**Projet : gestion d'une équipe de sport**

**Votre équipe favorite a besoin de vous !**

L'entraîneur vous demande de réaliser une application qui l'aidera à faire les sélections des joueurs pour les matchs de SON équipe.

Il souhaite pouvoir administrer la liste de ses joueurs (avec leurs noms et prénoms, leur numéro de licence, leur date de naissance, leur taille et leur poids) ainsi que celle des matchs (avec la date et l'heure, le nom de l'équipe adverse, le lieu de rencontre - domicile ou extérieur -, et le résultat qui sera saisi une fois le match terminé).

Il souhaite également pouvoir ajouter des notes personnelles (commentaires) sur chaque joueur et préciser leur statut : Actif, Blessé, Suspendu, ou Absent.

Avant chaque match il veut pouvoir choisir la liste des joueurs qui participeront, en précisant qui sera titulaire et qui sera remplaçant, et le poste occupé par chaque participant. Pour constituer ces feuilles de matchs, il ne faudra proposer à l'entraîneur que les matchs à venir, et seulement les joueurs actifs.

Après chaque match, il souhaite pouvoir évaluer la performance de chaque joueur ayant participé au match ; l'évaluation peut être mise en oeuvre par un système de notation (de 1 à 5 par exemple) ou un système d'étoiles par exemple.

Enfin, il souhaite avoir des statistiques qui l'aideront dans sa prise de décision. Ces statistiques devront donner :

* Le nombre et le pourcentage de victoires, le nombre de défaites et le nombre de matchs nuls.
* Un tableau donnant, pour chaque joueur, son statut actuel, le poste sur lequel il a été le plus performant, son nombre de titularisations, son nombre de « remplacements » (nombre de match où le joueur est positionné en tant que remplacant), la moyenne de ses évaluations de l’entraîneur, le nombre de matchs consécutifs auxquels il a participés, et le pourcentage de matchs qu'il a gagnés parmi ceux auxquels il a participé).

**CONSIGNES**

**Ce projet doit être réalisé en binôme.**

**Vous devez impérativement respecter les contraintes suivantes :**

* Le code SQL doit être séparé de votre code HTML et CSS, par exemple sous la forme d'une librairie contenant différentes fonctions permettant d'interagir avec la base de données.
* Les accès à la base de données doivent être faits via PDO (toutes les fonctions mySQL et mySQLi sont interdites).
* Votre application doit être accessible seulement après authentification de l'utilisateur (i.e., de l'entraîneur). Le(s) mot(s) de passe ne devront pas être stocké(s) en clair dans la base.
* Vous devez utiliser Git pour la gestion du code source de votre projet.
* Vous devrez prendre en compte la prévention aux injections SQL.

Vous pouvez choisir le sport d'équipe de votre choix (Football, Rugby, Basketball, Volleyball, etc.), vous ferez les adaptations nécessaires à chaque sport (nombre de titulaires par match par exemple).

Avant de vous lancer dans le développement prenez le temps de bien réfléchir à votre application dans sa globalité. N'hésitez pas à faire des maquettes des différents écrans et posez-vous des questions d'ordre pratique (par exemple, est-ce qu'il ne serait pas intéressant de mémoriser le statut d'un match pour savoir s'il a déjà été préparé ou pas ?).

Gardez à l'esprit que l'application devra être pratique à utiliser et accessible à des non informaticiens (c'est-à-dire un entraîneur de sport).

* **Modèle de données**

Sur papier ou à l'aide de l'outil de votre choix, réaliser le modèle de données pour cette application.

**APRÈS L'AVOIR FAIT VALIDER** par votre enseignant, créer la base de données MySQL correspondante.

* **Gestion des joueurs et des matchs**

Créer les pages nécessaires à l'affichage, l'ajout, la modification, et la suppression des joueurs et des matchs.

* **Saisie des feuilles de match**

Créer une page permettant de faire une sélection parmi les joueurs actifs et de définir pour chaque joueur choisi s'il sera titulaire ou remplaçant. Si le nombre minimum de joueurs n'est pas atteint, la sélection ne devra pas pouvoir être validée. L'interface de sélection devra afficher les informations des joueurs : taille, poids, commentaires et évaluations de l'entraineur.

Adapter l'affichage des matchs pour permettre de visualiser et modifier la sélection.

* **Statistiques**

Si ce n'est pas déjà fait, modifier la page de modification d'un match pour permettre la saisie du résultat ainsi que les évaluations de l'entraîneur.

Créer ensuite une page affichant les statistiques suivantes :

* + Le nombre total et le pourcentage de matchs gagnés, perdus, ou nuls.
  + Un tableau avec pour chaque joueur : son statut actuel, son poste préféré, le nombre total de sélections en tant que titulaire, le nombre total de sélections en tant que remplaçant, la moyenne des évaluations de l'entraîneur, et le pourcentage de matchs gagnés lorsqu'il a participé.
  + Si possible, ajouter également le nombre de sélections consécutives (facultatif).
* **Cadre de l'application**

Sécuriser l'application en créant une page d'authentification (à l'aide d'un nom d'utilisateur et d'un mot de passe définis à l'avance). Aucune autre page de l'application ne devra être accessible si l'utilisateur n'est pas authentifié.

