

A Mini Project Report on

Text Extractor using Amazon AWS Services

Submitted in partial fulfillment of the requirements
of the degree of Bachelor in Engineering

Third Year of Engineering In
Computer Engineering

By

Aditya Sawant	TE4	45
Swapnil Sawant	TE4	46
Chinmay Shinde	TE4	50

Under the guidance of
Prof. Krupa Chotai



Department of Computer Engineering
Shah and Anchor Kutchhi Engineering College
Chembur, Mumbai-400088
University of Mumbai
2022-2023

Aim :

To extract the text from the image by using the AWS IAM, Textract, and AWS CLI service.

Problem Statement :

The goal of the project "Text Extractor using Amazon AWS Services" is to create a system that can automatically extract text from photos using the services provided by AWS IAM, Textract, and AWS CLI. The purpose of this project is to give businesses and organizations that need to handle a lot of photos with text an effective and affordable option. The software must be able to correctly recognize text included inside pictures, extract the content, and store it in a machine-readable format for future use. Optimising the system's efficiency to handle high numbers of photos, maintaining the confidentiality and privacy of the extracted data, and assuring the accuracy of text extraction are the project's main problems. The accomplishment of this project's objectives

Creating IAM user(demo_user) :

The screenshot shows the AWS IAM console interface for creating a new user. The browser tabs include 'Python 3-Address Code', 'TE 4 C batch CCL LAB', 'IAM Management Console', 'python - Google Drive', and 'CCL Project Report - Google Docs'. The URL is 'us-east-1.console.aws.amazon.com/iamv2/home?region=ap-south-1#/users/create'. The left sidebar shows the navigation menu with 'IAM > Users > Create user' selected. The main content area is titled 'Specify user details' and contains a 'User details' section. In this section, the 'User name' field is filled with 'demo_user'. Below the field, a note states: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and +, -, ., @, _ (hyphen)'. There is an unchecked checkbox for 'Provide user access to the AWS Management Console - optional' with a note: 'If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.' A blue information box contains the text: 'If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more'. At the bottom right of the form are 'Cancel' and 'Next' buttons. An 'Activate Windows' watermark is visible at the bottom right of the console window.

Python 3-Address Code | TE 4 C batch CCL LAB | IAM Management Console | python - Google Drive | CCL Project Report - Google Docs

us-east-1.console.aws.amazon.com/iamv2/home?region=ap-south-1#/users/create

Services | Search | [Alt+S]

Global | Swapnil Sawant

IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Specify user details

User details

User name

demo_user

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and +, -, ., @, _ (hyphen)

☐ Provide user access to the AWS Management Console - optional
If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel Next

Activate Windows
Go to Settings to activate Windows.

The screenshot shows the 'Permissions policies' step in the AWS IAM console. The browser tabs are the same as in the previous screenshot. The URL is 'us-east-1.console.aws.amazon.com/iamv2/home?region=ap-south-1#/users/create'. The left sidebar shows 'IAM > Users > Create user' selected. The main content area is titled 'Permissions policies (1/1076)'. It includes a search bar with the text 'Filter distributions by text, property or value'. Below the search bar is a table listing various AWS managed policies. The 'AdministratorAccess' policy is selected with a blue checkmark. At the bottom right of the console window, there is an 'Activate Windows' watermark.

Python 3-Address Code | TE 4 C batch CCL LAB | IAM Management Console | python - Google Drive | CCL Project Report - Google Docs

us-east-1.console.aws.amazon.com/iamv2/home?region=ap-south-1#/users/create

Services | Search | [Alt+S]

