Bạn có thể sử dụng thư viện **Leaflet.js**, một thư viện JavaScript mã nguồn mở, để hiển thị bản đồ OSM trên ứng dụng web.

**Hướng dẫn tích hợp OpenStreetMap vào chatbot**

1. **Thêm Leaflet.js vào dự án của bạn**: Để sử dụng OSM với Leaflet.js, bạn cần thêm các file JavaScript và CSS của Leaflet vào file HTML của chatbot. Dưới đây là cách bạn có thể làm:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Trò chuyện với Chatbot</title>

<link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css" />

<script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"></script>

<style>

#map {

height: 400px;

width: 100%;

}

</style>

</head>

<body>

<h1>Trò chuyện với Chatbot</h1>

<div>

<form id="chat-form" method="POST" action="/chat">

<label for="message">Nhập câu hỏi của bạn:</label>

<input type="text" name="message" id="message" required>

<button type="submit">Gửi</button>

</form>

</div>

<div>

<h3>Bản đồ hỗ trợ:</h3>

<!-- Div cho bản đồ -->

<div id="map"></div>

</div>

<script>

// Tạo bản đồ OpenStreetMap với Leaflet.js

var map = L.map('map').setView([10.8231, 106.6297], 13); // [vĩ độ, kinh độ], 13 là mức zoom

// Thêm layer từ OpenStreetMap

L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {

attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'

}).addTo(map);

// Thêm marker ví dụ (TP. Hồ Chí Minh)

var marker = L.marker([10.8231, 106.6297]).addTo(map)

.bindPopup('Vị trí: TP. Hồ Chí Minh')

.openPopup();

</script>

</body>

</html>

1. **Chi tiết các bước**:
   * **Bước 1**: Thêm file CSS và JavaScript của Leaflet từ CDN để sử dụng trong ứng dụng web của bạn.
   * **Bước 2**: Tạo một thẻ <div> với id="map" để hiển thị bản đồ.
   * **Bước 3**: Trong phần JavaScript, bạn tạo bản đồ bằng cách gọi L.map() và định vị bản đồ với setView([vĩ độ, kinh độ], zoom). Trong ví dụ trên, bản đồ được định vị tại **TP. Hồ Chí Minh** với tọa độ [10.8231, 106.6297].
   * **Bước 4**: Thêm các layer bản đồ từ OpenStreetMap bằng L.tileLayer().
   * **Bước 5**: Thêm một marker (điểm đánh dấu) tại một vị trí cụ thể với L.marker() và hiển thị hộp thông tin với bindPopup().
2. **Tùy chỉnh theo chatbot**:
   * Bạn có thể cập nhật tọa độ của bản đồ và marker dựa trên câu trả lời của chatbot. Nếu chatbot trả về một vị trí cụ thể dựa trên câu hỏi của người dùng, bạn có thể sử dụng JavaScript để cập nhật vị trí bản đồ và marker theo thời gian thực.
   * Ví dụ, nếu người dùng hỏi về vị trí của một địa điểm, chatbot có thể trả về tọa độ của địa điểm đó và bạn sẽ sử dụng chúng để cập nhật bản đồ.

// Ví dụ: Cập nhật bản đồ khi chatbot trả về vị trí

function updateMap(latitude, longitude, locationName) {

map.setView([latitude, longitude], 13);

L.marker([latitude, longitude]).addTo(map)

.bindPopup(`Vị trí: ${locationName}`)

.openPopup();

}

1. **Thêm logic chatbot trong Python**: Trong chatbot.py, bạn có thể tích hợp phản hồi của chatbot để cung cấp tọa độ cho một vị trí. Ví dụ, khi người dùng hỏi về khoảng cách từ một địa điểm đến địa điểm khác, chatbot có thể trả về thông tin bao gồm cả tọa độ để cập nhật bản đồ.

class GeminiChatbot:

def get\_response(self, message):

try:

# Giả sử chatbot trả về thông tin vị trí (tọa độ)

if "vị trí" in message.lower():

latitude = 10.8231

longitude = 106.6297

location\_name = "TP. Hồ Chí Minh"

return f"{location\_name} có tọa độ là ({latitude}, {longitude}).", latitude, longitude

else:

return "Xin lỗi, tôi không thể tìm thấy vị trí.", None, None

except Exception as e:

return "Có lỗi xảy ra.", None, None

OpenStreetMap và Leaflet.js là những công cụ mạnh mẽ để tích hợp bản đồ miễn phí vào chatbot của bạn. Bạn có thể dễ dàng tùy chỉnh bản đồ dựa trên phản hồi của chatbot, và người dùng có thể tương tác trực tiếp với bản đồ ngay trên giao diện chatbot.