prediction Process:**

- The algorithm makes predictions by matching user input against the dictionary of responses. If a match is found, the corresponding response is returned to the user. If no match is found, a default response is returned.

- **Deployment Strategy:**

- The chatbot system is deployed as a Flask-based web application, which provides a scalable and reliable platform for handling user requests.

RESULT

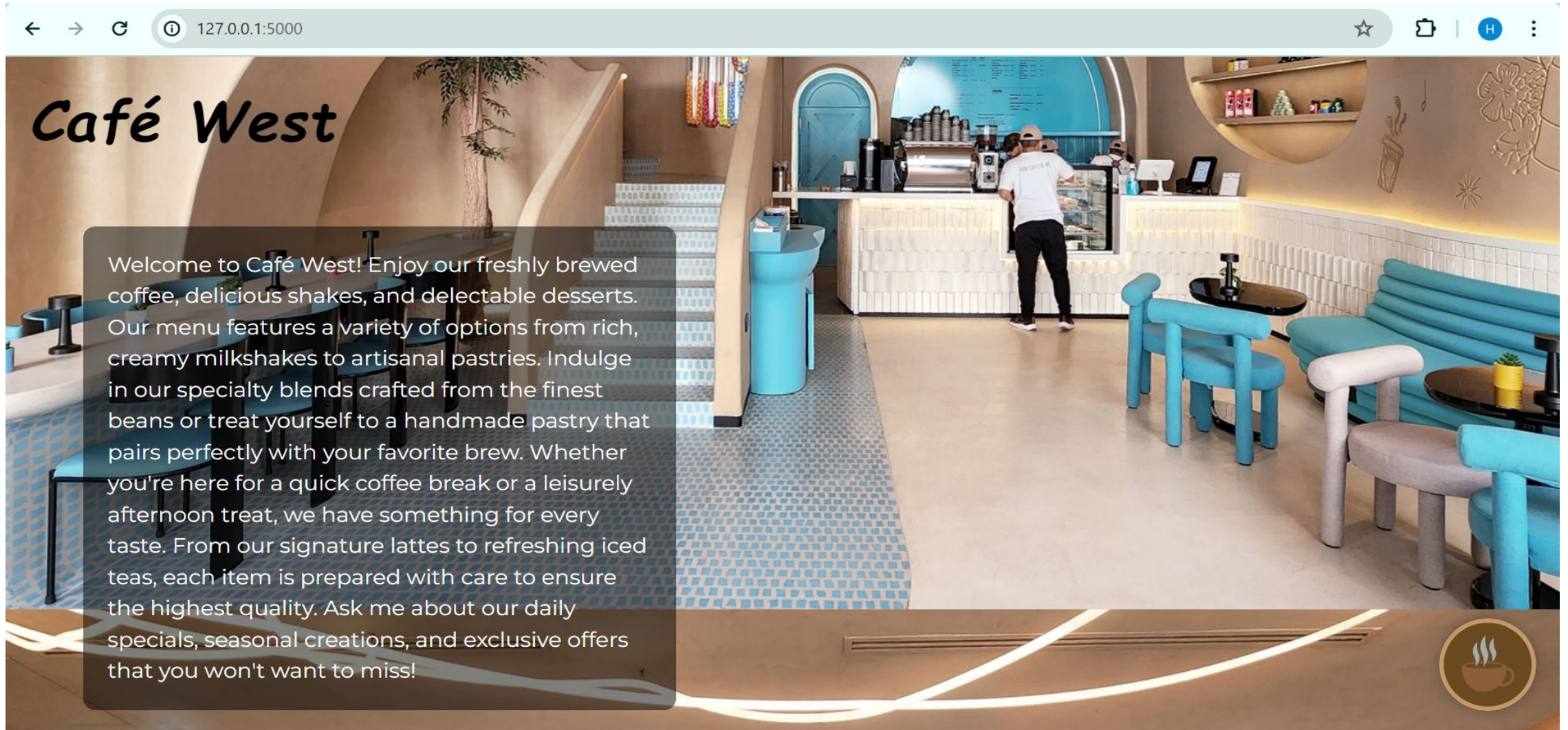


