Les petits plats

Fiche d’investigation fonctionnalité #2

Fonctionnalité : Système de recherche	Fonctionnalité #2
Problématique : Afin de rendre l’expérience de recherche la plus fluide possible, nous cherchons l’approche qui nous permettra d’avoir les meilleures performances. Ces deux approches seront testées dans un premier temps sur la barre de recherche.	

Solution 1 : Approche native (for, while…) Dans cette option, nous utiliserons les fonctions natives de JavaScript pour filtrer les recettes en fonction d'une chaîne de caractères saisie dans la barre de recherche. Une recette sera incluse dans les résultats si la chaîne de recherche est présente dans son titre, sa description ou l'un de ses ingrédients	
Avantages : - Contrôle total sur la logique de recherche - Meilleur compatibilité avec les anciennes version de javascript - Meilleur gestion de la mémoire	Inconvénients : - Moins lisible et plus verbeux - Moins performant sur de grands jeux de données

Solution 2 : Approche fonctionnelle (filter(), some(), every(), includes()) Dans cette option, nous utiliserons les méthodes JavaScript telles que **filter(), some() et every()** pour filtrer les recettes en fonction d'une chaîne de caractères saisie dans la barre de recherche. Une recette sera incluse dans les résultats si la chaîne de recherche est présente dans son titre, sa description ou l'un de ses ingrédients.	
Avantages : - Plus lisible et concis - Meilleure maintenabilité	Inconvénients : - L’ensemble du tableau doit être parcouru

Solution retenu :
En comparant les performances des deux approches sur différentes tailles de jeux de données (50, 100 et 150 150 recettes), il apparaît que **l’approche fonctionnelle** est plus rapide que l’approche native.