


Rappel : pour toute la durée du module, le travail attendu est un TRAVAIL PERSONNEL et de QUALITE PROFESSIONNELLE.

Dans la suite du site développé avec une page d'accueil et des pages de détail générées en JavaScript, vous devez maintenant :

- Basculer en mode « one page » : tous les éléments HTML générés sont placés dans une seule page.
- Seuls les éléments pertinents sont effectivement affichés (ex : quand le menu « Accueil » est actif, on voit la « section » contenant la liste des recettes / pokemons / guitares / chaussures, mais pas les « sections » détaillant chaque recette / ...).
- Vous n'avez le droit d'accéder qu'une et une seule fois à la structure de données qui définit vos éléments (ex : si vous avez défini `var recettes = [ {id: 'recette1', nom: 'Tarte aux pommes',...}, {...},...]`, vous ne pouvez accéder à cette variable qu'une seule fois, et vous ne pouvez pas la copier dans une autre variable (petits coquins ! ) )
- Vous devez rajouter une fonction de recherche qui permette de filtrer les éléments affichés. Cette fonction doit accepter :
  - Une chaîne de caractères simple : seules les recettes (ou autres) contenant quelque part cette chaîne restent visibles
  - Une expression du type `caractéristique [=, >, >=, <, <=] valeur`, par exemple `ingredient = sucre` : seules les recettes dont l'un des ingrédients est le sucre restent affichées
  - Une combinaison d'expression du type `caractéristique1 = valeur & NOT(caractéristique2 = valeur2)`, par exemple : `ingredient = sucre & not(niveau > 3)` : seules les recettes dont l'un des ingrédients est le sucre et dont la difficulté n'est pas supérieure à 3 restent affichées
- Cette recherche doit être basée sur la sémantique du HTML (interdiction de rechercher dans la structure de données : vous devez explorer le DOM et jouer sur le CSS `visibility` ou `display` pour afficher / effacer les structures pertinentes) ; la recherche ne doit pas être perdue après la consultation d'une fiche de détail (voir exemple ci-dessous)
- La syntaxe est libre : vous devez avoir les opérateurs égal, plus grand, plus petit, plus grand ou égal, plus petit ou égal, ET, OU, NON, mais vous pouvez définir librement les symboles qui leur correspondent.
- Il faut donc un popup pour guider l'utilisateur dans votre syntaxe
- Pensez que certaines requêtes peuvent ne pas faire sens : `ingredient > sucre` ne veut pas dire grand-chose => il faut vérifier ce type d'erreur et les indiquer à l'utilisateur

Exemple d'IHM :

Lors du chargement de la page, toutes les sections (liste – en bleu - et toutes les sections de détail – en gris) sont générées. Seule la liste est visible, avec toutes ses présentations synthétiques. Quand l'utilisateur demande à consulter un oiseau, la liste (en bleu) devient invisible (et donc toutes les présentations) et la fiche de l'oiseau (en gris) devient visible. Lorsque l'utilisateur fait une recherche, toutes les fiches deviennent invisible, la liste est affichée et seules les présentations d'oiseaux correspondant à la recherche restent visibles dans la liste. Si l'utilisateur clique sur une présentation, même mécanisme : on affiche la fiche oiseau. Mais quand on revient sur l'accueil, on retrouve la liste filtrée par la recherche.

A European bee-eater (Merops apiaster) is shown perched on a bare, thin branch. The bird has a distinctive yellow throat and breast, a dark cap, and a long, pointed beak. Its wings and back are a mix of brown and blue. The background is a soft, out-of-focus green, suggesting foliage.

]

