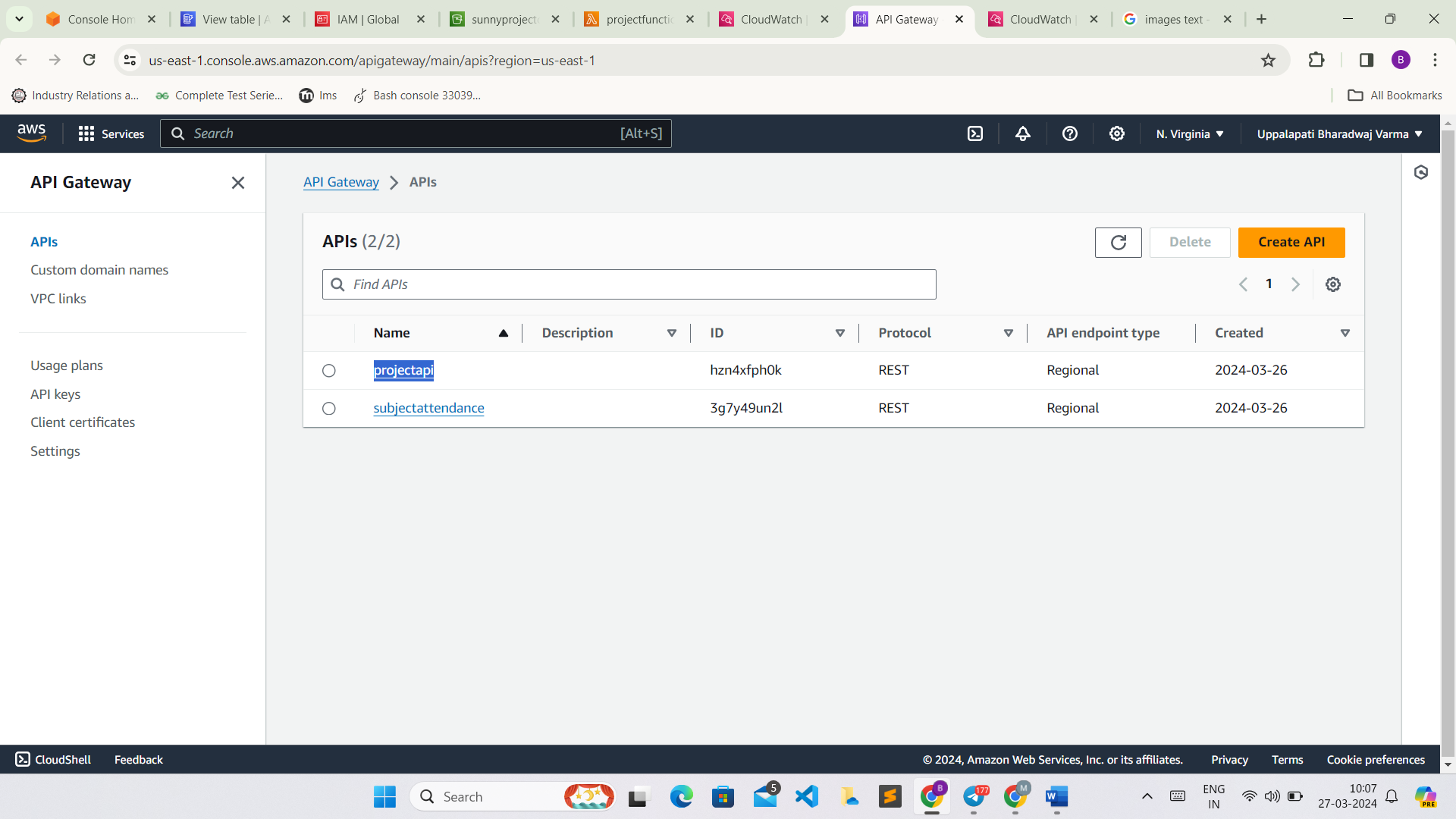
Develop a Telegram bot for converting images to text.

