ocr_product_scanner/

├── app.py # Streamlit/Flask frontend

├── ocr_module.py # OCR logic (Tesseract or EasyOCR)

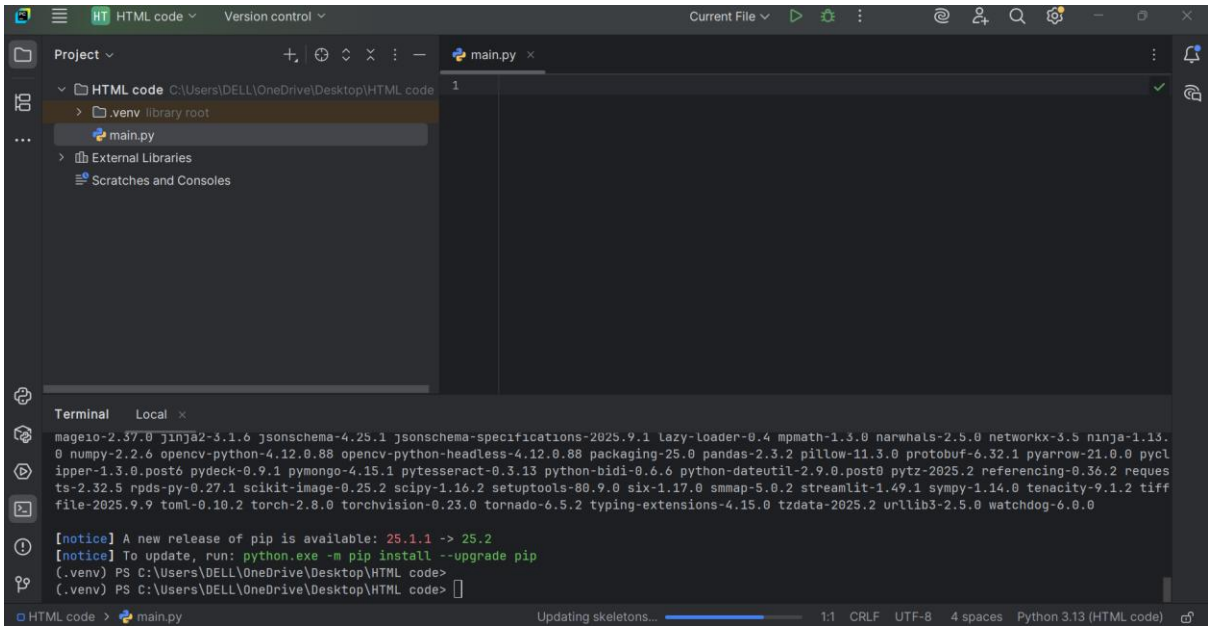
├── db_module.py # MongoDB query functions

├── utils.py # Helper functions (preprocessing, matching)

