|  |  |  |  |
| --- | --- | --- | --- |
| **Compétences à évaluer** | **Critère d’évaluation** | **Indicateurs** | **Validée** |
| Installer un framework PHP | Affichage de la page d’accueil du framework | La page d’accueil s’affiche |  |
| Associer une URL à une page | La configuration du fichier de routing. | Le squelette du site est disponible au format papier.  La page demandée s’affiche dans le navigateur. |  |
| Associer une URL à un controller | La création d’un controller  La configuration du fichier de routing pour appeler un controller | Schéma papier de transmission des requêtes entre client et serveur.  Le fichier controller est créé.  La page demandé s’affiche dans le navigateur. |  |
| Afficher une vue associée à un controller | Création d’une VIEW  Appel d’une VIEW depuis un Controller | Les résultats des 2 formulaires s’affichent. |  |
| Utiliser un template pour éviter les redondances de code HTML. | Création d’un template ‘Master’  Utilisation de l’héritage des VIEW. | Les vues des différentes pages ne comportent pas de duplication de code HTML. |  |
| Compréhension des concepts POO en PHP. | Création du mémo papier PHP | Réalisation à 100% des tutos codecademy sur le PHP Objet.  Le mémo papier fait référence aux concepts : attributs / getter / setter / instancier / héritage (*innheritance*) / surcharger (*overriding*) |  |
| Mise en oeuvre des concepts POO en PHP. | Développement du jeu “Warrior” | Les 16 premiers points de contrôle du jeu sont validés (vert). |  |
| Compréhension de l’architecture Model / View / Controller | Schéma sur papier décrivant l’architecture MVC | Le schéma sur papier décrivant l’architecture MVC présente les étapes réalisées par le framework pour afficher une page intégrant des données dynamiques. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Compétences à évaluer** | **Critère d’évaluation** | **Indicateurs** | **Validée** |
| Créer un Model avec l’ORM. | Creation d’un Model.  Utilisation du Model dans le Controller.  Récupération des données du Model dans la page. | Le fichier Model est créé.  La page affiche des données issues de la BDD sans utiliser de requête SQL. |  |
| Créer un formulaire et enregistrer les données en BDD (via le model) | Récupération des données d’un formulaire.  Manipulation des données du model : INSERT / UPDATE / DELETE | Les données enregistrées dans la BDD sont identiques aux données saisies dans les formulaires.  Les données de la BDD s’affichent sur les pages.  Le document de description fonctionnelle contient toutes les pages, tous les objets et toutes les routes du site |  |
| Mettre en place des relations entre les Models. | Definition des relations entre les différents Models: One To One, One To Many, Many to Many | On peut accéder aux données des Model liées au Model principal depuis les CONTROLLER/VIEW. |  |
| Sécuriser l’accès à son application via l’authentification | Mise en place et fonctionnement de l’authentification. | Un utilisateur peut se connecter via un Login + Mot de passe.  L’utilisateur authentifié peut accéder aux données privatives.  Un utilisateur non identifié n’est pas autorisé à accéder à des donnés dont il n’est pas le propriétaire. |  |

# 