

TELECOM SudParis

Projet Cassiopée - Equipe 65
Plateforme d'Apprentissage en IA et Data Science

Livrable Métier



BAKKALI Sami
GRIFFON Matthieu
RAMSIS Adam
THOMAS Alicia

Encadrants :
TOUNKARA Thierno, VOYER Robert

2024



Plan

1. Introduction.....	3
2. Fichier Source.....	4
2.1 Repository GitHub.....	4
2.2 Structure et description des fichiers principaux.....	4
3. Guide d'installation.....	5
3.1 Pré requis.....	5
3.2 Commande d'installation.....	5
4. Guide d'utilisation.....	7
4.1 Interface utilisateur.....	7
5. Documentation technique.....	9
5.1 Architecture de la plateforme.....	9
5.2 Intégration de l'IA.....	10
A. Annexes.....	11
A.1 Perspectives envisagées.....	11
A.2 Contacts.....	12

1. Introduction

Le projet Cassiopée présenté a pour ambition de développer une plateforme éducative pour l'enseignement de l'IA et de la data science, en utilisant les dernières avancées en IA telles que les modèles de langage : ChatGPT-4 ou d'autres LLM. L'objectif est d'offrir, via cette plateforme, une expérience d'apprentissage immersive et interactive aux élèves managers (avec des exercices interactifs et des sessions théoriques et pratiques ainsi qu'un chatbot et un suivi des élèves automatique et autonome tel un professeur particulier virtuel).

Les concepts d'IA et de data sciences intégrées dans la plateforme ont été, dans un premier temps, choisis par nos soins, priorisant l'impact pédagogique et l'intérêt des élèves managers. En effet, l'intelligence artificielle et la data science sont des domaines d'apprentissage essentiels pour les futurs managers: L'IA permet par exemple d'automatiser des tâches complexes, tandis que la data science permet de prendre des décisions basées sur des analyses prédictives. Deux possibilités s'offrent alors à l'étudiant sur notre plateforme: apprendre à partir de cours déjà implémentés provenant de professeurs et d'encadrants spécialisés dans ces domaines ou télécharger directement sur la plateforme leurs cours personnels afin de travailler dessus et de mieux les comprendre.

De plus, au cours du projet, nous avons cherché à comparer plusieurs types de LLM ayant des applications concrètes pour notre plateforme (génération de textes, de vidéos, de traduction, d'analyse...) afin d'identifier l'efficacité des différentes approches d'enseignements, de s'améliorer et d'optimiser la plateforme. Après analyse, nous avons intégré l'IA dans notre plateforme de telle sorte à créer un support d'apprentissage personnalisé et adapté à tout type d'élève. Notre solution dispose donc d'un chatbot basé sur les API ChatGPT permettant un soutien sur les parties théoriques et pratiques en text-to-text et image-to-text. L'objectif à long terme serait l'ajout d'exercices interactifs permettant d'imiter l'approche d'un professeur particulier avec un suivi personnel, des feedbacks instantanés et des recommandations personnalisées.

2. Fichier Source

2.1 Repository GitHub

Le code source de la plateforme Cassiopée est disponible sur GitHub. Le repository inclut tous les fichiers nécessaires pour exécuter et développer la plateforme moyennant une clé API pour le fonctionnement du chatbot et l'installation des bibliothèques nécessaires (Voir section 3). Vous pouvez accéder au repository via le lien suivant : <https://gitlabens.imtbs-tsp.eu/matthieu.griffon/cassiopee.git>

2.2 Structure et description des fichiers principaux

Le repository GitHub est structuré en backend et frontend de la manière suivante :

- **server.py** : Script Python qui configure et exécute le serveur backend de l'application. Il gère notamment les requêtes et interactions avec les modèles d'IA (la clé API), la base de données et le frontend (partie HTML et Javascript).
- **init_db.py** : Script Python pour initialiser la base de données SQLite, utilisée par la plateforme, contenant les informations et données nécessaires à son fonctionnement (cours, exercices et gestion de profils notamment)
- **/cours** : Répertoire contenant les fichiers de cours en PDF et autres formats.
- **/exercices** : Répertoire contenant les exercices interactifs et leurs solutions.
- **/templates** : Répertoire contenant les modèles HTML
- **index.html** : Fichier HTML principal qui permet la formation de la plateforme.
- **/static** : Répertoire contenant les fichiers statiques comme les feuilles de style CSS et les scripts JavaScript.
- **style.css** : Fichier CSS contenant les styles pour l'interface utilisateur
- **script.js** : Fichier JavaScript responsable de l'interactivité sur la page web.
- **user.txt**: Fichier comprenant des profils utilisateurs déjà intégré dans la base de donnée
- **requirement.txt**: Fichier permettant d'installer les dépendances et bibliothèques nécessaires au projet
- **.env**: Fichier utilisé pour stocker des variables d'environnement de manière sécurisée et organisée (tel que la clé API)
- **/.venv**: Répertoire configuré pour gérer un environnement virtuel python si besoin
- **/Documentation**: Répertoire contenant les fichiers de documentation du projet
- **README.md** : Fichier de documentation pour l'utilisation du projet.

