

# INF3190 - Introduction à la programmation web

TP1 Automne 2021 - Club de lecture

Carl Simard et Johnny Tsheke



# Chapitre 1

## TP1 Automne 2021 – Club de lecture

Ce travail vise l’acquisition des compétences sur les concepts/langages fondamentaux du web.

**Coté client** : L’accent est mis sur le développement coté client avec le html 5, css, Javascript, l’utilisation des frameworks et des multimédias, la conception et le design d’un site web à la fois adaptatif (*responsive*) et accessible (au sens WAI - Web Accessibility Initiative).

**Coté serveur** : Une configuration et un développement de base pour le déploiement du site.

### 1.1 Conception d’un site web

Un club de lecture désire aider les écoles primaires à augmenter la quantité de livres dans leurs bibliothèques. Pour y arriver, le club école désire vendre des livres pour enfant en ligne. Ainsi, il augmentera ses ventes tout en aidant les écoles primaires. Les usagers auront un code fourni par leur école. Ce code pourra être associé, par la suite, au compte de l’usager. À chaque achat, 10% des ventes avant taxes seront ajoutés en crédit dans le compte de l’école afin de leur permettre d’acheter de nouveaux livres.

Pour tester son idée, le club vous demande de lui développer un site Internet prototype selon les spécifications et les précisions données dans ce document. A ce stade, on n’implémentera pas *backend* (coté serveur) des fonctionnalités administratives ni celles du panier d’achat.

### 1.1.1 Spécification fonctionnelles

Le fonctionnement du site doit respecter les exigences suivantes.

- La page d'accueil doit permettre d'afficher de l'information pertinente comme les spéciaux de la semaine avec la liste des livres en rabais. Un administrateur doit être capable d'ajouter des annonces marketing sur la page d'accueil.
- Un livre doit avoir un code de produit, un auteur, un titre, un prix, un rabais (0% par défaut), une description, un niveau scolaire (maternelle à 6e année) et une image.
- L'utilisateur doit être capable de faire une recherche d'un livre par titre.
- L'utilisateur doit être capable de faire une recherche d'un livre par catégorie.
- L'utilisateur doit être capable d'ajouter un livre dans son panier ou le retirer. Dans cette première partie du travail, cela consistera essentiellement à prévoir des boutons à cet effet et afficher un compteur ou le montant total des livres sélectionnés.
- L'utilisateur doit être capable de créer un compte utilisateur afin de voir l'historique de ses achats, mais le compte n'est pas nécessaire afin de faire un achat. Il faut donc prévoir une page d'inscription/création de compte.
- Pour chaque utilisateur, on gardera le nom, le prénom, un courriel, un mot de passe, un numéro de téléphone, une adresse de livraison et le code fourni par l'école (Voir format du code plus bas).
- L'utilisateur doit entrer son courriel et un mot de passe afin de s'authentifier à son compte.
- Pour chaque école, on garde le nom de l'école, l'adresse de l'école, le code de l'école, un courriel et un mot de passe.
- Le compte école doit permettre de consulter le crédit de l'école.
- L'administrateur doit entrer son nom d'utilisateur et un mot de passe afin de s'authentifier avec le compte **admin**.
- L'administrateur doit être capable de gérer (ajouter, modifier et enlever) l'inventaire des livres.
- L'administrateur doit être capable de valider les comptes des écoles.
- L'administrateur doit être capable de modifier les images marketing.
- L'administrateur doit être capable d'ajouter des rabais qui seront affichés avec l'article.

### 1.1.2 Spécifications techniques

La réalisation de ce TP nécessitera de développer des pages web en HTML 5, CSS/SCSS/SASS et Javascript/jQuery/~~Typescript~~. Le développement se fera sous forme d'une application composée essentiellement des pages statiques avec éventuellement des aspects dynamiques en *frontend* (coté utilisateur).

Sauf autorisation explicite, aucun autre langage ne sera accepté pour cette première partie. A part **bootstrap** (<http://getbootstrap.com/>), **jquery** (<https://jquery.org/>) et/ou autres librairies autorisées explicitement, aucune librairie

autre que la votre ne peut être utilisée pour le front-end. ~~Il convient de préciser explicitement qu'aucun thème (bootstrap ou autre) n'est autorisé.~~ On peut se servir des textes, des codes de couleurs, des images et outils disponibles sur les sites suivants :

