WhatsApp Bot Using

ChatGPT API

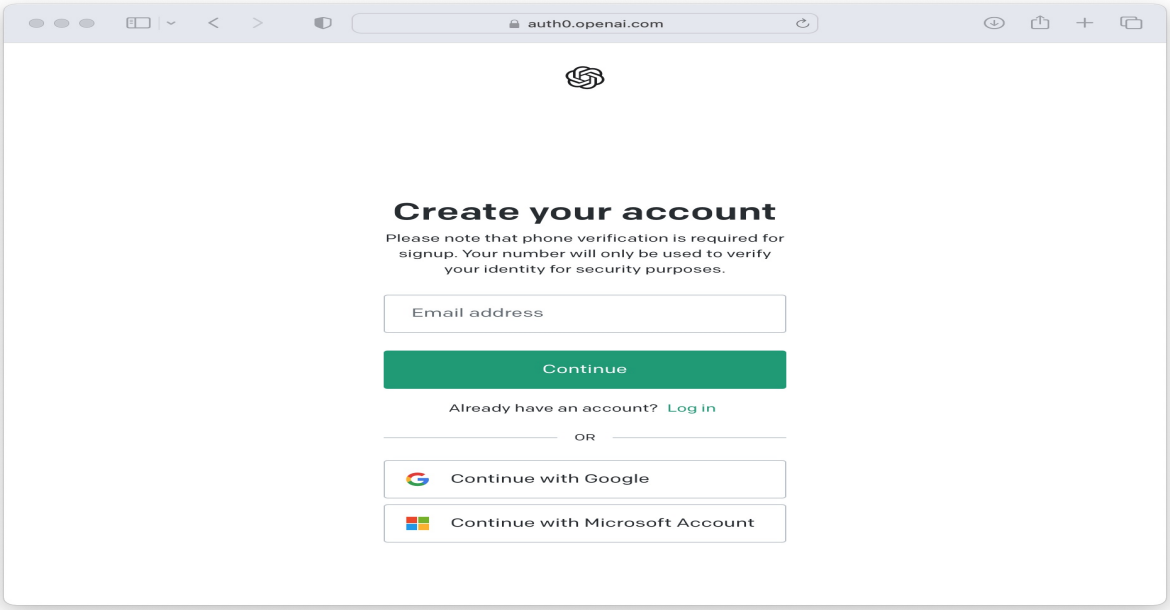
1. Abstract

This document outlines the high-level design (HLD) for a WhatsApp bot project that connects Twilio and OpenAI to provide automated answers to user questions using the OpenAI GPT-3 language model. The project is implemented in Python and utilizes the Flask framework. The bot is capable of handling text messages only.

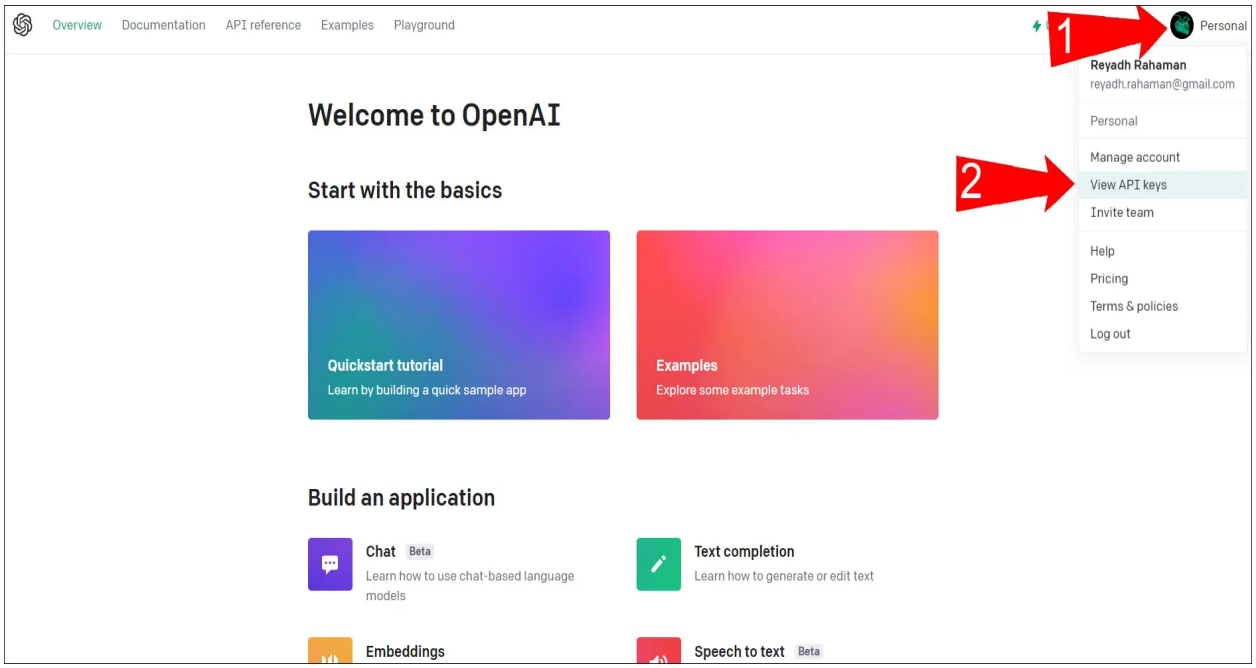
To use this project, several prerequisites are necessary. First, an OpenAI API key is required, which can be obtained by creating an account on the OpenAI website. Additionally, a Twilio project is needed, and the Account SID and Auth Token for that project must be obtained. An API requesting applications such as Postman or Insomnia is also required, along with NGROK for local testing.

2. How to obtain open_api_key?

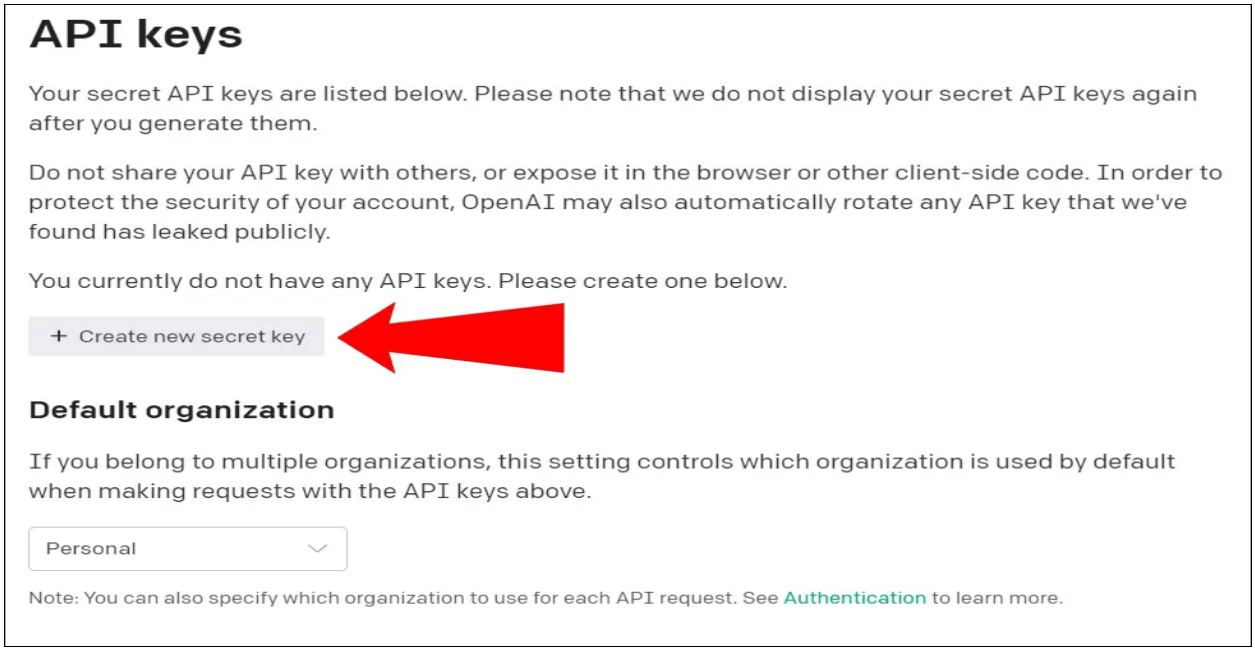
2.1 Sign in or sign up for your OpenAI account Link: https://platform.openai.com/



