



Laboratoire de développement Web

Travail août : site pour ressources.

Peeters Cédric
cedric.peeters@hers.be

Bachelier en Informatique de Gestion
Activité d'apprentissage du bloc 2
Année académique 2024-25

1 Introduction

Ce travail a pour but de vous faire utiliser certaines bases vues dans les cours de l'unité de Web 2.

2 Travail

Votre site permettra aux internautes de consulter une liste de ressources utiles (PDF, liens vers une vidéo, etc.) pour des cours et à d'autres de poster des ressources. Il sera à réaliser en PHP mais vous devrez utiliser du SCSS et du Javascript là où vous le jugerez nécessaire.

Vous êtes libre d'organiser ça en autant de pages que vous le souhaitez tant que ça reste utilisable. Mais vous devez penser Mobile First.

Ce travail peut être fait seul ou en groupe de 2 étudiants. L'examen sera malgré tout individuel et vous demandera de modifier votre projet (par exemple ajouter l'une ou l'autre fonctionnalité et/ou modifier une fonctionnalité existante).

2.1 Énoncé

Votre site en PHP, SCSS, Javascript et Bootstrap respectera les consignes suivantes :

1. La page d'accueil de votre site doit se nommer `index.php` et être disponible à tout le monde. Elle doit lister les cours pour lesquels il existe des ressources. Lorsqu'on clique sur un cours, on peut voir les ressources disponibles pour celui-ci. Mais tant qu'on n'est pas connecté, on ne peut pas consulter ces ressources, on peut donc uniquement les lister.
2. Cette page d'accueil doit présenter chaque cours dans une *card*¹. Cela devra afficher les *cards* les unes en dessous des autres sur mobile. Lorsque c'est visualisé sur un écran d'ordinateur, ça doit afficher les *cards* 3 par 3 (il y aura donc 3 *cards* sur une ligne, puis 3 sur la suivante, et ainsi de suite). Un cours est caractérisé par un titre, un bloc

1. On entend par *card*, une carte utilisée dans Bootstrap.



(bloc 1 par exemple), une section (info ou kiné par exemple). On doit également voir dans la card le nombre de ressources disponibles pour le cours concerné.

3. Toujours sur cette page d'accueil, les cards doivent être triées selon la date de dernier ajout d'une ressource. Donc par exemple si j'ai deux cours : C et Java, et que C a une ressource qui a été ajoutée en juillet 2025 et que Java a une ressource qui a été ajoutée en août 2025, alors Java devra apparaître avant C puisque c'est Java qui a la ressource la plus récente. On tient donc compte de la date d'ajout de la ressource la plus récente pour faire ce tri.
4. Depuis cette page, un utilisateur² doit pouvoir arriver sur une page de connexion où il peut se connecter avec un matricule et un mot de passe. Si un utilisateur entre l'URL d'une autre page sans être connecté, ça doit lui indiquer qu'il doit se connecter et lui proposer un lien vers la page de connexion. Un utilisateur connecté est soit un administrateur, soit un utilisateur normal.
5. Un utilisateur peut se déconnecter à tout moment.
6. Le nom, le prénom et l'avatar de l'utilisateur connecté doivent être présents dans l'entête de chaque page qu'il consulte jusqu'à sa déconnexion. S'il n'a pas d'avatar, une image générique sera mise à la place. Cette dernière est disponible sur Moodle sous le nom "buste.jpg".
7. Un utilisateur normal doit pouvoir poster une ressource pour un cours. Mais cette ressource sera ajoutée au cours concerné uniquement quand un administrateur aura validé cet ajout. Une ressource doit avoir un titre, une date d'ajout, un type (ici on se limite à image jpg, image png, lien URL et PDF) ainsi que le nom et le prénom de l'utilisateur qui l'a postée. La date d'ajout est la date à laquelle l'utilisateur normal a posté cette ressource et non la date à laquelle un administrateur a validé l'ajout.
8. Un administrateur peut donc valider une ressource proposée à l'ajout par un utilisateur normal. Il doit aussi pouvoir refuser l'ajout d'une ressource proposée.
9. Un utilisateur normal doit pouvoir voir un historique des ressources qu'il a proposées. Cet historique doit clairement faire apparaître le statut de la ressource : en attente de validation, validée ou refusée.
10. Un utilisateur connecté, qu'il soit administrateur ou non, doit pouvoir cliquer sur une ressource listée pour un cours et ainsi la visualiser : voir l'image, le PDF , ou le lien URL qui lui doit être cliquable.
11. Un utilisateur connecté doit également pouvoir marquer une ressource comme lue. Et lorsqu'il liste les ressources, on doit pouvoir voir clairement lesquelles ont déjà été lues et lesquelles ne l'ont pas encore été.
12. Les données (citations et utilisateurs) devront être persistantes à l'aide d'une base de données MySQL. Vous êtes libre d'y ajouter toute autre donnée que vous jugées utiles. Ne stockez pas les images ni les PDF en BD, stockez juste leur chemin relatif.
13. Vous devez créer des classes pour l'accès aux données et ce sont ces classes qui seront appelées dans vos pages (par exemple lorsque vous listerez les cours, vous appellerez l'une de ces classes pour avoir la liste des cours). Vous ne devez donc pas avoir de requête directement rédigées dans vos pages.