Global | Swapnil Sawant

IAM > Users > Create user

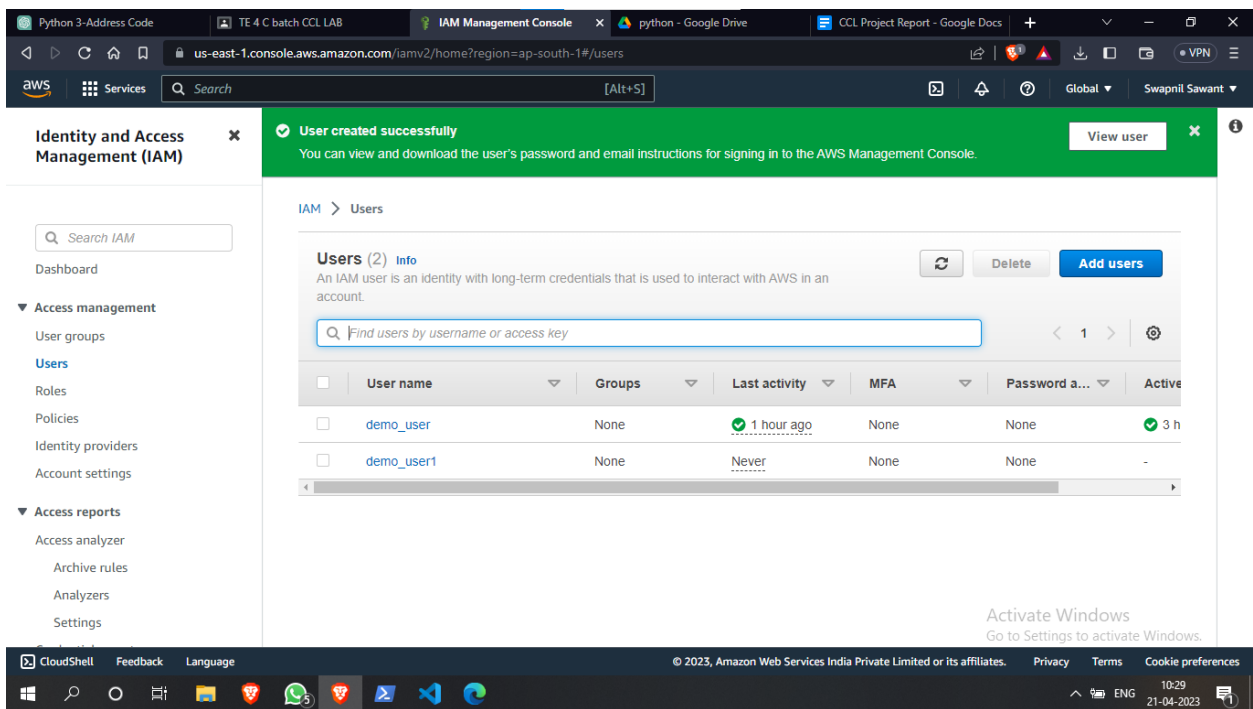
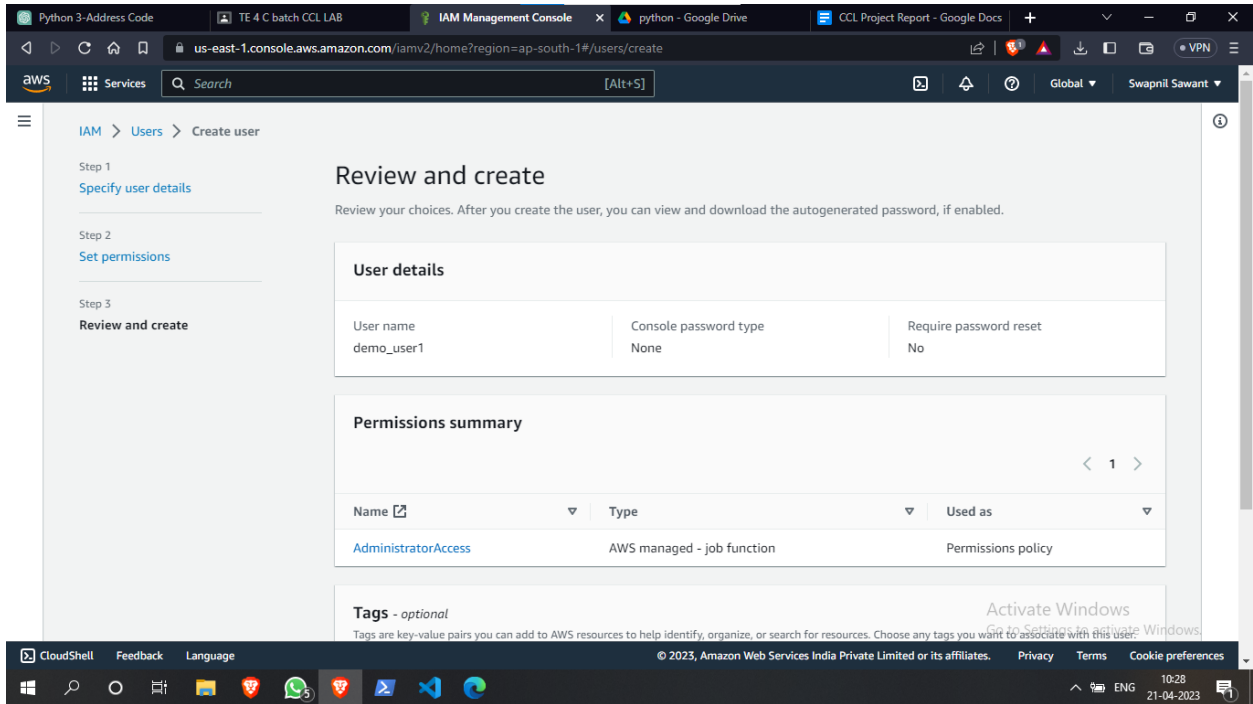
Permissions policies (1/1076)

Choose one or more policies to attach to your new user.

Filter distributions by text, property or value

	Policy name	Type	Attached entities
<input type="checkbox"/>	AccessAnalyzerServiceRole...	AWS managed	0
<input checked="" type="checkbox"/>	AdministratorAccess	AWS managed - job function	1
<input type="checkbox"/>	AdministratorAccess-Amplify	AWS managed	0
<input type="checkbox"/>	AdministratorAccess-AWSE...	AWS managed	0
<input type="checkbox"/>	AlexaForBusinessDeviceSet...	AWS managed	0
<input type="checkbox"/>	AlexaForBusinessFullAccess	AWS managed	0
<input type="checkbox"/>	AlexaForBusinessGatewayE...	AWS managed	0
<input type="checkbox"/>	AlexaForBusinessLifesized...	AWS managed	0
<input type="checkbox"/>	AlexaForBusinessNetworkP...	AWS managed	0

Activate Windows
Go to Settings to activate Windows.



The screenshot shows the AWS IAM console for the user 'demo_user'. The left sidebar contains navigation links for Identity and Access Management (IAM), Access management, and Access reports. The main content area displays the user's summary, including their ARN, console access status, and access keys. The 'Permissions' tab is selected, showing a list of permissions policies.

demo_user

Summary

ARN arn:aws:iam::510027780046:user/demo_user	Console access Disabled	Access key 1 AKIAXNQAGJ7HM5EV23WA - Active Used today. Created today.
Created April 21, 2023, 07:04 (UTC+05:30)	Last console sign-in -	Access key 2 Not enabled

Permissions policies (1)

Permissions are defined by policies attached to the user directly or through groups.

Find policies

Activate Windows
Go to Settings to activate Windows.

Give programmatic access to the user using the access key creation.

The screenshot shows the 'Create access key' wizard for the user 'demo_user'. The wizard is divided into three steps: Step 1 (Access key best practices & alternatives), Step 2 (optional, Set description tag), and Step 3 (Retrieve access keys). The first step is currently active, displaying information about access key best practices and alternatives.

Access key best practices & alternatives

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

- ☒ **Command Line Interface (CLI)**
You plan to use this access key to enable the AWS CLI to access your AWS account.
- ☐ **Local code**
You plan to use this access key to enable application code in a local development environment to access your AWS account.
- ☐ **Application running on an AWS compute service**
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
- ☐ **Third-party service**
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
- ☐ **Application running outside AWS**

Activate Windows
Go to Settings to activate Windows.