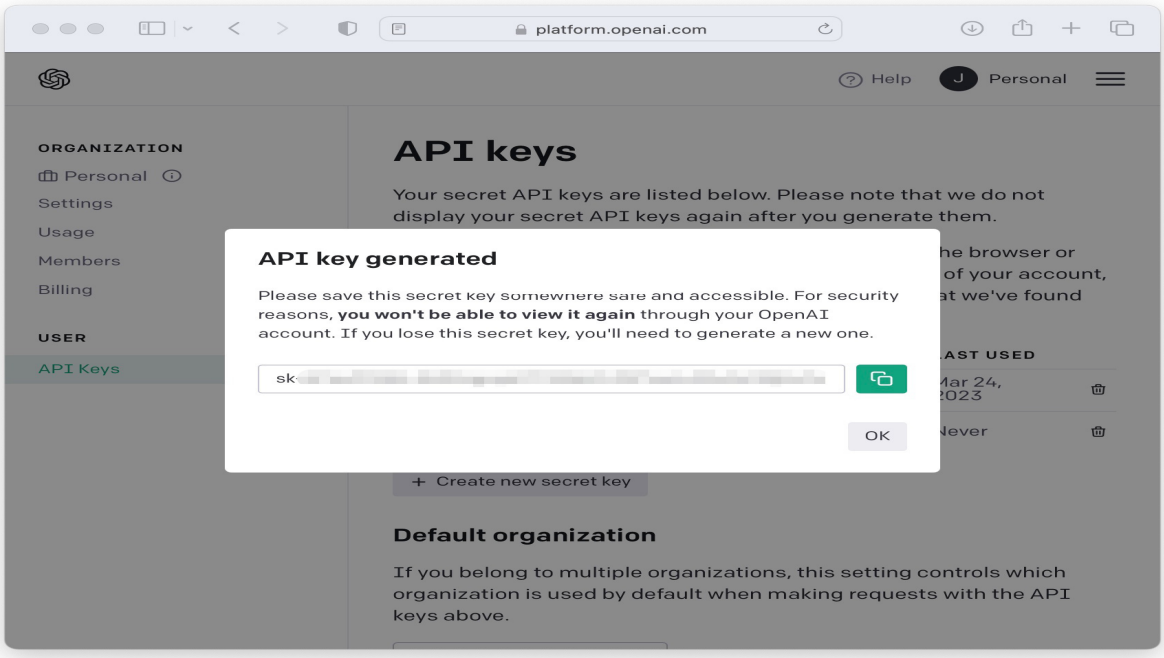
2.2 Click on your name in the top-right corner to bring up a dropdown menu. Then, click the “View API keys” option.



2.3 Click on create new secret key button

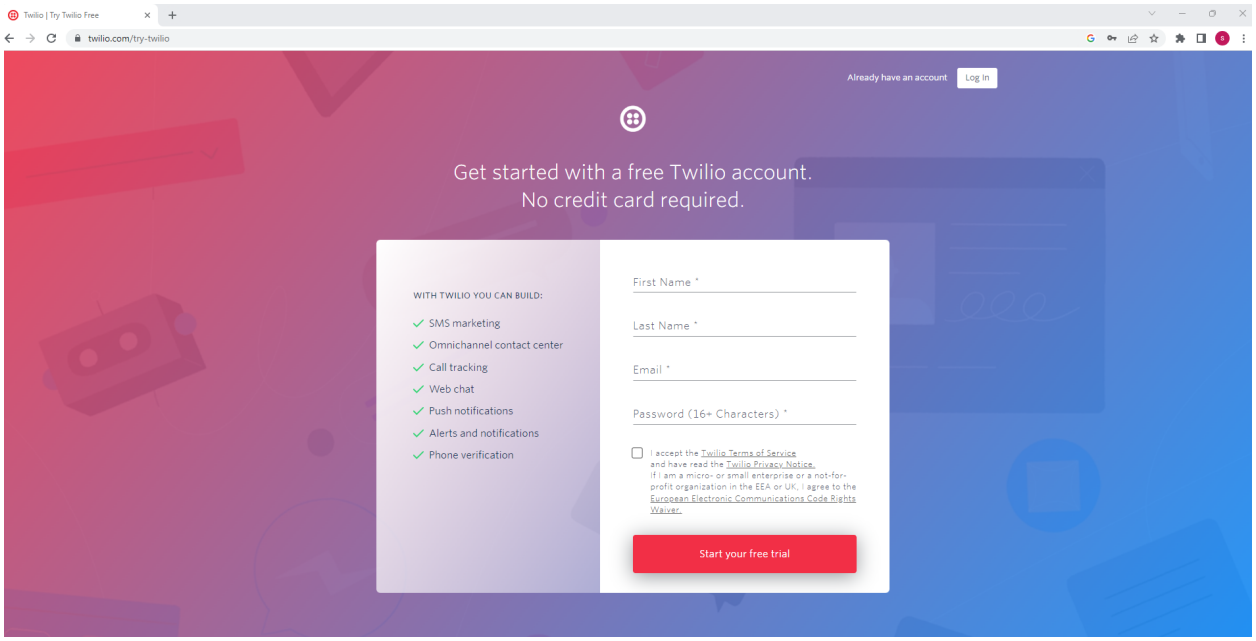


2.4 Then click the green icon on the popup page to copy your OpenAI secret key to a safe place

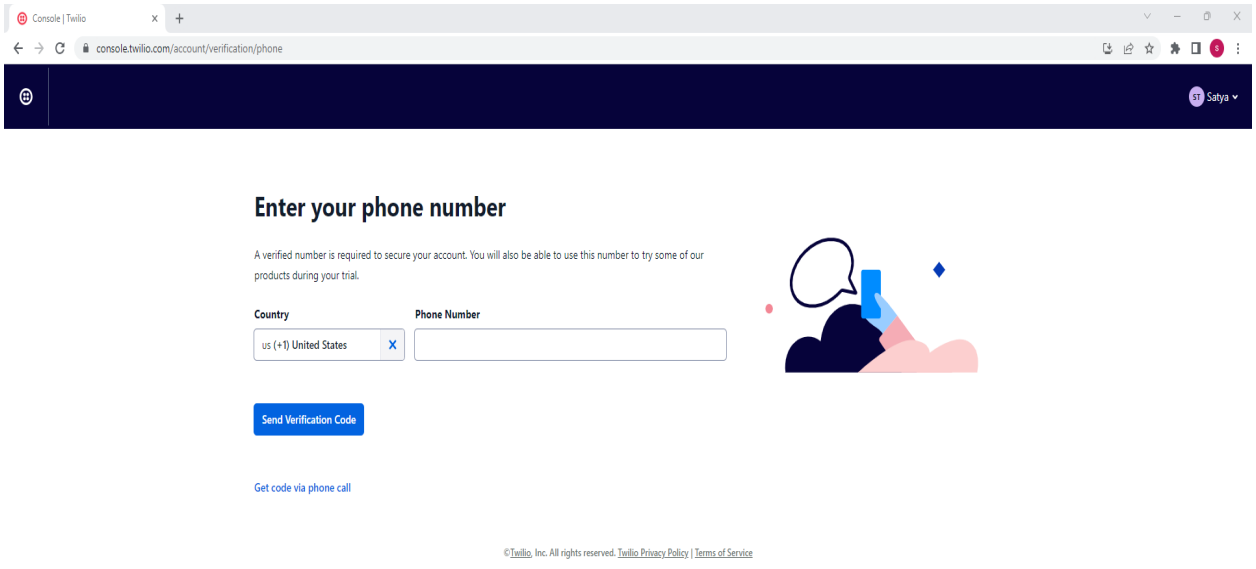


3. How to create a Twilio account?

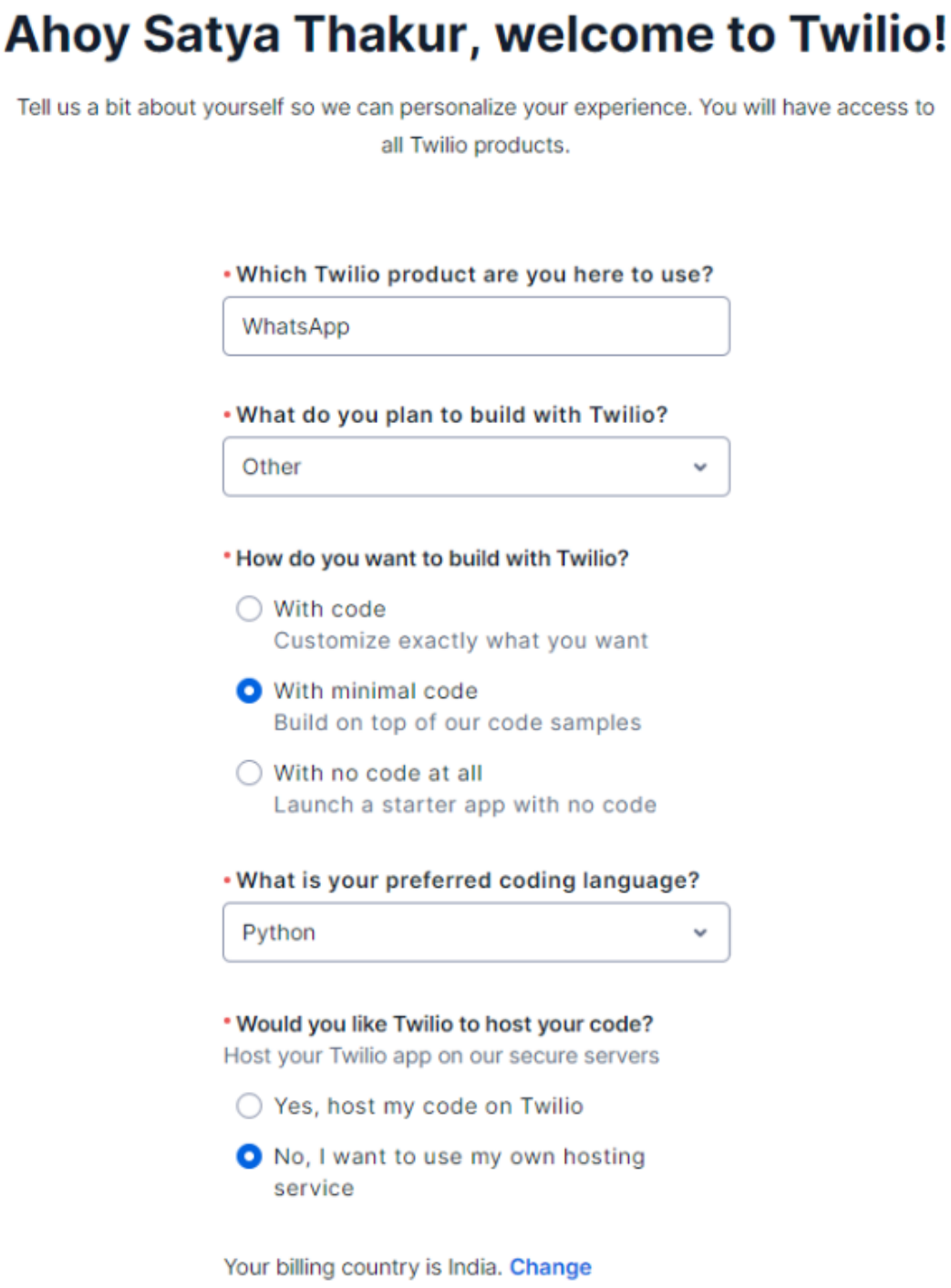
3.1 Create account for Twilio using link https://www.twilio.com/try-twilio



3.2 Input your phone number for authentication purposes, click on Send Verification Code button. You will receive a validation code you need to verify withTwilio.



