

Devoir-projet : mini-site

2020-2021

Licence Informatique, semestre 6

Pascal Vanier — Jean-Marc Lecarpentier

Devoir-projet : mini-site

2020-2021

Objectifs

L'objectif du projet est de réaliser un site qui met en oeuvre les principes de développement web vus tout au long du module Technologies Web.

Sujet

Vous devez réaliser, par groupes de 4 étudiant.e.s, un mini-site qui présente des « choses » que vous allez modéliser. On doit donc pouvoir présenter une liste de ces « choses » et voir le détail d'un item.

Dans la suite de l'énoncé, nous désignerons ces « choses » par des « **objets** », mais ce ne sont pas nécessairement des objets au sens propre. Cela pourra être par exemple des livres, loutres, chansons, pays, voitures, pots de chambre, recettes, disques, personnes, événements historiques, arbres, boîtes de camembert, etc...

Préférez un sujet original (en particulier, autre chose que votre page perso, et autre chose qu'un sujet informatique), mais vous ne serez pas jugés sur le choix du sujet. Cependant, ne reprenez pas tels quels les **objets** « poème », « couleur » et « animal » manipulés en cours et en TP.

Contraintes techniques

- Le site devra être hébergé sur le serveur web du département.
- Le code source de votre site devra être déposé sur le *git* de la forge du

département Informatique

- Vous devrez stocker les **objets** et les comptes utilisateur dans une base de données MySQL ou PostgreSQL.
- Vous devrez utiliser l'**architecture MVCR** vue en cours et détaillée lors des TP, avec en particulier une bonne séparation des classes métier (le modèle), de l'affichage (vues), de la gestion des requêtes HTTP (routeur), de la gestion des actions (contrôleurs), de la validation des données (*builders*), et des détails de stockage en base de données.
- Les pages générées devront être **valides HTML5** (pas nécessairement valides CSS, le validateur W3C n'étant pas forcément à jour sur les spécifications).
- Tout le code doit être écrit par vous : vous ne pouvez pas utiliser de frameworks et le plagiat sera sévèrement puni.

Réalisation de base (15 points)

Cette réalisation de base est **notée sur 15**.

Votre site devra avoir les fonctionnalités de base détaillées ci-après.

- Sans authentification, un utilisateur a accès à :
 - la liste de tous les **objets** (mais pas le détail).
 - la page de création de compte.
 - une page à *propos* qui contient :
 - **le numéro de groupe, les numéros étudiants des membres du groupe**
 - liste les points réalisés (en particulier les compléments, c.f. plus bas).
 - la répartition des tâches dans le groupe.
 - les principaux choix en matière de design, modélisation, code, etc...
 - tout ce qu'il vous semble utile de nous signaler.
- Les utilisateurs authentifiés peuvent :
 - voir la page de détail de chaque **objet**.
 - ajouter de nouveaux **objets**.
 - modifier/supprimer les **objets** qui lui appartiennent (mais pas ceux des autres).

Compléments (5 points)

Vous pouvez réaliser jusqu'à 3 compléments (pas plus), **notés sur 5**. Les compléments seront choisis dans la liste ci-dessous (le nombre d'asterisques

indique la difficulté) :

- (*) Une recherche d'**objets**
- Associer des images aux objets (en choisir un seul parmi les trois) :
 - (*) Un **objet** peut être illustré par une image (et une seule, non modifiable) uploadée par le créateur de l'objet.
 - (***) Un **objet** peut être illustré par zéro, une ou plusieurs images (modifiables) uploadées par le créateur de l'objet.
 - (****) Un **objet** peut être illustré par une (ou plusieurs) images (modifiables), uploadées par le créateur de l'objet et l'upload de cette image aura une barre de progression (pensez à tester avec une connexion lente ou avec de grosses images).
- (**) Site responsive
- (*) Tri de la liste des **objets** (par date etc)
- (*) Gestion par un admin des comptes utilisateurs
- (**) Pagination de la liste (avec N **objets** par page)
- (***) Commentaires sur un objet
- (**) Dans le formulaire d'inscription, vérification en temps réel de la disponibilité du login
- (***) Fonctionnalité rester connecté, avec une durée de validité (plusieurs jours par exemple) paramétrable par l'administrateur du site.

