Implement ChatGPT in a Flask Application

Last Updated : 24 May, 2023

You can build dynamic and interactive chat interfaces by integrating ChatGPT into a Flask application. This article's instructions can help you integrate ChatGPT into your Flask project and give users engaging chat experiences. Improve the user interface, try out new prompts, and look into new options to make your chat program more functional.

Steps to Implement ChatGPT in a Flask Application Using API

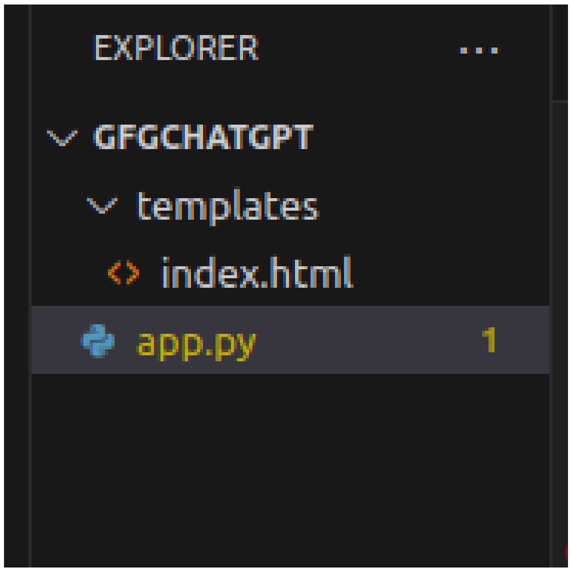
Integrating ChatGPT into a Flask web application allows you to build interactive chat interfaces and provide dynamic conversational experiences to users. Let's see the overall steps in summary:

- Setting up the Flask Project.
- Define a Flask app.py that handles the chat functionality.
- Configure Django's URL routing to connect the chat view to a specific URL pattern.
- Designing the User Interface.

- Testing and Running the Application.

Implement ChatGPT in a Flask Application

Folder Structure



app.py: Creating the Chat View

In this code, you need to have a valid OpenAI API key, which you should replace with 'YOUR_API_KEY'. The Flask application listens for requests on the root route '/', and when a POST request is received, it generates a response based on the user's prompt using the OpenAI API. The response is then returned as JSON.

Python3

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


app = Flask(__name__)

# OpenAI API Key
openai.api_key = 'YOUR_API_KEY'

def get_completion(prompt):
    print(prompt)
    query = openai.Completion.create(
        engine="text-davinci-003",
        prompt=prompt,
        max_tokens=1024,
        n=1,
        stop=None,
        temperature=0.5,
    )

    response = query.choices[0].text
    return response

@app.route("/", methods=['POST', 'GET'])
def query_view():
    if request.method == 'POST':
        print('step1')
        prompt = request.form['prompt']
        response = get_completion(prompt)
        print(response)

        return jsonify({'response': response})
    return render_template('index.html')


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

templates/index.html: Creating User Interface

Create an HTML template for the chat interface using HTML, CSS, and Bootstrap, and Set up the necessary JavaScript and AJAX code to handle user interaction.

HTML

```
<!-- query.html -->
<html>
<head>
    <title>Query</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/js-cookie@3.0.0/dist/js.cookie.min.j
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/boots
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/boots
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.1€
    <script>
```