3.3 Once your verification code is verified. You need to input the form below with filled details.



4. How to Set up the Twilio WhatsApp Sandbox?

Twilio WhatsApp Sandbox is a feature provided by Twilio, a cloud communications platform, that allows developers to test and prototype their WhatsApp integration using Twilio's APIs in a controlled environment. It provides a virtual testing environment where you can send and receive WhatsApp messages without going through the full approval process from WhatsApp.

The Twilio WhatsApp Sandbox is a development environment provided by Twilio that allows developers to test and prototype WhatsApp-integrated applications. It provides a simulated environment for sending and receiving messages via WhatsApp without the developer spending money or needing her real WhatsApp account.
Key features of the Twilio WhatsApp sandbox are:

Testing and Prototyping:

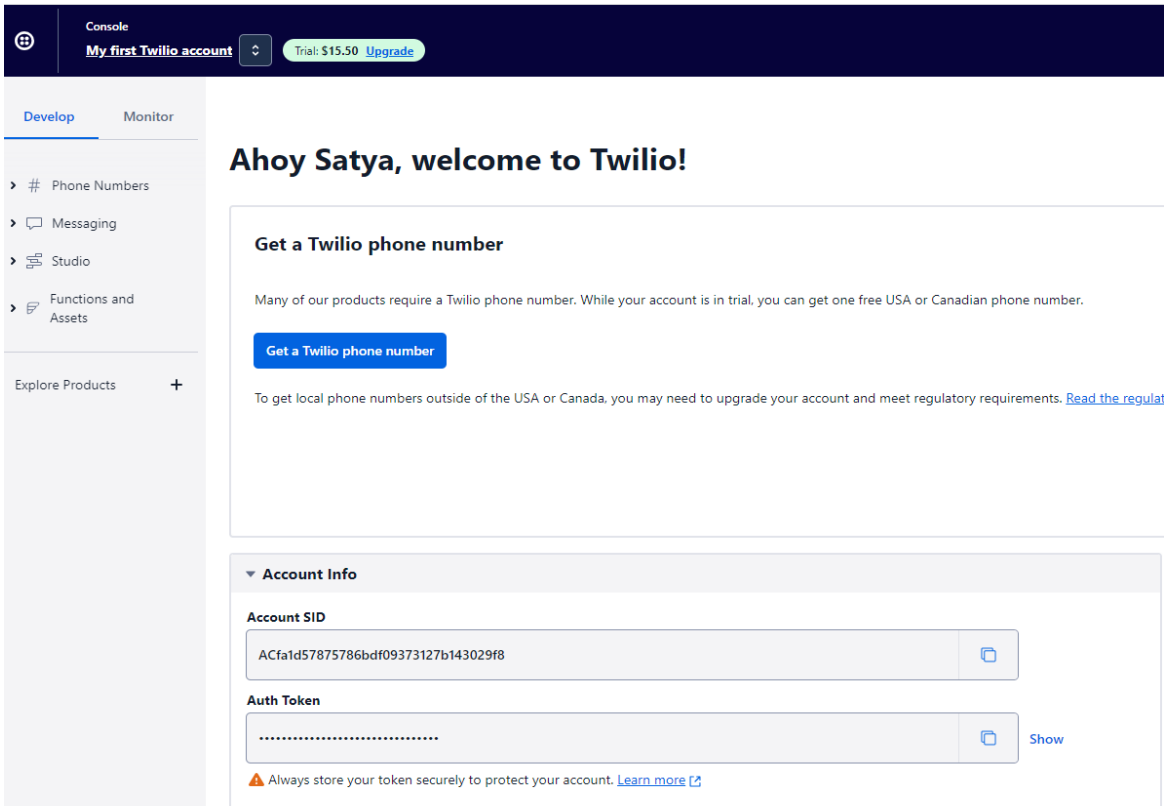
The sandbox allows developers to experiment and build applications that use her Twilio API to interact with her WhatsApp. With it, you can send and receive text girlfriend messages, images and other media files through WhatsApp.
Virtual Phone Number: Twilio provides a virtual phone number dedicated to the WhatsApp sandbox. These numbers are associated with his Twilio account and act as senders and receivers of WhatsApp messages in a sandbox environment. Predefined Templates: The sandbox includes predefined templates for sending messages in various scenarios, such as: B. Account Verification, Order Updates, Reservation Reminders. These templates help developers get up and running quickly and test common use cases.

Webhooks and Callbacks:
Twilio provides her webhooks and callbacks to receive incoming messages and deliver them to your application. You can configure your application's endpoints to receive, process, and respond to messages.
Usage Limits: The WhatsApp sandbox has limits on the number of messages you can send and receive. It is primarily designed for testing and development purposes and its use is limited to a small number of WhatsApp phone numbers.
The Twilio WhatsApp Sandbox is a valuable tool for developers looking to integrate WhatsApp functionality into their applications. This allows you to test and validate your code, experiment with different features, and make sure your application works as expected before moving it to production.

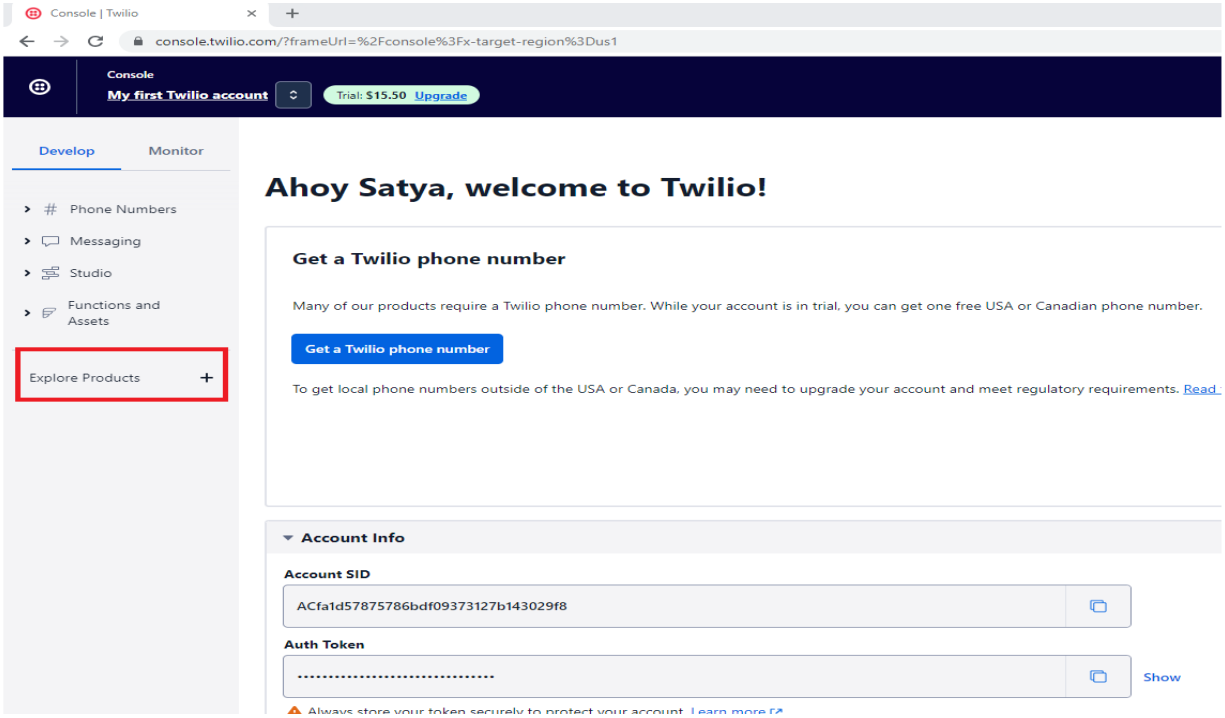
4.1 To activate the sandbox the dialogue box you need to click on the confirm button.

By leveraging the Twilio WhatsApp Sandbox, developers can efficiently build, test, and iterate on their WhatsApp integration without needing immediate access to WhatsApp's production environment. It provides a safe and controlled space to ensure the smooth functioning of your WhatsApp-enabled applications before going live.