3. Guide d'installation

3.1 Pré requis

Avant de commencer l'installation de la plateforme, assurez-vous d'avoir les éléments suivants :

- **Python 3.10 ou supérieur** : Le langage de programmation utilisé pour le développement du backend
- **pip** : L'outil de gestion des packages Python (afin d'installer les dépendances et bibliothèques)
- **Git** : Pour cloner le repository GitHub.
- **Flask** : Le micro-framework utilisé pour créer le serveur web.
- **Éditeur de code**: Pour éditer les fichiers HTML, CSS et JavaScript si besoin.
- **Clé API ChatGPT** : Nécessaire pour intégrer le chatbot utilisant ChatGPT-4

3.2 Commande d'installation

Voici les étapes à suivre pour pouvoir utiliser notre plateforme (version linux à adapter sous Windows):

1. Cloner le repository GitHub dans un nouveau dossier

```
mkdir NouveauDossier  
cd NouveauDossier  
git clone https://gitlabens.imtbs-tsp.eu/matthieu.griffon/cassiopee.git  
cd DossierCloné
```

2. Vérifier la présence de tous les documents

```
ls  
→ si erreur: vérifier que vous êtes placés sur la bonne branche main (git branch  
puis git checkout <nom_de_la_branch>)
```

3. Installer les dépendances et bibliothèques nécessaires

```
pip install -r requirements.txt
```

4. Configurer la clé API ChatGPT

→ **Obtenez une clé API ChatGPT** depuis OpenAI puis ajoutez là dans un fichier .env a la racine du projet à la ligne: OPENAI_API_KEY='Votre clé'

5. Initialiser la base de données

```
python init_db.py
```

6. Lancer le serveur :

```
python server.py
```

→ **Le serveur est maintenant en cours d'exécution et accessible** via `http://127.0.0.1:5000` dans votre navigateur web.

4. Guide d'utilisation

4.1 Interface utilisateur

La plateforme Cassiopée propose une interface utilisateur intuitive, immersive et interactive. Voici les principaux éléments de l'interface :

- **Page d'accueil:**
 - **Bannière principale** : Présente le nom de la plateforme.
 - **Menu de navigation** : Un menu horizontal en haut de la page permet l'accès rapide aux sections Cours, Exercices et À propos.
 - **Section de profil**: Permet de se connecter ou de s'inscrire sur la plateforme. Permet, une fois connecté, de se déconnecter.

- **Menu de navigation** :
 - **Cours** : Accès à la bibliothèque de cours disponibles.
 - **Exercices** : Accès aux exercices interactifs.

- **Section Cours** :
 - **Liste de cours** : Présente tous les cours disponibles, avec leurs descriptions et leur niveau de difficulté. Une option de filtrage permet de chercher un cours par son nom.
 - **Visualisation de cours** : Les cours peuvent être visualisés directement sur la plateforme sous forme de PDF ou autres formats.
 - **Téléchargement** : Les utilisateurs peuvent télécharger leurs propres documents de cours (et y associer un nom, une description et un niveau de difficulté) à la bibliothèque pour travailler dessus.

