



BTS SIO – Option SLAM

Documentation d'épreuve

Rédacteur	Version	Date	Nb pages
DIETRICH Johann	1.2	31/03/2023	17

**Mise en place d'un système d'affichage,
d'ajout et de suppression d'articles de vente.**

SOMMAIRE

1	CAHIER DES CHARGES	3
1.1	Introduction	3
1.2	Expression fonctionnelle du besoin	4
1.3	Contraintes.....	5
1.4	Gestion des droits d'accès.....	6
2	DESCRIPTION DES ENVIRONNEMENTS	7
3	METHODOLOGIE.....	8
3.1	Méthodologie et versioning.....	8
3.2	Gestion des tests de la solution	8
3.3	Rédaction de la documentation	8
3.4	Gestion de projet	9
4	MISE EN OEUVRE	10
4.1	Page d'affichage.....	10
4.2	Ajouter un article	12
4.3	Supprimer un article	14
5	GESTION DE LA MAINTENANCE (CORRECTIVE / EVOLUTIVE).....	16
5.1	Evaluation de la qualité de la solution.....	16
5.2	Procédure de correction d'un dysfonctionnement	16
6	BILAN DU PROJET.....	17
6.1	Validation des exigences point par point.....	17
6.2	Axes d'amélioration	17
6.3	Compétences acquises	17

1 Cahier des charges

1.1 Introduction

Type de mission
Mission effectuée à l'école durant des heures de cours dédiées, lors d'un projet plus grand nommé Bricobrac.
Contexte
Une société non existante souhaite vendre ces outils variés grâce à un site de E-commerce.
Demande du client
Le client souhaite avoir la possibilité de vendre, ajouter et modifier des articles sur son site.
Budget disponible
0
Outils disponibles
Editeur de code (VS Code) avec langages de programmations : PHP, html, CSS Machine virtuelle Debian 10 Framework CSS Bootstrap

1.2 Expression fonctionnelle du besoin

Liste des fonctionnalités attendues :

Front office

Page d'accueil :

- Page permettant d'afficher des articles

Page d'AJout :

- Ajouter un article a la liste

Page de suppression :

- Supprimer un article de la liste

Back Office

- Ajouter des articles
- Supprimer des articles
- Afficher des articles

1.3 Contraintes

Générales

L'équivalent d'environ deux mois de temps de travail.

Juridiques

Aucune information personnelle enregistrée donc aucune problématique juridique (dans cette partie).

Techniques

Obligation d'utiliser les technologies imposées : PHP, html, CSS

Ergonomique

Le site et les pages ont besoin d'être, responsive et d'avoir une esthétique basique.

1.4 Gestion des droits d'accès

Administrateur
Possède les droits d'accéder à toutes les pages et de supprimer/ajouter des articles

Utilisateurs
Possèdent le droit de regarder la liste d'articles

Visiteurs
Peuvent voir la liste d'article

2 Description des environnements

Environnement de développement

J'ai utilisé une VM Debian 10 avec PHP, Apache. Simplement car c'est la chose avec laquelle je suis le plus à l'aise.

J'ai utilisé une base de données MySQL avec PHPMyAdmin pour l'interface graphique de celle-ci.

3 Méthodologie

3.1 Méthodologie et versioning

Utilisation de GitHub pour faire des sauvegardes et du Versionning du projet, sur une branche principale.

Méthodologie simple, sur principe de CamelCase, de tri de dossier basé sur leur langage de programmation.

3.2 Gestion des tests de la solution

unitaires

Sachant que toutes les pages modifient la liste d'article, il suffit d'avoir la liste d'article parfaitement bonne. A partir de cela, chaque page a simplement besoin d'être testé individuellement et de vérifier sur la page de liste si les actions demandés fonctionnent.

fonctionnel

Une méthode assez simple de test consistant simplement en :

- La création d'un nouvel article
- Regarder si cet article est bien inclus dans la liste
- Supprimer l'article

Un paterne tout simple permettant de voir l'exécution de chacune des pages.