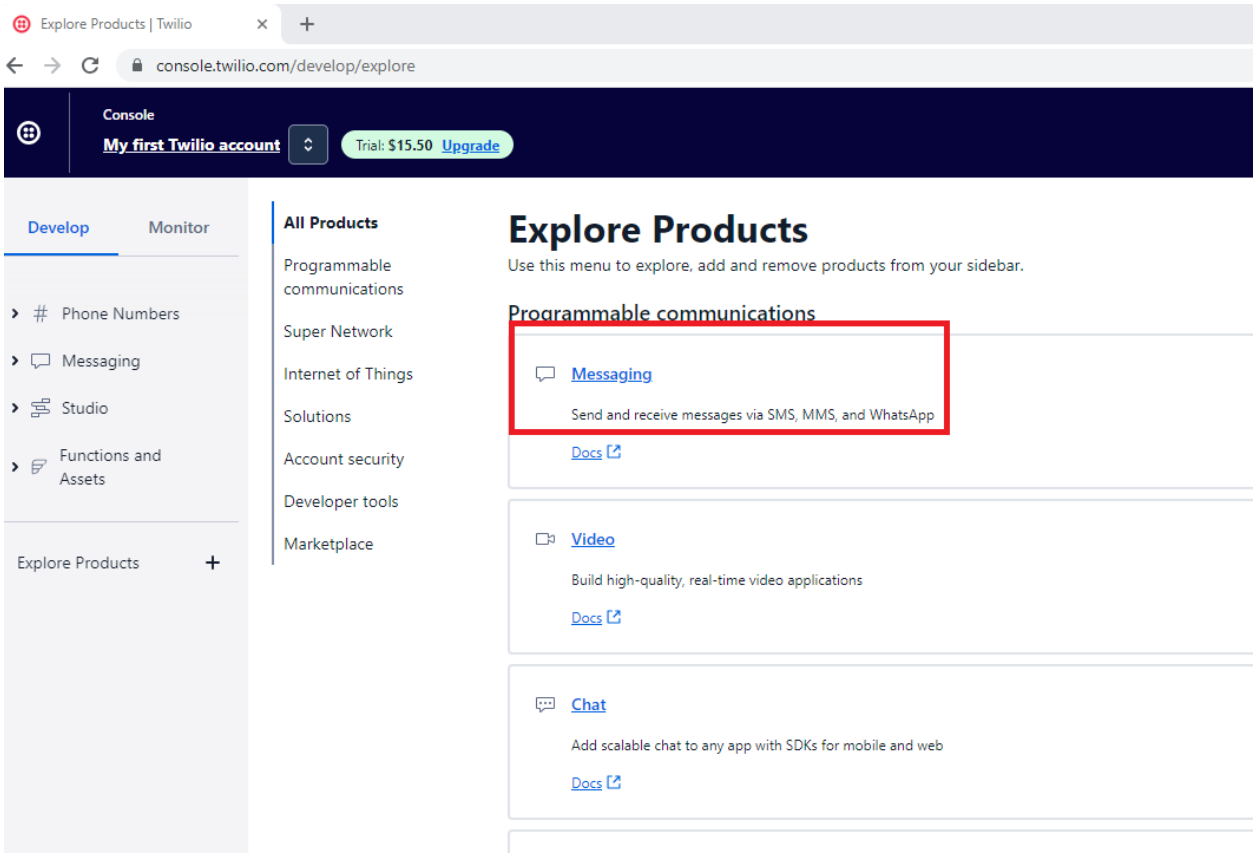
4.2 Now you need to open console using link: https://console.twilio.com/



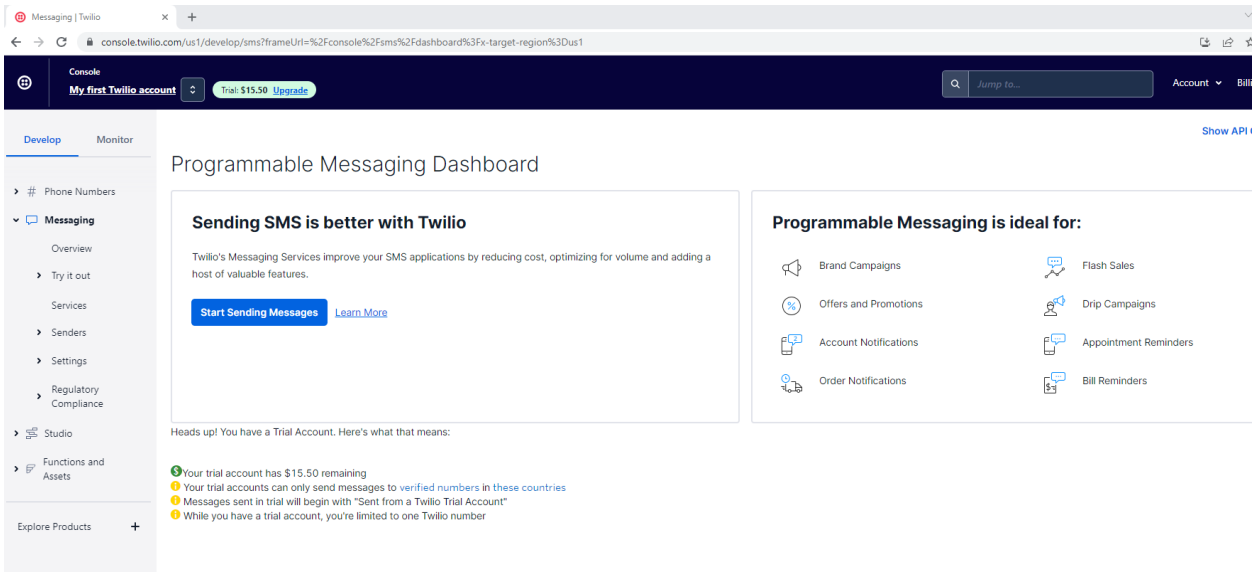
4.3 Now click on Explore Products in the left sidebar



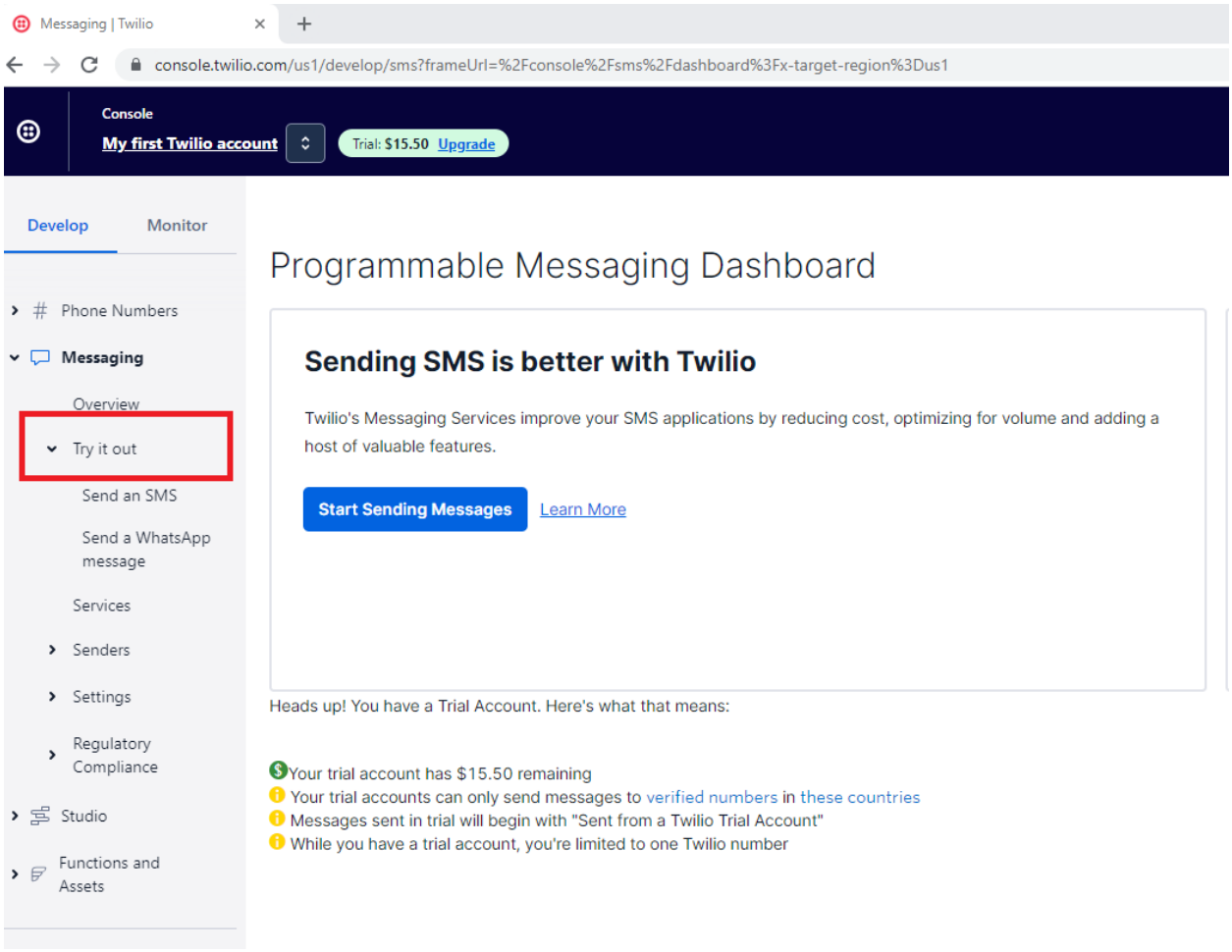
4.4 Now you need to select first option ie. Messaging



