***MENU “Site Web”***

**. Introduction**

* **Nom du projet** : Site Web de Recettes
* **Objectif** : Fournir une plateforme intuitive où les utilisateurs peuvent rechercher, ajouter, modifier, et partager des recettes culinaires.
* **Public cible** : Amateurs de cuisine, chefs débutants et expérimentés.

**2. Fonctionnalités Principales**

**2. Fonctionnalités**

**2.1 Fonctionnalités Fonctionnelles**

Les fonctionnalités fonctionnelles décrivent les actions et services que le site doit fournir.

1. **Gestion des recettes** :
   * Les utilisateurs peuvent rechercher des recettes par mots-clés, catégorie, ou ingrédient.
   * Les utilisateurs enregistrés peuvent créer, éditer, et supprimer leurs propres recettes.
   * Possibilité d'ajouter des images aux recettes.
2. **Page d'accueil dynamique** :
   * Affichage des recettes les plus populaires et récemment ajoutées.
3. **API REST** :
   * CRUD (Create, Read, Update, Delete) pour les recettes, utilisateurs, et commentaires.
   * Recherche d'ingrédients ou de catégories via des endpoints spécifiques.

**2.2 Fonctionnalités Non Fonctionnelles**

Les fonctionnalités non fonctionnelles décrivent les exigences de performance et qualité.

1. **Performance** :
   * Le site doit répondre en moins de 2 secondes pour 95 % des requêtes.
2. **Design** :
   * Interface utilisateur responsive et accessible.
   * Utilisation cohérente des composants Bootstrap pour une expérience uniforme.
3. **Extensibilité** :
   * Code modulaire pour faciliter les ajouts futurs (nouvelles catégories, fonctionnalités).
4. **Compatibilité** :
   * Fonctionne sur les navigateurs modernes (Chrome, Firefox, Safari, Edge).
   * Optimisation pour mobile, tablette et ordinateur (responsive design).
5. **Maintenance** :
   * Documentation claire pour les développeurs.
   * Gestion des erreurs et logs via un système centralisé.

**3. Architecture du Projet**

* **Structure du Front-End** :
  + React avec des composants fonctionnels et hooks.
  + Gestion de l'état avec Redux ou Context API.
  + Routage avec React Router.
* **Structure du Back-End** :
  + Flask pour les endpoints RESTful.
* **Intégration Front-End/Back-End** :
  + Communication via API avec Axios ou Fetch.

**4. Hébergement**

* **Front-End** : Hébergement sur Netlify ou Vercel.
* **Base de Données** : MYSQL (hébergée sur le cloud, ex. ElephantSQL).

**5. Livrables**

* Code source pour le front-end (GitHub repository).
* Code source pour le back-end (GitHub repository).

**6. Délais**

* Phase de conception : 2 semaines. / 1 jour
* Développement front-end : 4 semaines. /4 jours
* Développement back-end : 4 semaines. / 4 jours

**7. Contraintes Techniques**

* **Front-End** : React 18.x, Bootstrap 5.
* **Back-End** : Python 3.9+, Flask 2.x.
* **Base de Données** : MYSQL.

**8. Critères de Validation**

* Interface utilisateur responsive et fonctionnelle.
* Toutes les fonctionnalités du cahier des charges sont implémentées.
* Tests d'intégration et unitaires réussis.
* Documentation complète.