

# **SAE 6.01 : Audit 1**

## **Groupe C : Destina**

# Sommaire

I. Présentation du projet audité

II. Analyse technique

III. Évaluation fonctionnelle

IV. Documentation et maintenabilité

V. Recommandations et conclusions

## **I. Présentation du projet audité**

### **Objectif du projet**

Le projet Destina a pour objectif de proposer aux utilisateurs des suggestions de destinations en fonction des mots qu'ils ont saisi dans la barre de recherche. Grâce à l'algorithme, l'application génère en temps réel des recommandations pertinentes, facilitant ainsi l'exploration et la découverte de nouvelles destinations.

### **Technologies utilisées**

Ce projet utilise différentes technologies telles que Deno pour l'exécution du serveur, mais également Typescript, Node.js et npm ou encore Visual Studio Code.

### **Fonctionnalités principales**

L'application propose une recherche dynamique permettant d'obtenir instantanément des suggestions de destinations en fonction des entrées utilisateur. Elle est dotée d'une interface web interactive et intuitive qui favorise une expérience utilisateur fluide.

### **Utilisateurs cibles**

Destina s'adresse principalement aux voyageurs et aux personnes souhaitant explorer de nouvelles destinations en toute simplicité. L'application peut également intéresser les professionnels du tourisme cherchant à enrichir leur offre de recommandations et donc pouvoir attirer le public.

### **Contexte d'utilisation**

Cette application peut être utilisée dans le cadre de l'organisation d'un voyage par un groupe de personnes. Elle peut également être utilisée par les offices de tourisme à la recherche de nouvelles recommandation à proposer à leur public.

## **II. Analyse technique**

### **Qualité du code**

Le projet est bien structuré, l'idée de séparer le front-end et le back-end est une idée intéressante et permet une séparation claire des responsabilités. De plus, l'utilisation de TypeScript apporte une rigueur supplémentaire en renforçant la typage des données et en réduisant les risques d'erreurs. Le code quant à lui, est bien aéré et bien indenté, ce qui facilite la lecture et la compréhension.

### **Performance**

Les données étant préchargées, le lancement du serveur est assez lent mais une fois sur l'application, le temps de réponse lorsque l'on tape quelque chose dans la barre de recherche est plutôt satisfaisant même si il arrive que cela mette un peu plus de temps par moment. Cela est peut être dû au fait les données.

### **Sécurité**

L'application ne présentant pas de système de connexion ou d'inscription, il n'y a pas de menaces directes pour les données personnelles de l'utilisateur. Cependant, Il est essentiel de vérifier la gestion des entrées utilisateur pour éviter les vulnérabilités telles que les injections SQL.

## **III. Évaluation Fonctionnelle**

### **Adéquation aux objectifs**

Le projet Destina répond parfaitement à son objectif principal en proposant un outil de recommandation basé sur les recherches saisies par l'utilisateur. Ce projet permet bien de chercher toutes sortes de destinations dans le monde entier et ce selon les préférences de l'utilisateur.

### **Expérience utilisateur**

L'application est relativement simple à utiliser, l'interface n'est pas surchargée et il est simple de naviguer entre les différents menus.

## **Bugs et dysfonctionnements**

Il n'y a pas de bugs apparent dans l'application ou de dysfonctionnement.

## **IV. Documentation et maintenabilité**

### **Commentaires dans le code**

Le projet contient des fichiers qui sont bien commentés. Certains fichiers du projet ont des commentaires clairs qui expliquent le fonctionnement des différentes parties du code. Ces commentaires facilitent la compréhension du code. Cependant, il y a également des fichiers qui manquent de commentaires.

### **Documentation**

Une documentation est fournie pour expliquer comment fonctionne l'application. Cette documentation est plutôt claire, cependant, elle reste incomplète. Il manque certaines informations concernant le projet tels que les technologies choisies par exemple ou encore les objectifs du projet.

