

ÉVOLUTION **D'UNE** **APPLICATION**

SAE S6.A.01

AUDIT RÉALISÉ POUR LE GROUPE
MIAM-O-PEDIA



Miam-O-Pedia

ARUMUGADAS RAVICHANDRAN
BUT INFORMATIQUE

Sommaire

Présentation du projet audité	3
1. Le projet	3
a. Objectif du projet	3
b. Technologies Utilisées	3
c. Fonctionnalités Principales	3
2. Public Cible et Contexte d'Utilisation	4
Analyse Technique du Projet Miam-O-Pedia	5
1. Qualité du Code	5
2. Performance	6
3. Sécurité	6
Évaluation Fonctionnelle du Projet Miam-O-Pedia	8
1. Objectifs du Projet	8
2. Facilité d'utilisation et Ergonomie	8
3. Bugs et Dysfonctionnements	8
J'ai ajouté cet article en favoris.	10
Documentation et Maintenabilité du Projet Miam-O-Pedia	12
1. Commentaire du Code	12
2. Documentation Fournie	12
3. Évolution du projet : vers une application plus complète et polyvalente	12
Recommandations et conclusions	15
1. Points forts du projet	15
2. Points faibles	15
3. Axes d'amélioration	15

Présentation du projet audité

1. Le projet

a. Objectif du projet

Miam-O-Pedia est une application web conçue pour faciliter la recherche et la gestion d'articles alimentaires. Son principal objectif est d'offrir aux utilisateurs une plateforme intuitive où ils peuvent explorer des recettes, sauvegarder leurs plats favoris et recevoir des recommandations personnalisées en fonction de leurs goûts culinaires.

b. Technologies Utilisées

L'application repose sur un backend en Flask, un framework léger et modulaire, et une base de données MongoDB, choisie pour sa flexibilité dans la gestion de données non structurées. Le frontend est développé en HTML, CSS et JavaScript, permettant une interface fluide et interactive, notamment grâce à AJAX qui optimise l'expérience utilisateur en mettant à jour les contenus sans recharger la page.

c. Fonctionnalités Principales

Miam-O-Pedia propose plusieurs fonctionnalités clés pour faciliter la recherche et la gestion des articles alimentaires. Son moteur de recherche avancé permet aux utilisateurs de rechercher facilement un aliment ou un plat spécifique. Un système de filtres permet d'affiner la recherche en précisant si l'on souhaite un plat, un dessert ou une entrée. Les utilisateurs disposant d'un compte peuvent sauvegarder leurs recettes favorites et bénéficier d'un système de recommandations basé sur leurs préférences. De plus, ils ont la possibilité de créer des catégories personnalisées et d'y ajouter des articles alimentaires, comme par exemple regrouper leurs plats préférés sous une même thématique. Une page d'administration est également intégrée, offrant aux administrateurs la possibilité d'ajouter de nouveaux plats à la base de données.

2. Public Cible et Contexte d'Utilisation

Miam-O-Pedia s'adresse à un large public, allant des passionnés de cuisine aux personnes cherchant simplement de l'inspiration pour leurs repas quotidiens. Que l'on soit amateur, professionnel ou simplement curieux, l'application permet de découvrir facilement de nouvelles recettes et d'organiser ses préférences alimentaires. Elle convient aussi bien aux étudiants cherchant des idées simples et rapides qu'aux familles souhaitant planifier leurs repas, ou encore aux personnes ayant des restrictions alimentaires désirant trouver des plats adaptés. Accessible sur ordinateur et mobile, Miam-O-Pedia est conçue pour une utilisation fluide et intuitive, que ce soit à la maison, au travail ou même en faisant les courses.

Analyse Technique du Projet Miam-O-Pedia

Le projet **Miam-O-Pedia**, une application web de gestion d'armurerie, a été analysé sous plusieurs aspects techniques : la qualité du code, la performance, et la sécurité.

1. Qualité du Code

Le projet *Miam-O-Pedia* présente une architecture bien organisée, avec une séparation claire des responsabilités, facilitant ainsi la maintenance et l'évolutivité.

