

Licence MIASHS Deuxième Année

Rapport de projet informatique

Bibliothèque Intelligente

Projet réalisé du 10 septembre 2024 au 6 janvier 2025

Github : <https://github.com/Ayb92k/BibliothèqueIntelligente.git>

Membres du groupe

OUAJJOU Ayoub 43009775

Kpinso-folly Joël 43007549

Remerciements

Nous tenons à exprimer notre profonde gratitude envers toutes les personnes qui ont contribué à la réalisation de ce projet intitulé « **Bibliothèque Intelligente** ».

Tout d'abord, nous remercions chaleureusement **Mr Delbot**, pour son accompagnement précieux, ses conseils éclairés et sa disponibilité tout au long du développement de ce projet.

Nous adressons nos sincères remerciements aux membres de notre équipe pour leur engagement, leur collaboration exemplaire et leur persévérance, qui ont permis de surmonter les défis rencontrés et de faire avancer ce projet vers son aboutissement.

Enfin, nous sommes reconnaissants envers **l'université de Nanterre**, qui a mis à notre disposition les ressources et les outils nécessaires pour mener à bien ce travail.

Ce projet a été une aventure enrichissante, rendue possible grâce à l'engagement de chacun. Nous vous adressons nos plus sincères remerciements pour votre contribution inestimable.

Table des matières

1	Introduction	4
2	Bilan	5
2.1	Analyse des contraintes	5
2.2	Intégration des contraintes dans les recommandations	5
2.3	Personnalisation	5
2.4	Tests et ajustements	5
3	Implémentation	6
3.1	Mise en place de l'environnement de développement	6
3.2	Interface utilisateur	6
3.3	Intégration de l'API OpenAI pour les recommandations	6
3.4	Gestion des recommandations par défaut	
4	Difficultés rencontrées	7
5	Bilan	8
5.1	Conclusion	8
5.2	Perspectives	8
6	Annexes	11
A	Cahier des charges	11
B	Exemple d'exécution du projet	13
C	Manuel utilisateur	15

1 Introduction

Le projet « **Bibliothèque Intelligente** » a été imaginé pour répondre à un besoin croissant dans la gestion de contenu numérique. Dans un monde où les utilisateurs jonglent avec une multitude de ressources multimédias – livres, musiques, vidéos et autres fichiers – il devient essentiel de disposer d’un outil simple, intuitif, et performant pour organiser ces contenus. Le défi principal était d’offrir une solution capable de s’adapter aux préférences des utilisateurs tout en leur proposant des recommandations personnalisées et pertinentes.

La particularité de ce projet réside dans l’intégration de l’intelligence artificielle, en s’appuyant sur l’API OpenAI, pour générer des suggestions adaptées en fonction des interactions passées des utilisateurs. L’objectif était d’exploiter les dernières avancées technologiques pour non seulement faciliter la gestion de bibliothèques personnelles, mais aussi enrichir l’expérience utilisateur grâce à des recommandations contextuelles et ciblées. Cependant, nous avons également anticipé les éventuelles limitations de l’IA en implémentant un système de recommandations par défaut, garantissant ainsi une continuité du service en toutes circonstances.

Ce projet a également été conçu pour être accessible à tous, avec une interface utilisateur moderne et épurée. Nous avons porté une attention particulière à la convivialité de l’application, en veillant à ce que les fonctionnalités principales, telles que l’ajout, la recherche ou la consultation de contenus, soient faciles d’accès même pour des utilisateurs non techniques. La simplicité, combinée à l’efficacité, a guidé toutes nos décisions de conception.

D’un point de vue technique, nous avons opté pour des technologies modernes et éprouvées. Le choix de React pour le frontend a permis de créer une interface réactive et dynamique, tandis que l’utilisation de Vite a grandement amélioré les temps de développement grâce à son environnement rapide. Par ailleurs, TailwindCSS a joué un rôle clé dans la mise en place d’un design cohérent et attrayant. En ce qui concerne l’API OpenAI, elle a été intégrée de manière à fournir des recommandations intelligentes en temps réel, tout en respectant les contraintes liées à la performance et aux quotas de requêtes.

Cependant, ce projet ne s’est pas déroulé sans défis. L’intégration de l’API OpenAI s’est révélée complexe à plusieurs reprises, notamment en raison des limitations de requêtes qui empêchaient parfois l’obtention de réponses. Cette difficulté nous a conduits à développer une stratégie d’adaptation, incluant des mécanismes de gestion des erreurs et des recommandations par défaut. En outre, l’absence d’un backend dédié a nécessité une approche différente, où toute la logique et les données sont gérées directement côté client.

