



Rapport de Stage

20 juin 2023

Enseignant Référent : Philippe Roose

Tuteur : Mamadou Lamine Gueye

Maître de stage : Ernesto Exposito Garcia

Etudiant : Ewan Burasovitch

Remerciements

La réalisation de ce rapport n'aurait pas été possible sans la contribution de plusieurs personnes à qui je souhaite exprimer ma profonde gratitude.

Je tiens d'abord à adresser toute ma reconnaissance à mon tuteur, Mamadou Gueye, pour son soutien constant, ses précieux conseils et son expertise en ingénierie pédagogique, qui ont été d'une aide inestimable tout au long de ce projet.

Je remercie également mon maître de stage, Ernesto Exposito, ainsi que Nicolas EVAIN pour l'encadrement exceptionnel et les conseils avisés. Leur soutien et expertise ont été déterminants pour la progression et la qualité du projet Connect Unita.

Je souhaite exprimer ma gratitude à l'ensemble de l'équipe de l'UPPA impliquée dans Connect Unita, pour leur collaboration, leur soutien et leur esprit d'équipe, qui ont grandement contribué à la réussite de ce projet.

Je tiens à remercier particulièrement les acteurs externes qui ont joué un rôle crucial dans le développement et la mise en œuvre de notre projet :

Alexandre Paboeuf, pour son travail essentiel sur l'intégration multilingue des vidéos, rendant nos contenus accessibles à un public international.

Enfin, je souhaite remercier tous ceux qui, de près ou de loin, ont contribué à la réussite de ce projet par leur soutien, leurs conseils et leurs encouragements.

Remerciements	2
Introduction	5
Contexte	6
Equipe	7
Projet	8
Environnement de travail	9
Sujet	10
Objective	10
Problématique	10
La veille concurrentielle	11
Comparaison des solutions existantes	11
Solution Proposé	13
Vue d'ensemble de la solution	13
Navigation sur le site	14
Fonctionnalités	15
Vidéo interactives avec des questions à choix simple.	15
Vidéo interactives avec des questions à choix multiple.	18
Génération de questions via la vidéo	19
Tutoriel	20
Comment créer une Vidéo	20
Intégration dans Moodle	21
Architecture du Projet	23
Conception des différents services	23
Diagramme de cas d'utilisation	23
Diagramme de séquence Questions Choix Simple	24
Diagramme de séquence QuestionGenerator	25
Diagramme de séquence Accès Tuto & FAQ	26
Schéma de l'architecture	27
Architecture VScode	29
Revu de code	30
API python fichier de l'api, app.py :	30
Imports et Initialisation :	31
Route principale (/) :	31
Route /interactive_video_multiple_choice_question :	32
Route /interactive_video_simple_choice_question :	33
Route /upload_video :	33
Gestion des erreurs :	34
Exécution de l'application :	34
API python fichier h5p_generator.py :	35
Paramètres de la fonction :	35
Définition des paramètres stricts :	35
Manipulation des fichiers et des chemins :	36
Chargement et manipulation des données :	36
Modification du fichier JSON de contenu (content.json) :	36

	4
Paramètres et options pour chaque interaction :	37
Écriture et finalisation du contenu JSON :	37
Création du fichier H5P :	38
API python fichier generate_request.py :	38
Import des modules :	38
Définition de l'application Flask :	39
Fonction generate_questions_and_answers :	39
Fonctions utilitaires (remove_special_characters, clean_quotes, save_csv, check_file_existence) :	39
Fonction process_video_to_csv :	41
Point d'entrée principal (__name__ == "__main__") :	41
API python fichier utils.py :	42
Classe WhisperModel	42
Fonction translate_text	43
Fonction load_data_file	43
Fonction video_speech_to_text	44
Fonction modify_video_restrictions	44
Fonction zip_files_in_folder	44
Fonction video_duration	45
Fonction extract_audio	45
Fonction modify_video_url	46
Fonction add_subtitles	46
Fonction vtt_files_creation	47
Fonction find_previous_duration	49
Planification du Projet	50
Diagramme de Gantt	50
Tableau RACI	51
Outils de Collaboration	52
Mes travaux	53
Tâches réalisées	53
Tâches Backend	53
Tâches Fronted	55
Compétences apportées	56
Création du Projet sur GitLab	57
Comment partager du code	62
Utilisation de Docker	64
Pourquoi Docker	64
Fichier docker-compose.yml	64
Fichier dockerfile	65
Récapitulatif des Commandes Docker	67
Dépendance importer	68
Choix et Intégration des Fonctionnalités	69
Toutes les dépendances ont été choisies en comparant les différentes solutions trouvées lors de nos recherches. Nous avons donc effectué un comparatif entre les solutions existantes.	69
Speech-To-Text	69
Traduire le texte	69
Générateur de Questions/ Réponses IA	70
Choix de l'IA pour création Vidéo Avatar	70
Conclusion	71

Introduction

Je vais commencer par me présenter. Je suis Ewan Burasovitch, étudiant en BUT informatique. Dans le cadre de cette formation, j'ai cherché à réaliser un stage afin de compléter ma deuxième année. Ce stage vise à consolider les compétences acquises au sein de ma formation, telles que le développement et l'optimisation d'applications, l'administration des systèmes informatiques, la gestion des données, ainsi que le travail en équipe et la gestion de projet.

Ce stage représente également une opportunité pour moi de gagner ma première expérience professionnelle dans le domaine que je souhaite exercer à l'avenir.

Suite à un entretien avec mon actuel maître de stage et le chef des relations internationales de l'UPPA, j'ai été accepté pour effectuer un stage au sein de l'équipe Connect Unita pour une durée de 10 semaines, du 21/05/2024 au 26/07/2024, sur un projet spécifique présenté lors de l'entretien.

Le stage est une période temporaire de mise en situation en milieu professionnel, au cours de laquelle le stagiaire acquiert des compétences professionnelles et applique les connaissances de sa formation, en vue de l'obtention d'un diplôme ou d'une certification, tout en favorisant son insertion professionnelle. Le stagiaire se voit confier une ou plusieurs missions conformes au projet pédagogique défini par son établissement d'enseignement et approuvées par l'organisme d'accueil.

Le programme du stage est établi conjointement par l'établissement d'enseignement et l'organisme d'accueil, en fonction du programme général de la formation dispensée.

Contexte

Equipe

Lors de ce stage, j'ai collaboré avec deux équipes distinctes. La première équipe, à laquelle je suis rattaché, est Unita. Cette équipe est une alliance de six universités soutenue par l'Union Européenne, dont l'UPPA est l'un des membres fondateurs. Lancée en novembre 2020 pour une durée de trois ans, cette alliance participe activement à la construction de l'espace européen de l'enseignement supérieur. Mon stage s'inscrit dans le cadre de l'un des nombreux projets en cours au sein de cette alliance.

Ce projet, porté par Ernesto Exposito – vice-président en charge des relations internationales à l'UPPA –, a été officiellement lancé le 1er février 2022 pour une durée de trois ans.

L'objectif principal de ce projet est le développement d'une plateforme collaborative et d'une méthodologie pour l'innovation dans l'enseignement et l'apprentissage qui non seulement vise à atteindre le succès dans l'acquisition de connaissances et de compétences mais permet également de briser les barrières de la distance géographique, linguistique, temporelle et sociale favorisant ainsi l'inclusion et la coopération internationale.

Plus précisément, notre projet répondra aux objectifs suivants :

- Répondre aux besoins d'intégration des systèmes de gestion de la formation (LMS) et des espaces physiques d'apprentissage des étudiants et des enseignants des institutions de l'alliance UNITA et des futurs partenaires associés, afin d'éliminer les barrières géographiques, linguistiques et temporelles et de soutenir les espaces d'apprentissage numérique collaboratif. Dans ce but, une plateforme d'intégration offrant une solution innovante de services de traduction automatique et de distribution géographique des contenus d'apprentissage et de suivi synchrone ou asynchrone des étudiants par les enseignants dans le cadre de coopérations internationales sera conçue et développée.
- Offrir un cadre unifié pour guider les enseignants dans la transformation numérique et pédagogique lors de la création ou de l'adaptation de ressources et d'activités d'apprentissage afin d'éliminer les obstacles qui limitent la collaboration internationale dans l'enseignement et l'apprentissage. Ce cadre fournira à la fois une méthodologie et un modèle techno-pédagogique visant à intégrer les dimensions

géographiques, multilingues et hybrides de l'apprentissage et à favoriser la collaboration internationale dans la construction et la diffusion des formations.

- Créer un Living Lab international d'enseignement et d'apprentissage pour les enseignants et les étudiants de l'alliance UNITA, ainsi que pour la communauté internationale, grâce à la co-conception et la coproduction des enseignements, orientés à la fois au transfert et à la multiplication des compétences en matière d'innovation technologique et pédagogique ainsi que sur les thématiques prioritaires pour l'alliance UNITA (patrimoine culturel, énergies renouvelables et économie circulaire). Le living lab d'enseignement et d'apprentissage permettra également le développement de mobilités virtuelles entre les membres du consortium UNITA ainsi que d'autres membres externes et offrira un espace de coopération en matière d'innovation pédagogique et numérique.

Le living lab permettra le développement de mobilités virtuelles pour les membres du consortium UNITA comme pour des personnes extérieures à l'alliance. Il offrira un espace de coopération en matière d'innovation pédagogique et numérique.

C'est ici dans le cadre du deuxième point que le projet qui porte mon stage va s'intégrer.

Puis une autre équipe entre en collaboration dans ce projet portée par un second stagiaire. Alexandre, cette équipe se nomme IREKIA

Le projet IREKIA, lancé par l'Université de Pau et des Pays de l'Adour (UPPA), s'inscrit dans l'initiative "Excellence sous toutes ses formes" de l'Agence Nationale de la Recherche, dans le cadre du quatrième volet du Programme d'Investissements d'Avenir (PIA). L'objectif principal de ce projet est d'élargir l'offre éducative et de renforcer les capacités de recherche dans la région du Pays basque, tout en intégrant de manière plus étroite les formations dans le tissu économique local. IREKIA se distingue par son approche innovante, en se focalisant notamment sur l'hybridation des enseignements et l'utilisation des Learning Analytics pour optimiser le suivi pédagogique.

Equipe IREKIA

LABEL
EXCELLENCE(S)
IREKIA

Christian La Borderie
Responsable Scientifique

Stéphanie Chalivoix
Chargée du suivi de
projet IREKIA

Cécile Garcia
Chargée du développement
des fablabs

Sandrine Pottier
Chargée du learning center
IREKIA

Manon Harrewyn
Responsable FabLab Eaux

Camille Contreras
Responsable de l'Observatoire

Alexandre Paboeuf
Stagiaire Hybridation IA/LA

Figure 1 : Organigramme IREKIA.

Projet

Ce stage en informatique s'inscrit dans le cadre du projet Connect-UNITA et s'articule principalement autour de l'hybridation des enseignements dans le domaine de l'enseignement supérieur. Plus précisément, il vise à mettre en place des approches pédagogiques novatrices facilitant l'analyse des traces d'apprentissage, collectées tout au long du processus d'apprentissage de l'étudiant.

Afin de collecter ces traces, les ressources pédagogiques doivent être adaptées pour s'intégrer de manière fluide dans l'écosystème numérique, notamment le Learning Management System (LMS). Lors des interactions digitales avec le LMS, l'étudiant génère des traces en interagissant avec les ressources d'apprentissage, connues sous le nom d'objets d'apprentissage (Learning Objects).

Ces traces constituent une mine d'informations exploitables dans le cadre des Learning Analytics, offrant ainsi aux apprenants et enseignants une vision approfondie des processus d'apprentissage. L'analyse des traces propose quatre niveaux d'analyse : descriptif, diagnostic, prédictif et prescriptif, fournissant des réponses aux questions cruciales sur ce qui s'est passé, pourquoi cela s'est produit, que va-t-il se passer, et que peut-on faire pour améliorer les résultats.

Cependant, l'hybridation des enseignements demeure une tâche complexe, entravée par le manque de temps des enseignants pour effectuer cette transition digitale. Grâce aux avancées de l'intelligence artificielle, une automatisation partielle de la création des ressources pédagogiques devient possible. L'idée sous-jacente est que l'enseignant se concentre sur l'essence et le contenu de ses ressources pédagogiques, tandis que l'IA intervient pour transformer ces objets d'apprentissage dans un format compatible avec le LMS, permettant ainsi la collecte des traces d'apprentissage.

Pour orchestrer cette intégration, le projet s'appuie sur HTML 5 Package (H5P), une solution open source. L'objectif du stage est de développer une application permettant, à partir d'un design préétabli de ressources pédagogiques, la création automatique d'un objet H5P. L'intelligence artificielle sera employée pour intégrer des fonctionnalités telles que la dimension multilingue et l'intégration automatique de questions interactives. Ces objets d'apprentissage H5P pourront être directement intégrés dans le LMS, favorisant ainsi les Learning Analytics via la collecte des traces d'apprentissage, conformément au standard xAPI. Ces actions sont développées au sein de l'alliance UNITA et plus particulièrement dans le cadre du projet Connect-UNITA et à la tâche 3.3 (UNITA Innovation in Teaching & Learning Initiative) concernant les aspects multi-langue, pédagogiques et relatifs aux Learning Analytics.

Environnement de travail

Le stage se déroule dans les locaux de l'université de l'UPPA. Nous disposons de notre propre ordinateur avec un écran supplémentaire. Nous travaillons en binôme dans une salle dédiée à ce projet. La proximité des doctorants qui supervisent le

projet facilite la transmission des informations, rendant les échanges plus naturels et accélérant la réalisation des tâches.

Une salle de réfectoire équipée du nécessaire est à notre disposition, et la proximité des infrastructures étudiantes est très appréciée.

Sujet

Ce stage se déroule dans le cadre du projet Connect-UNITA, une initiative européenne portée par l'UPPA. Il vise à développer une application capable d'automatiser la création de ressources pédagogiques interactives grâce à l'intelligence artificielle. Ces ressources seront intégrées dans les systèmes de gestion de l'apprentissage (LMS) pour faciliter l'analyse des traces d'apprentissage et promouvoir l'innovation pédagogique. Le stagiaire travaillera sur un environnement Docker, impliquant la construction et le déploiement de services web et API.

Objectif

Activités confiées :

1. Analyse des designs pédagogiques
2. Automatisation des questions interactives
3. Développement d'une solution intégrant l'IA générative
4. Tests et déploiement
5. Compatibilité xAPI

Compétences à acquérir ou à développer :

- Maîtrise des technologies web, en particulier HTML, CSS et JavaScript
- Intelligence artificielle
- Administration de systèmes de gestion de bases de données
- Systèmes et réseaux informatiques

Problématique

Comment les nouvelles technologies, telles que l'intelligence artificielle et les Learning Analytics, ainsi qu'une intégration multilingue, peuvent-elles être utilisées pour optimiser le suivi pédagogique, enrichir les méthodes d'enseignement, et améliorer l'engagement et la réussite des étudiants ?

La veille concurrentielle

La veille concurrentielle des solutions existantes révèle plusieurs initiatives similaires visant à intégrer des technologies avancées dans l'éducation et l'edtech, tout en garantissant la souveraineté et la sécurité des données. Le projet fournit une API qui peut être intégrée aux logiciels éducatifs gratuits couramment utilisés tels que Moodle et BigBlueButton (BBB). Ces plateformes open source offrent une grande flexibilité pour répondre aux besoins spécifiques des utilisateurs et enrichir ces systèmes avec des fonctionnalités telles que la génération automatique de quiz et de contenus pédagogiques interactifs.

Nous collaborons également avec des sociétés edtech influentes sur le marché français, comme Ubicast et Wooclap. Ces partenariats exploitent les technologies et solutions innovantes de ces entreprises pour fournir des outils pédagogiques avancés. Les premiers contacts ont jeté les bases d'une collaboration fructueuse, notamment dans la collecte et l'analyse du matériel de cours, ainsi que dans l'évaluation des quiz générés par l'IA. L'API collectera différents types de matériel pédagogique (textes, documents, vidéos) et recueillera les avis des enseignants sur la pertinence des questions et réponses générées par l'IA, garantissant ainsi une amélioration continue de la qualité des quiz et des contenus présentés.

Un aspect crucial du projet est la fourniture d'une alternative souveraine à l'API Chat GPT, permettant aux enseignants et aux établissements éducatifs français d'éviter les risques liés à la récupération et à l'utilisation de leurs données par des entités extérieures. L'API offre une solution sécurisée, où les enseignants gardent le contrôle total sur leurs contenus et peuvent révoquer tout quiz ou contenu jugé peu pertinent. De plus, en automatisant la création de quiz et de contenus pédagogiques, l'API permet aux enseignants de gagner du temps, leur permettant ainsi de se concentrer sur l'enseignement et l'accompagnement des étudiants, tandis que les

étudiants bénéficient d'outils et de méthodes pédagogiques efficaces et adaptés à leurs besoins spécifiques.

Comparaison des solutions existantes

La veille concurrentielle des solutions existantes dans le cadre du projet Irekia révèle plusieurs initiatives similaires visant à intégrer des technologies avancées dans l'éducation et l'edtech, avec un accent particulier sur l'amélioration de l'interactivité et du suivi pédagogique. Parmi ces initiatives, des plateformes telles que Moodle, Ubicast, Wooclap, l'API Chat GPT et l'API Aristote illustrent comment les technologies peuvent être utilisées pour développer des outils éducatifs innovants et interactifs.

- Moodle

Moodle est une plateforme d'apprentissage open source largement adoptée dans le milieu académique. Elle propose une gamme étendue de fonctionnalités pour la création de cours en ligne, la gestion des inscriptions, le suivi des progrès des étudiants et l'évaluation des apprentissages. Grâce à sa flexibilité et à ses possibilités de personnalisation, Moodle permet l'intégration d'une variété d'outils tiers, facilitant ainsi une approche d'enseignement hybride. Les Learning Analytics intégrées à Moodle offrent la capacité de suivre et d'analyser les performances des étudiants en temps réel, permettant une intervention pédagogique rapide et ciblée.

- Ubicast

Ubicast propose une solution de capture et de diffusion de contenu vidéo dédiée à l'éducation. Elle permet l'enregistrement de cours en direct et leur diffusion en ligne avec des fonctionnalités interactives comme l'ajout de chapitres, de sous-titres et de questions intégrées. La plateforme offre également des outils d'édition vidéo simples, permettant aux enseignants de créer des contenus attrayants sans nécessiter de compétences techniques avancées. Ubicast facilite l'hybridation des enseignements en rendant les cours accessibles et interactifs pour les étudiants à distance.

- Wooclap

Wooclap est un outil interactif conçu pour dynamiser les cours en présentiel et à distance. Il permet aux enseignants de poser des questions en direct, de réaliser des sondages et des quiz, ainsi que de recueillir des feedbacks en temps réel. Les réponses des étudiants sont collectées et analysées, offrant une vue d'ensemble de leur compréhension et de leur engagement. Wooclap s'intègre facilement avec des plateformes d'apprentissage comme Moodle, enrichissant l'expérience pédagogique en rendant les cours plus interactifs et participatifs.

- API Chat GPT

L'API Chat GPT, développée par OpenAI, est une interface permettant d'intégrer des capacités de traitement du langage naturel dans diverses applications éducatives. Elle peut être utilisée pour créer des assistants virtuels, des outils de tutorat automatique et des systèmes de réponse aux questions en temps réel. En exploitant l'intelligence artificielle, l'API Chat GPT fournit des explications et des réponses adaptées aux besoins spécifiques des étudiants, améliorant ainsi leur expérience d'apprentissage et leur suivi pédagogique.

- API Aristote

L'API Aristote est une solution développée pour structurer et standardiser les données éducatives, facilitant leur intégration et leur exploitation dans divers contextes pédagogiques. Elle permet la création d'objets pédagogiques interactifs et personnalisés, enrichis de données provenant de multiples sources. L'API Aristote encourage la collaboration et le partage de ressources éducatives tout en assurant la souveraineté et la sécurité des données. En combinant ces fonctionnalités avec des outils d'analyse avancés, l'API Aristote contribue à optimiser les stratégies pédagogiques et à améliorer les résultats éducatifs.

Le tableau suivant présente les caractéristiques des solutions étudiées par rapport aux objectifs attendus du projet Connect Unita :

Caractéristiques	Moodle	Ubicast	Wooclap	API Chat GPT	API Aristote
Intégration de données	Oui	Oui	Oui	Oui	Oui
Génération automatique de quizz	Non	Oui	Oui	Oui	Oui
Collecte de supports pédagogiques	Oui	Oui	Non	Oui	Oui
Avis des enseignants sur les quizz	Non	Non	Non	Non	Oui
Souveraineté des données	Oui	Oui	Oui	Non	Oui
Sécurité des données	Oui	Oui	Oui	Non	Oui
Gérer le type de questions	Non	Non	Non	Non	Non
Créer ses propres questions	Non	Non	Non	Non	Non
Retravailler les questions de l'IA	Non	Non	Non	Non	Non
Accepte les liens Vidéos	Oui	Non	Non	Non	Non

Figure 2: Tableau travaux solutions existantes .

