

Université Cadi Ayyad
École Supérieure De Technologie-Safi
Département : Informatique
Filière : Génie Informatique

Rapport de stage

Education & AI

Réalisé par :

Mlle. RGUIBI Farah
Mlle. ASSAF Aya

Encadré par :

M. EL ABDELLAOUI Said
Mme. ELKOURCHI Asmaa

ANNÉE UNIVERSITAIRE : 2025/2026

Cahier de charge

1 Contexte

Avec l'essor de l'intelligence artificielle, de nouvelles opportunités s'ouvrent dans le domaine de l'éducation. Ce projet vise à développer une plateforme éducative intelligente capable de générer automatiquement des supports pédagogiques (résumés, QCM, fiches de révision), de fournir une assistance via un chatbot éducatif, et de suivre la progression des apprenants. La plateforme comportera une version web et une application mobile accessible à tous.

2 Objectifs du projet

- Générer automatiquement des supports éducatifs à partir de documents PDF.
- Intégrer un chatbot éducatif interactif basé sur des modèles NLP.
- Proposer des QCM adaptés et des fiches de révision.
- Mettre en place un système de répétition espacée pour faciliter la mémorisation.
- Déployer une interface web et mobile intuitive.
- Suivre la progression des utilisateurs (enseignants et apprenants).

3 Choix des modèles IA

Nous avons opté pour plusieurs modèles de traitement automatique du langage naturel (NLP), chacun spécialisé dans une tâche précise.

BART (Facebook AI)

Utilisation : Résumé automatique de documents.

Avantages :

- Excellente qualité de résumé grâce à l'architecture encoder-decoder.
- Open-source, pré-entraîné sur de grands corpus.
- Bon compromis entre précision et vitesse.

Inconvénients :

- Relativement lourd en ressources.
- Moins rapide que les modèles distillés.

DistilBERT (Hugging Face)

Utilisation : Questions-réponses (QéR), extraction d'informations.

Avantages :

- Plus rapide et plus léger que BERT.
- Performant pour les tâches de QéR.
- Facile à intégrer dans une API Flask.

Inconvénients :

- Moins précis que BERT sur certains jeux de données.
- Limitation sur la taille des textes d'entrée.

MarianMT (Helsinki-NLP)

Utilisation : Traduction automatique multilingue.

Avantages :

- Supporte de nombreuses paires de langues.
- Modèle optimisé pour la traduction.
- Léger et rapide.

Inconvénients :

- Moins fluide que certains modèles commerciaux (DeepL, Google).
- Moins adapté aux textes techniques ou spécialisés.

Bard (Google)

Utilisation : Génération avancée de questions pour QCM.

Avantages :

- Capacité à générer des questions logiques et pertinentes.
- Bonne compréhension contextuelle.
- Utile pour des QCM d'un niveau plus élevé ou personnalisé.

Inconvénients :

- Non open-source.
- Dépendance à une API externe.

Flan-T5 (Google)

Utilisation : Génération de texte, résumés, dialogue conversationnel.

Avantages :

- Entraîné sur de nombreuses consignes (instruction tuning).
- Bon générateur de texte polyvalent.
- Bonne compatibilité avec Hugging Face.

Inconvénients :

- Consomme beaucoup de mémoire.
- Moins spécialisé pour certaines tâches (ex : QéR).

Mistral (via OpenRouter + LangChain)

Utilisation : Réponses intelligentes à partir de documents PDF, génération de texte cohérente, conversation contextuelle avancée.

Avantages :

- Modèle léger et performant pour les tâches de génération et de raisonnement.
- Compatible avec LangChain pour les chaînes de traitement complexes.
- Très utile avec des embeddings pour répondre à partir de contexte PDF.
- Peut être utilisé via OpenRouter sans serveur local.

Inconvénients :

- Nécessite une API key tierce (OpenRouter).
- Moins précis que GPT-4 pour des questions très ouvertes.
- La qualité dépend fortement de la qualité des chunks extraits.

4 Technologies choisies

4.1 Backend IA/NLP — Flask + Transformers

Avantages :

- Léger et simple à coder.
- Idéal pour créer des APIs (résumé, QCM, chatbot).
- Compatible avec la librairie Hugging Face.

Inconvénients :

- Non asynchrone, donc limité pour les grosses charges.
- Moins adapté aux architectures complexes.

4.2 Application mobile — React Native

Avantages :

- Un seul code pour Android et iOS.
- Compatible avec Expo.
- Large écosystème (UI, navigation, plugins).

Inconvénients :

- Moins performant qu'une app native.
- Bugs possibles avec des modules natifs spécifiques.

4.3 Application Web — React / Next.js

Avantages :

- Framework moderne, rapide et performant.
- SSR (Server Side Rendering) avec Next.js pour un meilleur SEO.
- Code réutilisable avec React Native.

Inconvénients :

- Courbe d'apprentissage plus longue.
- Plus difficile à configurer pour les débutants.

4.4 Base de données — MySQL

Avantages :

- Robuste, relationnelle, bien standardisée.
- Bonne intégration avec Flask via SQLAlchemy.
- Open-source.

Inconvénients :

- Moins flexible que NoSQL pour des données très dynamiques.
- Configuration initiale plus complexe.

5 Différence entre Django et Flask

Flask

- Microframework léger, minimaliste, plus flexible.
- Idéal pour des APIs simples et des projets modulaires.
- Moins de configuration initiale.
- Parfait pour un projet de recherche ou prototype IA rapide.

Django

- Framework full-stack tout-en-un.
- Intègre ORM, système d'authentification, panneau d'administration.
- Plus adapté à des projets complexes orientés base de données.
- Configuration initiale plus lourde.