├── sample_images/ # Input images for testing

├── requirements.txt # Dependencies

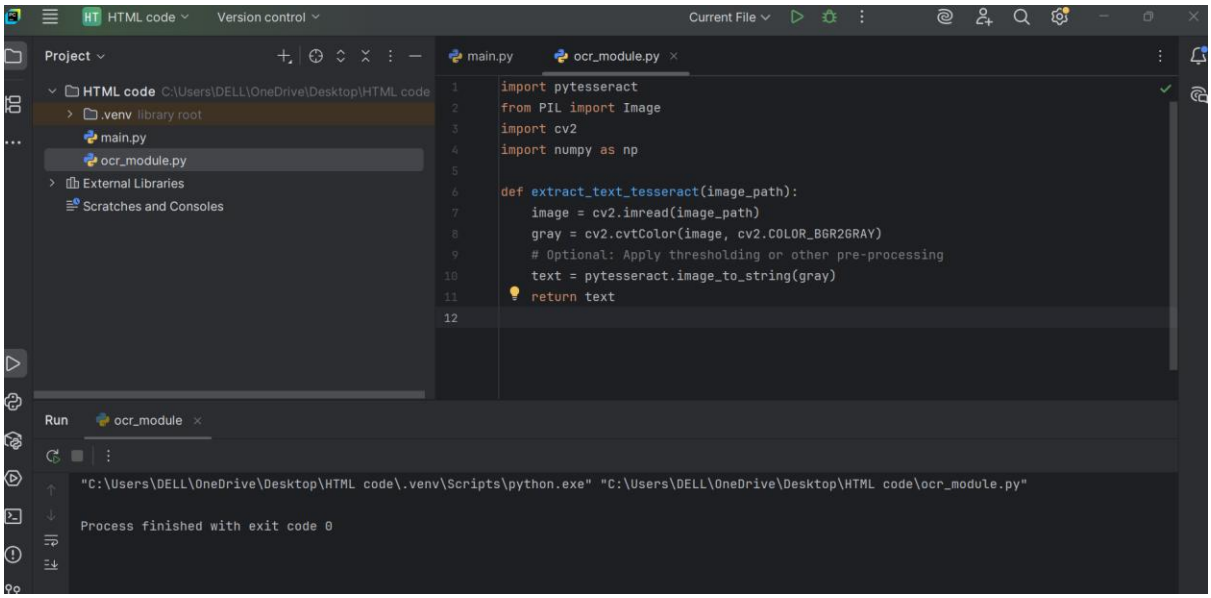
├── README.md # Instructions + overview

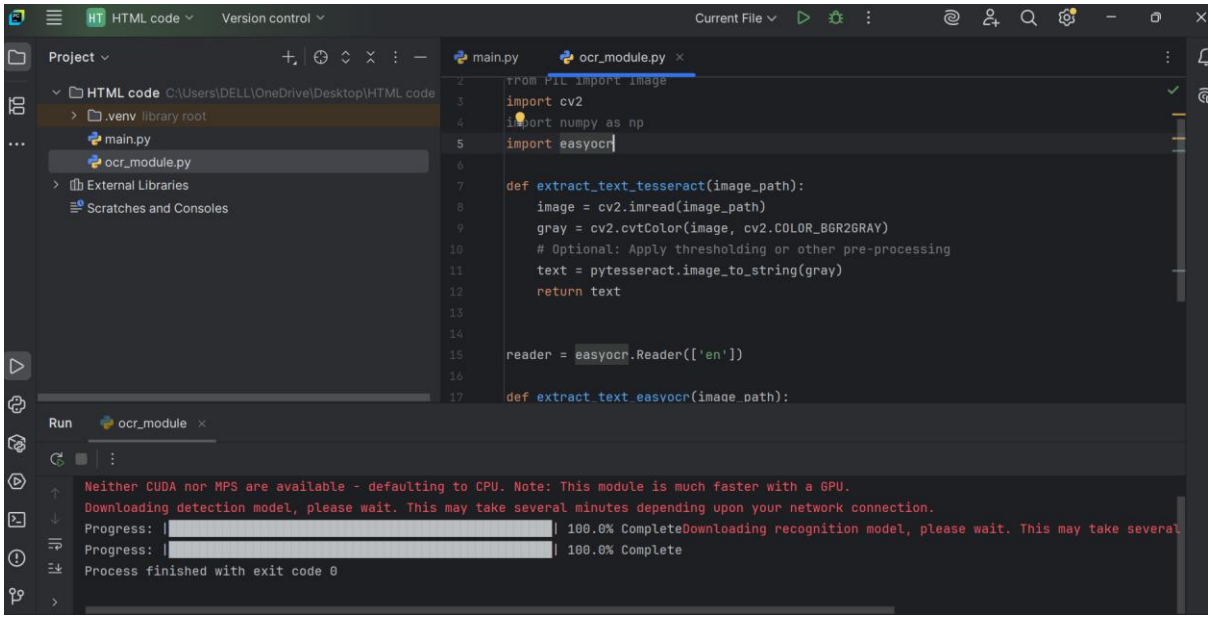


pip install pytesseract pillow opencv-python pymongo streamlit easyocr

```
import pytesseract
from PIL import Image
import cv2
import numpy as np

def extract_text_tesseract(image_path):
    image = cv2.imread(image_path)
```