L'architecture suit une structure modulaire grâce à Flask, où chaque composant est isolé dans un dossier spécifique :

- Les **routes** sont définies dans `app/routes/`, garantissant une gestion propre des différentes API et endpoints.
- Les **modèles de données** sont centralisés dans `app/models/`, facilitant l'interaction avec MongoDB.
- Les **templates** HTML dynamiques sont stockés dans `app/templates/`, assurant une séparation entre la logique backend et l'affichage.
- Les **fichiers statiques** (CSS, JavaScript, images) sont bien organisés dans `app/static/`, permettant une gestion efficace du front-end.
- Un dossier `app/utils/` contient des fonctions utilitaires pour simplifier certaines tâches, améliorant ainsi la réutilisabilité du code.

De plus, la gestion de la base de données est centralisée dans `bdd_utils/`, avec des scripts dédiés pour l'initialisation (`bdd_creation.py`) et la création d'un administrateur (`bdd_create_admin.py`).

L'utilisation de **MongoDB** offre une approche flexible et évolutive, particulièrement adaptée aux données hétérogènes comme celles du proje

Le projet applique de bonnes pratiques en matière de **nommage des variables et des fichiers**, ce qui améliore sa lisibilité et réduit la dette technique.

Certaines fonctionnalités JS sont directement insérées dans les fichiers HTML. Bien que cela fonctionne, il serait préférable de les externaliser entièrement dans les fichiers **déjà existants** dans `app/static/js/`, pour garantir une meilleure maintenabilité et éviter la dispersion du code.

2. Performance

L'optimisation des performances a été au cœur du développement de Miam-O-Pedia, garantissant une utilisation fluide et réactive. L'application offre une expérience sans ralentissement, assurant une navigation rapide et intuitive. Même lors de l'exploration approfondie des contenus ou de l'application de multiples filtres, la réactivité reste optimale. L'équipe a veillé à ce que chaque interaction soit instantanée, permettant aux utilisateurs d'accéder rapidement aux informations recherchées sans aucune attente.

Le projet Miam-O-Pedia affiche une excellente performance grâce à une gestion optimisée des requêtes et une recherche rapide, même en cas de fautes de frappe, grâce à l'algorithme de Levenshtein. Celui-ci permet une tolérance aux erreurs en ajustant dynamiquement la distance de similarité, bien que sa complexité quadratique puisse devenir coûteuse sur des bases de données volumineuses. Pour pallier cela, plusieurs optimisations ont été mises en place, notamment l'indexation de la base de données, un pré-filtrage intelligent des résultats et l'utilisation d'AJAX pour une mise à jour dynamique sans rechargement de la page.

L'intégration de la recherche avec AJAX dans Miam-O-Pedia a permis une navigation fluide et ininterrompue. Cette approche optimise la réactivité tout en allégeant la charge du serveur. Grâce à ces choix technologiques et aux optimisations mises en place, l'application Miam-O-Pedia assure une expérience rapide et efficace, parfaitement adaptée aux besoins des utilisateurs.

3. Sécurité

Sécurisation des mots de passe

L'une des failles de sécurité majeures dans votre code concerne la gestion des mots de passe. Actuellement, les mots de passe sont stockés en clair dans la base de données. Cette pratique est extrêmement risquée, car en cas de fuite de données, les mots de passe des utilisateurs seraient facilement récupérables. Pour résoudre ce problème, il est essentiel d'utiliser des techniques de hachage et de salage des mots de passe avant de les stocker. Des bibliothèques comme `bcrypt` ou `werkzeug.security` permettent de sécuriser les mots de passe en générant des hachages robustes, rendant ainsi les données bien plus difficiles à exploiter.

Protection contre les injections et validation des données

Une autre vulnérabilité dans votre code réside dans le risque d'injections de type MongoDB ou d'autres formes d'injections dans les entrées utilisateur. Même si MongoDB est généralement plus résistant aux injections SQL classiques, des requêtes malformées peuvent toujours être exploitées. De plus, certaines données comme les emails ou les catégories ne sont pas suffisamment validées avant d'être insérées dans la base de données. Pour renforcer la sécurité, il est crucial d'ajouter une validation rigoureuse des entrées, en utilisant des bibliothèques spécialisées pour valider des données telles que les

emails ou les formats de texte. Une validation des données réduit également les risques d'injections malveillantes.