Python 3-Address Code | TE 4 C batch CCL LAB | IAM Management Console | python - Google Drive | CCL Project Report - Google Docs

us-east-1.console.aws.amazon.com/iamv2/home?region=ap-south-1#/users/details/demo_user1/create-access-key

Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

IAM > Users > demo_user1 > Create access key

Step 1
Access key best practices & alternatives

Step 2 - optional
Set description tag

Step 3
Retrieve access keys

Retrieve access keys

Access key
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIAZNQAGJ7HNROEDZ5Q	***** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

Activate Windows
Go to Settings to activate Windows.

CloudShell | Feedback | Language | © 2023, Amazon Web Services India Private Limited or its affiliates. | Privacy | Terms | Cookie preferences

10:33 21-04-2023

Note down the access key and secret access key in the file.

Python 3-Address Code | TE 4 C batch CCL LAB | IAM Management Console | python - Google Drive | CCL Project Report - Google Docs

us-east-1.console.aws.amazon.com/iamv2/home?region=ap-south-1#/users/details/demo_user1/section=security_credentials

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users
- Roles
- Policies
- Identity providers
- Account settings

Access reports

- Access analyzer
- Archive rules
- Analyzers
- Settings

Access keys (1)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. [Learn more](#)

Create access key

Access key	Status	Created	Last used	Last used region	Last used service
AKIAZNQAGJ7HM5EV23WA	Active	3 hours ago	1 hour ago	ap-south-1	textract

SSH public keys for AWS CodeCommit (0)
Use SSH public keys to authenticate access to AWS CodeCommit repositories. You can have a maximum of five SSH public keys (active or inactive) at a time. [Learn more](#)

Actions | Upload SSH public key

Activate Windows
Go to Settings to activate Windows.

CloudShell | Feedback | Language | © 2023, Amazon Web Services India Private Limited or its affiliates. | Privacy | Terms | Cookie preferences

10:33 21-04-2023

Configure the user for command line in powershell:

Download all the dependencies needed.

```
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\swapn> python --version
Python 3.9.12
PS C:\Users\swapn> pip3 --version
pip 21.2.4 from C:\Users\swapn\anaconda3\lib\site-packages\pip (python 3.9)
PS C:\Users\swapn> pip3 install boto3
pip 21.2.4 from C:\Users\swapn\anaconda3\lib\site-packages\pip (python 3.9)
PS C:\Users\swapn> pip3 install boto3
Requirement already satisfied: boto3 in c:\users\swapn\anaconda3\lib\site-packages (1.21.32)
Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in c:\users\swapn\anaconda3\lib\site-packages (from boto3) (0.5.0)
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in c:\users\swapn\anaconda3\lib\site-packages (from boto3) (0.10.0)
Requirement already satisfied: botocore<1.25.0,>=1.24.32 in c:\users\swapn\anaconda3\lib\site-packages (from boto3) (1.24.32)
Requirement already satisfied: urllib3<1.27,>=1.25.4 in c:\users\swapn\anaconda3\lib\site-packages (from botocore<1.25.0,>=1.24.32->boto3) (1.26.9)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in c:\users\swapn\anaconda3\lib\site-packages (from botocore<1.25.0,>=1.24.32->boto3) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\swapn\anaconda3\lib\site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.25.0,>=1.24.32->boto3) (1.16.0)
PS C:\Users\swapn> pip install awscli
Collecting awscli
  Downloading awscli-1.27.117-py3-none-any.whl (4.1 MB)
    |#####| 4.1 MB 1.3 MB/s
Collecting PyYAML<5.5,>=3.10
  Downloading PyYAML-5.4.1-cp39-cp39-win_amd64.whl (213 kB)
    |#####| 213 kB 1.3 MB/s
Collecting botocore==1.29.117
  Downloading botocore-1.29.117-py3-none-any.whl (10.7 MB)
    |#####| 10.7 MB 1.1 MB/s
Requirement already satisfied: rsa<4.8,>=3.1.2 in c:\users\swapn\anaconda3\lib\site-packages (from awscli) (4.7.2)
Requirement already satisfied: colorama<0.4.5,>=0.2.5 in c:\users\swapn\anaconda3\lib\site-packages (from awscli) (0.4.4)
Collecting docutils<0.17,>=0.10
  Downloading docutils-0.16-py2.py3-none-any.whl (548 kB)
    |#####| 548 kB 1.1 MB/s
Collecting s3transfer<0.7.0,>=0.6.0
  Downloading s3transfer-0.6.0-py3-none-any.whl (79 kB)
    |#####| 79 kB 793 kB/s
Requirement already satisfied: urllib3<1.27,>=1.25.4 in c:\users\swapn\anaconda3\lib\site-packages (from botocore==1.29.117->awscli) (1.26.9)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in c:\users\swapn\anaconda3\lib\site-packages (from botocore==1.29.117->awscli) (2.8.2)
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in c:\users\swapn\anaconda3\lib\site-packages (from botocore==1.29.117->awscli) (0.10.0)
Requirement already satisfied: six>=1.5 in c:\users\swapn\anaconda3\lib\site-packages (from python-dateutil<3.0.0,>=2.1->botocore==1.29.117->awscli) (1.16.0)
Requirement already satisfied: pyyaml<0.1.3 in c:\users\swapn\anaconda3\lib\site-packages (from rsa<4.8,>=3.1.2->awscli) (0.4.0)
Installing collected packages: botocore, s3transfer, PyYAML, docutils, awscli
  Attempting uninstall: botocore
    Found existing installation: botocore 1.24.32
    Uninstalling botocore-1.24.32:
      Successfully uninstalled botocore-1.24.32
  Attempting uninstall: s3transfer
    Found existing installation: s3transfer 0.5.0
    Uninstalling s3transfer-0.5.0:
      Successfully uninstalled s3transfer-0.5.0
Successfully installed PyYAML-5.4.1 awscli-1.27.117 docutils-0.16

Activate Windows
Go to Settings to activate Windows.
```

```
Windows PowerShell
Uninstalling docutils-0.17.1:
  Successfully uninstalled docutils-0.17.1
WARNING: You are using pip version 21.2.4, however your system has a python package
python-dateutil-1.8.0 requires setuptools, which is not installed.
Successfully installed PyYAML-5.4.1 awscli-1.27.117 docutils-0.16
PS C:\Users\swapn> pip install awscli
Requirement already satisfied: awscli in c:\users\swapn\anaconda3\lib\site-packages (1.27.117)
Requirement already satisfied: s3transfer<0.7.0,>=0.6.0 in c:\users\swapn\anaconda3\lib\site-packages (from awscli) (0.6.0)
Requirement already satisfied: colorama<0.4.5,>=0.2.5 in c:\users\swapn\anaconda3\lib\site-packages (from awscli) (0.4.4)
Requirement already satisfied: PyYAML<5.5,>=3.10 in c:\users\swapn\anaconda3\lib\site-packages (from awscli) (5.4.1)
Requirement already satisfied: docutils<0.17,>=0.10 in c:\users\swapn\anaconda3\lib\site-packages (from awscli) (0.16)
Requirement already satisfied: botocore==1.29.117 in c:\users\swapn\anaconda3\lib\site-packages (from awscli) (1.29.117)
Requirement already satisfied: rsa<4.8,>=3.1.2 in c:\users\swapn\anaconda3\lib\site-packages (from botocore==1.29.117->awscli) (4.7.2)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in c:\users\swapn\anaconda3\lib\site-packages (from botocore==1.29.117->awscli) (2.8.2)
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in c:\users\swapn\anaconda3\lib\site-packages (from botocore==1.29.117->awscli) (0.10.0)
Requirement already satisfied: six>=1.5 in c:\users\swapn\anaconda3\lib\site-packages (from python-dateutil<3.0.0,>=2.1->botocore==1.29.117->awscli) (1.16.0)
Requirement already satisfied: pyyaml<0.1.3 in c:\users\swapn\anaconda3\lib\site-packages (from rsa<4.8,>=3.1.2->awscli) (0.4.0)
PS C:\Users\swapn> aws configure --profile demo_user
File association not found for extension .py
AWS Access Key ID [None]: AKIAKNQAGJ7HMS5EV23WA
AWS Secret Access Key [None]: n0gEhZ78gqWJQE3gpW5dSpJA2biF5SKKYCOb04N
Default region name [None]: ap-south-1
Default output format [None]: json
PS C:\Users\swapn> cd .aws
PS C:\Users\swapn\aws> dir

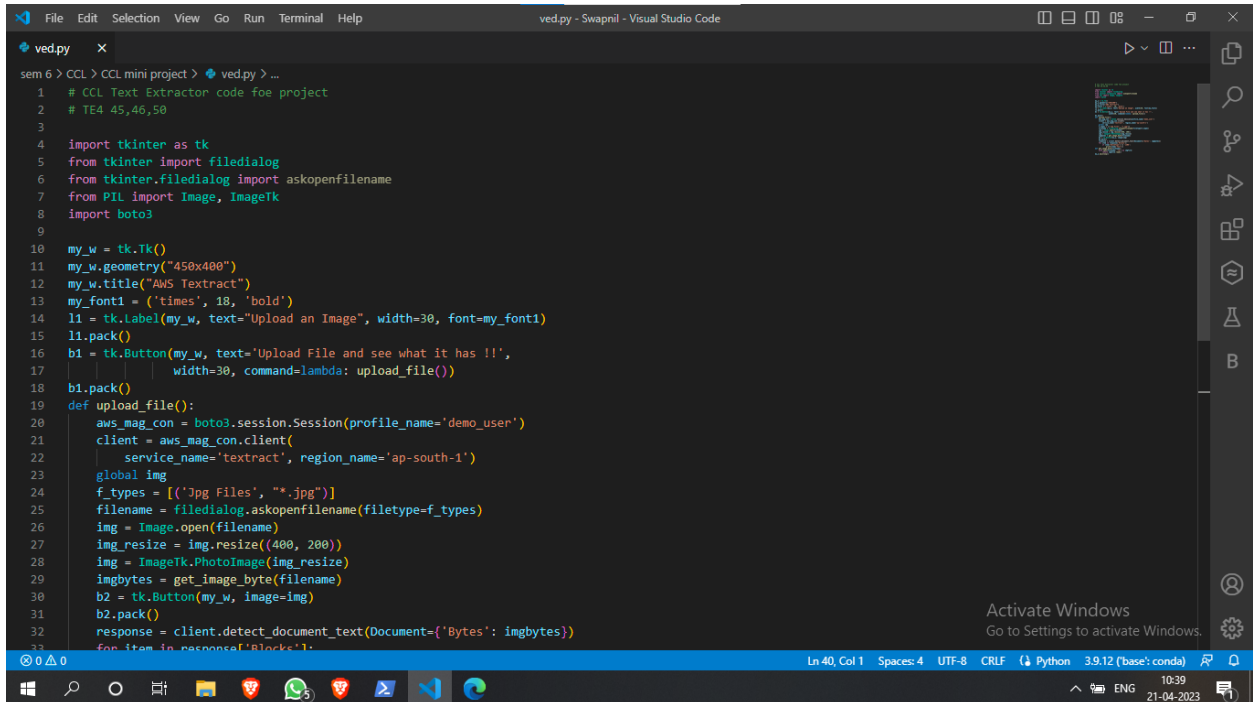
Directory: C:\Users\swapn\aws

Mode                LastWriteTime         Length Name
----                -
-a-----         21-04-2023    07:31           57 config
-a-----         21-04-2023    07:31          121 credentials

PS C:\Users\swapn\aws> type config
[profile demo_user]
region = ap-south-1
output = json
PS C:\Users\swapn\aws> type credentials
[demo_user]
aws_access_key_id = AKIAKNQAGJ7HMS5EV23WA
aws_secret_access_key = n0gEhZ78gqWJQE3gpW5dSpJA2biF5SKKYCOb04N
PS C:\Users\swapn\aws>

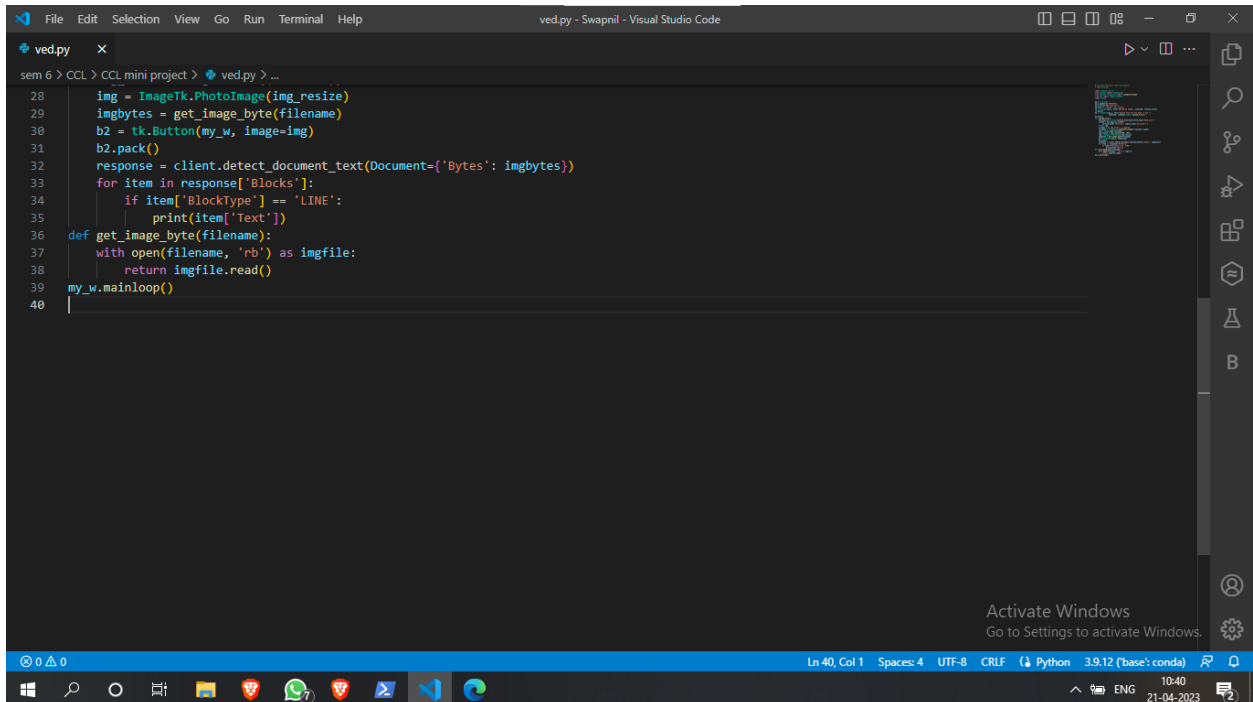
Activate Windows
Go to Settings to activate Windows.
```