**Textes** : <http://lipsum.com/>

**Images** : <https://pixabay.com/>

**Icônes** : <https://fontawesome.com/> (icônes gratuits seulement)

**Couleur** : <https://imagecolorpicker.com/>, <https://htmlcolorcodes.com/>,  
<https://www.w3schools.com/colors/>

**Contraste couleurs** : <https://webaim.org/resources/contrastchecker/>

**Design accessibilité** : <https://webaim.org/resources/designers/>, <https://www.w3.org/WAI/>

**Validation accessibilité** : <http://wave.webaim.org/>

- ◊ Les couleurs doivent minimalement respecter le niveau de conformité AA du standard WCGA 2. Ce qui nécessite un ratio d'au moins 4.5 : 1 pour le texte normal et d'au moins 3 : 1 pour du texte large. Un texte large commence à 14 pt (points) et gras ou 18 pt. Pour plus d'informations, voir le site de contraste des couleurs donné ci-dessus et <https://www.w3.org/WAI/WCAG21/quickref/?showtechniques=143#contrast-minimum>
- ◊ Une liste des recommandations pour développer un site accessible est disponible sur <https://webaim.org/standards/wcag/checklist>. On peut aussi trouver quelques exemples sur <https://www.dbswebsite.com/blog/ada-compliance-starts-with-accessible-design/>.

Le code de l'application sera structuré de la manière suivante.

- Les pages HTML seront dans le dossier racine de votre site Internet. Par défaut, le site affiche le fichier `index.html`.
- Tous les liens internes doivent être des URLs relatifs pour permettre la portabilité du site
- Votre propre code CSS sera placé dans des fichiers CSS séparés des documents HTML. Et tous vos fichiers CSS seront dans le sous dossier `css`.
- De même, vos fichiers JS éventuels pour le frontend devront-être placés dans le sous dossier `js`
- Un sous-dossier `images` sera créé pour mettre les images.
- Les bibliothèques autorisées (ex : Bootstrap) peuvent-être installés dans les sous-dossiers énumérés plus haut ou dans leur propres sous-dossiers.
- On peut utiliser une adresse de courriel factice dans les liens des courriels demandés éventuellement dans ce travail.

On pourra tester le déploiement avec l'application `node.js` disponible sur <https://gitlab.com/INF3190/tpapp.git>. La conception et la programmation doit permettre de faire évoluer le site assez facilement. En effet, le TP 2 sera une

extension du TP1 et comportera, notamment, le développement de la partie coté serveur (*backend*).

### 1.1.3 Description de la présentation

Cette section comporte une brève description de la présentation attendue du site. Les aspects visuels et esthétiques sont laissés à votre imagination. Il convient de garder en esprit que la beauté et la convivialité du site font partie de l'évaluation.

Voici quelques spécificités attendues :

- Toutes les pages du site doivent être identifiables par un même design.
- Un menu de navigation sous forme d'une barre horizontale au dessus de chaque page.
- La page d'accueil aura un *carousel* (slideshow) bootstrap composé des diapos qu'on peut faire défiler avec les boutons **Précédent** et **Suivant** situés respectivement à gauche et à droite des diapos. Chacun de ces diapos a une image (on vous laisse le choix de l'image) et un lien distinct vers une page (ex : un livre en spécial) ou un site spécifique.
- Le menu **Le club** conduit vers une page avec une vidéo Youtube, Vimeo ou autre intégrée avec **iframe**. La vidéo est supposée contenir un message du responsable du club, mais on vous autorise d'intégrer n'importe quelle vidéo en lien avec la programmation web. La page contient également d'autres informations du club (adresse de contact).
- Le menu **contacts** conduit vers une pages avec les photos, nom, fonctions et contacts des employés du club (Au moins 5 personnes).
- le menu **Chercher** doit conduire vers la page de recherche.
- Vous pouvez ajouter d'autres menus selon votre design.
- Tous les formulaires doivent respecter au moins les exigences suivantes
  - noms : composé de une à 25 lettres de l'alphabet latin (majuscules et minuscules acceptées)
  - Le code usage a le format suivant : 4 lettres majuscules de l'alphabet latin suivi de 8 chiffres arabes ( les chiffres arabes sont 0, 1, ... 9) et terminé de 2 lettres majuscules de l'alphabet latin. Cette condition est à valider avec une expression régulière.
  - Le code école est composé de 2 lettres majuscules de l'alphabet latin.
  - Téléphone composé de 10 chiffres arabes. Si le premier chiffre est 0, le deuxième chiffre ne peut pas être 0. Si le premier chiffre est différent de 0, le deuxième peut-être 0 ou un autre chiffre. Ces conditions doivent se valider avec une expression régulière.
  - Courriel doit avoir le format d'un courriel.
  - Photo, permet de téléverser (upload) une photo
  - Boutons radios : Au moins un formulaire doit offrir des choix avec des boutons radios. Ces boutons devront être dans un même **fieldset** et au moins un bouton devra-être coché avant de pouvoir soumettre le formulaire.
  - Case à cocher : Au moins un formulaire doit proposer des choix avec des cases à cocher. Par défaut aucune case ne sera cochée, mais il

faudra cocher au moins une case avant de soumettre le formulaire. Pour cela, utiliser du code javascript/jquery en *frontend* pour s'assurer qu'au moins une case est cochée.

