

fiche d'investigation de fonctionnalité

Fonctionnalité : Recherche de recettes + tags

Problématique : accéder rapidement à la fonctionnalité de recherche à une recette qui correspond au besoin de l'utilisateur en rapport avec les ingrédients, du nom des recettes ou leur description.

Architecture : Permet de trier parmi les 50 recettes disponibles via un champ de recherche et 3 filtres (ingrédients, ustensiles et appareils). S'il n'y a pas de recherche, le tableau que nous appellerons tableau initial, contenant les 50 recettes, sera envoyé aux filtres pour qu'ils puissent prendre le relais.

Détails : 1 champ de recherche, 3 sélecteur (filtres)

Algorithme 1 : Méthode d'Array filter/include

Dans cet algorithme, le code est composé de méthode Array (filter et include) qui permettent de segmenter les données des recettes simplement.

Avantages : écriture simple du code, utilisation pratique de méthodes de tableau avancées, meilleure compréhension du code

Inconvénients : connaître ce que les méthodes Array retournent si utilisées.

Algorithme 2 : Boucle native FOR + méthode d'Array include

Dans cet algorithme, le code est composé de méthode Array (include) et une boucle native FOR qui permettent de segmenter les données de chaque recette.

Avantages : JS Natif et connaissances de bases, simplicité et décomposer le fonctionnement des méthodes avancées de tableau.

Inconvénients : complexifie la compréhension du code et sa clarté. Complicé à maintenir.

Solution retenue : L'algorithme 1 est la solution retenue. Malgré une performance légèrement en dessous de l'algorithme Boucle FOR, il reste maintenable et reste simple niveau compréhension. Il est donc préférable de renoncer à une partie de la performance au profit de la maintenabilité.