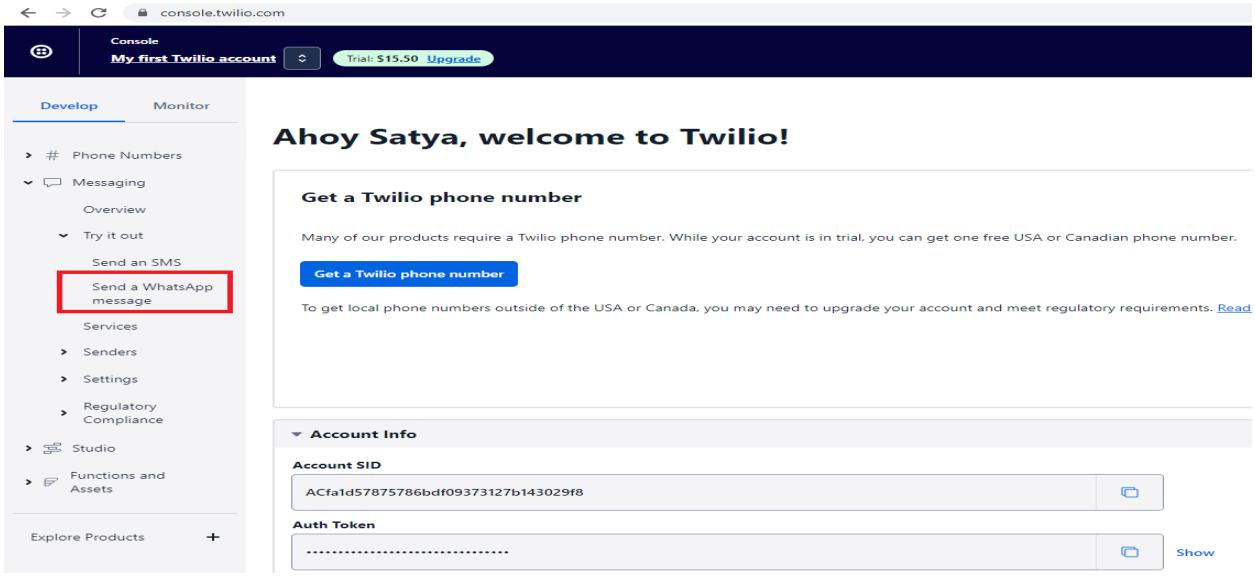
4.5 Messaging dashboard looks something like below



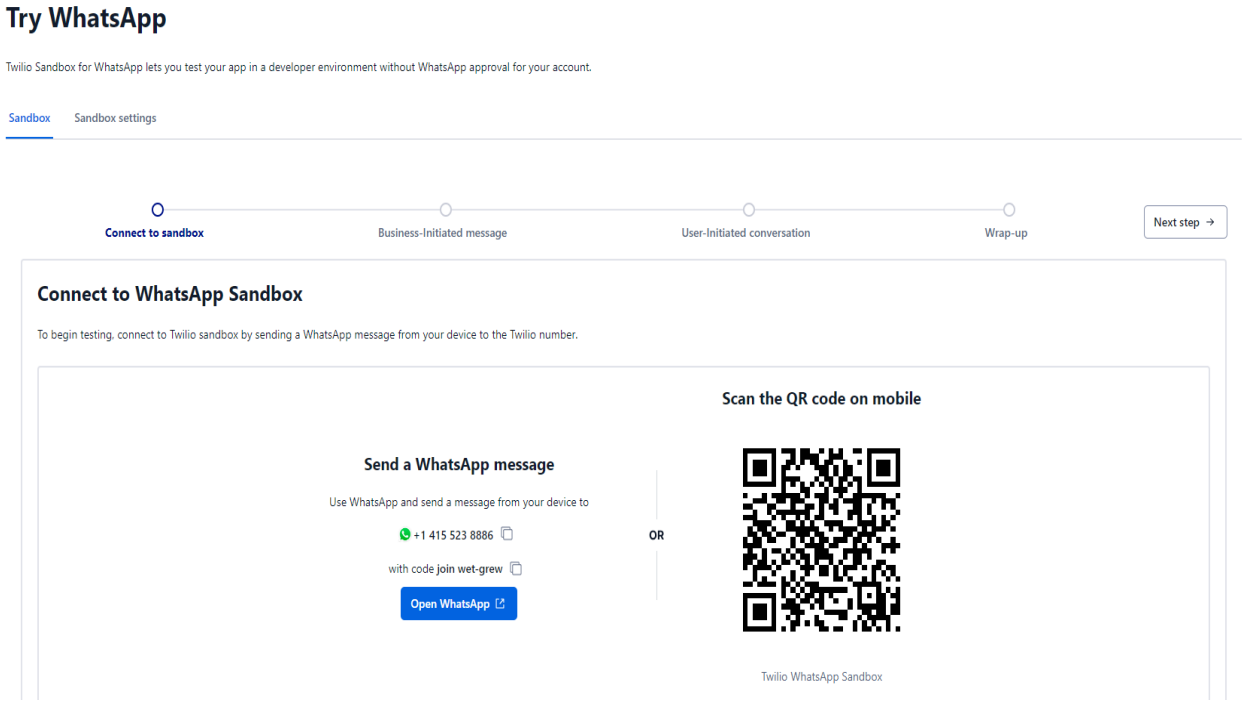
4.6 From Messaging sidebar click on Try it Out



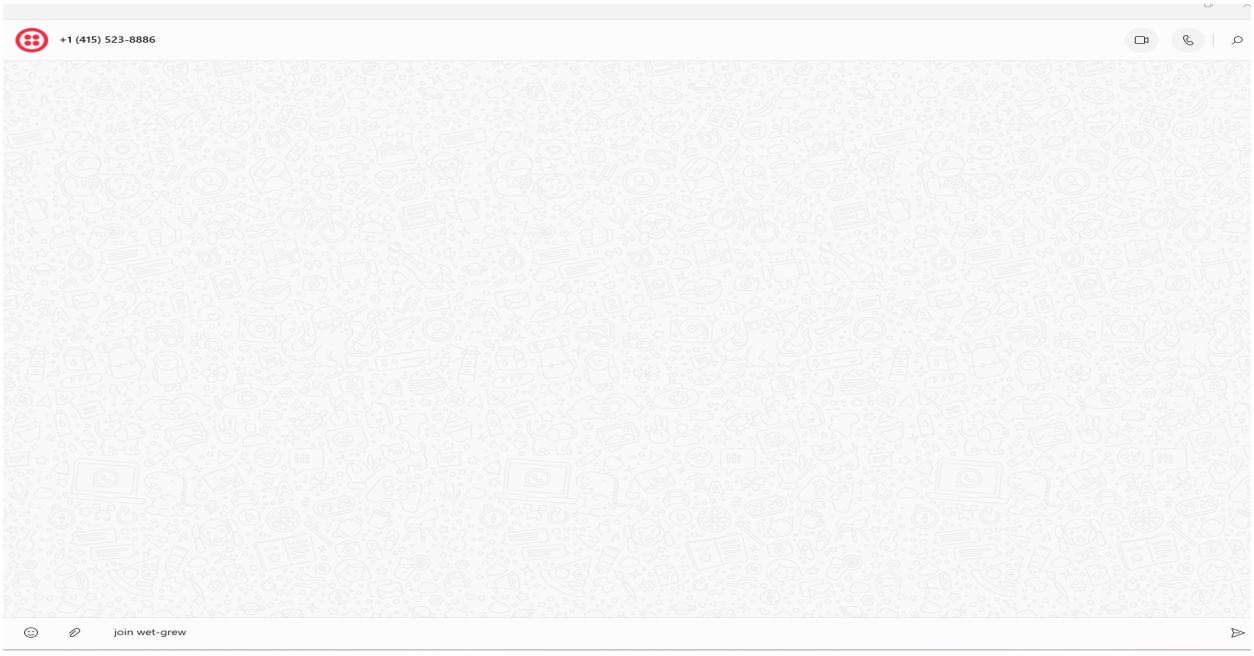
4.7 Click on Send a Whatsapp message from left sidebar



