

CAHIER DES CHARGES

Gestion de Portfolio de Projets

1. Présentation du projet

Ce projet consiste à développer une application web permettant de gérer un portfolio de projets personnels. L'application permettra d'afficher, d'ajouter, de modifier et de supprimer des projets via une interface web.

2. Objectifs pédagogiques

Ce projet vous permettra de mettre en pratique :

- L'utilisation d'une base de données relationnelle avec MySQL
- L'implémentation complète d'un CRUD (Create, Read, Update, Delete)
- La manipulation de données avec PHP et SQL
- L'intégration HTML/CSS pour une interface utilisateur fonctionnelle
- La gestion des formulaires et la validation des données
- La sécurisation des requêtes SQL (prévention des injections SQL)
- Une protection contre la faille XSS

3. Fonctionnalités attendues



3.1. Affichage des projets (READ)

La page d'accueil doit afficher la liste de tous les projets sous forme de blocs (cards). Chaque base de données bloc doit contenir :


- Le titre du projet (cliquable et renvoie vers le projet en détail)
- Une description courte (maximum 200 caractères)

3.2. Affichage d'un projets (READ)

Lorsqu'on clique sur le titre du projet, on arrive sur cette page qui donne les détails du projet.

- Le titre du projet
- Une description courte (maximum 1000 caractères)
- La date de création
- Auteur du projet
- Un lien vers le projet (URL GitHub, site en ligne, etc.)
- Des boutons "Modifier"  et "Supprimer" 

3.3. Ajout d'un projet (CREATE)


Sur la page d'accueil, on doit trouver un bouton "Ajouter"  qui conduit à un formulaire permettant d'ajouter un nouveau projet avec les champs suivants :

- Titre (obligatoire, texte, max 100 caractères)


- Description (obligatoire, texte, max 500 caractères)
- URL du projet (optionnel, format URL valide)
- Date de création (automatique)

Validation : Tous les champs obligatoires doivent être remplis avant l'enregistrement.

3.4. Modification d'un projet (UPDATE)

En cliquant sur le bouton "Modifier" , l'utilisateur accède à un formulaire pré-rempli avec les données du projet sélectionné. Il peut modifier tous les champs et enregistrer les modifications.

3.5. Suppression d'un projet (DELETE)

En cliquant sur le bouton "Supprimer" , L'utilisateur est redirigé vers une page qui demande la confirmation de la suppression du projet en rappelant son titre.

4. Spécifications techniques

4.1. Structure des fichiers

Organisation recommandée du projet :

- index.php : Page d'accueil affichant la liste des projets
- show.php : Page du projet en détail
- add.php : Formulaire d'ajout d'un projet
- update.php : Formulaire de modification d'un projet
- delete.php : Script de suppression d'un projet

4.2. Sécurité

- Utiliser des requêtes préparées (PDO) pour toutes les interactions avec la base de données
- Valider et filtrer toutes les entrées utilisateur
- Gérer les erreurs sans exposer d'informations sensibles
- Afficher les données utilisateur en vous protégeant de la faille XSS.

5. Contraintes techniques

- Utilisation obligatoire de PHP natif (pas de framework)
- base de données MySQL
- CSS personnalisé, **libre** et responsive (framework CSS autorisé)
- Code commenté et indenté correctement
- Respect des bonnes pratiques de nommage (variables, fonctions, fichiers)

6. Livrables

- Code source complet du projet (fichiers PHP, CSS, SQL)
- Documentation utilisateur expliquant comment installer et utiliser l'application
- Fichier README.md avec description du projet, instructions d'installation, lien GitHub et lien Jira

7. Bonus

- Système de recherche par mots-clés
- Mettre en place un système de connexion. Seuls les utilisateurs connectés pourront créer, modifier et supprimer un projet
- Afficher, dans chaque carte projet, une image téléchargeable lors de la création d'un projet via un `<input type="file">`

8. Critères d'évaluation

Critère	Description
HTML/CSS/Responsive	Structure correcte, respect des types et contraintes
CREATE	Formulaire fonctionnel, validation, insertion en BDD
READ	Affichage correct de tous les projets
UPDATE	Modification fonctionnelle avec formulaire pré-rempli
DELETE	Suppression avec confirmation
Sécurité	Requêtes préparées, validation et affichage des données
Gestion de projet	Jira / GitHub (commits et branches)
Interface	Design cohérent et ergonomique

9. Gestion de projet

Ticket-0 : Conception de la base de données

- Lister les tables nécessaires
- Lister les champs pour chaque table
- Lister les propriétés de chaque champ (INT, VARCHAR(1000), TEXT, DATE, NOT NULL, AUTO_INCREMENT, ...)
- Créer les tables dans phpMyAdmin

Ticket-1 : mise en place de l'environnement de travail

- Prise en main du Kanban
- Mise en place du repository
- Mise en place de l'espace de travail et de l'architecture de dossier
- Lancement du docker et connexion à la BDD

Ticket-2 : squelette de la page d'accueil

- Implementation du bloc header
- Implémentation du bloc main : vide
- Implementation du bloc footer (optionnel)
- Implémentation du style CSS général de la page (Responsive mobile first)
- Refactorisation

Ticket-3 : bloc main

- Affichage des projets avec un foreach
- Implémentation du style CSS du bloc main
- Refactorisation

Ticket-4 : page add.php

- Création du formulaire HTML
- Gestion de la soumission du formulaire
- Style CSS
- Refactorisation

Ticket-5 : page update.php

- Création du formulaire HTML
- Gestion de la soumission du formulaire
- Style CSS
- Refactorisation

Ticket-6 : page delete.php

- Création du formulaire HTML
- Gestion de la soumission du formulaire
- Style CSS
- Refactorisation

Ticket-7 : page update.php

- Création du formulaire HTML
- Gestion de la soumission du formulaire
- Style CSS
- Refactorisation