2100031737

Bharadwaj Varma

1.create a new API from API gateway



2.go to DynamoDB and create a table

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

3.Navigate to S3 and create a bucket with a unique name

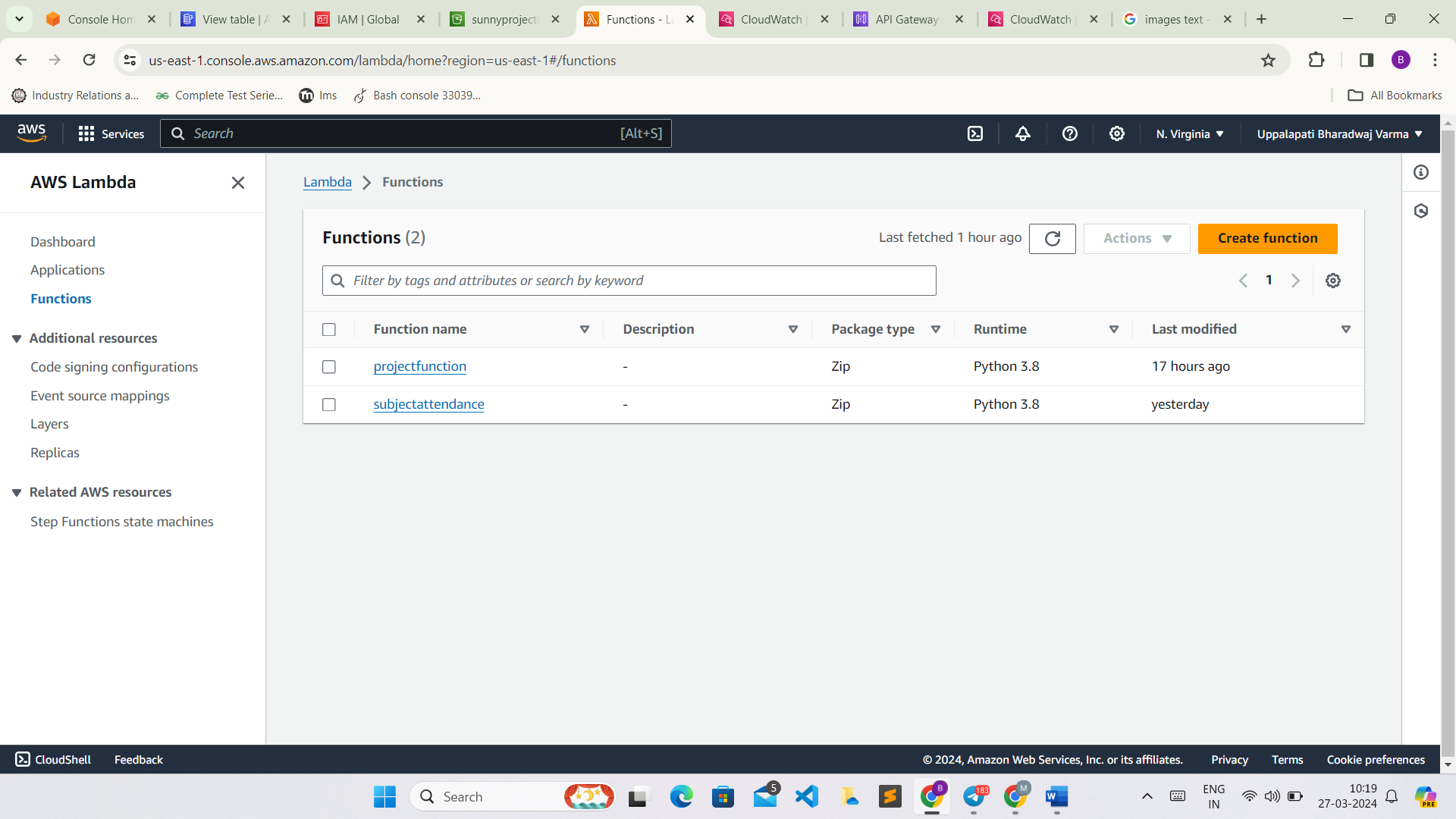
A screenshot of a computer

Description automatically generated

A screenshot of a computer

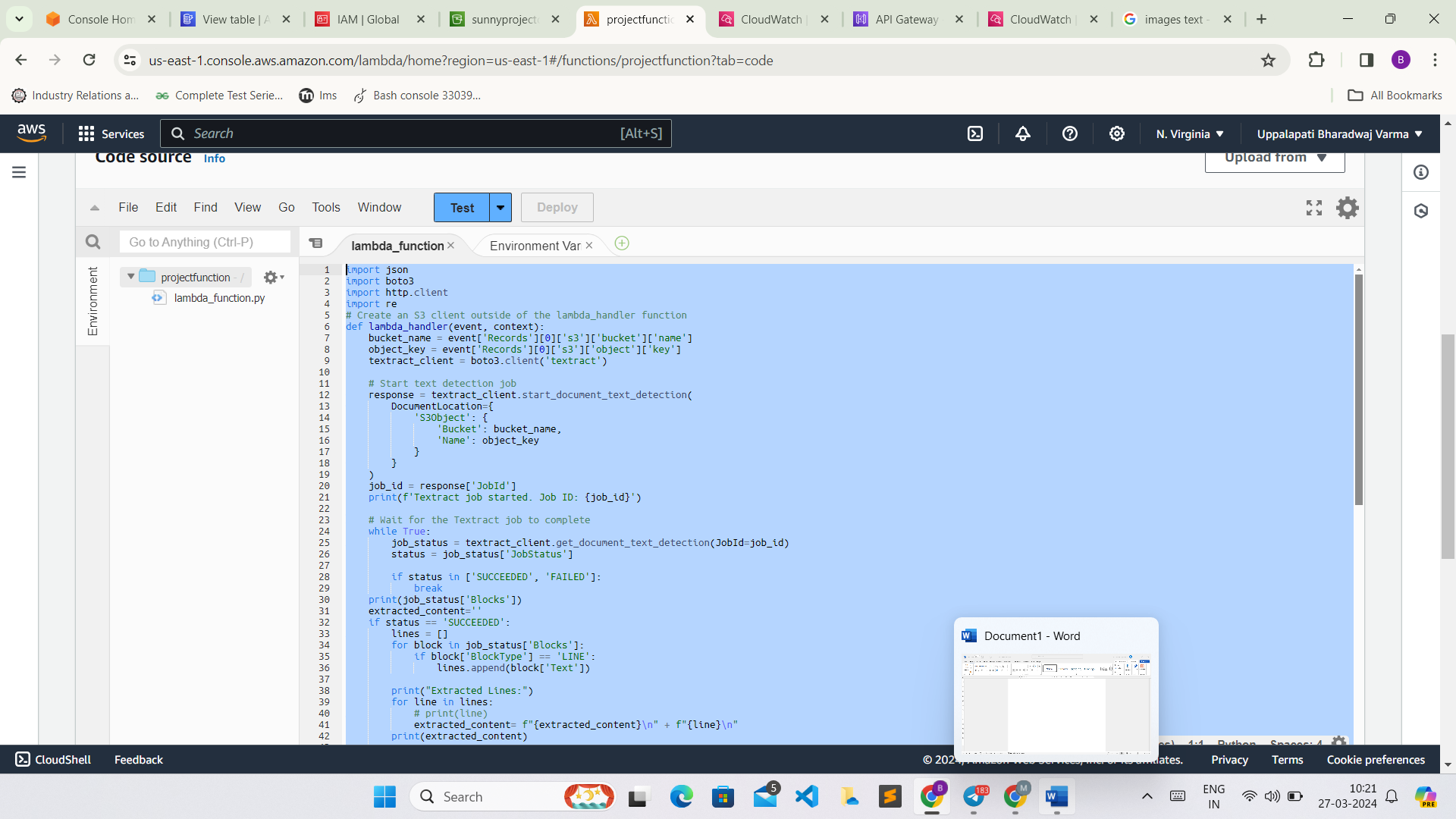
Description automatically generated

4.create a lambda function



A screenshot of a computer

Description automatically generated



CODE:

import json

import boto3

import http.client

import re

# Create an S3 client outside of the lambda\_handler function

def lambda\_handler(event, context):

bucket\_name = event['Records'][0]['s3']['bucket']['name']

object\_key = event['Records'][0]['s3']['object']['key']

textract\_client = boto3.client('textract')

# Start text detection job

response = textract\_client.start\_document\_text\_detection(

DocumentLocation={

'S3Object': {

'Bucket': bucket\_name,

'Name': object\_key

}

}

)

job\_id = response['JobId']

print(f'Textract job started. Job ID: {job\_id}')

# Wait for the Textract job to complete

while True:

job\_status = textract\_client.get\_document\_text\_detection(JobId=job\_id)

status = job\_status['JobStatus']

if status in ['SUCCEEDED', 'FAILED']:

break

print(job\_status['Blocks'])

extracted\_content=''

if status == 'SUCCEEDED':

lines = []

for block in job\_status['Blocks']:

if block['BlockType'] == 'LINE':

lines.append(block['Text'])

print("Extracted Lines:")

for line in lines:

