

## EXAMEN FINAL

### ÉNONCÉ

**Examen : My Scoreboard**

**Durée : 4 heures**

**Projet :**

*Un client vous propose le projet suivant : créer une plateforme qui permet de gérer des matchs et scores à des parties de jeux de société. La plateforme sera librement accessible : on pourra ajouter des joueurs par leur pseudonyme et leur e-mail et ajouter des jeux à la liste de jeux de société d'une part ; et créer des parties autour d'un jeu et d'une liste de joueurs d'autre part.*

---

### EXERCICE 1 – MISE EN LIGNE DU PROJET

#### 2 POINTS

- **Une mise en ligne de votre projet** de sorte à pouvoir le consulter « en production » :
  - Créer un compte sur <https://fr.000webhost.com> ou <https://infinityfree.net> ou l'hébergeur de votre choix et uploader vos codes sources de l'évaluation sur ce serveur.
  - Vous devrez rendre l'adresse url de votre site web afin que le correcteur puisse évaluer votre travail.
  - Par exemple : <http://www.example.com/evaluation-php/>
  - Ne mettez pas de fichier *index.php* ni *index.html* afin que le correcteur puisse avoir accès à votre listing de fichier.
  - En fin d'évaluation, pensez à uploader l'archive \*.zip de votre projet afin que le correcteur puisse récupérer votre code source directement sur le serveur.

## EXERCICE 2 – CREATION DE LA BASE DE DONNEES

### 3 POINTS

- **A - Créez une base de données** selon les indications suivantes :

DATABASE : wf3\_php\_final\_prenom

TABLE : game (id, title, min\_players, max\_players)

TABLE : player (id, email, nickname)

TABLE : contest (id, game\_id, start\_date, winner\_id)

TABLE : player\_contest (id, player\_id, contest\_id)

- **B - Pré-remplissez la base** avec les données suivantes :

- o Table **GAME (id, title, min\_players, max\_players)**

23	7 Wonders	2	7
24	Ticket to Ride	2	5
25	Pandemic	2	4
26	Munchkin	3	6

- o Table **PLAYER (id, email, nickname)**

1	luke.skywalker@rogue.sw	Luke
2	amidala.padme@naboo.gov	Padme
3	han.solo@millenium-falcon.com	HanSolo
4	chewbacca@wook.ie	Chewbie
5	rey@jakku.planet	Rey

- o Table **CONTEST (id, game\_id, start\_date, winner\_id)**

1	23	2019-12-25	2
2	25	2020-12-25	

- o Table **PLAYER\_CONTEST (id, player\_id, contest\_id)**

1	1	1
2	2	1
3	3	1
4	4	1
5	5	1
6	2	2
7	3	2
8	5	2

- **C - Exportez votre base de données** dans un fichier *database.sql*

---

### EXERCICE 3 – AJOUTER DES DONNEES

5 POINTS

- **En respectant une architecture MVC tout au long de l'examen, créez les formulaires suivants :**
  - Ajouter un jeu (game)
  - Ajouter un joueur (player)
  - Ajouter un match (contest)
- Vous réaliserez des contrôles de saisie sur chaque champ. Les champs seront typés, validés en HTML et en PHP.

---

### EXERCICE 4 – AFFICHER DES DONNEES

5 POINTS

- **Sur une page d'accueil, vous afficherez les données suivantes :**
  - Liste des joueurs (dans un tableau)
  - Liste des jeux disponibles (dans un tableau)
  - Liste des matchs, en « grisant » ou isolant visuellement les matchs qui n'ont pas commencé. Les données affichées doivent être :
    - Nom du jeu
    - Nombre de joueurs enregistrés
    - Date de démarrage
    - Pseudonyme du gagnant du match
    - Triées par date de démarrage décroissante

---

### EXERCICE 5 – AFFICHER ET GERER UN MATCH

5 POINTS

- **Vous créerez une page « Match » affichant le détail d'un match, accessible via un lien dans le tableau des matchs en page d'accueil.**
  - Sur la page d'un match, vous afficherez un formulaire permettant d'ajouter un joueur au match.
  - Vous afficherez la liste des joueurs du match, en affichant leur pseudonyme et email
  - Sur cette page, si le nombre de joueur n'est pas suffisant/trop important, vous l'indiquerez à l'utilisateur.
  - Vous intégrerez une option permettant de supprimer un joueur d'un match
  - Vous intégrerez un moyen d'indiquer qui est le gagnant du match. Cette option ne peut pas être disponible tant que la date de démarrage du match n'a pas encore été dépassée.