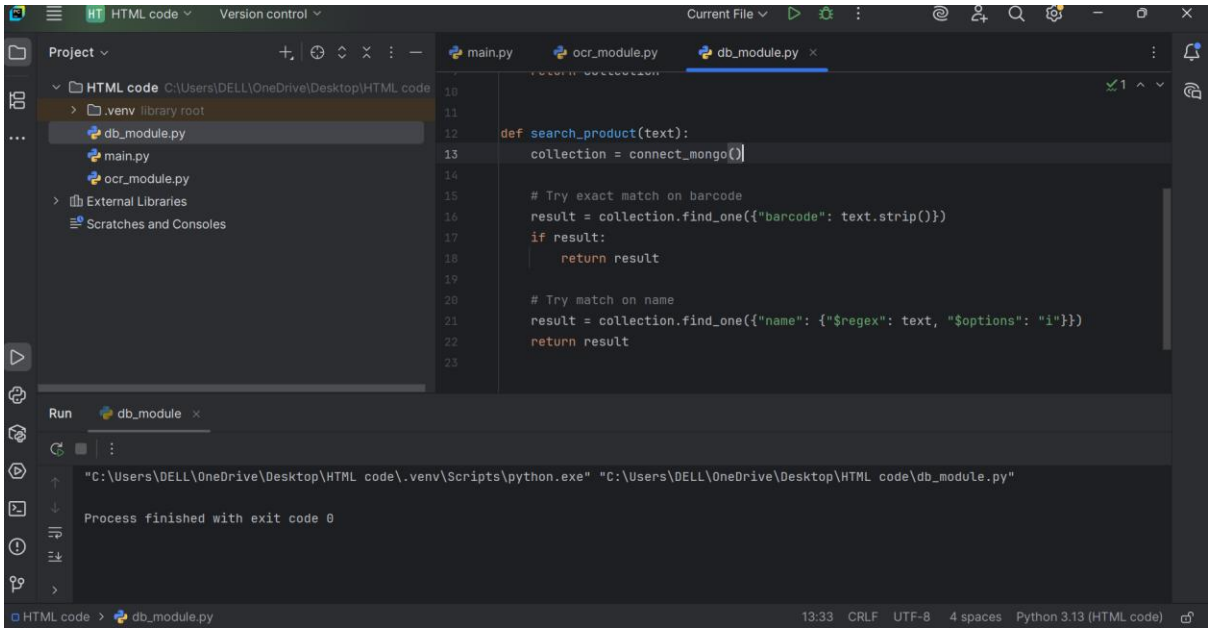
```
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    # Optional: Apply thresholding or other pre-processing
    text = pytesseract.image_to_string(gray)
    return text
```

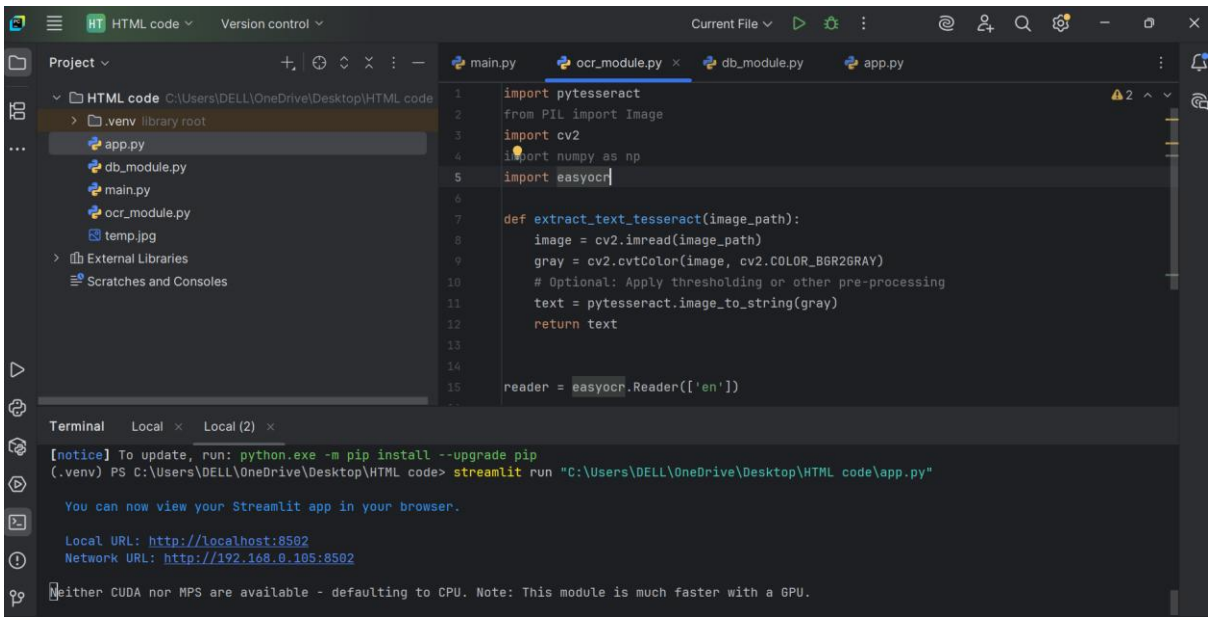





```
import easyocr

reader = easyocr.Reader(['en'])
```