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

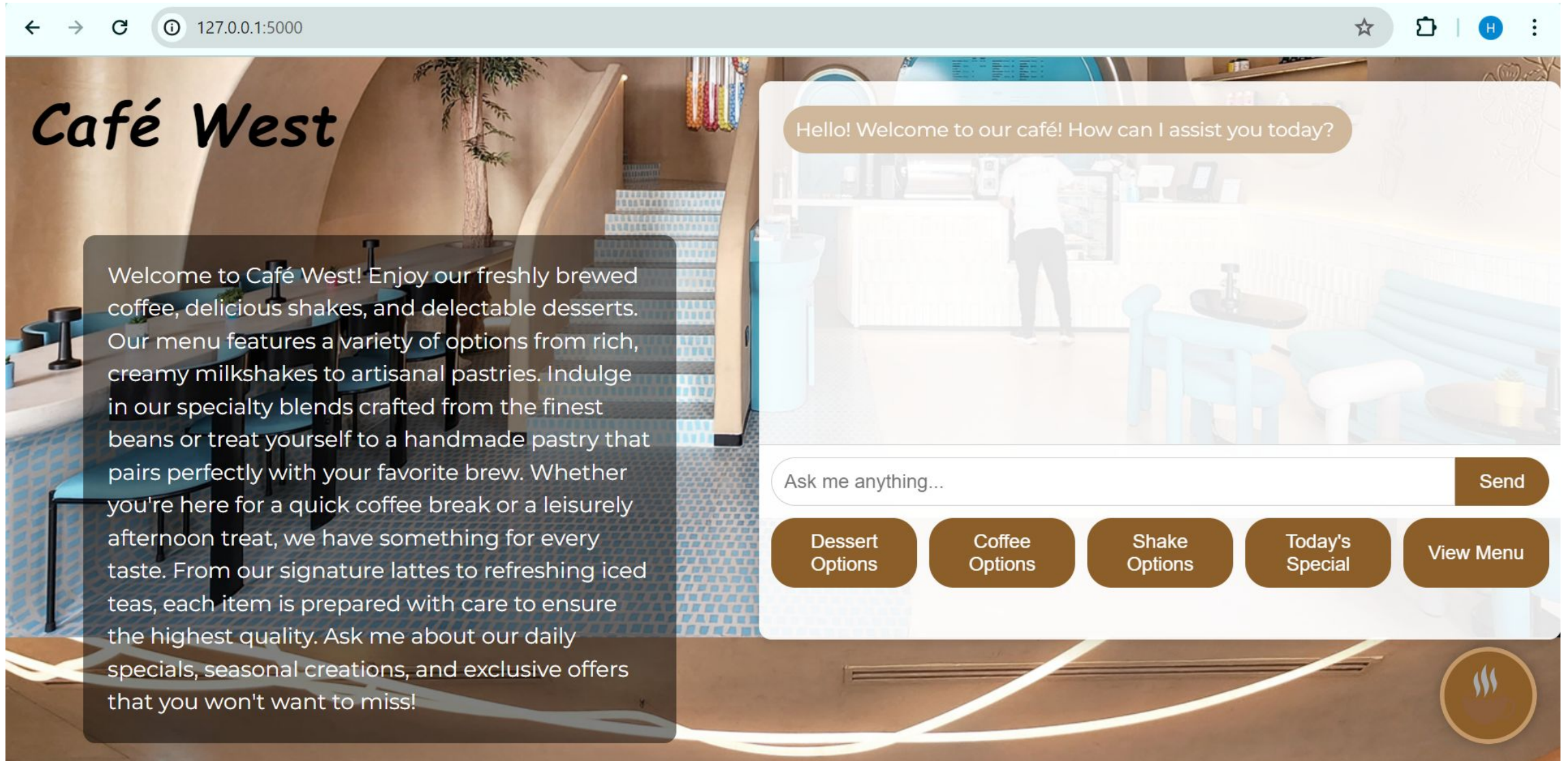
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\dell> cd cafe_chatbot
PS C:\Users\dell\cafe_chatbot> python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 846-553-956
```

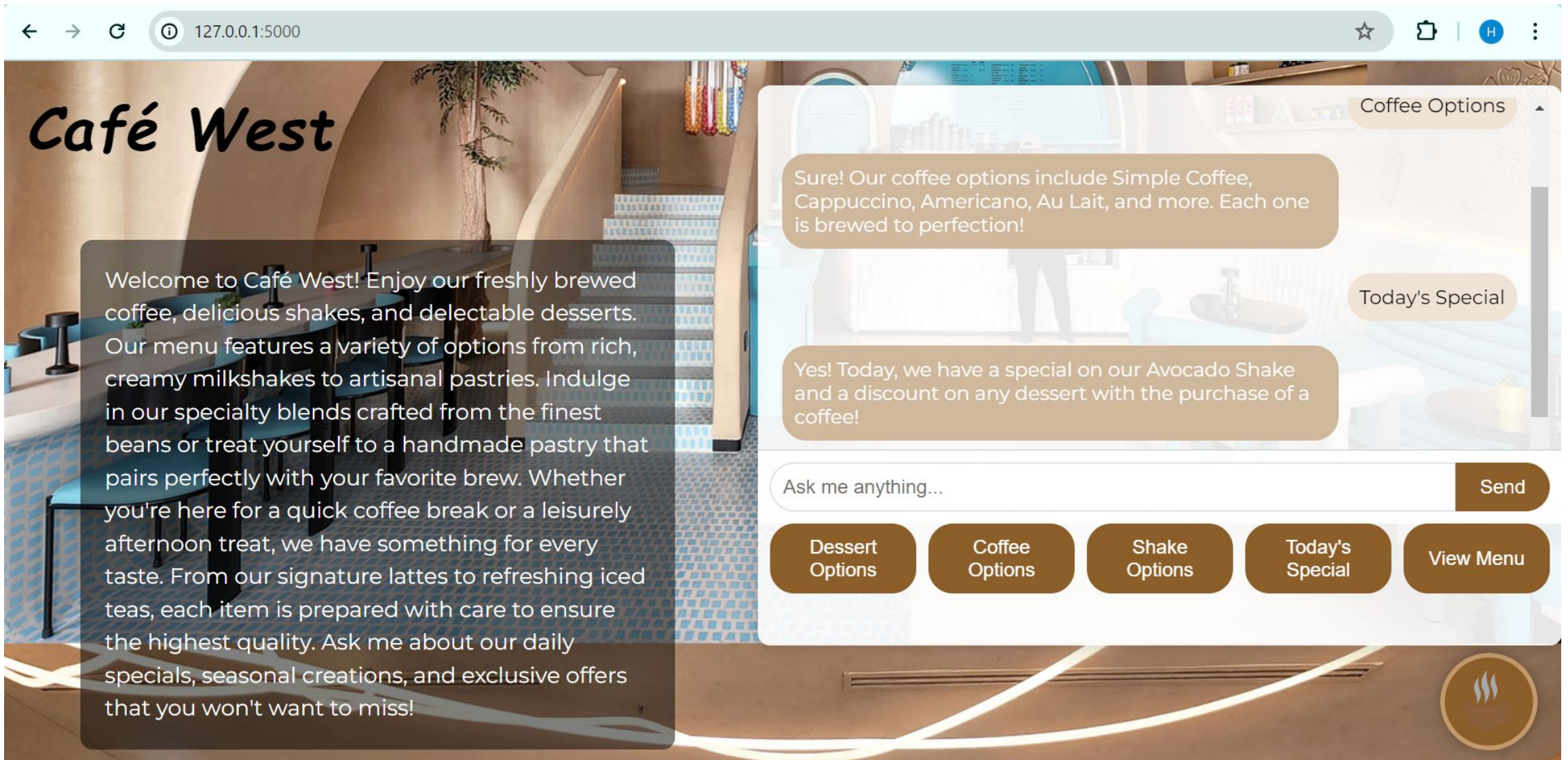

RESULT



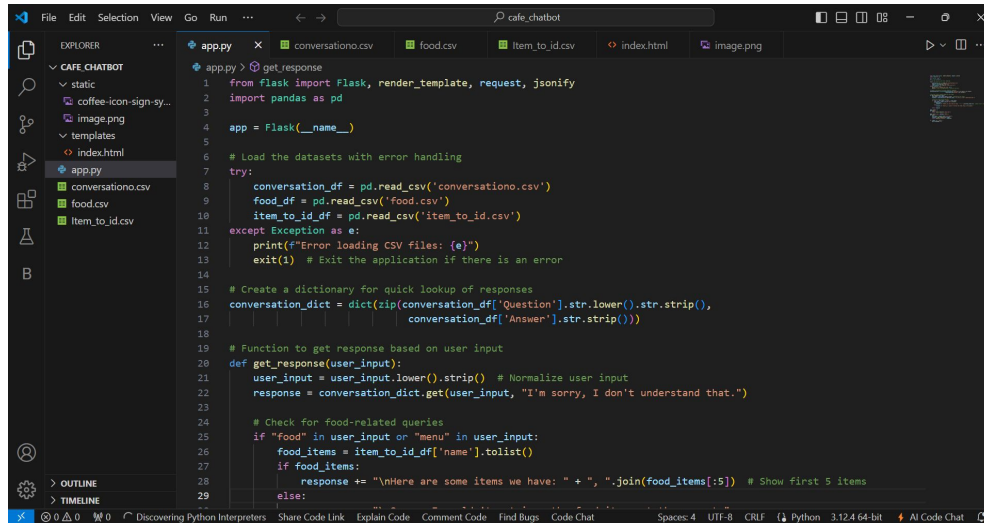
RESULT



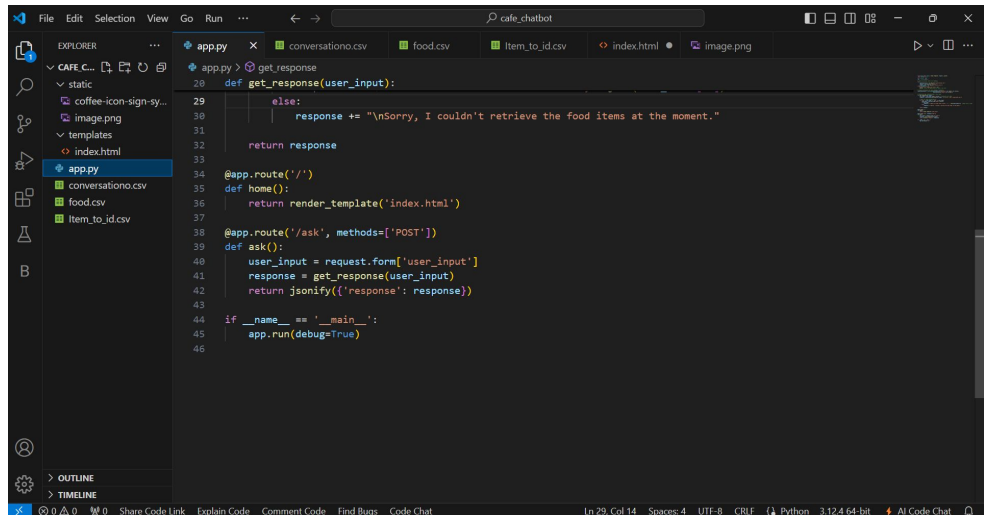
RESULT



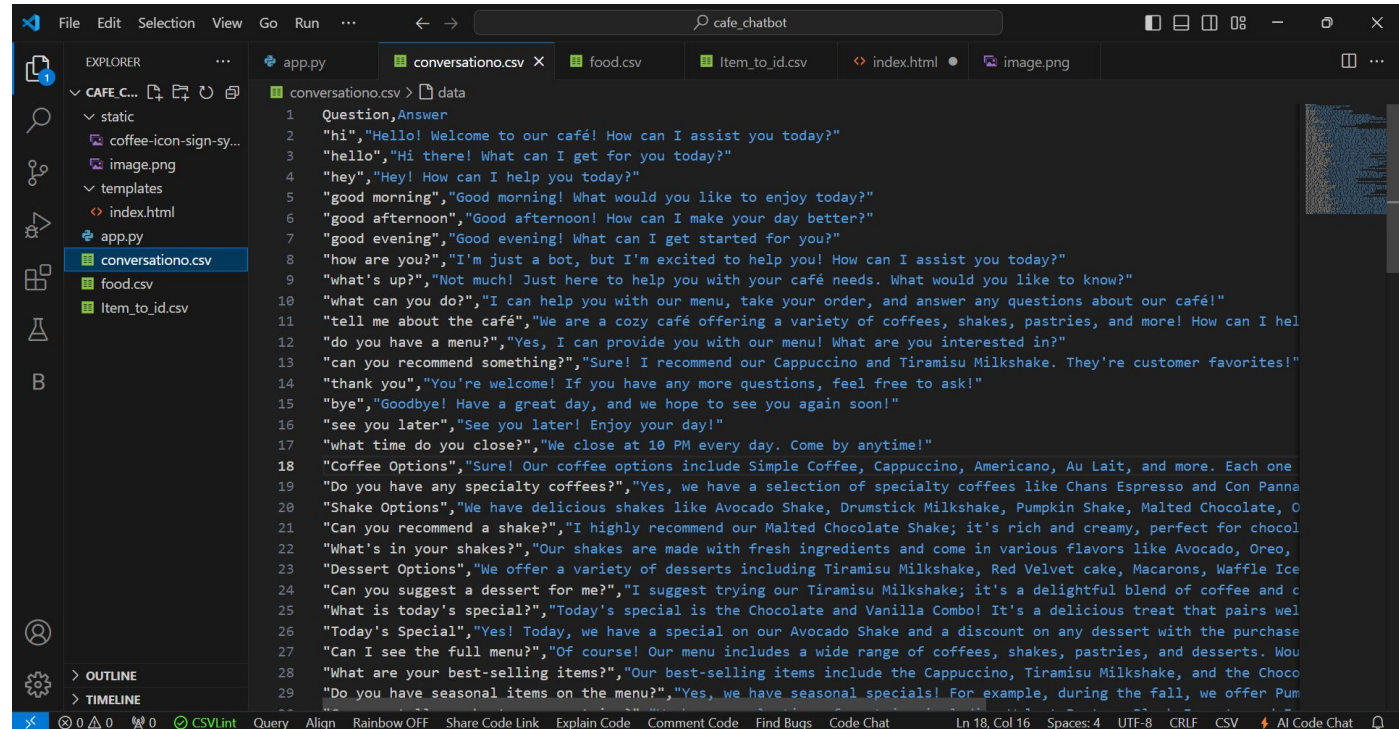
IMPLEMENTATION



