### **Task: Develop an Intelligent Document Processing System**

**Objective:** Design and implement an Intelligent Document Processing System using Django, Python OCR and AI libraries, and a MySQL/MariaDB database. The system should be capable of extracting text from documents, processing the extracted information using AI models, and storing the results in the database.

**Requirements:**

1. **Django Application Development:**
   * Set up a Django project with the following features:
     + **User Management:** Implement user authentication and authorization.
     + **Document Upload:** Create a form to upload documents (PDFs, images).
     + **Document Processing:** Implement endpoints to trigger document processing and retrieve results.
2. **OCR Integration:**
   * Integrate an OCR library (e.g., Tesseract via pytesseract, EasyOCR) to extract text from uploaded documents.
   * Ensure support for multiple languages and handle different document formats.
3. **AI Processing:**
   * Use an AI library (e.g., Hugging Face Transformers, spaCy) to analyze the extracted text. This could involve:
     + **Named Entity Recognition (NER):** Identify and classify entities (e.g., names, dates, locations).
     + **Text Classification:** Categorize documents based on their content.
     + **Sentiment Analysis:** Analyze sentiment if applicable.
4. **Database Design:**
   * Design and implement a MySQL/MariaDB schema to store:
     + **Users:** Information about system users.
     + **Documents:** Metadata of uploaded documents.
     + **Extracted Text:** Text extracted from documents.
     + **AI Results:** Results from the AI processing, including any identified entities or classifications.
5. **Backend Integration:**
   * Create Django models and views to handle:
     + **Document Upload and Retrieval:** Store and retrieve documents and their processed results.
     + **AI Processing Results:** Save and display results from AI models.
6. **Frontend Implementation:**
   * + Allow users to upload documents.
     + Display processing progress and results.
     + Provide visualization for AI processing results (e.g., entity highlighting, classification charts).
7. **Testing and Validation:**
   * Implement unit tests and integration tests to ensure:
     + Correct functionality of OCR and AI processing.
     + Proper interaction between the Django application and the MySQL/MariaDB database.
     + Robust handling of edge cases and errors (e.g., failed OCR, AI processing errors).
8. **Documentation:**
   * Document the codebase, including:
     + Setup instructions.
     + API endpoints and their usage.
     + Explanation of the AI models and OCR integration.
     + How to extend or modify the system.

**Deliverables:**

* A fully functional Django application meeting the above requirements.
* Source code with comprehensive comments and documentation.
* A database schema design document.
* Test cases and results.
* Deployment instructions.

**Technical Skills Required:**

* Advanced Python programming.
* Expertise in Django framework and MySQL/MariaDB.
* Experience with OCR libraries (e.g., Tesseract, EasyOCR).
* Familiarity with AI/ML libraries (e.g., Hugging Face Transformers, spaCy).
* Knowledge of frontend technologies (e.g., HTML/CSS, JavaScript frameworks).

**Estimated Time:**

* 4-5 days depending on complexity and additional features.

**Additional Notes:**

* Ensure the system adheres to best practices in terms of security, performance, and scalability.
* Consider implementing asynchronous processing for handling large documents or heavy AI tasks.

**Submission Process** :  
 Create a project on github and provide a github link.

Note : Code should be written by the candidate. Copied code will not be considered.

[2:57 pm, 22/8/2024] +91 84272 57531: 2 automatic communications(Push Notifications) on the app to the users that have pending payments with the following templates -

1st message - one day before the instalment is due

Template:

Ciao! La tua rata per la NFS ACADEMY è in scadenza domani, ricordati di saldarla. Per maggiori informazioni consulta l'app di NFS Academy oppure contatta la segreteria della sede di Milano al n° 371 335 5310.

Grazie!

2nd message - ten days later if the payment has not been made or has only been made partially.

Template:

Ciao! La tua rata per la NFS ACADEMY è scaduta il [insert date], ricordati di saldarla. Per maggiori informazioni consulta l'app di NFS Academy oppure contatta la segreteria della sede di Milano al n° 371 335 5310.

Grazie!

[2:59 pm, 22/8/2024] +91 84272 57531: You will have to create your own scenario with this requirement and write the cide using python laguage. The testing will be on local host and that should show the success. After you complete the task please pass on the script to me so that I can check this. And also please Mention all the steps on the document outlining how to deploy this code on server. Thanks!