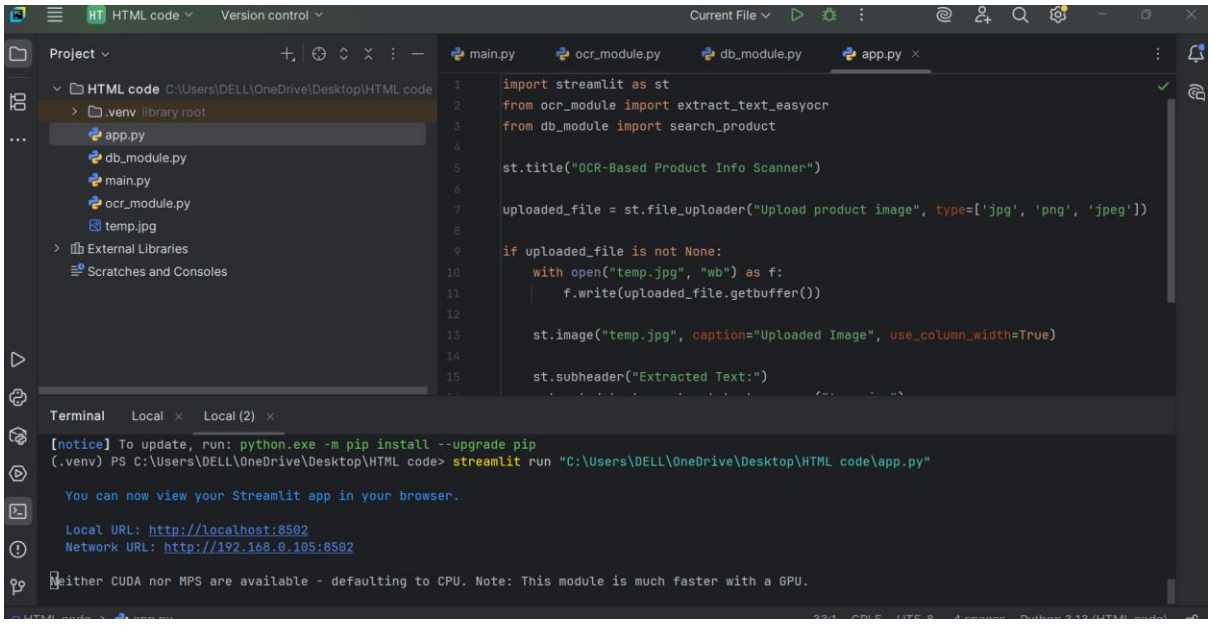
```
def extract_text_easyocr(image_path):
    result = reader.readtext(image_path, detail=0)
    return " ".join(result)
```






```
from pymongo import MongoClient

def connect_mongo():
```

```
    url = "mongodb+srv://ingredoai2:BXM6Hc1R57Hkiofy@cluster0.zimunh9.mongodb.net/"
    client = MongoClient(url)
    db = client["your_database_name"]  # Replace with actual DB name
    collection = db["your_collection_name"]  # Replace with actual collection
    return collection

def search_product(text):
    collection = connect_mongo()

    # Try exact match on barcode
    result = collection.find_one({"barcode": text.strip()})
    if result:
        return result

    # Try match on name
    result = collection.find_one({"name": {"$regex": text, "$options": "i"}})
    return result
```



```
import streamlit as st
from ocr_module import extract_text_easyocr
from db_module import search_product

st.title("OCR-Based Product Info Scanner")

uploaded_file = st.file_uploader("Upload product image", type=['jpg', 'png', 'jpeg'])

if uploaded_file is not None:
    with open("temp.jpg", "wb") as f:
        f.write(uploaded_file.getbuffer())

    st.image("temp.jpg", caption="Uploaded Image", use_column_width=True)

    st.subheader("Extracted Text:")
    extracted_text = extract_text_easyocr("temp.jpg")
    st.write(extracted_text)

    st.subheader("Searching Database...")
    result = search_product(extracted_text)

    if result:
        st.success("Product Found!")
        st.json({
            "Name": result.get("name", "N/A"),
            "Brand": result.get("brand", "N/A"),
            "Ingredients": result.get("ingredients", "N/A"),
            "Nutrition Facts": result.get("nutrition_facts", "N/A"),
            "Categories": result.get("categories", "N/A"),
        })
    else:
```

```
        st.error("Product not found.")
```

OCR-Based Product Info Scanner

This tool allows you to scan a product image and retrieve detailed information by matching it with a MongoDB database.

Features

- OCR text extraction (using EasyOCR or Tesseract)
- Product search via MongoDB
- Web interface using Streamlit
- Outputs product name, brand, ingredients, nutrition facts, and categories

Setup Instructions

1. Clone the repo
2. Install dependencies:
   ```
   pip install -r requirements.txt
   ```
3. Run the app:
   ```
   streamlit run app.py
   ```

Sample Input

Add your test images in `/sample_images/`.

Output

- Extracted text from label

- Retrieved product details (if found)

Demo

Include a few screenshots or a short Loom/video demo here.