Limiter les tentatives de connexion (attaque par force brute)

L'une des meilleures pratiques pour sécuriser l'accès à une application est de limiter le nombre de tentatives de connexion infructueuses. En l'absence d'une telle protection, un attaquant pourrait essayer plusieurs mots de passe jusqu'à trouver le bon, en utilisant une méthode de force brute. Pour contrer cela, il est recommandé d'implémenter un mécanisme de limitation des tentatives de connexion, ce qui permettrait de bloquer temporairement un compte après plusieurs tentatives échouées. Cette protection peut être implémentée à l'aide de bibliothèques comme **Flask-Limiter**, qui permet de restreindre le nombre de requêtes par utilisateur sur une période donnée.

Évaluation Fonctionnelle du Projet Miam-O-Pedia

1. Objectifs du Projet

Miam-O-Pedia répond efficacement à son objectif principal : faciliter la recherche d'informations sur les aliments en fournissant des résultats pertinents, même en cas d'erreur de saisie. Grâce à l'algorithme de Levenshtein, l'application tolère les fautes de frappe, ce qui améliore l'expérience utilisateur. De plus, l'organisation modulaire du projet et l'utilisation d'une base de données MongoDB permettent une gestion fluide et évolutive des informations.

2. Facilité d'utilisation et Ergonomie

L'interface utilisateur a été conçue pour être simple et intuitive. La navigation est fluide, avec un formulaire de recherche bien intégré et un affichage dynamique des résultats. L'utilisation d'AJAX permet de charger les informations sans nécessiter un rechargement complet de la page, garantissant ainsi une expérience utilisateur agréable et réactive.

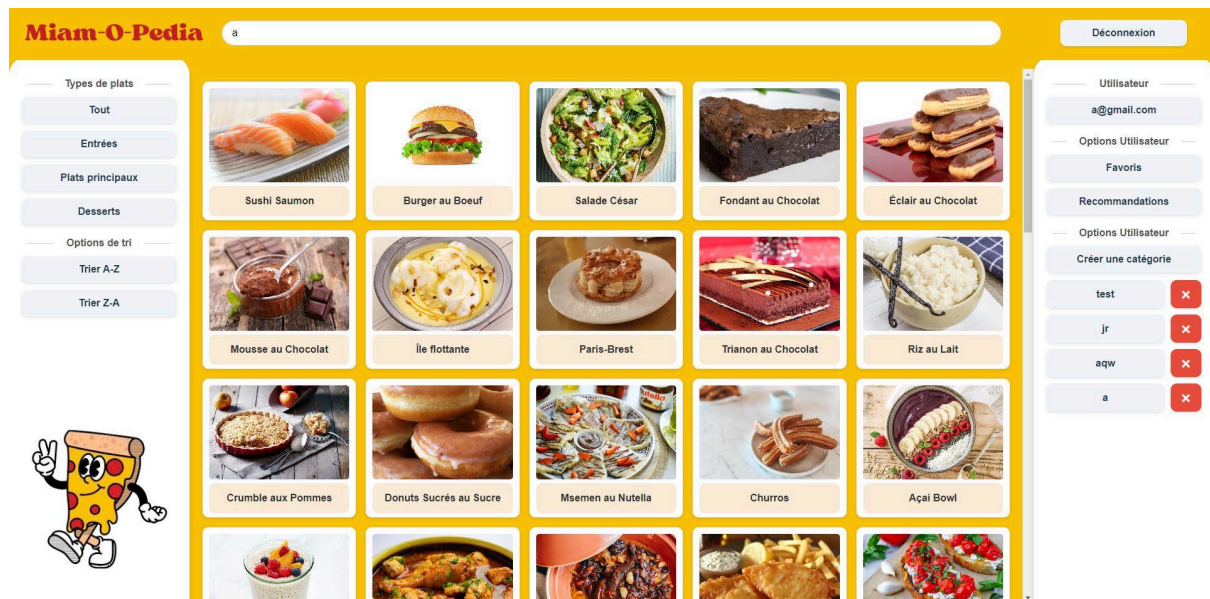
Cependant, un problème majeur concerne l'adaptabilité du site aux différents formats d'écran. Actuellement, Miam-O-Pedia n'est pas responsive sur les appareils mobiles, ce qui limite considérablement son accessibilité. Sur smartphone, l'interface ne s'adapte pas correctement, rendant la navigation difficile, voire inutilisable dans certains cas. Étant donné que de nombreux utilisateurs accèdent aux sites web via leur téléphone, cette absence de responsivité est un frein important à l'adoption du service. Pour améliorer cela, il serait pertinent d'optimiser le design en intégrant un système de mise en page flexible (comme Bootstrap ou CSS Grid) et en effectuant des tests sur plusieurs tailles d'écran.

