

Workshop Archi. logicielle Web

Objectifs pédagogiques : Construire un site web dynamique complet

Outils utilisés : Flask, serveur de Bdd

Langages : HTML, CSS, Json, Python, SQL, et un peu de Javascript

Concepts : Serveur Web, pages dynamiques, templates, MVC, API, REST

Sujet

L'objectif est de créer un serveur capable de stocker une liste d'étudiants, de leurs points forts (dev, montage, musique, dessin, etc). En recensant les talents de chaque étudiant, on pourra ensuite monter des groupes pour les projets, qui listeront les atouts de l'équipe.

Autre sujet possible : liste des stages/entreprises/étudiants (avec leur année de promo).

Conditions de réalisation

En **Flask**, pour la partie back, avec un moteur de **Bdd** (SQLlite toléré)

Usage du **git** très très recommandé, et notamment pour montrer la liste des commits et de la répartition du travail

L'API doit être réalisée en **REST** (le plus possible), et les échanges au format **JSON** (sauf les quelques pages de présentation globale)

MVP (*minimal viable product*)

- la description du projet, l'organisation de l'équipe
- le MCD qui va servir dans l'application
- Proposer la gestion de base des éléments choisis (liste, ajout, modif, suppression)
- Pouvoir réaliser une association (par exemple, pouvoir ajouter supprimer un talent à un étudiant)

V1

- les associations doivent fonctionner, la globalité du site aussi (consultation, ajout, suppression)
- un petit topo sur l'organisation de l'équipe, et le travail de chacun

Modalité de réalisation

Par équipe, trinôme MAXIMUM. Choisissez un nom pour l'équipe, et un chef d'équipe chargé des rendus. Les rendus se font sur le e-learning

Modalités de rendu

Le chef d'équipe déposera le **MVP mardi soir**, puis la **V1 jeudi soir**.

Lors de la soutenance finale, il faudra fournir **un document** (quelques pages) présentant l'équipe, le schéma MCD, le schéma de la base et la liste des end-points utilisés dans l'application.

Évaluation

Une soutenance notée de 10 minutes aura lieu le vendredi après midi, avec présentation de votre solution, et une note sur les différents rendus.

La partie front et la présentation NE SONT PAS EVALUÉES DANS CE WORKSHOP, seul le back m'intéresse.

Ce qui est attendu :

- au moins 3 tables, et une association plusieurs à plusieurs
- une API REST
- Des routes claires, un MVC propre
- Un MCD fourni (dans le MVP puis au final si il a changé)
- Un lien sur le git, pour vérifier la bonne répartition du travail

Best practices du git

<https://webpick.info/les-bonnes-pratiques-de-git-pour-un-developpeur/>

<https://pierrebelin.fr/blog/guide-complet-ameliorer-son-process-git/>