**Assignment: OCR and Document Search Web Application Prototype**

**Note - DO NOT REPLY TO THE EMAIL TO SAY “I ACCEPT” OR ANYTHING.**

**DIRECTLY SUBMIT THE ASSIGNMENT TO -** [**https://forms.gle/7FfL7bP1zeGKtezt6**](https://forms.gle/7FfL7bP1zeGKtezt6)**.   
DO NO SEND ANY EMAILS UNLESS ABSOLUTELY NECESSARY.**

**FEEL FREE TO MAKE ANY ASSUMPTIONS IF ANYTHING IS UNCLEAR & MENTION IT IN YOUR NOTE.**

**Objective**

Develop and deploy a web-based prototype that demonstrates the ability to perform Optical Character Recognition (OCR) on an uploaded image (in picture format) containing text in both Hindi and English. The web application should also implement a basic keyword search functionality based on the extracted text. The prototype must be accessible via a live URL.

**Scope of the Assignment**

This assignment focuses on creating a web application that allows users to upload a single image, processes the image to extract text using OCR, and provides a basic search feature. The application must be deployed and accessible online.

**Tasks**

**Task 1: Setup and OCR Implementation**

1. **Environment Setup**:
   * Set up a Python environment with the necessary libraries, including Huggingface Transformers, PyTorch, and any other dependencies required for OCR.
   * Explore the following OCR models and choose one to implement:
     + ColPali implementation of the new Byaldi library + Huggingface transformers for Qwen2-VL.
     + General OCR Theory (GOT), a 580M end-to-end OCR 2.0 model.
2. **OCR Model Integration**:
   * Implement the chosen OCR model to process a single uploaded image (JPEG, PNG, or other common picture formats) containing text in both Hindi and English.
   * Ensure the model successfully extracts text from the image and returns the extracted text in a structured format (JSON or plain text).

**Task 2: Web Application Development**

1. **Web Application**:
   * Develop a simple web application using **Gradio** or **Streamlit**.
   * The application should allow users to:
     + Upload an image file for OCR processing.
     + Display the extracted text from the image.
     + Enter keywords to search within the extracted text.
   * Display search results on the same page, highlighting the matching sections.

**Task 3: Deployment**

1. **Deploy the Web Application**:
   * Deploy the web application on platforms like Hugging Faces, Streamlit Sharing, or any other suitable platform.
   * Ensure the application is accessible via a public URL.

**Deliverables**

1. **Code Submission**:
   * Python scripts for the web application, including the OCR processing and search functionality.
   * A README file explaining how to set up the environment, run the web application locally, and details about the deployment process.
2. **Live Web Application**:
   * The live URL of the deployed web application where the OCR and search functionalities can be tested.
3. **Extracted Text and Search Output**:
   * JSON or plain text output of the extracted text from the uploaded image.
   * Demonstration of the search functionality with example keywords.

**Evaluation Criteria**

* **Accuracy**: How well the OCR model extracts text from both Hindi and English sections of the image.
* **Functionality**: The web application should correctly handle image uploads, extract text, and allow keyword searches.
* **User Interface**: The web interface should be simple, intuitive, and functional.
* **Deployment**: The application must be accessible online, with a reliable deployment process.
* **Clarity**: Clear and concise documentation and code structure.
* **Completeness**: All deliverables are submitted and demonstrate the required functionality.

**Deadline**

* **Submission Deadline**: 1 week from receiving the assignment.

**Instructions for Submission**

* Submit a ZIP file containing all your code, the README file, and any additional resources (e.g., screenshots of the web application).
* Provide the live URL of the deployed web application.