

Documentation

Objectif

L'objectif de notre application est de fournir aux utilisateurs une plateforme pour la planification de repas, en leur permettant de créer des plans de repas personnalisés, de gérer leurs recettes, d'obtenir des informations nutritionnelles et de suivre leurs objectifs de santé.

Architecture

Notre application est développée en utilisant React Native, qui nous permet de créer des applications mobiles multiplateformes. Nous utilisons également les bibliothèques Expo et React Navigation pour faciliter le développement et la navigation au sein de l'application.

L'architecture de l'application suit un modèle de navigation par onglets, où chaque onglet représente une fonctionnalité spécifique. Nous utilisons un BottomTabNavigator pour afficher les différents écrans d'onglets.

Nous avons également utilisé un contexte (MealPlanProvider) pour gérer l'état global de l'application lié à la planification des repas.

Il y a 3 onglets :

- 1 - Health Goals
- 2 - Food Database
- 3 - Meal Planning

Health Goals est une page de l'application qui permet aux utilisateurs de définir leurs objectifs de santé, tels que la perte de poids, le maintien du poids ou la prise de poids, en fournissant des informations telles que l'âge, le genre, la taille, le poids, le niveau d'activité et l'objectif de santé souhaité. Cette page calcule ensuite le métabolisme de base (BMR) en fonction de ces informations, ce qui aide les utilisateurs à mieux comprendre leurs besoins caloriques quotidiens.

Food Database est une page de l'application qui permet aux utilisateurs de rechercher des informations nutritionnelles sur différents aliments. Les utilisateurs peuvent entrer le nom d'un aliment dans la barre de recherche (il pourra s'aider des suggestions émises à chaque nouvelle lettre que l'on tape), et la base de données affiche les détails nutritionnels tels que les calories, les macronutriments et les micronutriments de l'aliment recherché. Cela aide les utilisateurs à prendre des décisions alimentaires éclairées en connaissant la composition nutritionnelle des aliments.

Meal Planning est une page de l'application qui permet aux utilisateurs de créer des plans de repas personnalisés en fonction de leurs objectifs de santé et de leurs préférences alimentaires. Les utilisateurs peuvent sélectionner les aliments de la base de données alimentaire, les ajouter à leur plan de repas et organiser les repas pour chaque journée de la semaine. Cette fonctionnalité facilite la planification des repas équilibrés et aide les utilisateurs à suivre un régime alimentaire approprié pour atteindre leurs objectifs de santé.

Fonctionnalités réussies

- Page Health Goals complète
- Page Food Database complète
- Page Meal Planning complète

Création et affichage des plans de repas personnalisés pour chaque jour et chaque repas.

Ajout et suppression de plats dans les plans de repas.

Affichage des informations nutritionnelles et image pour chaque plat (grâce à l'API Edamam).

Suivi des objectifs de santé, tels que la perte de poids ou la prise de poids, en ajustant les calories recommandées.

Navigation fluide entre les différentes pages.

Ajout de l'autocomplétion (limite de 3 suggestions) pour faciliter la saisie des données d'un plat dans Food Database (grâce à l'API Edamam).

Fonctionnalités non réussies

Je voulais reprendre le template d'un design d'application qui a des similitudes avec notre appli, mais je n'ai pas réussi à implémenter le design à la fin.

Feuille de route


Voici la feuille de route pour les futures améliorations de l'application :


- Ajouter la possibilité de personnaliser les portions des recettes pour ajuster les informations nutritionnelles en conséquence.
- Intégrer des fonctionnalités de partage pour permettre aux utilisateurs de partager leurs plans de repas ou leurs recettes avec d'autres utilisateurs.
- Améliorer l'interface utilisateur et l'expérience utilisateur en ajoutant des animations et des interactions plus fluides.
- Optimiser les performances de l'application pour assurer une utilisation efficace des ressources du périphérique.

Captures d'écran


Page Health Goals :


Health Goals


 Age

 Gender


Masculin ▾

 Height (in cm)


 Weight (in kg)

 Activity Level


Moderate exer... ▾


 Health Goal

Weight mainte... ▾


 Calculate


Health Goals


 Age

 Gender


Masculin ▾

 Height (in cm)


 Weight (in kg)

