



Rapport technique de la réalisation d'un Blog en PHP Orienté Objet



1. Architecture MVC:

- *Modèle (Model):*
- *Vue (View):*
- *Contrôleur (Controller):*

2. Besoins Fonctionnels:

3. Base de Données "monblog":

4. Implémentation Technique:

5. Authentification:

6. Conclusion:



1. Architecture MVC:

Le modèle MVC a été choisi pour sa capacité à organiser le code de manière modulaire, favorisant la maintenabilité et la lisibilité. Les composants clés du projet sont répartis comme suit :

Modèle (Model):

- Base de données "monblog" comprenant les tables t_billet, t_commentaire et t_auteur.
- Classes PHP dédiées aux interactions avec la base de données.

Vue (View):

- Page d'authentification (VueAuthentification) liée à la table t_auteur.
- Page VueAccueil affichant la liste des billets depuis la table t_billet.
- Page VueBillet pour visualiser les billets et leurs commentaires.
- Page VueErreur pour la gestion des erreurs.

Contrôleur (Controller):

- Gestion des actions de l'utilisateur, liaison entre la vue et le modèle.
- Contrôleurs dédiés pour l'authentification, l'accueil, la gestion des billets, etc.

2. Besoins Fonctionnels:

Le projet répond aux besoins fonctionnels suivants :



- Authentification des utilisateurs via la page d'authentification.
- Accès à la liste des billets depuis la page VueAccueil.
- Possibilité d'ajouter un nouveau billet.
- Visualisation des commentaires et ajout de nouveaux commentaires depuis la page VueBillet.
- Gestion des erreurs via la page VueErreur.

3. Base de Données "monblog":

La base de données est constituée de trois tables :

- t_billet : Stocke les informations sur les billets.
- t_commentaire : Contient les commentaires associés aux billets.
- t_auteur : Utilisé pour l'authentification des utilisateurs.

4. Implémentation Technique:

- Utilisation de PHP pour le backend et le contrôleur.
- HTML, CSS, et JavaScript pour la partie frontend.
- Utilisation de classes PHP pour modéliser les entités (billet, commentaire, auteur) et les interactions avec la base de données.



5. Authentification:

Une page d'authentification permet aux utilisateurs d'accéder au blog après validation de leurs identifiants par rapport à la table t_auteur.

6. Conclusion:

Le projet a été réalisé en suivant les principes du design pattern MVC, assurant une séparation claire des préoccupations et facilitant la maintenance future. Les fonctionnalités du blog répondent aux besoins spécifiés, offrant une expérience utilisateur cohérente et intuitive.