Table des matières

[Remerciements 2](#_Toc82495293)

[I. Introduction 3](#_Toc82495294)

[II. Résumé du projet 4](#_Toc82495295)

[III. Cahier des charges 5](#_Toc82495296)

[1. Présentation du projet 5](#_Toc82495297)

[2. Public visé 5](#_Toc82495298)

[3. Aspects Ergonomie et Graphisme 5](#_Toc82495299)

[4. Aspects Fonctionnels et Techniques 6](#_Toc82495300)

[IV. Spécifications fonctionnelles et techniques 8](#_Toc82495301)

[A. Introduction 8](#_Toc82495302)

[B. Spécifications fonctionnelles 9](#_Toc82495303)

[C. Spécifications techniques 12](#_Toc82495304)

[V. Réalisations et compétences développées 15](#_Toc82495305)

[A. Maquettage du site web 15](#_Toc82495306)

[**1.** Extrait de la maquette **Desktop** 16](#_Toc82495307)

[B. Modélisation et conception de la base de données 17](#_Toc82495308)

[1. Analyse de la situation existante et des besoins 17](#_Toc82495309)

[VI. Situation de travail 29](#_Toc82495310)

[VII. Bilan et Conclusion 32](#_Toc82495311)

[A. Bilan 32](#_Toc82495312)

[B. Conclusion 33](#_Toc82495313)

Remerciements

Tout d’abord, je tiens à remercier et à témoigner toute ma reconnaissance aux personnes suivantes, pour leur dévouement et leur soutien tout au long de cette formation :

* Mes référents de formation Messieurs **Didier Bonneau** et **Olivier Loeb** pour leur bienveillance, leur sympathie et pour avoir partagé avec nous une partie de leur savoir-faire et de leurs expériences professionnelles.
* Toute l’administration de l’AFPA de Créteil et l’ensemble des formateurs pour leur coopération professionnelle.

1. Introduction

Dans le cadre de la formation et de la validation du titre professionnel Développeur Web et Web Mobile à l’Afpa, il est demandé aux étudiants d’effectuer un stage de deux mois en entreprise afin de découvrir la réalisation de projets informatique en milieu professionnel.

N’ayant pas obtenu un stage en entreprise, J'ai donc choisi, comme projet personnel, la création et le développement d'un site Web en m'appuyant sur des sites de bibliothèque

Ce projet me permet de mettre en pratique mes connaissances et mes compétences professionnelles au travers d’un dossier de projet ayant pour finalité la conception et le développement d’un site web avec un design responsive.

Ainsi, l’objectif est de développer un site internet de visualisation de livre dans la bibliothèque en mettant à la disposition de l’administrateur des outils de base pour gérer tout le contenu du site, mais également mettre à la disposition des internautes une interface pour voir les livres, afficher les détails des livres.

Pour mener à bien ce projet, j’ai tout d’abord listé et analysé les exigences requises, afin d’obtenir une expression précise des besoins. Cette analyse m’a permis de développer plus efficacement les différentes fonctionnalités en utilisant le Framework PHP.

Après avoir effectué des tests dans le but de comparer les exigences attendues aux résultats obtenus et d’améliorer ces derniers, j’ai fait le bilan du projet avant de conclure.

1. Résumé du projet

Dans le cadre de ma formation de Développeur Web et Web Mobile à AFPA, j’ai mis en œuvre le projet du site dénommé **BIBLIOTHEQUE.**

Il permet à mes clients, dans le cadre de leur lecture, de pouvoir voir s’il y a les livres dont ils ont besoin dans la bibliothèque.

Le site **Bibliothèque**, la bibliothèque interactive, doit :

* Visualisation des livres ajouter sur la page d’accueil du site en présentant les produits les plus vendus, les nouveautés, la boutique et également son équipe.
* Présenter un tableau de bord à l’administrateur du site lui permettant d’ajouter, d’éditer, de visualiser ou de supprimer des livres.