Ce rapport documente de manière détaillée les étapes de la conception, du développement et des tests de l’application « **Bibliothèque Intelligente** ». Nous y décrivons les objectifs initiaux, les choix technologiques, les fonctionnalités principales, ainsi que les solutions apportées aux divers défis rencontrés. Nous espérons que ce travail mettra en lumière l’effort collaboratif fourni par notre équipe et les compétences acquises au cours de ce projet.

L’aventure de ce projet nous a permis d’explorer des domaines variés, allant de la gestion de données numériques à l’intelligence artificielle en passant par le design d’interface utilisateur. C’est également un point de départ pour de futures améliorations, avec des perspectives prometteuses telles que l’introduction de fonctionnalités avancées comme la synchronisation entre appareils ou l’élargissement des catégories de contenus gérées.

En somme, le projet « **Bibliothèque Intelligente** » illustre notre volonté de créer un outil pratique, performant et à la pointe de la technologie, tout en mettant l'utilisateur au cœur de l'expérience.

Notre groupe composé de OUAJJOU Ayoub, Kpinso-folly Joël, a saisi l'opportunité d'exploiter l'intérêt commun pour soumettre l'ébauche d'un projet personnel innovant à notre professeur de l'année de L1 M.DELBOT François.

Ce rapport présente une vue d'ensemble du projet, ses objectifs, ses fonctionnalités clés, ainsi que les technologies proposées pour son développement.

2 Bilan

2.1 Analyse des contraintes

Au cours de l'élaboration du projet, plusieurs contraintes techniques et fonctionnelles ont été identifiées. L'intégration de l'API OpenAI représentait un défi majeur, notamment en termes de gestion des erreurs et des limites de requêtes imposées par le service. Par ailleurs, l'absence d'un backend dédié a nécessité une optimisation côté client pour assurer une fluidité d'utilisation. En plus des contraintes techniques, nous avons dû prendre en compte les attentes des utilisateurs, tels que la simplicité d'utilisation, la rapidité des résultats, et la pertinence des recommandations.

2.2 Intégration des recommandations par défaut

Face aux limitations constatées dans l'utilisation de l'API OpenAI (par exemple, des messages comme « Vous ne pouvez plus faire de requête »), nous avons décidé d'implémenter un système de recommandations par défaut. Ce mécanisme permet d'assurer une continuité du service en proposant des suggestions prédéfinies lorsque l'IA n'est pas disponible. Ces recommandations, choisies avec soin, garantissent une expérience utilisateur cohérente.

2.3 Personnalisation

L'un des objectifs majeurs du projet était de personnaliser l'expérience utilisateur en fonction de son historique. En se basant sur les interactions précédentes (livres lus, musiques écoutées, vidéos visionnées), l'application génère des recommandations spécifiques. Cette approche vise à maximiser la pertinence des suggestions tout en offrant une valeur ajoutée à l'utilisateur.

2.4 Tests et ajustements

Les phases de tests ont permis de valider le bon fonctionnement des fonctionnalités clés, notamment les recommandations générées par OpenAI et celles proposées par défaut. Ces tests incluaient des vérifications des performances, de la pertinence des résultats, et de la compatibilité de l'application sur différents navigateurs et appareils. Les ajustements réalisés après chaque itération ont contribué à améliorer l'expérience globale.

3 Implémentation

3.1 Mise en place de l'environnement de développement

Le projet a été développé dans un environnement moderne et flexible. Nous avons utilisé Node.js et React pour bénéficier de leur large écosystème et de leur efficacité. Vite a été adopté comme outil de bundling pour accélérer les temps de compilation, tandis que TailwindCSS a été employé pour concevoir une interface épurée et réactive.

La configuration initiale incluait également la gestion des variables d'environnement, telles que la clé API pour OpenAI, afin de garantir la sécurité et la modularité du projet.

3.2 Interface utilisateur

L'interface utilisateur a été pensée pour être claire et intuitive. Les utilisateurs peuvent naviguer facilement entre les différentes sections, ajouter ou rechercher des documents, et consulter leurs recommandations. L'esthétique moderne et minimaliste, combinée à des icônes interactives, assure une navigation fluide tout en mettant l'accent sur l'expérience utilisateur.