La page à *propos* doit indiquer les compléments réalisés.

Critères d'évaluation

Les points suivants seront particulièrement importants pour l'évaluation de votre projet.

- Site fonctionnel sur le serveur des pages étudiants
- **Respect des principes de l'architecture MVCR** vus en CM/TP.
- Qualité du code PHP réalisé.
- Organisation des dossiers et fichiers.
- Qualité de l'ergonomie et de l'accessibilité du site. Le design graphique du site en lui-même n'est pas spécialement pris en compte mais même avec un design très basique on veillera à avoir une présentation claire des pages.
- Non vulnérabilité aux principales failles de sécurité vues en cours (injections SQL et JavaScript, « fuite » de la base de données).
- N'hésitez pas à vous entraider, mais ne copiez pas le code des autres. Nous serons sévères envers les fraudes.

Rendu du devoir

- La date limite de rendu du devoir est le **30 novembre à 23:59:59**.
- Vous devrez :
 - Déposer **régulièrement** votre code sur le *git* de votre groupe sur la forge : <https://forge.info.unicaen.fr/>.
 - Déployer votre site sur le serveur de **chacun des membres** du groupe, à l'URL `https://dev-NUMETU.users.info.unicaen.fr/projet-inf5c-2020/`, **si un étudiant du groupe n'a pas le travail sur son serveur, sa note sera automatiquement de 0**. Ce répertoire a été créé automatiquement pour tout le monde ; **vous ne devez pas le supprimer ou le renommer**. Après la date limite, il sera rendu inaccessible en écriture jusqu'à ce que la correction soit terminée (excepté le répertoire `/users/NUMETU/www-dev/projet-inf5c-2020/upload/`, pour que l'*upload* de fichiers puisse continuer à fonctionner).
- Deux comptes utilisateurs doivent être créés sur le site en ligne avec les logins `vanier` et `lecarpentier` (mot de passe `toto` pour les deux). Si le complément de gestion administrateur a été réalisé, créer aussi un compte admin (mot de passe `toto`).
- Un *dump* de votre base de données (voir la commande [`mysqldump`](#)) sera placé dans le dossier `dump` de votre *git*.

Si vous n'utilisez que votre propre serveur pour développer votre site, **prévoyez de la marge** pour le déploiement sur le serveur du département. Entre la BD, les différences de versions de PHP et la configuration d'Apache, il peut y avoir énormément de problèmes imprévisibles.

Remarques sur l'architecture

Un des objectifs du DM est de vous faire appliquer l'architecture présentée en cours et détaillée dans le gros exercice filé vu en TP. Certains des choix faits dans cet exercice sont destinés à en simplifier l'énoncé : ils ne représentent pas une consigne ferme. Par exemple, si vous regardez les sources du site des poèmes ou des couleurs, vous verrez qu'il y a des variations. Vous pouvez vous aussi tout à fait faire des choix différents.

Cependant, **les principes de l'architecture MVCR doivent être respectés**. Plus vous vous éloignerez de la structure présentée en cours/TP, plus vous vous exposerez à la critique. Réfléchissez bien avant de faire des choix architecturaux trop différents (vous pouvez aussi évidemment en discuter avec nous), et si vous décidez de les faire, **justifiez-les** sur votre page « à propos ».

Remarque pour les spécialistes du PHP : il n'est pas interdit de réutiliser des bouts de code que vous auriez déjà écrits, mais ne reprenez pas tout le super framework perso sur lequel vous travaillez depuis deux ans. C'est très difficile à évaluer, ça prend beaucoup de temps, ce n'est pas très juste par rapport aux autres (car on n'évalue alors pas un simple DM mais le travail de plusieurs mois/années), et on ne peut pas vérifier votre implication personnelle dans le développement. Ça peut être frustrant pour vous, mais ce type de frustration est courant en informatique — et rien ne vous empêche d'intégrer proprement vos idées dans l'architecture demandée.