A part la photo, tous les champs présent dans un formulaire sont obligatoires et doivent respecter les conditions spécifiées avant de pouvoir soumettre un formulaire. Il vous appartient de déterminer les éléments qui doivent figurer dans chaque formulaire pour respecter les exigences de l'énoncé.

- Sur chaque page, on aura un pied de page contenant les informations suivantes disposées d'une manière appropriée et cohérente au design global du site.
  - Indication que ce site est développé dans le cadre du cours INF3190 à l'UQAM
  - Les nom(s) des développeurs (euses).
  - Le nom de chaque développeur (euse) aura un lien vers son dépôt git du TP. Ce lien s'ouvre dans une fenêtre surgissante (pop-up) grâce une fonction javascript/jquery.
  - Le mot **UQAM** aura un lien : <http://www.uqam.ca> qui s'ouvre dans un nouvel onglet.
  - Le mot **INF3190** aura aussi un lien vers l'url suivant et s'ouvre dans le même onglet que la page courante. <http://www.etudier.uqam.ca/cours?sigle=INF3190>

## 1.2 Consignes et instructions

Cette section contient les informations pratiques sur l'organisation du travail, la remise et l'évaluation du TP 1.

- Essayez au mieux possible de valider les documents pour se conformer aux normes HTML 5, CSS et d'accessibilité.
- Mettre un fichier markdown **README.md** qui explique comment déployer, démarrer et tester votre application.

### 1.2.1 Groupes, collaboration et sauvegarde

Ce travail pratique est à faire en groupe de deux étudiant.e.s. Il faut faire le choix de l'équipe sur moodle avant de commencer.

- Le TP 2 sera aussi dans le même groupe parce qu'il s'agira de la continuité du travail entamé au TP 1.
- L'utilisation d'un dépôt git sur le serveur de l'université (<https://gitlab.info.uqam.ca/>) est requis. Le dépôt sera nommé **inf3190-a21-tp**
- Le dépôt git doit resté privé au moins jusqu'à la fin du cours. Il faut le partager simplement avec le(la) coéquipier(ère).
- Il faudra également donner accès à votre enseignant et aux correcteurs désigné par l'enseignant sur la plateforme moodle.

- L'autre membre de l'équipe créera un dépôt git miroir sur son propre compte et donnera les même accès. Cela garantie à chaque membre de l'équipe de disposer de tout le code source même si l'autre membre abandonne le cours.

### 1.2.2 Barème correction

- ★ **30%** – Les fonctionnalités : Est-ce que les différentes opérations sont présentes ?
- ★ **20%** – Qualité du code : Est-ce que le code est facile à la maintenance ?
- ★ **20%** – Qualité de la conception Est-ce que le couplage et la cohésion de l'application est professionnelle ?
- ★ **20%** – Design et ergonomie : Est-ce que le design est professionnel et convivial ?
- ★ **10%** – Respect de l'accessibilité

Note :

- ◇ Une note de 0 sera attribuée automatiquement à toute l'équipe si l'une des conditions suivantes est rencontrée.
  - le site n'est pas portable
  - le code source n'est pas sur le dépôt git de l'université
  - le dépôt git n'est pas privé
  - le correcteur et/ou l'enseignant n'a pas accès au dépôt git
  - le code n'est pas dans la branche TP1\_A2021
  - on modifie ou on supprime la branche TP1\_A2021 après la remise et avant la fin de la session
- ◇ Une pénalité de 5% sera appliquée par heure de retard de remise.
- ◇ Aucune remise ne sera acceptée par courriel et aucune réponse ne sera réservée à un tel courriel.

### 1.2.3 Remise

Ce travail est à rendre sur moodle au plus tard le **24 octobre 2021 à 23H55**.

La remise se fera sous forme d'un fichier .md de rapport et contenant les liens vers les dépôts git de l'équipe. Le code et l'historique doivent-être identiques sur les deux dépôts parce que la correction sera faite sur un seul dépôt. Le nom du fichier .md à rendre sera constitué des **codes permanents** des membres. Voici un exemple de nom.

- groupe de 2 : ABCD123456-EFGD125634.md
- Travail individuel : ABCD123456.md

Un seul membre de l'équipe pourra faire la remise pour toute l'équipe. Il appartient aux membres de désigner la personne qui va rendre le travail et de s'assurer que cette remise soit faite dans le délai.