3.3 Intégration de l'API OpenAI pour les recommandations

L'intégration de l'API OpenAI a constitué une étape cruciale du projet. Les données de l'utilisateur, telles que les titres précédemment consultés, sont envoyées sous un format structuré pour générer des recommandations adaptées. L'algorithme d'OpenAI a été configuré pour fournir des réponses en JSON, permettant une exploitation directe dans l'application. Des mécanismes de gestion des erreurs ont également été mis en place pour assurer la robustesse du système.

3.4 Gestion des recommandations par défaut

Pour pallier les éventuelles défaillances de l'API OpenAI, nous avons conçu un système de recommandations par défaut. Ces suggestions sont basées sur des choix génériques mais pertinents, tels que des classiques ou des œuvres populaires, afin de maintenir une continuité de service. Ce système garantit que l'utilisateur reçoit toujours des recommandations, même en cas de problème technique.

4 Difficultés rencontrées

Le projet a rencontré plusieurs défis majeurs. Tout d'abord, l'intégration de l'API OpenAI s'est révélée complexe, notamment à cause des limitations de requêtes et des erreurs fréquentes comme « Vous ne pouvez plus faire de requête ». Cela a nécessité la mise en place d'un système de fallback robuste pour assurer une expérience utilisateur continue.

Ensuite, l'absence d'un backend dédié a imposé de gérer la logique et les données côté client, augmentant ainsi la charge du frontend. Enfin, l'optimisation des performances a demandé des ajustements supplémentaires pour garantir une fluidité sur divers appareils.

5 Bilan

Le projet « **Bibliothèque Intelligente** » représente une avancée significative dans notre compréhension et notre maîtrise des technologies modernes. Malgré les défis rencontrés, nous avons réussi à développer une application fonctionnelle qui répond à ses objectifs principaux : permettre aux utilisateurs de gérer leurs contenus numériques et leur offrir des recommandations personnalisées basées sur leurs préférences. L'intégration de l'API OpenAI a été une étape marquante, bien que ponctuée de limitations techniques, notamment des restrictions de requêtes. Nous avons surmonté ces obstacles en implémentant des recommandations par défaut, assurant ainsi une continuité du service. Ce projet nous a permis de consolider nos compétences en développement frontend, en gestion d'erreurs et en design d'expérience utilisateur. En somme, il constitue une base solide pour de futures améliorations et démontre notre capacité à mener à bien des projets ambitieux.

5.1 Conclusion

Le projet « Bibliothèque Intelligente » a été une étape majeure dans notre apprentissage du développement d'applications web modernes. Il a permis de combiner des technologies avancées, comme l'API OpenAI et React, avec une approche centrée sur l'utilisateur. Malgré les défis rencontrés, tels que les limitations imposées par l'API ou l'absence d'un backend dédié, nous avons su proposer des solutions adaptées pour garantir une expérience fluide et continue. L'application remplit ses objectifs principaux : elle permet une gestion efficace des contenus numériques tout en offrant des recommandations pertinentes, même en cas d'indisponibilité de l'IA. Ce projet témoigne de notre capacité à concevoir et à développer des solutions innovantes tout en répondant aux attentes des utilisateurs.

