Réalisé par Antoine CONTOUX et Paul LOISIL.

Notre configuration

L'ensemble des instructions pour déployer ce projet est disponible dans le fichier README.md.

Machines de développement

Sur nos machines, nous travaillons avec un environnement XAMPP.

Serveur de production

Le serveur est hébergé sur un pc de type "barebone" chez un des membres du groupe. Le nom de domaine est "cocktails.projects.antoinectx.fr" (enregistré avec OVH) et le certificat SSL est délivré par Let's Encrypt (avec certbot).

Configuration du serveur :

- Ubuntu Server 22.04.1 LTS
- Apache 2.4.52
- PHP 8.1.2
- MariaDB 10.6.11

Le VirtualHost donné en exemple dans le fichier README.md est celui utilisé sur le serveur.

Technologies utilisées

Back-end

- PHP
  - Composer, pour la gestion des dépendances
  - Eloquent, pour la gestion de la base de données
  - Phinx, pour la gestion des migrations
  - Slim, pour la gestion des routes

Ces technologies ont toutes été étudiées et utilisées à l'IUT Nancy-Charlemagne, excepté Phinx (découvert en entreprise durant le stage d'un des membres du groupe).

Front-end

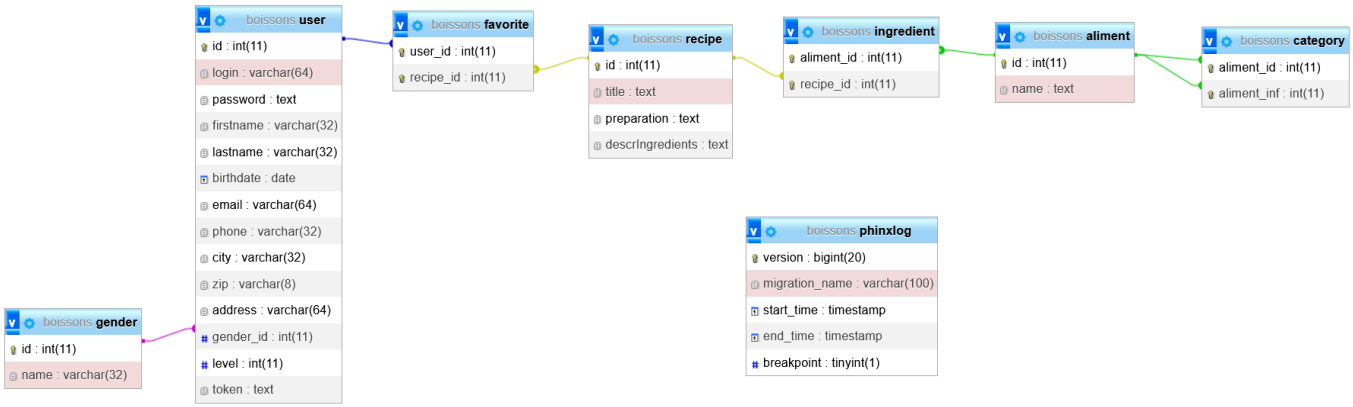
```
TODO
```

Analyse et solutions

Stockage des données

Base de données

Les données sont stockées dans une base de données MySQL/MariaDB. Le schéma de celle-ci est présenté ci-dessous.



La base de données est construite à partir des différentes migrations Phinx présentes dans le dossier src/api/db/migrations. C'est ce qui nous a permis de partager la même structure de base de données entre les différents membres du groupe et de développer rapidement. C'est donc ce qui explique la présence de la table `phinxlog` dans le schéma de la base de données ci-dessus.

Remplissage de la base de données

Le remplissage de la base de données ne peut être effectué que par un utilisateur ayant les droits d'administrateur. Il faut que la procédure d'insertion des données dans la base de données soit simple et rapide. Pour cela, nous avons choisi de la laisser à la charge de l'administrateur, depuis l'interface web.

La procédure est détaillée dans le fichier README.md.

Lors du remplissage, les tables `recipe`, `ingredient`, `aliment` et `category` sont remplis à partir du fichier src/assets/Donnees.inc.php. Ces tables sont vidées avant remplissage afin d'éviter les doublons.

Recherche

Lors de la recherche d'un cocktail, l'utilisateur peut rechercher un nom, mais aussi filtrer par ingrédient.

Front

```
TODO: parler de la hiérarchie des aliments, comment tu les affiches, et comment tu affiches les résultats
```

Back

Le front envoie une requête à l'API qui va récupérer les cocktails correspondants à la recherche de l'utilisateur. La requête est composée du texte de recherche, des ingrédients obligatoires et des ingrédients interdits.

Étapes de la recherche (effectuées par l'API) :

- On commence par récupérer qui contiennent au moins un mot de la recherche dans leur nom.
- On filtre pour ne garder que ceux qui contiennent tous les ingrédients obligatoires.

- On filtre pour ne garder que ceux qui ne contiennent aucun des ingrédients interdits.

Les résultats sont triés de la manière suivante avec un calcul de score :

- Pour chaque mot de la recherche, +1 si le nom du cocktail contient le mot ;
- Pour chaque mot de la recherche, +1 si un ingrédient du cocktail contient le mot ;
- Pour chaque mot de la recherche, on divise les points pour ce mot par 2 si le mot est entre parenthèses.

Formulaires

Front

```
TODO, si tu as qqc à dire
```

Back

Les formulaires de connexion et d'inscription sont gérés par l'API. L'API vérifie que les données envoyées par le front sont valides et renvoie une erreur si ce n'est pas le cas. Le détail des vérifications est présenté dans le fichier routes.md. Nous sommes assez stricts sur le mot de passe, car nous ne voulons pas que les utilisateurs puissent se connecter avec un mot de passe trop simple. Les paramètres optionnels vides sont ignorés. En revanche, les paramètres obligatoires vides sont considérés comme invalides.

Chaque vérification produit un message d'erreur différent, ce message est ensuite affiché par le front. Les messages renvoyés par l'API peuvent être traduits en plusieurs langues. Nous avons travaillé au départ en anglais, mais nous avons décidé de traduire en français car le front est en français.

Favoris

Front

```
TODO, si tu as qqc à dire
```

Back

L'API permet d'ajouter ou de supprimer de l'utilisateur courant une liste de cocktails en favoris. L'API vérifie seulement que l'id du cocktail est valide et que l'utilisateur est connecté. Il est possible de donner plusieurs fois le même id de cocktail ou de supprimer un cocktail qui n'est pas en favoris.

Connexion

Front

```
TODO, si tu as qqc à dire
```

Back

Lors de la connexion ou de l'inscription, si les champs sont valides, l'API renvoie le token de l'utilisateur. C'est ce qui permet de ne pas avoir à se reconnecter à chaque fois que l'on envoie une requête à l'API ou à garder en mémoire le mot de passe.

En prenant Discord comme exemple, nous avons décidé de changer le token lors du changement de mot de passe.