**AI Driven Information Retrieval from Image**

***Problem statement:***

In the modern digital landscape, organizations are inundated with vast amounts of visual data, including scanned documents, images, and photos. Extracting meaningful information from these images manually is not only time-consuming but also prone to errors, leading to inefficiencies. This project aims to address this challenge by implementing an automated information extraction solution from images. Utilizing advanced technologies such as Optical Character Recognition (OCR) and AI-based image analysis, we seek to streamline the process of extracting and organizing data from visual content.

***Objectives***

The primary objective of this project is to develop a Python API that accepts images and different folders as input, extracts any embedded text, and returns it in a clean and readable format. The API will be designed to accommodate a variety of image formats and types of text content, including printed, handwritten, and stylized text.

***Key Functionalities***

1. *Image File Input* 
   * Accept common image formats (JPEG, PNG, TIFF, etc.).
   * Allow users to upload images directly for transcription.
2. *Text Extraction* 
   * Implement OCR to transcribe text, accommodating different font styles, sizes, and orientations.
   * Support multilingual text, complex layouts, and mixed content (text and graphics).
3. *Output and Formatting* 
   * Return transcribed text in clear formats (plain text, JSON, CSU).
   * Include metadata like text location and confidence levels.

***Technologies***

* Python: The primary programming language for developing the API.
* Optical Character Recognition (OCR): Libraries such as Tesseract will be used for text extraction.
* Image Processing Libraries: Libraries like OpenCV and PIL will assist in pre- processing images to enhance OCR accuracy.
* Web Framework: Flask or FastAPI will be utilized to create the API endpoints.
* Data Formats: JSON and CSV libraries will be employed for structured output.
* Packages:
  + - Flask : A lightweight web framework for building web applications and APIs in Python.
    - Pytesseract : It allows you to extract text from images using Optical Character Recognition (OCR).
    - Pdfplumber : A Python library for extracting text, tables, and metadata from PDF files .
    - Whoosh : A pure python search engine library.

***Modules***

def allowed\_file(filename):

    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED\_EXTENSIONS

* Function to check if file has an allowed extension

def extract\_text\_from\_image(image\_path):

    try:

        image = Image.open(image\_path)

        text = pytesseract.image\_to\_string(image)

        return text

    except Exception as e:

* Function to extract text from an image using OCR

def extract\_text\_from\_pdf(pdf\_path):

    text = ""

    try:

        with pdfplumber.open(pdf\_path) as pdf:

            for page in pdf.pages:

                page\_text = page.extract\_text()  # Extract text from each page

                if page\_text:  # Only add non-empty text

                    text += page\_text + "\n"

    except Exception as e:

* Function to extract text from an pdf using pdfplumber

def index\_text(text):