2. Il n'y a pas de fonctionnalité d'ajout d'utilisateur, vous devrez donc en prévoir à l'avance dans votre système.



14. Le code SCSS et Javascript sera généré via nodejs. Vous aurez un dossier indépendant de votre projet PHP, où seront situés vos sources en js et scss. Un build (obtenu avec Vite) créera les fichiers (js et scss) qui seront utilisés par votre site en PHP.
15. Bootstrap sera installé également via Vite, via l'option B³ détaillée sur le site officiel.

3 Rapport

Écrivez un rapport qui :

1. contient une page de garde avec les informations pertinentes.
2. contient une brève introduction (qui n'est pas une copie de l'énoncé, c'est moi qui l'ai écrit donc je le connais, pas besoin de me le réécrire).
3. explique vos choix d'implémentation (par exemple comment avez vous structuré vos fichiers ? Quelles classes avez-vous créées ? Avez-vous utilisé des sessions ou des cookies et pour faire quoi ? etc.).
4. contient des captures d'écran de votre site avec des explications de ce qu'on y voit.
5. explique les difficultés rencontrées et les solutions pour résoudre ces difficultés.
6. explique comment déployer votre site Web afin de pouvoir l'utiliser correctement.
7. donne le nom d'utilisateur et le mot de passe de deux administrateurs du site et de 3 utilisateurs normaux afin que l'on puisse tester votre site. Pensez donc à mettre ces administrateurs dans votre script de création de BD pour votre projet.
8. contient une brève conclusion.
9. ne contient pas l'entièreté du code source (vu que vous posterez vos fichiers sources).
10. vous êtes libre d'y ajouter ce que vous souhaitez.

Si jamais votre site plante, votre rapport doit préciser quelles erreurs sont rencontrées et à quels endroits.

4 Remise du travail

Vous placerez vos fichiers sources (images comprises si vous en utilisez), ainsi que votre rapport au format PDF dans une archive au format zip⁴ et nommée *NomPrenomWeb.zip* (où *Nom* est votre nom et *Prenom* votre prénom, le tout sans lettre accentuée ni espace de celui qui poste le travail). **N'oubliez pas de mettre tous les fichiers utiles, comme par exemple les fichiers dans lesquels votre site lit des informations, ou encore votre script SQL de création de BD. Merci également de ne pas me remettre l'énoncé dans votre archive, je l'ai déjà.** Si une de ces consignes n'est pas respectée, des points sont retirés.

Votre archive est à rendre pour le mardi 12 août 2025 avant 23h59 via Moodle⁵.

3. <https://getbootstrap.com/docs/5.3/customize/sass/>

4. Le choix de ce format n'est pas anodin.

5. En cas de défaillance de Moodle, l'archive est à retourner par mail.