

# Évaluation du projet **SuperBlog**

*A lightweight CMS (blog) engine*

Vous trouverez dans ce document des informations relatives au projet dont l'objectif est de vous fournir les meilleures bases pour finaliser le projet que nous avons débuté ensemble.

La première chose à faire est de télécharger la dernière version de [LPB2024](#).

Actuellement, la dernière release est la 1.17.0. Rendez-vous dans la section PGMS. Vous trouverez à cet endroit la base du projet sous le nom de 10.superblog.

**Je vous conseille de lire ce document dans son entièreté une première fois et d'y revenir si besoin est.**

## A. Quelle est la situation actuelle de **SuperBlog** ?

Dans le répertoire 10.superblog se trouve toute la structure du projet avec tous les fichiers nécessaires au bon fonctionnement de l'application (voir le document : fil-conducteur-superblog.pdf). Vous trouverez dans cette structure un répertoire \_src/ à l'intérieur duquel se trouve deux fichiers SQL nécessaire à la création de la base de données (blog) dans laquelle nous gérons nos données :

- blog.sql : importez ce fichier pour créer ou réinitialiser la DB blog avec des mots de passe en clair pour les utilisateurs
- blog-pwd-hash.sql : importez ce fichier pour créer ou DB blog avec des mots de passe utilisateurs hashés.

Vous trouverez également ce document ainsi que le document appelé « fil conducteur ».

## B. Que faut-il faire pour remplir le contrat d'évaluation du projet **SuperBlog** ?

1. Il faut une page d'accueil (**index.php**) qui affiche les titres des articles **PUBLIES**. Articles enregistrés dans la table « articles » de notre DB blog.

2. Il faut, lorsque l'on clique sur le titre de l'article sur la page d'accueil, ouvrir une nouvelle page (**article.php**) qui va afficher l'article complètement, c.-à-d. Son titre et son contenu.
3. Il faut un formulaire d'identification (**login.php**). On utilise ce formulaire pour s'identifier sur l'application. On doit être identifié pour gérer le contenu (**manager.php**), ajouter du contenu (**add.php**), modifier du contenu (**edit.php**) et supprimer du contenu (**manager.php**). La balise action du formulaire appelle le fichier login.php lui-même et c'est dans le contrôleur au-dessus de la page que l'on effectue le processus d'identification. Une fois l'utilisateur identifié, ce dernier est redirigé vers la page de gestion du site (**manager.php**) à l'aide de la fonction PHP header()
4. Il faut une page de gestion (**manager.php**) qui va afficher les titres de tous les articles (publiés et non publié). En dessous du titre de chaque article, vous devez ajouter 3 liens / boutons. Un bouton « Modifier », un bouton « Lire », un bouton « Supprimer ».
5. Le bouton « Modifier » est un lien vers la page de modification (**edit.php**) et dans le lien, on passe l'ID de l'article à modifier. On teste et on récupère l'ID dans le contrôleur de la page de modifications et on charge les données de l'article correspondant à l'ID. On placera les données reçues dans le formulaire de la page de modification et on valide la modification en cliquant sur le bouton modifier en bas de page. On affiche un message en fonction du retour de la fonction de mise à jour.
6. Le bouton « Supprimer » est un lien qui vers la page de gestion (**manager.php**), page sur laquelle on se trouve. Dans le lien, on passe également l'ID de l'article à supprimer. On teste et on récupère l'ID et lance la fonction de suppression d'un article. On affiche un message en fonction du retour de la fonction de suppression.
7. Le bouton « Lire » est un lien identique à celui qui se trouve sur notre page d'accueil (**index.php**). C'est un lien vers la page qui affiche l'article au complet (**article.php**) avec l'ID de l'article en question.
8. La page de modification d'un article (**edit.php**). Si vous avez bien fait votre lien dans la page « **manager.php** » vous devez recevoir l'ID dans l'URL et à l'aide du \$\_GET['id'] vous récupérer les données de l'article en utilisant la

fonction `getArticleByIDDB()` et vous placez les données dans le formulaire. La balise action du formulaire envoie les données du formulaire vers la page **manager.php** et c'est dans le contrôleur de la page de gestion que vous vérifiez le formulaire que vous recevez et lancez la fonction de mise à jour. On affiche un message en fonction du retour de la fonction.

9. La page d'ajout d'un article (**add.php**) démarre avec un formulaire vierge que l'utilisateur va compléter pour ensuite envoyer son nouvel article (les données du formulaire) vers notre page de gestion (**manager.php**). Et vous commencez à avoir l'habitude maintenant, c'est dans le contrôleur de la page de gestion que l'on va vérifier le formulaire que l'on reçoit (rappelez-vous la balise `input hidden` en bas de formulaire qui possède un attribut `value` avec le nom du *form* (idem avec la page **edit.php**), pour ensuite lancer la fonction qui va ajouter un nouvel article dans la base de données. On vérifie le retour de la fonction pour afficher un petit message à l'utilisateur pour l'informer du résultat de l'ajout d'un nouvel article, en principe, une bonne nouvelle !
10. La page d'identification (**login.php**) est un simple formulaire avec un champ *email* et un champ *password*. L'attribut action du formulaire est initialisé sur **login.php**, on aurait pu le laisser vide, car vide = page sur laquelle on se trouve, autrement dit **login.php**. Dans le contrôleur en haut de la page, on fait une série de tests comme nous en avons l'habitude maintenant, y compris le fait de vérifier que l'utilisateur a bien rempli le champ email et le champ mot de passe, si c'est le cas, on lance la fonction d'identification. En fonction de la base de données que vous utilisez (mot de passe en clair ou mot de passe hashés) ou utilisera la fonction `userIdentificationDB()` ou `userIdentificationWithHashPwdDB()`. On vérifie le retour de la fonction. Si la fonction retourne un tableau avec les données de l'utilisateur, c'est que l'utilisateur est reconnu par le système et on redirige l'utilisateur vers la page de gestion (**manager.php**).
11. Le bouton de déconnexion dans le menu (en rouge) appelle le script `logout.php` et détruit la session en cours, mais en recrée une vierge tout de suite après et redirige vers la page d'accueil à la fin du script.