    writer = AsyncWriter(ix)  # Open an asynchronous writer for the index

    writer.add\_document(content=text)  # Add document to index

    writer.commit()  # Commit changes

* Function to index the extract text in whoosh

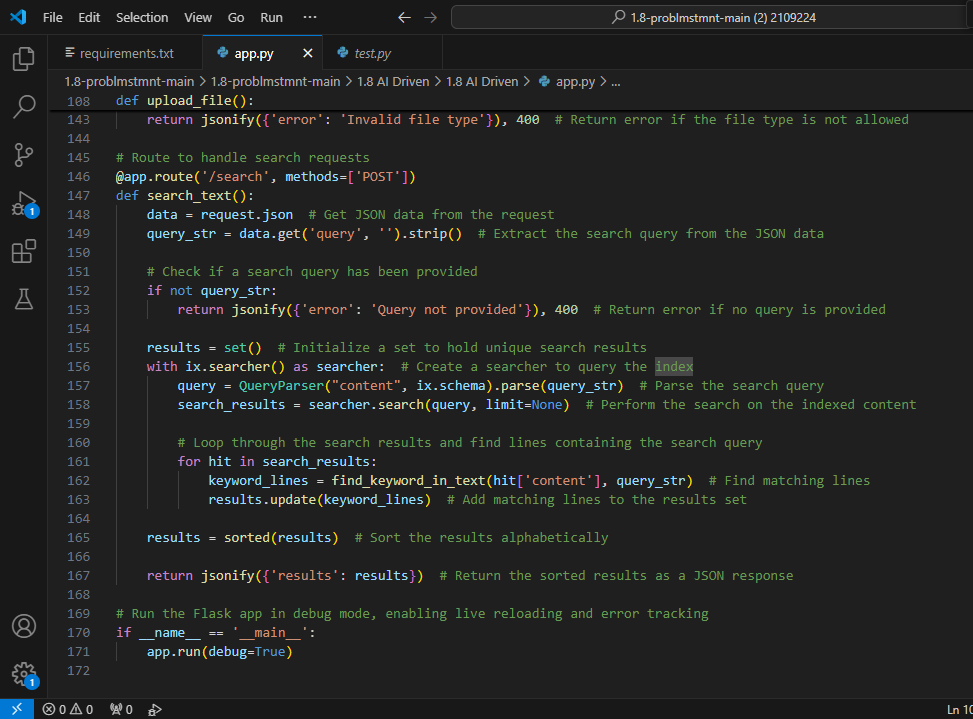
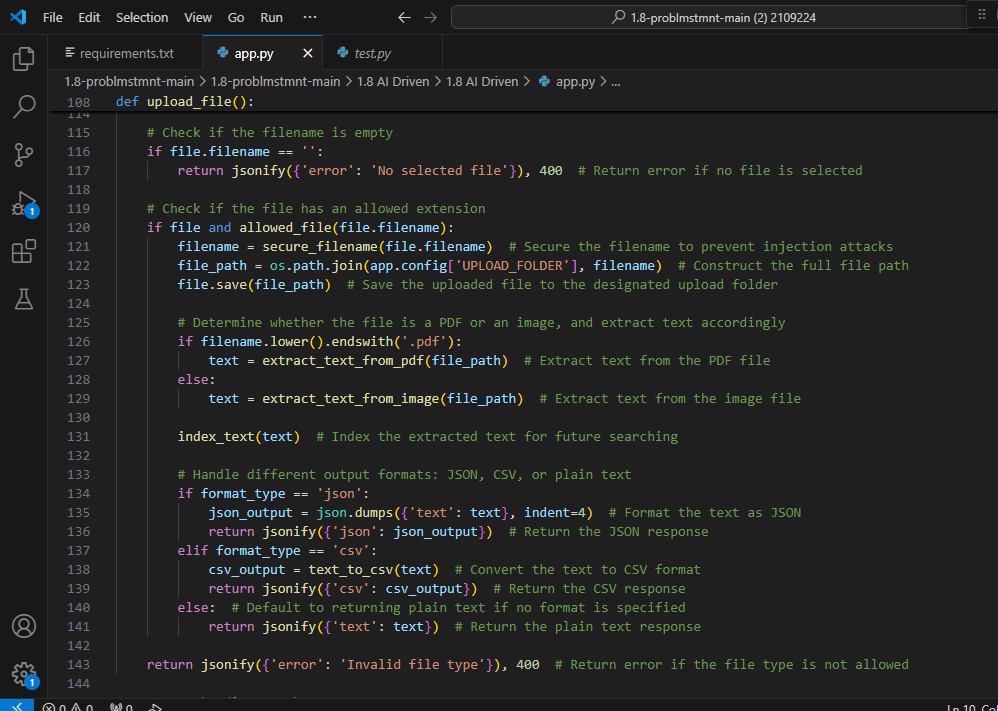
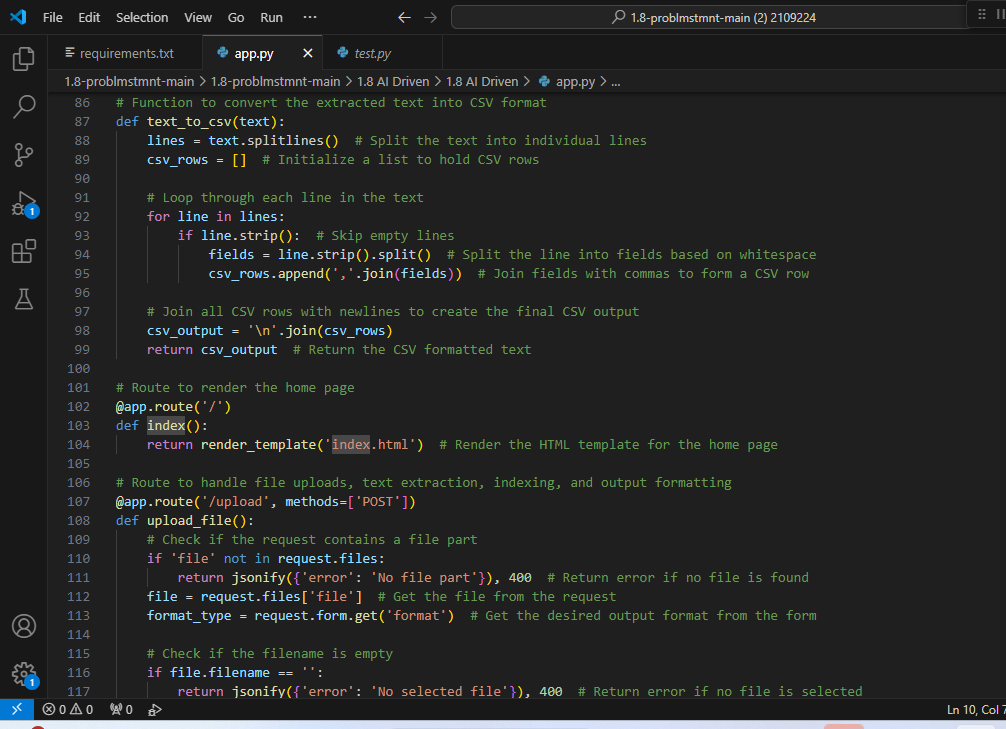