**Compétences développées :**

Dans ce projet j’ai :

* Maquetté l’application web via l’outil **pencil**
* Réalisé une interface utilisateur web statique et adaptable
* Développé une interface utilisateur web dynamique
* Développé les composants d'accès aux données
* Développé la partie back-end de l’application web

1. Cahier des charges

Afin de cadrer le projet **BIBLIOTHEQUE**, j’ai réalisé avec ma cliente le cahier des charges qui relate l’ensemble de ses attentes et besoins. Elle a exprimé les besoins suivants :

* Développer la présence en ligne de sa bibliothèque
* Permettre à ses clients de visualiser ses livres, faire une réservation en ligne.
* Gérer ses produits et réservation dans un back-office sécurisé.

1. Présentation du projet

BIBLIOTHEQUE, est un projet de site complet qui permet à une bibliothécaire de développer son activité sur le web.

Le site web donnera une meilleure visibilité à bibliothèque et lui permettra de vendre ses produits en ligne.

1. Public visé

Sa clientèle actuelle est tout publique.

1. Aspects Ergonomie et Graphisme

Les différentes couleurs et choix typographique permettront à un utilisateur de repérer les fonctionnalités qui s’offrent à lui.

Le code des couleurs principales du site est :

1. **Maron clair**
2. **Noir**

Le site doit rester aéré et s’adapter au périphérique sur lequel il est consulté. La mise en page du site doit faciliter au maximum la navigation à l’aide d’une présentation claire et intuitive

1. Aspects Fonctionnels et Techniques
2. Arborescence du site
3. Description fonctionnelle

|  |  |  |
| --- | --- | --- |
| **Objectif** | **Description** | **Contraintes** |
| Pouvoir mettre à jour le contenu du site sans retoucher le code source des pages | Permettre à l’administrateur du site  D’ajouter, d’éditer, de visualiser ou de supprimer du contenu en n’utilisant que la partie front-end du site. | Restreindre ces actions aux administrateurs |
| Permettre aux visiteurs de visualiser les livres | Un visiteur non connecté pourra naviguer et regardé les livre que la bibliothèque dispose |  |
| La navigation doit être simple et rapide. | Les utilisateurs pourront trouver, de façon intuitive et fluide, les produits désirés. |  |
| Le site doit être sécurisé | Le site ne doit pas être vulnérable. Il doit être protégé d’utilisations malveillantes : injection de script, de requêtes... | Toute requête SQL doit être sécurisée  Toutes les entrées utilisateur doivent être filtrées |

1. Contenus présents sur le site

Chaque page du site doit avoir un titre.

Les livres qui sont dans la bibliothèque sont classés dans différentes catégories : action, aventure, horreur …

Chaque livre est illustré par une photo.

1. Prestations attendues

|  |  |
| --- | --- |
| Spécifications techniques et fonctionnelles | 12 juillet |
| Charte graphique et Adaptation web | 15 juillet |
| Maquette du site | 23 juillet |
| Développement Front-end et Back-end du Site web complet | 27 août |

1. Spécifications fonctionnelles et techniques

## Introduction

Cette partie présente les spécifications fonctionnelles et techniques du projet de site « **BIBLIOTHEQUE »**.

* Les spécifications fonctionnelles ont pour but de décrire :
* L’ensemble des fonctionnalités du site web
* Les objets manipulés
* Leurs buts et leurs principes de fonctionnement
* Les écrans utilisateurs mettant en œuvre les fonctionnalités de l’application
* Le but, le type et le caractère obligatoire de chacun des champs présents sur les écrans de saisie, ainsi que les actions possibles à partir des écrans.
* Les spécifications techniques elles, répertorient tous les aspects techniques utiles au projet, comme les contraintes matérielles, logicielles, humaines et sécuritaires.

Elles ont pour but de décrire :

* Les environnements matériels et logiciel
* La mise en œuvre de l’application
* Les exigences de programmation
* Le déploiement de l’application
* Les éléments de sécurité mis en place

## Spécifications fonctionnelles

* Utilisateurs

Tout visiteur du site peut voir les livre enregistré par la bibliothécaire.



* Authentification de la bibliothécaire

Pour ce connecté la bibliothécaire doit connais sont Identifiant et son mot de passe.