Il est a noter, qu'en cas de problème, la mise en place de variables temporaires est utilisée afin de voir l'emplacement d'un problème.

3.3 Rédaction de la documentation

vous pourriez mettre un exemple de doc, du style un manuel utilisateur pour une fonctionnalité de création de compte dans le backoffice par exemple

manuel user exemple :

- ➔ Mettre une FAQ sur le site web pour montrer à l'utilisateur comment passer une commande
- ➔ Et mettre cette procédure dans ce doc Word pour illustrer cette compétence

3.4 Gestion de projet

3.4.1 Budget

Budget de 0 / personnel

Simple utilisation d'éditeur de texte gratuit et d'utilisation de virtualisation pour servir de serveur hôte.








4 Mise en oeuvre

Pour commencer, il faut déjà avoir la structure de ce que l'on veut faire, donc dans notre cas, le nombre de pages que l'on veut créer, la forme de celles-ci et la forme de la base de données qui justement servira aux autres pages.

Il faut :

- Une base de données avec une table pour stocker les articles
- Trois pages pour que chacune d'entre elles aient une des actions demandées (afficher, ajouter et supprimer)
- Et une certaine visibilité des chacune des pages

Dans notre cas, la base de données et la table ont déjà été donnés donc la voici :

	id	nom	reference	prix_ht	taxe	promotion
 Supprimer	1	marteau de menuisier bois verni	81968453	8.90	20	NULL
 Supprimer	2	marteau massette fibre de verre	80166978	21.90	20	NULL
 Supprimer	3	maillet de menuisier bois	82039106	14.90	20	NULL
 Supprimer	4	marteau arrache-clou	81968500	12.90	20	NULL
 Supprimer	5	tournevis électricien plat	74936295	1.95	20	NULL
 Supprimer	6	tournevis électricien isolé plat	67337361	5.10	20	NULL
 Supprimer	7	tournevis testeur de tension plat	76292503	2.90	20	NULL

Voici donc toutes les informations qu'il va falloir afficher sur une des pages.

4.1 Page d'affichage

La page d'affichage est la plus simple à effectuer, celle-ci ne demande qu'une requête SQL (permettant de retirer des informations de la base de données) et de les afficher de manière propre.

Pour commencer cela, il faut d'abord effectuer une connexion à la BDD grâce au code ci-dessous :

```
<?php
$DB_host = "localhost";
$DB_user = "root";
$DB_pass = "aaaa";
$DB_name = "bricobrac";

try
{
    $conn = new PDO("mysql:host={$DB_host};dbname={$DB_name}", $DB_user, $DB_pass);
    $conn->exec("set names utf8mb4");
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
}
catch(PDOException $e)
{
    echo $e->getMessage();
}
?>
```

Les variables changeant bien évidemment selon les informations de la BDD (nom, mot de passe, utilisateur et host).

La prochaine partie, est le début de l’affichage graphique des articles. Avant même de les afficher, il faut créer la base du tableau qui constituera son affichage. Celui-ci est fait de telle manière :

```
<center>
  <br>
  <table class="table" id="articles">
    <thead>
      <tr>
        <th scope="col">Nom</th> <!-- défini la forme du tableau -->
        <th scope="col">Référence</th>
        <th scope="col">Prix</th>
      </tr>
    </thead>
```

Ce qui nous donne ceci en temps que base du tableau.

Nom	Référence	Prix
-----	-----------	------

A partir de cela, il nous reste plus qu’a prendre les articles de la base de données et de les afficher. Ceci grâce a une fonction foreach, créant une nouvelle ligne du tableau pour chaque article qu’il y a dans la BDD.

```
$data = $conn->query("SELECT * FROM articles")->fetchAll(); /* insère dans une nouvelle
ligne pour chaque colonne dans la bdd d'articles */
foreach ($data as $article) {
    echo "<tr>";
    echo "<th> $article[nom]</th>";
    echo "<th> $article[reference]</th>";
    echo "<th> $article[prix_ht] € </th>";
    if (isset($_SESSION['id'])){
        echo "<th><form action='insertion_panier.php' class='panier' method='post'>
        <button name='valpanier' value='$article[id]'>Ajouter</button></form></th>";
    }
    echo "</tr>";
}
?>
</tbody>
</table>
</center>
```

Tout ceci nous permet d’avoir un affichage complet de tout les articles comme montré ci-dessous.