Solution Proposé

Vue d'ensemble de la solution

Pour le projet Connect Unita, nous proposons une solution technologique sous la forme d'une application web destinée à faciliter l'hybridation des cours pour les enseignants. Pour ce faire, nous utilisons l'outil H5P, qui permet d'intégrer des questions interactives dans les vidéos pédagogiques. Notre plateforme propose plusieurs services pour automatiser cette intégration, simplifiant ainsi le processus de manière efficace et intuitive.

Pour accéder au site, vous devez être connecté au réseau eduroam de l'Université via le wifi ou en utilisant OpenVPN Connect pour une connexion à distance. Voici l'URL pour y accéder : <http://10.3.154.163:8081/>

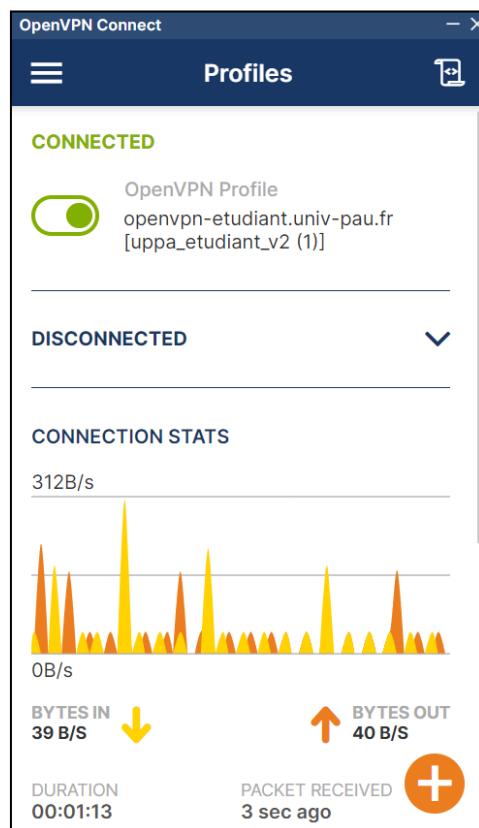


Figure 3: OpenVPN Connect.

Navigation sur le site

Lorsque vous arrivez sur le site, vous pouvez facilement naviguer entre différents services grâce au menu situé en haut de la page. La page "Home" vous accueille avec une vue d'ensemble du site. Si vous avez besoin d'aide pour comprendre le fonctionnement des services, la page "Tutoriel" est à votre disposition avec des

guides détaillés. Pour toute question supplémentaire, la section FAQ fournit des réponses aux interrogations courantes. Et si vous avez besoin d'une assistance plus personnalisée, vous pouvez contacter directement l'équipe de développement via la section "Contact".

H5P PROJECT

HOME SIMPLE CHOICE QUESTION MULTIPLE CHOICE QUESTION TUTORIEL QUESTIONGENERATOR CONTACT

Figure 4: Navigation

Fonctionnalités

Les fonctionnalités doivent répondre à des besoins précis. C'est pourquoi nous avons découpé notre application en plusieurs fonctionnalités distinctes. Nous allons d'abord examiner et expliquer les différentes fonctionnalités du côté utilisateur, afin d'avoir une bonne compréhension de la demande et des informations fournies à l'application. Cela nous permettra de réaliser efficacement les tâches demandées.

Vidéo interactives avec des questions à choix simple.

SIMPLE CHOICE QUESTION

Video Name:

ex: VideoCours

?

Data File: [Download Base Data File](#)

[Choisir un fichier] Aucun fichier choisi

?

Video File:

[Choisir un fichier] Aucun fichier choisi

?

Video URL:

<https://example.com/video.mp4>

?

Strict:

?

Languages: ?

French

English

Portuguese

Spanish

Italian

German

Ukrainian

Romanian

Download

Figure 5: Simple Choice Question.

La première sur notre liste est la création de vidéo interactive avec des questions à choix simple. Cette fonctionnalité nécessite plusieurs entrées spécifiques pour fonctionner correctement. Voici les détails des entrées requises :

- **Nom de la vidéo** : Il est essentiel de fournir un nom pour la vidéo que vous souhaitez utiliser. Le nom sera le nom du fichier en .h5p qui est l' output final de cette fonctionnalité.
- **Fichier Excel avec les questions** : Vous devez télécharger un fichier excel contenant les questions associées à la vidéo. Ce fichier doit respecter une structure particulière pour que les questions soient correctement intégrées. Cette structure se compose comme suite :

question	reponseA	reponseB	reponseC	reponseD	reponseE	duration	renvoi
Quel age as-tu?	18	21	20			15	
Quel est ton prenom?	Alexandre	Thomas	Pierre	Paul	Charles		
tu habites dans quelle ville	Anglet	Bayonne	Bordeaux	Agen		36	

Figure 6: Fichier Excel Simple.

Dans la première colonne, nous avons les différentes questions que l'utilisateur souhaite intégrer dans sa vidéo interactive. Sur chaque ligne, se trouvent toutes les réponses possibles pour chaque question.

Une fois les réponses listées, nous avons une colonne intitulée "duration" qui indique le timecode auquel la question doit apparaître dans la vidéo. Enfin, la colonne "renvoi" représente un autre timecode correspondant à l'endroit de la vidéo où la réponse correcte est expliquée. Si ce timecode est renseigné par le professeur, l'étudiant sera redirigé vers ce moment précis en cas de réponse incorrecte, afin d'obtenir l'explication.

ATTENTION : Celui-ci change est la structure pour la fonctionnalité à choix simple. La structure devra changer selon la fonctionnalité ou nous l'utiliserons.

- **Vidéo** : Vous pouvez soit télécharger une vidéo dans différents formats, soit fournir un lien vers une vidéo hébergée sur une autre plateforme. Nous acceptons différents types de video : .mp4, .avi, .mov, .wmv, .flv, .mkv, .mpeg, .m4v, .webm, .3gp. Voici un exemple .mp4:



Figure 7: Video input.

- **Mode vidéo strict** : Le mode vidéo strict est une option représentée par une case à cocher, permettant de déterminer si la vidéo doit être en mode strict ou non. Le mode strict impose des contraintes spécifiques sur la manière dont les questions sont présentées et répondues. Nous allons entrer plus en détail sur ce mode.

Les principales caractéristiques du mode strict sont les restrictions appliquées à la vidéo. L'une des restrictions majeures est l'incapacité de naviguer librement au sein de la vidéo.



Figure 8: Navigation Mode Strict.

En mode strict, l'étudiant est obligé de valider toutes les réponses de la ressource. Cela signifie que lorsqu'il se trompe sur l'une des questions, la vidéo revient à la question précédente ou au début de la vidéo pour la première question. Cette contrainte garantit que l'étudiant comprend et retient les informations avant de progresser dans la vidéo.

- **Langues pour les sous-titres** : Vous avez la possibilité de choisir parmi différentes langues pour la création des sous-titres. Cela permet de rendre la vidéo accessible à un public plus large. L'alliance unie a apporté des barrières de langue, c'est pourquoi il est primordial de toujours avoir une option de traduction afin de se rendre accessible à tous les membres de l'alliance et plus encore. Cette option se présente sous la forme d'une liste de langues que l'application prend en compte et que l'on peut cocher.

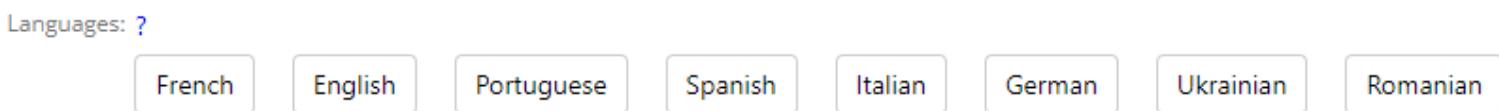


Figure 9: Langues.

- **Ajout manuel des sous-titres** : Il est également possible d'entrer les sous-titres manuellement, offrant une flexibilité supplémentaire pour la personnalisation des contenus. À chaque langue sélectionnée dans la liste des sous-titres, une case apparaît où l'utilisateur a la possibilité de rentrer les sous-titres par lui-même. Cela fait gagner énormément de temps, car nous n'avons pas à utiliser de modèle de traduction. C'est pourquoi, malgré la possibilité de traduire automatiquement, nous avons gardé cette option.

Languages: ?



French Subtitle:

?

Figure 10: Sous titre Manuel.

Il faut donc ajouter le texte de la vidéo traduit dans la langue sélectionnée dans cette zone ci-dessus.

Vidéo interactives avec des questions à choix multiple.

MULTIPLE CHOICE QUESTION

Video Name:

ex: VideoCours

Data File: [Download Base Data File](#)

Aucun fichier choisi

Video File:

Aucun fichier choisi

Video URL:

<https://example.com/video.mp4>

Strict:

Languages:

Figure 11: Multiple Choice Question.

Cette fonctionnalité est essentiellement la même que celle à choix simple, mais plusieurs réponses peuvent être justes pour une question. Elle dispose donc essentiellement des mêmes inputs que la précédente : **nom, fichier Excel, vidéo, mode strict, langues**. À une différence près :

- **Le fichier excel:** Celui- ci change sa structure, afin de bien gérer les différentes réponses qui peuvent être multiples. Voici donc le nouveaux fichier excel ou csv que l'utilisateur doit rentrer dans l'application.

question	reponseA	result	reponseB	res	reponseC	res	reposeD	res	reposeE	res	duratio	renvoi
What is a function?	An element of math that describes a specific relation between two sets of numbers	v	A machine that takes an input and produces an output according to a rule	f	A type of equation that can be solved using algebra	v	A concept used in calculus	f	A fundamental idea in math that represents a relationship	v	60	
What is the example function?	$f(x) = 2x + 3$	v	$f(x) = x^2 + 1$	f	$f(x) = 5x - 2$	f	$f(x) = x/2$	f	$f(x) = 10x + 5$	f	30	
How do you evaluate f(4)?	By replacing x with 4 in the function $f(x) = 2x + 3$	v	By solving a equation for x equal to 4	v	By using algebraic manipulations	v	By guessing and checking	f	By substituting x with 4 in the given function	v	60	
What is the importance of practicing functions?	It helps you become an expert in math	v	It allows you to understand complex concepts	f	It enables you to solve problems efficiently	f	It makes learning math more enjoyable	f	It helps you develop problem-solving skills	v	30	

Figure 12: Fichier Excel Multiple.

Dans la première colonne, nous avons les différentes questions que l'utilisateur souhaite intégrer dans sa vidéo interactive. Sur chaque ligne, se trouvent toutes les réponses possibles pour chaque question. Dans la colonne adjacente à chaque réponse, il est indiqué si elle est correcte ou incorrecte, représenté respectivement par "v" (vrai) et "f" (faux).

Une fois les réponses listées, nous avons une colonne intitulée "duration" qui indique le time code auquel la question doit apparaître dans la vidéo. Enfin, la colonne "renvoi" représente un autre timecode correspondant à l'endroit de la vidéo où la réponse correcte est expliquée. Si ce timecode est renseigné par le professeur, l'étudiant sera redirigé vers ce moment précis en cas de réponse incorrecte, afin d'obtenir l'explication.

Génération de questions via la vidéo

QUESTION GENERATOR

Video Name:

Create your CSV file with questions.

test

Video File:

Choisir un fichier video.mp4

Upload

Figure 13: Question Generator.

Le service "QuestionGenerator" génère automatiquement un fichier .csv contenant des questions et des réponses basées sur la vidéo que vous téléchargez. Ce fichier peut ensuite être utilisé dans les services "SimpleChoiceQuestion" ou "MultipleChoiceQuestion" pour créer automatiquement un fichier .h5p. Voici les détails des entrées requises :

- **Le nom** : Tout d'abord, nous devons entrer le nom du fichier CSV qui sera généré.
- **La vidéo** : Nous devons également pouvoir sélectionner la vidéo sur laquelle nous souhaitons générer les questions. Cette vidéo sera analysée par une IA qui produira des questions liées à son contenu. Les questions seront ensuite

exportées sous format CSV, adapté à l'intégration dans la fonctionnalité de choix simple.

Voici donc un exemple de output de la fonctionnalité :

question	reponseA	reponseB	reponseC	reponseD	reponseE	duration	renvoi
Quelle est la valeur de $f(4) = 2x + 3$ pour $x = 4$?		6	10	5	8	12	100
Quel est le résultat de la fonction fegale $2x + 1$ si $x = 2$?		5	3	4	9	11	90
Laquelle des fonctions suivantes décrit une relation spécifique entre deux ensemble de nombres?	$F(x) = 2x + 1$.	$F(x) = 2x - 1$.	$F(x) = x + 1$.	$F(x) = 3x - 2$.	$F(x) = 4x + 3$.		80
Quel est le rôle de la fonction dans une relation spécifique entre deux ensemble de nombres?	Une machine qui transforme un nombre en un autre selon une règle précise.	Un élément fondamental en mathématiques pour décrire les relations entre les nombres.	Une méthode pour résoudre les équations simples.	Un outil pour calculer les valeurs des fonctions.			50
Quelle est la caractéristique de la fonction qui la rend utile?	Elle permet de transformer un nombre en un autre selon une règle précise.	mais avec la pratique	elle devient simple.	Elle est utilisée pour résoudre les équations simples.	Elle est un élément fondamental en mathématiques pour décrire les relations entre les nombres.		100

Figure 14: Output Question Generator.

TUTORIAL

Choose a service:

[Simple Choice Question](#)

[Multiple Choice Question](#)

[Question Generator CSV](#)

[.h5P Upload Moodle](#)

Steps to use the application:

Generate Questions/Answers Using AI

Step 1: Go to the [Question Generator CSV](#) page

Step 2: Enter a name for your CSV file

Step 3: Upload your video

Step 4: Click the "Download" button to generate questions and answers in the .csv file with AI

FAQ

How to contact support?

You can contact us via the contact page or send an email to alexandre.paboeuf@etud.univ-pau.fr

What is Strict mode?

Strict mode is designed so that students must watch the entire video and answer all questions. Therefore, navigating through the video is not allowed, and if a mistake is made on a question, the student is automatically taken back to the previous question.

What is the 'duration' column for?

The 'duration' column is important because it allows you to place your questions at the desired points in the video. For example: If my video is 60 seconds long and I have 2 questions to ask. I want my first question to be in the middle of my video, so I write '30' in the duration column for the first question. Then I want my second question to be at the very end of my video, so I leave the column EMPTY, and the question will automatically be at the end of the video.

© 2024 Your Application. All rights reserved.

Tutoriel

Figure 15: Tutorial.

Afin que tout type d'utilisateur puisse utiliser notre solution, nous avons ajouté une page "Tutoriel" qui permet à n'importe qui de comprendre le fonctionnement des différentes fonctionnalités que nous proposons. Celle-ci se compose d'une simple explication avec les différentes étapes et d'une FAQ qui permet de prévoir quelques questions récurrentes.

Comment créer une Vidéo

Si vous ne disposez pas de votre propre vidéo, prenez rendez-vous pour utiliser le studio RapidMooc de l'UPPA. Installez-vous devant un fond vert : le studio vous filme pendant que vous partagez vos connaissances à travers une vidéo. Il est équipé d'un micro-cravate pour une meilleure qualité audio et vous permet d'utiliser un PowerPoint ainsi qu'un prompteur, tout en contrôlant le défilement de vos diapositives à distance avec une télécommande.



Figure 16: RapidMooc.

Intégration dans Moodle

Maintenant que vous disposez de tous les éléments nécessaires à l'utilisation de notre solution, utilisez notre service et obtenez un fichier .h5p. Vous pouvez l'intégrer facilement dans votre cours Moodle. Cette intégration se fera comme suit:

- Vous rendre à l'endroit de votre cours où vous souhaitez ajouter la vidéo.

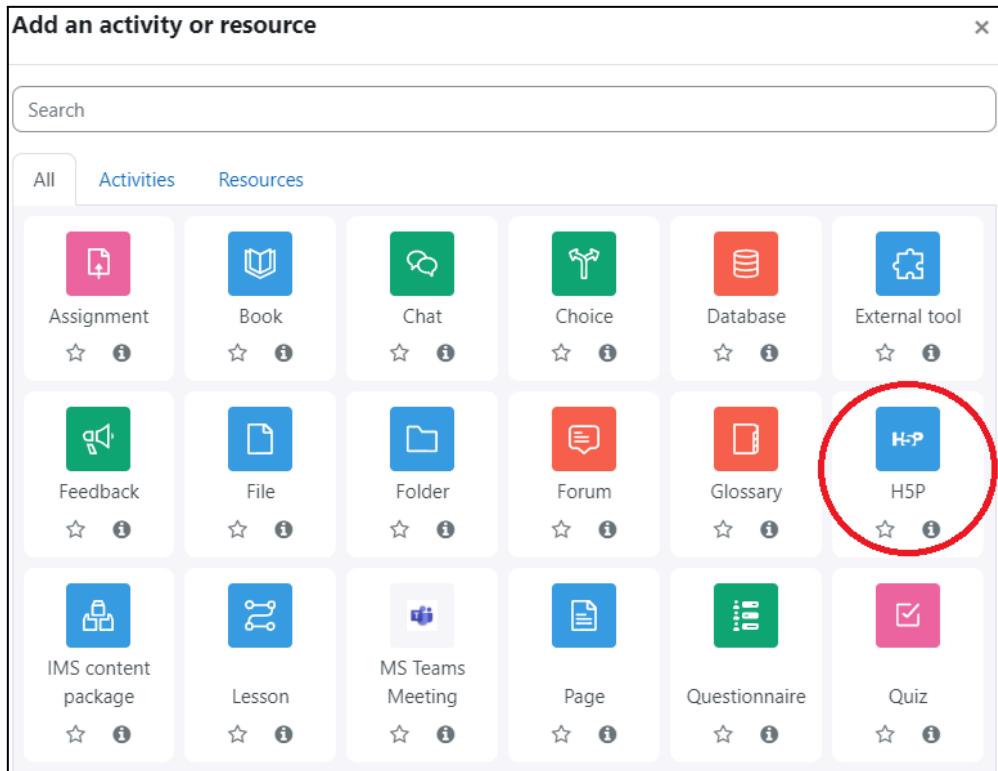


Figure 17: Moodle H5P.

- Donner un titre à la vidéo, une description, tétudiantrser votre fichier .h5p puis sauvegarder le contenu.

Adding a new H5P to Topic 1

General

Name:

Description:

Dans ce cours, nous allons explorer :

- La définition d'une fonction et son importance en mathématiques
- Les différents types de fonctions : linéaires, quadratiques, exponentielles, etc.
- Comment représenter les fonctions graphiquement et interpréter ces représentations
- Des exemples pratiques et des exercices pour renforcer votre compréhension

Display description on course page

Package file:

Accepted file types:
Archive (H5P) .h5p

Figure 18: Moodle H5P.

- Vos questions seront ainsi intégrées à votre vidéo, offrant une expérience d'apprentissage interactive et enrichie.

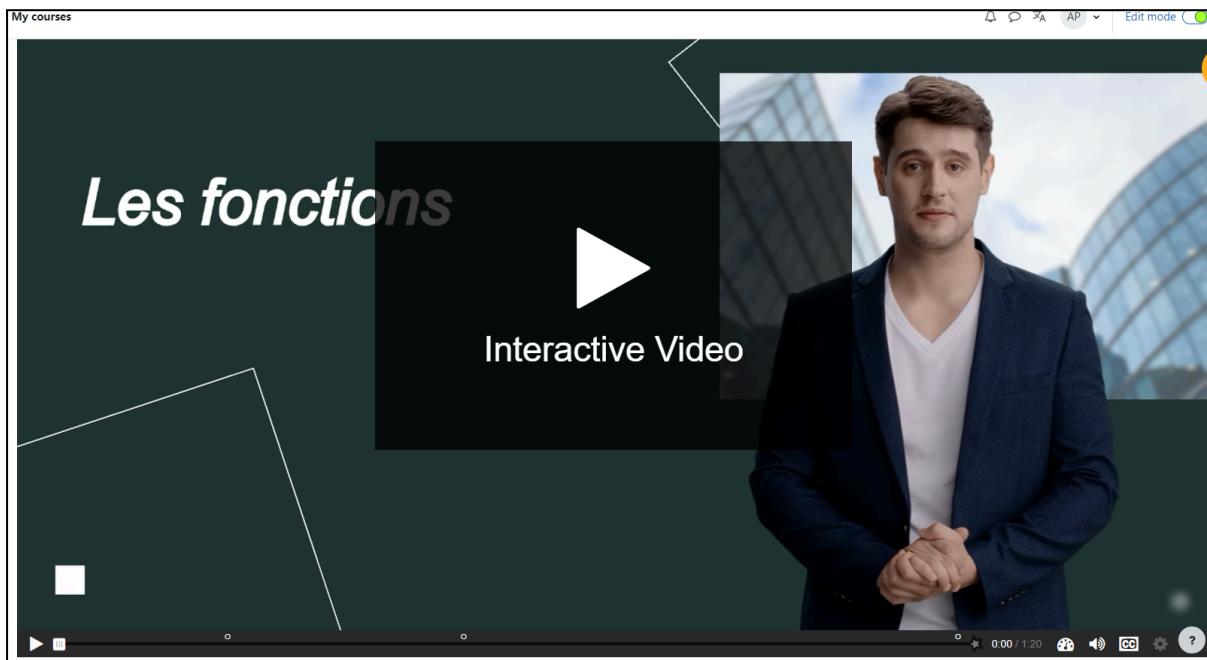


Figure 19: Vidéo Exemple.