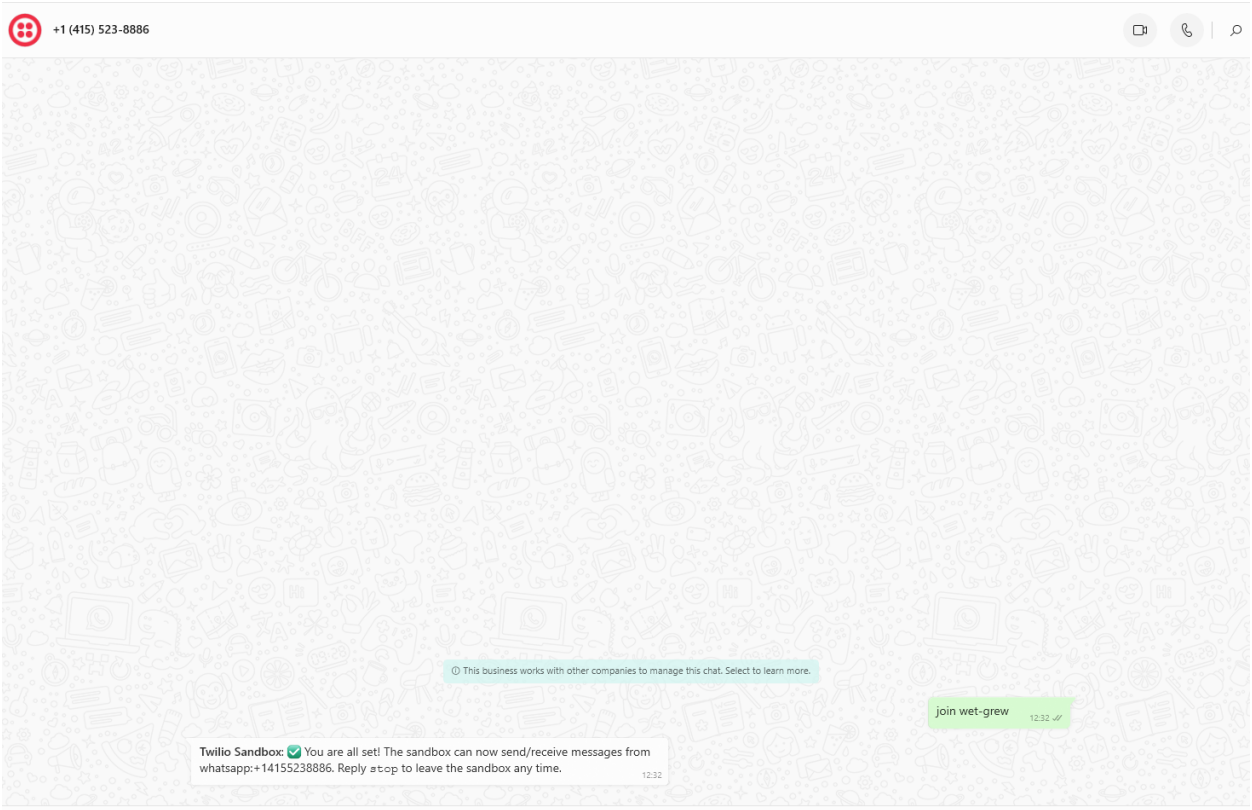
4.8 Save the WhatsApp number and code for your reference



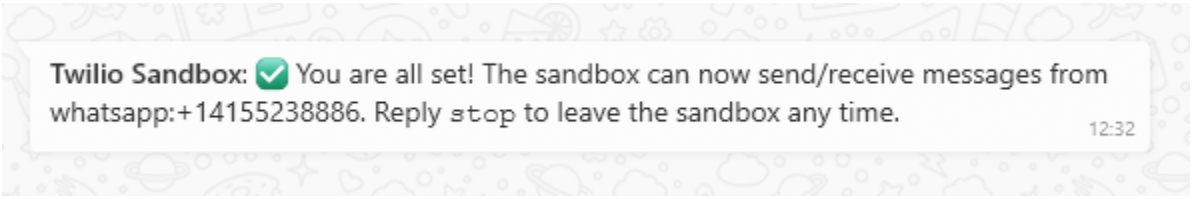
4.9 Open your WhatsApp using the same number you have copied in previous step



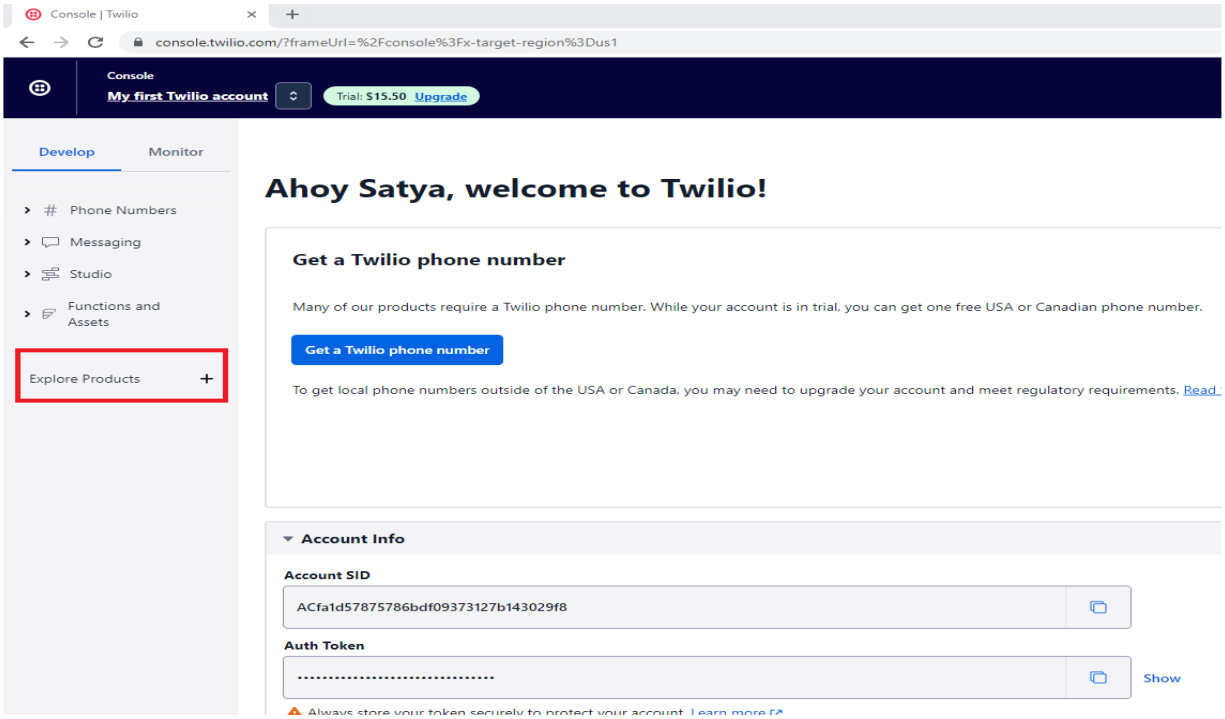
4.10 Then send your code that you have copied in previous steps.



4.11 Then you will receive reply something like below



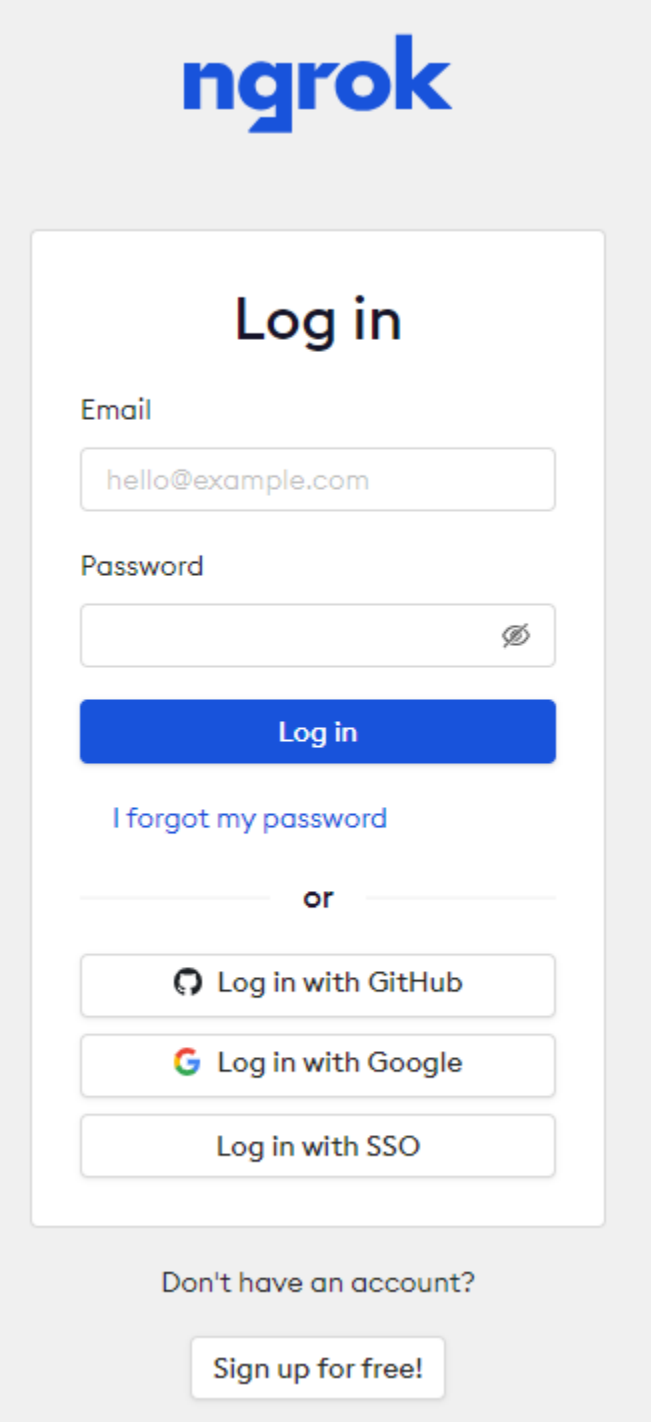
4.12 Open console using link: https://console.twilio.com/



