

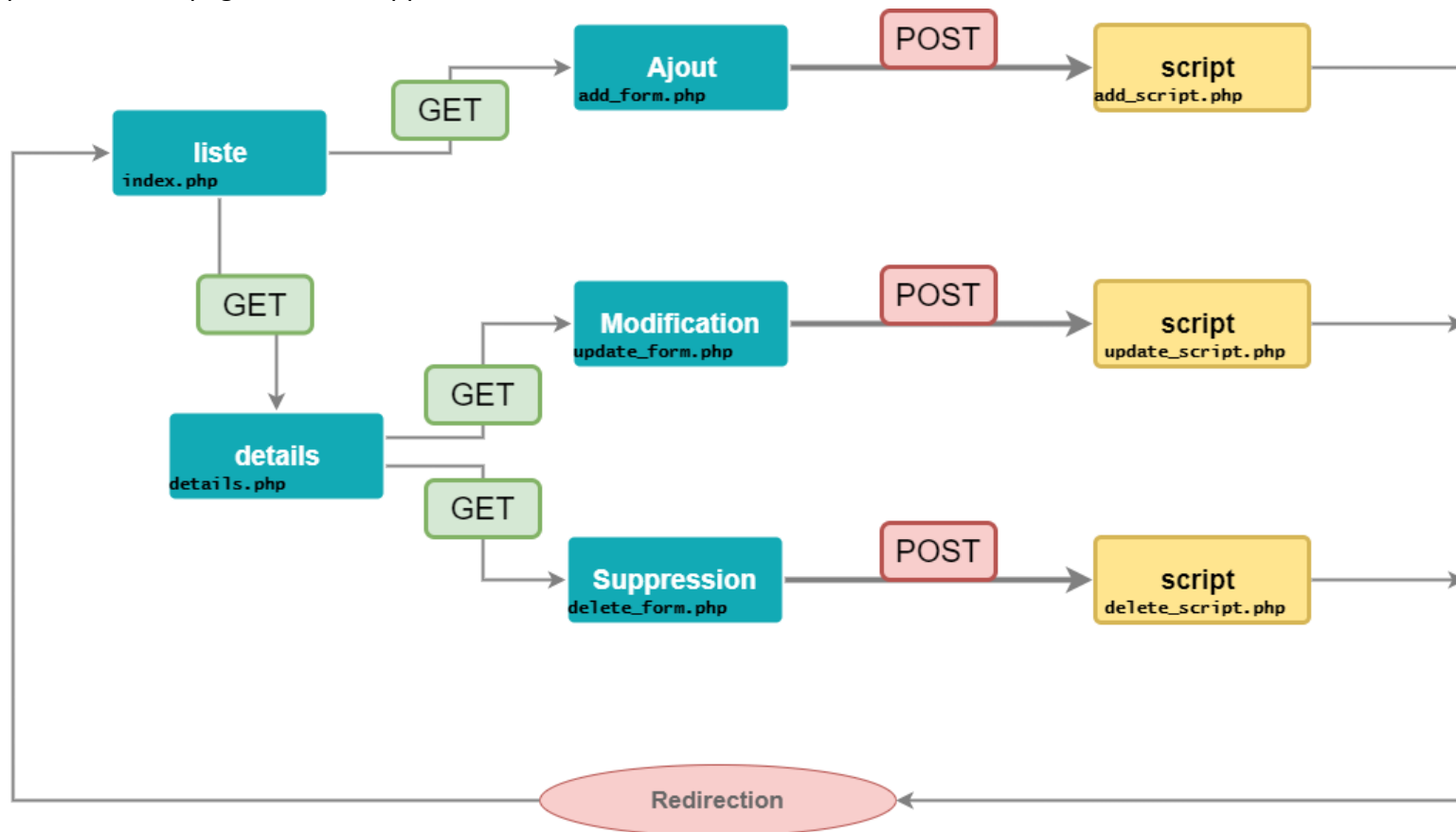
Jarditou : cahier des charges

Objectifs

Dans ce cours, vous avez appris à extraire des données provenant d'une base, en ajouter, en supprimer et en modifier. Vous allez pouvoir maintenant construire un projet avec tous les éléments nécessaires à la gestion d'un enregistrement :

- Afficher tous les enregistrements d'une table : page de liste (qui sera aussi la page d'accueil),
- Afficher le détail d'un enregistrement,
- Saisir un nouvel enregistrement : formulaire d'ajout

Voici le schéma présentant les pages de cette application :



Tout cela doit bien sûr être consultable sur mobile et agréable à l'oeil !

La page de liste

La première page, ci-dessous, doit afficher la liste des enregistrements se trouvant dans la table produits.