```
1 from flask import Flask, render_template, request, jsonify
2 import pandas as pd
3
4 app = Flask(__name__)
5
6 # Load the datasets with error handling
7 try:
8     conversation_df = pd.read_csv('conversationo.csv')
9     food_df = pd.read_csv('food.csv')
10    item_to_id_df = pd.read_csv('item_to_id.csv')
11 except Exception as e:
12     print(f"Error loading CSV files: {e}")
13     exit(1) # Exit the application if there is an error
14
15 # Create a dictionary for quick lookup of responses
16 conversation_dict = dict(zip(conversation_df['Question'].str.lower().str.strip(),
17                             conversation_df['Answer'].str.strip()))
18
19 # Function to get response based on user input
20 def get_response(user_input):
21     user_input = user_input.lower().strip() # Normalize user input
22     response = conversation_dict.get(user_input, "I'm sorry, I don't understand that.")
23
24     # Check for food-related queries
25     if "food" in user_input or "menu" in user_input:
26         food_items = item_to_id_df['name'].tolist()
27         if food_items:
28             response += "\nHere are some items we have: " + ", ".join(food_items[:5]) # Show first 5 items
29     else:
```

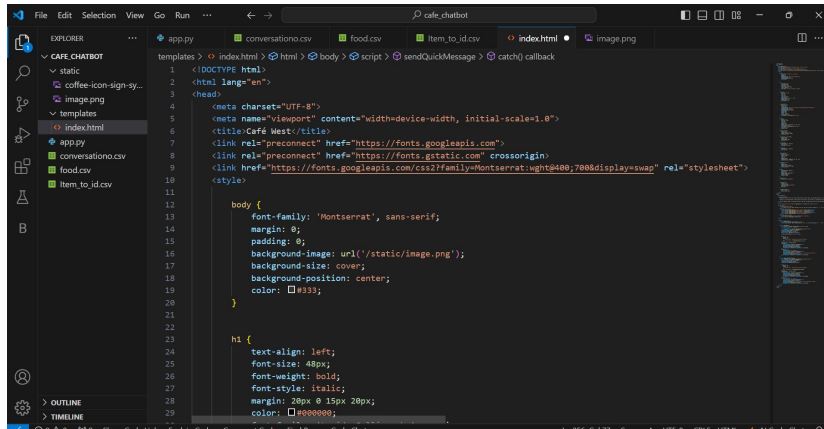


