

# Projet - Championnat de ski

Projet en groupe - Temps estimés : 15 jours

Technologies, outils et concept travaillés : PHP, Javascript, POO, MVC, Réécriture d'url, Outil de test d'API, serveur web local, IDE, PhpMyadmin, Mysql, Git, Composer, SASS, Webpack

## Description du projet

Une association gère une compétition de championnat de ski. C'est un championnat dont l'encadrement est réalisé par les membres de l'ESF. Votre objectif est de réaliser un outil permettant de :

- Générer un document Excel qui sera complété lors de la compétition
- Gérer l'affichage des résultats

## Processus de l'application :

- Au départ, l'application doit permettre de créer une épreuve et d'en saisir les compétiteurs
- Quand l'intégralité des participants est saisi, l'application pourra permettre d'exporter un fichier Excel prérempli et qui sera complété par les organisateurs durant la compétition
- Une fois la compétition terminée ce fichier devra être réimporté dans l'application et intégrer les données dans une base de données. L'application ne contiendra donc pas de formulaire permettant la saisie des temps en direct.
- Dans le cas d'un nouvel envoi d'un fichier, les données précédemment saisies sont écrasées

## Base de données :

Données	Définitions	Entrées
Epreuve	Chaque année est organisé une épreuve faisant participer de nombreuses personnes	Nom de la station de ski Date de l'épreuve
Participant	C'est une personne prenant part à la compétition	Nom Prénom Date de naissance Adresse email Photo

Catégorie	Un participant participe à l'épreuve est inscrit dans une seule catégorie. Les catégories peuvent évoluer d'une année à une autre	Nom de la catégorie : M1, M2, M3, Senior, V, Snow, Nouvelle Glisse
Passage	Chaque concurrent qui participe à l'épreuve a droit à deux passages. Chaque passage est donc chronométré. La durée est en minutes/secondes/centièmes	Temps réalisé

Fonctionnalité :

Une fois le fichier complété et importé par les gestionnaires de l'épreuve, l'application devra permettre d'afficher les résultats sous diverses formes :

- **Classement général** : toutes catégories confondues, le classement est complet
- **Classement par catégories** : pour chaque catégorie, seuls les trois premiers sont affichés. Un écart type sera aussi affiché pour chaque catégorie entre le premier et le dernier.

***Le résultat de chaque participant est composé de la moyenne de ses deux passages.***

Les +

Sécuriser votre API avec une authentification

---

## Les étapes :

- Créer le MCD ensemble et confirmer votre MCD par votre coach
- Créer le MLD ensemble et confirmer votre MLD par votre coach
- Créer le modèle Physique de données ensemble et confirmer votre Modèle Physique par votre coach

## Pour le back :

- Mettre en place le router et définir les routes
- Développement des modèles
- Développement des contrôleurs

## Pour le front :

- Réaliser les wireframe de toutes vos pages
- Réaliser l'intégration des pages via Javascript
- Récupérer les données via Ajax.

Durant toutes les étapes Essayer par vous-même

- Fait un point régulier avec les autres apprenants participant au projet
  - Faite un Trello de vos différentes tâches
  - Vous êtes en difficultés ? demander à un apprenant de votre groupe de vous expliquer les bases
  - Tu as compris ? prend le temps d'aider un autre apprenant
  - Tu as finis ? ajoute des fonctionnalités
  - Tous le long du projet versionne ton projet
- 

## Ressources

- [Bonnes pratiques API Rest](#)
  - [Créer une API Rest en PHP](#)
  - [Introduction au DOM](#)
  - [Introduction aux Web Component](#)
- 

Notes pour le formateur

## Activité du matin

- Définition d'une API
  - Méthodes HTTP : GET, POST, PUT et DELETE
  - PHP : Envoyer au navigateur un code de réponse
  - Ajax : fetch, async, wait
  - Ajax : Librairie Axios
  - Javascript : POO
- 

#### Revision #1

Created Wed, Dec 2, 2020 10:09 AM by [Emmanuel Bailly-Salins](#)

Updated Thu, Dec 3, 2020 8:37 PM by [Emmanuel Bailly-Salins](#)