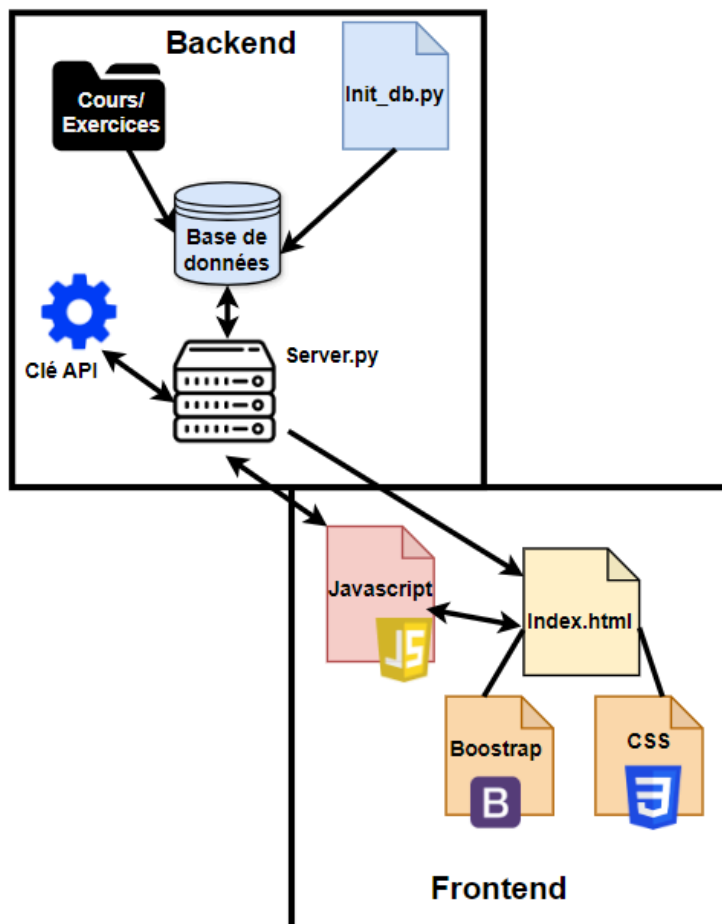
- **Section Exercices** :
 - **Liste d'exercices** : Présente tous les exercices disponibles, avec leurs descriptions et leur niveau de difficulté. Une option de filtrage permet de chercher un cours par son nom.
 - **Visualisation des exercices**: Les exercices peuvent être visualisés directement sur la plateforme sous forme de PDF ou autres formats.
 - **Téléchargement** : Les utilisateurs peuvent télécharger leurs propres documents d'exercices (et y associer un nom, une description et un niveau de difficulté) à la bibliothèque pour travailler dessus.

- **Chatbot :**
 - **Interface de conversation :** Une fenêtre de chat intégrée sur les parties cours et exercices permet aux utilisateurs de poser des questions ou de coller une image (tel un screenshot) afin d'obtenir des réponses en temps réel text-to-text ou image-to-text.
 - **Historique des conversations :** Les utilisateurs peuvent consulter l'historique de leurs interactions avec le chatbot (cet historique est dépendant du profil utilisateur)
 - **Feedback :** Le chatbot retourne, en plus, à chaque prompt un feedback personnalisé sur la question de l'utilisateur permettant de le renvoyer vers des cours ou chapitres associées (parcours d'apprentissage personnalisé) mais également de lui poser des questions de type QCM pour voir sa compréhension.

5. Documentation technique

5.1 Architecture de la plateforme

L'architecture de la plateforme Cassiopée est conçue pour être modulaire facilitant ainsi l'ajout de nouvelles fonctionnalités.



- **Backend :**

- **Base de données SQLite** : Une base de données est intégrée dans la plateforme pour stocker les informations sur les utilisateurs, les cours, et les exercices. Elle est initialisée et manipulée via le script `init_db.py`.
- **Flask** : Le framework web est utilisé pour développer le backend de la plateforme et notamment `server.py`. Il gère les requêtes HTTP avec le frontend et l'interface utilisateur mais également les interactions avec la base de données, la gestion de session et la clé API.
- **API ChatGPT** : Intégrée via le fichier `server.py`, elle permet de d'offrir une assistance en temps réel aux utilisateurs. La clé API est configurée pour interagir avec le modèle ChatGPT-4.

- **Frontend :**
 - **HTML/CSS/JavaScript** : Ces fichiers sont utilisés pour créer l'interface utilisateur. Le fichier index.html constitue la plateforme dont le style est défini dans style.css. Des scripts en JavaScript permettent l'interactivité de la plateforme.
 - **Bootstrap** : Une bibliothèque front-end est utilisée pour créer des interfaces utilisateur réactives et modernes. Bootstrap est intégré dans le fichier index.html pour améliorer le design et la réactivité de la plateforme.