Write the code for the text extractor :



```
ved.py x
sem 6 > CCL > CCL mini project > ved.py > ...
1 # CCL Text Extractor code for project
2 # TE4 45,46,50
3
4 import tkinter as tk
5 from tkinter import filedialog
6 from tkinter.filedialog import askopenfilename
7 from PIL import Image, ImageTk
8 import boto3
9
10 my_w = tk.Tk()
11 my_w.geometry("450x400")
12 my_w.title("AWS Textract")
13 my_font1 = ('times', 18, 'bold')
14 l1 = tk.Label(my_w, text="Upload an Image", width=30, font=my_font1)
15 l1.pack()
16 b1 = tk.Button(my_w, text="Upload File and see what it has !!",
17               width=30, command=lambda: upload_file())
18 b1.pack()
19 def upload_file():
20     aws_mag_con = boto3.session.Session(profile_name='demo_user')
21     client = aws_mag_con.client(
22         service_name='textract', region_name='ap-south-1')
23     global img
24     f_types = [('Jpg Files', '*.jpg')]
25     filename = filedialog.askopenfilename(filetype=f_types)
26     img = Image.open(filename)
27     img_resize = img.resize((400, 200))
28     img = ImageTk.PhotoImage(img_resize)
29     imgbytes = get_image_byte(filename)
30     b2 = tk.Button(my_w, image=img)
31     b2.pack()
32     response = client.detect_document_text(Document={'Bytes': imgbytes})
33     for item in response['Blocks']:
```