### **Installation et configuration**

La documentation d'installation est fournie et décrit de manière claire les étapes nécessaires pour installer le projet. Elle inclut certains prérequis notamment Deno. Cependant, cette documentation est incomplète. Certaines informations concernant par exemple la base de données, ou encore le reste des technologies nécessaires ne sont pas mentionnées dans la documentation.

### **Facilité d'évolution**

L'application semble relativement simple à faire évoluer, cependant pour cela il est nécessaire de bien commenter le code et documenter le projet. Certaines améliorations peuvent être effectuées et de nouvelles fonctionnalités développées.

## **V. Recommandations et Conclusions**

### **Points forts**

Le projet Destina présente plusieurs atouts. Tout d'abord, l'usage de technologies moderne telles que TypeScript qui est très utile pour une meilleure gestion des erreurs ou encore Deno qui permet une gestion plus simple des dépendances.

### **Points faibles**

Malgré ces points positifs, certaines améliorations peuvent être apportées. Une revue du code permettrait d'optimiser les algorithmes de recherche et de suggestion pour que l'application réponde toujours rapidement même si la base contient un plus grand nombre de données ou qu'il y a plusieurs requête à la fois. L'application manque également de fonctionnalités.

### **Propositions d'amélioration**

Pour améliorer le projet, il serait intéressant d'ajouter des tests afin de détecter les éventuels dysfonctionnements. Optimiser les algorithmes pourrait être utile pour améliorer les performances. Développer un système de recommandations en adéquations avec les favoris de l'utilisateur pourrait être intéressant. Cela implique également d'instaurer un système de compte, d'inscription, etc.

# **SAE 6.01 : Audit 2**

## **Groupe M : Cookeasy**

# Sommaire

I. Présentation du projet audité

II. Analyse technique

III. Évaluation fonctionnelle

IV. Documentation et maintenabilité

V. Recommandations et conclusions



## **I. Présentation du projet audité**

### **Objectif du projet**

Le projet Cookeasy a pour objectif de proposer aux utilisateurs un outil permettant de rechercher des plats ainsi des suggestions en fonction des aliments, des cultures ou encore des régimes qu'ils ont saisi dans la barre de recherche.

### **Technologies utilisées**

Ce projet est séparé en deux entités : une partie client et une partie serveur. Chaque partie utilise des technologies différentes qui ont leur utilités. Par exemple, la partie client inclut l'utilisation de TypeScript, Next.js ou encore TailwindCSS et Zod. La partie serveur, quant à elle, utilise du Python et plus précisément le framework Fast API. On retrouve également l'utilisation de MongoDB ainsi que de Loguru (logs) et Asyncio (opérations asynchrones). Enfin on peut citer Locust pour les tests de performances et Doxygen qui permet de gérer la documentation.

### **Fonctionnalités principales**

Les utilisateurs ont la possibilité de se créer un compte ainsi que de se connecter à ce dernier afin d'accéder à leur compte. Ce compte leur permet ensuite de créer leur propres recettes en renseignant différents éléments tels que le nom du plat, les ingrédients utilisés et ils peuvent même y incorporer une photo.

### **Utilisateurs cibles**

Les utilisateurs se considérant comme des "cuisiniers du dimanche" à la recherche de recettes simples à réaliser peuvent être tentés par cette application qui pourrait leur permettre de trouver des recettes à propos de plats du monde entier, ce qui leur permettrait d'enrichir leur culture. Les personnes à la recherche d'un régime spécifique pourrait également trouver leur bonheur à l'aide de ce produit.

## **II. Analyse technique**

### **Qualité du code**

Le code est assez robuste, ce qui le rend parfois plutôt complexe à lire et à comprendre. Cependant, l'idée de diviser le code en une partie client et une autre partie serveur est plutôt bonne car cela nous permet notamment de séparer la partie front de la partie back. De plus, cela ajoute une séparation claire des responsabilités. Enfin, il pourrait être intéressant d'évoquer les redondances dans le code ou encore l'organisation un peu plus floue des répertoires ainsi que des fichiers dans le côté serveur.

