Projet Web-side Client-Side

Organisation de travail:

- Côté client : HUANG Xiaohan
- Côté serveur: Sellami Asma & BEN TARJEM Salma.

Les fonctionnalités qu'on a préparées :

1. Fonctionnalités de rendu dimanche
    - Rechercher des aliments existants dans la base de données en fonction de critères donnés.
    - Attribuer une note globale aux aliments en fonction d'une combinaison de critères. (il faut lancer le insertScore.js)
    - Attribuer des prix au produits par distributeur ("Monoprix", "LidIt"....) (il faut lancer le insertPrice.js)
    - Ajouter une recette simple et rechercher les ingrédients appropriés dans la base de données.
    - Voir les détails de chaque aliments.
    - Trier les aliments par prix et par score.
    - Authentification ,Register.

2. Fonctionnalité amélioration pour la présentation (dans branche *amélioration*)
    - Mis en jour de prix de produit

Partie serveur:

Une fois que vous avez cloné ou téléchargé ce projet, vous devez vous assurer d'avoir installé Mongo DB, puis lancez la commande suivante pour apporter tous les packages requis pour ce projet: **npm install**.

I. Bibliothèques utilisées:

- Express: un framework pour construire des applications web basées sur Node.js
- Mongoose: pour gérer les relations entre les données et fournit la validation du schéma.
- body-parser:objet exposé diverses usines pour créer des middlewares
- Jsonwebtoken:(JWT) pour les jetons de signature.
- Passport-jwt: middleware pour obtenir et vérifier les fichiers JWT.
- Bootstrap: en tant que cadre de style

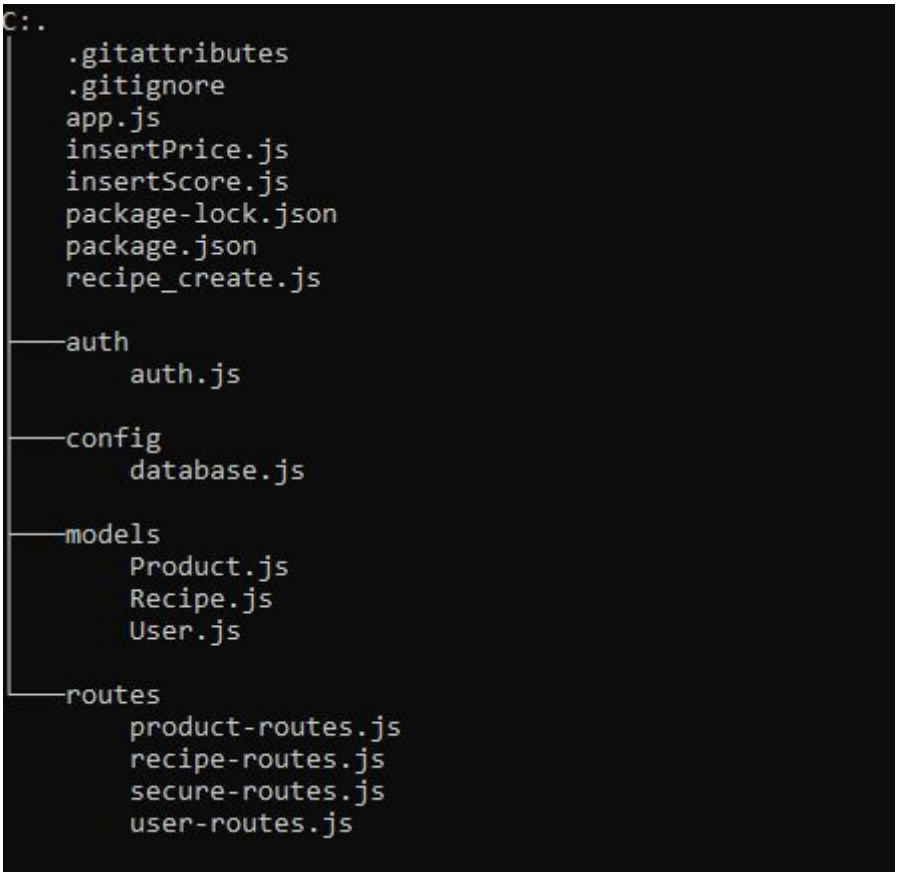
II. Installation:

```
npm install express --save
npm i mongoose
npm i body-parser
```

npm install --save bcrypt body-parser express jsonwebtoken mongoose passport passport-local passport-jwt
npm install method-override
npm install moment

III. Architecture de projet:

- Cette figure présente l'arborescence de notre projet



Notre projet comporte:

- 3 modèles: Product ,Recipe,User.
- 4 routes:product-routes ,user-routes,recipe-routes,secure-routes.

pour tester la partie serveur juste lancez la commande :
node app.js. ou **nodemon start** .

IV. Préparation de la base de données :
Avant de commencer nous avons importé la base de données à l'aide de la commande suivante :
mongoimport -d products -c products --file products.json.
après,nous avons nettoyé et éliminé les aliments qui n'avaient ni catégories,ni nom ,ni nutriments nous avons utilisé les commandes suivantes :

db.products.remove({"nutriments":{$exists: false}})

db.products.remove({"product_name":{$exists: false}})
db.products.remove({"catégories":{$exists: false}}).

Il faut lancer les fichier insertPrice.js et insertScore.js pour ajouter les prix et les scores au produits, ceux sont tâches qui doivent être réalisées une fois, c'est ce qui explique le faite qu'elles sont séparées du app.js.

Partie Client:

- **Framework utilisé:** Vue CLI
- **Installation besoin:**
    - `npm install -g vue-cli`
    - `npm install webpack`
    - `npm install webpack-dev-server`
    - `npm install cross-env`
    - `npm install babel-loader`
    - `npm install babel-core`
    - `npm i babel-preset-env`
    - `npm install --save-dev babel-preset-stage-3`
    - `npm install vue`
    - `npm i vue-loader`
    - `npm install vue-template-compiler`
    - `npm install css-loader`
    - `npm install file-loader`
    - `npm install vue-router`

- **Lancement de client:** Dans terminal, sur le chemin de client, entrez la ligne de commande "`npm run dev`"
- **Adresse de client:** http://localhost:8888