

# Programmation Web

---

## Énoncé du projet:

Le but du projet est de construire un site e-commerce. Voici ce qui sera attendu en terme de fonctionnalités:

1. Sans avoir à se connecter, l'utilisateur pourra / verra:
  - a. afficher la liste de tous les articles (avec un ordre par défaut qui sera défini par un administrateur du site) ayant une quantité > 0 **(1)**
  - b. ordonner cette liste par nom de l'article (croissant ou décroissant) ou par prix (croissant ou décroissant aussi) **(1+1=2)**
  - c. voir sur un panneau à gauche de la page une liste de filtre. Il pourra donc filtrer par catégorie de l'article ou par tranche de prix (prix compris entre min et max) **(1+1=2)**
  - d. la liste paginée: on ne pourra pas voir plus de 10 articles par page (le chiffre est petit pour ne pas avoir besoin de créer beaucoup d'articles pour tester). S'il essaye d'accéder à une page supérieure au max de pages qu'il peut avoir, on le renvoie sur la page 1. **(1+1=2)**
  - e. s'il choisit de filtrer et/ou trier la liste, le nouveau résultat revient à la page 1 **(1)**
  - f. cacher ou montrer des filtres **(1)**
  - g. ajouter un article à son panier directement depuis la liste **(1+1/ajax)=2**
  - h. accéder à son panier qui devra montrer tous les articles de son panier avec leurs prix, et la somme totale **(1)**
  - i. supprimer un article de son panier **(1)**
  - j. s'inscrire au site (vérifier qu'il n'a pas déjà un compte, /!\ email unique) en donnant: nom, prénom, email et mot de passe (il devra être crypté) **(1)**
  - k. Se connecter au site avec email + mot de passe (toujours crypté) **(1)**
2. Une fois connecté, l'utilisateur pourra:
  - a. se déconnecter **(1)**
  - b. Valider son panier (/!\ le panier devra être conservé si l'utilisateur vient juste de connecter) **(1)**
  - c. Payer ses articles (un faux paiement avec juste une fausse CB) **(1)**
  - d. Voir la liste de ses commandes (date, nombre d'articles et total de la commande) **(1)**
3. Une fois connecté en tant que compte administrateur, un administrateur pourra:
  - a. Ajouter des articles au site: nom, quantité, catégorie et prix **(1)**

- b. se déconnecter (1)
- c. Il n'aura pas accès au reste du site (1)
- d. définir l'ordre d'affichage de la liste d'articles par défaut (1)

tips:

- pour tout ce qui concerne les filtres et l'ordre des articles, pensez à utiliser le passage des paramètres par l'url (GET)
- pour tous les autres formulaires, passez les paramètres en POST
- Pensez à la session pour la gestion du panier et pour la gestion de la connexion de l'utilisateur
- Organisez bien votre BDD pour vous faciliter la tâche derrière
- Pensez à l'AJAX pour l'ajout dans le panier
- Pensez à jQuery pour cacher / montrer les filtres facilement
- Faites les fonctionnalités petit à petit (ex: d'abord afficher la liste, puis ajouter l'ordre, puis ajouter les filtres...). Sinon, ça vous paraîtra trop difficile.

**/>! Vous devez faire la fonctionnalité 3 sinon on ne peut pas administrer le site**

Notation:

- à côté de chaque fonctionnalité, il y'a les points du barème (23 au total). Ce total de points sera ensuite ramené sur 18 (18 \* nombre de points acquis / 23)
- +1 si vous ajoutez des urls jolies (écriture d'un petit routeur)
- +1 si vous utilisez les namespaces et l'autoloading
- +1 si vous utilisez les exceptions pour vos vérifications
- +(1 à 5) si vous écrivez des tests automatisés (la note dépendra de leurs pertinences)

La note maximale est de 20 :)

Contraintes:

- Backend (serveur) en PHP
- Interdiction d'utiliser un framework (ex: symfony, laravel...)
- Possibilité d'utiliser un framework CSS (bootstrap 4 ou Bulma)
- Un seul fichier devra être modifié pour la connexion à la BDD

**/>!**

- Vous devrez un petit fichier README.txt pour m'expliquer très rapidement comment démarrer votre application (BDD et autre si besoin)
- Pensez à me donner un fichier .sql pour créer et peupler la BDD

**Bonne chance à tous :)**