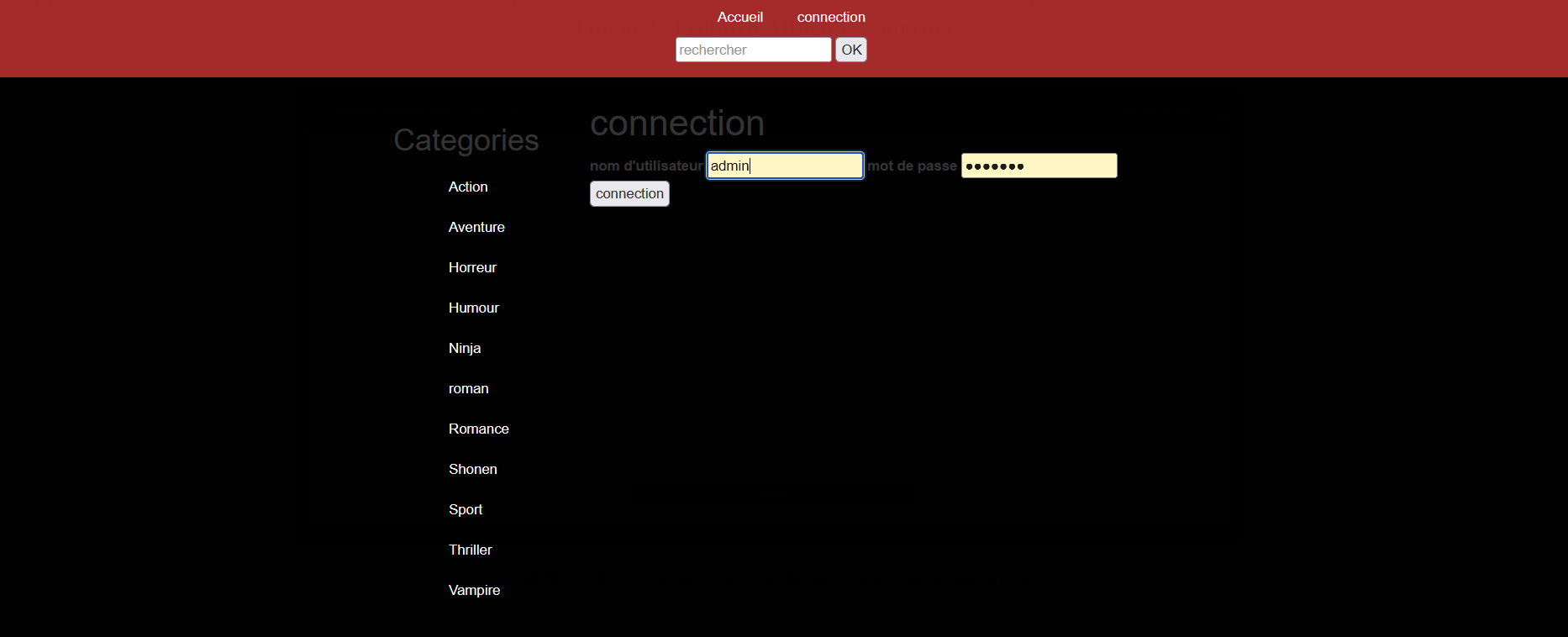


Figure 4: Formulaire de connexion

* Administration du site

Tous les utilisateurs ont accès au livres présent.

Par contre, seuls l’administrateurs du site auras accès a plusieurs onglets.

