

# Projet : Application de Recettes de Cuisine

## Description Générale

Cette application permettra aux utilisateurs de rechercher, ajouter et consulter des recettes de cuisine. Elle inclura également une fonctionnalité pour marquer les recettes comme favorites. Le projet utilisera React JS pour l'interface utilisateur, TypeScript pour la gestion des types et des états, ainsi que HTML/CSS pour le design et la mise en page.

## Fonctionnalités Détails

### 1. Recherche de Recettes

- **Barre de recherche avec suggestions automatiques :**
  - Affichage des suggestions basées sur les caractères saisis.
  - Recherche en temps réel dans une base de données de recettes.

### 2. Affichage des Recettes

- **Liste des Recettes :**
  - Affichage des recettes sous forme de cartes avec un aperçu rapide (image, titre, temps de préparation).
  - Possibilité de filtrer les recettes par catégorie (ex: entrées, plats principaux, desserts) et par type de cuisine (ex: italienne, asiatique).
- **Détails de la Recette :**
  - Page détaillée affichant le titre, l'image, la liste des ingrédients et les étapes de préparation.
  - Informations additionnelles comme le temps de préparation, le nombre de portions et des conseils de cuisine.

### 3. Ajout de Recettes

- **Formulaire de Soumission :**
  - Formulaire permettant d'ajouter une nouvelle recette avec des champs pour le titre, la description, les ingrédients, les étapes de préparation, la catégorie et une image.
  - Validation des champs pour s'assurer que toutes les informations nécessaires sont fournies. La vérification des champs se fera pendant le remplissage de l'utilisateur (Exemple : lorsque l'utilisateur écrit dans le champ « titre » tant que le texte ne fait pas minimum 3 caractère il y aura une erreur d'afficher sous le champs indiquant « le titre doit faire minimum 3 caractères ».)

### 4. Favoris

- **Ajout aux Favoris :**
  - Bouton permettant d'ajouter ou de retirer une recette des favoris.
  - Stockage des recettes favorites dans le local storage pour persister entre les sessions.
- **Affichage des Favoris :**
  - Page dédiée affichant toutes les recettes marquées comme favorites.

# Architecture de l'Application

## Composants React

- **App Component** : Composant principal gérant la navigation et l'état global de l'application.
- **Header Component** : Composant pour le menu de navigation (recherche, ajout de recettes, favoris).
- **RecipeList Component** : Composant pour afficher la liste des recettes avec filtres et recherche.
- **RecipeDetail Component** : Composant pour afficher les détails d'une recette spécifique.
- **RecipeForm Component** : Composant pour le formulaire d'ajout de recette.
- **FavoriteRecipes Component** : Composant pour afficher les recettes favorites.

## Gestion de l'État avec TypeScript

- **State Management** :
  - Utilisation de `useState` et `useReducer` pour la gestion des états locaux et globaux.
  - Utilisation de TypeScript pour définir les types des recettes, des actions et des états.
- **Actions et Reducers** :
  - Définir des actions pour ajouter, supprimer, rechercher des recettes et gérer les favoris.
  - Utiliser des reducers pour manipuler l'état global de l'application.

## Technologies et Outils

### Frameworks et Bibliothèques

- **React JS** : Pour la création des composants de l'interface utilisateur.
- **TypeScript** : Pour le typage statique et la gestion des états.
- **React Router** : Pour la navigation entre les différentes pages de l'application.

### Design et Mise en Page

- **HTML/CSS** : Pour la structure et le style de l'application.
- **Flexbox/Grid** : Pour la mise en page responsive.

### Outils de Développement

- **Visual Studio Code** : IDE recommandé pour le développement.
- **Git/GitHub** : Pour la gestion de versions et le contrôle de source. (BONUS)

## Détails de Déploiement

### Environnement de Développement

- **Node.js et npm** : Pour gérer les dépendances et exécuter les scripts de développement.

- **Setup (BONUS) :**
  - Cloner le dépôt GitHub.
  - Installer les dépendances avec `npm install`.
  - Lancer l'application en mode développement avec `npm start`.
- **Build et Déploiement (BONUS) :**
  - Générer une version optimisée pour la production avec `npm run build`.
  - Déployer sur une plateforme d'hébergement comme Vercel, Netlify ou GitHub Pages.

## Documentation

- **README (BONUS) :**
  - Instructions pour installer et lancer le projet.
  - Description des fonctionnalités et des composants.
  - Explications sur les choix techniques et l'architecture du projet.
- **Commentaires :**
  - Ajouter des commentaires pertinents dans le code pour expliquer la logique et les choix de conception.

Présentation le jour noté « ECF » dans votre planning avec un support de présentation contenant des maquettes réalisées sur Figma et la planification des tâches effectués. Le passage du candidat devra durer environ 15-20 minutes.

## Compétences évaluées :

- 1 : Installer et configurer son environnement de travail en fonction du projet
- 2 : Développer des interfaces utilisateur
- 4 : Contribuer à la gestion d'un projet informatique
- 5 : Analyser les besoins et maquetter une application
- 10 : Préparer et documenter le déploiement d'une application (BONUS)