Je pense que nous avons fait le tour des pages de notre application.

## Les cadeaux de fin d'années :)

Ce que **vous n'êtes pas obligé de faire**, car déjà présent dans le code de la dernière mise à jour :

1. La page **index.php** : la page d'accueil du site, nous l'avons faite ensemble le lundi 18/12/2023
2. La page **article.php** : la page qui affiche l'article en entier, nous l'avons également faite conjointement.
3. La page **login.php** : la page et tout le système d'identification sont opérationnels, vous pouvez vous connecter à l'application (avec les deux adresses e-mail présentes dans la table users de la base de données blog : **john@mail.com** ou **alain@mail.com** avec les mêmes mots de passe : 1234 pour les deux).
4. La page **add.php** : Le formulaire (HTML) est présent et fonctionnel. Le formulaire est envoyé sur la page **manager.php** et le contrôleur PHP est opérationnel également (une redirection si déjà connecté, quelques variables initialisées et un simple contrôle de la connexion à la DB).

## Que vous reste-t-il à faire concrètement ?

1. La page **edit.php** : Le formulaire est terminé à 95%, il vous reste à placer les données dans les balises title et content, la checkbox est déjà configurée à condition que vous initialisiez une variable nommée \$article avec les données reçues par la fonction qui va chercher les données en DB et terminer le contrôleur PHP en haut de la page.
2. La page **manager.php** : Cette page est créée, possède sa structure HTML et un contrôleur avec des instructions de bases et beaucoup de commentaires pour vous aider. Vous devez la terminer et la rendre complètement fonctionnelle.
3. **Vous devez vérifier** que toute l'application fonctionne parfaitement, c.-à-d. :

- Une page d'accueil qui n'affiche que les articles publiés
- Une page article qui affiche l'article sélectionné au complet
- La possibilité de vous connecter
- Avoir un menu différent si vous êtes connecté (gérer et ajouter en plus)
- Une page de gestion fonctionnelle qui affiche tous les articles et qui permet de modifier, lire et supprimer chacun d'entre eux.
- Vous devez pouvoir ajouter un nouvel article
- Vous devez pouvoir modifier un article
- Vous devez pouvoir supprimer un article
- Vous devez afficher des messages à l'utilisateur pour l'informer que tout s'est bien passé ou non.

Je vous ai aidé au maximum de ce que je pouvais faire. Cette évaluation aura un certain poids dans l'évaluation globale des travaux de l'année (hors examen de fin d'année), environ 25%. J'espère que vous arriverez toutes et tous à rendre un projet complet. Je vous conseille de bien lire ce document, ainsi que le fil conducteur et surtout de bien lire le code et les commentaires présents dans les fichiers. Essayer au maximum de comprendre ce que j'ai écrit et bien évidemment de comprendre le code que vous écrivez.

Si vous démarrez du code présent dans LPB2024, il ne vous reste pas des milliers de lignes de code à écrire. Si vous commencez un projet au départ de votre code personnel, vous pouvez vous inspirer et utiliser le code présent dans le SuperBlog de LPB2024 ou non en fonction de vos facilités ou de vos besoins.

**Vous devez me rendre votre projet SuperBlog au plus tard le dimanche 7 à minuit.** Une archive de votre projet au format zip ou rar serai parfait.

En ce qui concerne la base de données, j'utiliserai la base de données **blog** avec mot de passe en clair ou hashé en fonction de la DB et du type d'identification que vous aurez choisi dans le contrôleur de la page login.php.

Je vous déconseille d'utiliser une autre DB que la DB blog que nous avons utilisé jusqu'à présent, une DB personnelle par exemple, si vous ne maîtrisez pas un minimum les bases de données relationnelles. Sachez que dans ce cas, vous devrez me fournir un export de votre DB au format SQL, ce qui pourrait éventuellement être plus délicat.

**Si vous avez fini à l'avance en partant du code de SuperBlog dans LPB2024, essayez de réécrire l'application vous-même en partant de rien. Vous commencez par la base, la page d'accueil et la page article et vous construisez votre application petit à petit en l'améliorant et en la complétant progressivement. C'est de cette façon que vous apprendrez et progresserez dans votre apprentissage de la programmation.**

**JE RESTE DISPONIBLE SI VOUS AVEZ  
BESOIN DE MOI :)**

**Mon adresse e-mail personnelle : [alain@kelleter.be](mailto:alain@kelleter.be)**

**Je vous souhaite une de bonnes fêtes de fin d'années, profitez bien de votre famille, de vos proches et de vos amis. On se revoit le 08 janvier pour un débriefing du projet SuperBlog et un nouveau langage, le JavaScript :)**