Le libellé du produit doit être cliquable pour permettre d'afficher le formulaire de détail. Ce lien doit transmettre en paramètre l'identifiant (clé primaire) de l'enregistrement souhaité :

```
<a href="detail.php?pro_id=<?php echo $row->pro_id; ?>" title="Modifier">Modifier</a>
```

La page de liste doit proposer également un lien vers le formulaire d'ajout.

La page de détail

Elle doit afficher toutes les informations présentes dans la table produits, avec donc une requête SELECT.

- Un lien Modifier permet d'afficher le formulaire de modification ci-dessous :

Cette page doit afficher tout le détail d'un produit, la clé primaire du produit est transmise dans l'url via la méthode GET (lien présent sur le libelle du produit dans la page de liste) :

```
<a href="detail.php?pro_id=<?php echo $row->pro_id; ?>" title="Modifier">Modifier</a>
```

Le formulaire d'ajout

Il doit permettre de saisir toutes les informations présentes dans la table produits, c'est-à-dire chaque colonne. Attention, pour que la requête d'insertion fonctionne, il faut que les valeurs saisies correspondent aux types des colonnes.

Pour la catégorie et la photo, la saisie doit se faire pour le moment dans un champ texte normal (entrez un entier pour la catégorie et 'jpg' pour les photos), nous verrons plus tard la bonne méthode.

Vous devez donc créer un formulaire et l'appeler par exemple *_formulaireajout.html*. Un clic sur le bouton *Envoyer* déclenche l'appel de la page spécifiée dans l'attribut action, par exemple *_scriptajout.php*.

Ce script doit effectuer les actions suivantes :

1. Récupération des informations transmises par le formulaire.
 2. Enregistrement des données dans la base (à l'aide d'une requête INSERT)
 3. La date de modification ne doit pas être renseignée (valeur NULL en base) puisqu'il n'y a jamais eu de modification.
 4. Redirection vers la page de liste. Cette redirection s'effectue à l'aide de la fonction PHP `header("Location:liste.php");`
-

Le formulaire de modification

Il doit permettre de modifier **toutes** les informations présentes dans la table produits.

Vous devez donc créer un formulaire et l'appeler par exemple *formulaire_modif.html*. Un clic sur le bouton *Envoyer* déclenche l'appel de la page spécifiée dans l'attribut action, par exemple *_scriptmodif.php*.

Ce script doit effectuer les actions suivantes :

1. Récupération des informations transmises par le formulaire.
2. Enregistrement des données dans la base (à l'aide d'une requête UPDATE).
3. La date d'ajout ne doit pas être modifiée, par contre cette fois il faut renseigner la date de modification.
4. Redirection vers la page de liste.

La suppression

Aucun formulaire nécessaire, mais un lien placé sur la page de détail qui au clic demande un message de confirmation Javascript.

- Si on confirme, la page de destination de ce lien effectue une requête DELETE. Le lien doit donc transmettre la clé primaire de l'enregistrement à supprimer,
- Si on annule, l'enregistrement ne doit pas être supprimé,
- Dans les 2 cas, redirection vers la page de liste.

Contrôler les données saisies par l'utilisateur

Vous pouvez appliquer le contrôle des saisies en Javascript, en termes d'ergonomie l'utilisateur voit les erreurs en temps réel. Mais n'oubliez pas que du point de vue sécurité, le Javascript peut être désactivé dans le navigateur, la bonne pratique est donc de **toujours filtrer les saisies côté serveur**, dans notre cas en PHP.

Sécuriser les requêtes SQL

N'oubliez pas de contrôler le contenu des champs envoyés par l'utilisateur.

Et utilisez la technique des requêtes préparées.

Choix d'une catégorie

Dans la base de données, une contrainte de clé étrangère a été définie sur la colonne `pro_cat_id` de la table `produits`. Cela signifie qu'il est impossible de classer un produit dans une catégorie qui n'existerait pas.

Pour éviter ce problème, il faut donc afficher dans les formulaires d'ajout et de modification les catégories existantes dans une liste déroulante après avoir fait une requête SQL pour les extraire.

Dans chaque balise `<option>`, l'attribut `value` correspondra à la clé primaire de la table `categories` et le texte affiché à l'utilisateur sera le nom de la catégorie. Lorsque le formulaire sera posté, le numéro de catégorie sera transmis et on pourra le stocker dans la colonne `pro_cat_id` de la table `produits`.

Chargement d'une photo

[Lire cette ressource.](#)

Mettre en œuvre le téléchargement de fichier dans le formulaire d'ajout pour pouvoir charger la photo d'un produit.

Consignes :

- Respectez la charte de nommage "`pro_id.extension`".
- Ne pas mettre en œuvre les notions de droits ni d'API Drag & Drop.