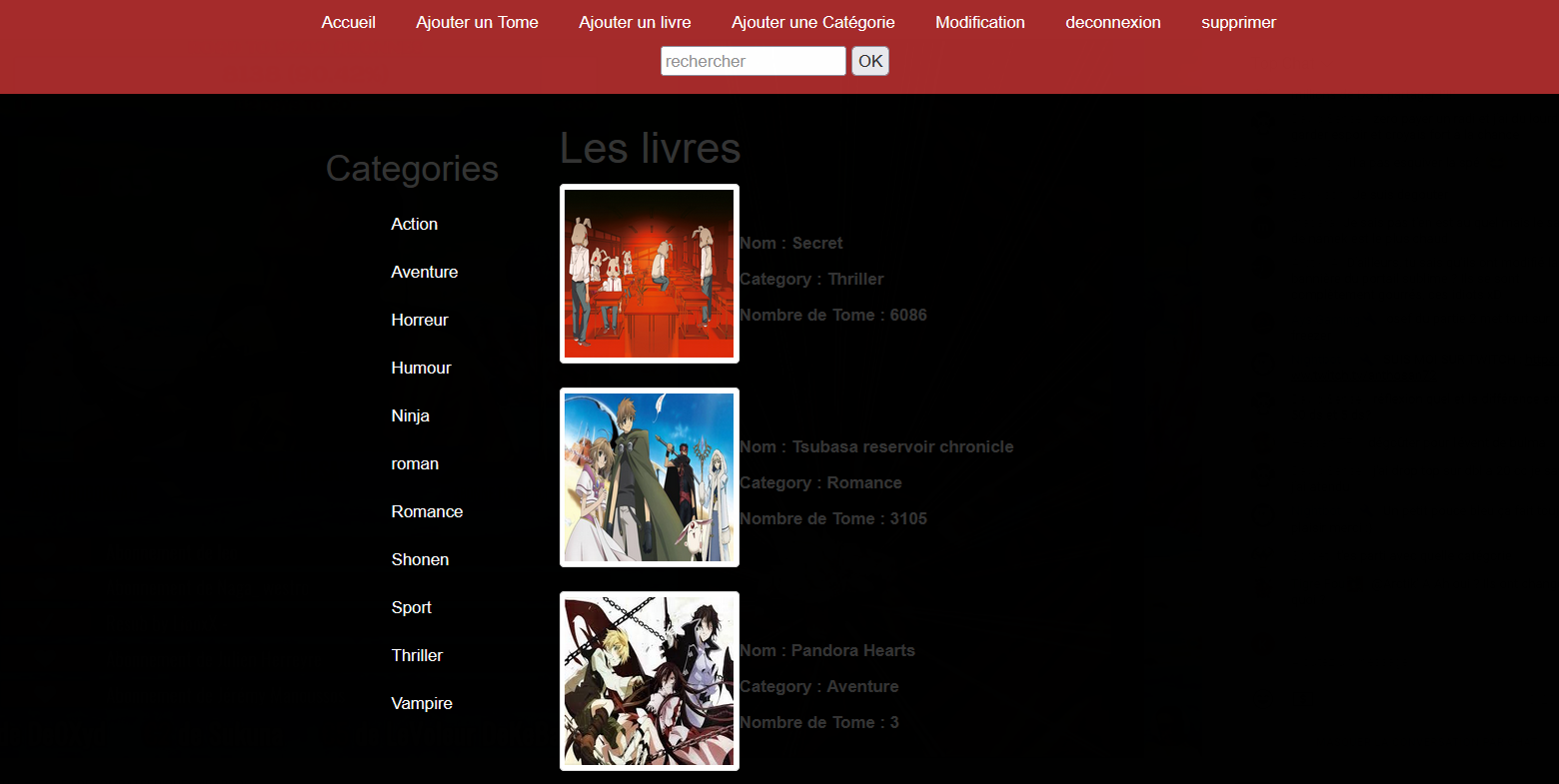


Figure 5 : espace bibliotécaire

Pour l’ajouter des livres dans la bibliothèque interactive, la bibliothécaire doit se connecter et cliquer sur l’onglet Ajouter un livre.