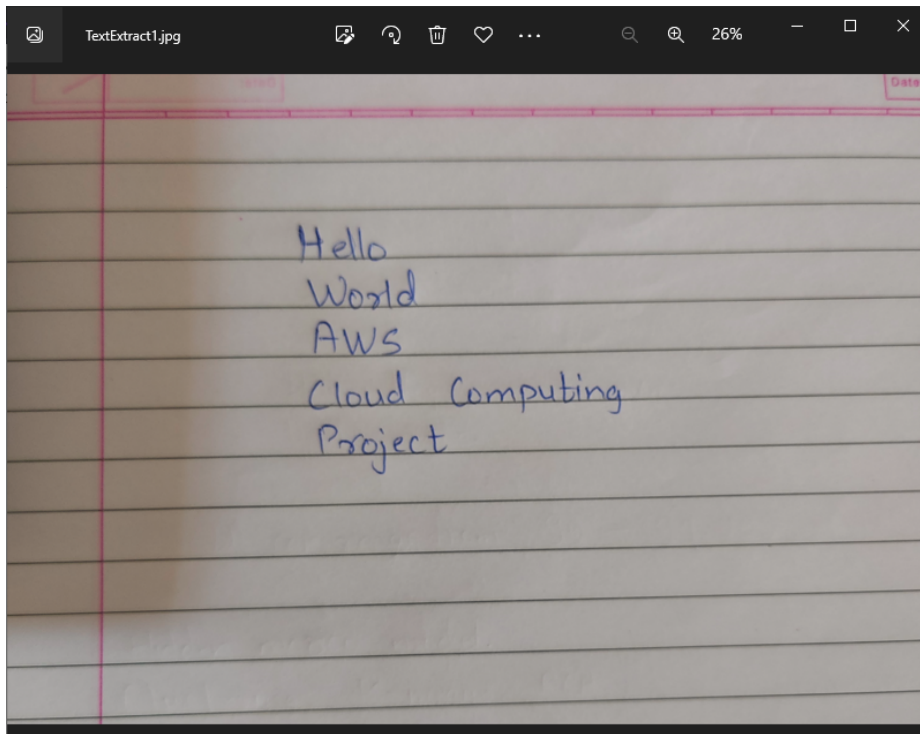
Ln 40, Col 1 Spaces: 4 UTF-8 CRLF Python 3.9.12 (base: conda)



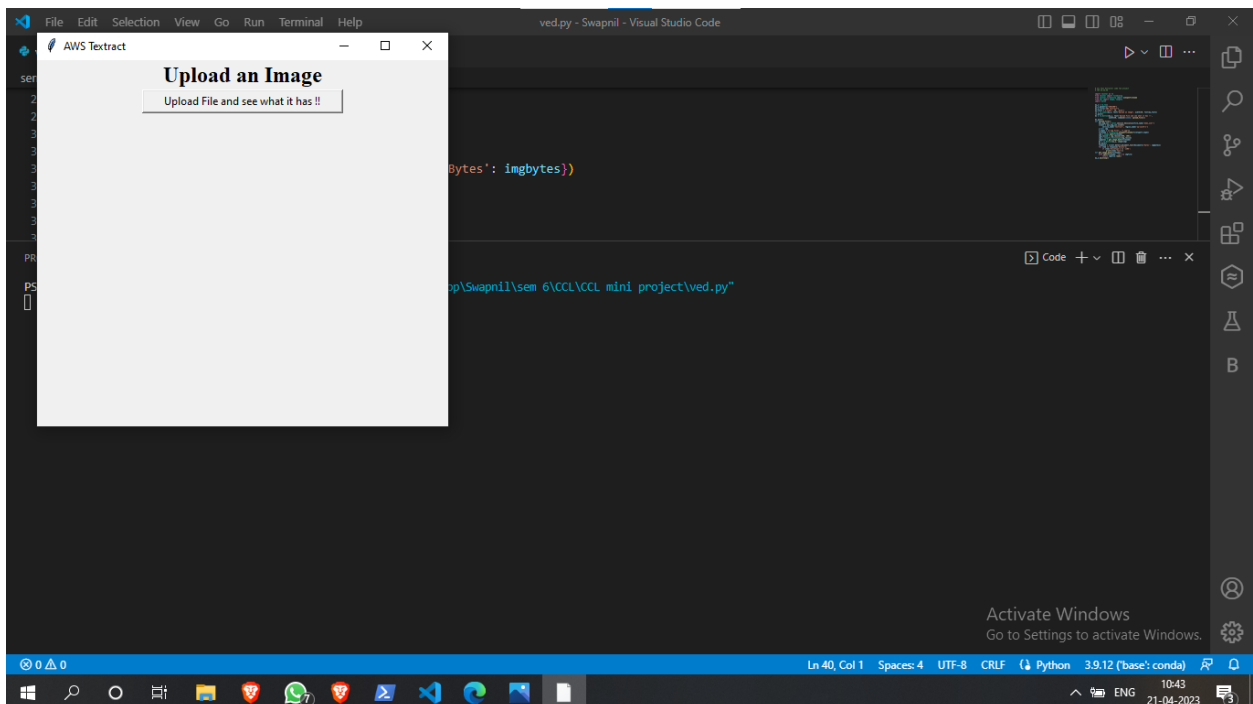
```
ved.py x
sem 6 > CCL > CCL mini project > ved.py > ...
28     img = ImageTk.PhotoImage(img_resize)
29     imgbytes = get_image_byte(filename)
30     b2 = tk.Button(my_w, image=img)
31     b2.pack()
32     response = client.detect_document_text(Document={'Bytes': imgbytes})
33     for item in response['Blocks']:
34         if item['Blocktype'] == 'LINE':
35             print(item['Text'])
36 def get_image_byte(filename):
37     with open(filename, 'rb') as imgfile:
38         return imgfile.read()
39 my_w.mainloop()
40
```