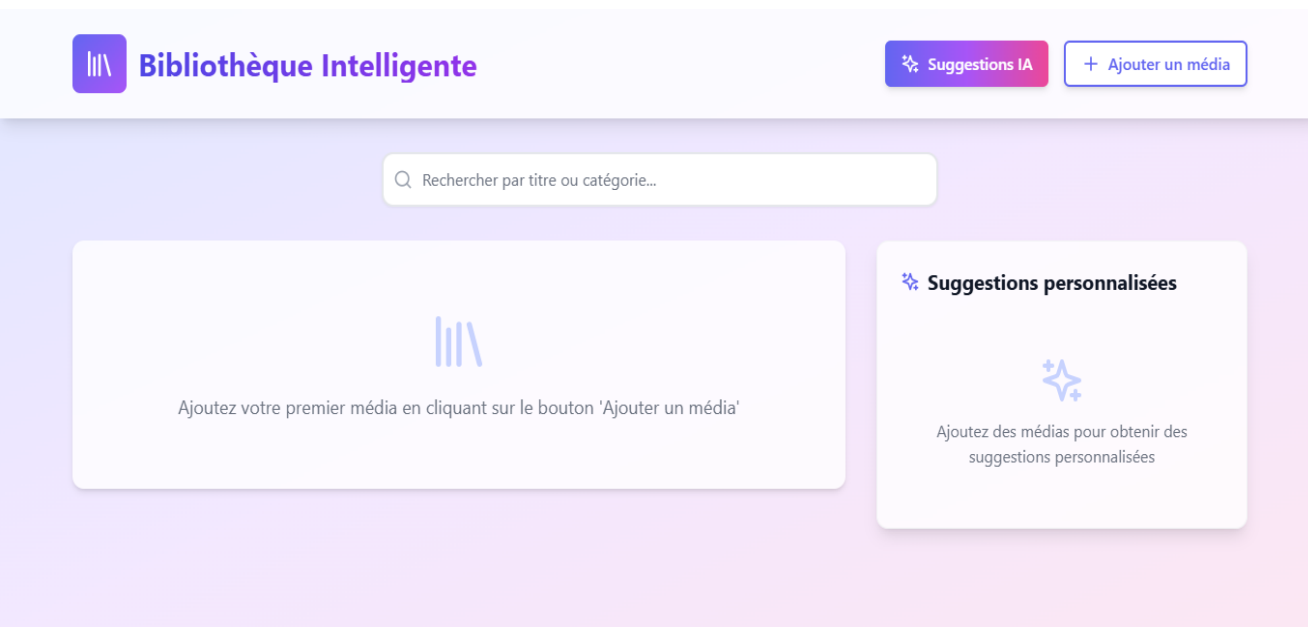
5.2 Perspectives

Pour aller au-delà de cette première version, plusieurs pistes d'amélioration sont envisagées. Tout d'abord, le développement d'un moteur de recherche avancé pourrait renforcer la navigation et l'accessibilité des contenus pour les utilisateurs. Ensuite, la synchronisation entre plusieurs appareils serait une fonctionnalité essentielle pour offrir une continuité d'utilisation, quelle que soit la plateforme. Par ailleurs, l'intégration de nouvelles catégories de contenus, telles que des podcasts ou des fichiers professionnels, élargirait le champ d'application de l'outil. Enfin, la mise en place d'un backend robuste pourrait permettre une gestion centralisée des données, ouvrant la voie à des fonctionnalités plus complexes comme des analyses statistiques sur les préférences utilisateur. Ces perspectives offrent un potentiel de croissance significatif pour faire de « Bibliothèque Intelligente » un outil encore plus puissant et polyvalent.

6 Captures d'écran

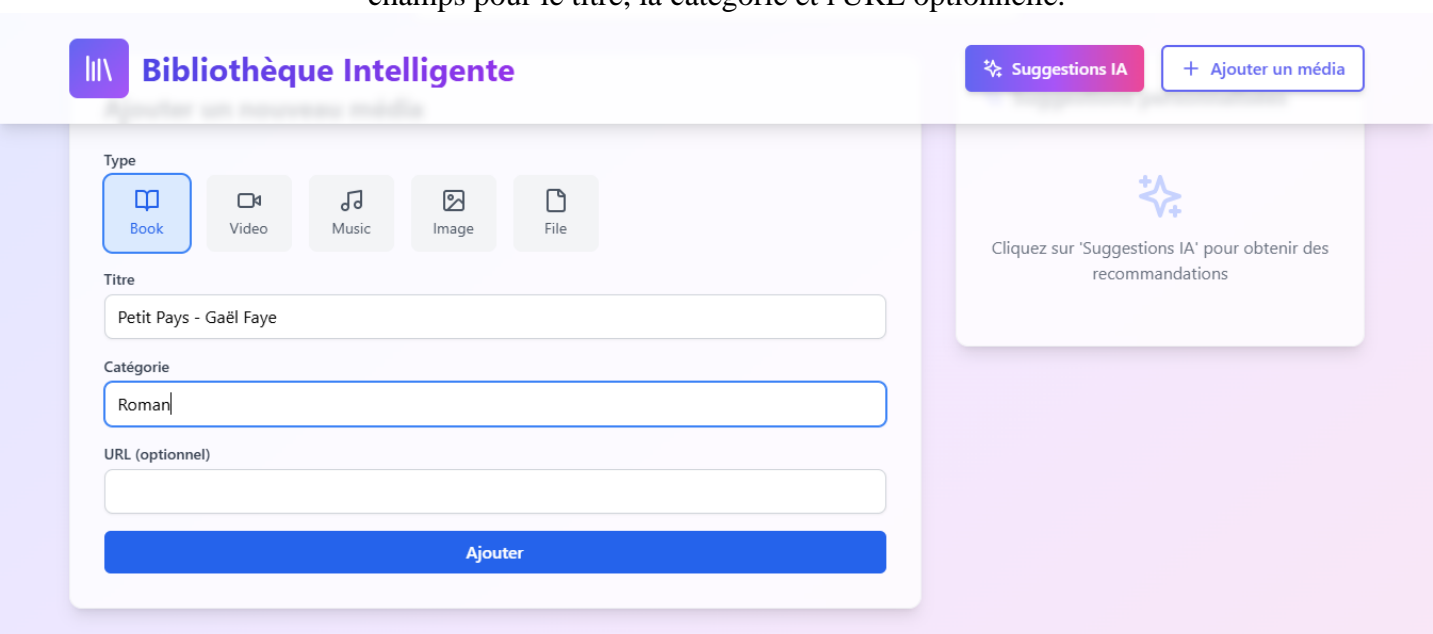
Capture d'écran 1 : Interface principale de la Bibliothèque Intelligente

