|  |  |
| --- | --- |
| **Fonctionnalité #** | **Fonctionnalité #** |
| **Problématique :**  Filtrer les recettes dans l’interface utilisateur.  Accéder rapidement à une recette correspondant à un besoin de l’utilisateur dans les recettes déjà reçues. | |

|  |  |
| --- | --- |
| **Option 1 :** Mise en œuvre à l'aide de boucles imbriquées (For ou ForEach). | |
| **Avantage :** | **Inconvénients** |
| Boucles imbriquées « For » :   * Pas besoin de rechercher/itérer dans toutes les données si les premières conditions sont remplies.   Boucles imbriquées « ForEach » :   * Pas besoin de rechercher/itérer dans toutes les données si les premières conditions sont remplies. * Plus lisible que la boucle « For ». * Moins de code que la boucle « For ». | * Moins lisible. * Plus de code. (17 lignes). * Mais moins lisible que « Filter et some » * Mais plus de code que l'algorithme utilisant des fonctions d'ordre supérieur. |
| Le cas d’utilisation commence lorsque l’utilisateur entre au moins 3 caractères dans la barre de recherche principale. | |

|  |  |
| --- | --- |
| **Option 2 :** Implémentation à l'aide de fonctions d'ordre supérieur (Filter, include & some). | |
| **Avantage :** | **Inconvénients** |
| * Moins de code. (3 lignes). * Lisibilité. | * Nécessaire pour rechercher/itérer dans toutes les données (nom, description et ingrédients). |
| Le cas d’utilisation commence lorsque l’utilisateur entre au moins 3 caractères dans la barre de recherche principale. | |

|  |
| --- |
| **Solution retenue :**  Après avoir effectué des tests, en utilisant le site JSBEN.CH pour mesurer les performances des algorithmes, je recommande d’utiliser l’algorithme qui utilise les fonctions d’ordre supérieur « filter, include & some ». C’est mieux pour la vitesse ainsi que pour la lisibilité du code. (**Algorithme 2).** |

**Annexes**

|  |
| --- |
|  |

**Figure 1 – Algorithme 1**

|  |
| --- |
|  |

**Figure 2 – Algorithme 2**

**Un échantillon des résultats des 72 tests effectués sur les algorithmes :**

|  |  |  |
| --- | --- | --- |
|  | | |
| **Titre :** le mot cible à trouver est dans le titre… | | |
| JSBEN.CH : | | |
| Salade méditerranéenne fraiche au chèvre | Crème pâtisserie | Poulet coco réunionnais |
|  |  |  |
| Filter: 6361 op/s  forEach : 6303 op/s  for Loop : 6259 op/s | Filter: 5932 op/s  For Loop: 5914 op/s  forEach: 4618 op/s | For Loop: 6254 op/s  Filter : 6150 op/s  forEach : 5599 op/s |
| https://jsben.ch/GEFvF | <https://jsben.ch/2c1bb> | <https://jsben.ch/9E2Zo> |
|  | | |
|  | | |
| **Description :** le mot cible à trouver se trouve dans la description… | | |
| JSBEN.CH : | | |
| cocotte | étalez | réfrigérateur |
|  |  |  |
| forEach : 6293 op/s  For Loop : 6269 op/s  Filter : 6071 op/s | Filter : 5679 op/s  For Loop : 5550 op/s  forEach : 5525 op/s | Filter : 5939 op/s  forEach : 5910 op/s  For Loop: 4536 op/s |
| <https://jsben.ch/AyUG4> | <https://jsben.ch/00PSO> | <https://jsben.ch/1RUUf> |
|  | | |
| **Ingrédients :** le mot cible à trouver se trouve dans les ingrédients… | | |
| JSBEN.CH : | | |
| Pâte Brisée | Crème fraîche | Pâte Feuilletée |
|  |  |  |
| Filter : 6262 op/s  ForEach : 6097 op/s  For Loop : 5999 op/s | For Loop : 6491 op/s  Filter : 5184 op/s  ForEach : 5091 op/s | Filter : 6351 op/s  ForEach : 6347 op/s  For Loop : 6275 op/s |
| <https://jsben.ch/9ucL0> | <https://jsben.ch/DA3xm> | <https://jsben.ch/Jse3k> |

**Figure 3 – Un échantillon des résultats**

Les tests de performance ont été effectués 72 fois.

6 mots ont été choisis dans chaque champ : le titre (nom), la description et les ingrédients.

Les tests ont été exécutés 4 fois pour chaque mot ciblé.

L'algorithme qui a obtenu le score le plus élevé le plus souvent a été considéré comme le plus efficace.

*(Pour les résultats complets, veuillez consulter la feuille de calcul « résultats des tests »)*

|  |
| --- |
|  |

**Figure 4 - Le modèle complet (en utilisant l'algorithme 2)**