****

**Superior University Lahore**

***Lab Task # 10***

**Name:** Muhammad Abdullah Kashif

**Roll No**: SU92-BSAIM-F23-070

**Instructor**: Sir Rasikh Ali

# Course: Programming for Artificial Intelligence (Lab)

**Restaurant Chatbot using Flask**

**1. Project Overview**

This project creates a restaurant chatbot web app using Flask as the backend and HTML/CSS/JavaScript for the frontend. The chatbot can answer questions related to the restaurant's menu, opening hours, location, delivery services, and more.

**3. Backend (Flask) - app.py**

**Step 1: Import Libraries**

from flask import Flask, render\_template, request, jsonify

import random

from datetime import datetime

These modules are used for web server handling (Flask), template rendering, AJAX response handling, and random response generation.

**Step 2: Initialize Flask App**

app = Flask(\_\_name\_\_)

This line initializes the Flask app.

**Step 3: Restaurant Data**

restaurants = [ { ... } ]

A list of dictionaries containing all necessary data about the restaurant: menu, location, hours, dishes, delivery, etc.

**Step 4: Home Route**

@app.route('/')

def home():

return render\_template('index.html')

This route serves the main page (index.html) when a user accesses the root URL.

**Step 5: Chat Response Endpoint**

@app.route('/get\_response', methods=['POST'])

def get\_response():

user\_message = request.json['message']

response = generate\_response(user\_message.lower())

return jsonify({'response': response})

Handles the AJAX POST request and sends back the bot's reply using generate\_response.

**Step 6: Generate Bot Response**

def generate\_response(message):

# checks for different user message types

This function contains logic to handle various user inputs like greetings, menu inquiries, delivery options, etc., by matching keywords in the message.

**4. Frontend (HTML, CSS, JS)**

**HTML - templates/index.html**

* Contains the structure of the chat interface.
* Includes:
  + Input field
  + Chat area
  + Submit button
  + JavaScript and CSS links

<form id="chat-form">

<input type="text" id="user-input" placeholder="Type a message...">

<button type="submit">Send</button>

</form>

<div id="chat-box"></div>

**CSS - static/css/style.css**

* Styles the chat interface, chat bubbles, font, and layout.
* Adds background colors and spacing for a neat UI.

**JavaScript - static/js/script.js**

document.getElementById("chat-form").onsubmit = async function(e) {

e.preventDefault();

const input = document.getElementById("user-input").value;

// Show user message

// Fetch bot response using fetch('/get\_response')

// Append bot response to chat

}

* Sends user input to the Flask backend
* Appends both user and bot messages to the chat box

**5. Virtual Environment (venv)**

**Step 1: Create a Virtual Environment**

python -m venv venv

**Step 2: Activate it**

venv\Scripts\activate

**Packages**

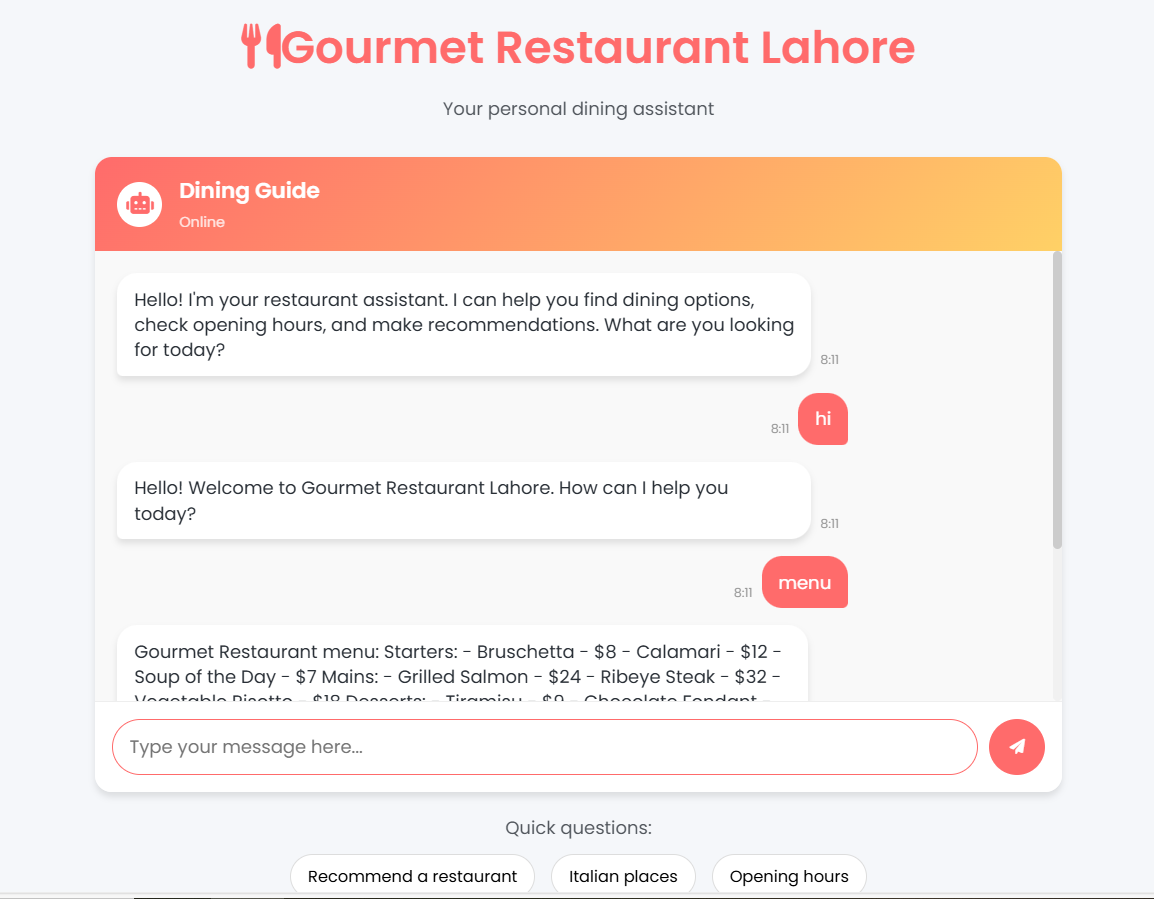
pip install flask

**6. Running the App**

flask run

Then visit http://127.0.0.1:5000 in your browser to interact with the chatbot.

**Output:**

****