



Projet M.I.A.G.E

Description

M.artini

I.zarra

A.maretto

G.in

E.au de vie

Sujet

Une appli concernant des recettes de cocktails avec au moins 4 pages:

1. Accueil / Suggestion du jour
2. Filtrage de cocktails
3. Favoris / déjà faits
4. A propos / infos externes ...

APIs possible :

[Free API | TheCocktailDB.com](#)

<https://fr.openfoodfacts.org/data>

URL à utiliser pour accéder aux données d'un produit :

[https://world.openfoodfacts.org/api/v0/product/\[code-barre\].json](https://world.openfoodfacts.org/api/v0/product/[code-barre].json)

Exemple : <https://world.openfoodfacts.org/api/v0/product/5000281031033.json>

Plugins capacitors à utiliser :

[Lire des codes barres](#)

[Enoncer la recette \(Text2Speech\)](#) ou [ça](#)

[Shaker pour valider et ajouter aux favoris](#)

[Local Notifications Capacitor Plugin API](#)

Réalisation :

Le fichier ReadMe du projet comporte un certain nombre d'informations que nous allons résumer ici. Pour plus d'information, n'hésitez pas à vous y reporter.

Après mûre réflexion et quelques idées proposées plus haut, nous avons fait le choix de conserver uniquement l'utilisation de l'API de cocktail et les deux plugins Capacitor que sont "Motion" et "localNotification". Nous avons décidé de retirer la fonctionnalité de la lecture de code-barres au vu de la complexité qu'aurait pu apporter ce développement. L'objectif de base avec ce plugin était de pouvoir par la suite utiliser l'API openfoodfacts afin de proposer des cocktail en rapport avec l'article scanné,

Globalement, des choix techniques ont été faits. On peut noter le fait d'utiliser le localstorage afin d'éviter de relancer des appels à chaque retour sur la page d'accueil. Le localstorage a également été utilisé dans le reste du projet comme pour la gestion des favoris pour des raisons de simplicité.

Des fonctionnalités ont été ajoutées en plus de ce qui avait été défini au début du projet comme l'utilisation du module i18n afin de permettre d'accéder à l'application en deux langues.