Notre application web simplifie l'intégration de questions interactives dans les vidéos pédagogiques grâce à ses services automatisés. En utilisant des fichiers Excel ou CSV ainsi que des API Python, nous proposons aux enseignants une solution pratique pour enrichir leurs cours et impliquer leurs étudiants de manière interactive.

Architecture du Projet

Conception des différents services

Diagramme de cas d'utilisation

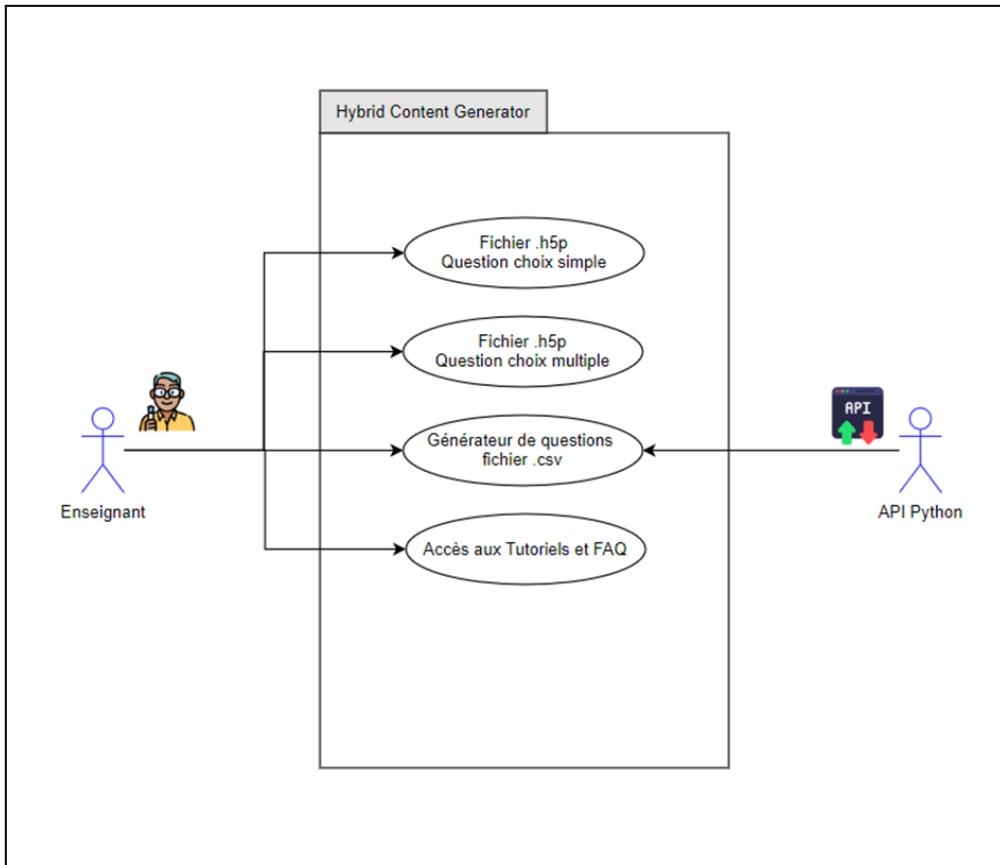


Figure 20: Diagramme Cas d'utilisation.

Fonctionnement Général

- Le système "Hybrid Content Generator" centralise les fonctionnalités de création et de gestion de contenus éducatifs. Il permet à l'enseignant de créer des vidéos interactives avec des questions sous différents formats : .h5p pour des questions à choix simple et multiple, et .csv pour une liste de questions générées par l'IA.
- L'accès aux tutoriels et à la FAQ offre un soutien continu à l'enseignant pour une utilisation efficace des outils disponibles.
- L'intégration avec l'API Python garantit l'automatisation et l'optimisation des processus de génération de contenu.

Ce diagramme illustre les principales fonctionnalités proposées par le "Hybrid Content Generator" et montre comment un enseignant peut interagir avec ces outils pour créer des ressources éducatives de manière efficace et simplifiée.

Diagramme de séquence Questions Choix Simple

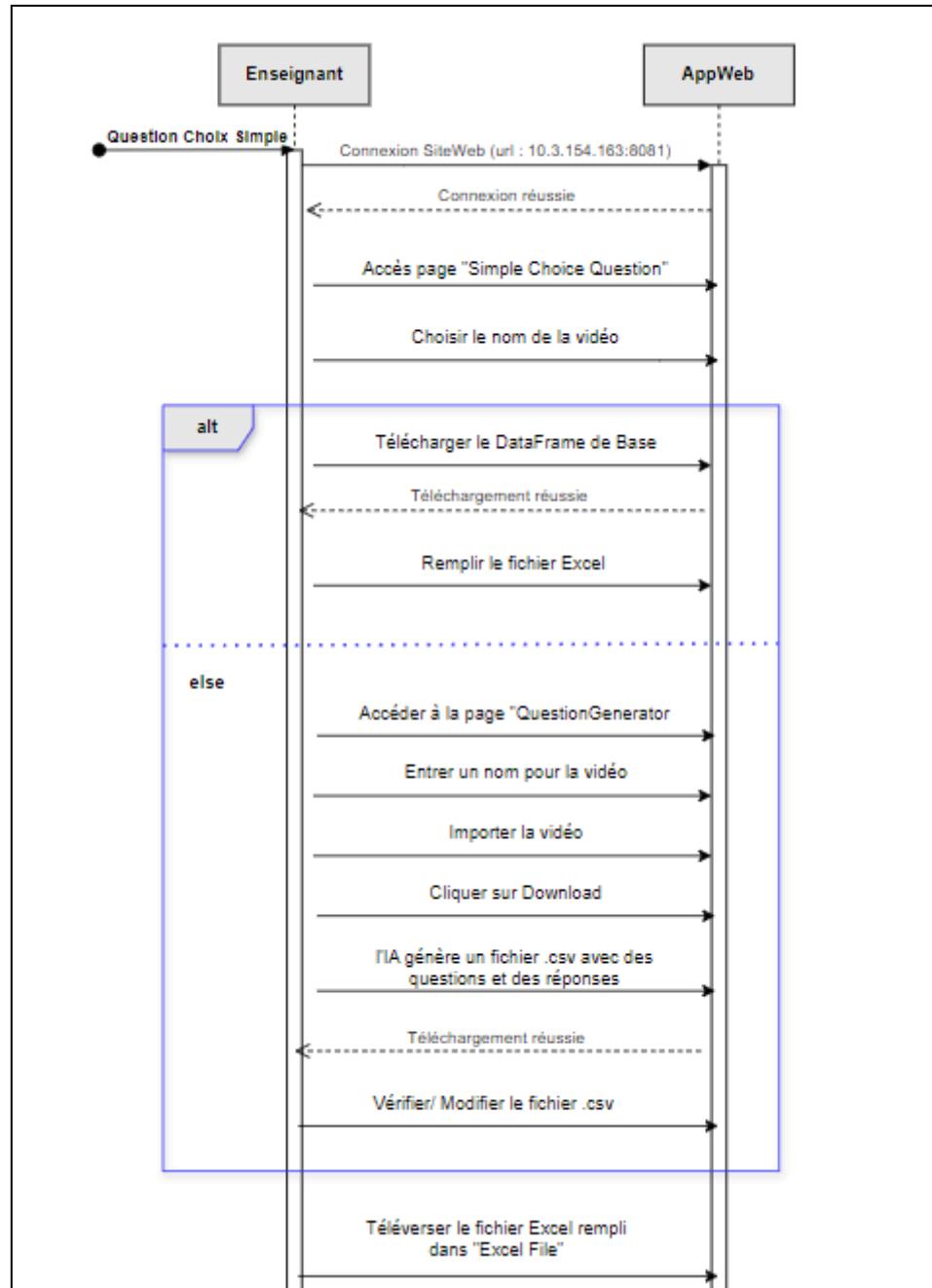


Figure 21: Diagramme Séquence Question Choix Simple P1.

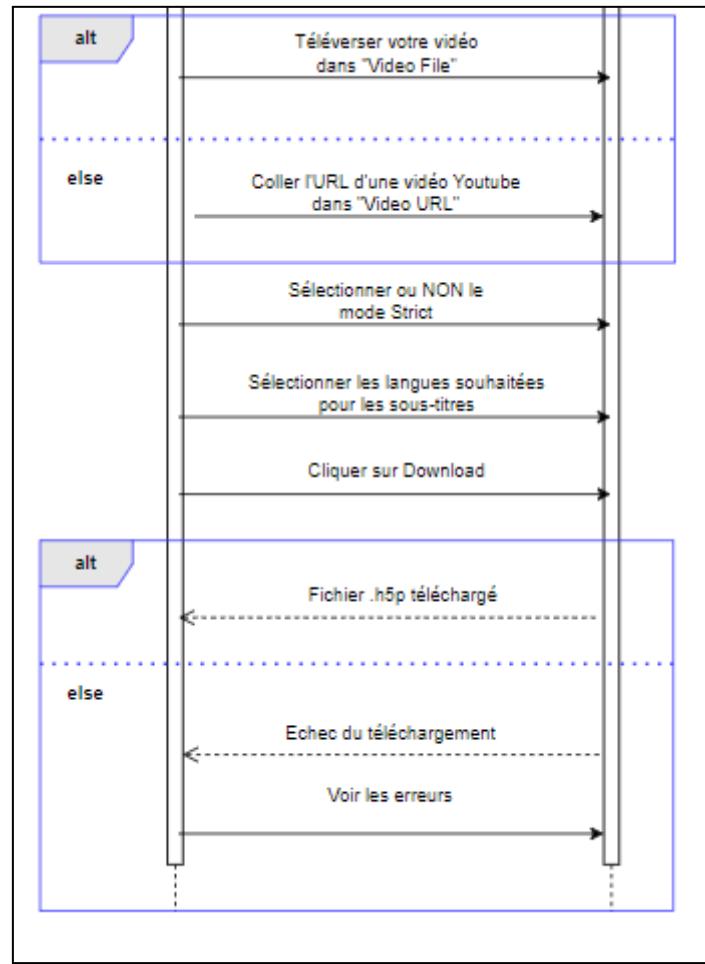


Figure 22: Diagramme Séquence Question Choix Simple P2.

Diagramme de séquence QuestionGenerator

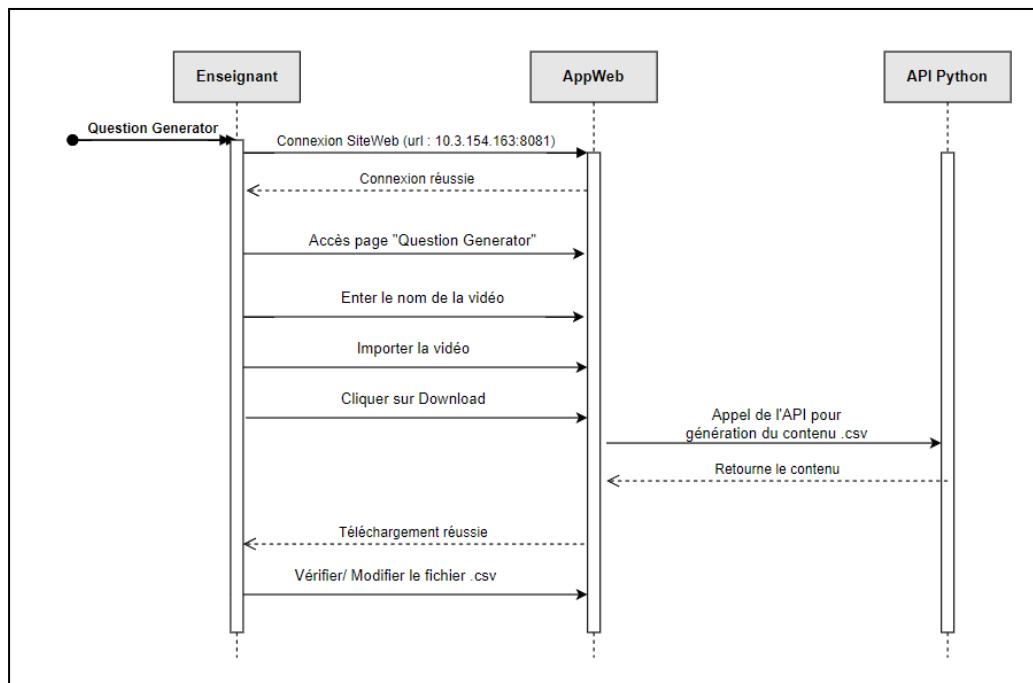


Figure 23: Diagramme Séquence Question Generator.

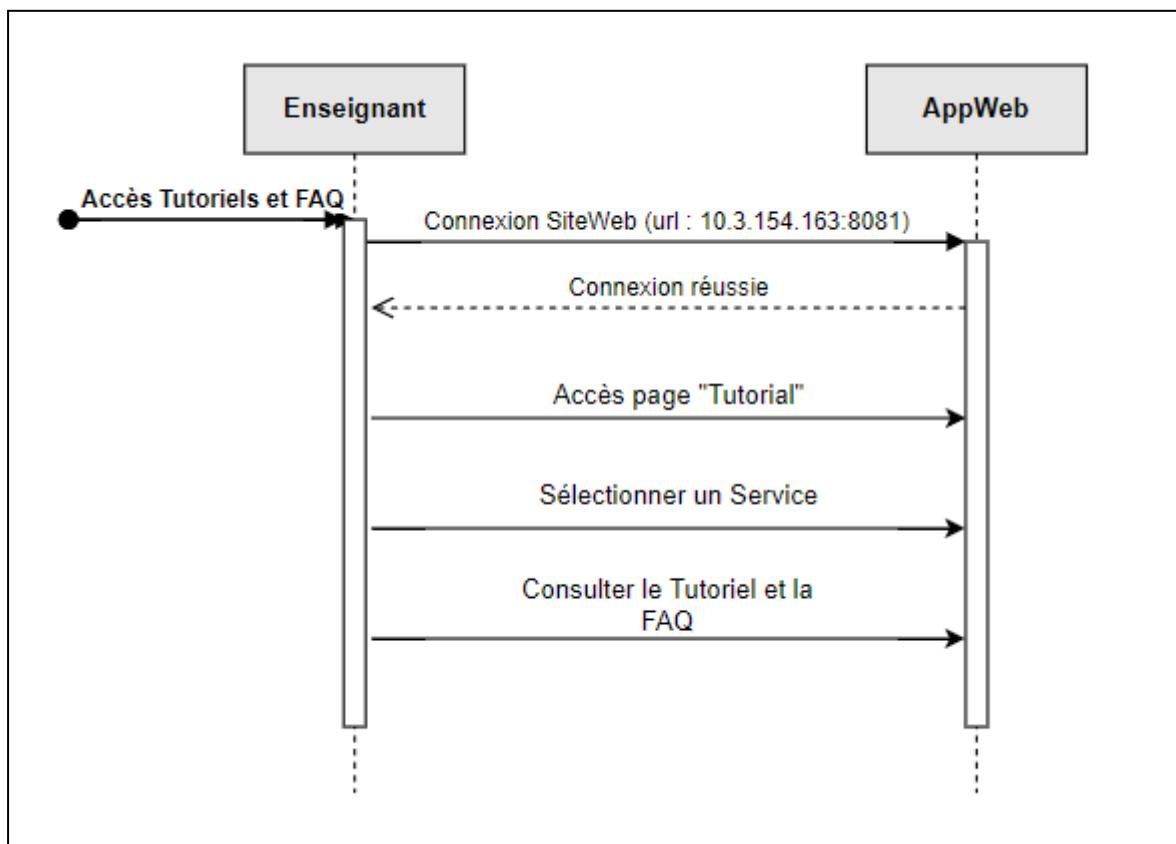
Diagramme de séquence Accès Tuto & FAQ*Figure 24: Diagramme Séquence Tutoriel & FAQ.*

Schéma de l'architecture

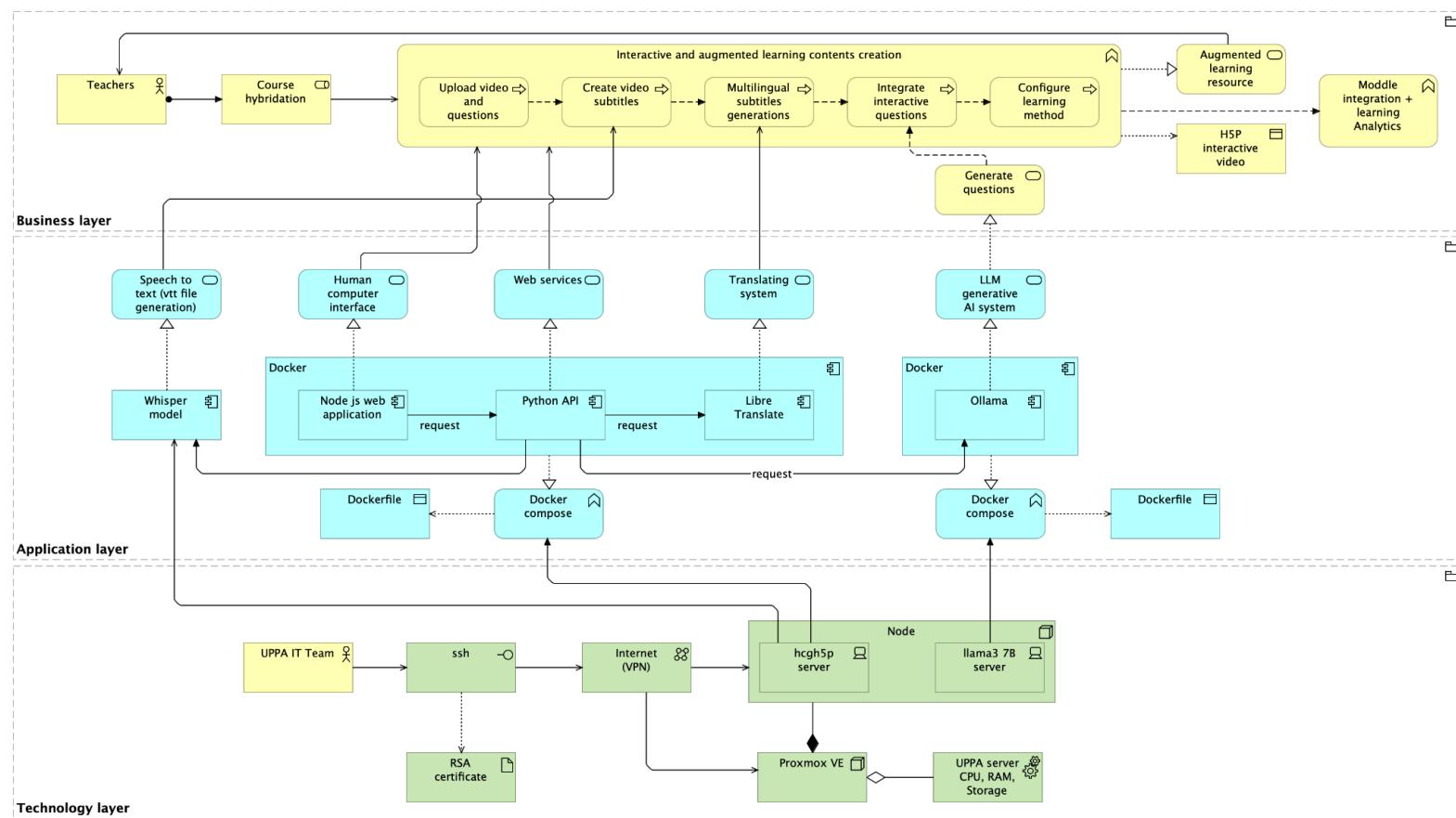


Figure 25 : Architecture globale.

