

Rapport de Projet :

Moteur de Blog PHP/MVC

Binôme : Ciserane Marius & Laporte Valentin

Contexte : Projet de développement Web (11 Janvier 2026)

Sujet : Création d'un blog complet respectant l'architecture MVC

1. Organisation et Environnement de Travail

Pour mener à bien ce projet, nous avons mis en place un environnement de développement partagé et des outils de communication adaptés.

- **Gestion de Version (VCS)** : Nous avons utilisé **GitLab** pour centraliser notre code. Cela nous a permis de travailler simultanément sur différents fichiers sans écraser le travail de l'autre et de garder un historique propre de nos modifications.
 - **Environnement Serveur** : Nous avons uniformisé nos environnements locaux en utilisant tous les deux **XAMPP** (Apache/MySQL), ce qui a évité les problèmes de compatibilité lors des fusions de code.
 - **Éditeurs de Code (IDE)** : Chaque membre a utilisé l'outil avec lequel il était le plus performant :
 - **Marius** : Visual Studio Code (VSCode).
 - **Valentin** : PhpStorm.
 - **Communication** : Les échanges ont été réguliers tout au long du projet, nous permettant de synchroniser nos avancées et de résoudre les blocages rapidement.
-

2. Répartition du Travail

Nous avons opté pour une répartition basée sur nos affinités techniques (Frontend vs Backend), tout en conservant une flexibilité nécessaire pour un travail en binôme.

Phase Commune : L'Architecture

Le projet a débuté par une phase de **Pair Programming** indispensable. Nous avons construit **ensemble** la structure MVC (les dossiers, l'autoloader, le routeur `index.php`). C'était crucial pour que chacun comprenne comment l'information circule avant de travailler sur ses tâches respectives.

Marius Ciserane : Focus Frontend & Intégration

- **Responsabilités principales** : Intégration des vues avec **Twig**, design global (CSS, Bootstrap), et interactivité en **JavaScript**.
- **Polyvalence** : Marius a également codé certains Modèles et Contrôleurs nécessaires à l'affichage des données (notamment pour le Dashboard).

Valentin Laporte : Focus Backend & Logique

- **Responsabilités principales** : Développement de la logique **PHP**, structure de la base de données, et implémentation du cœur fonctionnel.
 - **Polyvalence** : Valentin a aussi travaillé sur certaines vues Twig pour s'assurer que les données complexes s'affichaient correctement.
-

3. Difficultés Rencontrées et Solutions

Chaque membre a fait face à des défis techniques spécifiques liés à sa partie :

La Gestion des Rôles (Valentin)

- **Le problème** : Mettre en place un système de droits granulaires (RBAC) ne se limitait pas à un simple champ "admin" dans la base de données. Il fallait gérer les relations entre Utilisateurs, Rôles et Permissions.
- **La solution** : Une réflexion approfondie sur le schéma SQL (tables de jointure `Role_User`, `Role_Permission`) et la création de méthodes PHP robustes pour vérifier si un utilisateur possède une permission spécifique avant d'exécuter une action.

La Gestion du "Dark Mode" (Marius)

- **Le problème** : Le défi n'était pas seulement de changer les couleurs, mais de faire en sorte que le choix de l'utilisateur persiste entre les pages et soit immédiat, sans "flash" blanc au chargement.
 - **La solution** : Utilisation avancée de JavaScript et des variables CSS (`var(--bg-color)`), couplée au stockage local (`localStorage`) pour mémoriser la préférence de l'utilisateur.
-

4. Choix Techniques

Les contraintes techniques de ce projet nous étaient **imposées** par le cahier des charges pédagogique. Nous n'avons pas choisi la stack, mais nous avons compris la pertinence de ces choix :

- **PHP (Sans Framework)**
 - **Architecture MVC**
 - **Twig**
 - **Design Patterns**
-

5. Bilan et Apprentissages

Ce projet a été une étape fondamentale dans notre formation. Au-delà de la note ou du résultat final, voici ce que nous en retenons :

1. **Compréhension du MVC** : Le concept, qui paraissait abstrait au début, est devenu concret. Nous savons maintenant pourquoi et comment séparer la logique de l'affichage.
2. **Design Patterns** : L'utilisation du **Singleton** (pour la connexion BDD et la Session) est désormais acquise. Nous avons compris l'intérêt d'avoir une instance unique accessible partout.
3. **Travail Collaboratif** : Nous avons réussi à combiner nos forces (Front/Back) pour livrer un projet complet, en surmontant les bugs grâce à une communication efficace.

En conclusion, ce moteur de blog est fonctionnel, sécurisé, et repose sur une architecture saine que nous pourrions faire évoluer ou réutiliser pour de futurs projets.