

Rapport SAE 2.03

L'objectif de cette SAE était de transformer un site portfolio statique, réalisé lors de la SAE 1.04, en une application web dynamique, interactive et sécurisée, à l'aide du framework Flask. L'application devait permettre :

- La gestion des compétences par semestre et par bloc
- La validation du niveau d'acquisition des compétences via un formulaire
- La connexion sécurisée d'utilisateurs avec authentification
- Le déploiement du site dans un environnement conteneurisé avec Docker.

Ce projet m'a permis de consolider mes connaissances en développement web Python, en bases de données relationnelles, ainsi qu'en déploiement d'applications professionnelles.

Mise en place du projet

Reprise du portfolio de la SAE 1.04

Pour démarrer, j'ai repris le squelette HTML/CSS de mon portfolio créé en SAE 1.04. Celui-ci comportait déjà plusieurs pages : accueil, hobby, profil, etc. Ce socle m'a permis de :

- Garder une cohérence graphique et ergonomique,
- Y intégrer les éléments dynamiques Flask sans repartir de zéro,
- Organiser les templates avec Jinja2, pour éviter les redondances de code HTML.

Création de la base de données PostgreSQL

J'ai configuré une base de données nommée **dbaurel**, avec l'utilisateur **aurelien** et le mot de passe **progr00**.

La structure de la base respecte l'organisation suivante :

- Semestres : chaque semestre contient plusieurs blocs.
- Blocs de compétences : liés à un semestre.
- Compétences (apprentissages critiques) : liées à un bloc.
- Niveau d'acquisition : stocké dans une table liée à l'utilisateur et à la compétence.
- Utilisateurs : identifiants et mots de passe pour l'authentification.

Les relations entre ces entités sont modélisées avec SQLAlchemy, et permettent de faire des requêtes croisées facilement dans Flask.

Développement de l'application Flask

Le fichier principal **app.py** contient l'ensemble des routes et la logique du site :

- Route **/** : accueil du portfolio
- Route **/profil** : affichage des informations de l'étudiant
- Route **/competences** : affichage des compétences classées par semestre/bloc
- Route **/ajouter** : formulaire de validation de compétence
- Routes **/login** et **/register** : gestion de l'authentification utilisateur

Les données affichées dans les pages sont passées dynamiquement via templates Jinja2, avec des boucles et conditions pour s'adapter aux différents blocs et niveaux.

Mise en place de l'authentification

J'ai intégré un système de connexion utilisateur avec :

- Inscription (enregistrement dans la base PostgreSQL)
- Connexion et gestion de session via Flask-Login
- Redirection vers un espace protégé, accessible uniquement après connexion

Cela permet à chaque utilisateur de gérer ses propres compétences sans interférer avec celles des autres.

Déploiement avec Docker

L'ensemble du projet est conteneurisé grâce à Docker, ce qui le rend facilement portable et reproductible.

Les fichiers utilisés :

- **Dockerfile** : construction de l'image Flask, à partir d'une image Python officielle.
- **requirements.txt** : contient les dépendances du projet (Flask, psycopg2, SQLAlchemy, etc.).
- **docker-compose.yml** : définit deux services :
 - **web** (le serveur Flask exposé sur le port 5000)
 - **db** (le serveur PostgreSQL, contenant la base **dbaurel**)

Le lancement du projet se fait via **docker-compose up**, permettant de tout exécuter dans un environnement isolé.

Compétences développées

- Développement d'une application web dynamique avec Flask
- Modélisation et gestion de base de données relationnelle PostgreSQL
- Intégration d'un système d'authentification sécurisé
- Création de formulaires dynamiques et traitement des données utilisateur
- Utilisation de Docker et docker-compose pour le déploiement multi-conteneurs
- Structuration professionnelle d'un projet Python (app, routes, modèles, templates)

Conclusion

Cette SAE m'a permis de développer un projet complet et concret, regroupant les éléments essentiels d'un site web professionnel : base de données, sécurité, interface utilisateur et hébergement. Le fait d'avoir repris mon portfolio existant et de l'avoir rendu dynamique constitue une évolution directe de mes compétences en développement web.

Grâce à cette expérience, je suis désormais capable de concevoir, développer et déployer une application web fonctionnelle en environnement conteneurisé, tout en respectant une structure claire et maintenable.