```
29     else:
30         response += "\nSorry, I couldn't retrieve the food items at the moment."
31     return response
32
33 @app.route('/')
34 def home():
35     return render_template('index.html')
36
37 @app.route('/ask', methods=['POST'])
38 def ask():
39     user_input = request.form['user_input']
40     response = get_response(user_input)
41     return jsonify({'response': response})
42
43 if __name__ == '__main__':
44     app.run(debug=True)
45
46
```

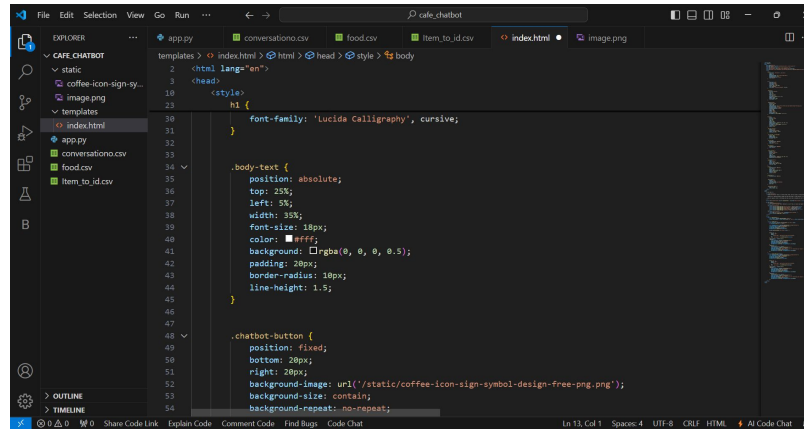


