

# Projet



## Les petits plats

Développez un algorithme de recherche  
en JavaScript

# Contexte du projet

Je suis freelance et j'ai été missionnée par l'entreprise "Les petits plats" en temps que Développeur Front-end pour une mission de 3 mois.

Après avoir édité des livres de cuisine pendant plusieurs années, l'entreprise a décidé de se lancer dans un nouveau projet : **réaliser son propre site de recettes de cuisine..**

Je suis chargée de créer l'interface du site Web en implémentant la fonctionnalité de recherche et en prenant en compte la fluidité du moteur de recherche.

# Outils

- fichiers JSON de 50 recettes
- la description du cas d'utilisation :

<https://s3-eu-west-1.amazonaws.com/course.oc-static.com/projects/Front-End+V2/P6+Algorithms/Cas+d%E2%80%99utilisation+%2303+Filtrer+les+recettes+dans+l%E2%80%99interface+utilisateur.pdf>

- la maquette figma :

<https://www.figma.com/file/xqeE1ZKIHUWi2Efo8r73NK/UI-Design-Les-Petits-Plats-FR>

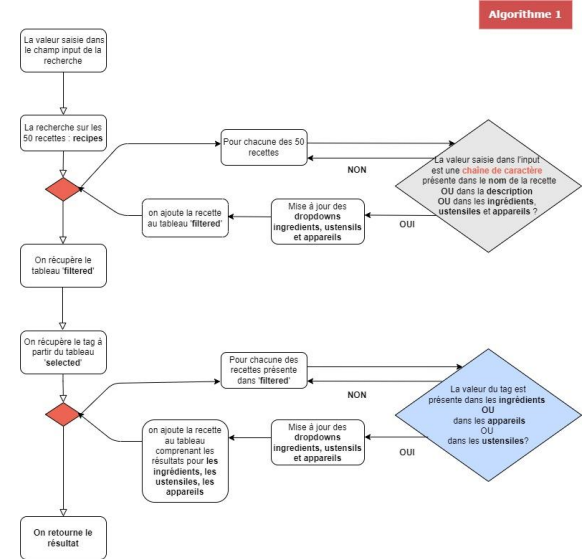
- la fiche d'investigation de fonctionnalité

# Décomposer la problématique

- Problématique du projet centrée essentiellement sur la recherche
- Réalisation de 2 versions du projet:
  - tester la performance
  - conserver l'algorithme le plus rapide et adapté au moteur de recherche
- Conception d'un schéma pour mieux comprendre la problématique

# Fiche d'investigation de fonctionnalité

- Décrire 2 types de recherches :
  - par chaînes de caractères
  - par termes
- Schématiser les algorithmes proposés
- Conserver la solution



# Implémentation du site Web

- Présenter l'interface Web:
  - sans bootstrap
  - utilisation du flexbox



# Dropdowns

- Recherche d'un élément qui filtre sur les données des recettes



- Filtre directement en cliquant sur l'élément
- Chaque sélection filtre les éléments dans chaque dropdown



# Code Dropdown

- construit sur la méthode Factory Pattern:
  - utilisation des classes
  - utilisation des extensions de classe
  - une approche plus solide pour la résolution de problème
  - code plus lisible

```
JS Filters.js  X
js > JS Filters.js > Filters > render
1  class Filters
2  {
3      constructor(type)
4      {
5          this.all = new Set();
6          this.filtered = new Set();
7          this.selected = new Set();
8          this.searchValue = '';
9          this.type = type;
10         this.renderDropdown();
11     }
12
13     //-- construit le HTML
14     build()
15     {
16         this.render();
17         this.listenForSelection();
18         this.listenForUnselect();
19     }
20 }
```



# La recherche

- La recherche d'un élément filtre sur les recettes mais aussi les dropdowns
- Si la saisie contient une erreur : message d'erreur

 Les petits plats

coco 

Rechercher un ingrédient ^

lait de coco	jus de citron	crème de coco
sucré	glacés	thon rouge (ou ...
concombre	tomates	carottes
citrons vert	poulet	coulis de tomates
oignon	poivron rouge	huile d'olives
pommes de terre	ail	pois cassé
beurre	crème fraîche	cumin
noix de muscade		

**Poisson Cru à la tahitienne** ⌚ 60min

**Poisson Rouge (ou blanc)** :  
grammes  
Céleriandre : 1  
Tomates : 2  
Carottes : 1  
Poisson : 1

Découper le thon en dés, mettre dans un plat et recouvrir de jus de citron vert (mieux vaut prendre un plat large et peu profond). Laisser mariner 1h30.

**Poulet coco réunionnais** ⌚ 80min

**Poulet** : 1  
**Lait de coco** : 400 ml  
**Couille de tomates** : 25 cl  
**Oignon** : 1  
**Poivron rouge** : 1  
**Huile d'olive** : 1

Découper le poulet en morceaux, les faire dorer dans une cocotte avec de l'huile d'olives. Salez et poivrez. Une fois doré, laissez mijoter 1h30.

 Les petits plats

coco 

**Aucune recette ne correspond à votre critère...**

Vous pouvez chercher d'autres mots-clés comme «**tarte aux pommes**», «**poisson**», ou découvrez de nouvelles recettes à l'aide de nos **filtres de recherche**.



