

# Énoncé de Projet : Application Web de Gestion de Liste de Tâches (To-Do List)

## Contexte :

Dans le cadre de votre apprentissage en développement web, vous allez concevoir et implémenter une application web de gestion de tâches (To-Do List). Cette application permettra aux utilisateurs de créer, visualiser, marquer comme terminées et supprimer des tâches. Le projet devra intégrer les technologies modernes de développement web, offrant une interface conviviale et des fonctionnalités robustes.

## Objectifs :

1. Concevoir une interface utilisateur intuitive et réactive en utilisant HTML, CSS, et JavaScript.
2. Implémenter une couche backend en PHP pour la gestion des données.
3. Utiliser une base de données MySQL pour persister les tâches.
4. Déployer l'application sur un serveur local ou distant pour la rendre accessible.

## Consignes :

### I. Analyse et Conception

#### 1. Description Fonctionnelle :

- Les utilisateurs doivent pouvoir ajouter une tâche avec un titre.
- Les tâches doivent être listées dans l'ordre d'ajout.
- Les utilisateurs doivent pouvoir marquer une tâche comme terminée (et l'indiquer visuellement).
- Les utilisateurs doivent pouvoir supprimer une tâche.
- (Optionnel) Ajouter une fonctionnalité de filtre (tâches terminées, non terminées, toutes).

#### 2. Description Technique :

- Utilisez HTML5 pour structurer la page.
- Stylez l'interface avec CSS3 pour rendre l'application attrayante.
- Gérez les interactions utilisateur avec JavaScript.
- Implémentez un backend en PHP pour gérer les tâches.
- Stockez les tâches dans une base de données MySQL.

### II. Réalisation Technique

#### 1. Frontend :

- Créez un formulaire d'ajout de tâche.
- Affichez les tâches dans une liste.

- Utilisez JavaScript pour gérer les interactions (ajout, suppression, marquage comme terminé).

## 2. Backend :

- Implémentez un script PHP pour recevoir les requêtes (ajout, suppression, récupération des tâches).
- Gérez les données dans une base de données MySQL.

## 3. Base de Données :

Créez une base `todolist` avec une table `tasks` :

```
```sql
CREATE TABLE tasks (
  id INT AUTO_INCREMENT PRIMARY KEY,
  title VARCHAR(255) NOT NULL,
  completed BOOLEAN DEFAULT FALSE
);
```
```

## 4. Communication Frontend-Backend :

- Utilisez fetch() ou AJAX pour interagir avec l'API PHP.

## III. Test et Validation

- Testez les fonctionnalités suivantes :
  - Ajout de tâches.
  - Marquage des tâches comme terminées.
  - Suppression de tâches.
  - Affichage correct des tâches (toutes et filtrées).
- Assurez-vous que l'application fonctionne sur différents navigateurs (Chrome, Firefox, etc.).

## IV. Déploiement

- Déployez l'application sur un serveur local (WAMP, XAMPP) ou un hébergement distant.
- Testez le fonctionnement en ligne et assurez-vous que toutes les fonctionnalités sont accessibles.

## Livrables :

1. **Code Source** : Organisé en dossiers (frontend/, backend/, etc.).
2. **Documentation** :
  - Explication des choix techniques (HTML, CSS, JavaScript, PHP, MySQL).
  - Guide d'installation et d'utilisation.
3. **Base de Données** : Fournir un script SQL pour recréer la base de données.
4. (Optionnel) **Vidéo Démonstration** : Montrer les fonctionnalités principales en action.

## Critères d'Évaluation :

### 1. Fonctionnalités :

- Ajout, suppression, et modification des tâches.
- Persistance des données avec MySQL.

### 2. Interface Utilisateur :

- Esthétique, ergonomie et réactivité.

### 3. Code :

- Structure claire, bien commentée et lisible.

### 4. Documentation :

- Instructions d'installation et d'utilisation précises.

### 5. Test :

- Fonctionnement sans bug.
- Adaptabilité à différents navigateurs.