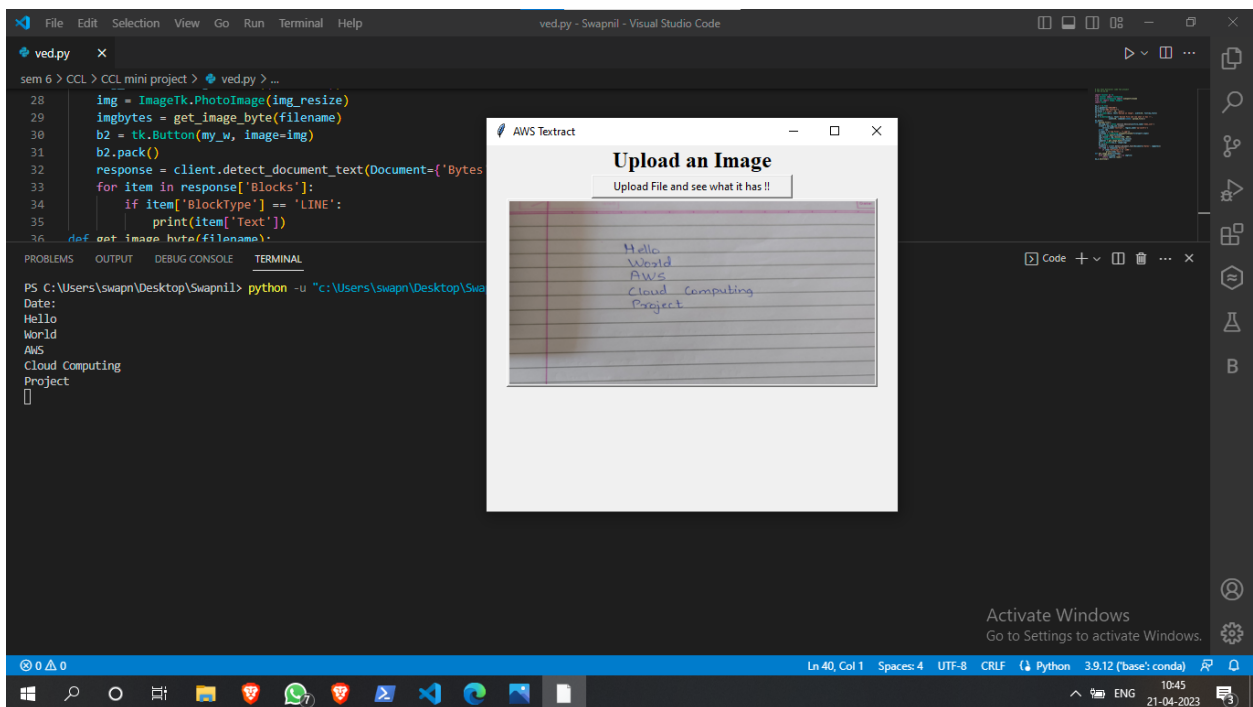
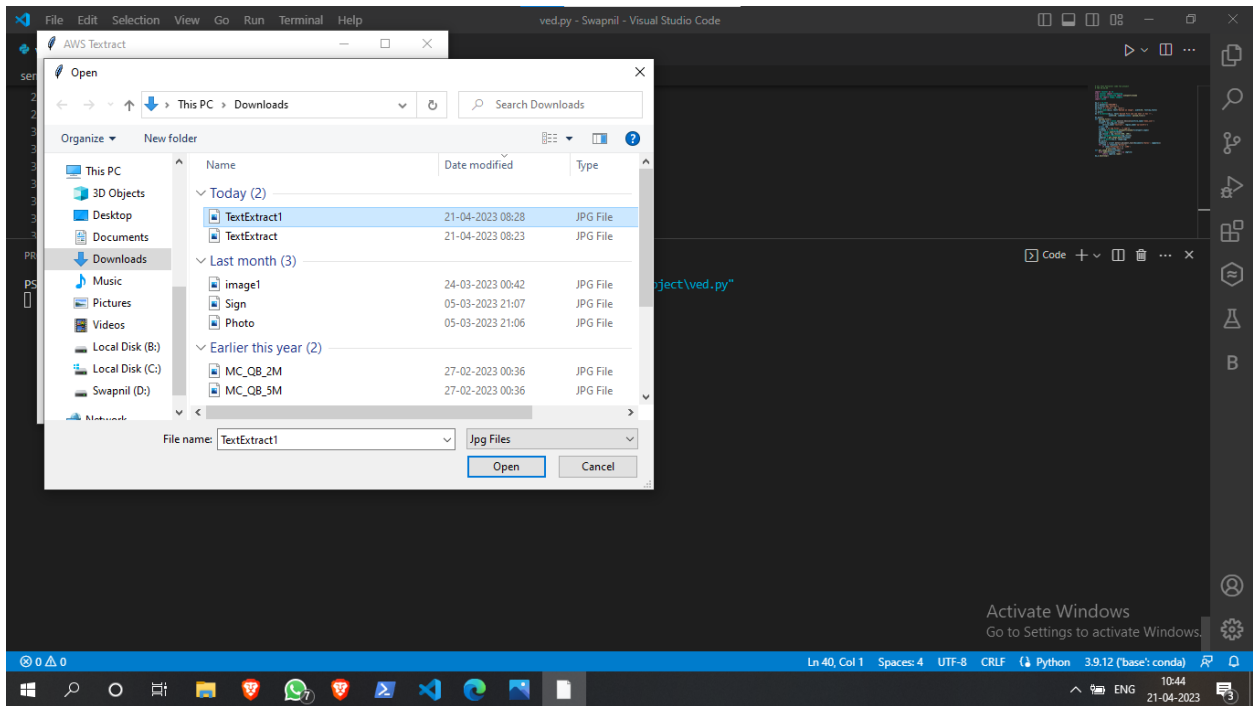
Ln 40, Col 1 Spaces: 4 UTF-8 CRLF Python 3.9.12 (base: conda)

INPUT :