# print(line)

extracted\_content= f"{extracted\_content}\n" + f"{line}\n"

print(extracted\_content)

pattern = r'(\d+)---'

print(event)

print(object\_key)

# Search for the pattern in the input string

match = re.search(pattern, object\_key)

chat\_id = match.group(1)

dynamodb\_client = boto3.resource('dynamodb')

dynamo\_table = dynamodb\_client.Table('project-table')

data={

'id': object\_key,

'extracted\_text':extracted\_content

}

table\_name='project-table'

response = dynamo\_table.put\_item(Item=data)

response=send\_telegram\_message(chat\_id,extracted\_content)

return response

return "Failed to extract text. Try Again"

def send\_telegram\_message(chat\_id, text):

telegram\_bot\_token = "7147499817:AAFVheTtaPDg2slHpFkJO-v7huwhregkewo"

host = "api.telegram.org"

path = f"/bot{telegram\_bot\_token}/sendMessage"

connection = http.client.HTTPSConnection(host)

headers = {"Content-type": "application/json"}

payload = {

"chat\_id": chat\_id,

"text": text

}

connection.request("POST", path, body=json.dumps(payload), headers=headers)

response = connection.getresponse()

# Read and print the response

response\_data = response.read().decode("utf-8")

print(response\_data)

connection.close()

return response\_data

5. Go to PythonAnywhere and Sign and open Bash Console. Paste the following command and run to install the necessary packages to run the bot.