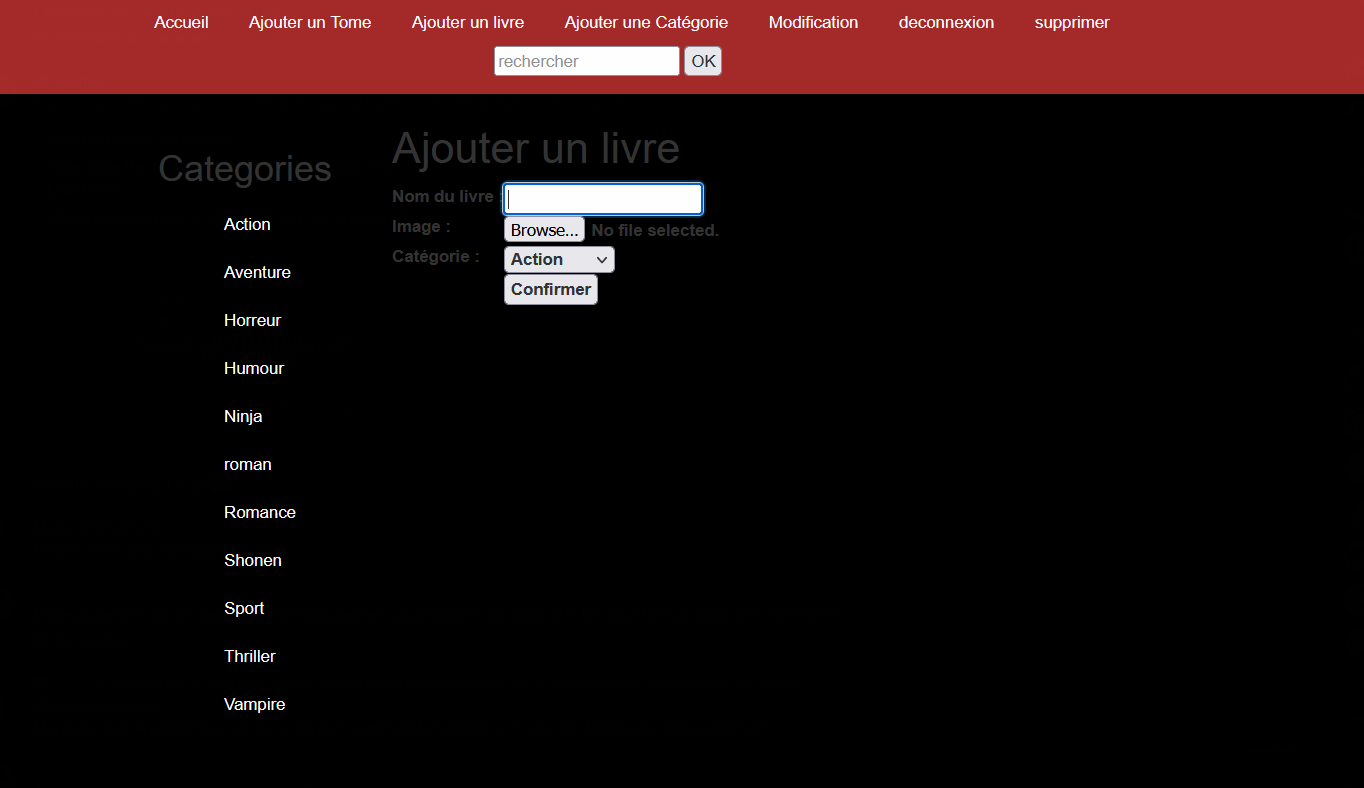


Figure 6 : Ajouter un livre

Editer un livre

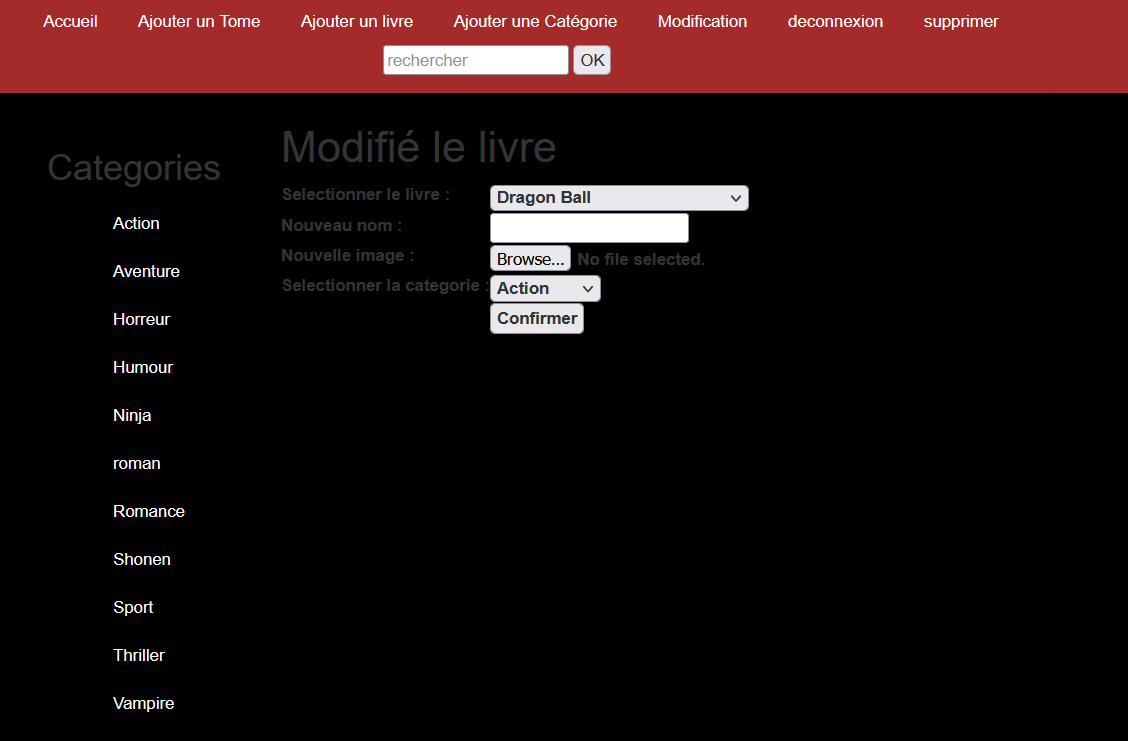


Figure 7 : Editer un produit

## Spécifications techniques

* Outils

J’ai utilisé plusieurs outils matériels et logiciels pour l’analyse et la conception du site web « **bibliothèque**» :

**Environnement matériel** :

|  |  |
| --- | --- |
| Pc bureaux | Acer Nitro N50-600 |
| RAM | 16,00 Go |
| Processeur | Intel® Core™ i5-8400U CPU @ 2.80GHz 2.80 GHz |
| Système d’exploitation | Windows 10 |
| Deuxième écran | HP 24’’ |

**Environnement Logiciel:**

* **Vs Code**

J’ai utilisé l’éditeur de code **Visual Studio Code**. C’est un éditeur open source et multiplateforme qui fonctionne sur macOS, Linux et Windows. La plate-forme prend en charge presque tous les langages de programmation parmi lesquels HTML, JavaScript, CSS, et PHP… Un débogueur intégré et GIT sont inclus dans Vs Code. Ce qui me permet de « commiter » et de « pousser » chaque module ou tache traités pour « versionner » et sauvegarder mon code dans GitHub.

Avec quelques plugins installés, j’ai pratiquement toutes les fonctionnalités d’un bon IDE.