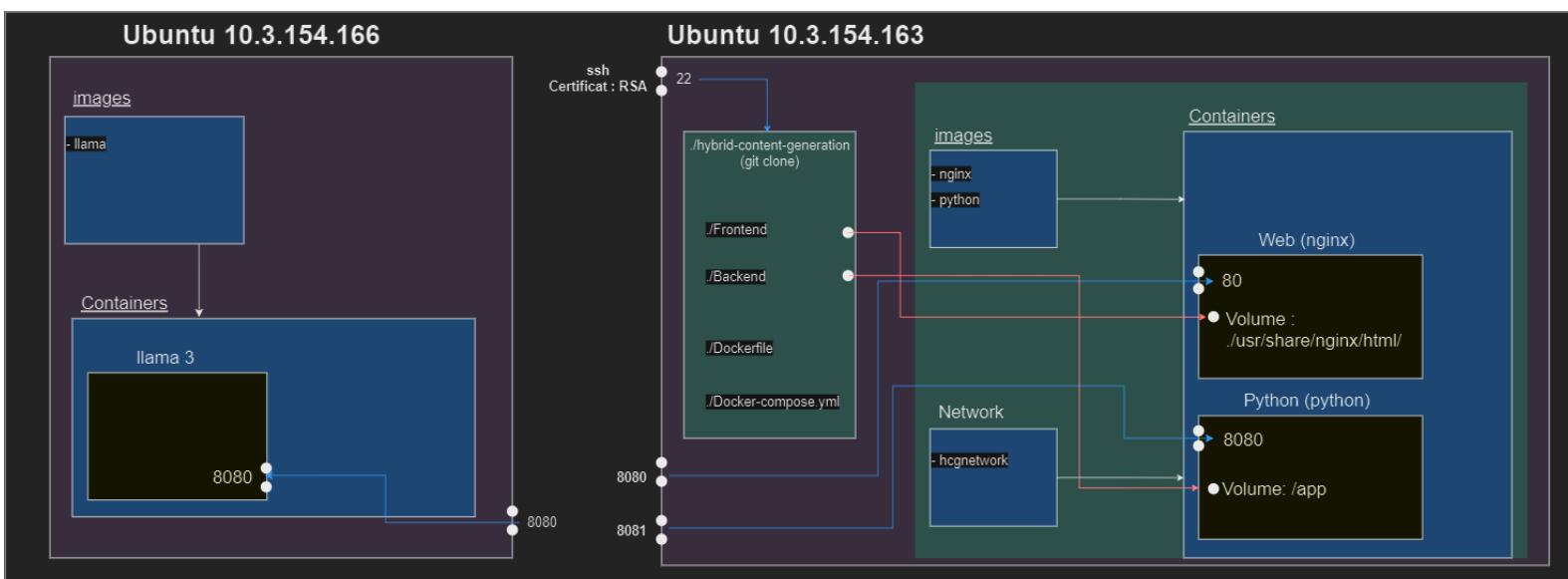


Figure 26 : Schémas Architecture.

Ces schémas fournis illustrent l'architecture d'un projet qui utilise des conteneurs Docker déployés sur deux machines Ubuntu distinctes. Voici une explication détaillée de chaque composant et de leur interaction :

Machine 1: Ubuntu 10.3.154.166

1. Images
 - llama : Cette image Docker est disponible sur cette machine.
2. Containers
 - llama 3 : Un conteneur créé à partir de l'image "llama". Ce conteneur est configuré pour exposer le port 8080, permettant l'accès aux services internes de ce conteneur via ce port.
3. Réseau
 - Le conteneur "llama 3" expose le port 8080 qui est ensuite redirigé pour une communication externe.

Machine 2: Ubuntu 10.3.154.163

1. Réseau
 - hognetwork : Un réseau Docker personnalisé pour la communication entre les conteneurs sur cette machine.
2. Images
 - nginx : Une image Docker pour le serveur web Nginx.
 - python : Une image Docker pour exécuter des applications Python.
3. Containers
 - Web (nginx) : Un conteneur basé sur l'image "nginx". Il expose le port 80 pour les requêtes HTTP, et utilise un volume monté à /usr/share/nginx/html/ pour servir le contenu statique.
 - Python (python) : Un conteneur basé sur l'image "python". Il expose le port 8080 pour accéder à l'application Python, et utilise un volume monté à /app pour les données de l'application.
4. Dépôt Git
 - ./hybrid-content-generation (git clone) : Ce dossier contient le code source cloné depuis un dépôt Git. Il inclut trois sous-dossiers principaux :
 - ./Frontend/ : Probablement contient le code pour l'interface utilisateur.
 - ./Backend/ : Probablement contient le code pour le backend de l'application.
 - ./Dockerfile et ./Docker-compose.yml : Fichiers de configuration pour Docker permettant de construire et orchestrer les conteneurs nécessaires pour le projet.

Interaction entre les Machines

- SSH Connection : Une connexion SSH est configurée entre les deux machines via un certificat RSA, permettant des opérations sécurisées comme le déploiement, la gestion des conteneurs, ou la synchronisation des services.

Points de Communication

- Ports exposés et mappés :

- Le port 8080 de "llama 3" sur Ubuntu 10.3.154.166 est accessible.
- Les ports 80 (Nginx) et 8080 (Python) sur Ubuntu 10.3.154.163 sont également exposés et mappés pour permettre l'accès aux services web et aux applications Python respectivement.

Architecture VScode

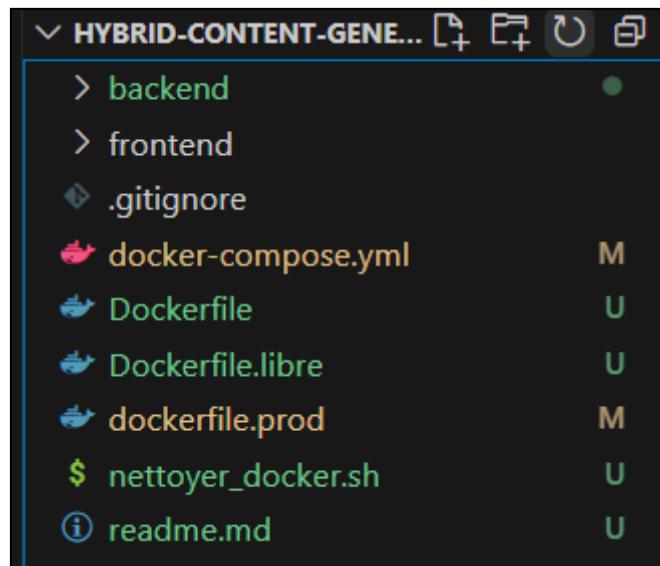


Figure 27: Architecture VSCode.



Figure 28: Architecture VSCode Backend.

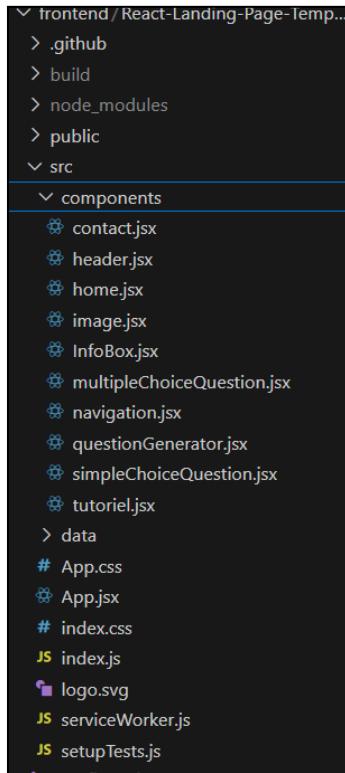


Figure 29: Architecture VSCode Frontend.

Revu de code

API python fichier de l'api, app.py :

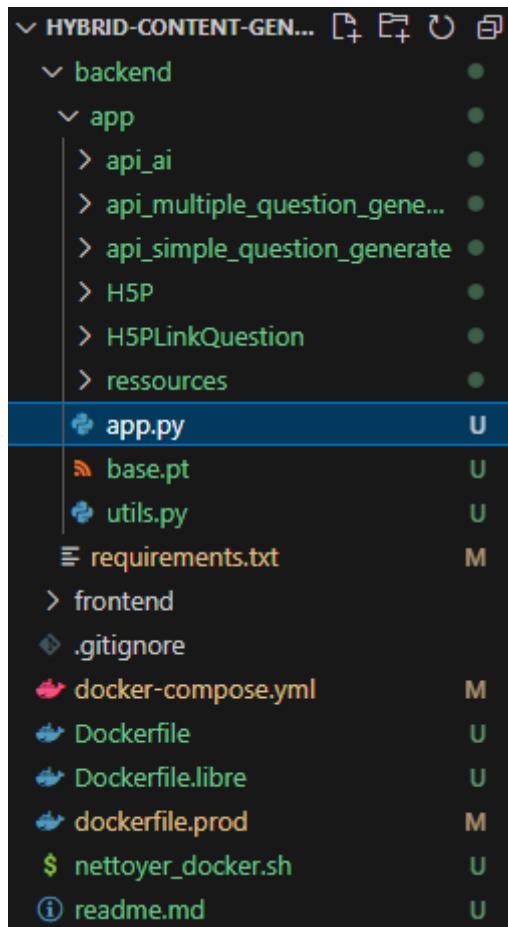


Figure 30: Fichier app.py.

Imports et Initialisation :

```
from flask import Flask, request, send_file, jsonify
from flask_cors import CORS
import json
import os
from flasgger import Swagger

from api_simple_question_generate.h5pLink_generator import create_h5p_interactions_link
from api_simple_question_generate.h5p_generator import create_h5p_interactions
from api_multiple_question_generate.h5p_generator import create_h5p_interactions_multiple
from api_multiple_question_generate.h5plink_generator import create_h5p_interactions_multiple_link
from api_ai.generate_request import process_video_to_csv
from utils import load_data_file
```

Figure 31: Import app.py.

- Le code commence par importer les modules nécessaires (`Flask`, `request`, `send_file`, `jsonify`, etc.).
- Il utilise également `Flask-CORS` pour gérer les CORS (Cross-Origin Resource Sharing) et `flasgger` pour la documentation Swagger.

Route principale (/) :

```
@app.route('/')
def index():
    """
    Route principale de l'API.
    ---
    responses:
        200:
            description: Message de bienvenue
            content:
                application/json:
                    schema:
                        type: object
                        properties:
                            message:
                                type: string
    """
    return jsonify({'message': 'Welcome to my Flask API!'})
```

Figure 32:Route (/).

- Cette route renvoie un simple message JSON indiquant le démarrage de l'API.
- Utilisation de `jsonify` pour retourner une réponse JSON.

Route /interactive_video_multiple_choice_question :

```
@app.route('/interactive_video_multiple_choice_question', methods=['POST'])
def interactive_video_multiple_choice_question():
    ...
```

Figure 33: Route (/interactive_video_multiple_choice_question).

- Cette route permet de générer un fichier H5P contenant des questions à choix multiples basées sur une vidéo et des données fournies.
- Paramètres attendus via formData : videoName, dataFile, videoUrl, videoFile, STRICT, languages, subtitles.
- La fonction vérifie d'abord la présence du fichier de données (dataFile) et l'enregistrer localement.

```
try:
    video_name = request.form.get('videoName')
    data_file = request.files.get('dataFile')
    video_url = request.form.get('videoUrl', '')
    video_file = request.files.get('videoFile')
    strict = request.form.get('STRICT', 'false')
    languages = request.form.get('languages', '[]')
    subtitles = request.form.get('subtitles', '[]')

    strict = strict.lower() == 'true'
    languages = json.loads(languages)
    subtitles = json.loads(subtitles)

    if data_file:
        data_file_ext = os.path.splitext(data_file.filename)[1]
        data_file_path = '/tmp/data_file' + data_file_ext
        data_file.save(data_file_path)
        data = load_data_file(data_file_path)
    else:
        return jsonify({'error': 'No data file provided.'}), 400
```

Figure 34: Route (/interactive_video_multiple_choice_question), code1.

- En fonction de la présence de videoUrl ou videoFile, elle appelle des fonctions (create_h5p_interactions_multiple_link ou create_h5p_interactions_multiple) pour générer le fichier H5P.
- Elle retourne le fichier généré en tant que réponse avec send_file.

```
if video_url:
    zip_file_path = create_h5p_interactions_multiple_link(data_file_path, video_url, video_name + '.h5p', strict=strict, languages=languages)
else:
    if not video_file:
        return jsonify({'error': 'No video file provided.'}), 400

    video_file_path = '/tmp/video.mp4'
    video_file.save(video_file_path)
    zip_file_path = create_h5p_interactions_multiple(data_file_path, video_file_path, video_name + '.h5p', strict=strict, languages=languages, subtitles=subtitles)

return send_file(zip_file_path, as_attachment=True, mimetype='application/zip', download_name=video_name + '.h5p')

except Exception as e:
    return jsonify({'error': str(e)}), 500
```

Figure 35: Route (/interactive_video_multiple_choice_question), code2.

Route /interactive_video_simple_choice_question :

```
@app.route('/interactive_video_simple_choice_question', methods=['POST'])
def interactive_video_simple_choice_question():
```

Figure 36: Route (/interactive_video_simple_choice_question).

- Similaire à la précédente, mais génère des questions à choix simple plutôt que multiples.
- Paramètres et traitement similaires à /interactive_video_multiple_choice_question.

```
if video_url:
    zip_file_path = create_h5p_interactions_link(data_file_path, video_url, video_name + '.h5p', strict=strict, languages=languages)
else:
    if not video_file:
        return jsonify({'error': 'No video file provided.'}), 400

video_file_path = '/tmp/video.mp4'
video_file.save(video_file_path)
zip_file_path = create_h5p_interactions(data_file_path, video_file_path, video_name + '.h5p', strict=strict, languages=languages, subtitles=subtitles)
```

Figure 37: Route (/interactive_video_simple_choice_question), code.

Route /upload_video :

```
@app.route('/upload_video', methods=['POST'])
def upload_video():
```

Figure 38: Route (/upload_video).

- Cette route permet de télécharger une vidéo, la traite pour générer un fichier CSV.
- Paramètres attendus via `formData:videoFile, videoName`.
- La vidéo est enregistrée localement, puis une fonction (`process_video_to_csv`) est appelée pour la convertir en CSV.

```

try:
    csv_dir = "./ressources/data/"
    for f in os.listdir(csv_dir):
        if f.endswith(".csv"):
            os.remove(os.path.join(csv_dir, f))

    if 'videoFile' not in request.files:
        return jsonify({'error': 'No file part'}), 400

    video_file = request.files['videoFile']
    if video_file.filename == '':
        return jsonify({'error': 'No selected file'}), 400

    video_name = request.form.get('videoName', os.path.splitext(video_file.filename)[0])

    video_file_path = os.path.join('/tmp', video_name + '.mp4')
    video_file.save(video_file_path)

    process_video_to_csv(video_file_path, video_name)

```

Figure 39: Route (/upload_video), code1.

- Le fichier CSV généré est retourné en réponse avec `send_file`.

```

        return send_file(
            csv_file_path,
            as_attachment=True,
            download_name=video_name + ".csv",
            mimetype="text/csv"
        )

    except Exception as e:
        return jsonify({'error': str(e)}), 500

```

Figure 40: Route (/upload_video), code2.

Gestion des erreurs :

- Chaque route inclut une gestion des erreurs avec des codes HTTP appropriés (`400` pour les erreurs client, `500` pour les erreurs serveur).
- Les erreurs sont renvoyées sous forme de réponse JSON avec des détails sur l'erreur.

Exécution de l'application :

- Le code vérifie si le script est exécuté directement (`__name__ == '__main__'`) et démarre l'application Flask sur le port `8080`.

```

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080)

```

Figure 41: Route (/upload_video), code3.

API python fichier h5p_generator.py :

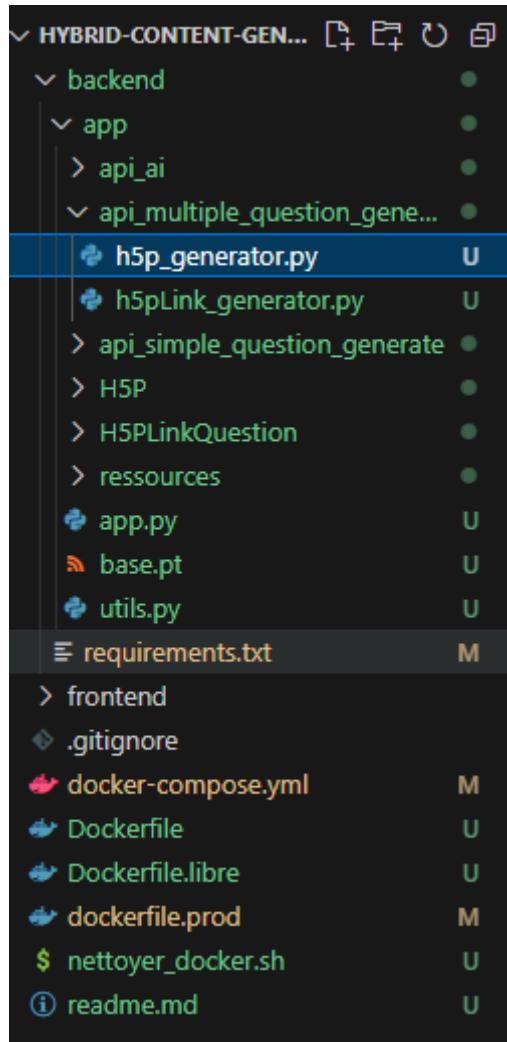


Figure 42: Fichier h5p_generator.

Paramètres de la fonction :

- **data_file_path**: Chemin vers le fichier de données (CSV ou Excel) contenant les questions, réponses, etc.
- **video_file_path**: Chemin vers le fichier vidéo utilisé pour les interactions H5P.
- **h5p_file_name**: Nom du fichier H5P à générer.
- **strict**: Booléen déterminant le mode strict pour les interactions H5P.
- **languages** : Liste des langues disponibles pour les sous-titres.
- **subtitles** : Liste des sous-titres pour la vidéo

```
def create_h5p_interactions_multiple(data_file_path, video_file_path, h5p_file_name, strict, languages=[], subtitles[]):
```

Figure 43: Fonction input create_h5p_interactions_multiple.

Définition des paramètres stricts :

- Configure les paramètres H5P en fonction du mode strict (`autoContinue`, `enableSolutionsButton`, `requireCompletion`).

```
# Définir les paramètres en fonction de la variable strict
if strict:
    autoContinue = False
    enableSolutionsButton = False
    requireCompletion = True
else:
    autoContinue = True
    enableSolutionsButton = True
    requireCompletion = False
```

Figure 44: create_h5p_interactions_multiple, code1.

Manipulation des fichiers et des chemins :

- Copie la vidéo spécifiée (`video_file_path`) dans le répertoire H5P.

```
# Copier la vidéo dans le répertoire H5P
shutil.copy2(video_file_path, './H5P/content/videos/video.mp4')
```

Figure 45: create_h5p_interactions_multiple, code2.

- Extrait l'audio de la vidéo pour la génération des sous-titres.

Chargement et manipulation des données :

- Charge les données du fichier spécifié (`data_file_path`) en utilisant `pandas`.

```
# Charger le fichier de données
if data_file_path.endswith('.csv'):
    df = pd.read_csv(data_file_path)
elif data_file_path.endswith('.xlsx'):
    df = pd.read_excel(data_file_path)
else:
    raise ValueError("Unsupported file format. Please provide a CSV or Excel file.")
```

Figure 46: create_h5p_interactions_multiple, code3.

- Calcule la durée totale de la vidéo.
- Tri et ajustement des durées des questions en fonction des données fournies.

Modification du fichier JSON de contenu (content.json) :

- Réinitialise les interactions dans le fichier JSON.
- Ajoute les interactions pour chaque question basée sur les données chargées.

```
# Traiter les durées des questions
durations_list = []
for i in range(len(df)):
    if pd.isnull(df.at[i, 'duration']):
        durations_list.append(total_video_duration)
    else:
        durations_list.append(float(df.at[i, 'duration']))

adjusted_durations = sorted(durations_list)
for i in range(1, len(adjusted_durations)):
    if adjusted_durations[i] == adjusted_durations[i - 1]:
        adjusted_durations[i] += 0.5

sorted_questions = df.assign(duration=adjusted_durations).sort_values(by='duration', ascending=True)
sorted_durations = adjusted_durations
```

Figure 47: create_h5p_interactions_multiple, code4.

Paramètres et options pour chaque interaction :

- Génère des UUID uniques pour chaque interaction.
- Prépare les réponses, les durées, les interactions H5P avec les paramètres appropriés.

```
# Ajouter les interactions
for i in range(num_questions):
    interaction_uuid = uuid.uuid4().hex

    # Préparer les réponses
    answers = []
    for col in df.columns:
        if col.startswith('reponse') and not pd.isnull(df.at[i, col]):
            result_col = 'result' + col[-1]
            correct = df.at[i, result_col] == 'v'
            answers.append({
                "text": f"<div>{df.at[i, col]}</div>\n",
                "correct": correct,
                "tipsAndFeedback": {
                    "tip": "",
                    "chosenFeedback": "",
                    "notChosenFeedback": ""
                }
            })

    duration = sorted_durations[i]

    if not pd.isnull(df.at[i, 'renvoi']):
        seekTo = df.at[i, 'renvoi']
    else:
        if strict:
            seekTo = trouver_nombre_avant(sorted_durations, duration) + 1
        else:
            seekTo = duration + 0.1
```

Figure 48: *create_h5p_interactions_multiple, code5.*

Écriture et finalisation du contenu JSON :

- Écrit les données mises à jour dans le fichier `content.json`.
- Ajoute les sous-titres au contenu JSON.

