AWS Chatbot

Step1: Create a Lambda Function for API Call

```
import json
import requests
import os

def lambda_handler(event, context):
    try:
        # Handle both direct test events and API Gateway events
        body = event.get("body")
        if isinstance(body, str):
            body = json.loads(body)
        elif body is None:
            body = event  # Treat entire event as input if 'body' key is missing

        user_input = body.get("message", "Who is Virat Kohli?")

        api_url = os.environ.get("LLM_API_URL")
        api_key = os.environ.get("LLM_API_KEY")
        model_name = os.environ.get("LLM_MODEL_NAME")

        headers = {
            "Authorization": f"Bearer {api_key}",
            "Content-Type": "application/json",
        }

        json_data = {
            "model": model_name,
            "messages": [{"role": "user", "content": user_input}],
        }

        res = requests.post(api_url, json=json_data, headers=headers)
        res.raise_for_status()
        llm_response = res.json()

        response_content = llm_response.get("choices", [{}])[0].get("message", {}).get("content", "No
response")

        return {
            'statusCode': 200,
            'body': json.dumps({'response': response_content})
        }

    except Exception as e:
        return {
            'statusCode': 500,
            'body': json.dumps({'error': str(e)})
        }
```

Step 2 : In Lambda

Go to configurations > Environment Variables

And Add:

LLM_API_URL --> sk-bf725748416143d88b7ea444d68f0c90
LLM_API_KEY --> llama3.2-vision:latest
LLM_MODEL_NAME --> https://chat.ivislabs.in/api/chat/completions

Step 3: Add a zip folder in Layer for importing modules To create a zip use Windows Powershell or cmd

Run these commands in cmd or Windows powershell

mkdir requests-layer
cd requests-layer
mkdir python
pip install requests -t python/ # Installs requests and stores in python directory

zip -r requests-layer.zip python #For Cmd

Compress-Archive -Path python -DestinationPath requests-layer.zip #For Windows PowerShell

Then add that layer to the Lambda

Step 4: API Gateway

Go to API Gateway → Create an HTTP API.

Choose “Add Integration” → Select your Lambda function.

Add route: POST /chat

Deploy → copy the Invoke URL.

Paste the copied Invoke URL in the next html file

Step 5 : Create an HTML and host it using S3 static hosting

index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>AI Chatbot</title>
  <style>
    body { font-family: Arial; margin: 20px; }
    #chatBox { border: 1px solid #ccc; padding: 10px; height: 300px; overflow-y: scroll; }
    input { padding: 8px; width: 70%; }
    button { padding: 8px; }
  </style>
</head>
<body>

<h2>My AI Chatbot</h2>
<div id="chatBox"></div><br>
<input type="text" id="userInput" placeholder="Type your message...">
<button onclick="sendMessage()">Send</button>

<script>
  async function sendMessage() {
    const input = document.getElementById("userInput").value;
    const chatBox = document.getElementById("chatBox");

    if (!input.trim()) return; // Prevent empty input

    chatBox.innerHTML += `<p><strong>You:</strong> ${input}</p>`;

    try {
      const res = await
fetch("https://e4iwkiwukj.execute-api.ap-south-1.amazonaws.com/CHatbotstage/chat", {
        method: "POST",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify({ message: input })
      });

      const data = await res.json();

      if (data.response) {
        chatBox.innerHTML += `<p><strong>Bot:</strong> ${data.response}</p>`;
      } else if (data.error) {
        chatBox.innerHTML += `<p><strong>Bot (error):</strong> ${data.error}</p>`;
      } else {
```

```
      chatBox.innerHTML += `<p><strong>Bot:</strong> No response</p>`;
    }

  } catch (error) {
    chatBox.innerHTML += `<p><strong>Bot (error):</strong> ${error.message}</p>`;
  }

  document.getElementById("userInput").value = "";
  chatBox.scrollTop = chatBox.scrollHeight; // Auto scroll to bottom
 }
</script>

</body>
</html>
```

Bucket Policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::my-chatbot-ui/*"
    }
  ]
}
```

Step 6 : In API Gateway to the route created add CORS

Allowed Origins: http://chatbotinaws.s3-website.ap-south-1.amazonaws.com
Allowed Methods: POST
Allowed Headers: Content-Type

Save and Deploy again

Take the static website link and run