**WampServer** est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l’aide du serveur **Apache2**, du langage de scripts **PHP** et d’une base de données **MySQL**. Il possède également **PHPMyAdmin** pour gérer plus facilement les bases de données.

* **Outil collaboratif :** **Trello**
* **Navigateurs :** Chrome, Firefox et Microsoft Edge. Ces navigateurs sont les plus utilisés pour tester le fonctionnement de l’application.
* **Conception et traitement de l’image :** Gimp 2.10.22, Adobe Photoshop Express.
* Langages informatiques

L’application est développée en **PHP** orienté objet (PHP POO) version 7.4.9 avec utilisation d’un Framework MVC et Html/Css3 + JavaScript/JQuery/Ajax.

Le « PHP: Hypertext Preprocessor », est un langage de script côté serveur, utilisé pour le développement de sites Web statiques et dynamiques ou applications web.

**Pourquoi Le PHP Orienté Objet ?**

PHP est un langage de programmation open source utilisé par une large communauté de développeurs et il est très bien documenté. Il possède de nombreuses bibliothèques fournies par la communauté.

PHP a une très grande famille de divers frameworks et CMS : Symfony, Laravel, WordPress…

* Architecture de l’application

Afin de garantir un fonctionnement correct, une meilleure performance et une maintenance facile, le choix de l’architecture adéquate dans la phase de conception de toute application est primordial.

