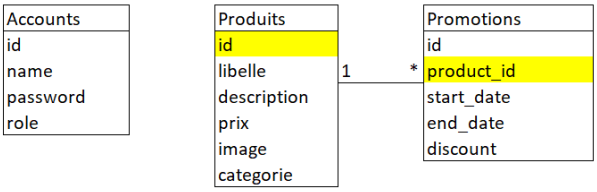
Documentation Technique

- Architecture
    - Architecture BDD

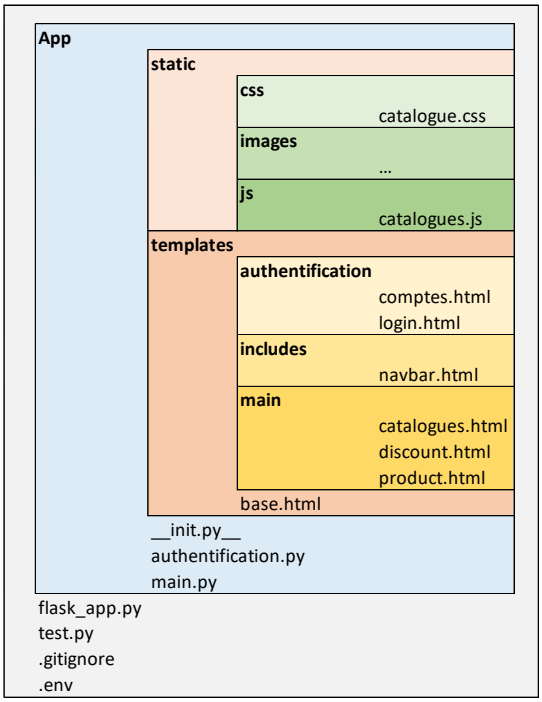


L'application, dans l'état actuel de la demande, nécessite une architecture des données assez simple.

Une table 'accounts' pour gérer les identifiants. Nous avons prévu un champ rôle au cas où il y aura besoin d'attribuer différentes permissions à l'avenir puisque nous sommes sur une application avec une partie back administrateur.

Une table 'produits' et une table 'promotions' pour la gestion respective de ces deux éléments. Il y a une clé étrangère dans la table promotions faisant référence à l'id du produit en question. Un produit peut avoir plusieurs promotions mais sur des intervalles de dates toujours différentes (logique mise en place dans les requêtes sql du backend). Une promotion, d'un point de vue de la base de données, ne correspond qu'à un seul produit. Dans l'application on peut rentrer simultanément 'la même promotion' pour plusieurs produits, mais dans la base elles seront toutes indépendantes.

- Architecture logicielle de l'application



A la racine du projet on trouve le fichier de lancement de l'application. Le dossier contenant l'application, les fichiers cachés (variables d'environnements, le gitignore …), le fichier de test.
Le dossier de l'applicatif contient le fichier d'initialisation de l'application ainsi que les fichiers gérants les différentes routes/fonctions.
On y trouve également un dossier static pour gérer les fichiers javascripts et css, mais également pour gérer les fichiers uploader par l'utilisateur (ici des images).
Enfin un dossier templates pour la partie front-end de l'application. On y trouve une template pour chaque page mais également une template pa élèment récurrent (ici la navbar) et un pour la tram générale/informations qui ne devraient pas changer d'une template/page à l'autre.

- Technologies
    - Langages de programmations
        - Postgresql
        - Python
        - Javascript
        - Html
        - Css
    - Frameworks
        - Flask
        - Jinja2
    - Bibliothèques
        - psycopg2
        - dotenv
        - jwt
        - functools
        - werkzeug
        - datetime
        - contextlib
        - uuid

- Sécurité
    - Tout d'abord le système d'authentification pour les administrateurs, avec un système de token et de hachage de mot de passe. Seulement les administrateurs, une fois authentifié, peuvent insérer des données dans le système.

    - Les mots de passe sont hachés avec Werkzeug et les tokens créés avec jwt. C'est le mot de passe haché qui est sauvegardé dans la base de données et utilisé par comparaison lors de la demande de connexion d'un compte. Le token quant à lui est créée lors de la connexion au compte, il est personnalisé et à durée limitée.

    - Le token est vérifié pour accéder à certain contenus/fonctionnalités grâce à un décorateur créée spécialement à cet effet et appelé token_required_auth.

    - Il est important les messages d'erreurs ne communiquent pas d'informations sensibles une fois en production.

    - Nous vous recommandons également de garder systématiquement les bibliothèques et frameworks utilisés dans l'application, à jour avec les derniers correctifs de sécurité.

- Test
    - Des tests ont été mise en place pour tester les différentes technologies dans 0unittest.py. Cela permet de comparer les technologies et de voir si elles correspondent aux attentes avant de les intégrer dans le projet.

    - Des tests ont été mise en place pour tester les différentes routes de l'applications dans test_flask_app.py. Cela permet de s'assurer que toutes les routes continuent de

fonctionner correctement tout au long de la phase de développement et même durant l'ensemble de son cycle de vie, ce qui facilite la maintenance de l'application.

- Hébergement
    - Considérant une application Flask, nous sommes partis sur un hébergement chez PythonAnywhere. Cela correspond donc parfaitement au projet. Ils permettent également d'héberger la base de données Postgresql. On pourra y héberger l'ensemble de l'application, de la base de données jusqu'au front.