pip install python-telegram-bot==13.14

A screenshot of a computer

Description automatically generated

6.Open telegram Bot father and create a new telegram bot

A blue box with white text

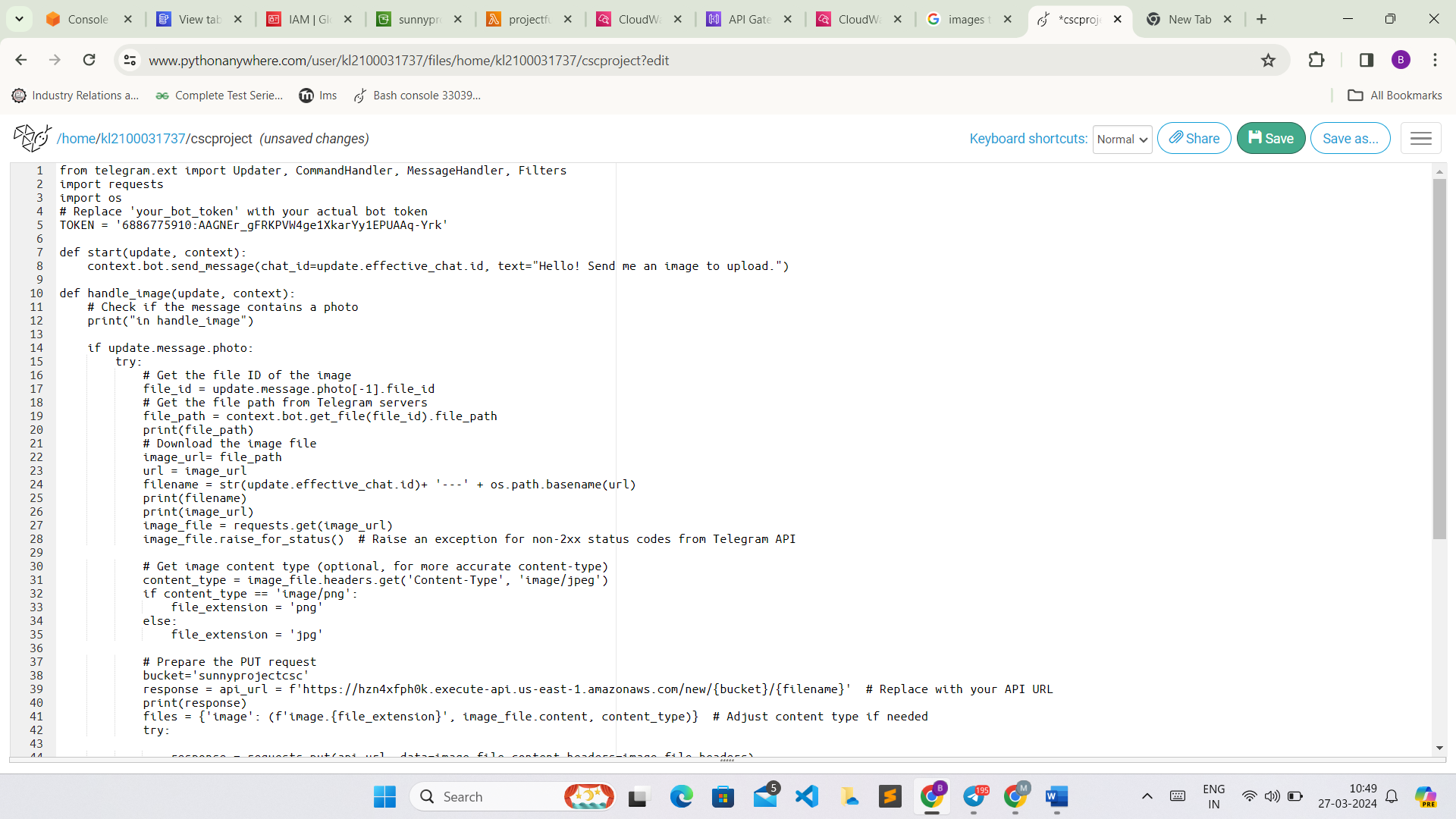
Description automatically generated

A screenshot of a chat

Description automatically generated

Do the following steps to create a bot

7.Copy token id and open a new .py file and push the code by changing the token id that is generated by your bot.



Code:

from telegram.ext import Updater, CommandHandler, MessageHandler, Filters

import requests

import os

# Replace 'your\_bot\_token' with your actual bot token

TOKEN = '6886775910:AAGNEr\_gFRKPVW4ge1XkarYy1EPUAAq-Yrk'

def start(update, context):

context.bot.send\_message(chat\_id=update.effective\_chat.id, text="Hello! Send me an image to upload.")

def handle\_image(update, context):

# Check if the message contains a photo

print("in handle\_image")

if update.message.photo:

try:

# Get the file ID of the image

file\_id = update.message.photo[-1].file\_id

# Get the file path from Telegram servers

file\_path = context.bot.get\_file(file\_id).file\_path

print(file\_path)

# Download the image file

image\_url= file\_path

url = image\_url

filename = str(update.effective\_chat.id)+ '---' + os.path.basename(url)

print(filename)

print(image\_url)

image\_file = requests.get(image\_url)

image\_file.raise\_for\_status() # Raise an exception for non-2xx status codes from Telegram API

# Get image content type (optional, for more accurate content-type)

content\_type = image\_file.headers.get('Content-Type', 'image/jpeg')

if content\_type == 'image/png':

file\_extension = 'png'

else:

file\_extension = 'jpg'

# Prepare the PUT request

bucket='sunnyprojectcsc'

response = api\_url = f'https://hzn4xfph0k.execute-api.us-east-1.amazonaws.com/new/{bucket}/{filename}' # Replace with your API URL

print(response)

files = {'image': (f'image.{file\_extension}', image\_file.content, content\_type)} # Adjust content type if needed

try:

response = requests.put(api\_url, data=image\_file.content,headers=image\_file.headers)

response.raise\_for\_status() # Raise an exception for non-2xx status codes

# Handle successful upload

context.bot.send\_message(chat\_id=update.effective\_chat.id, text='Image uploaded successfully!\n Please Wait! The image is getting processed....')

except requests.exceptions.RequestException as e:

# Handle errors from API request

context.bot.send\_message(chat\_id=update.effective\_chat.id, text=f'An error occurred while uploading the image: {e}')

except Exception as e: # Catch any other exceptions (e.g., from Telegram API calls)

context.bot.send\_message(chat\_id=update.effective\_chat.id, text=f'An unexpected error occurred: {e}')

else:

context.bot.send\_message(chat\_id=update.effective\_chat.id, text='Please send an image.')

def main():

updater = Updater(token=TOKEN, use\_context=True)

dp = updater.dispatcher

dp.add\_handler(CommandHandler("start", start))

dp.add\_handler(MessageHandler(Filters.photo, handle\_image)) # Use Filters.photo instead of filters.PHOTO

updater.start\_polling()

updater.idle()

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

main()

click run and open telegram then upload the image that contains text

A screenshot of a chat

Description automatically generated

Output:

A screenshot of a chat

Description automatically generated