```
1 Question,Answer
2 "hi","Hello! Welcome to our café! How can I assist you today?"
3 "hello","Hi there! What can I get for you today?"
4 "hey","Hey! How can I help you today?"
5 "good morning","Good morning! What would you like to enjoy today?"
6 "good afternoon","Good afternoon! How can I make your day better?"
7 "good evening","Good evening! What can I get started for you?"
8 "how are you?","I'm just a bot, but I'm excited to help you! How can I assist you today?"
9 "what's up?","Not much! Just here to help you with your café needs. What would you like to know?"
10 "what can you do?","I can help you with our menu, take your order, and answer any questions about our café!"
11 "tell me about the café","We are a cozy café offering a variety of coffees, shakes, pastries, and more! How can I hel
12 "do you have a menu?","Yes, I can provide you with our menu! What are you interested in?"
13 "can you recommend something?","Sure! I recommend our Cappuccino and Tiramisu Milkshake. They're customer favorites!"
14 "thank you","You're welcome! If you have any more questions, feel free to ask!"
15 "bye","Goodbye! Have a great day, and we hope to see you again soon!"
16 "see you later","See you later! Enjoy your day!"
17 "what time do you close?","We close at 10 PM every day. Come by anytime!"
18 "Coffee Options","Sure! Our coffee options include Simple Coffee, Cappuccino, Americano, Au Lait, and more. Each one
19 "Do you have any specialty coffees?","Yes, we have a selection of specialty coffees like Chans Espresso and Con Panna
20 "Shake Options","We have delicious shakes like Avocado Shake, Drumstick Milkshake, Pumpkin Shake, Malted Chocolate, O
21 "Can you recommend a shake?","I highly recommend our Malted Chocolate Shake; it's rich and creamy, perfect for chocol
22 "What's in your shakes?","Our shakes are made with fresh ingredients and come in various flavors like Avocado, Oreo,
23 "Dessert Options","We offer a variety of desserts including Tiramisu Milkshake, Red Velvet cake, Macarons, Waffle Ice
24 "Can you suggest a dessert for me?","I suggest trying our Tiramisu Milkshake; it's a delightful blend of coffee and c
25 "What is today's special?","Today's special is the Chocolate and Vanilla Combo! It's a delicious treat that pairs wel
26 "Today's Special","Yes! Today, we have a special on our Avocado Shake and a discount on any dessert with the purchas
27 "Can I see the full menu?","Of course! Our menu includes a wide range of coffees, shakes, pastries, and desserts. Wou
28 "What are your best-selling items?","Our best-selling items include the Cappuccino, Tiramisu Milkshake, and the Choco
29 "Do you have seasonal items on the menu?","Yes, we have seasonal specials! For example, during the fall, we offer Pum
```

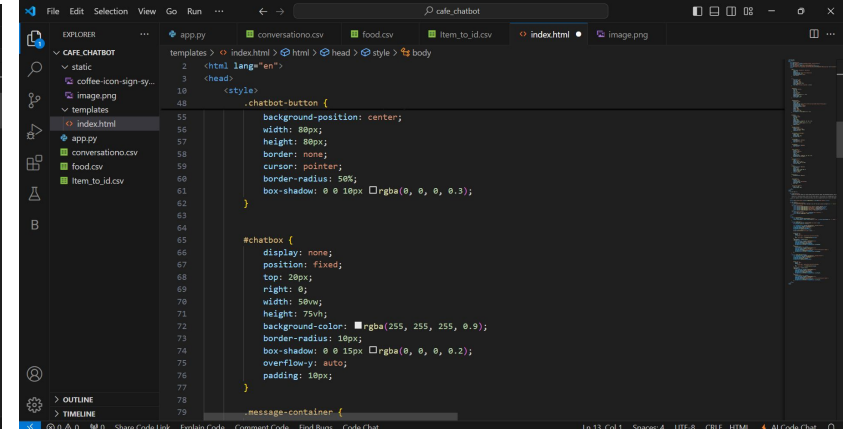

IMPLEMENTATION



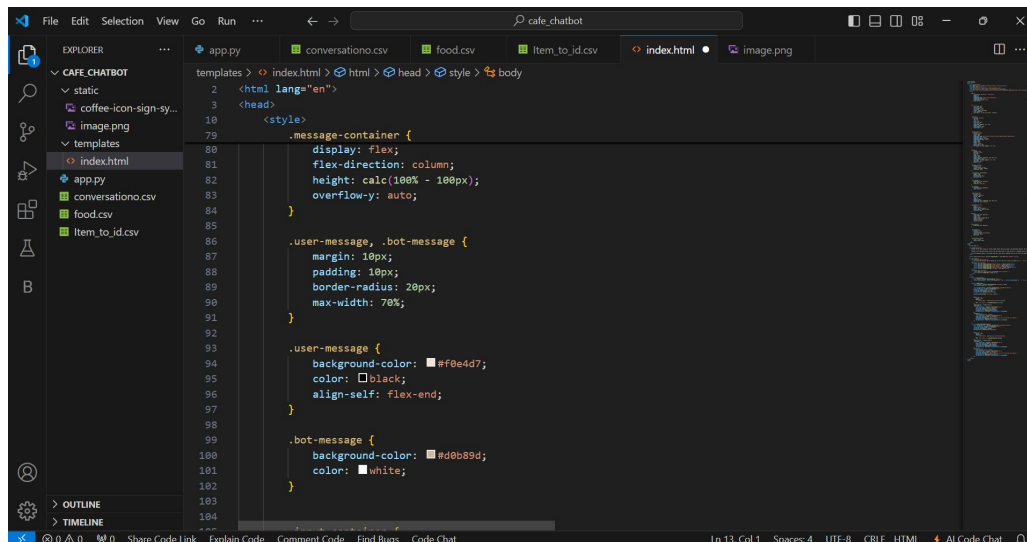
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Café West</title>
7 <link rel="preconnect" href="https://fonts.googleapis.com">
8 <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
9 <link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;700&display=swap" rel="stylesheet">
10 <style>
11
12 body {
13   font-family: 'Montserrat', sans-serif;
14   margin: 0;
15   padding: 0;
16   background-image: url('/static/image.png');
17   background-size: cover;
18   background-position: center;
19   color: #e333;
20 }
21
22
23 h1 {
24   text-align: left;
25   font-size: 40px;
26   font-weight: bold;
27   font-style: italic;
28   margin: 20px 0 15px 20px;
29   color: #000000;
30 }
```



