	Université de Corse - Pasquale PAOLI
	***Diplôme :* Licence SPI 2ème année parcours info** — **2023-2024**
	Module : Programmation Web Back End **TP N°3: Organisation MVC** **Enseignants : Evelyne Vittori- Ghinevra Comiti**

Ce TP n'est pas à rendre, il sera corrigé de manière collective pendant la séance du 14 mars.
Son objectif est la mise en pratique des concepts étudiés dans le cadre du CH2 du cours.

La façon dont nous avons réalisé le code PHP lors des TP1 et TP2 présente un certain nombre d'inconvénients :

- *Maintenance et réutilisation difficile* : les fichiers PHP ne sont pas structurés, du code est souvent répété d'un fichier à l'autre.
- *Développement difficile* : le code de l'application est mélangé avec le code de présentation et le code d'accès aux données. Ainsi, un développeur *backend* et un développeur *front-end* sont contraints de travailler sur le même fichier.
- *Sécurité :* les mots de passes d'accès à la base de données sont stockés dans un fichier dans un dossier publiquement accessible.

Afin d'y remédier, les applications web peuvent être structurées à l'aide de l'architecture modèle-vue-contrôleur (MVC). Celle-ci permet de séparer les couches de traitement, d'accès aux données et de présentation dans des fichiers distincts. La page 15 du diaporama CH2-Organisation du code permet d'illustrer les interactions entre ces différentes couches logiques dans le cadre d'une requête HTTP :

1. La requête de l'utilisateur est interprétée par le **contrôleur**. Celui-ci est responsable de vérifier les paramètres associés à une requête, de vérifier si l'utilisateur dispose des droits pour accéder à une page, etc.
2. Pour manipuler les données, le contrôleur utilise les services offerts par les **modèles**. Un modèle permet d'encapsuler des données (en RAM) et de jouer le rôle d'interface d'accès aux bases de données, fichiers, et services, où sont stockées les données de façon persistante.
3. Selon les opérations demandées par le contrôleur au modèle, celui-ci cherche à synchroniser les données en mémoire avec la base de données, à des fins de création, de récupération, de mise à jour, ou de suppression.
4. À l'aide des données encapsulées par les modèles, les contrôleurs sont en mesure de préparer la réponse à l'aide des vues.
5. Le contrôleur génère la **vue** adaptée à la requête de l'utilisateur, dont la construction dépend de variables que le contrôleur récupère depuis les modèles.
6. Enfin, le contrôleur effectue sa réponse à l'utilisateur en renvoyant la vue générée.

Dans ce TP, on vous demande de **reprendre le code de votre TP1 (ou choisir le code d'un autre étudiant sur Teams si vous n'avez pas pu finir le TP) et d'effectuer un refactoring** afin d'organiser votre code selon le modèle MVC (fonctionnel).
Cette ré-organisation devra être similaire à celle proposée dans l'exemple ex_mvcCH2 disponible sur l'ENT et expliquée dans le CH2 du cours.