

Explanation of Flask Web Application: ChatBot For DoT

1. Imports and Initialization

```
from flask import Flask, render_template, request, jsonify

app = Flask(__name__)
```

- Flask: A web framework for Python that facilitates the easy creation of web applications.
- render_template: Utilized for rendering HTML templates.
- request: Handles incoming requests to the server.
- jsonify: Converts Python dictionaries into JSON format for seamless response handling.

2. Data Structure

```
mtech_subjects = {
    'chemical': { ... },
    'civil': { ... },
    'computer': { ... },
    'electronics': { ... },
    'mechanical': { ... },
    'information security': { ... },
    'aviation': { ... }
}
```

- Purpose: This dictionary contains information about M.Tech subjects for various specializations (e.g., Chemical, Civil, Computer, Electronics, etc.). Each specialization includes a nested dictionary listing subjects for four semesters.

3. Routes

3.1. Home Route

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

- Description: This route (/) serves as the main page of the application. It renders the index.html template, which is likely the user interface for the chatbot.

3.2. API Route for Chatbot Messages

```
@app.route('/api/message', methods=['POST'])
def message():
    user_message = request.json.get('message').lower()
```

- Description: This route (/api/message) listens for POST requests containing a user's message. It extracts the message, converts it to lowercase for uniformity, and processes it to generate an appropriate response.

4. Response Logic

The chatbot generates different responses based on keywords found in the user's message. Here's a breakdown of its functionality:

4.1. M.Tech Courses

```
if any(keyword in user_message for keyword in ['m-tech', 'mtech', 'master in technology']):  
    bot_response = (  
        "M-Tech Courses:\n\n"  
        "1. M-Tech (Chemical and Biotechnology)\n"  
        ...  
    )
```

- Functionality: If the user mentions M.Tech, the bot responds with a list of available M.Tech courses.

4.2. PG Diploma Courses

```
elif any(keyword in user_message for keyword in ['pg diploma', 'post graduate diploma', 'pg', 'PG']):  
    bot_response = (  
        "PG Diploma Courses:\n\n"  
        "1. P.G. Diploma in Water Technology and Management\n"  
        ...  
    )
```

- Functionality: This section employs similar logic for responding to queries about PG Diploma courses.

4.3. Undergraduate Courses

```
elif any(keyword in user_message for keyword in ['undergraduate', 'ug', 'UG']):  
    bot_response = (  
        "Undergraduate Courses:\n\n"  
        "1. B.Sc. Animation and VFX\n"  
        ...  
    )
```

- Functionality: The bot responds with a list of undergraduate courses when the user mentions related keywords.

4.4. Gratitude Response

```
elif any(keyword in user_message for keyword in ['thank you', 'thanks', 'appreciate it']):  
    bot_response = "You're welcome! Please provide your feedback."  
    https://forms.gle/ivfqJatEswdSa1r78"
```

- Functionality: If the user expresses gratitude, the bot acknowledges it and provides a feedback link.

4.5. Subjects for Specific M.Tech Courses

```
elif 'subject' in user_message and any(course in user_message for course in  
mtech_subjects.keys()):  
    ...
```

- Functionality: If the user inquires about subjects for a specific M.Tech course, the bot identifies the course and checks for the specified semester.

5. Semester Checks

```
if 'semester' in user_message:  
    semester = [sem for sem in ['semester 1', 'semester 2', 'semester 3', 'semester 4'] if sem in  
user_message]  
    ...
```

- Functionality: The bot verifies if the user has mentioned a specific semester and retrieves the corresponding subjects for the selected M.Tech course.

6. Default Responses

```
else:  
    bot_response = "Please specify if you want information about M-Tech, PG Diploma, or  
Undergraduate courses!"
```

- Functionality: If none of the keywords match, the bot prompts the user to clarify their request.

7. JSON Response

```
return jsonify({'response': bot_response})
```

- Description: The bot sends back the generated response in JSON format.

8. Running the Application

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

- Description: This snippet runs the Flask application with debug mode enabled, which facilitates easier debugging during development.