

Fiche d'investigation de fonctionnalité

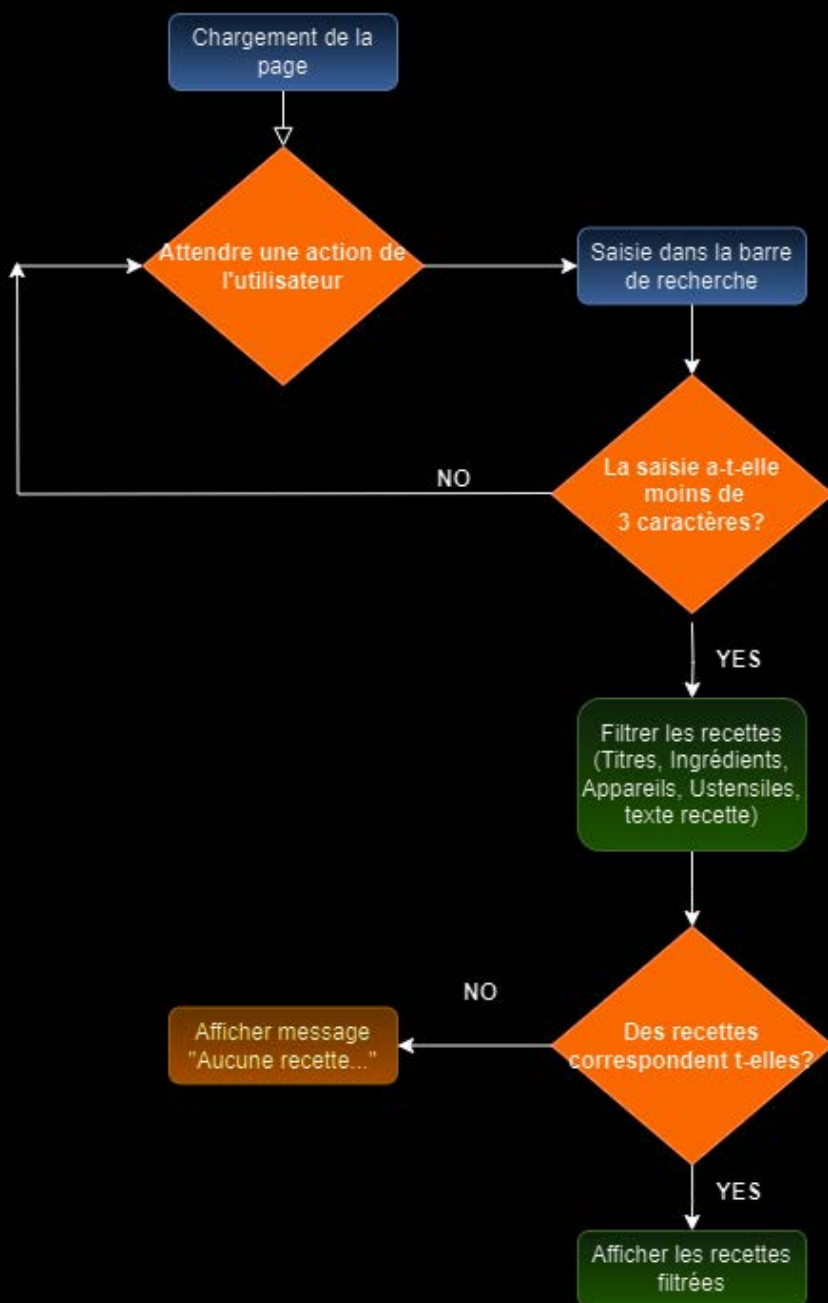
Fonctionnalité : Recherche principale	Fonctionnalité #1
Problématique : Nous cherchons à implémenter une fonction de recherche principale efficace et performante pour filtrer les recettes en fonction des entrées de l'utilisateur. L'objectif est d'optimiser la performance et la lisibilité du code.	