### **Performance**

La durée d'affichage des données est au alentours de 45 ms mais cette dernière peut augmenter dans le cas où la base de données est plus conséquente. Le temps de réponse suite au rafraîchissement ou encore lorsque l'on entre quelque chose dans la barre de recherche reste tout de même plutôt satisfaisant.

### **Sécurité**

Petit bémol niveau sécurité des données, il n'y a aucune garantie que le mot de passe soit bien complètement protégé. Hormis cela, les données des utilisateurs sont bien protégées. L'utilisation d'un token et du hachage sont également des bonnes pratiques.

## **III. Évaluation Fonctionnelle**

### **Adéquation aux objectifs**

Le site remplit bien l'objectif principal de proposer à ces utilisateurs diverses recettes et de leur permettre de rechercher différents plats contenus dans la base de données.

## **Expérience utilisateur**

L'interface graphique est simpliste et intuitive, pas difficile à comprendre et la navigation entre les différentes pages se fait donc facilement. Les couleurs sont bien accordées et ne dérangent pas l'utilisateur. Cependant, certaines fonctionnalités manquent à l'appel comme par exemple la possibilité pour l'utilisateur de consulter et modifier son profil.

## **Bugs et dysfonctionnements**

Premièrement, certaines erreurs ont été relevées lors du chargement de certaines recettes, que nous soyons connecté ou non. De plus, il arrive parfois que lorsque l'on tape un plat dans la barre de recherche, il y'a d'autres plat qui apparaissent à la place. On pourrait aussi ajouter que lors de l'entrée d'une lettre simple, l'application ne nous renvoie rien, or on pouvait s'attendre à l'affichage de l'ensemble des plats commençant par cette lettre. Enfin le triage des plats semble ne pas fonctionner correctement et il est en plus de cela dérangeant pour l'utilisateur car ce dernier est placé au-dessus des premières recettes.

## **IV. Documentation et maintenabilité**

### **Commentaires dans le code**

Il y'a énormément de fichiers qui ne sont pas commenté et ceux qui le sont ne le sont que partiellement, ce qui rend le code difficile à comprendre et cela peut rendre la tâche difficile pour quelqu'un qui souhaite réutiliser ce code.

### **Documentation**

Idem pour la documentation, les fichiers ne sont pas assez documentés voir pas du tout. Cependant une documentation d'installation est fournie et elle est peut-être un peu difficile à comprendre mais simple à réaliser une fois comprise.

## **Installation et configuration**

L'installation de l'application était assez ardue aux premiers abords car il était assez compliqué de s'y retrouver entre les fichiers, les dossiers qui sont présents dans le projet. La première chose à relevé est l'absence de documentation pour les utilisateurs Windows notamment. Certains commentaires étaient assez flou mais on peut finalement réussir à s'en sortir. De plus, les installations au niveau du serveur ne sont pas toujours précises et complètes, ce qui peut porter à confusion.

## **Facilité d'évolution**

L'application a encore beaucoup de potentiel et énormément de choses peuvent être améliorés. La division des responsabilités entre la partie serveur et la partie client facilite notamment ces évolutions. Cependant le manque de documentation et de commentaires dans le code peuvent représentés un frein au bon avancement du projet.

## **V. Recommandations et Conclusions**

### **Points forts**

Le projet Cookeasy possède de nombreux atouts notamment concernant l'ergonomie de l'application, la séparation des responsabilités ou encore plus généralement le concept en lui même.

### **Points faibles**

Or il faut faire attention aux différents bugs et dysfonctionnements apparents qui peuvent être un frein à l'avènement de nouveaux utilisateurs. Il peut également être intéressant de citer la documentation du code qui est plus que nécessaire au développement futur du projet

### **Propositions d'amélioration**

Un petit point qui pourrait apporter quelque chose de plus à l'application, serait pourquoi pas de récolter, exposer les régimes de certaines stars par exemple ou encore de sportifs de haut niveau ou plus généralement de créateur de contenu qui peuvent partager leur routine alimentaire avec toute une communauté.