**Titre du projet :**

**"EduLLM – Assistant IA d’Aide à l’Apprentissage des Étudiants BDIA avec un LLM"**

### Objectif du projet :

Créer une application web légère qui intègre un modèle de langage (LLM)\*\* pour assister les étudiants dans la compréhension des concepts enseignés en BDIA (Big Data, Machine Learning, IA, etc.).  
Le système sera capable de :

* répondre aux questions sur les cours,
* générer des résumés et des quiz,
* conseiller des ressources supplémentaires.

### **Pourquoi ce projet ?**

* Il permet aux étudiants d’**appliquer concrètement un LLM** sur un cas réel : l’éducation.
* Il touche plusieurs domaines : NLP, interface web, embedding, vectorisation de documents, etc.
* Il est faisable en 4 à 6 semaines avec des modules existants.

## ****Fonctionnalités principales de EduLLM :****

### 🧠 **Chatbot intelligent avec LLM personnalisé**

* + Répond à des questions sur les cours à partir de documents fournis (.pdf, .txt, etc.)
  + Le chatbot utilise un **LLM avec un système de RAG (Retrieval-Augmented Generation)** pour générer des réponses précises en se basant sur les cours.
  + Exemple : "Explique-moi l’algorithme K-means." → réponse extraite des notes de cours.

### 📄 **Chargement et résumé de documents**

* + L’utilisateur charge un cours (PDF ou texte brut).
  + Le LLM produit un résumé hiérarchisé selon le niveau de détail souhaité.
  + Résumé simplifié pour une meilleure compréhension.

### ❓ **Générateur de quiz intelligents**

* + Génère automatiquement des questions QCM à partir d’un cours donné.
  + Classe les questions selon la taxonomie de Bloom (connaissance, compréhension, application...).

### 📌 **Système de recommandation**

* + Analyse les lacunes de l’étudiant via ses interactions.
  + Propose des liens YouTube, articles ou autres documents pour approfondir.

## 🛠️ ****Architecture et technologies recommandées :****

| **Composant** | **Outils / Technologies** |
| --- | --- |
| **Frontend** | Streamlit, Flask + HTML/CSS |
| **Backend IA** | Python, LangChain, OpenAI API ou Llama 3 (via HuggingFace) |
| **Vector Store** | FAISS ou ChromaDB |
| **PDF Processing** | PyMuPDF, PDFMiner |
| **Embedding** | SentenceTransformers, OpenAI Embeddings, HuggingFace |
| **LLM** | OpenAI GPT-4-turbo, Mistral, LLaMA 3 (en local ou via API) |

## ****Déroulement du projet (exemple) :****

| **Semaine** | **Tâche** |
| --- | --- |
| S1 | Étude du besoin + collecte de documents de cours |
| S2 | Intégration du LLM via LangChain (ou autre framework) |
| S3 | Création du chatbot avec moteur de recherche sémantique (RAG) |
| S4 | Génération de quiz + interface simple |
| S5 | Tests utilisateurs + amélioration UX |
| S6 | Rendu final + rapport + présentation |