

Fiche d'investigation de fonctionnalité

Fonctionnalité : Recherche de recettes Fonctionnalité #2

Problématique: L'objectif est de proposer un moteur de recherche rapide et fluide, capable de trier efficacement un arge nombre de recettes en fonction d'un mot-clé et/ou du choix de(s) dropdown(s) séléctionné(s). Cette recherche est un

élément crucial de l'expérience utilisateur pour permettre un accès facile et rapide aux informations souhaitées.

Option 1: Programmation Fonctionnelle

Dans cette option, la recherche est effectuée en utilisant des méthodes fonctionnelles disponibles dans JavaScript comme filter, map, reduce, et forEach. L'algorithme parcourt les recettes, puis vérifie si le mot-clé est présent dans les ingrédients, ustensiles, ou le nom de la recette.

Avantages

- Utilisation des fonctionnalités modernes de JavaScript
- Code plus concis et lisible

Inconvénients

- Moins performant pour de très grands volumes de données
- ⊖ Moins optimal en termes de consommation mémoire

Nombre de champs minimum analysés : 3 (nom de la recette, ingrédients, ustensiles)

Option 2: Boucles Natives

Dans cette option, la recherche est effectuée via des boucles natives (for, while, etc.), permettant un contrôle plus direct du flux d'exécution. Cela implique que chaque recette est vérifiée de manière séquentielle et les résultats sont ajoutés manuellement à une collection.

Avantages

Meilleure performance sur de grands ensembles de données

Inconvénients

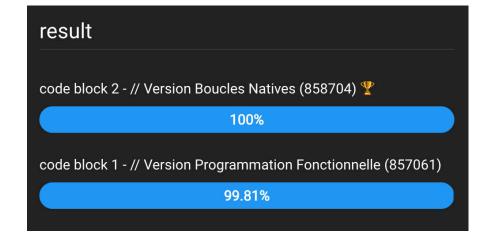
- ⊖ Code plus long et plus difficile à lire
- ⊖ Nécessite plus de gestion manuelle des états

1

Nombre de champs minimum analysés : 3 (nom de la recette, ingrédients, ustensiles)

Solution retenue:

Nous avons retenu l'option Programmation Natives. cf. captute ecran ci-dessous

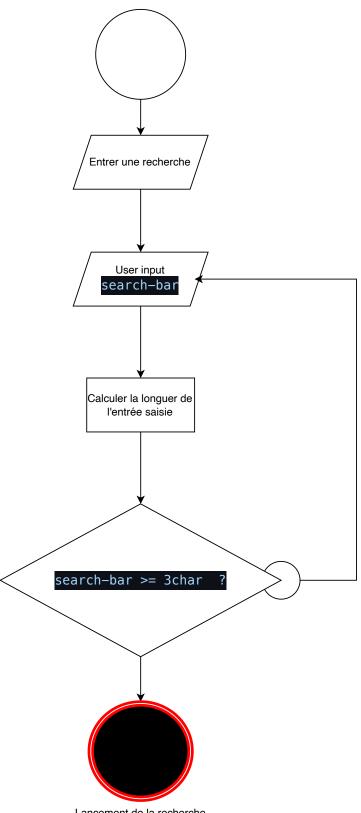




Annexes

cf. Draw.io file

Figure 1 - Alorigramme de la barre recherche



Lancement de la recherche



