

Module : M202 – Approche Agile

Filière : Développement digital – option web full stack

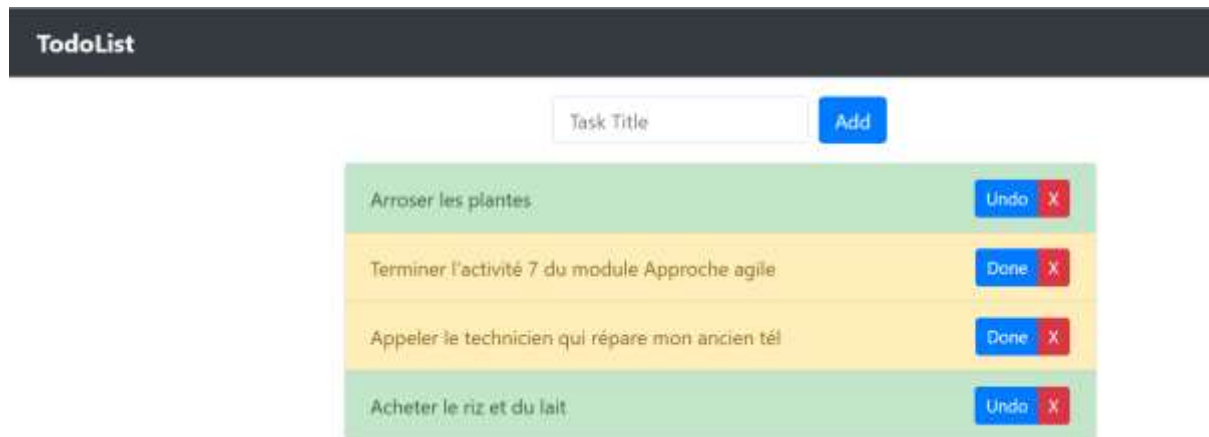
Activité 7

Objectifs :

- Manipulation de dépôts avec Git/GitLab

TP réalisé par binôme :

On souhaite créer une application « todo » : une application web permettant de gérer les tâches personnelles à faire, comme le montre la capture suivante :



Cette application est en php/Mysql et elle doit être réalisée en binôme, en respectant les consignes suivantes :

- Le projet doit être géré et enregistré dans GitLab.
- L'un des stagiaires sera chargé par la réalisation de la partie front end et l'autre développera la partie back end.

Git / GitLab :

1. Créer un nouveau projet gitlab ;
2. A l'aide de git local, lancez une commande « **clone** » pour cloner le répertoire distant dans un répertoire local de votre choix ;
3. Ajouter un fichier index.php :

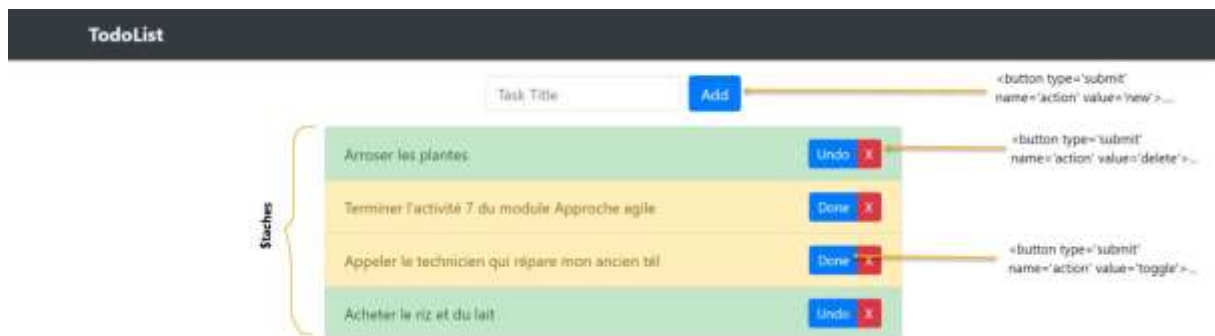
En haut de ce fichier, ajouter un ensemble de constantes définissant la configuration de la base de données :

```
// Database connection parameters.  
define('DB_USER', 'root');  
define('DB_PASS', '');
```

```
define('DB_NAME', 'todolist');
define('DB_HOST', '127.0.0.1');
define('DB_PORT', '3306');
```

4. Insérer par la suite, dans le même document index, la structure html ;
5. Ajouter et valider les modifications avec git ;
6. Apporter les modifications du fichier index.php au répertoire central à l'aide de la commande *push* ;

Réalisation du projet :



▪ Le développeur back-end:

- a. Il est amené à lire la liste des tâches enregistrées dans la table « todo » et les enregistrer sous forme d'une liste des objets dans une variable appelée « \$taches ». *(Cette liste doit être triée par date de création du plus récent au plus ancien)*
- b. Selon l'action reçue par la page index (une variable reçue en post), le développeur back end doit écrire les requêtes correspondantes, tel que :
 - **new** : sert à ajouter une nouvelle tâche à la table todo ;
 - **delete** : sert à supprimer la tâche de la table todo ;
 - **toggle** : sert à basculer la valeur du champ « done » entre true et false de la tâche en question.

Indication : pour faire ceci, vous pouvez utiliser :
update todo set done=1-done where id=...

A noter :

- Tous les formulaires envoient leurs valeurs avec la méthode « post »
- La tâche à traiter a toujours la valeur « id » comme name.
- Le code de cette partie doit être écrit en haut de la page index.

▪ Le développeur front-end:

En utilisant Bootstrap :

- a. Ajouter un navbar à la page index.php ;

- b. Ajouter un formulaire d'ajout d'une nouvelle tâche, tel que le « name » de la zone de texte est 'title' :

Add

- c. Ajouter en bas, une liste des tâches, en se basant sur la variable *\$taches* ; cette variable sera remplie par la liste provenant de la table todo et c'est le développeur back-end qui s'en charge.

- Afficher les tâches effectuées en utilisant cette classe : `list-group-item-success`
- Et les tâches qui ne sont pas encore réalisées en : `list-group-item-warning`

Arroser les plantes

Terminer l'activité 7 du module Approche agile

Appeler le technicien qui répare mon ancien tél

Acheter le riz et du lait

- d. Ajouter pour chaque tâche, un formulaire se composant de :
- Un input de type hidden contenant l'id de la tâche courante.
 - Deux boutons de type submit ; les deux auront le même name « action » et auront des valeurs différentes : toggle et delete.

Done X

Remarques :

- Chaque développeur doit créer sa branche et y travailler.
- A la fin de chaque question, testez vos travaux et lancez les commandes :
 - ✓ **git fetch** : pour télécharger les dernières modifications
 - ✓ **git diff origin/main** ou pointez sur la branche distante avec **git checkout origin/main** pour examiner les différences.
 - ✓ **git merge origin/main** : pour appliquer les changements au répertoire de travail.

✓ **git push -u origin nom_branche** : pour déposer les modifications dans le répertoire distant.

- L'un doit envoyer et traiter les merges request de l'autre.

Script de la base de données :

```
CREATE DATABASE IF NOT EXISTS todolist;
USE todolist;
CREATE TABLE IF NOT EXISTS `todo` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `title` varchar(2048) NOT NULL,
  `done` tinyint(1) NOT NULL DEFAULT '0',
  `created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`)
);
```