420-05C-FX **Développement application web** **Philippe et Imane[1]** **Automne 2023**	**TP 1** **Troqueur**

Modalités

- Ce travail compte pour 20% de votre note finale.
- Ce travail doit être fait individuellement.

Remise

Voir la date de remise sur LÉA.

Objectif

Construire un site permettant de faire des échanges d'objets avec Flask. Le thème et la mise en forme sont laissés au choix. Cette première version sera rudimentaire et ne permettra pas l'échange. Nous l'améliorerons dans la deuxième partie.

Base de données

1. Dans **phpMyAdmin**, vous devez créer une base de données nommée **tp1_troqueur** avec l'interclassement par défaut de MySQL.
2. Vous devez réutiliser l'utilisateur « garneau » créé en classe. Voici ses informations :
   - Utilisateur : garneau
   - Serveur : localhost
   - Mot de passe : qwerty123
   - Privilèges globaux : aucun
3. Vous devez lui donner les privilèges **Data** sur la BD **tp1_troqueur**.
4. Vous devez importer le script **tp1_troqueur.sql** pour créer les tables.
5. De plus, vous devez utiliser le fichier **bd.py** fourni en classe.

Fonctionnalités à développer

Sur toutes les pages

- Entête avec le nom de votre site cliquable, permettant de revenir à l'accueil.

[1] Adapté de Nicolas Richard

- Chaque route doit avoir un <title> avec | NomDuSite
- Un menu principal fait avec Bootstrap:
    - Avec des liens pour :
        - l'accueil;
        - l'ajout d'un objet;
        - la liste de tous les objets.
- Spécifier ses paramètres régionaux, entre en_US, en_CA et fr_CA.
    - Par défaut, votre site est en fr_CA.
- Un pied de page avec le nom de l'auteur et un copyright.
- Un design fluide, en fonction de la largeur d'écran disponible, sans *@media query*.

Page d'accueil

- Doit afficher les cinq derniers articles créés selon la date de création (décroissant).
- Chaque affichage d'un objet doit avoir la photo, le titre, ainsi qu'un lien vers sa page de détails.

Page de détails d'un objet

- Doit présenter toutes les informations d'un objet.
    - Doit afficher le titre, le description ainsi que la date de création.
        - La date doit être affichée en fonction des paramètres régionaux.
    - Doit afficher une photo.
        - La photo doit avoir être dans un dossier sous **/static/images/objets/** et son nom doit être unique.
        - C'est possible qu'un objet n'ait pas d'image. On affiche alors une image par défaut.
- Doit offrir un lien pour éditer l'objet.

Ajout/Modification d'un objet

- Doit permettre de spécifier le titre, la description , la photo et la catégorie de l'objet.

Validation des données

- Vous devez retirer les espaces blancs inutiles au début et à la fin des champs de saisie.
- Les messages d'erreur doivent être proche des champs en erreur.
- Aucun champ texte ne doit accepter des balises HTML : `<(.*)>.*?|<(.*) />`

Catégories

- Vous n'avez pas à faire de formulaire pour les catégories. Vous remplirez la base de données avec des catégories directement dans votre outil de gestion de base de données.

Objet

Champ	Contrainte(s)
Titre	Entre 1 et 50 caractères alphabétiques.
Description	Entre 5 et 2000 caractères.
Photo	Entre 6 et 50 caractères alphanumériques, incluant le tiret et le point. Par exemple a.jpg
Catégorie	Une référence à la table catégories

Code HTTP et pages d'erreur

Votre application doit avoir des pages d'erreur pour codes HTTP suivants :

Code HTTP	Tests
400	Paramètre manquant dans l'URL
404	Accès à une route inexistante Détails d'un objet inexistant Modification d'un objet inexistant
500	Erreur en lien avec la BD

Contraintes

- Vous devez appliquer toutes les bonnes pratiques vues dans vos cours 420-05B-FX et 420-05C-FC, qui s'appliquent dans ce travail. On pense, entre autres, aux bonnes pratiques de design web.
- Vous ne pouvez pas modifier la structure de la BD.
- Tous les formulaires modifiants la BD doivent utiliser l'approche Post-Redirect-Get.
- Pour GIT :
    - Tout le développement de votre site web doit être hébergé sur un dépôt GIT privé sur le service GitHub Vous devez ajouter l'utilisateur **mtimane (https://github.com/mtimane)**.
- Pour votre design graphique original :
    - Vous devez utiliser SASS pour produire le fichier CSS.
    - Vous devez utiliser le fichier du thème officiel de Bootstrap ou un de ceux de Bootswatch.
    - Toutes vos pages doivent être dérivée d'un seul modèle (*template*).
- Toutes les pages HTML générées doivent être valide d'après **Total Validator** avec le niveau **HTML 5**.
- Vous devez utiliser uniquement les modules Python installés en classe.
- Aucune programmation Javascript, hormis l'inclusion des scripts JS de Bootstrap.
- Tous les pages web produites doivent s'afficher correctement avec les navigateurs **Firefox** et **Chrome**.

À remettre

- Dans une archive Zip, vous devez remettre :
    - Un document nommé **dépôt git** avec l'URL de votre dépôt GIT.
    - Une exportation de votre BD avec données.
    - Un dossier **site-web/** avec tout votre site web.
        - Ne remettre que les fichiers nécessaires à la publication de votre site sur Internet.
        - À supprimer :
            - Aucun fichier CSS : il sera généré à partir de votre fichier SASS tel que vu en classe.
            - Aucun fichier requis par Git (*.git/*, *.gitignore*, etc.) ni PyCharm (*.idea/*) ni *__pycache__*

Évaluation

Voici une ébauche du barème de correction. Il pourra être modifié lors de la correction.

Éléments	Points
Sur toutes les pages	10
Accueil	10
Gestion des objets (photo, choix de la catégorie, validations, etc.)	40
Qualité des interfaces	10
Utilisation d'un template de base	7
Gestion de l'affichage des dates	8
Code HTTP et pages d'erreur	7
Utilisation de GIT	3
Respect des critères de remise	3
Pénalités • Retard • Problèmes d'exécution • Français (-1 pt par faute dans l'interface)	