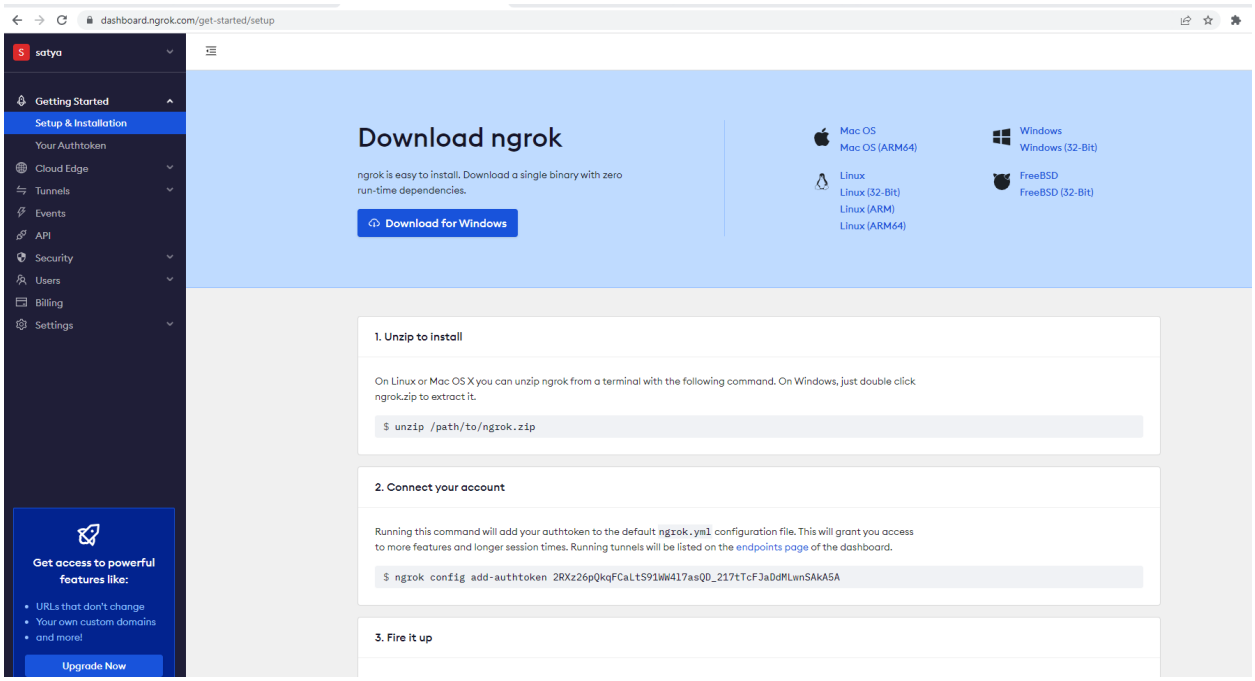
Save `TWILIO_ACCOUNT_SID and TWILIO_AUTH_TOKEN` for your reference.

5. NGROK for local testing

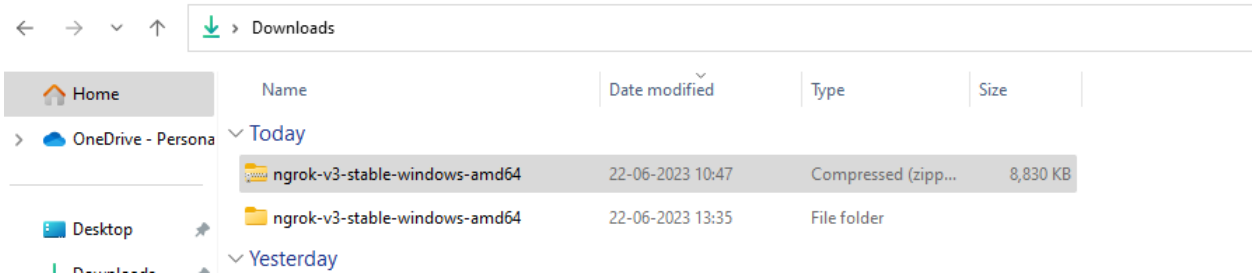
5.1 Create account for ngrok using link https://dashboard.ngrok.com/login



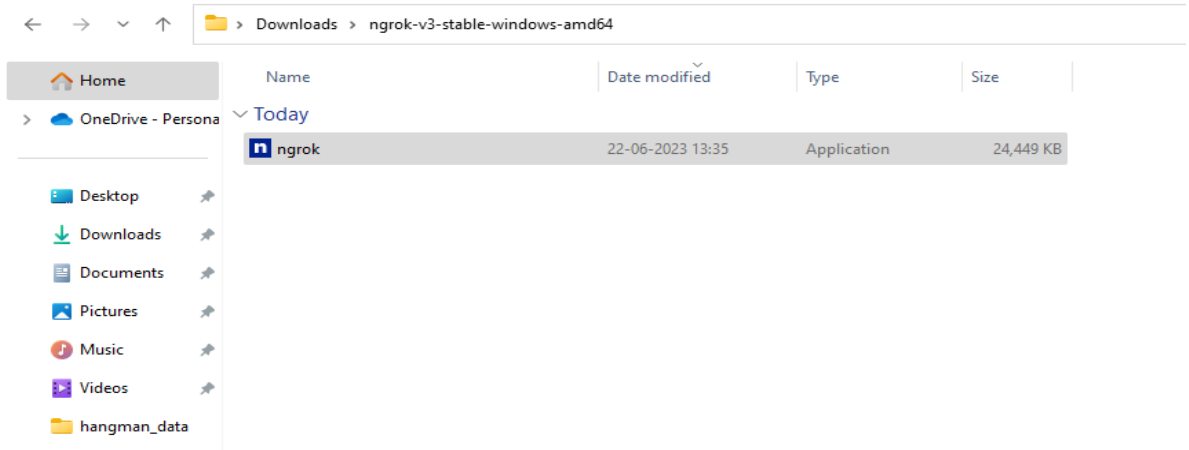
5.2 Once you logged in, then you can download software from Download button



5.3 Extract the downloaded folder



5.4 Open the extracted folder



6 . Basic Project Setup

6.1 Export environment variables.

After above setup done successfully, we can export environment using `python-dotenv`

`dotenv` loads the environment variable by reading them from a `.env` file which needs to be inside the project directory.

The `.env` file has declarations in the form of key-value pairs separated by =, following is an example of the contents of a `.env` file.

```
TWILIO_ACCOUNT_SID = ABC123

TWILIO_AUTH_TOKEN = SUPERSECRET123

OPENAI_API_KEY = ABC123

FROM=whatsapp:+14155238886
```

6.2 Loading the .env file

```
from dotenv import load_dotenv

load_dotenv()
```

6.3 Run run.py file and bot will be ready for the conversation.

```
python run.py
```

6.4 Install all the libraries from `requirements.txt` file.

```
pip install -r requirements.txt
```

Details of each requirement you can find in below table

LIBRARY NAME	DESCRIPTION
`openai`	It iis a Python library provided by OpenAI. It allows you to interact with OpenAI's APIs, including their language models.
`Twilio`	A Python library for integrating with Twilio APIs.
`Flask`	A Python web framework for building web applications
`python-dotenv==0.21.0`	This library allows you to load environment variables from a .env file into your Python script.

7 . Interacting openAI to generate chatbot responses

The `text_resources` function takes a prompt as input, interacts with OpenAI's language model, and returns a dictionary containing a status code and the generated response. If the interaction with the language model is successful, the status code is set to 1, and the generated response is included in the dictionary. If an exception occurs during the interaction, the status code is set to 0, and an empty string is returned as the response.

```
import os
import openai from dotenv
import load_dotenv
load_dotenv()
```

This line sets the API key for the OpenAI library by accessing the value of the environment variable named `CHATGPT_API_KEY`. The API key is required to authenticate and make requests to the OpenAI API.

```
openai.api_key = os.getenv('CHATGPT_API_KEY')
```

This function takes a string parameter prompt and returns a dictionary. The function is designed to interact with OpenAI's language model and generate a response based on the given prompt.

```
def text_resources(prompt: str) -> dict:
    Try:
```

This line makes a request to the OpenAI API by calling the `Completion`.create method. It provides various parameters to customize the behavior of the language model, such as the model to use (`text-davinci-003`), the prompt to start with (`'Human: {prompt}\nAI: '`), temperature, max tokens, and penalties for frequency and presence.

```
        response = openai.Completion.create(
            model='text-davinci-003',
            prompt=f'Human: {prompt}\nAI: ',
            temperature=0.9,
            max_tokens=150,
            top_p=1,
            frequency_penalty=0,
            presence_penalty=0.6,
            stop=['Human:', 'AI:']
        )
```

return {`'status': 1, 'response': response['choices'][0]['text']`}: If the API call is successful, this line returns a dictionary with a status key set to `1 (indicating success)` and a response key set to the generated text from the API response.

```
        return {
            'status': 1,
            'response': response['choices'][0]['text']
        }
    except:
        return {
            'status': 0,
            'response': ''}
```

8. Use Twilio to send a WhatsApp message

The code sets up the Twilio client with the account SID and authentication token, defines a function to send a WhatsApp message, and then uses the Twilio client to send the message by calling the `client.messages.create()` method with the necessary parameters.

```
import os

from twilio.rest import Client

from dotenv import load_dotenv

load_dotenv()

account_sid = os.getenv('TWILIO_ACCOUNT_SID')

auth_token = os.getenv('TWILIO_AUTH_TOKEN')

client = Client(account_sid, auth_token)

def send_message(to: str, message: str) -> None:

    _ = client.messages.create(

        from_=os.getenv('FROM'),

        body=message,

        to=to

    )
```

The function takes two parameters: `to` and `message`. The parameter is the WhatsApp number of the recipient, in the format `whatsapp:+09876543215`. The `message` parameter is the text message that you want to send.

