

Fiche d'investigation de fonctionnalité

Fonctionnalité : Filter les recettes dans l'interface utilisateur

Problématique : Pour se démarquer des autres plateformes de recettes, notre objectif est d'optimiser le moteur de recherche. Nous voulons garantir un accès rapide aux recettes répondant aux besoins de l'utilisateur parmi celles déjà disponibles. Les recettes seront filtrées en fonction de la saisie de l'utilisateur dans le champ principal (à partir de 3 caractères). Le filtre principal prend en compte le titre, les ingrédients et/ou la description. Les filtres de la recherche avancée seront automatiquement ajustés pour correspondre aux recettes affichées.

Option 1 – algorithme : Utilisation des boucles natives

Dans cette approche, les boucles natives while et for sont exploitées. Nous itérons à travers les recettes, et dès qu'un élément de la recette (titre, ingrédients, description) correspond à la recherche de l'utilisateur, la recette est ajoutée à la liste des résultats à afficher. Les filtres de la recherche avancée sont automatiquement actualisés pour correspondre aux recettes affichées.

Nombre de champs à remplir :

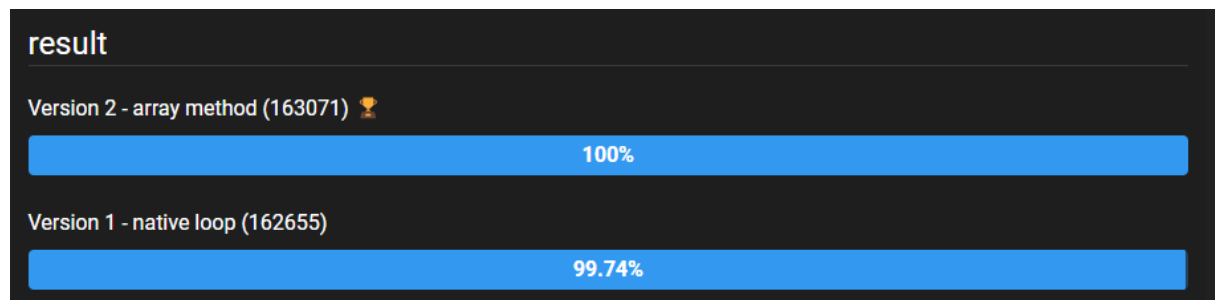
1 qui est le champ principal

Option 2 – algorithme : Programmation avec une méthode de l'objet Array

Dans cette approche, les méthodes filter et some sont mises en œuvre. Nous parcourons les recettes et les filtrons dès qu'un élément de la recette (titre, ingrédients, description) correspond à la recherche de l'utilisateur. Seules les recettes filtrées sont ensuite affichées. Les filtres de la recherche avancée sont automatiquement ajustés pour correspondre aux recettes affichées.

Nombre de champs à remplir :

1 qui est le champ principal



Conclusion :

1. Clarté du code :

- La version 2 utilise des méthodes d'objet Array comme « filter » et « some », ce qui rend le code plus concis et plus lisible.
- La version 1 utilise des boucles whiles/for, ce qui peut rendre le code légèrement plus verbeux et complexe à suivre pour certains développeurs.

2. Performance :

- En général, l'utilisation des méthodes d'objet Array comme « filter » est souvent plus efficace en termes de performances que l'utilisation de boucles while/for.

- b. Les méthodes d'objet Array sont optimisées et implémentées nativement dans la Javascript, ce qui les rend généralement plus rapides que les boucles.

3. **Maintenabilité :**

- a. La version 2 avec les méthodes d'objet Array est souvent plus facile à maintenir, car elle utilise des fonctions natives bien documentées et largement utilisées.
- b. La version 1 peut nécessiter une compréhension plus approfondie du fonctionnement des boucles whiles/for, ce qui peut rendre la maintenance plus difficile.

En considérant la clarté du code, les performances et la maintenabilité, la version 2 qui utilise les méthodes d'objet Array semble être un meilleur choix dans la plupart des cas.