Création du fichier H5P :

- Compile tous les fichiers nécessaires dans un fichier H5P compressé.
- Retourne le chemin du fichier H5P généré.

```
# Créer le fichier H5P
h5p_file = zip_files_in_folder('./ressources/content/'f'{h5p_file_name}', './H5P')

return h5p_file
```

Figure 49: *create_h5p_interactions_multiple, code6.*

API python fichier generate_request.py :

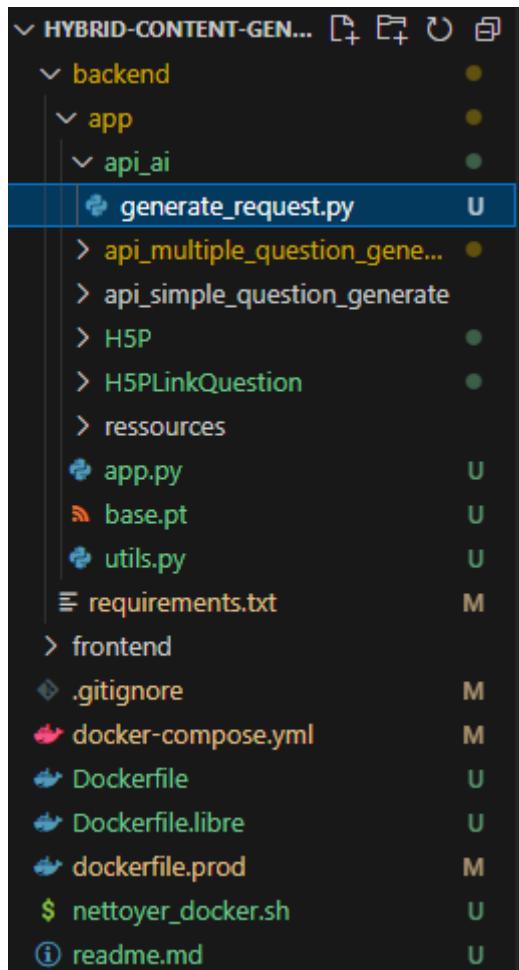


Figure 50: *genrate_request.py.*

Import des modules :

- `requests`: Pour effectuer des requêtes HTTP vers l'API llama3.
- `json, csv, os`: Pour la manipulation de données JSON, CSV et les opérations sur le système de fichiers.
- `unidecode`: Pour supprimer les caractères spéciaux du texte transcrit.
- `Flask, request, jsonify, send_file`: Pour créer une application web, gérer les requêtes HTTP et renvoyer des fichiers au client.
- `extract_audio, video_speech_to_text`: Fonctions personnalisées importées depuis `utils` pour extraire l'audio de la vidéo et transcrire le texte audio en utilisant un modèle.

```
import requests
import json
import csv
import os
from unidecode import unidecode
from flask import Flask, request, jsonify, send_file
from utils import extract_audio, video_speech_to_text
```

Figure 51: generate_request.py, imports.

Définition de l'application Flask :

- `app = Flask(__name__)`: Initialise une instance de l'application Flask.

Fonction generate_questions_and_answers :

```
def generate_questions_and_answers(transcribed_text):
```

Figure 52: generate_request.py, fonction generate_questions_and_answers.

- Utilise le texte transcrit pour générer une série de questions à choix multiples à l'aide de l'API llama3.
- Prépare un prompt décrivant les instructions et le texte transcrit, puis envoie une requête POST à l'API llama3.
- Traite la réponse JSON de l'API et retourne les questions générées.

```
payload = {
    "model": "llama3:instruct",
    "prompt": prompt,
    "stream": False
}
headers = {
    "Content-Type": "application/json",
    "User-Agent": "curl/7.68.0"
}

try:
    response = requests.post("http://10.3.154.166:8080/api/generate", data=json.dumps(payload), headers=headers)
    response.raise_for_status()
    return response.json().get("response", "")
except Exception as e:
    print(f'An error occurred: {e}')
```

Figure 53: generate_request.py, code1.

Fonctions utilitaires (`remove_special_characters`, `clean_quotes`, `save_csv`, `check_file_existence`):

- `remove_special_characters`: Utilise `unidecode` pour supprimer les caractères spéciaux du texte.

```
def remove_special_characters(text):
    return unidecode(text)
```

Figure 54: generate_request.py, remove_special_characters.

- `clean_quotes`: Supprime les guillemets autour du texte.

```
def clean_quotes(text):
    return text.strip('')'
```

Figure 55: generate_request.py, clean_quotes.

- `save_csv`: Enregistre le texte CSV nettoyé dans un fichier spécifié, en ajoutant un en-tête et en utilisant des virgules comme séparateurs.

```
def save_csv(csv_text, file_path):
    cleaned_csv_text = remove_special_characters(csv_text)

    # Diviser le contenu en lignes et supprimer toute ligne vide
    rows = [row for row in cleaned_csv_text.split('\n') if row.strip()]

    # Supprimer les deux premières lignes inutiles
    rows = rows[2:]

    # Remplacer les points-virgules par des virgules
    rows = [row.replace(';', ',') for row in rows]

    # Ajout de l'en-tête
    header = 'question,reponseA,reponseB,reponseC,reponseD,reponseE,duration'
    rows.insert(0, header)

    with open(file_path, 'w', newline='', encoding='utf-8') as csvfile:
        csvwriter = csv.writer(csvfile, quoting=csv.QUOTE_MINIMAL)
        for row in rows:
            # Nettoyer les guillemets pour chaque champ
            cleaned_row = [clean_quotes(field) for field in row.split(',')]
            csvwriter.writerow(cleaned_row)
```

Figure 56: generate_request.py, save_csv.

- `check_file_existence`: Vérifie si un fichier existe à l'emplacement spécifié.

```
def check_file_existence(file_path):
    full_path = os.path.abspath(file_path)
    if not os.path.exists(full_path):
        print(f"Erreur: le fichier {full_path} est introuvable.")
        return False
    return True
```

Figure 57: generate_request.py, check_file_existence.

Fonction process_video_to_csv :

- Prend en paramètre le chemin vers la vidéo (`video_path`) et le nom du fichier CSV à créer (`csv_file_name`).
- Initialise le chemin vers le fichier de sortie CSV (`csv_file_path`).

```
audio_path = "./ressources/videos/audio.mp3"
csv_file_path = f"./ressources/data/{csv_file_name}.csv"
```

Figure 58: generate_request.py, process_video_to_csv, code1.

- Vérifie l'existence des fichiers vidéo et audio requis.

```
if not check_file_existence(video_path):
    return
if not check_file_existence(audio_path):
    return
```

Figure 59: generate_request.py, process_video_to_csv, code 2.

- Extrait l'audio de la vidéo et transcrit le texte audio en utilisant `extract_audio` et `video_speech_to_text`.

```
extract_audio(video_path, audio_path)

model_path = "./base.pt"
transcribed_text = video_speech_to_text(audio_path, model_path)
```

Figure 60: generate_request.py, process_video_to_csv, code3.

- Utilise `generate_questions_and_answers` pour générer les questions à choix multiples à partir du texte transcrit. Puis on enregistre les questions générées dans un fichier CSV à l'aide de `save_csv`.

```
csv_text = generate_questions_and_answers(transcribed_text)
save_csv(csv_text, csv_file_path)
```

Figure 61: generate_request.py, process_video_to_csv, code4.

Point d'entrée principal (`__name__ == "__main__"`) :

- Lance l'application Flask en mode débogage (`debug=True`), ce qui permet de voir les erreurs directement dans le terminal.

API python fichier utils.py :

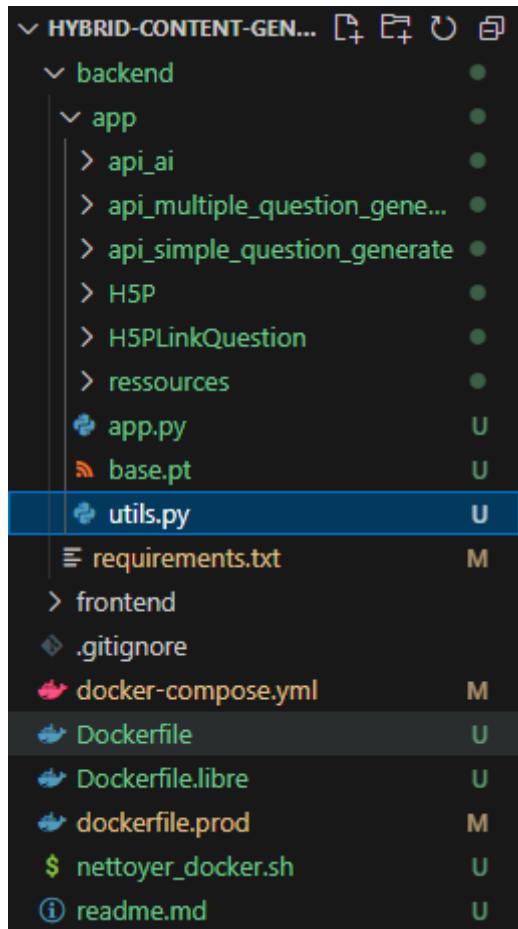


Figure 62: `utils.py`.

Classe `WhisperModel`

```

class WhisperModel:
    _model = None

    @classmethod
    def get_model(cls, model_path):
        if cls._model is None:
            cls._model = whisper.load_model(model_path)
        return cls._model

```

Figure 63:`utils`, class `Whisper`.

La classe Singleton pour le modèle Whisper est bien implémentée. Cela permet de ne charger le modèle qu'une seule fois, ce qui est efficace en termes de mémoire.

Fonction `translate_text`

```
def translate_text(text, source_lang, target_lang):
    url = "http://libretranslate:5000/translate"
    payload = {
        "q": text,
        "source": source_lang,
        "target": target_lang,
        "format": "text",
    }
    headers = {
        "Content-Type": "application/json"
    }

    response = requests.post(url, data=json.dumps(payload), headers=headers)

    if response.status_code == 200:
        translated_text = response.json().get('translatedText', '')
        return translated_text
    else:
        return "Error: Failed to translate text"
```

Figure 64:utils, translate_text.

La fonction utilise une API externe pour la traduction. Assurez-vous que le service `libretranslate` est disponible et accessible depuis votre environnement Docker.

Fonction `load_data_file`

```
def load_data_file(file_path):
    try:
        if file_path.endswith('.csv'):
            data = pd.read_csv(file_path)
        elif file_path.endswith('.xlsx'):
            data = pd.read_excel(file_path)
        else:
            raise ValueError("Unsupported file format. Please provide a CSV or Excel file.")
        return data
    except Exception as e:
        raise ValueError(f"Error loading data file: {str(e)}")
```

Figure 65:utils, load_data_file.

La fonction gère les fichiers CSV et Excel. Elle utilise `pandas` pour lire les fichiers, ce qui est performant et fiable.

Fonction video_speech_to_text

```
def video_speech_to_text(audio_path: str, model_path: str) -> str:
    try:
        model = WhisperModel.get_model(model_path)

        result = model.transcribe(audio_path)

        return result["text"]
    except Exception as e:
        return str(e)
```

Figure 66:utils, video_speech_to_text.

La fonction utilise Whisper pour transcrire l'audio en texte. La gestion des exceptions permet de capturer et de retourner les erreurs.

Fonction modify_video_restrictions

```
def modify_video_restrictions(fichier_json, valeur):
    with open(fichier_json, 'r') as f:
        contenu = json.load(f)

    if 'override' in contenu:
        contenu['override']['preventSkippingMode'] = 'both' if valeur else 'none'

    with open(fichier_json, 'w') as f:
        json.dump(contenu, f, indent=4)
```

Figure 67: utils, modify_video_restriction.

La fonction modifie un fichier JSON pour ajuster les restrictions vidéo. Elle vérifie la clé override et met à jour en conséquence.

Fonction zip_files_in_folder

```
def zip_files_in_folder(zip_path, folder_path):
    with zipfile.ZipFile(zip_path, 'w', zipfile.ZIP_DEFLATED) as zipf:
        for root, dirs, files in os.walk(folder_path):
            for file in files:
                file_path = os.path.join(root, file)
                rel_path = os.path.relpath(file_path, folder_path)
                zipf.write(file_path, arcname=rel_path)
    return zip_path
```

Figure 68: utils, zip_files_in_folder.

La fonction crée une archive ZIP d'un dossier donné. L'utilisation de `zipfile.ZipFile` est appropriée pour cette tâche.

Fonction `video_duration`

```
def video_duration(filename):
    try:
        result = subprocess.run(
            ["ffprobe", "-v", "error", "-show_entries", "format=duration",
             "-of", "default=noprint_wrappers=1:nokey=1", filename],
            stdout=subprocess.PIPE,
            stderr=subprocess.PIPE,
            check=True
        )
        return float(result.stdout)
    except subprocess.CalledProcessError as e:
        print(f"Error occurred while running ffprobe: {e.stderr.decode().strip()}")
        raise
    except FileNotFoundError:
        print("ffprobe not found. Please ensure ffmpeg is installed and ffprobe is accessible in your PATH.")
        raise
```

Figure 69: utils, video_duration

La fonction utilise `ffprobe` pour obtenir la durée de la vidéo. Assurez-vous que `ffprobe` est disponible dans votre environnement Docker.s

Fonction `extract_audio`

```
def extract_audio(video_path, audio_path):
    video = VideoFileClip(video_path)
    audio = video.audio
    audio.write_audiofile(audio_path)
    video.close()
```

Figure 70: utils, extract_audio.

La fonction extrait l'audio d'une vidéo et l'enregistre. moviepy est bien utilisé pour cette tâche.

Fonction modify_video_url

```
def modify_video_url(json_file_path, video_url):
    # Lire le fichier JSON
    with open(json_file_path, 'r') as f:
        data = json.load(f)

    # Modifier le chemin de la vidéo pour mettre l'URL
    data['interactiveVideo']['video']['files'] = [
        {
            "path": video_url,
            "mime": "video/YouTube",
            "copyright": {
                "license": "U"
            }
        }
    ]

    # Écrire les modifications dans le fichier JSON
    with open(json_file_path, 'w') as f:
        json.dump(data, f, indent=4)

    print("Chemin de la vidéo modifié et fichier JSON mis à jour.")
```

Figure 71: utils, modify_video_url.

La fonction met à jour l'URL de la vidéo dans un fichier JSON. La structure du JSON est respectée et les modifications sont bien appliquées.

Fonction add_subtitles

```
def add_subtitles(json_file_path, languages, output_path):
    # Lire le fichier JSON
    with open(json_file_path, 'r') as f:
        data = json.load(f)

    # Vider les sous-titres existants
    data['interactiveVideo']['video']['textTracks']['videoTrack'] = []

    # Ajouter les nouveaux sous-titres
    for lang in languages:
        subtitle = {
            "label": f"Subtitles_{lang}",
            "kind": "subtitles",
            "srcLang": lang,
            "track": {
                "path": f"files/subtitles_{lang}.vtt",
                "mime": "text/plain",
                "copyright": {
                    "license": "U"
                }
            }
        }
        data['interactiveVideo']['video']['textTracks']['videoTrack'].append(subtitle)

    # Écrire les modifications dans le fichier JSON
    with open(json_file_path, 'w') as f:
        json.dump(data, f, indent=4)
```

Figure 72: utils, add_subtitles.

La fonction ajoute des sous-titres à une vidéo en modifiant le fichier JSON. La gestion des chemins de fichiers et la structure JSON sont correctes.

Fonction vtt_files_creation

Initialisation du Dictionnaire de Sous-titres :

- Si des sous-titres sont fournis, ils sont ajoutés dans un dictionnaire après avoir été séparés en phrases.
- Si aucun sous-titre n'est fourni, la fonction utilise `video_speech_to_text` pour transcrire l'audio en texte.

```
if subtitles and any(subtitles):
    print("dic initialiser")
    subtitles_dict = {lang: subs.split('. ') for lang, subs in zip(languages, subtitles) if subs}
    for lang, subs in subtitles_dict.items():
        subtitles_dict[lang] = [phrase.strip() for phrase in subs if phrase.strip()]
```

Figure 73: utils, vtt_files_creation.

Traduction des Sous-titres :

- Si des sous-titres sont fournis mais qu'ils sont vides pour certaines langues, ces sous-titres sont traduits depuis le français (ou une autre langue source) vers les langues cibles en utilisant la fonction `translate_text`.

```
# If subtitles are provided but are empty, fill them with translations
for lang in languages:
    if lang not in subtitles_dict:
        if lang == "fr":
            text = video_speech_to_text(audio_path, model_path)
            phrases = text.split('. ')
            subtitles_dict["fr"] = [phrase.strip() for phrase in phrases if phrase.strip()]
        else:
            traductions = []
            for phrase in subtitles_dict["fr"]:
                if phrase: # Verify that the phrase is not empty
                    traduction = translate_text(phrase, "fr", lang)
                    traductions.append(traduction)
            subtitles_dict[lang] = traductions
```

Figure 74: utils, vtt_files_creation, traduction.

Calcul de la Durée par Phrase :

- La durée totale de la vidéo est divisée par le nombre de phrases pour déterminer la durée d'affichage de chaque phrase.

Formatage du Temps :

- Une fonction interne `format_temps` convertit les temps en format hh:mm.mmm, nécessaire pour les fichiers VTT.

```
def format_temps(temp):
    heures = int(temp // 3600)
    minutes = int((temp % 3600) // 60)
    secondes = int(temp % 60)
    millisecondes = int((temp - int(temp)) * 1000)
    return f"{heures:02}:{minutes:02}:{secondes:02}.{millisecondes:03}"
```

Figure 75: utils, vtt_files_creation, formatage.

Création des Fichiers VTT :

- Une fonction interne `creer_fichier_vtt` crée les fichiers VTT en utilisant les phrases traduites et les durées calculées.

```
def creer_fichier_vtt(subtitles, chemin_fichier):
    print(f"Création du fichier VTT: {chemin_fichier}")
    os.makedirs(os.path.dirname(chemin_fichier), exist_ok=True)
    with open(chemin_fichier, 'w', encoding='utf-8') as f:
        f.write("WEBVTT\n\n")
        for start, end, text in subtitles:
            f.write(f"{format_temps(start)} --> {format_temps(end)}\n")
            f.write(f"\n{text}\n\n")
```

Figure 76: utils, vtt_files_creation, Création fichier vtt.

Sauvegarde des Fichiers VTT :

- Les fichiers VTT sont sauvegardés dans le chemin de sortie spécifié pour chaque langue.

```

# Create subtitle files
chemins_fichiers = {}
for lang in languages:
    subtitles_lang = subtitles_dict.get(lang)
    if not subtitles_lang:
        print(f"Traduction des phrases en français pour la langue {lang}")
        traductions = []
        for i, phrase in enumerate(subtitles_dict["fr"]):
            if phrase: # Verify that the phrase is not empty
                traduction = translate_text(phrase, "fr", lang)
                traductions.append(traduction)
                print(f"Traduction en {lang} de la phrase {i + 1}: {traduction}")
        subtitles_lang = traductions

    subtitles_vtt = []
    temps_courant = 0.0
    for phrase in subtitles_lang:
        start_time = temps_courant
        end_time = start_time + duree_par_phrase
        subtitles_vtt.append((start_time, end_time, phrase))
        temps_courant = end_time

    chemin_fichier = os.path.join(output_path, f'subtitles_{lang}.vtt')
    chemins_fichiers[lang] = chemin_fichier
    creer_fichier_vtt(subtitles_vtt, chemin_fichier)

print("Fichiers de sous-titres créés :")
for lang in languages:
    print(chemins_fichiers[lang])

```

Figure 77: utils, vtt_files_creation, sauvegarde.