The function first imports the `twilio.rest` and `dotenv` modules. The `twilio.rest` module provides a Python interface to the Twilio API, which is used to send WhatsApp messages. The `dotenv` module is used to load environment variables from a `.env` file.

The function then gets the `account_sid` and `auth_token` environment variables,

which are used to authenticate with the Twilio API. It then creates a `Client` object, which is used to interact with the API.

The function then calls the `messages.create` method on the `Client` object. This method sends a WhatsApp message to the recipient specified by the `to` parameter. The `message` parameter is passed as the `body` argument to the `messages.create` method.

9. Flask Application

Flask application with two routes (`/ and /twilio/receiveMessage`) and defines corresponding functions to handle the requests.

```
from components.chatgpt_api import text_resources

from components.twilio_api import send_message

from flask import Flask, request
```

```
from dotenv import load_dotenv

load_dotenv()

app = Flask(__name__)


@app.route('/')

def home():

    return 'Processing successfully'


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

def receiveMessage():

    try:

        # Extract incoming parameters from Twilio

        message = request.form['Body']

        sender_id = request.form['From']

        # Get response from Openai

        result = text_resources(message)

        if result['status'] == 1:

            send_message(sender_id, result['response'])

    except:

        pass

    return 'OK', 200
```

Flask application that can be used to receive WhatsApp messages and respond to them with text generated by the ChatGPT API.

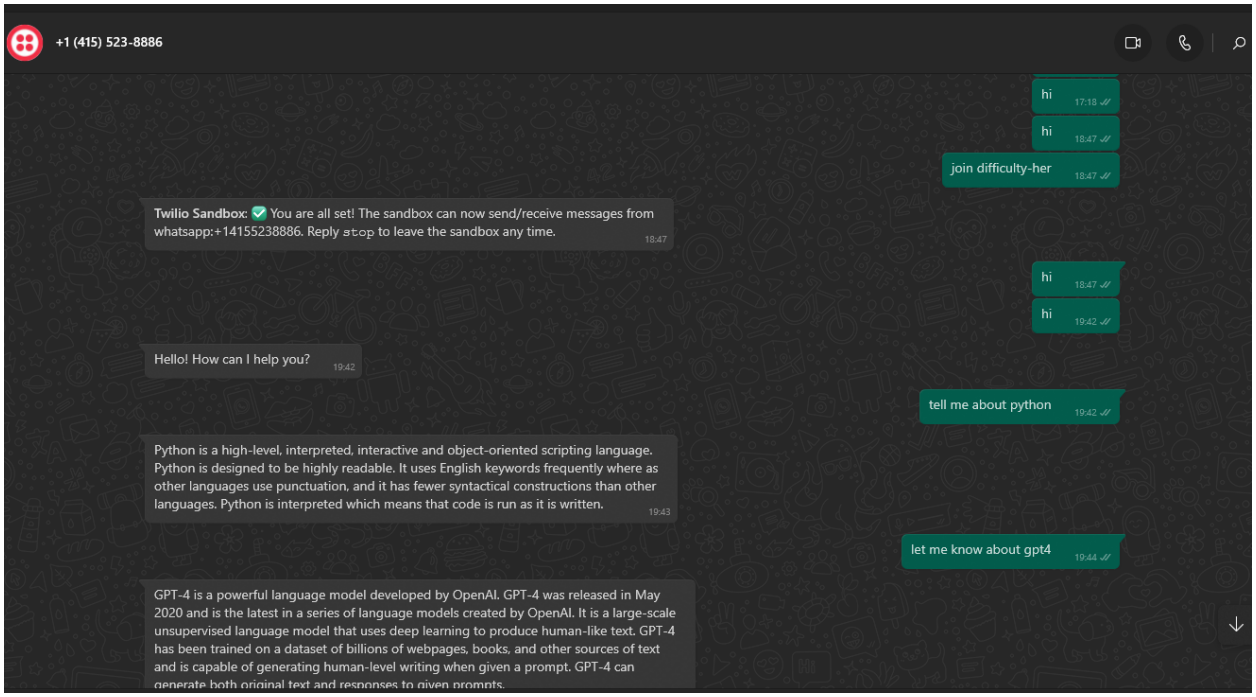
The `receiveMessage` route first extracts the incoming parameters from the request. The `Body` parameter contains the text of the message, and the From parameter contains the WhatsApp number of the sender.

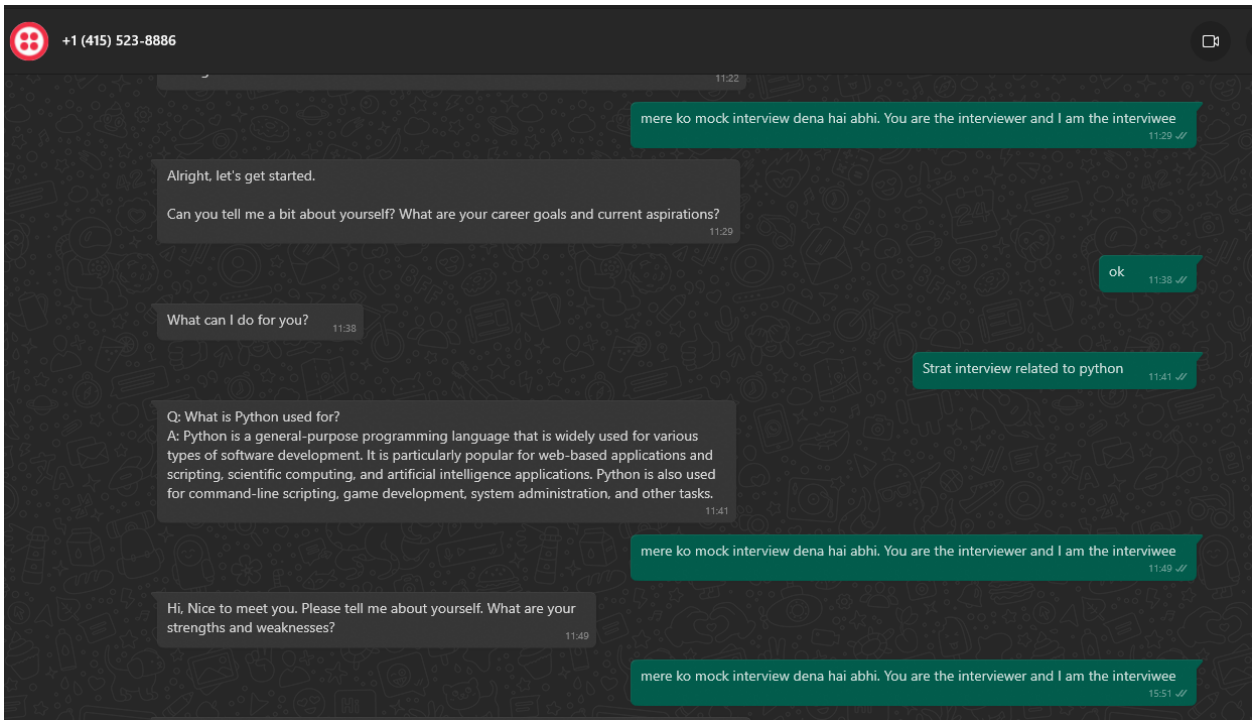
The `receiveMessage` route then calls the `text_resources` function to get a response from the ChatGPT API. The `text_resources` function takes the text of the message as input and returns a response object. The response object contains the status of the request and the generated text.

If the status of the request is 1, then the `receiveMessage` route calls the `send_message` function to send the generated text back to the sender

The `receiveMessage` route finally returns the text "OK" with a status code of 200. This indicates that the request was successful.

10. Results


+1 (415) 523-8886
hi 17:18
hi 18:47
join difficulty-her 18:47
Twilio Sandbox: You are all set! The sandbox can now send/receive messages from whatsapp:+14155238886. Reply stop to leave the sandbox any time. 18:47
hi 18:47
hi 19:42
Hello! How can I help you? 19:42
tell me about python 19:42
Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages. Python is interpreted which means that code is run as it is written. 19:43
let me know about gpt4 19:44
GPT-4 is a powerful language model developed by OpenAI. GPT-4 was released in May 2020 and is the latest in a series of language models created by OpenAI. It is a large-scale unsupervised language model that uses deep learning to produce human-like text. GPT-4 has been trained on a dataset of billions of webpages, books, and other sources of text and is capable of generating human-level writing when given a prompt. GPT-4 can generate both original text and responses to given prompts.


+1 (415) 523-8886
11:22
mere ko mock interview dena hai abhi. You are the interviewer and I am the interviwee 11:29
Alright, let's get started.
Can you tell me a bit about yourself? What are your career goals and current aspirations? 11:29
ok 11:38
What can I do for you? 11:38
Strat interview related to python 11:41
Q: What is Python used for?
A: Python is a general-purpose programming language that is widely used for various types of software development. It is particularly popular for web-based applications and scripting, scientific computing, and artificial intelligence applications. Python is also used for command-line scripting, game development, system administration, and other tasks. 11:41
mere ko mock interview dena hai abhi. You are the interviewer and I am the interviwee 11:49
Hi, Nice to meet you. Please tell me about yourself. What are your strengths and weaknesses? 11:49
mere ko mock interview dena hai abhi. You are the interviewer and I am the interviwee 15:51