TESSERACT OCR:

Tesseract OCR is a powerful open-source optical character recognition engine developed by HP and now maintained by Google. It is widely used for extracting text from images, scanned documents, and PDFs. Tesseract supports over 100 languages and can be trained to recognize custom fonts and handwriting. It works best with high-quality images and offers features like layout analysis and multi-column text recognition. Tesseract is commonly used in document digitization, data extraction, and computer vision projects, and it integrates well with Python and other programming languages through libraries like pytesseract.

CODE:

import pytesseract

from pdf2image import convert\_from\_path

import cv2

import numpy as np

import os

# Set Tesseract and Poppler paths

pytesseract.pytesseract.tesseract\_cmd = r"C:\Program Files\Tesseract-OCR\tesseract.exe"

poppler\_path = r"C:\Users\priya\Downloads\Release-24.08.0-0\poppler-24.08.0\Library\bin"

pdf\_path = "WT\_38.pdf"

# Convert PDF to high-resolution images

pages = convert\_from\_path(pdf\_path, dpi=500, poppler\_path=poppler\_path)

output\_text = ""

    print(f"[INFO] Processing page {i+1}/{len(pages)}")

    # Convert to OpenCV

    image = cv2.cvtColor(np.array(page), cv2.COLOR\_RGB2BGR)

    # --- PREPROCESSING FOR BEST OCR ---

    gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

    # Contrast stretch

    gray = cv2.normalize(gray, None, alpha=0, beta=255, norm\_type=cv2.NORM\_MINMAX)

    # Thresholding

    \_, thresh = cv2.threshold(gray, 0, 255, cv2.THRESH\_BINARY + cv2.THRESH\_OTSU)

    # Denoising

    clean = cv2.fastNlMeansDenoising(thresh, h=30)

    # Morphology to remove small dots

    kernel = cv2.getStructuringElement(cv2.MORPH\_RECT, (1, 1))

    clean = cv2.morphologyEx(clean, cv2.MORPH\_OPEN, kernel)

    # --- OCR with Tesseract ---

    config = r'--oem 3 --psm 6 -l eng'  # Best for paragraph reading

    text = pytesseract.image\_to\_string(clean, config=config)

    output\_text += f"\n--- Page {i+1} ---\n{text}"

# Save output

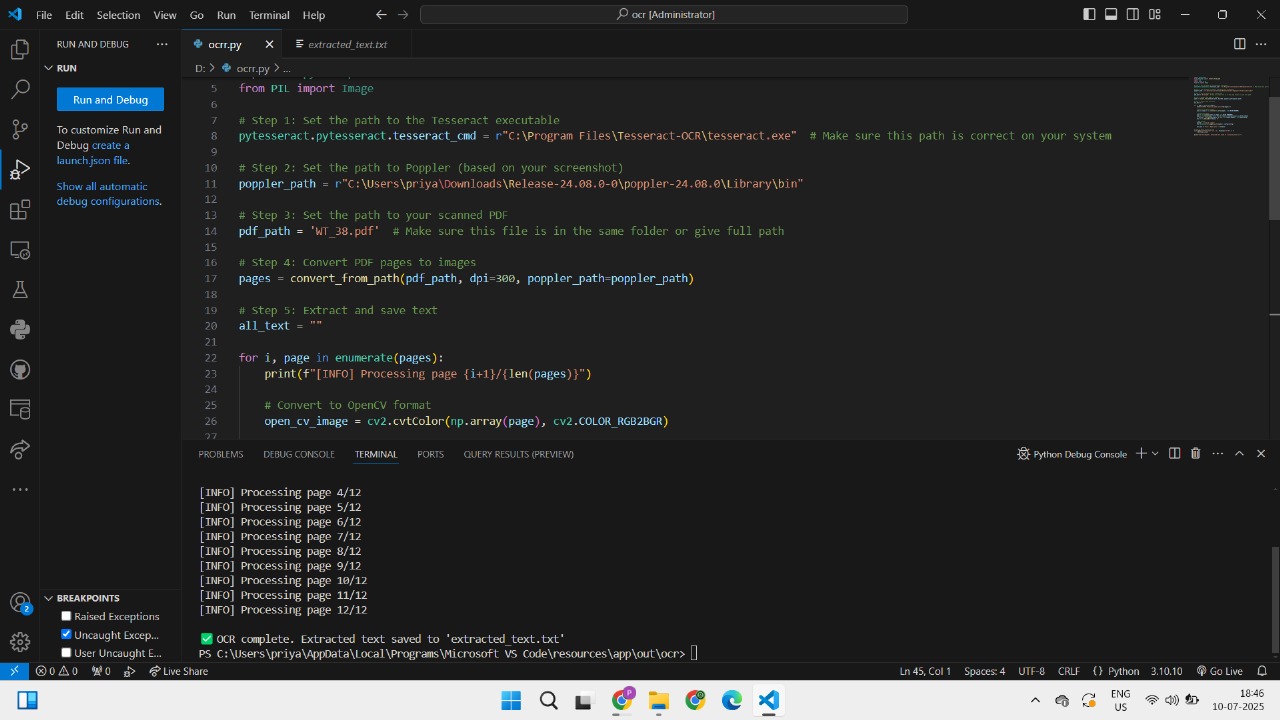
output\_file = "extracted\_text\_high\_accuracy.txt"

with open(output\_file, "w", encoding="utf-8") as f:

    f.write(output\_text)

print(f"[✅] OCR Completed. Text saved to: {output\_file}")

OUTPUT:



A screenshot of a computer

AI-generated content may be incorrect.