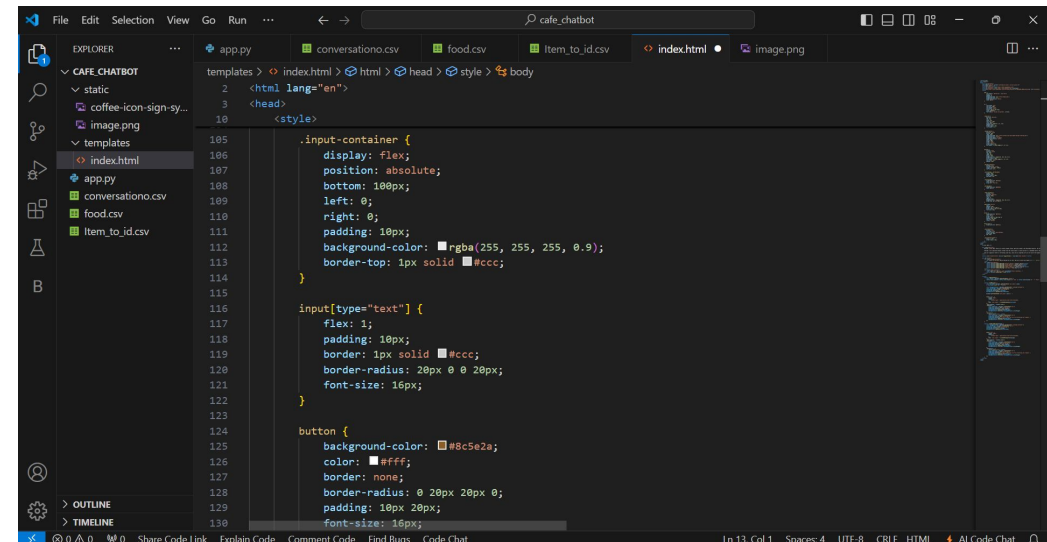
```
1
2 <html lang="en">
3 <head>
4 <style>
5
6 h1 {
7   font-family: 'Lucida Calligraphy', cursive;
8
9 }
10
11
12 .body-text {
13   position: absolute;
14   top: 25%;
15   left: 5%;
16   width: 35%;
17   font-size: 18px;
18   color: #e333;
19   background: #e333;
20   padding: 20px;
21   border-radius: 10px;
22   line-height: 1.5;
23 }
24
25
26 .chatbot-button {
27   position: fixed;
28   bottom: 20px;
29   right: 20px;
30   background-image: url('/static/coffee-icon-sign-symbol-design-free.png.png');
31   background-size: contain;
32   background-repeat: no-repeat;
33 }
```



```
1
2 <html lang="en">
3 <head>
4 <style>
5
6 .chatbot-button {
7   background-position: center;
8   width: 80px;
9   height: 80px;
10  border: none;
11  cursor: pointer;
12  border-radius: 50%;
13  box-shadow: 0 0 10px #e333;
14 }
15
16
17 #chatbot {
18   display: none;
19   position: fixed;
20   top: 20px;
21   right: 0;
22   width: 200px;
23   height: 75px;
24   background-color: #e333;
25   border-radius: 10px;
26   box-shadow: 0 0 10px #e333;
27   overflow: auto;
28   padding: 10px;
29 }
30
31 .message-container {
32 }
```

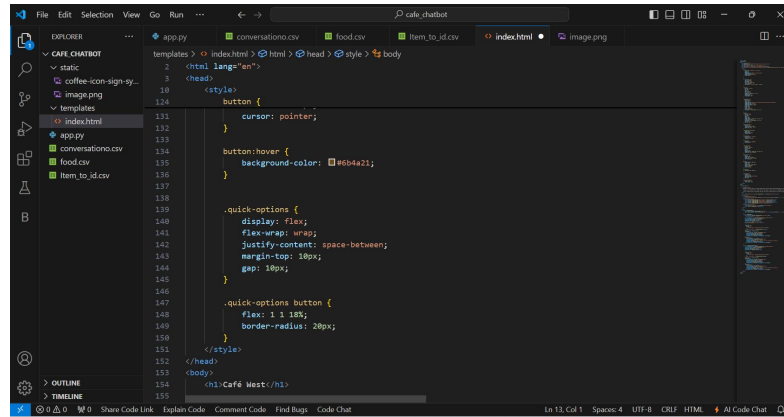


```
1
2 <html lang="en">
3 <head>
4 <style>
5
6 .message-container {
7   display: flex;
8   flex-direction: column;
9   height: calc(100% - 100px);
10  overflow-y: auto;
11 }
12
13
14 .user-message, .bot-message {
15   margin: 10px;
16   padding: 10px;
17   border-radius: 20px;
18   max-width: 70%;
19 }
20
21
22 .user-message {
23   background-color: #f0e4d7;
24   color: black;
25   align-self: flex-end;
26 }
27
28
29 .bot-message {
30   background-color: #d0b89d;
31   color: white;
32 }
```

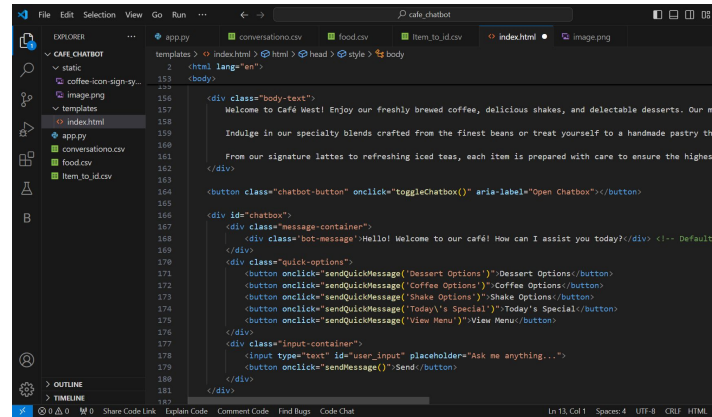


```
1
2 <html lang="en">
3 <head>
4 <style>
5
6 .input-container {
7   display: flex;
8   position: absolute;
9   bottom: 100px;
10  left: 0;
11  right: 0;
12  padding: 10px;
13  background-color: #e333;
14  border-top: 1px solid #e333;
15 }
16
17
18 input[type="text"] {
19   flex: 1;
20   padding: 10px;
21   border: 1px solid #e333;
22   border-radius: 20px 0 0 20px;
23   font-size: 16px;
24 }
25
26
27 button {
28   background-color: #8c5e2a;
29   color: #ffff;
30   border: none;
31   border-radius: 0 20px 20px 0;
32   padding: 10px 20px;
33   font-size: 16px;
34 }
```

