**Reprise du projet « weblog »**Rapport du 24/05/2025 par Mikael Debray

Weblog v0 semble être un projet de publication d’articles qui n’a jamais abouti.

Il présente à son état initial des anomalies et un manque de fonctionnalités essentielles à son fonctionnement normal. Ce rapport présente de façon synthétique les changements apportés pour faire fonctionner ce site.

Technologies utilisées :  
  
php, html, css   
Ckeditor  
Mysql pour la base

**Vue d’ensemble du site dans son état initial (V0) :**

*\*voir pour un diagramme*

FRONT :

Le squelette du site est présent, le dossier includes contient les éléments de pages personnalisable et réutilisable facilement pour les parties admin et publique du site. Grâce aux inclusions on peut faire apparaitre le pied de page et la barre de navigation appropriée pour la catégorie d’utilisateur connectée sur le site et de façon facilement maintenable. Le header permet d’appliquer automatiquement le style sur la page pour l’administrateur et le visiteur grâce aux fichiers css admin et public styling contenu dans le fichier css.  
Le visuel de la page d’accueil est déjà présent avec sa bannière. Il y a aussi celui du tableau de bord de l’administrateur avec la page user et create post ainsi que le menu et le header. On dispose aussi du visuel de la page de login.

BACK

Le fichier de deconnexion

Base de données :

Le fichier weblog.sql permet de créer la base. Cette dernière est relativement bien construite mais elle est imparfaite.

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.  
  
Elle se compose des tables posts, users, topics, roles,

Et de la table de jointure post\_topics

En commentaire il est suggéré de créer la table de jointure supplémentaire role\_user pour joindre les id des tables users et roles en obtenant les colonnes suivantes :  
role\_user(id, user\_id,role\_id)

Certaines tables (post\_topic, roles, topics, users, posts) ont des champs id mais la configuration AUTO\_INCREMENT et la clé primaire n’est pas correctement définies.

D’autre commentaires laissent entendre que la création de tables d’archive des posts et users supprimées serait souhaités

**Points notables :**L’architecture des dossiers..

Il existe des fichiers de style css pour styliser le site via un header

L’id dans la table post\_topic n’est pas auto incrémentée.

**Composants absents :**La logique métier est pratiquement inexistante.

* Il n’y pas de système d’enregistrement pour l’utilisateur :  
   Register.php est vide
* Les fichiers admin\_function.php, post\_functions.php et all\_functions.php sont vide. La logique métier est donc quasiment inexistante. Il manque toutes les fonctionnalités liées à la publication des articles, la validation des articles par l’administrateur, leur edition
* Le renvoi d’erreur est inexistant, le fichier errors.php est vide.
* La page principale index.php présente un section « Articles Récent » mais elle est vide.
* Le fichier single\_post est vide, il doit permettre d’afficher l’article complet.
* La fichier filtered\_post est vide. Cette page devrait pouvoir montrer les articles et permettre de les filtrer par sujet voir par auteur.

**Sécurité :**

Le mot de passe est **hashé en MD5,** aujourdh’ui c’est une faille de sécurité car cela peut se calculer « facilement »

Pas de gestion des roles pour l’accès aux pages réservées à l’administrateur

Composant présent :

* La page user.php existe et propose une interface avec listing, création, édition et suppression mais la logique est absente
* Le fichier Create\_post possède déjà la logique d’upload d’image et d’édition. Elle fonctionne avec CKeditor qui est déjà présent dans le projet.  
  Cependant la logique d’enregistrement est à créer
* Un tableau de bord pour l’administrateur est présent, et dirige l’administrateur vers la page user et post.php uniquement. De plus l’incrémentation des nombres de post, d’user et de posts publié est inexistante.   
  Une image contenant texte, Police, ligne, capture d’écran

  Le contenu généré par l’IA peut être incorrect.

Anomalie :  
Il y a une coquille sur la constante root\_path dans le fichier config \_FILE\_

**Actions effectuées :**

Durant toute l’optimisation du site la charte graphique a été respectée en réutilisant les fichiers de style css déjà présent à l’état initial. 🡪 admin\_styling.css ou public\_styling.css

**Prérequis :** En premier lieu on crée la base via le fichier fournit weblog.sql

Il faut modifier la page config.php de façon à pouvoir se connecter à la base de données :   
Rentrer le nom de la base, le login et le mot de passe de connexion :

BASE : weblog  
Login : root  
mot de passe : aaaaaaaa

Les fonctions sont placées de façon à préserver la logique initiale dans admin\_functions.php All\_functions.php et post\_funciton.php.

* Correction de l’anomalie dans le root\_path du fichier config.php   
  \_FILE\_ 🡪 \_\_FILE\_\_  
  

Sécurité :   
Le mot de passe est hashé avec password\_hash ce qui assure le dernier algorithme de chiffrement et donc la préservation du secret du mot de passe.

Des placeholders sont utilisé pour les requêtes SQL afin de prévenir des injections.

Acces verrouillé aux articles non publiés. La page single post pour un article non publié ne s’affiche pas et si l’on tente d’y accéder directement avec son lien la page affiche « Article non trouvé ou non publié »

**1) Modification dans la base de données :**

* Les droits des utilisateurs passent désormais par la table role\_user :
  + Création d’une table de jointure role\_user
  + Suppression du champ role dans users
  + Migration des anciens rôles utilisateurs dans la nouvelle table role\_user
* Ajout (ou correction) de la clause AUTO\_INCREMENT et définition correcte des clés primaires pour chaque table.
* Ajout des index et des clés étrangères pour garantir l’intégrité et éviter les duplications.

**2) Ajout des composants :**

* La page d’accueil doit afficher les articles Récemment publiés, ajout d’une fonction **getPublishedPosts**() pour récupérer les articles publiés en commençant par les plus récent.
* Ajout du contenu pour les liens ancrés présent dans la navbar

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

* Affichage du détails du post en cliquant sur « Lire la suite »  
  Une image contenant texte, capture d’écran, plante

  Le contenu généré par l’IA peut être incorrect.
* Affichage des post pour permettre à l’admin de les valider ou de les refuser. Ajout de getAllPosts() et création de la page posts.php
  + Une image contenant texte, logiciel, capture d’écran, Police

    Le contenu généré par l’IA peut être incorrect.Possibilité de supprimer un post
  + Possibilité de modifier un post (voir encore les cas speciaux si temps)
* Création et édition de post dans create\_post.php
* Ajout d’une gestion pour les messages d’erreur dans errors.php
* Verrouillage des pages admin avec la vérification du rôle pour la session et renvoi automatique vers la page d’identification si l’utilisateur n’est pas admin.
* Un auteur ne peut modifier que ses post, et n’a pas accès aux pages de gestion utilisateur et de gestion des topics qui sont réservé aux admin.
* Page filtered\_post pour afficher tout les posts et pouvoir les filtrer
* L’administrateur peut valider ou non la publication d’un post
* Dans le dashboard,
  + La 3eme rubrique permet désormais d’acceder à la gestion des sujets
  + Le nombre réel d’articles, d’user et de sujets s’affiche et se met à jour

Notes :

Une image contenant texte, capture d’écran, Logiciel multimédia, logiciel

Le contenu généré par l’IA peut être incorrect.

On a cré un compte et ce dernier est enregistré dans la base :  


Note soutenance :

Commencer par montrer enregistrement et login, puis qu’il est impossible d’acceder aux pages de l’administrateur sans être admin