Cette capture montre l'interface d'accueil où l'utilisateur peut commencer à gérer sa bibliothèque numérique, effectuer des recherches et accéder aux recommandations personnalisées.



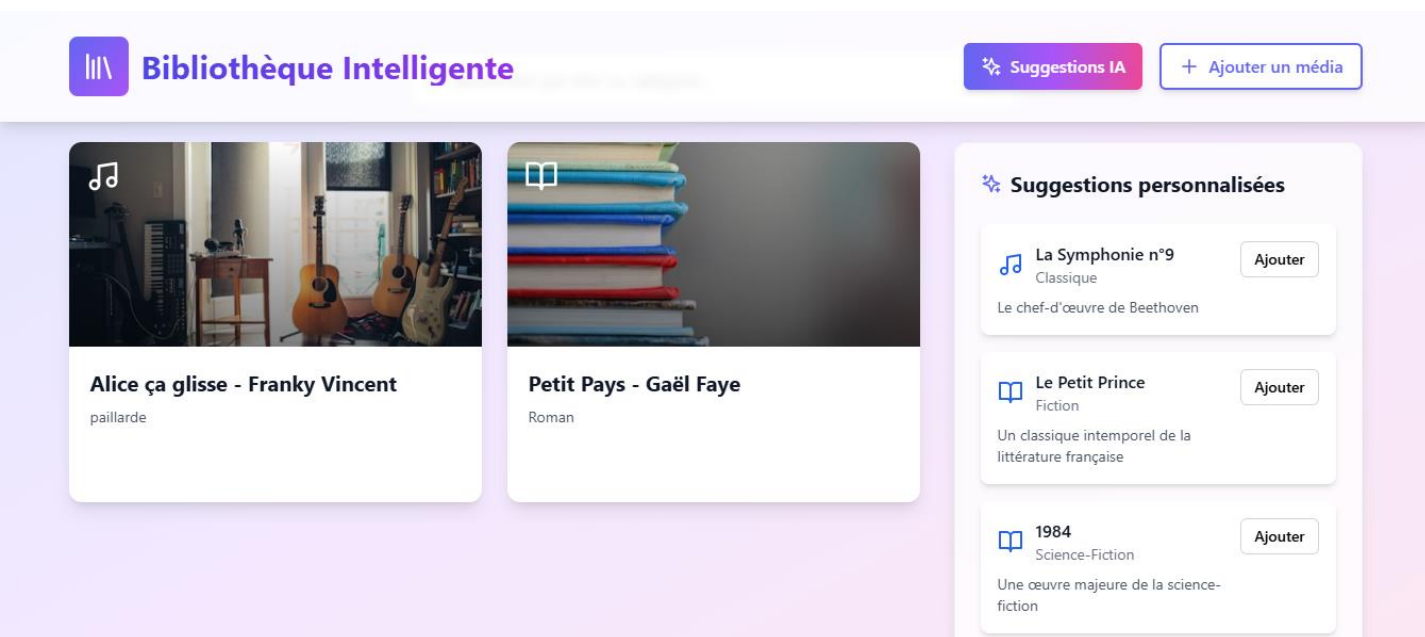
Capture d'écran 2 : Formulaire d'ajout d'un média

Cette capture illustre le processus d'ajout d'un nouveau média (livre, vidéo, musique, image ou fichier) avec les champs pour le titre, la catégorie et l'URL optionnelle.



Capture d'écran 3 : Aperçu de la bibliothèque avec suggestions IA

Ici, on voit les médias ajoutés à la bibliothèque personnelle ainsi que les suggestions personnalisées générées par l'intelligence artificielle.



Capture d'écran 4 : Fonction de recherche dans la bibliothèque

Cette capture met en évidence la fonctionnalité de recherche permettant à l'utilisateur de trouver rapidement des contenus spécifiques dans sa bibliothèque.