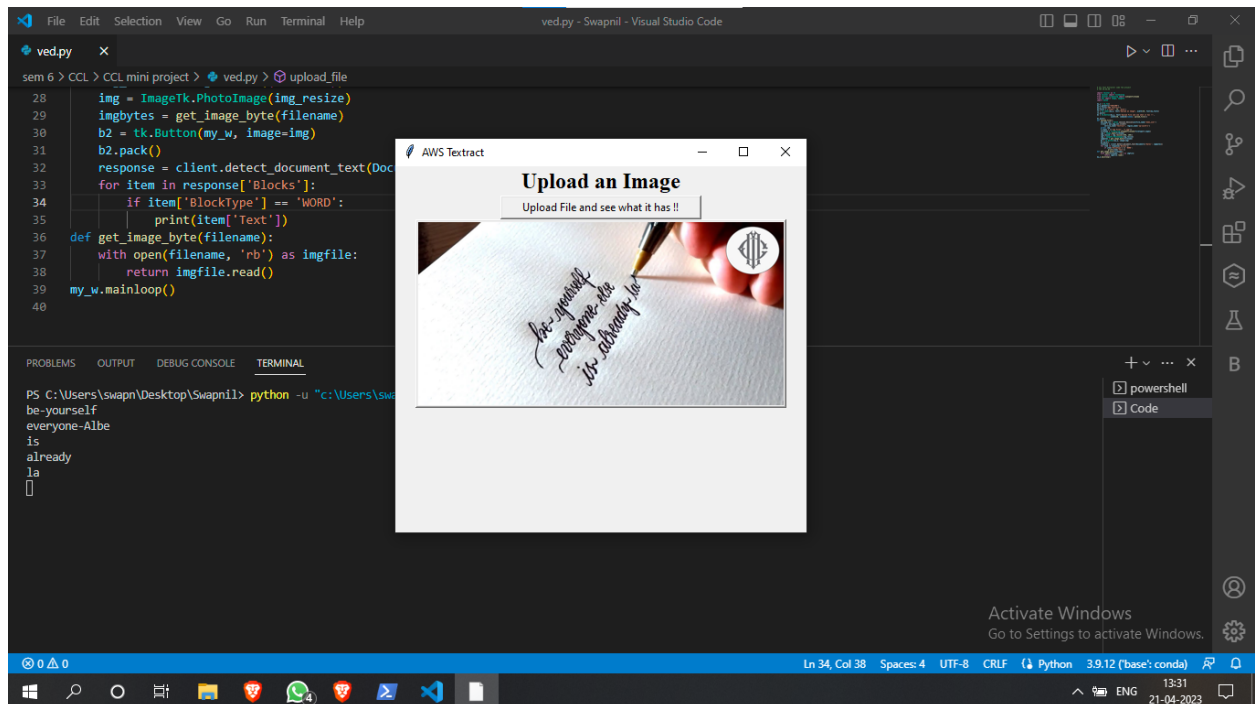
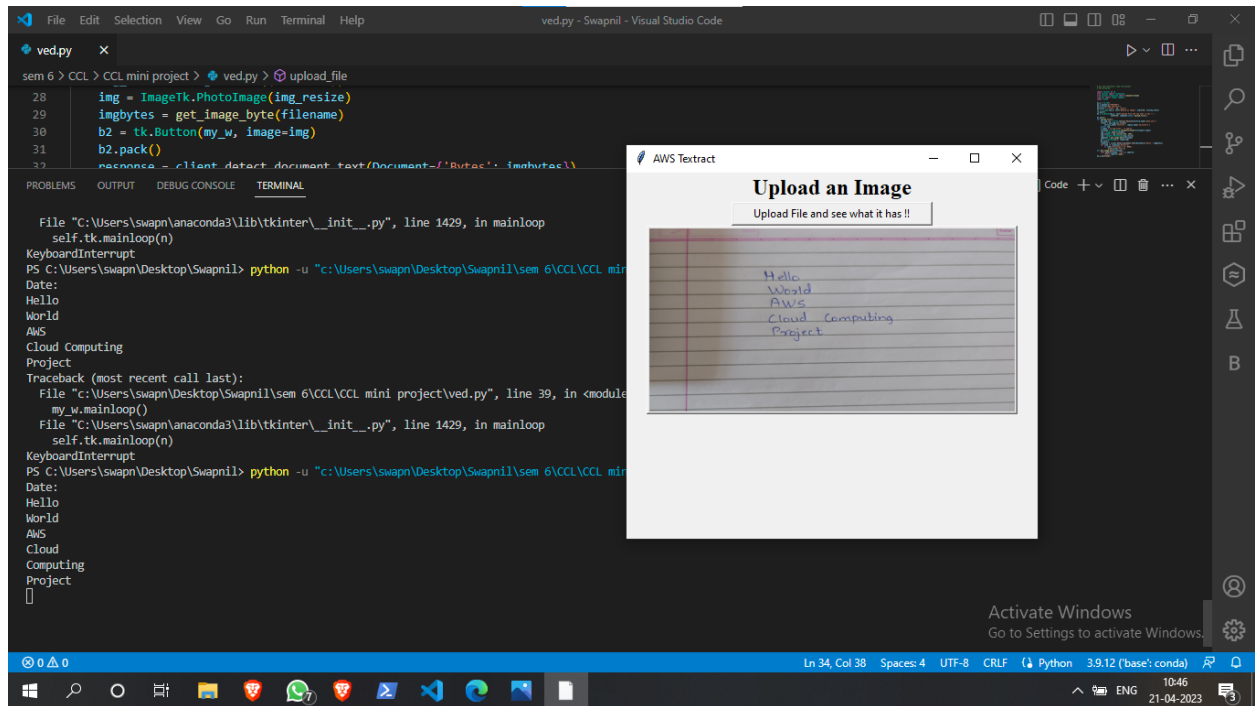


OUTPUT :





To display word by word



Conclusion :

In conclusion, creating a Text Extractor with Amazon AWS Services is a worthwhile project that has the potential to greatly increase commercial and organizational productivity. The system can automatically extract text from photos and save it in a machine-readable format for subsequent processing by utilizing the AWS IAM, Textract, and AWS CLI services. The project has a number of difficulties, such as guaranteeing text extraction accuracy, enhancing system performance, and guaranteeing the confidentiality and privacy of extracted data. However, with proper design and execution, the project may offer considerable advantages to companies and organizations, simplifying their processes, lowering manual labor requirements, and enhancing overall effectiveness. This project is an excellent illustration of how contemporary technology may be utilized to address complicated issues and improve our way of life.

References :

<https://aws.amazon.com/textract/>

<https://docs.aws.amazon.com/workspaces-web/latest/adminguide/getting-started-iam-user-access-keys.html>

<https://www.geeksforgeeks.org/how-to-extract-text-from-images-with-python/>