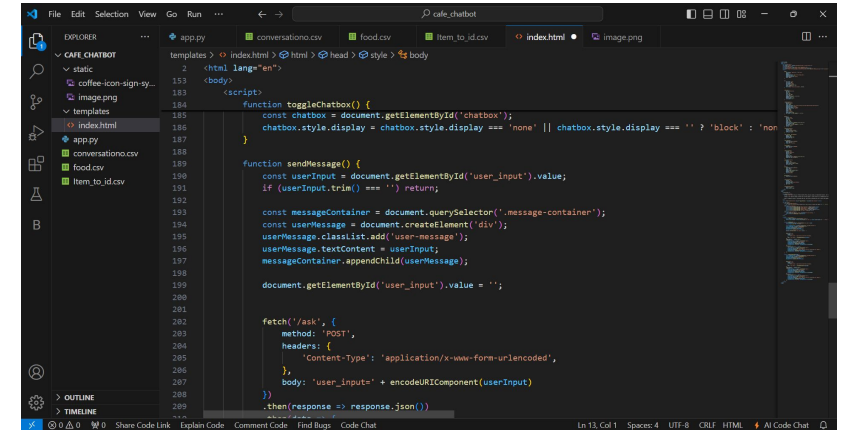
IMPLEMENTATION



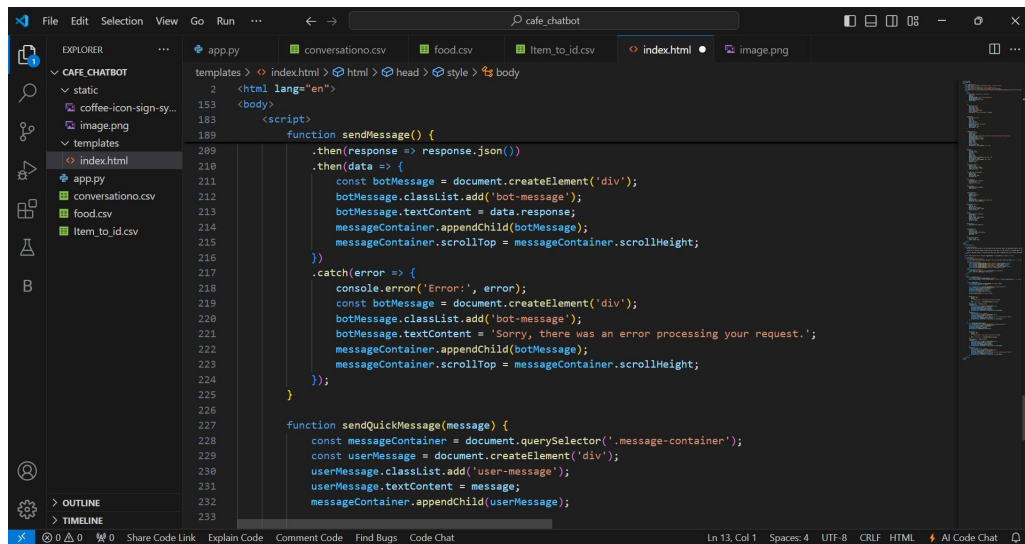
```
131 button {
132   cursor: pointer;
133 }
134
135 button:hover {
136   background-color: #808080;
137 }
138
139 .quick-options {
140   display: flex;
141   flex-wrap: wrap;
142   justify-content: space-between;
143   margin-top: 10px;
144   gap: 10px;
145 }
146
147 .quick-options button {
148   flex: 1 1 150px;
149   border-radius: 20px;
150 }
151
152 </style>
153 </head>
154 <body>
155   <!-- Café West -->
```