# La recherche par chaîne de caractères

- utilisation de la méthode IndexOf
  - renvoie le premier indice pour lequel on trouve un

élément donné dans un tableau.

- `indexOf` compare élémentRecherchéaux éléments

contenus dans le tableau en utilisant une égalité stricte (la même

méthode utilisée par l'opérateur `===`).

- utilisation de conditions, Foreach et  
boucle For

```
recipeIsValid(recipe)
{
    //terme recherché dans le titre
    if(recipe.name.toLowerCase().indexOf(this.searchValue) >= 0)
    {
        return true;
    }

    //terme recherché dans ustensils
    recipe.ustensils.forEach(ustensil => {
        if(ustensil.toLowerCase().indexOf(this.searchValue) >= 0)
        {
            return true;
        }
    })

    //terme recherché dans appareil
    if(recipe.appliance.toLowerCase().indexOf(this.searchValue) >= 0)
    {
        return true;
    }

    //terme recherché dans description
    if(recipe.description.toLowerCase().indexOf(this.searchValue) >= 0)
    {
        return true;
    }

    // terme recherché dans ingrédients
    for(let i=0; i < recipe.ingredients.lenght; i++)
    {
        let ingredient = recipe.ingredients[i].ingredient;

        if(ingredient.ingredient.toLowerCase().indexOf(this.searchValue) >= 0)
        {
            return true;
        }
    }
}
```

# La recherche par terme

- utilisation de la méthode split()
  - transforme une chaîne de caractère en tableau
- foreach

```
//terme recherché par ingrédients
this.ingredients.forEach(item=>
{
    item.ingredient.split(' ').forEach(word =>
    {
        word = word.replace(/[\.,\/#!$%^&\*;\{:}=\-_`~()]/g, "")
        if(this.isWordValid(word))
        {
            this.terms.add(word);
        }
    })
})
}

isWordValid(word)
{
    const forbiddenWords = [
        "abord","afin","ainsi","allaient","allo","allô","allons","après","assez","attendu",
        "aucun","aucune","aujourd","aujourd'hui","auquel","aura","auront","aussi","autre","autres",
        "aux","auxquelles","auxquels","avaient","avais","avait","avant","avec","avoir","ayant",
        "bah","beaucoup","bien","bigre","boum","bravo","brrrr","ça","car","ceci","cela",
        "celle","celle-ci","celle-là","celles","celles-ci","celles-là","celui","celui-ci",
        "celui-là","cent","cependant","certain","certaine","certaines","certains","certes",
        "ces","cet","cette","ceux","ceux-ci","ceux-là","chacun","chaque","cher","chère",
        "chères","chers","chez","chiche","chut","cinq","cinquantaine","cinquante","cinquantième",
        "cinquième","clac","clic","combien","comme","comment","compris","concernant","contre",
        "couic","crac","dans","debout","dedans","dehors","delà","depuis","derrière","des"
```

# Performance testée

- Performance tester avec **console.time()**

- Plus le chiffre est petit, plus la

recherche est rapide

exemple :

-algo 1 : monTimer1: 0.285888671875 ms

-algo 2: monTimer2: 2.5341796875 ms

```
//-- Paramètre de la recherche par chaîne de caractères
search()
{
    console.time("monTimer1"); // timer début performance

    let items = [];
    list.filtered.forEach(recipe =>
    {
        if(this.recipeIsValid(recipe))
        {
            items.push(recipe)
        }
    })

    console.timeEnd("monTimer1"); // timer fin performance
    return items;
}
```

# Github

[https://github.com/Laurene45/LaureneCourde\\_7\\_13092021/tree/algo\\_1](https://github.com/Laurene45/LaureneCourde_7_13092021/tree/algo_1)

The screenshot shows the GitHub interface for the repository `Laurene45 / LaureneCourde_7_13092021`. The repository is public. The navigation bar includes links for `Code`, `Issues`, `Pull requests`, `Actions`, `Projects`, `Wiki`, `Security`, `Insights`, and `Settings`. The `algo_1` branch is selected, showing 3 branches and 0 tags. The `Go to file`, `Add file`, and `Code` buttons are visible. The `Switch branches/tags` dropdown is open, showing a list of branches: `main` (default), `algo_1` (selected), and `algo_2`. A `View all branches` link is also present. The commit history table shows the following commits:

Commit Hash	Commit Message	Time Ago
68399a5	responsive SearchBar Algo1	18 hours ago
	mise à jour des 3 dropdowns et création de la recherche	21 days ago
	correction search algo 1	2 days ago
	ajout de fichier et dropdown ingredient en construction	2 months ago
	ajout de fichier et dropdown ingredient en construction	2 months ago
	modification search bar icon css	2 days ago
	correction search algo 1	2 days ago

# Validation W3C (algo1 et algo2)

- Html

**Document checking completed. No errors or warnings to show.**

Used the HTML parser.

- Css

Résultats de la validation W3C CSS de contact.css (CSS niveau 3 + SVG)

**Félicitations ! Aucune erreur trouvée.**

# Avez-vous des questions ?