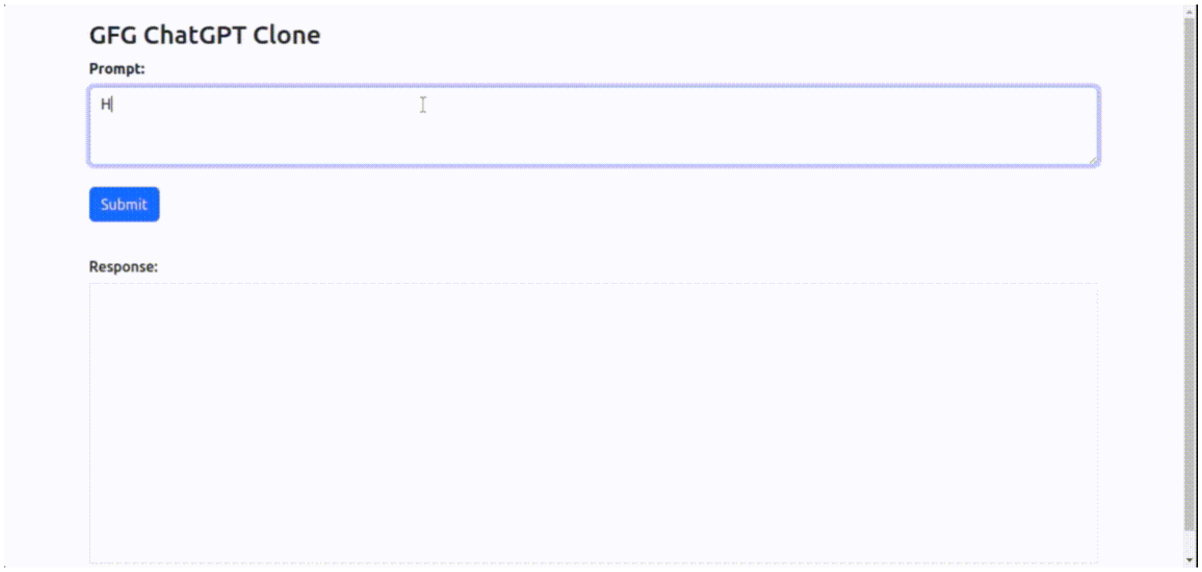
```
            $(document).ready(function() {
                // Send the form on enter keypress and avoid if shift is pressed
                $('#prompt').keypress(function(event) {
                    if (event.keyCode === 13 && !event.shiftKey) {
                        event.preventDefault();
                        $('form').submit();
                    }
                });
                $('form').on('submit', function(event) {
                    event.preventDefault();
                // get the CSRF token from the cookie
                var csrftoken = Cookies.get('csrftoken');

                // set the CSRF token in the AJAX headers
                $.ajaxSetup({
                    headers: { 'X-CSRFToken': csrftoken }
                });
                    // Get the prompt
                    var prompt = $('#prompt').val();
                    var dateTime = new Date();
                    var time = dateTime.toLocaleTimeString();
                    // Add the prompt to the response div
                    $('#response').append('<p id="GFG1">('+ time + ') <i class="bi bi
                    $('#response #GFG1').css({"color": "green", "width": "90%", "float
                    // Clear the prompt
                    $('#prompt').val('');
                    $.ajax({
                        url: '/',
                        type: 'POST',
                        data: {prompt: prompt},
                        dataType: 'json',
                        success: function(data) {
                            $('#response').append('<p id="GFG2">('+ time + ') <i clas
                            $('#response #GFG2').css({"color": "red", "width": "90%",
                        }
                    });
                });
            });
        </script>
    </head>
    <body>
        <div class="container p-3">
            <h3>GFG ChatGPT Clone</h3>
            <div class="mb-3">
                <form method="post" action="">

                    <label for="prompt" class="form-label"><strong>Prompt: </strong><
                    <textarea class="form-control" type="textarea" id="prompt" name="
                    <br>
                    <button class="btn btn-primary " type="submit">Submit</button>
                </form>
            </div>
            <br>
            <div class="mb-3">
                <h6>Response:</h6>
                <div class="container border overflow-auto h-50" id="response"></div>

            </div>
        </div>
    </body>
    </html>
```

Output:



To learn more about Chat GPT, you can refer to:

- [7 Best AI Businesses to Start with ChatGPT]
- [ChatGPT vs Google BARD]
- [ChatGPT's History Gone: Data Breach or AI Bug?]

Ready to dive into the future? Mastering Generative AI and ChatGPT is your gateway to the cutting-edge world of AI. Perfect for tech enthusiasts, this course will teach you how to leverage **Generative AI** and **ChatGPT** with hands-on, practical lessons. Transform your skills and create innovative AI applications that stand out. Don't miss out on becoming an AI expert – Enroll now and start shaping the future!

skrg141 1

Previous Article

Create a ChatBot with OpenAI and Gradio in Python

Next Article

Creating ChatGPT Clone in Python

Similar Reads

Documenting Flask Endpoint using Flask-Autodoc

Documentation of endpoints is an essential task in web development and being able to apply it in different frameworks is always a utility. This article discusses...

4 min read

How to use Flask-Session in Python Flask ?

Flask Session - Flask-Session is an extension for Flask that supports Server-side Session to your application.The Session is the time between the client logs in to...

4 min read

How to Integrate Flask-Admin and Flask-Login

In order to merge the admin and login pages, we can utilize a short form or any other login method that only requires the username and password. This is know...

8 min read

Minify HTML in Flask using Flask-Minify

Flask offers HTML rendering as output, it is usually desired that the output HTML should be concise and it serves the purpose as well. In this article, we would...

12 min read

Flask URL Helper Function - Flask url_for()

In this article, we are going to learn about the flask url_for() function of the flask URL helper in Python. Flask is a straightforward, speedy, scalable library, used f...

11 min read

How to Implement Filtering, Sorting, and Pagination in Flask

Python-based Flask is a microweb framework. Due to the fact that it doesn't require any specific tools or libraries, it is categorized as a micro-framework. It...

6 min read

Implement a Python REST API with Flask & Flasgger

Building a RESTful API in Python can be straightforward with Flask, a lightweight and flexible web framework. To add comprehensive documentation and...

4 min read

Profile Application using Python Flask and MySQL

A framework is a code library that makes a developer's life easier when building web applications by providing reusable code for common operations. There are ...

10 min read

Create Postman collection from Flask application using flask2postman

Prerequisite: Introduction to Postman, First App using Flask Since Postman is gaining popularity in the development domain, this article explains a way in...

3 min read

Build a Flask application to validate CAPTCHA

In this article, we are going a build a web application that can have a CAPTCHA. CAPTCHA is a tool you can use to differentiate between real users and...

5 min read

Article Tags : Python ChatGPT Flask Projects Python Flask

Practice Tags : python

Company
About Us
Legal
In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program
GeeksforGeeks Community

Languages
Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial
Tutorials Archive

DSA
Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Data Science & ML
Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Web Technologies
HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Python Tutorial
Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

Computer Science

DevOps