

Projet web

Présentation générale

CIR2 Brest/Caen/Nantes/Rennes - 2022

antoine.bouyer@isen-ouest.yncrea.fr

ayoub.karine@isen-ouest.yncrea.fr

benoit.lardeux@isen-ouest.yncrea.fr

thibault.napoleon@isen-ouest.yncrea.fr

christophe.vignaud@isen-ouest.yncrea.fr



Objectif

Concevoir et développer une application web de gestion de matchs sportifs.

Cette application devra permettre de rechercher un match, de réserver une place en tant que joueur, mais aussi de proposer un nouveau match.

Fonctionnement interne :

- **Partie *front-end*** : recherche, réservation et proposition d'un match
- **Partie *API*** : partie permettant de fournir les informations au front (*API REST*)
- **Communication *front-end* et *API*** : *AJAX*

- Analyse :
 - MCD
 - Maquette du site web (*FIGMA*)
 - Requête associée aux pages présentées
- Conception et développement :
 - HTML
 - CSS
 - JS
 - PHP
 - SQL
- Test
- Recette et mise en place dans le serveur

Cahier des charges

Fonctionnalités

1- Page 1 : Authentification (pas de OAuth2)

- Si le compte existe :
 - E-mail
 - Mot de passe
 - Bouton se connecter

Bonus :

- Gérer mot de passe oublié
- Utilisez les sessions

Cahier des charges

Fonctionnalités

1- Page 1 : Authentification (pas de OAuth2)

- Si le compte existe :
 - E-mail
 - Mot de passe
 - Bouton se connecter
- Sinon bouton créer un compte :
 - Prénom
 - Nom
 - Ville
 - Photo
 - Email
 - Vérification de l'existence de l'email dans la BDD
 - Mot de passe
 - Vérification du mot de passe
 - Bouton s'inscrire

Bonus :

- Lien de vérification envoyé par e-mail
- Captcha

Cahier des charges

Fonctionnalités

1- Page 1 : Authentification (pas de OAuth2)

2- Page 2 : Explorer

- Uniquement les matchs futurs avec possibilité de filtrer :
 - par ville
 - par sport
 - par période (+7j, +15j, +30j)
 - par match complet/non complet
- Pour chaque match, les informations suivantes seront affichées :
 - Sport
 - Ville
 - Nombre de joueurs maximum
 - Date
 - Nombre de joueurs inscrits
 - Heure

Bonus :

- Affichage des différents matchs sur une carte
- Affichage du plus proche (en date) au plus loin

Cahier des charges

Fonctionnalités

1- Page 1 : Authentification (pas de OAuth2)

2- Page 2 : Explorer

- En cliquant sur un match :
 - On voit le détail (organisateur, joueurs avec photo, adresse avec ville, heure de début, durée, prix, ...)
 - On peut s'inscrire et on attend la confirmation de l'organisateur

Bonus :

- Inscription non autorisée si le nombre maximum de joueurs est atteint

Cahier des charges

Fonctionnalités

1- Page 1 : Authentification (pas de OAuth2)

2- Page 2 : Explorer

- En cliquant sur un match
- Mes matchs :
 - § Prochains matchs en précisant si on est organisateur ou joueur ou les deux
 - § Matches passés en précisant si on est organisateur ou joueur ou les deux, statistiques :
 - § Score du match
 - § Meilleur(e) joueur(euse) du match

Bonus :

- Buteurs
- Possibilité de se désinscrire (2h maximum avant l'évènement)

Cahier des charges

Fonctionnalités

1- Page 1 : Authentification (pas de OAuth2)

2- Page 2 : Explorer

3- Page 3 : Organiser

- § Créer un match : type de sport, nombre de joueurs (min/max), adresse (ville, ...), heure, durée, prix, ...
- § Gérer les inscriptions : accepter ou refuser
- § Après le match, saisie des statistiques :
 - § Score du match
 - § Meilleur(e) joueur(euse) du match

Bonus :

- Buteurs en statistiques
- Inviter des joueurs
- Composer les équipes aléatoirement