Nom	Référence	Prix
marteau de menuisier bois verni	81968453	8.90 €
marteau massette fibre de verre	80166978	21.90 €
maillet de menuisier bois	82039106	14.90 €
marteau arrache-clou	81968500	12.90 €
tournevis électricien plat	74936295	1.95 €
tournevis électricien isolé plat	67337361	5.10 €
tournevis testeur de tension plat	76292503	2.90 €
tournevis sans fil	81900760	40.00 €

4.2 Ajouter un article

Pour cette partie-ci, il faut commencer par créer un formulaire remplissable avec les informations nécessaires pour un article (nom, prix, etc...) puis faire en sorte que les informations mises dans ce formulaire sont acceptées et mises dans la base de données.

En premier, donc, créer un formulaire basique comme ci-dessous avec toutes les informations nécessaires :

```
<center>
  <br>
  <label>Ajouter un article</label> <!-- formulaire pour ajouter un article -->
  <br><br>
  <form action='insertion_article.php' class="table" method="POST">
    <div class="form-group">
      <label for="nom">Nom</label>
      <input type="text" class="form-control" name="nom" id="nom" placeholder="Entrez
le nom de l'article" >
    </div>
    <div class="form-group">
      <label for="reference">Reference</label>
      <input type="text" class="form-control" name="reference"
id="reference" placeholder="Entrez la référence" >
    </div>
    <div class="form-group">
      <label for="prix">Prix</label>
      <input type="text" class="form-control" name="prix" id="prix" placeholder="Entrez
le prix" >
    </div>
    <div class="form-group">
      <label for="taxe">Taxe</label>
      <input type="text" class="form-control" name="taxe" id="taxe" placeholder="Entrez
la taxe (20 de base)" value=20 >
    </div>
    <div class="form-group">
      <label for="promotion">promotion</label>
      <input type="text" class="form-control" name="promotion" id="promotion"
placeholder="Entrez la promotion (si il y en a une)">
    </div>
    <div class="form-check">
      <input type="checkbox" class="form-check-input" name="nouv" id="nouv">
      <label class="form-check-label" for="nouv">Nouveauté ?</label>
    </div>
    <br>
    <button type="submit" class="btn btn-primary">Ajouter</button>
  </form>
</center>
```

Il est intéressant de noter que chaque « input » donc entrée d'information de l'utilisateur est faite de sorte a avoir un nom et id unique, permettant de récupérer les résultats individuellement pour les insérer dans la bonne partie de la base de données.

Tout ceci nous donne ce formulaire :

Ajouter un article

Nom

Reference

Prix

Taxe

promotion

☐ Nouveauté ?

A partir de ce moment, dès que l'on cliquera sur le bouton « ajouter », on sera redirigé sur une autre page temporaire qui permet simplement d'appeler la fonction qui insèrera les données dans la BDD. Celui-ci ressemble à ça.

```
<?php
session_start();
include_once "connection.php";
include_once "fonctions.php";

if(!empty($_POST)){ /* vérifie si le formulaire a été envoyé */
    insertion_articles($conn);
}
?>
```

(la variable \$conn correspond aux informations nécessaires a se connecter la BDD)

Donc a partir de cela, il faut créer la fonction « insertion_articles » permettant, comme le nom l'indique, d'insérer des articles dans la BDD.

```
function insertion_articles($conn)
{
    $sql = "INSERT INTO articles (nom, reference, prix_ht, taxe, promotion, nouveaute)
VALUES (?, ?, ?, ?, ?, ?)"; //Insertion des données
    $conn->prepare($sql)->execute([$_POST['nom'], $_POST['reference'], $_POST['prix'],
$_POST['taxe'], $promo, $nouv]);
    header("location:articles.php");
    exit;
}
```

Fonction assez simple utilisant la commande « insert into » pour insérer dans la base de données les informations reçues. Celle-ci redirige également vers la page montrant tout les articles après avoir été exécuté.

