
R3.01 - ProgWeb

TP 4 - Rugby World Cup

Contexte

Vous allez coder un site Web traitant les résultats de la Coupe de Monde de Rugby 2023.

Nous allons vous présenter un lot d'écrans et nos attentes. Vous avez ensuite la liberté de coder, en PHP uniquement, en respectant les impératifs suivants :

- Vous devez mettre en œuvre le modèle architectural MVC et les bonnes pratiques que vous avez apprises dans cette ressource R3.01.
- Vous devez stocker vos données en BDD.
- Vous ne devez pas utiliser de framework ou de librairies spécifiques autres que les modules standards déjà proposés par PHP.
- Vous devez utiliser des conteneurs Docker sur la même base que le TP3, ce qui inclut obligatoirement l'usage des procédures de sauvegarde et de restauration que vous avez déjà utilisées.
- Le fichier **dotenv** fourni avec le **docker-compose.yml** doit être renommé en **.env**
- Ce TP tient sur 2 séances (4H).

Introduction

Vous allez donc créer une nouvelle application Web permettant d'organiser les rencontres lors de la Coupe du Monde de Rugby, de consigner les résultats et de produire des classements.

Sur Moodle, vous disposez des fichiers suivants :

- **EQUIPES** : liste des équipes
Format du fichier : **POULE=Code pays/Nom pays,...**
- **MATCHES** : toutes les rencontres de poules (avec résultats)
Format du fichier :
CodePays1-CodePays2=Date,Score,PointsPays1-PointsPays2
où **CodePaysN** sont les deux pays qui se sont affrontés dans ce match, et **PointsPaysN** sont les points (entre **0** et **5**) que chaque pays (**1** ou **2**) a gagnés pour le classement, grâce au résultat de ce match.
- **LOGOS_RWC2023.zip** : les images des logos des équipes ainsi que le logo de la Coupe du Monde 2023 pour habiller vos pages.

Etape 1

Définissez le modèle relationnel (et créez manuellement les tables en BDD) permettant :

- Stockage des Équipes
- Stockage des matches (et des résultats) de la phase de poule
- Stockage des matches (et des résultats) de la phase finale

Note concernant les URLs

Une URL se terminant par un `/` provoquera l'appel d'un script **index.php** situé dans le dossier ciblé par l'URL. Par exemple à l'appel d'une URL `/users/etu/1a/`, le serveur Web exécutera le script **index.php** (s'il existe) situé dans le sous-dossier **1a** du sous-dossier **etu** du dossier **users** lui-même situé à la racine du serveur Web¹. Si le script **index.php** n'existe pas dans le dossier, vous aurez une erreur **403 FORBIDDEN**².

Les URLs qui vont vous être demandées dans ce TP reposent sur ce principe.

Si des paramètres sont passés à la suite d'une URL terminée par un `/`, ils seront présents dans la superglobale `$_GET` du script **index.php** qui est implicitement ciblé. Exemple d'une telle URL (sans la partie `scheme`³ + domaine du site) :

`/users/etu/1a/?id=55618&nom=LULU`

index.php recevra, dans `$_GET`, deux paramètres **id** et **nom** avec leurs valeurs associées (**55618** et **LULU**).

Etape 2

Remplissez votre BDD à partir des fichiers fournis. Pour cela, définissez une URL `/filldb/` qui effectue cette tâche et que vous pourrez appeler librement quand nécessaire pour (re)peupler votre base de données.

Pour des questions de simplicité pour les tests, placez sur la page d'accueil, des liens vers chaque URL demandée dans ce TP

Définissez aussi une URL `/cleardb/` qui supprime tous les enregistrements de vos tables. Comme avec l'URL `/filldb/`, cette URL `/cleardb/` vous sera très utile pour nettoyer votre base de données si besoin.

¹ La racine du site Web est, pour Apache, généralement dans `/var/www/html` (dans le conteneur) mais peut se trouver ailleurs pour d'autres serveurs Web tels que nginx. Ce point importe peu pour ce TP.

² Car en l'absence de fichier **index.php**, le serveur interprète cette URL comme une demande d'afficher la liste des fichiers présents dans le dossier, ce qui est généralement refusé par le serveur pour des questions de sécurité.

³ On appelle `scheme` [ski:m] la partie **http://** ou **https://**

Evidemment, dans une vraie application, on ne définirait pas de telles URLs qui sont des risques potentiels pour la sécurité. Ce n'est ici qu'une facilité pour les tests.

