Projet Trivia

Documentation technique

Contenu

Présentation de l’application

Il s'agit d'adapter pour l'entreprise un jeu (Trivia crack), qui permettra aux salariés de tester et d'entretenir leurs connaissances de façon ludique dans les domaines du Monde dans lequel ils travaillent. L’utilisateur appartient à un monde, dans ce monde il pourra jouer dans différents domaines qui possède chacun des questions.

Pour chaque bonne réponse, un point est attribué sinon c’est à l’autre joueur de jouer. Au bout de trois bonnes réponses, on joue pour une couronne (un domaine). Dès que l’utilisateur débloque toutes les couronnes, il gagne.

L’application contient donc une partie pour répondre au QCM et une partie administration permettant la gestion (ajout, modification, suppression) des domaines, des questions et des réponses.

I/ Caractéristiques techniques

- Langage de programmation : HTML5, CSS3, JavaScript (JQuery), PHP (5.4.12)
- Type de développement : Programmation Orientée Objet (POO)
- Application WEB
- Utilisation d'un ORM (Object Relational Mapper) basé sur les annotations sur les membres de données des classes
- Plateforme de développement : XAMPP (version 1.8)
- Système de Gestion de Base de Données : MySQL (version 5.0.11)
- Serveur Web : Apache (version 2.4.10)
- Navigateurs web : Google Chrome (version 34); pour l'accès à l'interface d'administration (PHPMyAdmin) de la base de données sous MySQL
- Environnement de Développement Intégré : PhpStorm (version 8.0.1)

II/ MVC

L'application utilise le patron de conception MVC (Modèles, Vues, Contrôleurs) afin de séparer les traitements, les données, et la présentation. Ce qui offre un cadre normalisé pour structurer l' application, et facilite le dialogue entre les différents concepteurs.
L'idée est de bien séparer les données, la présentation et les traitements.

III/ Réécriture des URL

Afin de manipuler des URL qui ne montrent pas l'arborescence de l'application (Assurance de confidentialité des données), les URL du projet se présentent sous la forme suivante :
nomServeur / nomProjet / nomContrôleur / nomMéthode / param1 / param2 / …

Cette réécriture donnera lieu à :
$obj = new nomContrôleur();
$obj->nomMéthode(param1, param2);

Par exemple, l'URL : http://localhost/trivia/CQuestion/afficherQuestion/22, appellera la méthode afficherQuestion du contrôleur CQuestion à laquelle on passera 22 en paramètre.

IV/ Arborescence du projet

Dossiers de l'application :

Nom du package	Rôle
classes	Contient les classes métier permettant l'instanciation des objets métier
controllers	Contient les contrôleurs de l'application
css	Contient les feuilles de style CSS
images	Contient toutes les images
js	Contient toutes les fonctions JavaScript/ Jquery utilisées
technics	Contient les classes techniques de l'application
views	Contient toutes les vues de l'application
sql	Contient le script de création de la base de données

Classes métier:

Nom	Descriptif
BaseObject	Classe de base dont héritent la majorité des autres classes métier pour avoir un id
Couronne	Classe permettant l'instanciation d'une couronne
Domaine	Classe permettant l'instanciation d'un domaine
Joueur	Classe permettant l'instanciation d'un joueur
Monde	Classe permettant l'instanciation d'un monde
Partie	Classe permettant l'instanciation d'une partie
Probleme	Classe permettant l'instanciation d'un problème
Question	Classe permettant l'instanciation d'une question
Reponse	Classe permettant l'instanciation d'une reponse
Score	Classe permettant l'instanciation d'un score
Signalement	Classe permettant l'instanciation d'un signalement
Statistiques	Classe permettant l'instanciation d'une statistique

Classes techniques:

Nom	Descriptif
DAO	Classe passerelle entre Base de données et objets
Database	Classe d'accès à une base de données, encapsule un objet PDO
GUI	Gestion de l'affichage (évite les excès de code dans les vues)
JsUtils	Génération de scripts côté client
OrmUtils	Récupération des annotations de mappage relationnel/objet
RequestUtils	Récupération des variables POST ou GET
SessionUtils	Méthodes utilitaires liées à la session
SqlUtils	Méthodes utilitaires liées à SQL

Contrôleurs :

Nom	Descriptif
BaseController	Contrôleur de base dont héritent tous les autres contrôleurs
CDomaine	Contrôleur lié à la gestion des domaines
CJoueur	Contrôleur lié à la gestion du menu et du chargement du joueur
CPartie	Contrôleur lié à la gestion des parties
CQuestion	Contrôleur lié à la gestion des questions d'une partie
CReponse	Contrôleur lié à la gestion des réponses d'une question
CScore	Contrôleur lié à la gestion des scores des joueurs
CStatistiques	Contrôleur lié à la gestion des statistiques d'un joueur
CCouronne	Contrôleur lié à la gestion des couronnes d'un joueur
CMonde	Contrôleur lié à la gestion des mondes

V/ Bonnes pratiques et normes suivies

Structure de l'application :

L'application rivia, respectant le patron de conception MVC, doit respecter la structure suivante :

- Les classes métiers devront être stockées dans le dossier classes
- Les classes techniques devront être stockées dans le dossier technics
- Les contrôleurs devront être stockés dans le dossier controllers
- Les vues dans le dossier views
- Les images dans le dossier images

Classes métier :

- Leur nom doit correspondre à une table de la base de données
- Leur nom commence par une majuscule

Contrôleurs :

- Leur nom sera toujours préfixé par "C", pour montrer qu'il s'agit d'un contrôleur

Vues :

- Leur nom commence toujours par un "V" majuscule, permettant ainsi de déterminer rapidement qu'il s'agit d'une vue
- On s'efforcera de donner un nom pertinent aux vues, pour éviter de devoir lire le code pour déterminer leur utilité