5.2 Intégration de l'IA

L'intégration de l'IA dans la plateforme Cassiopée se fait principalement à travers l'utilisation de l'API ChatGPT, qui est utilisée pour le chatbot interactif. Voici les étapes et considérations techniques pour cette intégration :

- **API ChatGPT :**
 - **Configuration** : La clé API est configurée dans le fichier server.py à la ligne `openai.api_key = 'VOTRE_CLE_API_OPENAI'`. Cette clé permet d'accéder aux fonctionnalités de ChatGPT.
 - **Fonctionnement** : Lorsqu'un utilisateur interagit avec le chatbot, une requête est envoyée à l'API ChatGPT avec le texte et/ou l'image fourni par l'utilisateur. L'API renvoie alors une réponse en markdown qui est ensuite affichée à l'utilisateur via l'interface de chat.
 - **Modes de réponse** : Le chatbot peut répondre en mode text-to-text pour les questions textuelles, et en mode image-to-text pour les images collées par les utilisateurs. L'utilisateur peut également écrire un prompt comprenant des images et du texte en même temps.
- **Gestion des requêtes :**
 - **Envoi des requêtes** : Les requêtes sont envoyées à l'API ChatGPT via des appels HTTP à partir du serveur backend. En fonction du LLM choisi (dans notre cas ChatGPT-4), cette requête peut coûter quelques centimes proportionnellement au nombre de tokens.
- **Gestion des sessions :**
 - **Configuration** : Flask-Session est configuré dans le fichier server.py pour gérer les sessions côté serveur (plus sécurisé). Cela permet d'associer l'historique des prompts ChatGPT à un profil unique.
 - **Utilisation** : Les données de session peuvent être stockées et récupérées en utilisant l'objet session de Flask. En effet, `session['key'] = 'value'` stocke une valeur dans la session, et `value = session.get('key')` récupère cette valeur.

A. Annexes

A.1 Perspectives envisagées

L'évolution future de la plateforme Cassiopée comprend plusieurs axes d'amélioration et de développement que nous avons envisagés (débutés sur la branche develop de notre git). Voici les principales perspectives futures du projet :

- **Développement d'exercices interactifs avancés :**
 - Intégrer des exercices interactifs avec des feedbacks instantanés tels que la correction et les explications des erreurs mais aussi la création d'exercices plus adaptés et personnalisés à l'utilisateur, par l'IA, en fonction des difficultés de l'élève.
 - Lier ces exercices aux cours présents dans la bibliothèque de la plateforme pour une meilleure cohérence et un parcours pédagogique

- **Surveillance des progrès et personnalisation de l'apprentissage :**
 - Suivre, de façon autonome et automatique grâce à l'IA, les progrès des utilisateurs à chaque session théorique et pratique afin de leur fournir des recommandations et des parcours d'apprentissage personnalisés (ce qui permettrait d'imiter l'approche d'un professeur particulier)
 - Faire des suggestions et des redirections vers d'autres contenus en fonction des difficultés de l'élève
 - Introduire des éléments ludiques comme des jeux/défis et des systèmes de récompense pour encourager et motiver les élèves à travailler leur session de cours et d'exercices.

- **Intégration de nouveaux modèles de langage (LLM) :**
 - Incorporer des modèles de langage plus avancés, spécialisés et open-source pour enrichir les interactions et fournir des réponses encore plus précises.
 - Comparer plusieurs types de LLM ayant des applications concrètes pour la plateforme (génération de textes, de vidéos, de traduction, d'analyse) afin d'identifier les approches d'enseignement les plus efficaces et optimiser la plateforme.
 - Intégrer des fonctionnalités supplémentaires telles que le speech-to-text, le text-to-speech, la génération de vidéos... pour diversifier et améliorer les outils pédagogiques disponibles.

- **Paramétrage avancé du chatbot :**
 - Ajouter la possibilité de définir des paramètres pour chaque requête envoyée au chatbot, comme le niveau de précision de la réponse, la longueur de la réponse, le langage verbal...

- **Sécurité et gestion des profils :**
 - Renforcer la sécurité de la plateforme pour protéger les données des utilisateurs et garantir un environnement d'apprentissage sûr.
 - Améliorer la gestion des profils avec un accès à ses paramètres pour personnaliser son expérience (cours favoris, langage de préférence...)

- **Optimisation de l'interface utilisateur :**
 - Continuer à améliorer l'interface utilisateur pour assurer une navigation plus fluide et intuitive, en tenant compte des retours des utilisateurs.

A.2 Contacts

Pour toute question ou demande d'assistance, veuillez contacter l'un des membres de l'équipe du projet Cassiopée :

- **Nom :** Matthieu Griffon
 - **Email :** matthieu.griffon@telecom-sudparis.eu
 - **Téléphone:** 0645113732
- **Nom :** Alicia Thomas
 - **Email :** alicia.thomas@telecom-sudparis.eu
 - **Téléphone:** 0695712701
- **Nom :** Adam Ramsis
 - **Email :** adam.ramsis@telecom-sudparis.eu
 - **Téléphone :** 0618774245
- **Nom :** Sami Bakkali
 - **Email :** sami.bakkali@telecom-sudparis.eu
 - **Téléphone :** 0755645415