Mettre en place un menu qui sera affiché sur chaque page pour permettre à l'utilisateur de naviguer dans l'application. Ajouter tous les liens nécessaires entre les différentes pages.

* **Mise en forme**

Utiliser les feuilles de style (CSS) et les bases d'ergonomie logicielle pour faire en sorte que l'utilisation de l'application soit la plus agréable et intuitive possible.

NB : La priorité reste le code et les fonctionnalités, attention à ne pas perdre trop de temps sur la forme.

* Liste des hébergeurs en ligne gratuits

Voici une liste d'hébergeurs en ligne gratuits et qui intègrent la pile Apache/Mysql/PHP. **N'oubliez pas de vous assurer de la compatibilité** de votre environnement de développement (a priori, votre ordinateur) avec celui de l'hébergeur que vous choisissez !

* + AlwaysData : <https://www.alwaysdata.com/fr/>
  + InfinityFree : <https://www.infinityfree.com/>
  + 000webhost : <https://www.000webhost.com/>
  + PlanetHoster : <https://www.planethoster.com/fr/World-Lite>
  + Byet : <https://byet.host/>

**1. Decide on the Sport**

Choose the sport you want to manage (e.g., football, basketball, volleyball). This choice will affect things like the **number of players per team** and any sport-specific positions or roles.

**2. Sketch the Data Model**

Based on the requirements, create a sketch (on paper or digitally) of the main tables and relationships. Here’s a basic outline to help you get started:

* **Players Table**: Stores player information.
  + player\_id (Primary Key)
  + name
  + license\_number (Unique ID for each player)
  + birth\_date
  + height
  + weight
  + status (Active, Injured, Suspended, Absent)
  + comments
* **Matches Table**: Stores match details.
  + match\_id (Primary Key)
  + date
  + time
  + opponent
  + location (Home/Away)
  + result (Win/Loss/Draw, entered after the match)
* **Player\_Match Table**: Links players with each match to track their participation and role.
  + player\_match\_id (Primary Key)
  + match\_id (Foreign Key referencing Matches)
  + player\_id (Foreign Key referencing Players)
  + role (Starter/ Substitute)
  + position (e.g., Forward, Goalkeeper for football)
  + performance\_rating (Rating after the match)

**3. Draw the Data Model**

If you're using a tool (like Modelio, Lucidchart, or a similar diagram tool), create the tables and relationships based on the fields above. If not, you can simply draw it out on paper to show relationships (e.g., Player\_Match links player\_id and match\_id as foreign keys to the respective tables).

**4. Validate the Model**

Once you have a sketch, validate it by considering questions like:

* Are all required fields included?
* Does each table serve a specific purpose?
* Do the relationships make sense?

Once you've completed this, let me know, and we can move on to **creating the database** in MySQL based on this model!

**1. Decide on the Sport ( football)**

**2. Sketch the Data Model**

**efining the data model specifically for football and then set up the database.**

**Football-Specific Data Model Adjustments**

**In football, the starting lineup typically includes 11 players, with specific positions such as Goalkeeper, Defender, Midfielder, and Forward. Let’s tailor the model to reflect this.**

**Finalized Tables**

1. **Players Table: This table will store each player’s profile and status.**
   * **player\_id (Primary Key): Unique identifier for each player.**
   * **name (VARCHAR): Full name of the player.**
   * **license\_number (VARCHAR, Unique): A unique ID specific to each player.**
   * **birth\_date (DATE): Date of birth.**
   * **height (FLOAT): Player’s height in meters.**
   * **weight (FLOAT): Player’s weight in kilograms.**
   * **status (ENUM): Can be "Active," "Injured," "Suspended," or "Absent."**
   * **comments (TEXT): Personal notes on the player.**
2. **Matches Table: Stores each match’s information.**
   * **match\_id (Primary Key): Unique identifier for each match.**
   * **date (DATE): Date of the match.**
   * **time (TIME): Time of the match.**
   * **opponent (VARCHAR): Name of the opposing team.**
   * **location (ENUM): Either "Home" or "Away."**
   * **result (ENUM): Will be filled after the match as "Win," "Loss," or "Draw."**
3. **Player\_Match Table: Tracks which players participate in each match and their roles.**
   * **player\_match\_id (Primary Key): Unique identifier for each record.**
   * **match\_id (Foreign Key): Links to the match\_id in Matches.**
   * **player\_id (Foreign Key): Links to the player\_id in Players.**
   * **role (ENUM): Either "Starter" or "Substitute."**
   * **position (ENUM): Football-specific positions such as "Goalkeeper," "Defender," "Midfielder," "Forward."**
   * **performance\_rating (INT): Rating from 1 to 5 or star-based.**

**3. Draw the Data Model**

**4. Validate the Model**