3. Bugs et Dysfonctionnements

Cas n°1

Hormis le problème de responsivité, il y a aussi un autre problème au niveau du moteur de recherche, voici un exemple :

Par exemple, je tape la lettre 'a', et cela ne m'affiche pas les articles qui correspondent à la lettre 'a'.

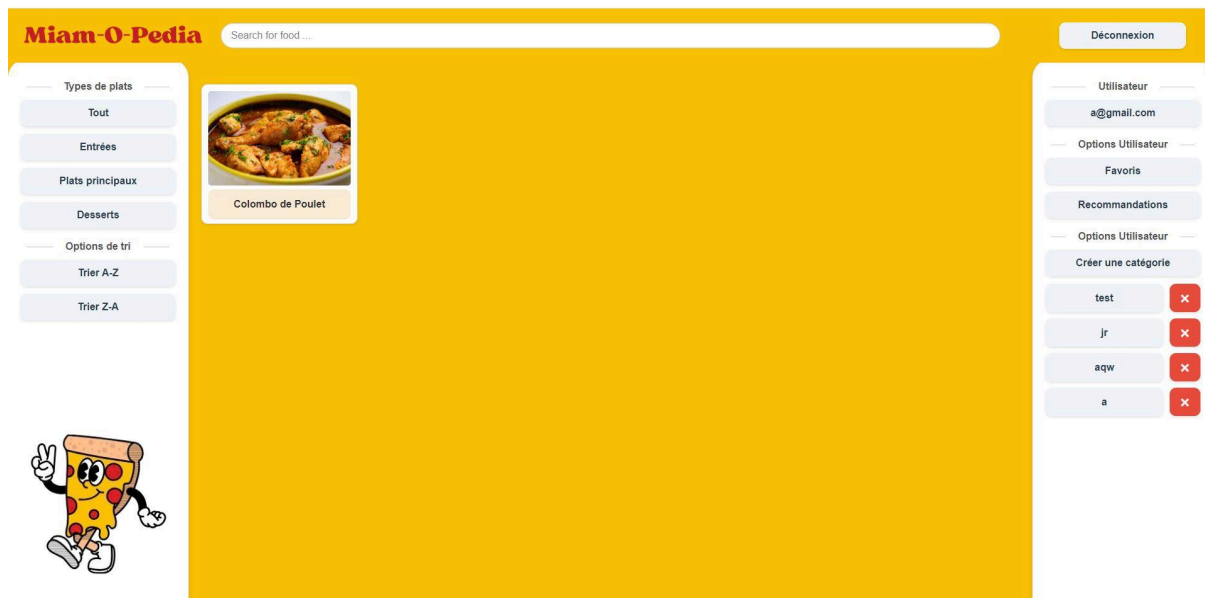


Maintenant, si j'écris un mot plus précisément, cela ne me met pas le produit en référence. Donc, il y a un problème au niveau de la barre de recherche, il ne fonctionne pas très bien. Dans certains cas, il fonctionne bien, dans d'autres non.