Fonction find_previous_duration

```

def find_previous_duration(liste, nombre):
    # Vérifie si le nombre est le premier élément de la liste
    if liste[0] == nombre:
        return 0
    # Trouve l'index du nombre dans la liste
    index = liste.index(nombre)
    # Retourne le nombre juste avant
    return liste[index - 1]

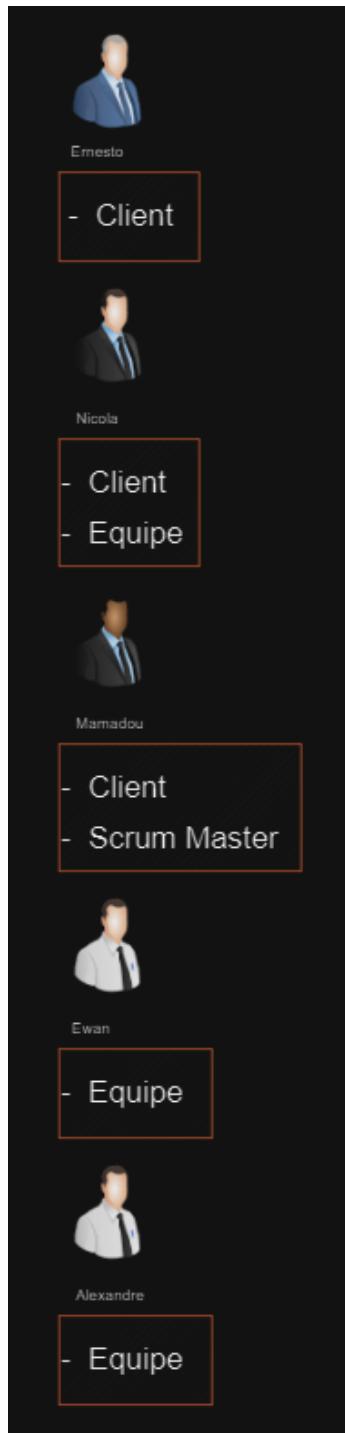
```

Figure 78: utils,find_previous_duration.

La fonction retourne la durée précédente dans une liste donnée. Elle est simple et efficace.

Planification du Projet

Pour la partie de la méthode de planification et de gestion du projet, nous avons choisi la méthode Scrum. Celle-ci se compose de plusieurs rôles : l'équipe, qui est composée des membres développant le projet. les clients, qui sont les personnes imposant leurs demandes et contraintes. Et enfin, le Scrum Master, qui se charge de veiller au respect de la mise en pratique des méthodes. Voici les rôles des membres du projet :



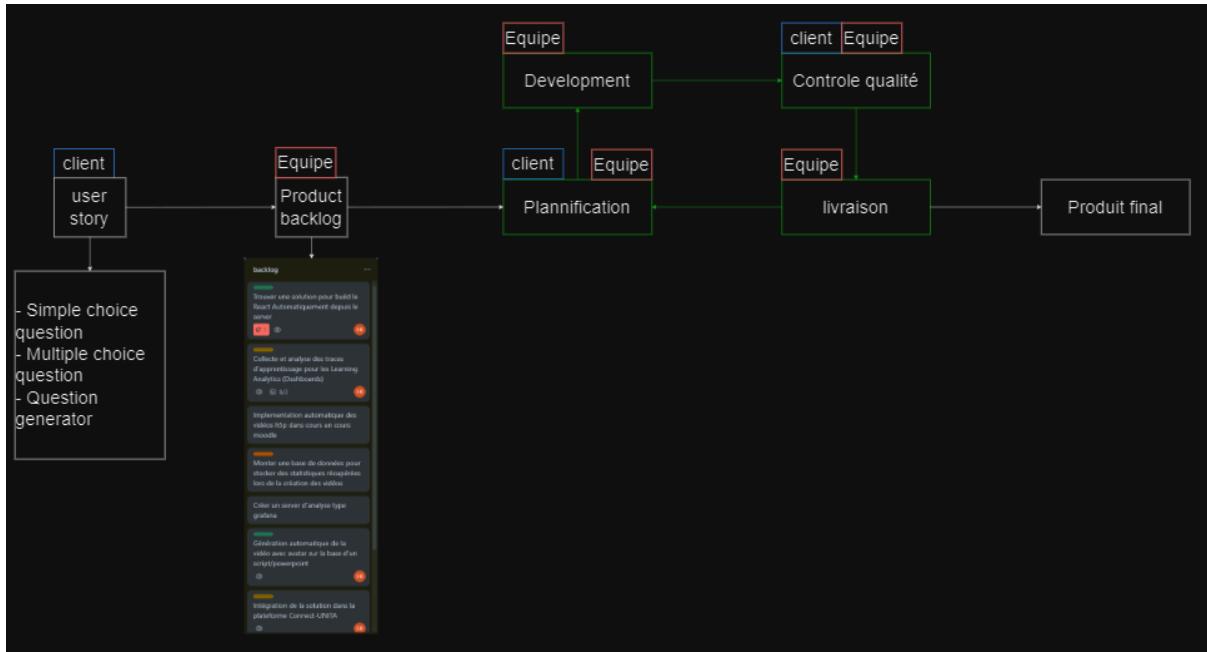


Figure 79: Schéma SCRUM .

Avant mon arrivée, Exposito, Evain et Lamine ont décrit une vision du projet du point de vue de l'utilisateur. Ils ont pu créer une liste de tâches à accomplir et établir un planning. De plus, à mon arrivée, nous avons également pu redéfinir une "user story" afin de m'éclairer sur toutes les visions du projet et les types de tâches à effectuer. Pour la suite, chaque semaine, l'équipe du projet se réunit pour examiner les progrès de chacun sur le projet. Ces réunions hebdomadaires permettent de partager les avancées, de discuter des défis rencontrés et de planifier les prochaines actions. Cette pratique favorise la transparence et la coordination au sein de l'équipe, assurant que tous les membres restent alignés sur les objectifs du projet et nous permettant d'ajuster rapidement nos stratégies si nécessaire.

Planning prévisionnel

Tâches	2-15 avril	16-30 avril	1-15 mai	16-31 mai	1-15 juin	16-30 juin
1. Analyse des designs pedagogiques						
2. Mettre en place une solution d'Hybridation						
3. Développement d'une AppWeb						
4. Création de services complémentaires						
5. Savoir manipuler le RapidMooc						
6. Learning Analytics						
7. Test et Déploiement						

Figure 80: planning prévisionnel.

Tâches du Projet

1. Analyse des designs pédagogiques
 - Durée : 2-15 avril
 - Description : Cette tâche consiste à examiner et analyser les designs pédagogiques existants pour comprendre les besoins et les exigences du projet.
2. Mettre en place une solution d'hybridation
 - Durée : 16-30 avril

- Description : Durant cette période, une solution pour combiner différentes approches pédagogiques (traditionnelle et numérique) sera mise en place.
3. Développement d'une AppWeb
- Durée : 1-15 mai
 - Description : Le développement de l'application web se fera sur toute la période de mai. Cela inclut la conception, le codage, et les tests initiaux de l'application.
4. Création de services complémentaires
- Durée : 16 mai au 30 juin
 - Description : Cette tâche implique la création de services supplémentaires qui compléteront l'application web, comme des API ou des services d'intégration.
5. Savoir manipuler le RapidMooc
- Durée : 16-30 juin
 - Description : Cette phase consiste à former l'équipe ou les utilisateurs à l'utilisation de RapidMooc, un outil de création de contenu vidéo éducatif.
6. Learning Analytics
- Durée : 16-30 juin
 - Description : Pendant cette période, les outils d'analyse des données d'apprentissage seront développés et mis en place pour suivre et évaluer l'efficacité pédagogique des contenus créés.
7. Test et Déploiement
- Durée : 1 mai au 30 juin
 - Description : Cette dernière phase consiste à tester l'ensemble du système (application web et services complémentaires) et à déployer la solution finale.

Tableau RACI

Tâches	Alexandre PABOEUF	Ewan Burosovitch	Nicolas Evain	Mamadou Lamine	Christian La Borderie	Sandrine Pottier	Stéphanie Chalivoix	Ernesto Exposito
1	R	C	A	I	I	A	I	I
2	R	C	R	C	C	C	C	C
3	R	R	A	C	C	C	C	C
4	R	R	R	C	C	C	C	C
5	R	I	I	A	A	A	I	I
6	A	A	A	R	I	I	I	I
7	R	R	R	A	A	C	C	A

Figure 81: Tableau RACI.

Légendes :

- Tâche 1 : Analyse des designs pédagogiques
- Tâche 2 : Mettre en place une solution d'Hybridation

- Tâche 3 : Développement d'une AppWeb
 - Tâche 4 : Création de services complémentaires
 - Tâche 5 : Savoir manipuler le RapidMooc
 - Tâche 6 : Learning Analytics
 - Tâche 7 : Tests et Déploiement
-
- R : Responsable
 - A : Acteur
 - C : Consulté
 - I : Informé

Outils de Collaboration

Pour répartir les tâches et suivre l'avancement du projet, nous avons créé un groupe sur Trello, un logiciel de collaboration en ligne. Trello nous permet d'organiser notre travail de manière visuelle et interactive, facilitant ainsi la communication et la gestion des tâches. Le groupe comprend les membres du projet Connect Unita et de Irekia, Mamadou Lamine, ainsi que Nicolas Evain, Alexandre PABOEUF

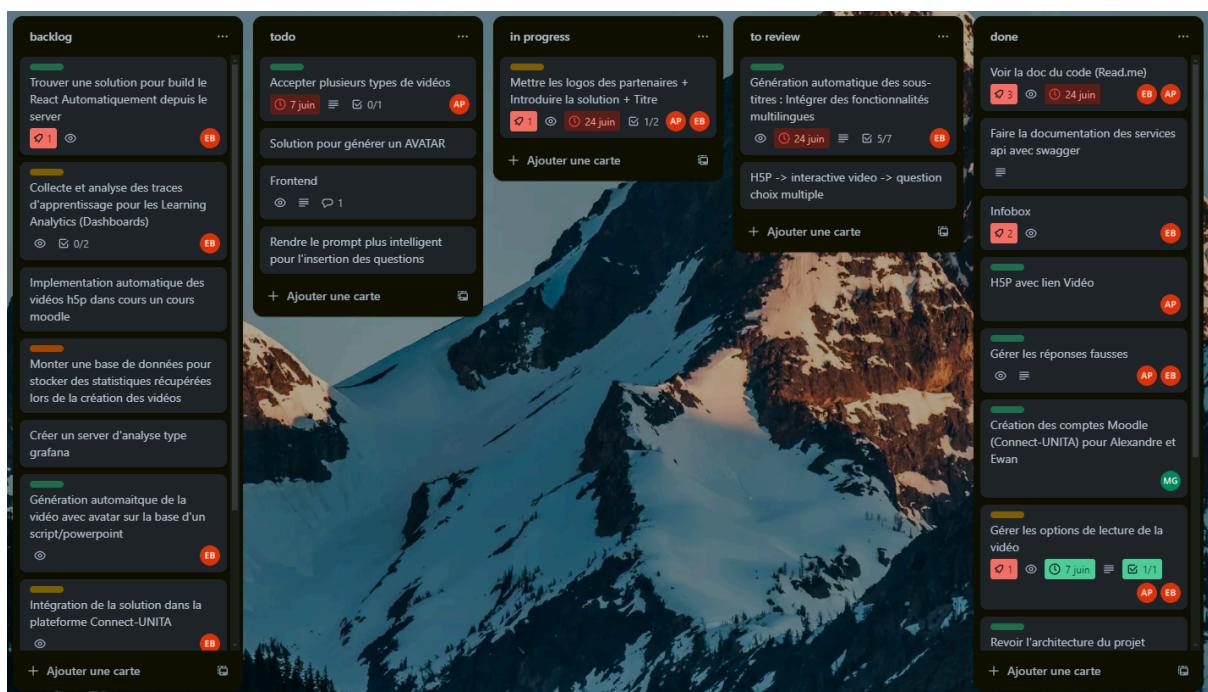


Figure 82: Trello.

Chaque tâche est représentée par une carte sur Trello, et ces cartes sont organisées en différentes colonnes correspondant aux étapes du processus de travail. Nous avons mis en place cinq colonnes principales pour suivre l'avancement des tâches :

- Backlog ⇒ Liste des tâches prévues pour l'avenir.
- Todo ⇒ Liste des tâches à effectuer.
- In Progress ⇒ Liste des tâches en cours.
- To Review ⇒ Liste des tâches terminées en attente de validation.
- Done ⇒ Liste des tâches validées.

Pour chaque tâche, nous :

- Donnons un nom.
- Assignons un ou plusieurs membres.
- Ajoutons une étiquette en fonction de l'importance de la tâche (importante, moyenne, faible).
- Fixons une date limite.
- Fournissons une description.
- Créons une checklist avec les sous-tâches.

Voici un exemple de tâche.

Génération des questions par intelligence artificielle

Dans la liste done

Membres: AP, EB

Étiquettes: Important

Notifications: Suivie

Ajouter à la carte: Membres, Étiquettes, Checklist, Dates, Pièce jointe, Image de couvert..., Champs personnali...

Description: Générer un fichier excel templatisé avec les questions et les réponses via un outil Gen AI

ATTENTION : que se passe-t-il pour une vidéo de 1 heure ??? (faire en batch)

Checklist:

- ✓ Speech-to-text
- ✓ Ecrire le prompt
- ✓ Appel vers l'api d'un outil gen AI (chatgpt)
- ✓ Récupérer le csv ou xlsx
- ✓ Revoir le prompt enginering
- ✓ Création d'une VM Ubuntu
- ✓ Déploiement d'un modèle, type LLaMA 3, 7 milliards de paramètres
- ✓ Création d'une API pour récupérer une requête du serveur en backend
- ✓ Développement d'une fonction python; gérer les requêtes; envoie du prompt modèle
- ✓ Fonction python; Récupère les questions et réponses et les mettre dans un Excel

Masquer les tâches cochées | Supprimer

Power-Ups: + Ajouter des Pow...

Automatisation: + Ajouter un bouton

Actions: → Déplacer, Copier, Créer un modèle, Archiver, Partager

Ajouter un élément

Activité: Afficher les détails

AP | Écrivez un commentaire...

Figure 83: Tâche Trello.

Mes travaux

Tâches réalisées

Tâches Backend

1. Préparation et Configuration Initiale
 - Installer les dépendances : Assurez-vous que toutes les bibliothèques nécessaires (comme pandas, shutil, etc.) sont installées.
 - Configurer l'environnement : Mettre en place l'environnement de développement (création de dossiers, configuration de fichiers).
2. Gestion des Données
 - Charger les fichiers de données : Importer et lire les fichiers CSV ou Excel contenant les questions et les réponses.
 - Manipuler les données : Préparer et ajuster les données pour les interactions vidéo.
3. Manipulation des Fichiers Vidéo et Audio
 - Copier les fichiers vidéo : Copier les fichiers vidéo nécessaires vers les dossiers appropriés.
 - Extraire l'audio : Extraire l'audio des fichiers vidéo pour la transcription..
4. Transcription et Sous-titres
 - Transcrire l'audio : Utiliser un modèle de transcription pour générer des sous-titres à partir de l'audio.
 - Créer des sous-titres VTT : Générer des fichiers de sous-titres en format VTT pour les langues spécifiées.
5. Gestion des Interactions Vidéo
 - Calculer les durées : Calculer et ajuster les durées pour chaque question/interactions.
 - Générer des UUIDs : Créer des identifiants uniques pour chaque interaction.
 - Construire les interactions : Préparer les données d'interactions vidéo avec les questions, réponses, et autres paramètres.
 - Mettre à jour le fichier content.json : Ajouter les nouvelles interactions dans le fichier JSON de contenu.
6. Création et Gestion des Fichiers H5P
 - Ajouter les sous-titres : Intégrer les fichiers de sous-titres générés dans le contenu H5P.
 - Compresser les fichiers H5P : Créer un fichier H5P compressé à partir du contenu préparé.

7. Utilisation de l'API et des Services Externes

- Utiliser des APIs de transcription : Si nécessaire, intégrer des services de transcription via des APIs externes.
- Gérer les appels API : Implémenter des appels API pour la conversion et la transcription audio.

8. Validation et Vérification

- Vérifier les données : S'assurer que les données importées et manipulées sont correctes et complètes.
- Faire une revue de code afin d'améliorer les aspects pouvant l'être.
- Tester les interactions : Tester les interactions vidéo générées pour garantir qu'elles fonctionnent comme prévu.

9. Déploiement

- Préparer le serveur : Configurer le serveur pour accueillir l'application web.
- Déployer le code : Uploader et configurer le code backend sur le serveur universitaire.
- Configurer les permissions : S'assurer que les fichiers et dossiers ont les permissions appropriées.

Tâches Fronted

10. Conception et mise en place de React

- Initialiser un nouveau projet ReactJS. Afin d'avoir les bases d'un framework solide.
- Configurer les routes avec React Router.

11. La partie Accueil

- Créer un composant Home pour la page d'accueil
- Ajouter le texte de bienvenue et une description des fonctionnalités du site.

12. La partie SimpleChoiceQuestion

- Créer un composant SimpleChoiceQuestion avec les champs de formulaire nécessaires (nom de la vidéo, fichier de données, fichier vidéo, URL de la vidéo, sous-titres, etc.).
- Ajouter la validation des champs du formulaire (vérification des formats de fichier, URL valide, etc.).
- Ajouter des messages d'information avec le composant InfoBox.
- Ajouter la fonctionnalité pour sélectionner les langues des sous-titres et gérer les changements de texte.
- Implémenter la soumission du formulaire avec fetch pour envoyer les données au serveur.
- Gérer l'état de la soumission et les éventuelles erreurs.

13. La partie MultipleChoiceQuestion

- Créer un composant MultipleChoiceQuestion avec les champs de formulaire nécessaires (nom de la vidéo, fichier de données, fichier vidéo, URL de la vidéo, sous-titres, etc.).
- Ajouter la validation des champs du formulaire (vérification des formats de fichier, URL valide, etc.).
- Ajouter des messages d'information avec le composant InfoBox.
- Ajouter la fonctionnalité pour sélectionner les langues des sous-titres et gérer les changements de texte.
- Implémenter la soumission du formulaire avec fetch pour envoyer les données au serveur.
- Gérer l'état de la soumission et les éventuelles erreurs.

14. La partie QuestionGenerator

- Créer un composant QuestionGenerator pour générer des questions automatiquement.
- Valider les champs du formulaire.
- Utiliser fetch pour soumettre les données et gérer l'état de soumission.

15. La partie Tutoriel & FAQ

- Créer un composant TutorialFAQ pour afficher les tutoriels et les FAQ.
- Ajouter le contenu des tutoriels et des FAQ.

16. Design du site

- Concevoir la mise en page générale du site (en-tête, pied de page, navigation).
- Insertion des logos partenaires
- Choisir un thème et une palette de couleurs cohérente.
- Appliquer le style aux différents composants (formulaires, boutons, etc.).
- Assurer que le site est réactif et s'affiche correctement sur tous les appareils (desktop, tablette, mobile).

Compétences apportées

Des tâches comme la création de nouveaux services dans l'API Python m'ont permis de développer des applications. J'ai pu développer de nouvelles fonctionnalités pour l'API, représentées par ses routes dans le code. Le respect du cahier des charges était une priorité, et les différentes contraintes m'ont obligé à m'adapter constamment, comme l'utilisation de solutions open source, gratuites et déployables en local. J'ai également pu mettre en place des principes algorithmiques adaptés à nos besoins, tels qu'un Singleton. Les conventions de nommage ont été respectées dans ce code, nous avons choisi un étiquetage en anglais, en suivant les conventions apprises en cours. Les revues de code ont également mis en lumière les mises à jour à réaliser dans les semaines à venir. L'apprentissage d'un nouveau type

de documentation avec Swagger a été très bénéfique, même si la réalisation d'un bon vieux README a été effectuée.

Lors de ce projet, j'ai pu manipuler des données nécessaires aux fonctionnalités de l'application. Ces données sont transmises par fichiers plats, soit en CSV soit en XLSX, et sont fournies par l'utilisateur. J'ai également amélioré le frontend, qui était initialement en HTML/CSS, en développant une application avec le framework ReactJS pour une meilleure sécurité. De plus, nous avons déployé un modèle d'IA Llama3 sur un serveur Ubuntu, permettant ainsi de lui faire des requêtes et de générer des questions à partir d'un texte.

J'ai également développé l'API Python qui constitue le backend de l'application, celle-ci est appelée au niveau du framework ReactJS. Nous avons également réalisé une autre API Python chargée de gérer les requêtes pour le modèle Llama3.

Nous avons également créé une connexion SSH avec un certificat pour sécuriser la connexion. Nous avons pu gérer quelques failles du code, comme les fuites de mémoire, qui ont été résolues grâce à l'adoption de bonnes pratiques, telles que l'ajout de modèles algorithmiques. J'ai aussi pu gérer des données de traces analytiques avec des outils dédiés.

Ce projet m'a demandé de comprendre les besoins multilingues, ainsi que les fonctionnalités possibles à ajouter à notre solution et la manière de les intégrer. J'ai mis en place un suivi hebdomadaire du projet et un Trello, ce qui a permis une meilleure compréhension des différentes tâches réalisées et à réaliser dans le cadre du projet.