Cahier des charges

Fonctionnalités

1- Page 1 : Authentification (pas de OAuth2)

2- Page 2 : Explorer

3- Page 3 : Organiser

4- Page 4 : Profil

§ Editer le profil :

§ Age

§ Ville

§ Forme sportive

§ Noter l'application web

§ Mot de passe

§ Photo de profil

§ Statistiques :

§ Nombre de matchs joués

§ Notifications :

§ Organisateur : demande de validation

§ Joueur : inscription acceptée ou refusée

Bonus :

- Nombre de buts en statistiques

Cahier des charges

Fonctionnalités

Interactions client – serveur :

La partie *API* doit permettre de :

- Répondre aux requêtes du *front* (client) en respectant l'architecture *REST* :
 - Récupération de données: *GET*
 - Ajout de données : *POST*
 - Modification de données : *PUT*
 - Suppression de données : *DELETE*
- Interroger la base de données
- Renvoyer des informations au format *JSON* (obligatoire)

Base de données/PHP :

- Toutes les informations doivent être stockées dans une base de données relationnelle modélisée en utilisant les règles de normalisation classique
- Utilisation de *PDO* en *PHP* pour les interactions avec la base de données

Cahier des charges

Technologies à utiliser

Partie front-end :

Le front doit être développé uniquement avec les technologies :



Les bibliothèques suivantes, et uniquement celles-ci, peuvent être utilisées :



Pour l'authentification :

≡ *PHP* possible

Partie API :

- Exclusivement en 
- Utilisation des communications 
- SGBD :  
- échange de données en *JSON*

Déploiement :

Configuration obligatoire de la machine virtuelle :

- *Apache 2*
- *PHP 7.4*
- *Postgresql11 ou MySQL5.7*

Cahier des charges

Contraintes

Mise en page :

Il faut que la mise en page du front prenne en compte un en-tête et un pied de page commun à toutes les pages.

Accès :

L'accès au site web sera via l'adresse *IP* de la *VM*

L'application doit être :

- Responsive
- Évolutive : possibilité de rajouter de nouvelles fonctionnalités
- Ergonomique : facilité d'utilisation, homogénéité des informations, respect des normes utilisées du Web

Le code doit être :

- Correctement architecturé
- Réexploitable : code lisible, code bien commenté, noms de variables/classes/fonctions explicites
- Séparé en plusieurs fichiers (par fonctionnalités)

Cahier des charges

Fonctionnalités bonus

De nombreuses autres fonctionnalités pourront être ajoutées, exemple :

- Ajout d'un script Python d'insertion de données (fortement recommandé)
- Ajout d'un chat
- Météo sur le lieu du match
- Calendrier des matchs
- Itinéraire depuis ma position
- Tutoriel d'utilisation de l'application (avec un carroussel)
- ...

Cahier des charges

Compétences techniques

Utilisées et consolidées :

- Programmation Web :
 - *Front* : HTML/CSS/Bootstrap
 - *API* : PHP
 - Communication client/serveur : AJAX
- Base de données :
 - Modélisation: *JMerise*
 - Requêtes SQL : MySQL ou PostgreSQL
- Serveur Web :
 - Déploiement et gestion d'un serveur Web

Apprises :

- Maquette : *Figma*
- Développement d'une application web complète

Déroulement

Calendrier

Organisation du projet au jour le jour :

