

# Projet Développement Web Sup de Cuisine

**Auteur :** CARREAU Paul-Antoine

**Modules :** MISE EN SITUATION DEV - HTML/CSS/JS

## Introduction

Ce document récapitule les principaux défis rencontrés lors du développement de l'application de recherche et de filtrage "Sup de Cuisine" et les solutions qui ont été mises en place pour y répondre. L'objectif est de fournir un retour d'expérience technique et fonctionnel sur le projet.

## Difficultés Rencontrées et Solutions Apportées

### A. Fonctionnalité de Recherche et Filtrage en Temps Réel

**Difficulté :** Implémenter une recherche dynamique qui se met à jour à chaque caractère saisi et qui combine les critères de recherche principaux avec les tags de filtre (ingrédients, ustensiles, appareil) a nécessité une gestion efficace des performances.

**Solution :** La recherche a été optimisée pour ne se déclencher qu'après la saisie de 3 caractères minimum, réduisant ainsi la charge de calcul. Les ensembles (Set) en JavaScript ont été utilisés pour gérer les tags, permettant une gestion rapide des intersections de filtres et améliorant la fluidité de l'application.

### B. Mise à Jour Dynamique des Options de Filtre dans les Dropdowns

**Difficulté :** S'assurer que les options de filtre dans les listes déroulantes (ingrédients, appareils, ustensiles) reflètent uniquement les options encore disponibles parmi les recettes affichées a nécessité des mises à jour en temps réel de ces options.

**Solution :** La fonction populateFilterOptions a été modifiée pour rafraîchir les options dans les dropdowns en fonction des recettes restantes après chaque application de filtre, assurant une cohérence entre les filtres disponibles et les recettes affichées.

### C. Intersection des Tags de Filtre

**Difficulté :** Garantir que les recettes affichées répondent simultanément à tous les filtres sélectionnés, et non à chaque filtre individuellement, a demandé une logique d'intersection de filtres complexe.

**Solution :** Une logique d'intersection de filtres a été développée pour vérifier chaque recette selon tous les tags actifs, de manière à ne retenir que les recettes correspondant à tous les critères. Cette logique a été testée de manière itérative pour garantir son bon fonctionnement.

### D. Gestion des Cas Particuliers dans l'Interface Utilisateur

Difficulté : Gérer les cas particuliers, comme afficher un message “Aucun résultat” lorsque les filtres ne renvoient aucune recette, ou éviter des listes déroulantes vides.

Solution : Un affichage conditionnel a été mis en place pour afficher un message d’erreur lorsque la recherche ou les filtres n’aboutissent à aucun résultat. Des sécurités ont également été ajoutées pour éviter les états vides dans les dropdowns de filtres.

## **E. Tests et Validation**

Difficulté : Assurer une couverture de tests suffisante, en particulier pour les éléments dynamiques, et intégrer le JavaScript sans bibliothèque externe.

Solution : Des tests unitaires et d'intégration ont été développés pour couvrir la logique de filtrage, la mise à jour des dropdowns, et la recherche. Les tests ont été adaptés pour fonctionner indépendamment du DOM, et des validations manuelles ont été faites pour garantir la conformité W3C.

## **Synthèse des Apprentissages Clés**

---

Utiliser des structures de données performantes comme les ensembles (Set) simplifie la gestion des filtres et des intersections. Les applications en temps réel bénéficient des stratégies de limitation (comme le déclenchement de recherche après un seuil de caractères) pour éviter des calculs inutiles. Une approche de tests robuste améliore la fiabilité des fonctionnalités dynamiques et des traitements de données dans l’application.