**Project : Chatbot Using Python**

**Phase 3 : Development Part 1**

To start building our chatbot with GPT-3 integration using the Transformers library and create a web app with Flask, we can follow these steps:

**Step 1: Set Up Your Environment**

Ensure we have Python installed on your system. If not, you can download and install it from the official Python website (https://www.python.org/downloads/).

Create a new Python virtual environment to manage your project dependencies. You can use a tool like `virtualenv` or `conda` to do this. Here's how you can create a virtual environment using `virtualenv`:

# Install virtualenv if you haven't already

pip install virtualenv

# Create a new virtual environment

virtualenv chatbot-env

# Activate the virtual environment (on Windows, use "chatbot-env\Scripts\activate")

source chatbot-env/bin/activate

**Step 2: Install Required Libraries**

With our virtual environment activated, we can install the necessary libraries. We'll need the Transformers library for GPT-3 integration and Flask for building the web app. We can also use `pip` for this: pip install transformers flask

**Step 3: Get an API Key for GPT-3**

We will need an API key to access GPT-3. You can get one from the OpenAI platform. We'll also need to install the OpenAI Python package and set up your API key:

pip install openai

Once you have our API key, you can set it as an environment variable. For example, in our Python code or in our terminal:

export OPENAI\_API\_KEY=your-api-key

**Step 4: Create the Flask web**

We can create a basic Flask app with routes to interact with your chatbot. Here's an example of a simple Flask app:

from flask import Flask, request, jsonify

import openai

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

# Your GPT-3 API key

openai.api\_key = "your-api-key-here"

@app.route('/chat', methods=['POST'])

def chat():

user\_input = request.form.get('user\_input')

# Use GPT-3 to generate a response

response = openai.Completion.create(

engine="text-davinci-002",

prompt=user\_input,

max\_tokens=50,

)

return jsonify({"response": response.choices[0].text})

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

This code sets up a Flask app with a `/chat` route, which takes user input and sends it to GPT-3 to generate a response. The response is then returned as JSON.

**Step 5: Run the Flask App**

Run your Flask app using the following command:

python your\_app.py

Your app should now be running, and you can access it in your browser or using a tool like `curl` or `requests` to interact with your chatbot.

Team Members :

ROHAN SAMUEL J

C. MONESH KUMAR

P.K.M HARICHEAN

A. JEEVA

M. NITHISH KRISHNAN