Semaine 1					Semaine 2				
Lundi	Mardi	Mercredi	Jeudi	Vendredi	Lundi	Mardi	Mercredi	Jeudi	Vendredi
	<ul style="list-style-type: none"> Mock-ups Modèle BDD Interfaces JS/PHP 	<ul style="list-style-type: none"> Recette MCD + Maquette 	<ul style="list-style-type: none"> Vérification des machines virtuelles par les encadrants Développement de l'application web 	<ul style="list-style-type: none"> Développement de l'application web 	<ul style="list-style-type: none"> Développement de l'application web 	<ul style="list-style-type: none"> Développement de l'application web 	<ul style="list-style-type: none"> Développement de l'application web 	<ul style="list-style-type: none"> Développement de l'application web 	<ul style="list-style-type: none"> Recette fonctionnelle <i>Dépôt du rendu final avant 12h sur Moodle</i>
<ul style="list-style-type: none"> Présentation du sujet, des attendus et des recettes Préparation de la machine virtuelle 	<ul style="list-style-type: none"> Mock-ups modèle BDD Interfaces JS/PHP 	<ul style="list-style-type: none"> Recette MCD + Maquette <i>Dépôt du MCD et de la maquette avant 18h sur Moodle</i> 	<ul style="list-style-type: none"> Développement de l'application web 	<ul style="list-style-type: none"> Développement de l'application web 	<ul style="list-style-type: none"> Développement de l'application web 	<ul style="list-style-type: none"> Développement de l'application web 	<ul style="list-style-type: none"> Développement de l'application web 	<ul style="list-style-type: none"> Développement de l'application web 	<ul style="list-style-type: none"> Recette fonctionnelle

1 seul rendu par groupe

Travail en binôme :

- Chaque étudiant dans le binôme connaît l'ensemble du projet : utiliser un gestionnaire de code (*Git* par exemple)
- Attention à bien se répartir le travail
- Ajoutez une page dans la présentation qui répond à la question : Qui a fait quoi ?

Ressources externes :

- Tous les documents sont autorisés
- Attention à utiliser avec une grande précaution tout document extérieur : site de vulgarisation, forum, code d'autrui

Documentation du projet :

- Au fur et à mesure
- Standardisée
- Livraison de code ou de document :
 - Ne pas attendre la dernière minute pour poster un livrable
 - Préparer des livrables intermédiaires (surtout pour les sources) ;
 - Sauvegarder régulièrement vos données
 - Mise en production :
 - Vérifier votre configuration
 - Faire régulièrement des mises en production pour vérifier le fonctionnement

Format de l'archive :

- Archive ZIP, TGZ, 7ZIP, pas de RAR : projetweb_groupeX.zip (remplacer X par votre numéro de binôme)

Le rendu intermédiaire doit contenir :

- § Un fichier PDF contenant un lien cliquable qui dirige vers la maquette conçue avec Figma (vérifiez si vous pouvez l'ouvrir via une navigation privée de votre navigateur)
- § Le MCD et le MPD

Le rendu final doit contenir :

- Un fichier README (format *markdown*) détaillant l'installation de votre application et les logins et mots de passe de quelques utilisateurs
- Un fichier PDF contenant un lien cliquable qui dirige vers la maquette conçue avec Figma (vérifiez si vous pouvez l'ouvrir via une navigation privée de votre navigateur)
- Un schéma de la base de données (MCD et MPD)
- Les scripts SQL de création des tables et d'insertion du jeu de test
- L'intégralité de votre code commenté
- Vos ressources (images, icônes...)

Attention

Les livrables seront à poster sur l'intranet. Tout retard sera sanctionné (l'heure du réseau fait foi).
Les fichiers au mauvais format ou mauvais nommage seront pénalisés.

Barème indicatif :

- Présentation *MCD* et maquette (rendu intermédiaire) : 20%
- Recette fonctionnelle : 40%
- Évaluation du code (rendu final) : 40%

Remarques :

- Malus possible sur des membres du groupe si l'investissement est jugé trop faible
- Possibilité d'être interrogé durant le projet de façon individuelle
- Plagiat sévèrement sanctionné pour TOUS les membres du/des groupe(s)

Ressources et groupe

Chemin du projet sur Moodle :

Groupes:

Accueil ► Cours ► DEUXIEME ANNEE ► CIR2 ► CIR2 Brest ► Informatique

Généralités

Projet Web CIR2

Projet 2021-2022

 Presentation du Projet Web CIR2

 Groupes de projet et VM

 Rendu intermédiaire

 Rendu final

Exemple de connexion ssh :

Pour user 1 :

ssh user1@10.10.51.71

MDP : user1

Pour user 2 :

ssh user2@10.10.51.71

MDP : user2

Des questions ?