

Web Serveur - L2

Votre client, après avoir trouvé beaucoup trop de (mauvaises) excuses pour la gestion chaotique de son organisation, a décidé de faire appel à vous pour développer une application web collaborative de gestion de tâches. Des discussions interminables avec lui ont permis d'établir un cahier des charges.

Cahier des charges

- Une tâche comporte un **titre**, , un **statut** (*accomplie ou à faire*) et peut avoir une **description**, une **liste de sous-tâches**, **une liste de personnes** et **une liste d'équipes rattachées**.
- L'application doit être **multi-utilisateurs** et supporter *l'authentification et l'enregistrement*.
- Un-e utilisateur/trice est une personne qui a un **nom**, peut avoir une **description** et une **équipe**. Elle s'authentifie avec son **e-mail** et son **mot de passe**.
- Une équipe a un **nom** et une **liste de personnes** membres de l'équipe, elle peut être *créeée, rejoindre ou quittée* par une personne.
- Une personne peut *ajouter, modifier ou supprimer* une tâche.
- Une tâche *peut-être assignée à une ou plusieurs personnes/équipe*, elle peut également être *désassignnée ou réassignée*.
- Une tâche peut être *privée* : visible seulement par la personne qui a créé la tâche et les personnes/ équipes qui y ont été associées ; ou *publique* : visible par tout le monde.
- Une personne peut voir le profil d'une autre personne/équipe ainsi que les tâches publiques qui lui ont été assignées.

Quelques conseils

Avant de se mettre à développer

- Lister et détailler les entités dont vous allez avoir besoin.
- Établir routes de votre application.
- Préparer les différentes vues de votre application.

Outils

Il existe plusieurs framework MVC en fonction de votre préférence de language, dans ce TP nous utiliserons **Django** qui est écrit en Python ([documentation](#)).

Pour démarrer

Dans le dossier de votre projet, il faudra d'abord **installer Django** dans un **environnement virtuel**.

```
python -m venv tp-env
# UNIX
source tp-env/bin/activate
# Windows
tp-env\Scripts\activate
python -m pip install Django
```

Ensuite il faut créer un projet et une application

```
django-admin startproject task_manager .
python manage.py startapp tasks
```

Vous pouvez vous inspirer du [tutoriel](#) pour les premières application.

Travailler en groupe

C'est également une bonne idée d'utiliser git, d'autant plus que vous pouvez vous faire un compte sur l'instance [Gitlab de l'ISIMA](#). Cela vous permet de travailler plus facilement en groupe et de sauvegarder votre travail.

Autre

- Lisez bien le sujet et n'hésitez pas à [me poser des questions](#) ;
- Il n'y a pas qu'une solution d'implémentation pour ce TP ;
- Je préfère corriger votre code que celui de ChatGPT et ses homologues.

Modalités de rendu

- Le TP se fait en *binôme* de la *même séance* ;
- Le rendu se fait sur l'espace dédié Moodle en déposant :
 - Le code du projet (vous pouvez également donner un lien fonctionnel d'un dépôt git accessible);
 - Un fichier `README.md` ou un autre document décrivant le fonctionnement de votre application (routes, etc.);
 - Les noms, prénoms et groupes de votre binôme ;

Barème général (à titre indicatif)

1. Fonctionnalités (8 points) ;
2. Interface utilisateur (4 points) ;
3. Qualité du code (4 points) ;
4. Documentation et présentation (2 points) ;
5. Respect général des consignes (2 points).