1. Project Description

The AI-PoweredStudy Assistant is a system process learning materials such as PDFs, books and lecture slides and transform them into structured interactive study resources.
The system will support:

- Automatic summarization of English content.
- Information extraction (keywords, entities, concepts) to highlight them.
- Search and retrieval (RAG) for an information within the learning materials ? (can act as a chatbot).
- Document processing (OCR + NLP for extracting text from scanned documents).
- Content generation (NLG) such as quizzes, study notes, and flashcards.

---

2. Machine Learning & NLP Models

We will adopt a two path approach:
Option 1 (possible baseline Models)

- Summarization: TextRank, TF-IDF extractive summarization, Seq2Seq with attention.
- Keyword Extraction: RAKE, TextRank, AraVec embeddings.
- Information Extraction (NER): Pre-trained BERT , DistilBERT or RoBERTa for NER.
- Quiz Generation: Template-based + similarity scoring with embeddings.
- Question Answering: Retrieval-based QA using FAISS + embeddings.

Option 2 (Fine-Tuning Small LLMs)
If resources permit, we will fine-tune small/medium Arabic language models:

- Summarization: Fine-tuned T5 / BART with LoRA.
- Question Answering & Chatbot: Fine-tuned BERT , DistilBERT or RoBERTa.
- Search & Retrieval (RAG): Integrate fine-tuned model with FAISS/Weaviate for semantic search.
- NLG (Notes & Quizzes): LLM-generated content (summaries, study notes, flashcards).

---

3. Tools & Technologies

- Programming Language: Python
- Data Handling & Preprocessing: NLTK, SpaCy Tesseract OCR (for scanned PDFs).
- Baseline NLP Models: TextRank, RAKE, Seq2Seq (PyTorch/TensorFlow).
- Deep Learning Frameworks: PyTorch, Hugging Face Transformers.
- Gen Ai: LangChain.
- Vector Search & Retrieval: FAISS, Weaviate, or ChromaDB.
- Deployment: FastAPI.
- Version Control & Tracking: GitHub, MLflow.
- Datasets (Arabic) (Not The Final datasets):
  - NER: CoNLL-2003 or OntoNotes 5.0.
  - Question Answering: SQuAD (Stanford Question Answering Dataset) 1.1 or 2.0.
  - Large-Scale Corpus: C4 (Colossal Clean Crawled Corpus) or the OSCAR English subset.
  - Custom Dataset: From textbooks, PDFs, and slides.

---

4. Milestones

- Milestone 1: Data collection (QA datasets, OCR from PDFs/slides) + preprocessing pipeline.
- Milestone 2: Implement baseline NLP models (summarization, keyword extraction, quiz generation).
- Milestone 3: Build QA chatbot + retrieval-based search system.
- Milestone 4: Fine-tuning experiments on small LLMs (if feasible).
- Milestone 5: Integration, lightweight deployment (FastAPI demo), and final report preparation.

---

5. Group Members & Roles:

Of course! Here is a detailed work breakdown for your team, assigning specific parallel tasks to each member to ensure everyone contributes across all project domains.

---

1. Ahmed Ibrahim abd el aziz (Team Leader)

Ahmed will focus on project leadership, baseline summarization, and the core advanced RAG model.

- **Project Management & Integration:**
  - Lead project planning and task coordination using GitHub Projects/Issues.
  - Manage the main GitHub repository, review pull requests, and oversee final system integration.
- **Summarization (Baseline):**
  - Implement the **TextRank** algorithm for extractive summarization.
  - Prepare and preprocess a portion of the custom dataset for this task.
- **Search & Retrieval (Advanced RAG):**
  - Take the lead on fine-tuning **BERT/DistilBERT for Question Answering**.
- **Content Generation:**
  - Woking on LLM generated study notes.

2. Youssuf Yasser Mohamed Rabie

Youssuf will handle a baseline model, advanced content generation, and its corresponding API.

- **Data Preprocessing:**
  - Prepare and preprocess the custom dataset specifically for **Content Generation** (e.g., creating text-to-quiz or text-to-notes pairs).
- **Summarization (Baseline):**
  - Implement a **Seq2Seq with attention** model in PyTorch/TensorFlow for abstractive summarization. This serves as a baseline for the fine-tuned T5.
- **Content Generation (Advanced NLG):**
  - Use the fine-tuned language models to generate **high-quality study notes and advanced quizzes**.
- **Search & Retrieval (RAG):**
  - Set up and manage the **FAISS/ChromaDB vector database**.
  - Implement the pipeline for chunking documents, generating embeddings, and indexing them in the vector store.
  -
- **Backend:**
  - Develop the FastAPI endpoint for all **Content Generation** features (e.g., /generate-quiz, /generate-notes).

3. Michael Samy Wiliam Ghobrial

Michael's focus is on Information Extraction (NER) and setting up the foundational vector database for the RAG system.

- **Information Extraction (Baseline & Advanced):**
    - Implement NER using a pre-trained **BERT**.
    - **Fine-tune** the BERT model on the OSIAN dataset for improved accuracy on your specific domain.
- **Tools:**
    - Set up and manage **MLflow** for tracking all model experiments and results.
    -

4. Mina Maher Fouad

Mina will handle the initial document ingestion (OCR) and baseline information extraction, and will build the main chat API and the deployment.

- **Document Processing:**
    - Implement the complete **OCR pipeline using Tesseract** to extract text from scanned PDFs and images.
- **Information Extraction (Baseline):**
    - Implement keyword extraction using algorithm.
- **Search & Retrieval (RAG):**
    - Develop the primary **FastAPI endpoint for the chatbot/RAG system** (/chat) that handles user queries.
- **Integration:**
    - Work with Ahmed and Michael to integrate the retrieval logic and the QA model into the chatbot API.
- **Deployment:**
    - Will deploy our project in a VM using Docker

5. Mohamed Sameh Elbably

Mohamed will be responsible for the advanced summarization model

- **Data Preprocessing:**
    - Prepare and preprocess the **dataset** for large-scale model fine-tuning.
- **Summarization (Advanced):**
    - Lead the fine-tuning of **T5 or BART with LoRA** for high-quality abstractive summarization.
- **Backend:**
    - Develop the FastAPI endpoint for the **Summarization** feature (/summarize).

6. Mahmoud Mohamed Mahmoud

Mohamed will focus on the foundational retrieval logic, model evaluation, and the document upload API.

- **Data Preprocessing:**
    - Prepare and preprocess the dataset for Ahmed's QA model.
- **Search & Retrieval (Baseline RAG):**
    - Implement the core **retrieval logic** using FAISS. This script will take a user query, embed it, and retrieve the most relevant text chunks from the vector database.
- **Model Evaluation:**
    - Develop and implement the complete **evaluation framework**. This includes writing scripts to calculate **ROUGE** for summarization, **F1-score** for NER, and **Accuracy/EM** for QA.
- **Backend:**
    - Develop the FastAPI endpoint for **document uploading and processing** (/upload), which will trigger the OCR and embedding pipeline.