



TP - CV

Contexte

Dans le cadre du cours de PHP nous allons réaliser un projet web.

Le but est de créer plusieurs pages, de faire persister les données tout en suivant les bonnes pratiques.

En **rouge**: prioritaire

En **magenta**: à ne pas prioriser pour le moment

Consignes

Vous devez réaliser un site CV/Portfolio comprenant plusieurs pages et proposant diverses fonctionnalités pour gérer des CV/Portfolio.

Modalités de rendu

Afin d'évaluer la qualité de votre travail vous devez obligatoirement me **rendre un lien Github** du type: <https://github.com/Clemios/Ynov-Php>.

Quelqu'un qui n'a jamais fait de PHP doit pouvoir démarrer votre projet.

Contenu Attendu

Pages du site web

Il doit y avoir au moins:

- Une **page d'accueil** (landing page statique)
- Une page contact
- Une **page CV modifiable** (cf. exemples de code du cours)
- Une **page qui liste des "projets"**
- Une page de login (cf. exemples de code du cours)
- Une page logout (cf. exemples de code du cours)
- Une **page profil**
- Une page **"panneau d'administration"**

Fonctionnalités du site web

Pour que votre projet soit validé, les fonctionnalités suivantes doivent être implémentées:

General

- Votre site doit posséder une **en-tête** et un **pied de page**
- La navigation se fait via un **menu**
- Une fois connecté on voit son nom et prénom sur l'interface

Contact

- La page affiche un **formulaire qui envoie des mails**
- La page affiche une carte de la ville qui vous représente

Authentification

- Possibilité de se **connecter/déconnecter**
- Possibilité d'ajouter des utilisateurs (avec des **rôles**)
- **Utiliser compte google (SSO - OAuth2)**

CV

- **Quand un utilisateur est connecté, il voit ses infos sur le CV**
- Un utilisateur connecté peut **modifier ses informations de CV**
- Un utilisateur connecté peut **customiser le style de la page CV**
- N'importe quel utilisateur peut choisir le CV à afficher
- **N'importe quel utilisateur peut télécharger le CV en tant que PDF**

Projets/Portfolio

- Un **utilisateur connecté** peut ajouter des projets,
- Un **admin** peut customiser le style de la page Projets/Portfolio
- Il est possible de mettre des projets en favoris
- **N'importe quel utilisateur peut faire une recherche parmi les projets**
- **Il est possible d'ajouter des commentaires aux projets**
- **Les projets sont soumis à validation par un admin. Une fois validé le projet est visible.**

Profil

- **Utilisateur connecté** peut modifier ses informations

Panel Admin

- **Possibilité de gérer les utilisateurs**
- **Possibilité de gérer les projets**

“Objets”

Voici le minimum attendu en terme d'objets et de données associées

→ **user:**

- ◆ email
- ◆ first_name
- ◆ last_name
- ◆ password
- ◆ rôle

→ **cv:**

- ◆ title
- ◆ description
- ◆ skills{object-JSON} (skill: title, description, years_of_experience)
- ◆ experiences-external (experience: title, start_date, end_date)
- ◆ educations-external (education: school, start_date, end_date)

→ **project:**

- ◆ title
- ◆ description
- ◆ image

Outils

[Bootstrap](#)

Fournit des classes CSS permettant d’avoir des composants prêts à l’emploi.

Fournit également des classes CSS dites “utilitaires”

Alternative: [Tailwind](#)

[jQuery](#)

Fournit une surcouche à Javascript permettant avec une syntaxe simplifiée d'interagir avec le DOM.

[SweetAlert2](#)

Librairie Javascript qui permet d’avoir de jolies alertes

[Notie](#) / [Toastr](#)

Librairies Javascript permettant de facilement avoir de jolis toasts