Etape 3

La page d'accueil doit afficher la liste des pays de la compétition par ordre alphabétique. Elle va être complétée au fur et à mesure de l'avancement de vos travaux. Rappel de ce qui a été indiqué précédemment : cette page doit aussi faire figurer un menu de liens vers les URLs demandées dans le sujet (**/filldb/**, **/cleardb/** etc.)






Etape 4

Définissez une URL **/matches/** permettant d'afficher, en respectant le format suivant, la liste des matches joués en phase de poules :

Vendredi 8 septembre 2023							
	France	27	+4				
	Nouvelle-Zélande	13	0				
Samedi 9 septembre 2023							
	Italie	52	+5		Irlande	82	+5
	Namibie	8	0		Roumanie	8	0
	Australie	35	+5		Angleterre	27	+4
	Georgie	15	0		Argentine	10	0
Dimanche 10 septembre 2023							
	etc...						

Etape 5

Définissez une URL **/team-result/?code=<code_equipe>** qui affiche les résultats d'une équipe donnée par son **<code_equipe>**, de la manière suivante :

 FRANCE - Poule A - 1ère - 18 pts				
	08/09/2023	France - Nouvelle-Zélande	Gagné	27-13
	14/09/2023	France - Uruguay	Gagné	27-12
	21/09/2023	France - Namibie	Gagné	96-0
	06/10/2023	France - Italie	Gagné	60-7

Il n'est pas utile de prévoir un lien dans le menu d'accueil mais, par contre, modifiez la **<table>** de l'étape précédente pour que les noms des équipes soient cliquables et mènent vers cette URL (en adaptant évidemment la partie **<code_equipe>**).

Faites aussi en sorte que les noms des équipes adverses de la **<table>** ci-dessus soient aussi cliquables pour mener aux résultats de l'équipe en question.

Etape 6

Les quarts de finale (tour **Q**) font s'opposer :

- Quart **Q1** : le 1^{er} de la poule **C** et le second de la poule **D**
- Quart **Q2** : le 1^{er} de la poule **B** et le second de la poule **A**
- Quart **Q3** : le 1^{er} de la poule **D** et le second de la poule **C**
- Quart **Q4** : le 1^{er} de la poule **A** et le second de la poule **B**

Les demi-finales (tour **D**) font s'opposer :

- Demi **D1** : le vainqueur de **Q1** et le vainqueur de **Q2**
- Demi **D2** : le vainqueur de **Q3** et le vainqueur de **Q4**

La finale fait s'opposer les vainqueurs (tour **F**) de **D1** et **D2**

La "petite finale" fait s'opposer les perdants (tour **P**) de **D1** et **D2**

Codez une URL **/fill-stage/<tour>** qui :

- Vérifie que le **<tour>** (parmi **Q, D, F** et **P**⁴) n'a pas déjà été rempli avec des scores
- Calcule (ou recalcule si pas encore été remplis) les rencontres du tour

Définissez une URL **/set-result/<match>** qui, en fonction du **<match>** (parmi **Q1** à **Q4**, **D1**, **D2**, **FI** (finale) et **PF** (petite finale), affiche la rencontre des deux équipes et demande le résultat du match. A ce stade de la compétition, comme il s'agit de tours à élimination directe, seul le score importe, il n'est plus question de points attribués.

⁴ Pour **Q**uart, **D**emi, **F**inale et **P**etite finale

Si le résultat a déjà été saisi mais que le tour suivant n'a pas été calculé, on peut encore modifier le résultat avec cette URL **/set-result/<match>**

Etape 7



Définissez une URL **/tableau/** pour l'affichage de la page suivante, en le remplissant avec les résultats des quarts, demi et finales. L'exemple donné en page suivante montre les quarts de finale déjà remplis mais ils ne le seront pas au départ. C'est l'appel de l'URL **/fill-stage/<tour>** qui se charge de ce remplissage.

Les URLs **/set-result/<match>** qui menaient jusqu'ici vers l'accueil à la soumission de leur formulaire doivent désormais renvoyer vers cette page.

Un clic sur une des équipes du tableau de la page suivante doit mener vers l'URL qui convient : **/set-result/<match>**

	Galles	14 oct.
	Argentine	

		20 oct.

	Irlande	14 oct.
	Nouvelle-Zélande	

		28 oct.

	Angleterre	15 oct.
	Fidji	

		21 oct.

	France	15 oct.
	Afrique du Sud	