Option 1 : Implémentation avec boucle FOR (branche main) Cette option utilise des boucles `for` traditionnelles pour parcourir et filtrer les recettes.	
Avantages <ul style="list-style-type: none"> ⊕ Performance potentiellement meilleure pour de très grands ensembles de données ⊕ Contrôle précis sur l'itération ⊕ Possibilité d'interrompre la boucle prématurément si nécessaire 	Inconvénients <ul style="list-style-type: none"> ⊖ Code plus verbeux et potentiellement moins lisible ⊖ Plus susceptible aux erreurs de logique (par exemple, erreurs d'indexation) ⊖ Moins expressif en termes d'intention du code
Nombre de caractère minimum dans la barre de recherche : 3	

Option 2 : Implémentation avec méthodes d'array (filter) (branche feature/functional-search) Cette option utilise des méthodes d'array fonctionnelles comme `filter` pour parcourir et filtrer les recettes.	
Avantages <ul style="list-style-type: none"> ⊕ Code plus concis et plus lisible ⊕ Approche déclarative, exprimant clairement l'intention du code ⊕ Réduction des risques d'erreurs liées à l'indexation ⊕ Facilité de maintenance et de modification 	Inconvénients <ul style="list-style-type: none"> ⊖ Peut être légèrement moins performant pour de très grands ensembles de données ⊖ Potentiellement moins flexible pour des logiques de filtrage très complexes ⊖ Courbe d'apprentissage pour les développeurs moins familiers avec la programmation fonctionnelle
Nombre de caractère minimum dans la barre de recherche : 3	

Solution retenue : Option 2 (méthodes d'array)
Justification : <ol style="list-style-type: none"> 1. Code plus lisible et maintenable (Code plus concis et expressif, facilite la compréhension et maintenance à long terme). 2. Réduction des erreurs d'indexation 3. Performance adaptée à l'application 4. Alignement avec les pratiques JavaScript modernes 5. Facilité d'extension et de modification future


Annexes



Algorithme fonctionnalités barre de recherche principale

JSBEN.CH

BENCHMARKBROWSE

Benchmark Functionnalités search 

RUN TESTS

GENERATE PAGE URL

NEW BENCHMARK


Description


Setup block

(useful for function initialization. it will be run before every test, and is not part of the benchmark.)

Boilerplate block


(code will executed before every block and is part of the benchmark. use it for data initializing.)

code block 1 

code block 2 

+

result

code block 1 (57896) 

100%

code block 2 (56550)

97.68%

Résultat BenchMark Des 2 fonctionnalités de recherche de la barre principale source : JSBEN.CH

3

CODE BLOCK 1

```
function searchRecipesNativeFor(query, recipes) {
  query = query.toLowerCase().trim();
  let filteredRecipes = [];
  for (let i = 0; i < recipes.length; i++) {
    if (recipeMatchesSearch(recipes[i], query)) {
      filteredRecipes.push(recipes[i]);
    }
  }
  return filteredRecipes;
}

// Appel de la fonction pour le benchmark
runSearch(searchRecipesNativeFor);
```

CODE BLOCK 2

```
function searchRecipesFilter(query, recipes) {
  query = query.toLowerCase().trim();
  return recipes.filter(recipe => recipeMatchesSearch(recipe, query));
}

// Appel de la fonction pour le benchmark
runSearch(searchRecipesFilter);
```

CODE Boilerplate block

```
const recipes = [
  // Liste Tableaux de recettes ici
];

function recipeMatchesSearch(recipe, query) {
  return recipe.name.toLowerCase().includes(query) ||
    recipe.description.toLowerCase().includes(query) ||
    recipe.ingredients.some(ing => ing.ingredient.toLowerCase().includes(query));
}

// Fonction de test commune
function runSearch(searchFunction) {
  const queries = ["poulet", "citron", "four", "salade"];
  queries.forEach(query => {
    searchFunction(query, recipes);
  });
}
```

La fonction run Search permet d'exécuter les tests pour les deux versions de la fonction de recherche