Je me suis parfaitement intégré dans l'équipe. Nous avons été rapidement inclus dans les différentes réunions afin de nous présenter à l'équipe Connect Unita et de présenter notre solution à ses membres. Que ce soit en anglais pour les échanges internationaux ou en français lors des réunions internes, j'ai toujours su me positionner et m'intégrer lors de ces événements pour présenter le meilleur de notre travail. Je me suis également intégré aux différentes tâches en collaboration avec le deuxième stagiaire travaillant sur le projet. J'ai pu directement récupérer des tâches à réaliser seul ou en collaboration. Nous avons pu comprendre l'architecture interne des membres des deux équipes, Connect Unita et IREKIA.

Création du Projet sur GitLab

1. Crédit de la clé :

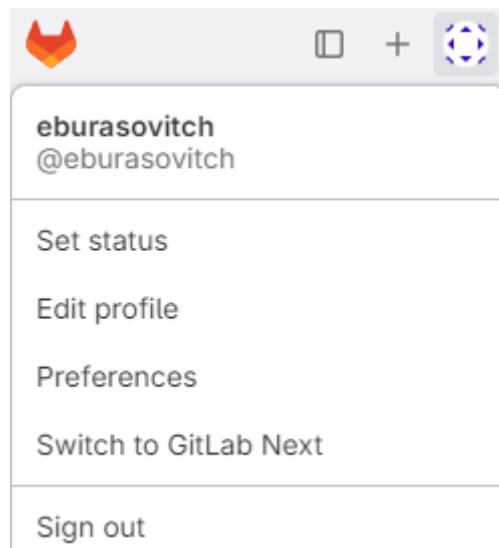


Figure 84: GitLab.

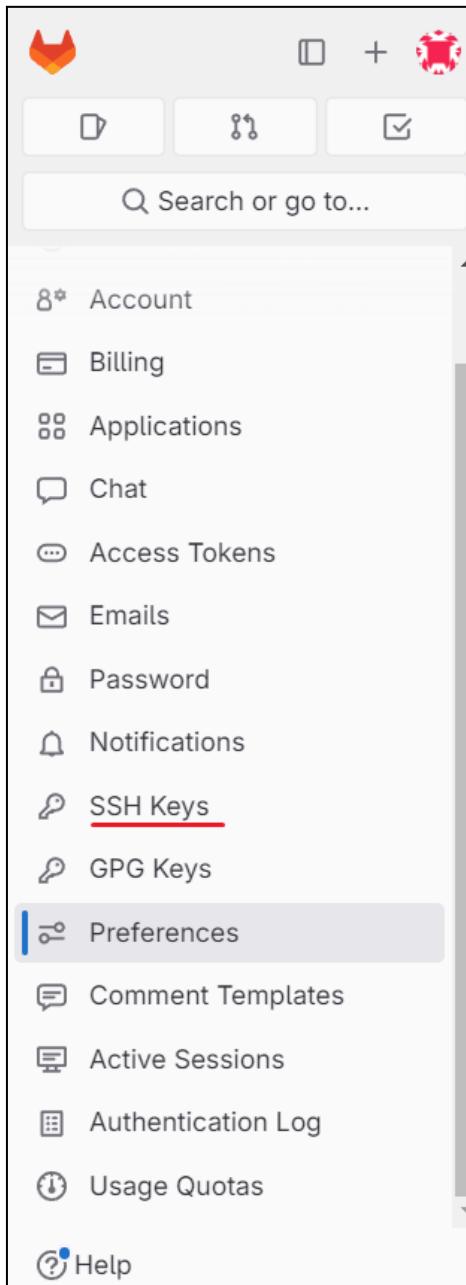


Figure 85: GitLab.

SSH Keys						
SSH keys allow you to establish a secure connection between your computer and GitLab. SSH fingerprints verify that the client is connecting to the correct host. Check the current instance configuration .						
Your SSH keys <small>1</small>						
Title	Key	Usage type	Created	Last used	Expires	Actions
Utilisateur 2@DESKTOP-4HJT5MO	b8:62:81:4b:e5:e1:46:89:2a:ed:9b:1b:0e:17:5e:68	Authentication & Signing	21 hours ago	21 hours ago	Never	Revoke Delete

Figure 86: GitLab Clé.

```

Utilisateur 2@DESKTOP-4HJT5MO MINGW64 ~/.ssh
$ history
1
2
3
4
5
6
7
8 cd /
9 ls
10 cd ~
11 pwd
12 cd /
13 cd c
14 cd ..
15 ls -al
16 cd c
17 ls
18 cd Users
19 ls
20 cd Utilisateur 2
21 cd Utilisateur 2
22 cd Utilisateur\ 2/
23 cd /
24 cd ~
25 pwd
26 ls -al
27 cd .ssh
28 mkdir .ssh
29 ls -l
30 lq -al | .ssh
31 lq -al | grep .ssh
32 ls -al | grep .ssh
33 cd .ssh
34 ssh-keygen -t ed25519
35 ssh-keygen -t ed25519
36 ls -al
37 eval $(ssh-agent -s)
38 ssh-add ./
39 pwd
40 ssh-add /c/Users/Utilisateur\ 2/.ssh/id_rsa_ed25519
41 ls -al
42 nano config
43 cat config
44 history

```

Figure 87: GitLab commandes.

- Accédez au répertoire souhaité (si le chemin contient un espace, par exemple 'Utilisateur 2', insérez un antislash pour gérer l'espace, comme dans 'Utilisateur\ 2').
- Créer le dossier .ssh
- Vérifiez si le dossier a été créé avec succès en utilisant la commande ls -al | grep .ssh, qui liste tous les éléments contenant '.ssh' dans le répertoire courant.
- Ensuite, accédez à ce dossier avec la commande "cd .ssh"
- Choisir une clé la plus adaptée :

Algorithm	Public key	Private key
ED25519 (preferred)	id_ed25519.pub	id_ed25519
ED25519_SK	id_ed25519_sk.pub	id_ed25519_sk
ECDSA_SK	id_ecdsa_sk.pub	id_ecdsa_sk
RSA (at least 2048-bit key size)	id_rsa.pub	id_rsa
DSA (deprecated)	id_dsa.pub	id_dsa
ECDSA	id_ecdsa.pub	id_ecdsa

Figure 88: GitLab clé.

- On génère une clé avec le bon algo : “ssh-keygen -t ed25519”
- On vérifie que l’agent est bien actif : “eval \$(ssh-agent -s)”
- On ajoute une clé spécifique à l’agent ssh : “ssh-add /c/Users/Utilisateur\2/.ssh/id_gitlab_ed25519”
- Pour finir on rentre dans config ce code :

```
Utilisateur 2@DESKTOP-4HJT5MO MINGW64 ~/ssh
$ cat config
Host gitlab.com
PreferredAuthentications publickey
IdentityFile ~/.ssh/id_gitlab_ed25519
```

Figure 89: GitLab authentification clé public.

afin de se connecter automatiquement au gitLab sans rentrer son mail, son mdp.

- Se rendre sur la homepage de gitlab et choisir le projet

Figure 90: GitLab.

- Copier l’adresse ssh

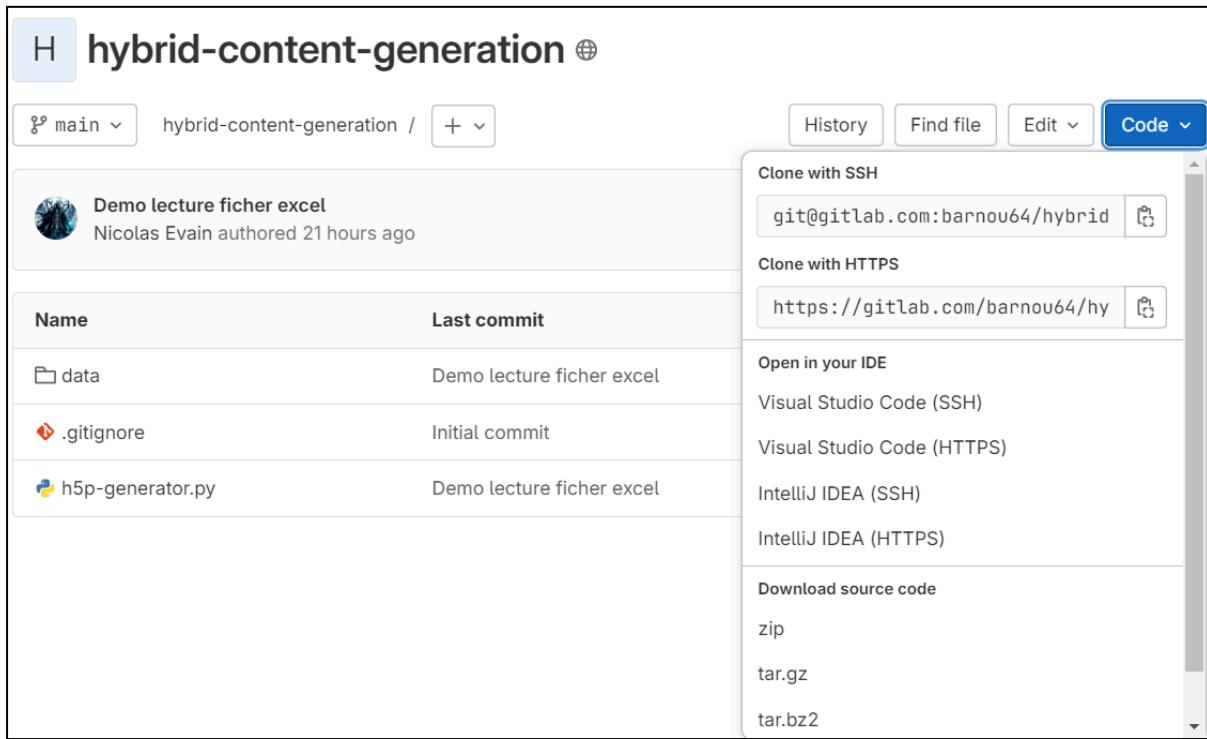


Figure 91: GitLab adresse ssh.

- Coller l'adresse sur VisualStudioCode

```
PS C:\Users\Utilisateur 2\Desktop\Universite Montaury\Licence 3\stage\IREKIA\Code\hybrid-content-generation>
git clone git@gitlab.com:barnou64/hybrid-content-generation.git
Cloning into 'hybrid-content-generation'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (9/9), 7.16 KiB | 3.58 MiB/s, done.
```

Figure 92: GitLab VsCode.

Comment partager du code

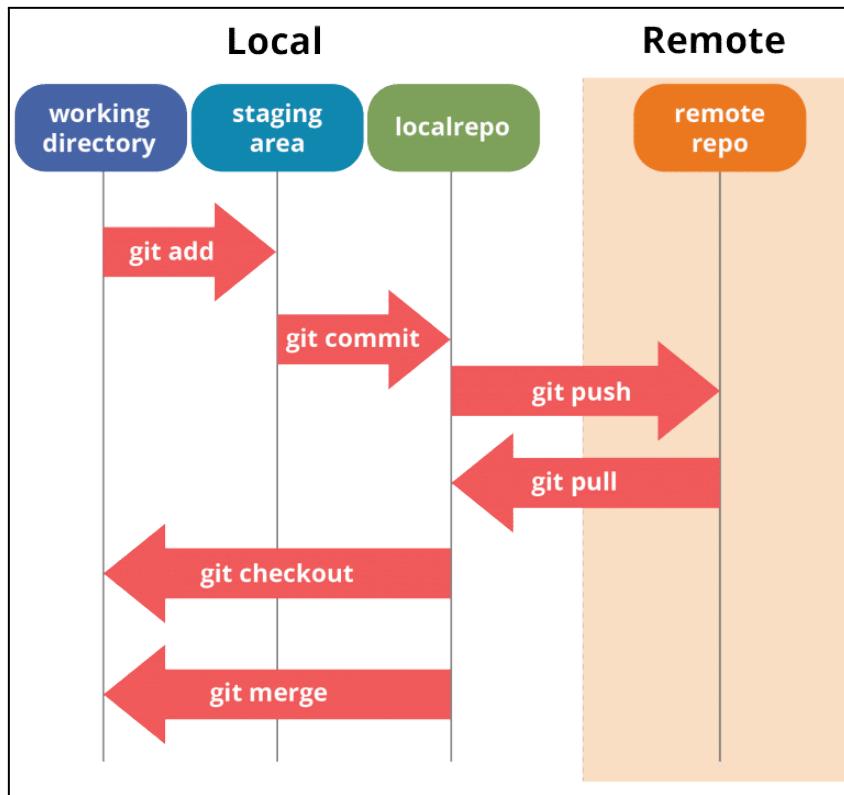


Figure 93: GitLab commandes.

Installer python sur machine + VSCode:

- Se rendre sur le site web "<https://www.python.org/downloads/>" puis télécharger la version adaptée au système d'exploitation
- Lancer vscode
- Fichier > Préférences > Paramètres

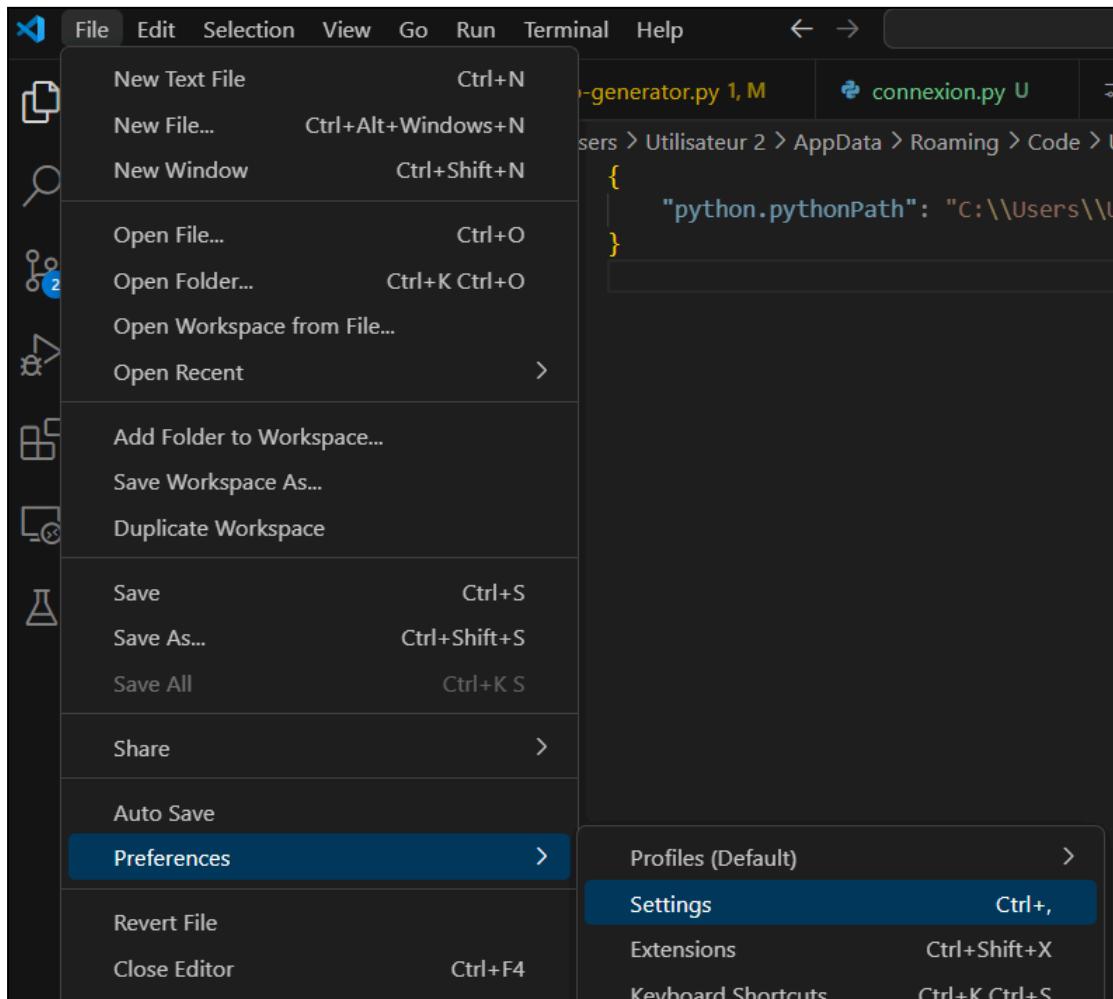


Figure 94: Installer Python VScode.

- Taper “Python Path” dans la barre de recherche

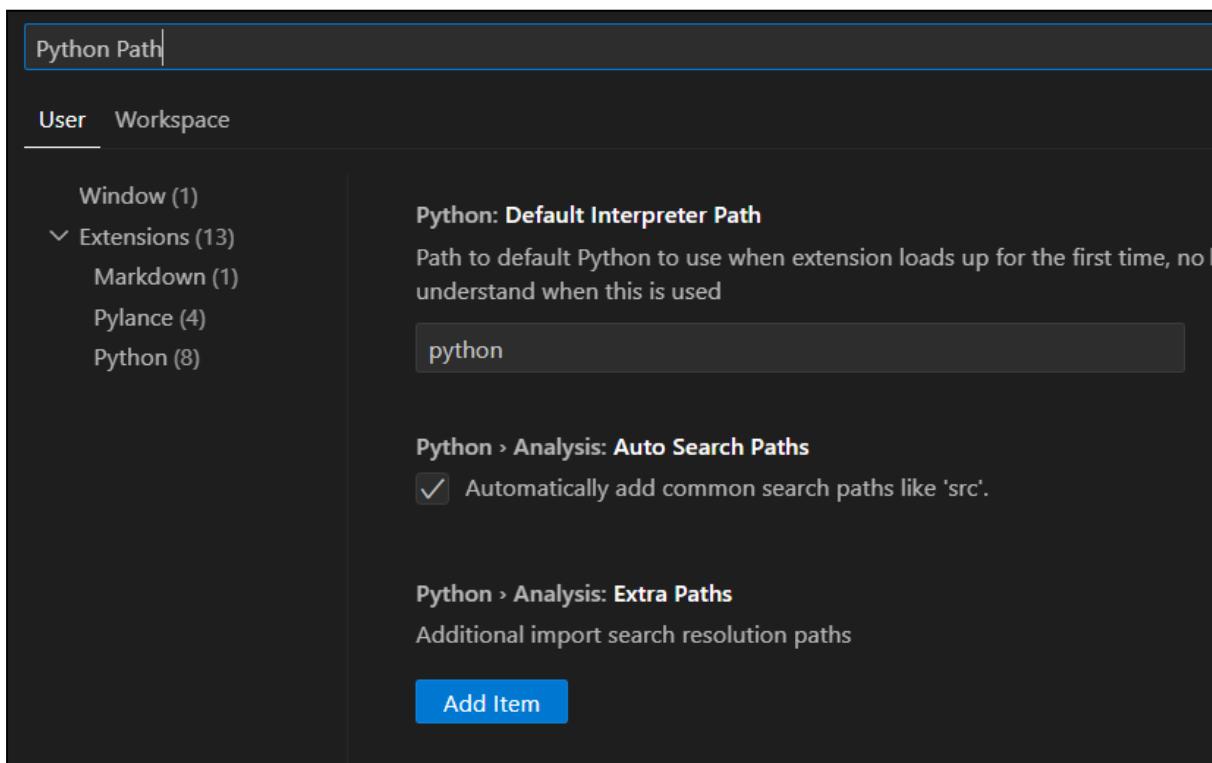


Figure 95: Installer Python VScode..

- Puis cliquer sur “Edit settings.json” et rentrer le chemin du fichier python.exe

```

1  {
2    "python.pythonPath": "C:\\Users\\Utilisateur 2\\AppData\\Local\\Programs\\Python\\Python312\\python.exe",
3  }
4

```

Figure 96: Installer Python VScode..

Utilisation de Docker

Pourquoi Docker

L'utilisation de Docker dans nos projets apporte de nombreux avantages, notamment la portabilité, l'isolation des processus et la facilité de déploiement. Grâce à Docker, nous pouvons encapsuler notre application et ses dépendances dans un conteneur, garantissant ainsi que le code a les mêmes fonctionnalités quel que soit l'endroit où il est exécuté.

Fichier docker-compose.yml

Le fichier docker-compose.yml est une partie importante de la configuration de Docker. Il nous permet de définir et de gérer plusieurs conteneurs Docker en une seule commande. Ce fichier spécifie la configuration de chaque service, leurs interactions, leur mise en réseau et les volumes requis. Cela simplifie grandement la configuration de l'ensemble de nos environnements de développement et de production.

```

version: '3.8'

services:
  python-api:
    build:
      context: .
      dockerfile: dockerfile.prod
    ports:
      - "8080:8080"
    volumes:
      - ./backend/app:/app
    networks:
      - hcgnetwork

  web:
    build:
      context: .
      dockerfile: Dockerfile
    ports:
      - "8081:80"
    networks:
      - hcgnetwork

  libretranslate:
    build:
      context: .
      dockerfile: Dockerfile.libre # Utilisez un Dockerfile spécifique pour LibreTranslate si nécessaire
      args:
        with_models: "true" # Passer l'argument with_models=true pour le téléchargement des modèles
    ports:
      - "8083:5000"
    environment:
      - http_proxy=http://cache.univ-pau.fr:3128
      - https_proxy=https://cache.univ-pau.fr:3128
    networks:
      - hcgnetwork

networks:
  hcgnetwork:
    driver: bridge

```

Figure 97: docker-compose.yml.