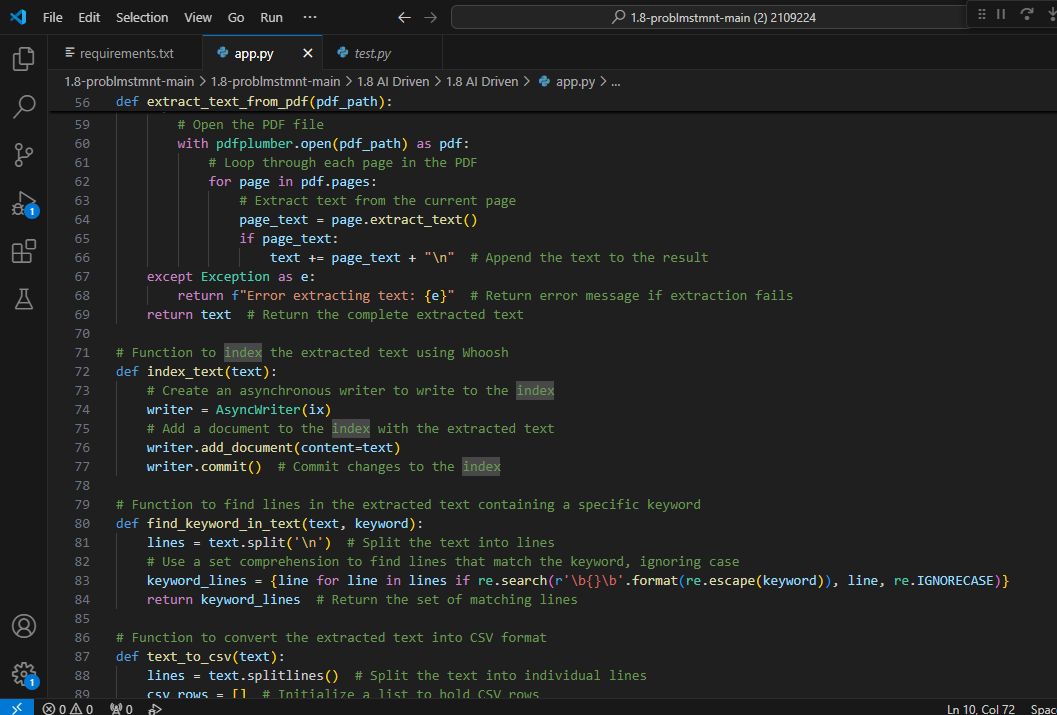
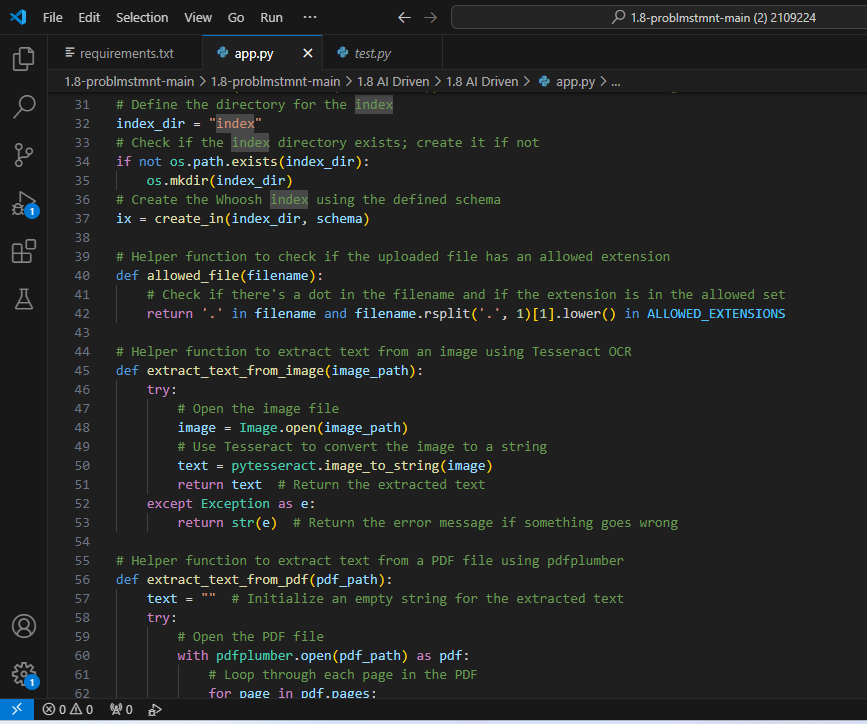
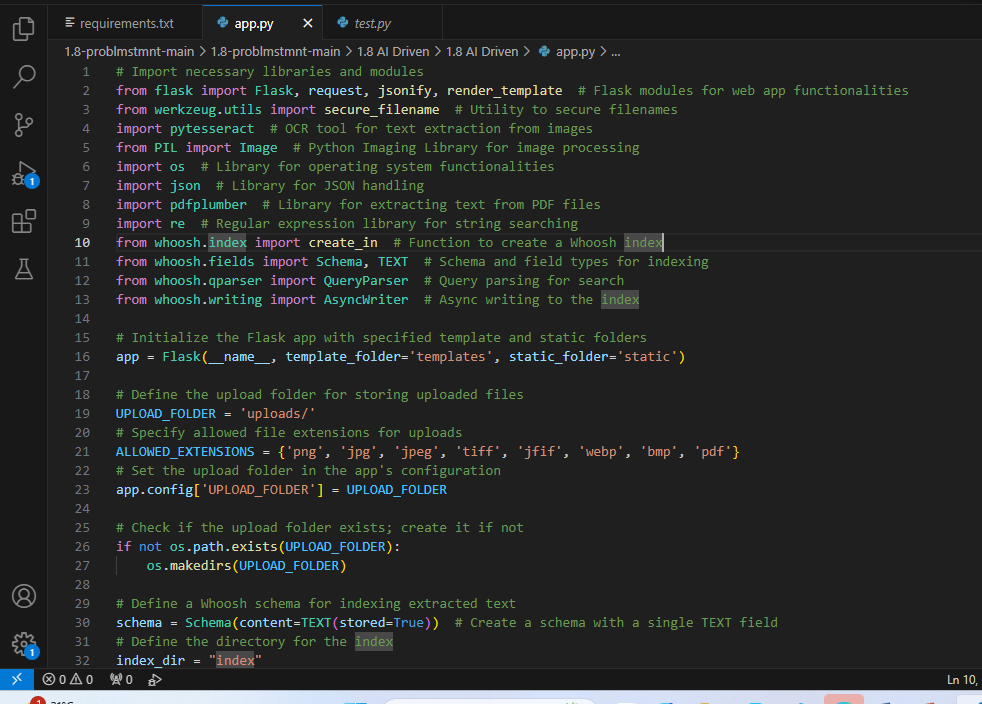
def find\_keyword\_in\_text(text, keyword):