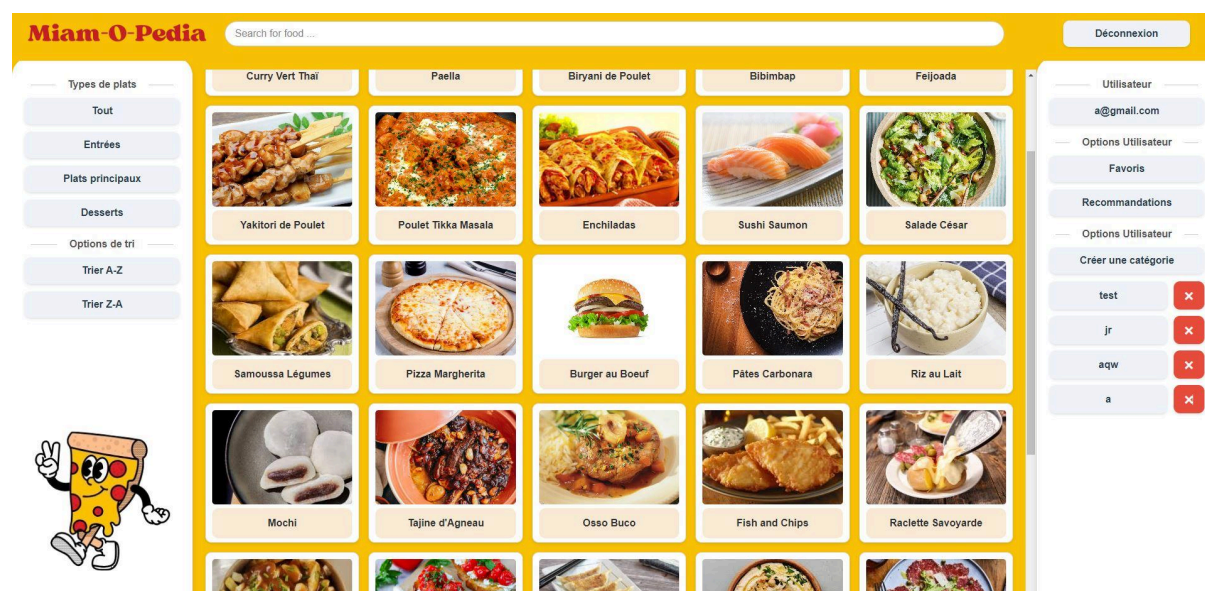


Cas n°2

Il y a un autre problème qui concerne la partie recommandation. Voici un exemple :
J'ai ajouté cet article en favoris.

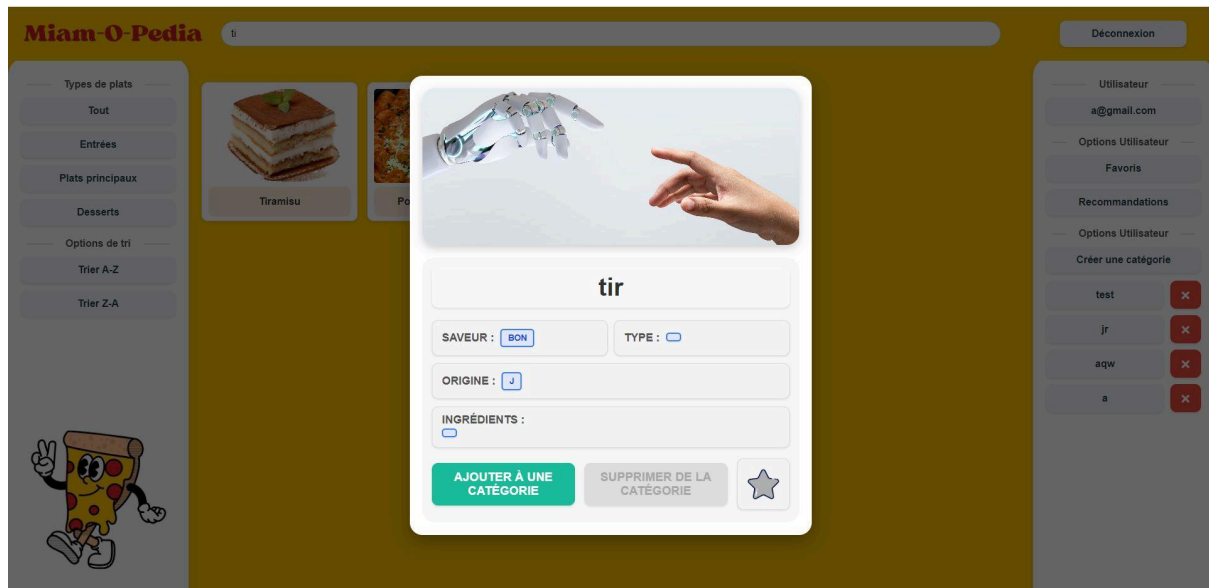


La rubrique "Recommandation" est censée afficher les articles alimentaires correspondant à mes favoris.



Le souci est que la rubrique "Recommandation" me propose aussi des articles qui ne sont pas en lien avec mes favoris. Par exemple, j'ai enregistré un aliment spécifique en favori, mais le système me recommande des mochi, alors que même les ingrédients listés dans ces mochi ne correspondent pas à ceux que j'ai sélectionnés. Cela semble être un petit problème dans le code, probablement au niveau du filtre ou de l'algorithme de correspondance des recommandations.

Cas n°3



Maintenant, il y a aussi un autre petit détail qui pourrait être amélioré.

