

PHP 420-P62-SU Travail pratique

Pondération :

30% de la note finale

Livraison (remise) :

- Date de remise : Mardi 22 août 2017 avant 13H.
- Site déployé sur un hébergement public (Hébergement ISI Sentora ou hébergement externe personnel).
- **Archive du répertoire du site** complet déposée dans le dossier : X:\Gilles Pénissard_78_P62_E17\remise_TP. Si une base de donnée est utilisée, l'export SQL intégral doit être présent dans l'archive sous forme d'un fichier .sql.
- **Dépôt git** (git repository) de développement du projet à jour. L'adresse du dépôt doit être indiquée dans un fichier de documentation dans l'archive livrée. Le dépôt git doit comprendre **au minimum** 30 (trente) commits **pertinents**.

Objectif

Réaliser un site simple complet en PHP, sans outils, avec une approche réaliste, en utilisant le système de gestion de version git.

Introduction

Vous devez réaliser un site qui présente un certain nombre d'items (par exemple des articles, des produits, des livres, des comptes, des services, des articles de presse). Quelle que soit leur nature, ces items devront appartenir à différentes catégories répertoriées. L'utilisateur pourra parcourir l'ensemble des items et effectuer une sélection de certains items, persistante au rechargement de page.

Exigences fonctionnelles

Le produit devra comporter les éléments fonctionnels suivants :

1. La gestion d'une liste d'au moins 15 items. Chaque item doit comporter au moins les caractéristiques suivantes:
 - a. nom
 - b. description détaillée
 - c. image
 - d. catégorie
 - e. un autre attribut de votre choix (prix, dimensions, etc)
2. Au minimum 3 catégories d'items
3. Une **vue** de présentation des items (un type de catalogue) permettant de voir
 - a. Tous les items à la fois (toutes catégories confondues)

- b. Les items regroupés, classés ou filtrés par catégorie
4. L'utilisateur pourra enregistrer une sélection d'items (par exemple panier, préférences, favoris) et cette sélection sera persistante (conservée après un changement ou un rechargement de la page).
5. ~~Une connexion/déconnexion de l'utilisateur. Cet état sera bien entendu conservé après un changement de page ou un rechargement de la page. L'utilisateur ne pourra accéder à certaines fonctionnalités que s'il est connecté (par exemple, achat, envoi de message, visite de page privée).~~
6. Un **formulaire riche avec filtrage et validation des champs en PHP**. Le formulaire devra comporter au minimum les champs suivants :
 - a. Trois champs de saisie (input, textarea)
 - b. Un champ sélect, simple ou multiple
 - c. Un champ d'options regroupées (des radios ou des check-box regroupés)

Coté serveur (en PHP donc), **tous** les champs sans exception devront donner lieu à

1. Un filtrage des données
2. Une validation avec des critères réalistes, réalisé en PHP.

Le formulaire ne sera pas accepté tant que l'ensemble des critères ne seront pas remplis. Il sera réaffiché à l'utilisateur avec les données préalablement saisies et des indications de critères à respecter.

Exigences techniques

Le produit doit répondre aux exigences techniques suivantes:

1. Il comportera un système simple de "templating" mis en œuvre en PHP. Cela signifie que les pages du site devront être composées à partir de **blocs PHP-HTML modulaires de type vue (view)**. En conséquence, le code HTML ne sera pas répété dans les différents fichiers de pages. Cette technique fait usage des inclusions de fichiers PHP (*require, include*).
2. La persistance requise au paragraphe des exigences fonctionnelles devra être réalisée coté serveur (par exemple, à l'aide de sessions PHP ou d'une base de donnée). Par exemple la connexion / déconnexion utilisateur réalisée à l'aide des sessions ou la gestion d'un panier virtuel.
3. ~~Le site doit comporter une page d'erreur pour le type HTTP 404.~~

Autres exigences

- Le produit ne devra générer **aucune erreur PHP** dans aucune circonstance.
- Votre produit devra être un tant soit peu agréable à regarder et à utiliser.
- Même si HTML, Javascript et CSS ne sont pas les priorités de ce cours, votre produit ne doit comporter **aucunes erreurs** dans ces 3 langages dans aucune circonstance. En outre, vous devrez respecter les standards (par exemple, localisation des fichiers, nommage) appris dans les cours précédents.

Structure des fichiers du projet

Concernant les fichiers de votre projet, nous suggérons à titre d'exemple l'arborescence minimale suivante (répertoires en gras et entre <>):

- Racine Projet (project root)
 - index.php
 - catalog.php
 - reservation.php
 - ~~confirmation.php~~
 - **<view>**
 - _header.php
 - _head.php
 - _footer.php
 - ...
 - **<common >**
 - _utils.php
 - ...
 - **<data>**
 - ~~database_export.sql~~
 - ~~_db_defines.php~~
 - **<users>**
 - ~~_register.php~~
 - **<products>**
 - ~~_catalog.php~~
 - ~~_product.php~~
 - **<style ou css>**
 - main.css
 - ...
 - **<images>**
 - **<js ou script>**
 - main.js
 - ...
 - ... (autres répertoires que vous jugerez utiles)

Plan de développement

Nous suggérons le plan de développement suivant:

1. Le prototypage du produit (voir ci-dessous).
2. Création de la structure fichier de l'application
3. Respect du développement de l'application selon une architecture s'inspirant de MVC et ne **comportant pas de répétition de code HTML**.
4. [Option] Intégration d'une base de données:

- a. Développement du modèle de données de l'application.
 - b. Création de la base de données MySQL.
 - c. Création d'une couche d'accès aux données (code managers de données)
5. Test et débogage de l'application
6. Optimisation du site

Le prototypage

Pour cette étape, vous pouvez partir d'un site HTML statique s'il en existe un (ex : un ancien travail pratique ou un projet de site HTML statique). Dans ce cas, faites de multiples captures d'écran des différentes pages du site.

A défaut, vous pouvez proposer un prototype de site (réalisé avec photoshop, drawio, illustrator,... ou tout autre application de dessin / prototypage. Vous devez fournir des fichiers dans un format répandu – JPG, PNG, HTML, PDF, WORD)

Base de données

L'utilisation d'une base de données, bien qu'encouragée, n'est pas exigée dans ce travail.

Si vous mettez en place une base de données:

- Vous devez fournir un fichier export (dump) de votre BD comportant aussi bien la structure que les données initiales nécessaires au fonctionnement de votre produit (**fixtures**).
- Les paramètres d'accès à votre base de données doivent être présents dans les fichiers du produit sous forme de constantes PHP (fichier `_DB_DEFINES.php` par exemple).
- Pour un plein potentiel, votre BD pourra gérer tous les éléments d'un schéma CRUD (Create/Read/Update/Delete).