



 PROJET EN COURS

Créez votre premier blog en PHP


60%

MISSION **GUIDE MENTOR** COURS RESSOURCES ÉVALUATION

Note d'accompagnement

Dans ce projet l'étudiant va utiliser le langage PHP pour développer son propre blog. Celui-ci doit être réalisé from scratch et doit comporter un certains nombre de fonctionnalités, l'occasion donc de mettre en application les différentes notions acquise jusqu'ici.

Aucun framework ni front-end, ni back-end, n'est autorisé pour la partie PHP, l'idée étant que l'étudiant réalise le site « à la main » afin de renforcer sa maîtrise du langage PHP, mais également de l'architecture MVC et de la persistance des données avec le SQL. Il peut cependant utiliser les outils et librairies étant utilisées comme **moteur de template** (comme Twig) à la condition d'être intégrés grâce à Composer. Mais l'étudiant ne peut pas utiliser de framework ou de librairies javascript comme React, VueJS ou Angular.

 Pour les étudiants qui ont démarré leur projet avant le 30 juin 2020, l'utilisation de tels frameworks est autorisée car les consignes à ce propos n'était pas claires à ce moment là.

Assurez-vous que l'étudiant soit à l'aise avec les concepts de base de la programmation (boucle, variable, fonction...) et qu'il soit familier avec la programmation orienté objet surtout les notions de base.

Recommandations

L'utilisation de librairie externe est autorisé pour le projet à condition qu'elles soient intégrés via [Composer](#). Il peut être intéressant pour l'élève de s'initier à Composer, de comprendre son fonctionnement et son utilité. L'installation d'une librairie comme [Twig](#) est donc un bon exercice pour à la fois appréhender la librairie et se familiariser avec Composer et l'installation de packages.

Au niveau de l'architecture du projet, orienter l'élève vers une solution simple type MVC. L'idée est que l'étudiant comprenne ce qu'il fait et qu'il **maîtrise** le code produit.

Sensibiliser l'élève au principe "[KISS](#)" et évitez qu'il tombe dans [overengineering](#).

PHP est un langage très permissif et pour que l'étudiant ai de bon réflexe et qu'il adopte facilement les bonnes pratiques faites en sortes que le code respecte les PSR-1, PSR-12 et PSR-4 (si il utilise l'autoloader de Composer). Gardez à l'esprit que le blog servira peut-être de vitrine à votre étudiant, le code doit donc être propre et lisible (commentaire, naming...).

Il est recommandé d'utiliser l'autoloader de Composer plutôt que d'utiliser la fonction "require" ou "include" pour appeler ses classes.

Rappelez à votre étudiant le concept de "Never trust user input" et assurez-vous qu'il n'y a pas de failles de sécurité (XSS, CSRF, SQL injection, session hijacking, upload possible de script php...).

Pour rappel le blog doit être créé from scratch et la principale difficulté est souvent le syndrome de la page blanche, de manière générale il est très difficile de mettre en place une architecture et d'apporter une solution technique à un besoin fonctionnel. Soyez donc vigilant et anticiper les éventuels blocages, si besoin accompagner l'étudiant dans la mise en place des premiers fichiers/premières classes.

Le but n'est pas de développer le projet à la place de l'élève mais simplement de l'orienter vers la bonne direction comme vous savez si bien le faire :-)

Aidez l'élève à découper les fonctionnalités en tâche et à procéder par itérations, voici un exemple pour l'affichage d'un post.

- Création de la route /post/{id}
- Création d'un Controller PostController
- Création de la méthode qui récupère/affiche le post
- Création d'une vue pour afficher le post
- Création d'une entité Post
- Création d'une classe pour requêter la base de données
- Création de la méthode pour récupérer un Post en base de données

D'une session à l'autre fixer des objectifs clair et atteignable, il ne faut pas hésiter à détailler ce qui doit être fait **surtout au début du projet**.

Des issues doivent être créer pour ce projet pensez qu'une fonctionnalité peut correspondre à une issue, créer les dès le début du projet pour que l'étudiant puisse suivre son avancement.

Soutenance

Assurez-vous que le projet réponde aux besoins fonctionnelles, que l'étudiant soit bien l'auteur du code produit, qu'il sache où et quoi modifier si on veut ajouter une fonctionnalité.

C'est la première fois que l'étudiant va montrer du code qu'il aura lui même produit, l'idée n'est pas de lire tout le code mais d'expliquer le code pour 1 ou 2 fonctionnalité.

Erreurs récurrentes

- Ne pas respecter les PSR
- Ne pas mettre en place de sécurité (XSS, CSRF, SQL injection, session hijacking, upload possible de script php...)
- Ne pas profiter de l'orienté objet et dupliquer le code
- Ne pas passer par des entités
- Avoir une seule et même classe qui requête à la base de données les Post, Commentaires, Users.
- Ne pas personnaliser les pages d'erreurs
- Mettre beaucoup de logique dans la vue
- Ne pas faire d'héritage de templates
- Ne pas respecter le principe d'encapsulation
- Ne pas hydrater ses objets venant de la base de données
- Ne pas versionner son projet (absence de branche, issues, Pull request)
- Ne pas créer d'issue
- Ne pas valider les commentaire soumis

Critères d'évaluation

Estimer une tâche et tenir les délais

- ☐ Mise en place d'issues
- ☐ Estimations des tâches

Gérer ses données avec une base de données

- ☐ Classe de connexion à la base de données
- ☐ Présence d'entités

Proposer un code propre et facilement évolutif

- ☐ Respect des PSR
- ☐ Pas beaucoup de logique dans les vues
- ☐ Naming cohérent, code autodocumenté et commentaires présents (si besoin)

Assurer le suivi qualité d'un projet

- ☐ Présence d'un rapport indiquant la qualité du code produit

Choisir une solution technique adaptée parmi les solutions existantes si cela est pertinent

- ☐ Utilisation d'une version récente de PHP
- ☐ (facultatif) Utilisation de l'autoloader de composer
- ☐ (facultatif) Utilisation de Twig

Créer et maintenir l'architecture technique du site

- ☐ Création d'un projet "from scratch"
- ☐ Mise en place d'une architecture cohérente
- ☐ Présence d'un fichier README

Créer une page web permettant de recueillir les informations saisies par un internaute

- ☐ formulaire de contact.
- ☐ formulaire d'ajout de post.
- ☐ formulaire de modification d'un post.
- ☐ formulaire d'ajout de commentaire (soumis à validation).

Analyser un cahier des charges

- ☐ Ensemble des fonctionnalités présentes
- ☐ Respect des contraintes technique

Conceptualiser l'ensemble de son application en décrivant sa structure (Entités / Domain Objects)

- ☐ Présence des schémas UML

Rédiger les spécifications détaillées du projet

- ☐ Présence des spécifications (exemple : schéma d'arborescence du site)

OPENCLASSROOMS

OPPORTUNITÉS

POUR LES ENTREPRISES

 Français 

Qui sommes-nous ?

Nous rejoindre 

Former et recruter

Alternance

Devenir mentor 

EN PLUS

Financements

Devenir coach carrière 

Boutique 

Expérience de formation

AIDE

Mentions légales

Forum



FAQ

Blog 

Conditions générales d'utilisation

Politique de protection des données personnelles

Cookies

Accessibilité