Si, par exemple, un administrateur décide d'ajouter un nouvel article sans spécifier, par exemple, le type ou la saveur du produit, lorsque l'utilisateur veut consulter le produit, cela affiche un champ vide ou incomplet. Il serait préférable que, dans ce cas, rien ne soit affiché plutôt qu'un champ incomplet. À part cela, ce sont vraiment des détails.

Documentation et Maintenabilité du Projet

Miam-O-Pedia

1. *Commentaire du Code*

Le code est bien commenté, avec des commentaires clairs et précis qui expliquent les principales fonctions et blocs de code. Cela permet de faciliter la compréhension et la maintenance du code pour les développeurs qui devront intervenir dessus à l'avenir.

2. *Documentation Fournie*

Par ailleurs, l'équipe nous a fourni une documentation d'installation qui est claire et bien rédigée. Elle est conçue de manière à être compréhensible même pour une personne n'ayant pas de connaissances techniques. Cependant, il serait peut-être utile d'ajouter quelques précisions supplémentaires. Par exemple, un lien direct vers la page d'installation de MongoDB serait un ajout pratique pour faciliter l'expérience d'un utilisateur lambda.

D'autre part, il n'y a pas de documentation précise sur l'utilisation de l'application, comme le but de chaque page ou un aperçu du fonctionnement global. Il serait essentiel de modifier cela pour offrir une meilleure compréhension du logiciel aux utilisateurs. Une documentation utilisateur complète, qui explique clairement les fonctionnalités et le fonctionnement de l'application, serait un véritable atout pour l'adoption du logiciel, réduisant ainsi le besoin d'assistance technique.

3. *Évolution du projet : vers une application plus complète et polyvalente*

Le projet **Miam-O-Pedia** a été conçu avec une architecture flexible qui permet de l'adapter et de l'améliorer facilement au fil du temps. En effet, sa structure modulaire et bien pensée garantit qu'il est possible d'ajouter de nouvelles fonctionnalités sans perturber le fonctionnement global de l'application. Cela fait de ce projet une base solide, sur laquelle il est facile d'ajouter des fonctionnalités et d'apporter des améliorations en fonction des retours des utilisateurs et des évolutions du marché.

Évolution des fonctionnalités

L'une des premières pistes d'évolution pour **Miam-O-Pedia** pourrait être l'enrichissement de la base de données de recettes. Par exemple, il serait possible d'ajouter des détails plus précis, tels que des quantités exactes d'ingrédients en fonction du nombre de personnes (par exemple, "50g de poulet pour 4 personnes"). Cela améliorerait la précision des recettes proposées et offrirait une expérience plus pratique pour les utilisateurs, leur permettant de suivre les instructions de manière optimale selon leur situation.

Système de propositions utilisateur

Une autre amélioration envisageable serait d'intégrer un système de soumission de recettes par les utilisateurs eux-mêmes. Cela permettrait à chacun de proposer ses idées de recettes, plats ou entrées, enrichissant ainsi la base de données du site. Pour garantir la qualité des contenus partagés, un processus de modération pourrait être mis en place. Les propositions soumises par les utilisateurs seraient validées par des modérateurs avant d'être publiées, permettant à la communauté de contribuer activement tout en maintenant une certaine rigueur dans les informations partagées.

Amélioration des recommandations

Pour rendre l'expérience encore plus personnalisée, une évolution du système de recommandations pourrait être envisagée. En utilisant des algorithmes avancés, l'application pourrait analyser les habitudes et préférences des utilisateurs pour proposer des recettes encore plus adaptées à leurs goûts. Cela pourrait inclure des recommandations basées sur les recherches précédentes, les aliments fréquemment recherchés, ou même des tendances populaires au sein de la communauté.

Ajout de fonctionnalités sociales

Dans une perspective d'évolution, l'intégration d'aspects sociaux pourrait aussi enrichir l'expérience de l'utilisateur. Par exemple, il serait possible d'ajouter un système de partage, où les utilisateurs peuvent publier leurs propres réalisations, commenter les recettes des autres ou encore participer à des défis culinaires. Ce système renforcerait la dimension communautaire de l'application et permettrait à chaque utilisateur de s'impliquer davantage.

Gestion d'un espace collaboratif

En complément des fonctionnalités sociales, un espace collaboratif pourrait être ajouté pour permettre aux utilisateurs de se regrouper autour de différentes cuisines, régimes alimentaires ou défis. Cela offrirait un environnement d'échange et de partage, où les utilisateurs peuvent interagir, poser des questions, partager des astuces et créer des liens avec d'autres passionnés de cuisine.