 Activity Level


Extra active ▾


 Health Goal


Weight loss ▾


 Calculate


Caloric Intake : 2766 calories


 Health Goals

 Food Database

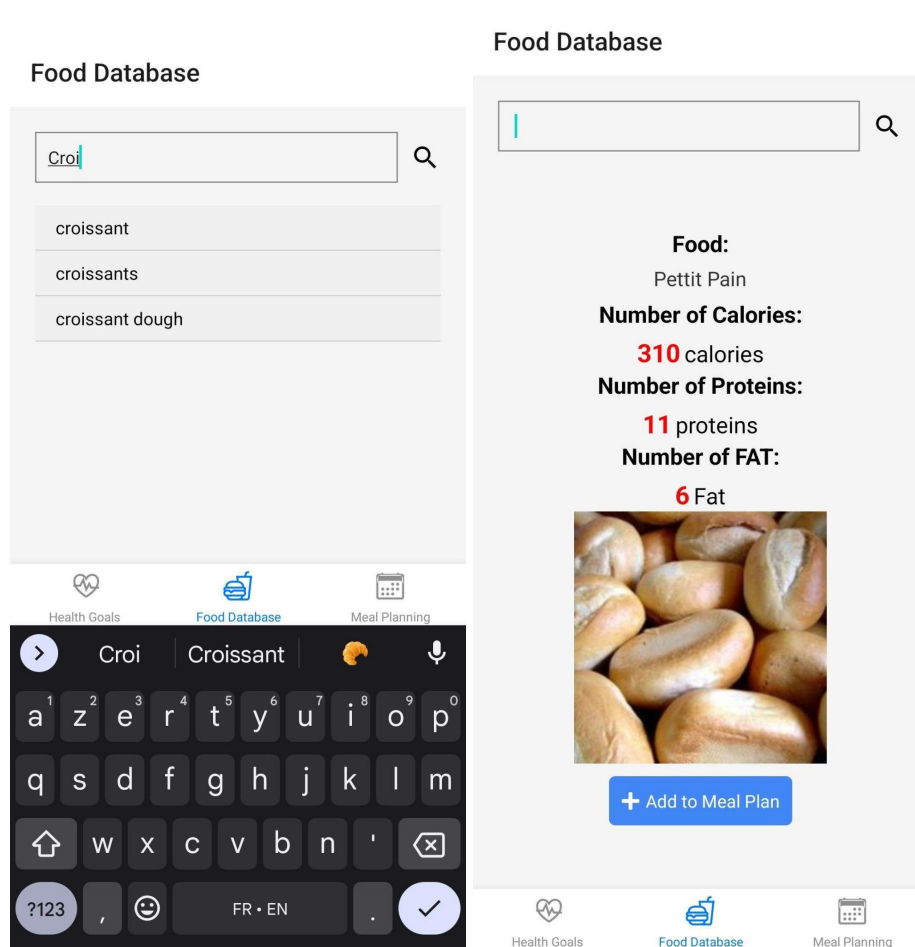
 Meal Planning

 Health Goals

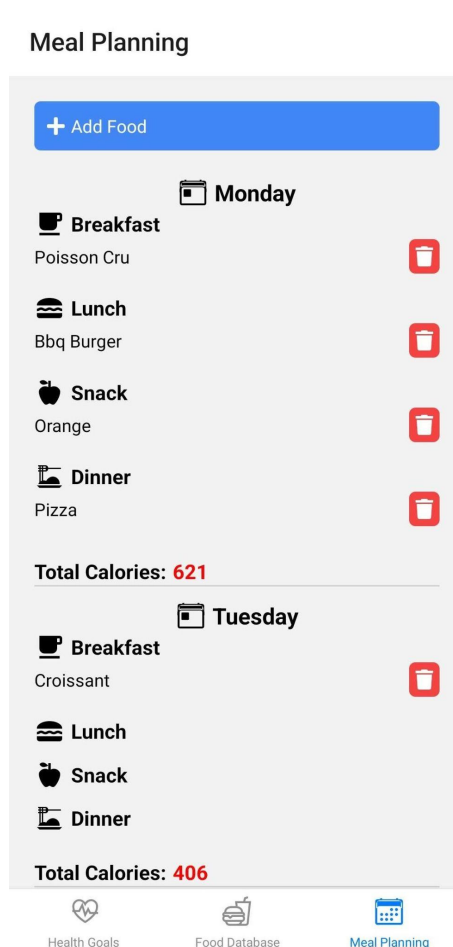
 Food Database

 Meal Planning

Page Food Database :



Page Meal Planning :



Bottom Tab Navigator:



Logo :



Difficultés rencontrées

L'une des principales difficultés auxquelles nous avons été confrontés lors du développement de cette application a été la gestion de l'état global de l'application. Nous avons utilisé le contexte (MealPlanProvider) pour partager les données entre les différents composants, mais cela a nécessité une bonne compréhension de la gestion de l'état et de la synchronisation des données.

De plus, l'intégration de bibliothèques tierces et la configuration de la navigation ont été des défis mineurs, mais nous avons pu les surmonter en consultant la documentation et en effectuant des recherches approfondies.