**Documentation Technique – Projet EcoCook**

**1. Présentation générale**  
**Technologies utilisées** :

* Vue.js (frontend)
* Vite (outil de build)
* Node.js + Express (backend)
* MySQL (base de données)
* LocalStorage (sauvegarde côté client)
* CSS (Styling léger pour réduire l'impact écologique)

**Objectif** :  
Proposer un site web écoresponsable de recettes de cuisine, respectant les principes du Green IT, permettant de :

* Consulter des recettes filtrées par saison, CO2, nombre de personnes.
* Rechercher une recette par son nom.
* Se connecter / s’inscrire.
* Ajouter des recettes aux favoris.

**2. Architecture du projet**

**Frontend** (/src/views/) :

* AccueilView.vue → Page d'accueil
* CatalogueView.vue → Liste de toutes les recettes + filtres
* RechercheView.vue → Recherche de recettes par nom
* FavorisView.vue → Liste des favoris
* LoginView.vue → Connexion
* RegisterView.vue → Inscription
* AboutView.vue → Page de présentation Green IT

**Backend** (/controllers/) :

* recipeapi.route.js → Routes API pour gérer les recettes, utilisateurs, favoris

**Base de données** :

* Tables : users, recipes, ingredients, recipe\_ingredients, favorites, seasonal\_catalog

**3. Base de données**

**Nom de la base** : eco\_cook

**Tables principales** :

* users : stocke les utilisateurs (id, email, mot de passe hashé, rôle, username).
* recipes : stocke les recettes (id, titre, description, instructions, image, saison, empreinte carbone, nombre de personnes).
* ingredients : stocke les ingrédients disponibles.
* recipe\_ingredients : table de liaison recettes/ingrédients (quantité, unité).
* favorites : table reliant un utilisateur à ses recettes favorites.
* seasonal\_catalog : catalogue des fruits/légumes selon les saisons.

**4. Fonctionnalités**

**Côté utilisateur :**

* **Page d’accueil** : Présentation du projet et catalogue de saison.
* **Catalogue** : Filtres dynamiques (saison, CO2, personnes).
* **Recherche** : Recherche d’une recette par son titre.
* **Favoris** : Liste personnalisée de recettes favorites pour les utilisateurs connectés.
* **Connexion/Inscription** : Accès aux fonctionnalités avancées après connexion.

**Côté administrateur :**

* Gestion potentielle de nouvelles recettes, d'ingrédients (non finalisé dans la V1).

**5. Technologies et choix Green IT**

* Vue.js : framework rapide, léger et efficace.
* Images compressées et tailles réduites.
* Filtres effectués en local (pas de requêtes serveur inutiles).
* Favoris stockés dans le navigateur pour limiter la charge serveur.
* Design simple et épuré : pas d'animations lourdes, pas de vidéos.
* Utilisation de composants réutilisables pour éviter la duplication de code.
* Possibilité d'hébergement sur un service écoresponsable (type OVH Green ou Netlify).

**6. Installation et déploiement**

**Backend**

1. Se placer dans backend/
2. Installer les dépendances : npm install
3. Lancer le serveur : node server.js

**Frontend**

1. Se placer dans frontend/
2. Installer les dépendances : npm install
3. Lancer le serveur Vite : npm run dev