Recommandations et conclusions

Après une analyse approfondie du projet Miam-O-Pedia, plusieurs points forts et axes d'amélioration ont été identifiés.

1. Points forts du projet

Le projet **Miam-O-Pedia** présente plusieurs points forts qui méritent d'être soulignés. Tout d'abord, l'architecture du projet est bien structurée, facilitant ainsi la maintenance et l'évolution. L'application repose sur une séparation claire des responsabilités entre le frontend et le backend, ce qui rend la gestion du code propre et modulaire. L'utilisation de Flask et MongoDB permet une gestion flexible et évolutive des données, tout en étant adapté à un volume croissant d'informations. Le système de recherche avancée, qui intègre l'algorithme de Levenshtein pour tolérer les fautes de frappe, améliore significativement l'expérience utilisateur. De plus, l'interface utilisateur est fluide et intuitive, avec une mise à jour dynamique via AJAX qui permet une expérience agréable sans rechargement complet de la page. En termes de sécurité, l'application fait usage de bonnes pratiques comme l'utilisation de **bindValue** pour prévenir les injections SQL, ainsi que la protection contre les attaques XSS, ce qui garantit une certaine sécurité de base.

2. Points faibles

Cependant, malgré ses nombreux atouts, le projet **Miam-O-Pedia** présente également des aspects qui nécessitent des améliorations. Un des problèmes majeurs est l'absence de **réactivité** du site, particulièrement sur les appareils mobiles. Actuellement, le design n'est pas responsive, ce qui rend l'application difficile à utiliser sur des smartphones. Cela limite l'accessibilité et l'adoption du service, notamment dans un contexte où une grande partie des utilisateurs accède à des sites via leurs téléphones. De plus, certains bugs fonctionnels affectent l'expérience utilisateur, notamment le moteur de recherche, qui ne retourne pas toujours les résultats attendus, même pour des recherches simples. Le système de recommandations présente également des incohérences, puisqu'il propose parfois des plats qui ne correspondent pas aux favoris de l'utilisateur. En termes de gestion des contenus, un autre détail concerne l'interface d'administration, qui laisse des champs vides lorsque certaines informations (comme le type ou la saveur) ne sont pas spécifiées pour un nouvel aliment. Cette gestion des données incomplètes peut nuire à la présentation et à la qualité des informations affichées aux utilisateurs.

3. Axes d'amélioration

Pour améliorer l'expérience utilisateur, plusieurs actions concrètes peuvent être envisagées. En premier lieu, l'**optimisation de la responsivité** est primordiale. Le projet devrait intégrer un système de mise en page flexible, comme **Bootstrap** ou **CSS Grid**, afin de garantir une utilisation fluide sur les différents types d'appareils. Il serait également pertinent de procéder

à des tests de compatibilité avec des écrans de différentes tailles afin d'identifier d'éventuelles anomalies. Concernant le moteur de recherche, il serait utile d'effectuer des tests pour en comprendre les causes profondes et corriger les erreurs de retour des résultats. Une solution pourrait être de revoir l'algorithme ou de réimplémenter un système de recherche plus robuste, en s'assurant que toutes les entrées, même partielles, renvoient les bons résultats. En ce qui concerne la section des **recommandations**, il est recommandé d'améliorer l'algorithme de correspondance, en affinant les critères de similarité et en s'assurant qu'il tienne compte des préférences exactes des utilisateurs, plutôt que de recommander des produits sans rapport. Enfin, pour éviter l'affichage de **champs incomplets** lors de l'ajout de nouveaux produits par les administrateurs, une validation stricte des informations avant l'enregistrement dans la base de données pourrait être mise en place.

Conclusion

Le projet **Miam-O-Pedia** représente une base solide avec un potentiel considérable pour devenir une plateforme de référence dans la gestion des articles alimentaires et des recettes. L'application offre une bonne structure technique, une interface agréable et une recherche efficace grâce à des algorithmes intelligents. Toutefois, pour garantir une adoption plus large et améliorer l'expérience utilisateur, il est essentiel de corriger certains bugs fonctionnels, d'optimiser la responsivité pour les appareils mobiles et de renforcer l'algorithme de recommandations. En appliquant ces améliorations, **Miam-O-Pedia** pourra offrir une plateforme encore plus performante et user-friendly, attirant ainsi un public plus large et fidèle.