Fichier dockerfile

Un Dockerfile est un script qui contient une série d'instructions permettant d'assembler une image Docker. Chaque instruction du Dockerfile ajoute un calque à l'image finale. Ce fichier définit exactement comment notre application doit être construite, quelles dépendances doivent être installées et quelles configurations doivent être appliquées. Cela nous permet de créer des images Docker reproductibles et cohérentes à chaque fois.

```

# Utiliser une version spécifique de Node.js
FROM node:14-alpine AS builder

WORKDIR /usr/src/app

# Configurations du proxy
ENV http_proxy http://cache.univ-pau.fr:3128
ENV https_proxy http://cache.univ-pau.fr:3128

RUN yarn config set proxy $http_proxy && yarn config set https-proxy $https_proxy

# Copier les fichiers package.json et yarn.lock
COPY ./frontend/React-Landing-Page-Template/package.json ./frontend/React-Landing-Page-Template/yarn.lock ./

# Nettoyer le cache Yarn et installer les dépendances
RUN yarn cache clean && yarn install --frozen-lockfile

# Ajouter les dépendances manquantes
RUN yarn add mini-css-extract-plugin schema-utils chalk

# Copier le reste du code de l'application
COPY ./frontend/React-Landing-Page-Template .

# Réinstaller les dépendances pour s'assurer que tout est correctement installé
RUN yarn install

# S'assurer que react-scripts est exécutable
RUN chmod +x node_modules/.bin/react-scripts

# Construire l'application
RUN yarn run build

# Utiliser nginx pour servir le build
FROM nginx:alpine
COPY --from=builder /usr/src/app/build /usr/share/nginx/html
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]

```

Figure 98: Dockerfile.

```

# Utiliser l'image de base Python
FROM python:3.8-slim

# Variables d'environnement pour le proxy si nécessaire
ENV http_proxy http://cache.univ-pau.fr:3128
ENV https_proxy http://cache.univ-pau.fr:3128

# Répertoire de travail
WORKDIR /app

# Copier les fichiers nécessaires
COPY .

# Script pour télécharger et intégrer les modèles de langues si besoin
RUN if [ "$with_models" = "true" ]; then \
    pip install libretranslate && \
    libretranslate --download-models && \
    echo "Models downloaded successfully" ; \
else \
    pip install libretranslate ; \
fi

# Exposer le port 5000 (optionnel)
EXPOSE 5000

# Commande par défaut pour lancer l'API
CMD ["libretranslate", "--host", "0.0.0.0", "--port", "5000"]

```

Figure 99: Dockerfile.libre

```

FROM python:3.9-slim

ENV PYTHONDONTWRITEBYTECODE 1
ENV PYTHONUNBUFFERED 1
ENV http_proxy http://cache.univ-pau.fr:3128
ENV https_proxy http://cache.univ-pau.fr:3128

WORKDIR /app

COPY ./backend/requirements.txt .

RUN apt-get update && apt-get upgrade -y &&
apt-get install -y zip ffmpeg git && pip install --upgrade pip && pip install --no-cache-dir -Ur requirements.txt

EXPOSE 8080

ENV http_proxy=
ENV https_proxy=

CMD ["python", "app.py"]

```

Figure 100: Dockerfile.prod

Récapitulatif des Commandes Docker

1. Docker Images

Lister les images Docker disponibles sur votre machine :
docker images

Télécharger une image depuis Docker Hub :
docker pull <nom_image>:<tag>

2. Docker Containers

Lister les conteneurs en cours d'exécution :
docker ps

Lister tous les conteneurs (en cours d'exécution et arrêtés) :
docker ps -a

Démarrer un conteneur :
docker start <id_conteneur>

Arrêter un conteneur :
docker stop <id_conteneur>

Exécuter un nouveau conteneur :
docker run -d --name <nom_conteneur> <nom_image>:<tag>

Supprimer un conteneur :
docker rm <id_conteneur>

3. Docker Volumes

Lister les volumes Docker :

`docker volume ls`

Créer un volume Docker :

`docker volume create <nom_volume>`

Supprimer un volume Docker :

`docker volume rm <nom_volume>`

4. Docker Networks

Lister les réseaux Docker :

`docker network ls`

Créer un réseau Docker :

`docker network create <nom_reseau>`

Supprimer un réseau Docker :

`docker network rm <nom_reseau>`

Dépendance importer

Toutes les dépendances sont importées grâce au fichier requirements.txt, qui est installé dans le Dockerfile. Voici ces dépendances :

```
Flask==3.0.3
Flask_Cors==4.0.0
openpyxl==3.1.2
pandas==2.2.2
moviepy
pydub
openai>=1.0.0
isodate
youtube-dl
pytube
pydub
speechrecognition
langdetect
requests==2.26.0
unidecode
openai-whisper
moviepy
torch
transformers
flasgger
```

Figure 101: requirement.txt.

Choix et Intégration des Fonctionnalités

Toutes les dépendances ont été choisies en comparant les différentes solutions trouvées lors de nos recherches. Nous avons donc effectué un comparatif entre les solutions existantes.

Speech-To-Text

Critères	whisper	DeepSpeech	Google Speech-to-Text	Azure Speech Service
Temps de transcription	Modéré	Modéré	Rapide	Rapide
Quantité langues dispo.	Très large	Moyen	Très large	Très large
Précision	Élevée	Élevée	Très élevée	Très élevée
Facilité d'utilisation	Oui	Oui	Oui	Oui
Support de formats audio	Large	Large	Large	Large
Portabilité	Oui	Oui	Oui	Oui
Gratuit	Oui	Oui	Limité (payant au-delà)	Limité (payant au-delà)

Figure 102: Tableau comparaison outils Speech-To-Text.

Nous avons choisi d'utiliser Whisper pour plusieurs raisons clés. Tout d'abord, il offre une grande précision, essentielle pour une transcription fiable. Ensuite, Whisper prend en charge un large éventail de langues, ce qui le rend idéal pour les applications multilingues. Bien que le temps de transcription soit modéré, ce point est compensé par la gratuité de Whisper, contrairement à d'autres options payantes après un certain nombre d'utilisations. De plus, il est facile à utiliser et compatible avec divers formats audio, assurant une intégration fluide dans nos systèmes existants.

Traduire le texte

Critères	LibreTranslate	Facebook NLLB	MTranslate	Whisper	Google T5
Précision	Élevée	Faible	Élevée	Faible	Élevée
Quantité de langues dispo.	Élevée	Élevée	Élevée	Faible	Faible
Rapidité de traitement	Élevé	Faible	Modéré	Modéré	Élevé
Support de formats	Large	Large	Large	Large	Large
Gratuit	Oui	Oui	Limité (payant au-delà)	Oui	Limité (payant au-delà)
Facilité d'utilisation	Oui	Oui	Oui	Oui	facile

Figure 103: Tableau comparaison outils Traduction.

Nous avons choisi d'utiliser LibreTranslate pour plusieurs raisons impérieuses. Tout d'abord, sa grande précision garantit des traductions de haute qualité, cruciales pour notre travail. De plus, il prend en charge un grand nombre de langues, augmentant ainsi sa polyvalence. LibreTranslate se distingue également par sa vitesse de traitement, permettant une utilisation efficace et rapide. Il prend en charge de nombreux formats et est entièrement gratuit, ce qui en fait une option financièrement viable. Enfin, la facilité d'utilisation de LibreTranslate en fait un outil pratique et accessible à tous les utilisateurs.

Générateur de Questions/ Réponses IA

Critère	LLAMA 3	ChatGPT	Mistral	Gemini
Temps de réponse API	Modéré	Rapide	Modéré	Rapide
Précision	Élevée	Très élevée	Élevée	Très élevée
Facilité d'utilisation API	Moyenne	Très élevée	Moyenne	Très élevée
Documentation API	Complète	Très complète	Complète	Très complète
Support de formats CSV	Oui	Oui	Oui	Oui
Coût	Oui	Payant	Limité quota	Payant

Figure 104: Tableau comparaison outils LLM.

Nous avons choisi de déployer LAMA 3 localement pour plusieurs raisons. Premièrement, il offre une grande précision, essentielle pour nos besoins en traitement de données. De plus, le coût de LAMA 3 est plus avantageux que celui d'autres options payantes comme ChatGPT et Gemini. Bien que les temps de réponse de l'API soient modérés et que la facilité d'utilisation soit moyenne, ces aspects sont compensés par une documentation complète de l'API et la prise en charge du format CSV. Ces fonctionnalités font de LAMA 3 une solution pratique et rentable pour notre infrastructure locale.

Choix de l'IA pour création Vidéo Avatar

Pour évaluer les meilleures solutions pour l'avenir du projet, nous avons comparé Synthesia et HeyGen, deux technologies permettant de remplacer l'enseignant par un avatar. Les critères analysés incluent les templates, les avatars, la voix, le temps de création des vidéos, l'interface utilisateur, les sous-titres, la traduction, la résolution, l'assistance et l'API. HeyGen a obtenu une note finale légèrement supérieure (8,47) par rapport à Synthesia (8,14). À ce jour, aucune décision d'abonnement n'a été prise, mais cette recherche fournira une base solide pour prendre des décisions futures visant à améliorer l'enseignement avec des avatars numériques.

	Templates	Avatar	Voix	Temps Vidéos	Utilisateur	Sous-titres	Traduction	Résolution	Assistance	API	note finale
Synthesia	8	8	7	10	10	7	7	9	9	7	8,14
HeyGen	8	8	9	10	10	7	8	9	6	9	8,47
Poids	8	8	8	9	6	7	7	7	6	10	

Figure 105: Recherche opérationnelle IA.

Stratégies	Note finale
Synthesia	8,14
HeyGen	8,47

Figure 106: Recherche opérationnelle IA.

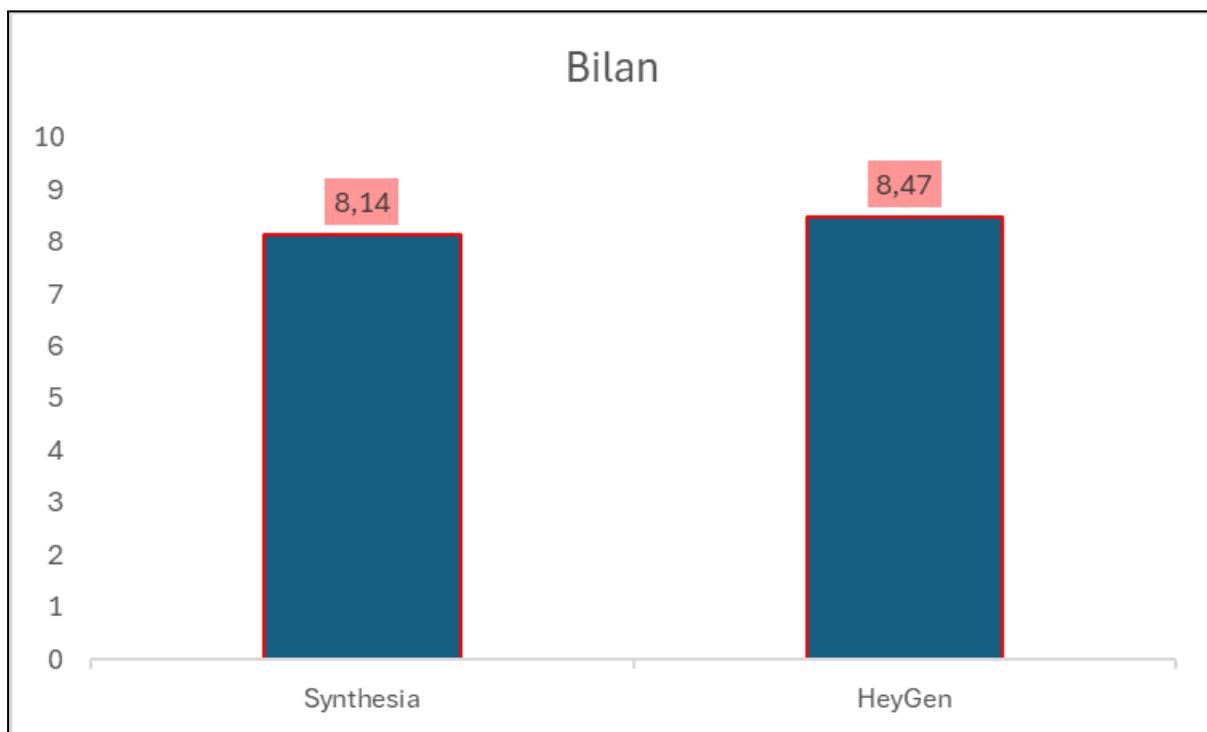


Figure 107: Recherche opérationnelle IA.

Conclusion

Lors de ce projet, j'ai pu mettre en valeur mes différentes compétences dans un cadre de travail réel comportant diverses contraintes. Ce projet m'a permis de découvrir l'importance cruciale de l'hybridation des enseignements et de l'intégration des nouvelles technologies, telles que l'intelligence artificielle et les Learning Analytics, dans l'enseignement supérieur. Il m'a également montré l'importance d'automatiser les services de création des ressources pédagogiques pour faciliter le travail des enseignants. Les contraintes linguistiques m'ont rappelé l'importance de rendre nos travaux accessibles à tous.

En ce qui concerne le développement, ce stage m'a montré comment m'adapter à des contraintes imposées qui impactent directement la qualité du projet. J'ai également appris à déployer une solution dans un environnement avec des contraintes telles que celles d'un serveur proxy. Ces compétences et connaissances seront des atouts précieux pour ma future carrière, ouvrant de nombreuses opportunités dans le domaine de l'innovation pédagogique et du développement des technologies éducatives.

Annexe

Table des illustrations

<i>Figure 1 : Organigramme IREKIA.</i>	8
<i>Figure 2: Tableau travaux solutions existantes .</i>	13
<i>Figure 3: OpenVPN Connect.</i>	14
<i>Figure 4: Navigation</i>	14
<i>Figure 5: Simple Choice Question.</i>	15
<i>Figure 6: Fichier Excel Simple.</i>	16
<i>Figure 7: Video input.</i>	16
<i>Figure 8: Navigation Mode Strict.</i>	17
<i>Figure 9: Langues.</i>	17
<i>Figure 10: Sous titre Manuel.</i>	17
<i>Figure 11: Multiple Choice Question.</i>	18
<i>Figure 12: Fichier Excel Multiple.</i>	18
<i>Figure 13: Question Generator.</i>	19
<i>Figure 14: Output Question Generator.</i>	19
<i>Figure 15: Tutorial.</i>	20
<i>Figure 16: RapidMooc.</i>	21
<i>Figure 17: Moodle H5P.</i>	21
<i>Figure 18: Moodle H5P.</i>	22
<i>Figure 19: Vidéo Exemple.</i>	22
<i>Figure 20: Diagramme Cas d'utilisation.</i>	23
<i>Figure 21: Diagramme Séquence Question Choix Simple P1.</i>	24
<i>Figure 22: Diagramme Séquence Question Choix Simple P2.</i>	25
<i>Figure 23: Diagramme Séquence Question Generator.</i>	25
<i>Figure 24: Diagramme Séquence Tutoriel & FAQ.</i>	26
<i>Figure 25 : Architecture globale.</i>	27
<i>Figure 26 :Schémas Architecture.</i>	27
<i>Figure 27: Architecture VSCode.</i>	29
<i>Figure 28: Architecture VSCode Backend.</i>	29
<i>Figure 29: Architecture VSCode Frontend.</i>	30
<i>Figure 30: Fichier app.py.</i>	30
<i>Figure 31: Import app.py.</i>	31
<i>Figure 32:Route (/).</i>	31
<i>Figure 33: Route (/interactive_video_multiple_choice_question).</i>	32
<i>Figure 34: Route (/interactive_video_multiple_choice_question), code1.</i>	32

	76
<i>Figure 35: Route (/interactive_video_multiple_choice_question), code2.</i>	32
<i>Figure 36: Route (/interactive_video_simple_choice_question).</i>	33
<i>Figure 37: Route (/interactive_video_simple_choice_question), code.</i>	33
<i>Figure 38: Route (/upload_video).</i>	33
<i>Figure 39: Route (/upload_video), code1.</i>	33
<i>Figure 40: Route (/upload_video), code2.</i>	34
<i>Figure 41: Route (/upload_video), code3.</i>	34
<i>Figure 42: Fichier h5p_generator.</i>	35
<i>Figure 43: Fonction input create_h5p_interactions_multiple.</i>	35
<i>Figure 44: create_h5p_interactions_multiple, code1.</i>	36
<i>Figure 45: create_h5p_interactions_multiple, code2.</i>	36
<i>Figure 46: create_h5p_interactions_multiple, code3.</i>	36
<i>Figure 47: create_h5p_interactions_multiple, code4.</i>	37
<i>Figure 48: create_h5p_interactions_multiple, code5.</i>	37
<i>Figure 49: create_h5p_interactions_multiple, code6.</i>	38
<i>Figure 50: genrate_request.py.</i>	38
<i>Figure 51: genrate_request.py, imports.</i>	39
<i>Figure 52: genrate_request.py, fonction generate_questions_and_answers.</i>	39
<i>Figure 53: genrate_request.py, code1.</i>	39
<i>Figure 54: genrate_request.py, remove_special_characters.</i>	40
<i>Figure 55: genrate_request.py, clean_quotes.</i>	40
<i>Figure 56: genrate_request.py, save_csv.</i>	40
<i>Figure 57: genrate_request.py, check_file_existence.</i>	40
<i>Figure 58: genrate_request.py, process_video_to_csv, code1.</i>	41
<i>Figure 59: genrate_request.py, process_video_to_csv, code 2.</i>	41
<i>Figure 60: genrate_request.py, process_video_to_csv, code3.</i>	41
<i>Figure 61: genrate_request.py, process_video_to_csv, code4.</i>	41
<i>Figure 62: utils.py.</i>	42
<i>Figure 63: utils, class Whisper.</i>	42
<i>Figure 64:utils, translate_text.</i>	43
<i>Figure 66:utils, video_speech_to_text.</i>	44
<i>Figure 67: utils, modify_video_restriction.</i>	44
<i>Figure 68: utils, zip_files_in_folder.</i>	44
<i>Figure 69: utils, video_duration</i>	45
<i>Figure 70: utils, extract_audio.</i>	45
<i>Figure 71: utils, modify_video_url.</i>	46
<i>Figure 72: utils, add_subtitles.</i>	46
<i>Figure 73: utils, vtt_files_creation.</i>	47
<i>Figure 74: utils, vtt_files_creation, traduction.</i>	47
<i>Figure 75: utils, vtt_files_creation, formatage.</i>	48
<i>Figure 76: utils, vtt_files_creation,Création fichier vtt.</i>	48
<i>Figure 77: utils, vtt_files_creation, sauvegarde.</i>	49
<i>Figure 78: utils,find_previous_duration.</i>	49
<i>Figure 79: Schéma SCRUM .</i>	50
<i>Figure 80: Planning prévisionnel .</i>	50
<i>Figure 81: Tableau RACI.</i>	51
<i>Figure 82: Trello.</i>	52
<i>Figure 83: Tâche Trello.</i>	53
<i>Figure 84: GitLab.</i>	57
<i>Figure 85: GitLab.</i>	58
<i>Figure 86: GitLab Clé.</i>	58
<i>Figure 87: GitLab commandes.</i>	59
<i>Figure 88: GitLab clé.</i>	60

	77
<i>Figure 89: GitLab authentification clé public.</i>	60
<i>Figure 90: GitLab.</i>	60
<i>Figure 91: GitLab adresse ssh.</i>	61
<i>Figure 92: GitLab VsCode.</i>	61
<i>Figure 93: GitLab commandes.</i>	62
<i>Figure 94: Installer Python VScode.</i>	63
<i>Figure 95: Installer Python VScode..</i>	63
<i>Figure 96: Installer Python VScode..</i>	63
<i>Figure 97: docker-compose.yml.</i>	65
<i>Figure 98: Dockerfile.</i>	66
<i>Figure 99: Dockerfile.libre</i>	66
<i>Figure 100: Dockerfile.prod</i>	67
<i>Figure 101: requirement.txt.</i>	68
<i>Figure 102: Tableau comparaison outils Speech-To-Text.</i>	69
<i>Figure 103: Tableau comparaison outils Traduction.</i>	69
<i>Figure 104: Tableau comparaison outils LLM.</i>	70
<i>Figure 105: Recherche opérationnelle IA.</i>	70
<i>Figure 106: Recherche opérationnelle IA.</i>	70
<i>Figure 107: Recherche opérationnelle IA.</i>	71