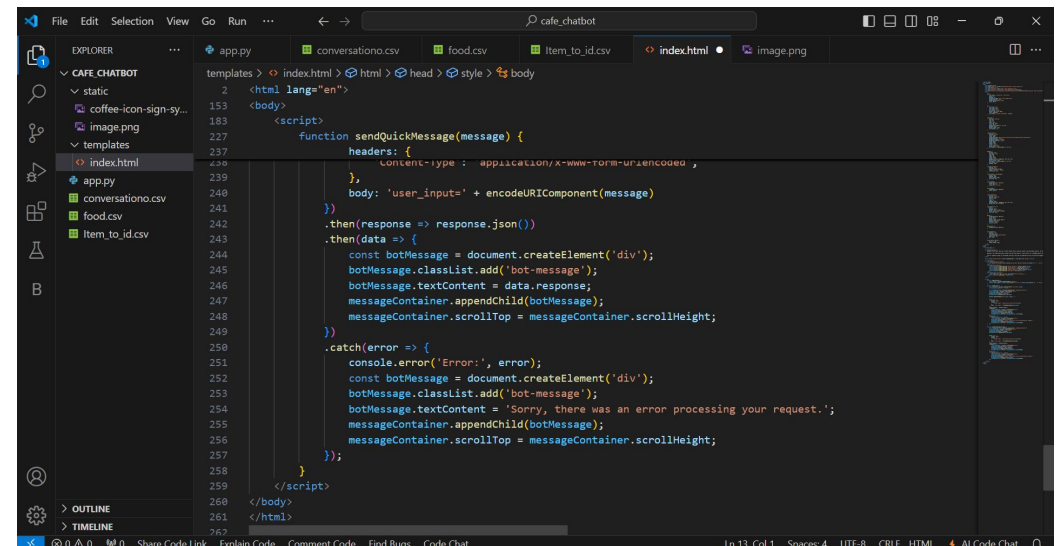
```
156 <div class="body-text">
157   Welcome to Café West! Enjoy our freshly brewed coffee, delicious shakes, and delectable desserts. Our menu
158   indulges in our specialty blends crafted from the finest beans or treat yourself to a handmade pastry that
159   From our signature lattes to refreshing iced teas, each item is prepared with care to ensure the highest q
160 </div>
161
162 <button class="chatbot-button" onclick="toggleChatbox()" aria-label="Open Chatbox"></button>
163
164 <div id="chatbox">
165   <div class="message-container">
166     <div class="bot-message">Hello! Welcome to our café! How can I assist you today?</div> <!-- Default me
167   </div>
168   <div class="quick-options">
169     <button onclick="sendQuickMessage('Dessert Options')">Dessert Options</button>
170     <button onclick="sendQuickMessage('Coffee Options')">Coffee Options</button>
171     <button onclick="sendQuickMessage('Shake Options')">Shake Options</button>
172     <button onclick="sendQuickMessage('Today's Special')">Today's Special</button>
173     <button onclick="sendQuickMessage('View Menu')">View Menu</button>
174   </div>
175   <div class="input-container">
176     <input type="text" id="user_input" placeholder="Ask me anything...">
177     <button onclick="sendMessage()">Send</button>
178   </div>
179 </div>
180 </body>
```



```
181
182
183 <script>
184   function toggleChatbox() {
185     const chatbox = document.getElementById('chatbox');
186     chatbox.style.display = chatbox.style.display === 'none' ? 'block' : 'none';
187   }
188
189   function sendMessage() {
190     const userInput = document.getElementById('user_input').value;
191     if (userInput.trim() === '') return;
192
193     const messageContainer = document.querySelector('.message-container');
194     const userMessage = document.createElement('div');
195     userMessage.classList.add('user-message');
196     userMessage.textContent = userInput;
197     messageContainer.appendChild(userMessage);
198
199     document.getElementById('user_input').value = '';
200
201     fetch('/ask', {
202       method: 'POST',
203       headers: {
204         'Content-Type': 'application/x-www-form-urlencoded',
205       },
206       body: 'user_input=' + encodeURIComponent(userInput)
207     })
208     .then(response => response.json())
```



```
209   .then(response => response.json())
210   .then(data => {
211     const botMessage = document.createElement('div');
212     botMessage.classList.add('bot-message');
213     botMessage.textContent = data.response;
214     messageContainer.appendChild(botMessage);
215     messageContainer.scrollTop = messageContainer.scrollHeight;
216   })
217   .catch(error => {
218     console.error('Error:', error);
219     const botMessage = document.createElement('div');
220     botMessage.classList.add('bot-message');
221     botMessage.textContent = 'Sorry, there was an error processing your request.';
222     messageContainer.appendChild(botMessage);
223     messageContainer.scrollTop = messageContainer.scrollHeight;
224   });
225
226
227   function sendQuickMessage(message) {
228     const messageContainer = document.querySelector('.message-container');
229     const userMessage = document.createElement('div');
230     userMessage.classList.add('user-message');
231     userMessage.textContent = message;
232     messageContainer.appendChild(userMessage);
233   }
```



```
234
235
236   function sendQuickMessage(message) {
237     headers: {
238       'Content-Type': 'application/x-www-form-urlencoded',
239     },
240     body: 'user_input=' + encodeURIComponent(message)
241   })
242   .then(response => response.json())
243   .then(data => {
244     const botMessage = document.createElement('div');
245     botMessage.classList.add('bot-message');
246     botMessage.textContent = data.response;
247     messageContainer.appendChild(botMessage);
248     messageContainer.scrollTop = messageContainer.scrollHeight;
249   })
250   .catch(error => {
251     console.error('Error:', error);
252     const botMessage = document.createElement('div');
253     botMessage.classList.add('bot-message');
254     botMessage.textContent = 'Sorry, there was an error processing your request.';
255     messageContainer.appendChild(botMessage);
256     messageContainer.scrollTop = messageContainer.scrollHeight;
257   });
258
259 </script>
260 </body>
261 </html>
262
```

CONCLUSION

- In this project, a chatbot system was designed and implemented to provide customers with a conversational interface to inquire about cafe information, such as menu items, prices, and operating hours. The proposed solution utilizes a rule-based system to match user inputs with pre-defined responses. The results of the project demonstrate the effectiveness of the proposed solution in providing accurate and helpful responses to user queries. The chatbot is able to understand a variety of questions and requests, and provides responses that are clear and concise.

FUTURE SCOPE

- Future enhancements could include integration with online ordering systems, personalization, multi-language support, emotional intelligence, voice assistant integration, loyalty program integration, real-time inventory management, customer feedback and ratings, chatbot analytics, and expansion to other platforms. Emerging technologies such as AI, ML and NLP and cloud computing can be leveraged to improve the chatbot's ability to understand and respond to customer queries, making it an even more powerful tool for cafes to improve customer engagement and drive business growth.

REFERENCES

- <https://www.sciencedirect.com/science/article/abs/pii/S1447677020302102>
- <https://www.kaggle.com/datasets/sonalibhoir/cafe-chatbot-dataset>
- <https://botpress.com/blog/chatbot-for-restaurants>
- <https://www.mdpi.com/2071-1050/15/7/5614>
- <https://www.sciencedirect.com/science/article/pii/S1877050922004689>

COURSE CERTIFICATE 1

In recognition of the commitment to achieve
professional excellence



Hamritha V

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence



Issued on: 24 JUL 2024

Issued by IBM

Verify: <https://www.credly.com/go/68Fno3C8>



COURSE CERTIFICATE 2

In recognition of the commitment to achieve
professional excellence



Hamritha V

Has successfully satisfied the requirements for:

Getting Started with Enterprise-grade AI



Issued on: 24 JUL 2024

Issued by IBM

Verify: <https://www.credly.com/go/gy7Xa64f>





THANK YOU