

EVALUATION PHP – MVC Orienté Objet

INSCRIPTION DE JOUEURS EN BASE DE DONNEES

Table des matières

INSCRIPTION DE JOUEURS EN BASE DE DONNEES	1
OBJECTIFS PEDAGOGIQUES	1
CONTEXTE	2
CRITERES D'EVALUATIONS	2
Fichiers fournis	4
Tâches à effectuer	4
RESULTAT ATTENDU	5

OBJECTIFS PEDAGOGIQUES

- Être capable de comprendre et de mettre en place une structure MVC
- Être capable de recevoir les données envoyées par un formulaire
- Être capable de sécuriser un formulaire d'envoi de donnée
- Être capable de communiquer avec une base de données pour insérer des données
- Être capable de communiquer avec une base de données pour lire des données
- Être capable de se prémunir des injections SQL
- Être capable de traiter et d'afficher les données provenant d'une base de données
- Être capable de comprendre le fonctionnement et l'intérêt de la POO
- Être capable de créer et d'utiliser les classes
- Être capable de créer et d'utiliser des objets
- Être capable de comprendre les notions d'héritage

Auteur :

Yoann DEPRIESTER

Date création :

16-01-2025

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

16-01-2025



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

EVALUATION PHP – MVC Orienté Objet

CONTEXTE

Un nouveau projet arrive sur le bureau du directeur de votre agence de développement. La Croix-Rouge souhaite mettre en place un serious game pour sensibiliser la population aux gestes de premiers secours.

Votre Chef de Projet, conjointement avec l'Architecte Logiciel et le Tech Lead, décide de développer ce serious game en PHP pour toucher un maximum de monde par le biais d'un site dédié. La structure du code doit pouvoir assurer une bonne maintenabilité, et l'emphase sera mis sur la gestion des données. C'est pourquoi le pattern Model – View - Controller , développé en Orienté Objet, a été choisi.

Au sein de l'équipe, il vous a été confié le développement de la gestion des utilisateurs. Il vous a été fourni le Diagramme de Classe du projet, un accès à une base de données remplis avec 3 utilisateurs tests, et la structure de fichier utilisée.

A vous de jouer pour développer les fonctionnalités demandées.

CRITERES D'EVALUATIONS

Propreté du code et indentation	3 points
Commentaire et Documentation des fonctions	2 points
Respect de la convention de nommage (nom en anglais, variable et fonction en camel case, constante en upper case, ...)	3 points
Création et manipulation de variables	2 points
Création et utilisation de fonctions	2 points
Manipulation des conditions et opérateurs logiques	2 points
Manipulation des boucles	2 points
Manipulation des tableaux	2 points
Ecrire du HTML sémantique	2 points
Utiliser la méthode appropriée pour transmettre les informations d'un formulaire	4 points
Sécuriser la réception de données clients	6 points
Hasher le mot de passe avant enregistrement en base de données	4 points
Vérifier qu'un utilisateur n'existe pas avant de l'enregistrer	4 points
Mettre en forme un affichage à partir d'un tableau de données	4 points
Gestion de cas d'erreur, notamment à l'aide de Try... Catch	4 points
Afficher des messages d'erreur ou de confirmation aux utilisateurs	4 points
Configurer correctement l'objet de connexion à la base de données	4 points

Auteur :

Yoann DEPRIESTER

Date création :

16-01-2025

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

16-01-2025

EVALUATION PHP – MVC Orienté Objet

Créer des classes et interfaces en respectant leur relation au sein du Diagramme de Classe	4 points
Créer des attributs typés et respectant l'encapsulation du Diagramme de Classe	4 points
Le constructeur de la classe ModelPlayer ne fait qu'assigner un objet de connexion PDO à l'attribut PDO en utilisant la fonction connect() du fichier utils.php	2 points
Créer des Getter et Setter pour chaque attribut au sein de la bonne classe	4 points
Les méthodes demandées sont créées dans la bonne classe et tirent parties des attributs de la classe	6 points
L'envoi des requêtes à la base de données prévient les risques d'injection SQL	6 points
Les include / require sont effectués au sein du fichier d'exécution	2 points
Les objets sont correctement instanciés et exploités	4 points
Lorsqu'un utilisateur est enregistré, il apparaît directement dans la liste sans avoir besoin d'actualiser la page	2 points
Le Controller transmet correctement les données à la View	4 points
La View affiche correctement les données	4 points
Les spécificités du pattern Model – View – Controller sont respectés	4 points
TOTAL	100 points

Auteur :

Yoann DEPRIESTER

Date création :

16-01-2025

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

16-01-2025



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

EVALUATION PHP – MVC Orienté Objet

Fichiers fournis

- Le script SQL : eval_supergame.sql
- Le Diagramme de Classe : eval_supergame_diag_class-CDA.png
- Une structure de fichier de base : structure_des_fichiers-CDA.png
- L'archive ZIP contenant la structure de fichier : supergame.zip

Tâches à effectuer

- 1) Dans MySQL Workbench, ouvrir et exécuter le script SQL fourni
- 2) Dézipper l'archive à la racine de votre server local (Laragon, XAMP, MAMP, ...)
- 3) Au sein de votre éditeur de code, dans le fichier utils.php, configurer correctement les paramètres du constructeur dans la fonction connect()

En vous appuyant tout à la fois sur le Diagramme de Classe fourni, et le script SQL fourni :

- 1) Dans le fichier viewPlayer.php, créer une classe dont la méthode displayView() renverra un formulaire permettant d'enregistrer un nouveau joueur, et une section qui permettra d'afficher la liste des joueurs avec leur pseudo et leur score
- 2) Dans les fichiers viewHeader.php et viewFooter.php, créer les classes ViewHeader et ViewFooter dont les méthodes displayView() renverront le html du header et du footer.
- 3) Dans le fichier interfaceModel.php, mettez en place l'interface InterfaceModel
- 4) Dans le fichier playerModel.php, mettez en place la classe ModelPlayer

NOTE :

add() est la méthode pour enregistrer un joueur avec son pseudo, son email, son score et son mot de passe en BDD.

getAll() est la méthode pour récupérer toute la liste des joueurs avec leurs données depuis la BDD.

getEmail() est la méthode pour récupérer les données d'un joueur unique que l'on cherche grâce à son email.

- 5) Mettez en place la classe abstraite AbstractController dans le fichier abstractController.php, puis mettez en place la classe PlayerController dans le fichier playerController.php

Auteur :

Yoann DEPRIESTER

Date création :

16-01-2025

Relu, validé & visé par :

☑ Jérôme CHRETIENNE
☑ Sophie POULAKOS
☑ Mathieu PARIS

Date révision :

16-01-2025

EVALUATION PHP – MVC Orienté Objet

Le fichier index.php va servir de fichier d'exécution :

- 1) Importez vos ressources dans le bon ordre
- 2) Créer un nouvel objet PlayerController
- 3) Demander à votre objet de lancer le rendu avec sa méthode render()

RESULTAT ATTENDU

Accueil

Inscription d'un Joueur

Yoann a été enregistré en BDD !

Flora

flora@gmail.com

Score : 500

Kevin31

theking@sfr.fr

Score : 900

Chuck Norris

beyondgod@hotmail.com

Score : 100

Yoann

depriester.yoann@neuf.fr

Score : 250

Auteur :

Yoann DEPRIESTER

Relu, validé & visé par :

- ☒ Jérôme CHRETIENNE
- ☒ Sophie POULAKOS
- ☒ Mathieu PARIS

Date création :

16-01-2025

Date révision :

16-01-2025



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.