



## W2 – Piscine MVC

Cloud Wac

*Responsables Pédagogiques*

[pedagowac@epitech.eu](mailto:pedagowac@epitech.eu)

## Sommaire

Détails administratifs.....	3
Restrictions.....	3
Jour 01 .....	4
Objectifs de la journée .....	4
Etapes .....	4
Jour 02 .....	5
Objectifs de la journée .....	5
Etapes .....	5
Jour 03 .....	6
Objectifs de la journée .....	6
Etapes .....	6
Jour 04 .....	7
Objectifs de la journée .....	7
Etapes .....	7
Jour 05 .....	8
Objectifs de la journée .....	8
Etapes .....	8

## Détails administratifs

- Le projet est à réaliser seul.
- Les sources doivent être rendues avec BLIH.
- Répertoire de rendu : **Piscine\_MVC\_Cloud\_Wac**

## Restrictions

Le but de ce projet est de faire une plateforme de stockage de fichiers en ligne. Vous devez utiliser le Framework... :

- S'il est prévu que vous fassiez la Piscine C : **Laravel**
- Si vous avez fait la Piscine PHP 2 : **CakePHP**

## Jour 01

### Objectifs de la journée

- Prise en main et installation du Framework.
- Inscription et connexion fonctionnels.

### Etapes

1. Installer le Framework.
2. Créer une page d'inscription avec un formulaire comportant les champs :
  - a. Nom d'utilisateur : username
  - b. Mot de passe : password
  - c. Nom : name
  - d. Prénom : lastname
  - e. Date de naissance : birthdate
  - f. Adresse mail : email
3. Créer une page d'accueil avec :
  - a. Lorsque l'on n'est pas connecté :
    - i. Un formulaire de connexion comportant les champs :
      1. Nom d'utilisateur : username
      2. Mot de passe : password
  - b. Lorsque l'on est connecté :
    - i. Afficher le nom d'utilisateur ainsi qu'un bouton pour se déconnecter.
4. Faire fonctionner les formulaires correctement.
5. Faire des tests, par exemple :
  - a. Inscrire un nouvel utilisateur et se connecter.
  - b. Supprimer cet utilisateur de la base de données.
  - c. Vérifier que l'utilisateur n'est plus connecté.

## Jour 02

### Objectifs de la journée

- Système d'upload de fichiers.
- Système de « CRUD » fichiers.

### Etapes

1. Créer une table dans la base de données, pour stocker toutes les informations relatives aux fichiers qui auront été uploadés, elle vous sera très utile par la suite.
2. Créer sur la page d'accueil une fois connecté, un formulaire permettant d'uploader un ou plusieurs fichiers. Il devra également être possible de glisser-déposer un ou plusieurs fichiers dans une zone définie.
  - a. Vous pouvez utiliser un plugin jQuery.
3. Implémenter un système de « CRUD » fichiers, à travers lequel chaque utilisateur doit pouvoir de manière très sécurisée :
  - a. Afficher la liste de tous ses fichiers.
  - b. Uploader de nouveaux fichiers.
  - c. Renommer ses fichiers.
  - d. Supprimer ses propres fichiers.
4. Implémenter un système de pagination pour afficher en plusieurs fois, la liste de tous les fichiers, s'il y en a beaucoup.
5. Faire des tests, par exemple :
  - a. Essayer d'éditer le nom d'un fichier qui n'existe pas.
  - b. Essayer de poster du code HTML, PHP.
  - c. Essayer les injections de type XSS, SQL.
  - d. Essayer « F5 » lors de l'upload d'un fichier.

## Jour 03

### Objectifs de la journée

- Système de dossiers et de restrictions.
- Partage des fichiers aux autres utilisateurs.

### Etapas

1. Définir votre architecture pour le stockage de tous les fichiers uploadés.
2. Implémenter des restrictions telles que :
  - i. Limitation d'un compte à 50 Mio.
  - ii. Limitation à 10 Mio par fichier.
3. Implémenter un système de CRUD pour permettre aux utilisateurs d'organiser leurs fichiers au travers un système de dossiers.
4. Implémenter un système de partage de fichiers / dossier, soit :
  - a. Public : Tout le monde a accès via une URL.
  - b. Limité : Une liste d'adresses mails (les comptes inscrits ont accès aux fichiers).
    - i. Inclure la possibilité de leur envoyer un mail lors d'un nouveau partage.
  - c. Privé : Personne n'a accès, hormis soi-même, le propriétaire.
5. Faire des tests.



50 Mio



51.200 Kio



52.428.800 octets

<http://fr.wikipedia.org/wiki/Octet>

## Jour 04

### Objectifs de la journée

- Système d'administration des utilisateurs et des fichiers.
- Système de prévisualisation des fichiers médias.

### Etapes

1. Créer une partie réservée à l'administration accessible depuis l'URL :
  - a. « /admin » qui affiche un récapitulatif avec :
    - i. Les 10 derniers inscrits.
    - ii. Les 10 derniers fichiers uploadés.
  - b. « /admin/users » pour tous les utilisateurs.
  - c. « /admin/files » pour tous les fichiers.
  - d. « /admin/<user> » pour tous les fichiers de l'utilisateur <user>.
    - i. Les utilisateurs ne peuvent donc pas se nommer « users » ou « files ».
2. Implémenter un système de type de compte (droits relatifs aux utilisateurs) :
  - a. Utilisateur : L'utilisateur peut uploader des fichiers.
  - b. Administrateur : L'utilisateur a accès à l'URL « /admin » et possède tous les droits.
3. Permettre à un administrateur de bloquer (ou débloquer) un compte.
4. Permettre de visualiser les fichiers médias de nature :
  - a. Audio : Il doit être possible d'écouter une musique en streaming.
  - b. Vidéo : Il doit être possible de regarder un film en streaming.
  - c. Images : Il doit être possible d'avoir un aperçu dans le navigateur.
5. Tester encore. Il doit être possible de revenir en arrière sur les pages de recherche sans avoir à revalider les envois de formulaire.



**Il y aura partout où c'est possible, des systèmes de CRUD et de pagination.**

## Jour 05

### Objectifs de la journée

- Page de contact
- Tests et debug

### Etapes

1. Implémenter un formulaire de contact accessible depuis l'URL « /contact ».
  - a. Il faut être connecté pour arriver sur cette page.
  - b. Le message doit être envoyé par mail à tous les administrateurs.
  - c. On ne peut pas envoyer plus de 2 messages par heure.
2. Tester tout votre site et débbugger.
3. Finaliser le design de votre site d'hébergement de fichiers.