4.3 Supprimer un article

Pour cette partie-ci, il faudra donc créer une autre page, un sélecteur d'article (pour choisir lequel supprimer) et une fonction permettant de justement supprimer l'article et toutes ses informations de la base de données.

```
<center>
<br><br>
<label>Sélectionnez un article à supprimer</label>
<br>
<form class="tablemod" method="POST">
  <select name="select" class="form-select" aria-label="Default select example">
    <option selected>Articles </option>
    <?php
      require "connection.php";
      $data = $conn->query("SELECT * FROM articles")->fetchAll();
      foreach ($data as $row) /* crée une ligne pour chaque article, avec le nom de
l'article a chaque fois */
      {
        echo "<option value=$row[id]> $row[nom] </option>";
      }
    ?>
  </select>
  <br><br><button type="submit" class="btn btn-primary">Supprimer</button>
</form>
</center>
```

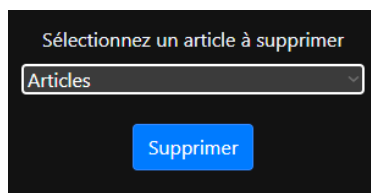
Le sélecteur d'article fonctionne de telle manière : Il commence par faire la forme du sélecteur avec une option de base (Articles), ensuite il fait une boucle foreach mélangé a une requête SQL qui permet d'afficher tout les noms de tous les articles en tant qu'options afin de les sélectionner.

Ensuite, il suffit de récupérer l'ID de l'article sélectionné et de faire une requête SQL afin de le supprimer :

```
$sql = 'DELETE FROM articles WHERE id=:id';
$stmt = $conn->prepare($sql);
$stmt->execute(['id' => $_POST['select'],]);

header("location:articles.php");
```

Grâce a ça, on peut maintenant supprimer des articles.

The screenshot shows a web form with a title "Sélectionnez un article à supprimer". Below the title is a dropdown menu with "Articles" selected. At the bottom of the form is a blue button labeled "Supprimer".

Sélectionnez un article à supprimer

- Articles ▼
- Articles ▲
- marteau de menuisier bois verni
 - marteau massette fibre de verre
 - maillet de menuisier bois
 - marteau arrache-clou
 - tournevis électricien plat
 - tournevis électricien isolé plat
 - tournevis testeur de tension plat
 - tournevis sans fil
 - jeu de tournevis
 - tournevis cruciforme
 - jeu de tournevis torx
 - tournevis boule cruciforme
 - scie de carreleur
 - lot de 2 lames pour scie à métaux
 - scie à métaux
 - scie égoïne de charpentier
 - boîte à onglet manuelle
 - scie japonaise
 - scie à bûche ▼

5 Gestion de la maintenance (corrective / évolutive)

5.1 Evaluation de la qualité de la solution

La solution est simple mais efficace, de simples pages effectuant des requêtes SQL selon des informations demandés, pour l'instant visible et utilisable par tout le monde.

5.2 Procédure de correction d'un dysfonctionnement

Isolation de la page problématique.

Vérification d'emplacement de problème à l'aide d'une variable.

Résolution du problème.

6 Bilan du projet

6.1 Validation des exigences point par point

Capacité d'afficher tout les articles de la BDD

Capacité de créer un article et l'insérer dans la BDD

Capacité de supprimer un article et le supprimer de la BDD

Tout ceci de manière propre et visible

6.2 Axes d'amélioration

Potentiel optimisation avec les fonctions.

6.3 Compétences acquises

Création d'une machine virtuelle hôte

Utilisation de Base de données dans le cas d'un site web

Utilisation de Framework CSS (Bootstrap)