

Documentation du Projet "VideTonFrigo"

Installation

Prérequis

Assurez-vous que vous avez Node.js et npm (Node Package Manager) installés sur votre machine.

Étapes d'installation

- 1) Création de l'application React

```
npx create-react-app vide-ton-frigo
```

- 2) Accéder au répertoire du projet

```
cd vide-ton-frigo
```

- 3) Démarrer l'application React

```
npm start
```

1. Ceci lance l'application React dans votre navigateur par défaut à l'adresse <http://localhost:3000>.

Structure du Projet

- public/ : Contient les fichiers statiques du projet.
- src/ : Contient le code source de l'application React.
 - assets/ : Stocke les fichiers multimédias tels que les images et les icônes.
 - components/ : Composants React réutilisables.
 - styles/ : Fichiers CSS pour la stylisation.
 - App.js : Composant principal de l'application.
 - index.js : Point d'entrée de l'application.

Composants Principaux

HomePage Component

- HomePage.js : Composant représentant la page d'accueil.
- styles-desktop.css : Fichier CSS contenant les styles pour les écrans de bureau.
- styles-mobile.css : Fichier CSS contenant les styles pour les écrans mobiles.

Ingrédients Component

- Ingrédients.js : Composant permettant aux utilisateurs de saisir et gérer leurs ingrédients.
- Ingredients.css : Fichier CSS contenant les styles pour ce composant.

Recettes Component

- Recettes.js : Composant affichant les recettes basées sur les ingrédients sélectionnés.
- Recettes.css : Fichier CSS contenant les styles pour ce composant.

Personnalisation

Styles et Mise en Page

- Les styles de la page d'accueil sont définis dans les fichiers CSS correspondants, adaptés aux écrans de bureau et mobiles.
- Les tailles des éléments, les couleurs et les polices peuvent être personnalisées selon les besoins.

Liens vers les Stores

- Le bouton "Télécharge-moi" sur la page d'accueil peut être lié aux boutiques d'applications respectives en modifiant les liens dans le fichier approprié.

Défilement en Douceur

- L'effet de défilement en douceur vers le bas de la page est implémenté avec un bouton "Scroll". Vous pouvez personnaliser cet effet en ajustant les styles et les références dans le code.

Fonctionnalités

Ajout d'ingrédients

- Les utilisateurs peuvent ajouter des ingrédients via un formulaire dédié. Les ingrédients sont ensuite affichés sous forme de liste.

Recherche de recettes

- L'application utilise une API pour rechercher des recettes en fonction des ingrédients fournis par l'utilisateur.

Sauvegarde locale

- Les ingrédients ajoutés par l'utilisateur sont sauvegardés localement dans le navigateur pour une utilisation future.

Dépendances

Liste des principales dépendances utilisées dans le projet :

```
"dependencies": {
```

```
"axios": "^0.21.1",    // Pour les requêtes HTTP
```

```
"react-router-dom": "^5.2.0" // Pour la gestion des routes
```

```
}
```

Vous pouvez installer ces dépendances en utilisant la commande suivante :

```
npm install axios react-router-dom
```

Étapes de Réflexion et de Réalisation

1. Idée Initiale et Brainstorming

L'idée de "VideTonFrigo" est née de la volonté de réduire le gaspillage alimentaire en utilisant les ingrédients disponibles pour créer des recettes. Nous avons brainstormé plusieurs fonctionnalités potentielles et avons décidé de commencer par une fonctionnalité de base permettant d'entrer des ingrédients et de rechercher des recettes.

2. Recherche et Planification

Nous avons recherché des API disponibles pour obtenir des recettes et avons opté pour l'API Spoonacular en raison de sa simplicité d'utilisation et de ses fonctionnalités. Nous avons également planifié la structure de l'application et listé les principaux composants nécessaires.

3. Conception de l'Interface Utilisateur

Nous avons esquissé plusieurs maquettes de l'interface utilisateur en utilisant des outils comme Figma. Nous avons décidé d'avoir une page d'accueil simple avec un formulaire pour entrer les ingrédients et un affichage des recettes en dessous.

4. Développement Initial

Nous avons initialisé le projet avec Create React App et avons commencé à développer les composants de base. Nous avons commencé par le composant "Ingredients" pour permettre aux utilisateurs d'ajouter et de gérer leurs ingrédients.

5. Intégration de l'API

Nous avons intégré l'API Spoonacular pour récupérer les recettes en fonction des ingrédients. Cela impliquait la création de requêtes HTTP avec Axios et la gestion des réponses de l'API.

6. Tests et Debugging

Nous avons écrit des tests unitaires pour les composants principaux afin de nous assurer que chaque partie fonctionnait correctement. Nous avons également effectué des tests manuels pour vérifier l'intégration de l'API et le fonctionnement global de l'application.

7. Améliorations et Optimisations

Après avoir obtenu une version fonctionnelle de base, nous avons recueilli des retours d'utilisateurs et avons apporté des améliorations, telles que le stockage local des ingrédients et une meilleure gestion des erreurs de l'API.

8. Déploiement

Nous avons préparé l'application pour le déploiement en optimisant les performances et en configurant les paramètres de production. L'application a été déployée sur [service de déploiement choisi] (ex. Vercel, Netlify).

9. Documentation

Enfin, nous avons documenté le projet, y compris l'installation, la structure du projet, les composants principaux, les fonctionnalités et les étapes de réflexion et de réalisation.