    lines = text.split('\n')  # Split the text into lines

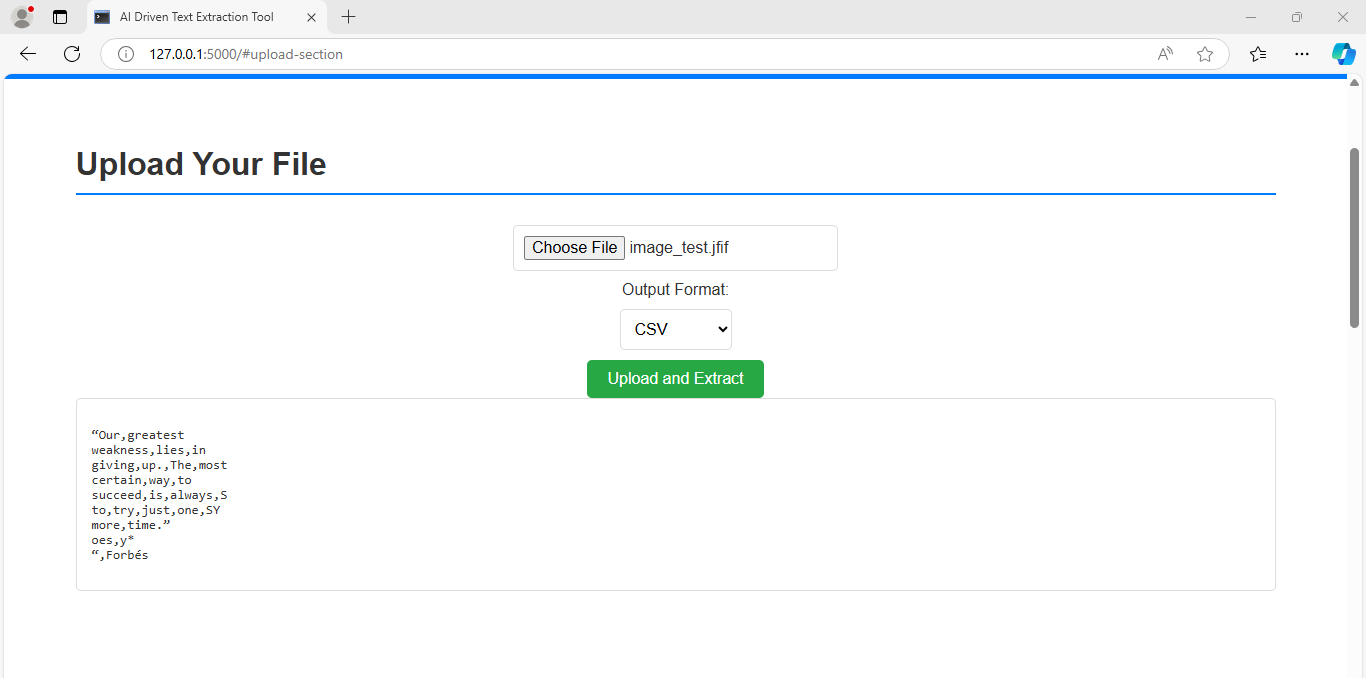
    keyword\_lines = {line for line in lines if re.search(r'\b{}\b'.format(re.escape(keyword)), line, re.IGNORECASE)}

    return keyword\_lines  # Return lines containing the keyword

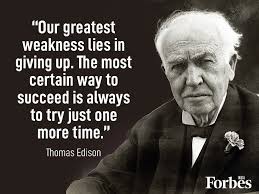
* Function to find lines in the text containing the specified keyword

***API***

***UI:***

******

***Image which we are given is:***

******

***Testing***

* + - * + Functional testing: upload, extraction, output generation
        + Accuracy Testing: Measure the accuracy of the extracted text against
        + end to end testing: The complete workflow from upload to output

***conclusion***

Our project provides a robust solution for automated information extraction from images. By leveraging OCR and Al technologies, it enhances efficiency and accuracy in data retrieval, catering to diverse organizational needs, Future enhancements may include improved multiple language support and advanced layout recognition capabilities.