Dans cette logique, j’ai opté pour l’architecture dite MVC.

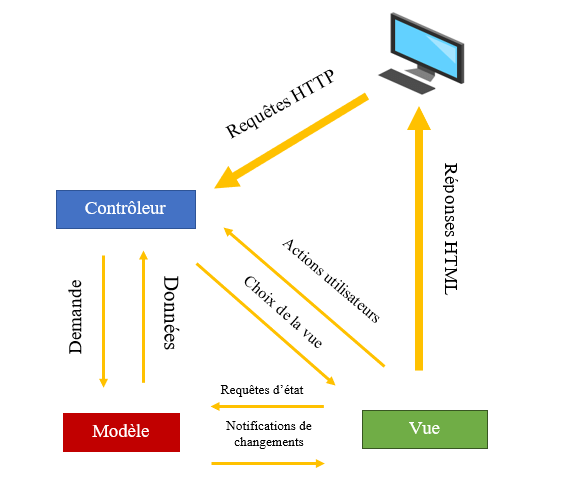


Figure 7 : Architecture de l’application : MVC

**Fonctionnement du pattern MVC**

MVC (Modèle – Vue – Contrôleur) est un patron de conception (ou *Design pattern* en anglais) qui permet de bien organiser son code source. Il nous aide à savoir quels fichiers créer, mais surtout à définir leur rôle. Le but de MVC est justement de séparer la logique du code en trois parties que l'on retrouve dans des fichiers distincts.

* **Modèle :** cette partie gère les données du site. Son rôle est d'aller récupérer les informations « brutes » dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur. On y trouve donc entre autres les requêtes SQL.
* **Vue :** cette partie se concentre sur la « présentation ». C’est par celle-ci que l’utilisateur interagit avec l’application. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu'elle doit afficher. On y trouve essentiellement du code HTML mais aussi quelques boucles et conditions PHP très simples, pour afficher par exemple une liste de messages.
* **Contrôleur :** cette partie gère la logique du code qui prend des décisions. C'est en quelque sorte l'intermédiaire entre le modèle et la vue : le contrôleur va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue. Le contrôleur contient exclusivement du PHP. C'est notamment lui qui détermine si le visiteur a le droit de voir la page (gestion des droits d'accès).

Cette architecture communément mise en place dans de très nombreux projets, permet une meilleure sécurisation du code, une maintenabilité plus aisée et d’assurer la cohérence entre les données et les traitements.

* Base de données



Pour la gestion des données du site, j’ai choisi le SGBDR (Système de Gestion de Base de Données Relationnelles) MySQL pour créer et gérer la base de données.

MySQL est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, PostgreSQL et Microsoft SQL Server.

MySQL fonctionne sur de nombreux systèmes d'exploitation différents, incluant FreeBSD, Linux, Mac OS X ou Windows.

1. Réalisations et compétences développées

## Maquettage du site web

Une maquette de site Web, ou **mockup** en anglais, est une représentation graphique simplifiée de la façon dont on veut agencer son site Web.

L’objectif unique est de montrer ce qu’on envisage, avant de commencer le développement du site web.

C’est une étape cruciale pour élaborer un site web agréable à utiliser et efficace en conversion.

J’ai utilisé l’outil **Pencil** pour créer mes maquettes desktop.

**Pourquoi Choisir Pencil ?**

**Pencil** est un outil de design collaboratif qui permet de prototyper les interfaces graphiques pour les UX/UI designers. Il aide notamment à concevoir sites web, applications, interfaces utilisateur. (Le Journal du Net)

Cet outil est une solution de design d’interfaces collaboratif et dynamique, qui peut être alimenté et partagé au sein d’une équipe de designers.

Il utilise une bibliothèque de composants graphiques pour une approche « design system » ; il permet également de créer des composants (menu, bouton, carte...) pour utiliser les mêmes propriétés du site (police, taille, couleurs…). Ceci afin de garder une certaine cohérence et une harmonie générale.

Il fait aussi appel à différents plugins réalisés par une large communauté de designers.

### Extrait de la maquette **Desktop**

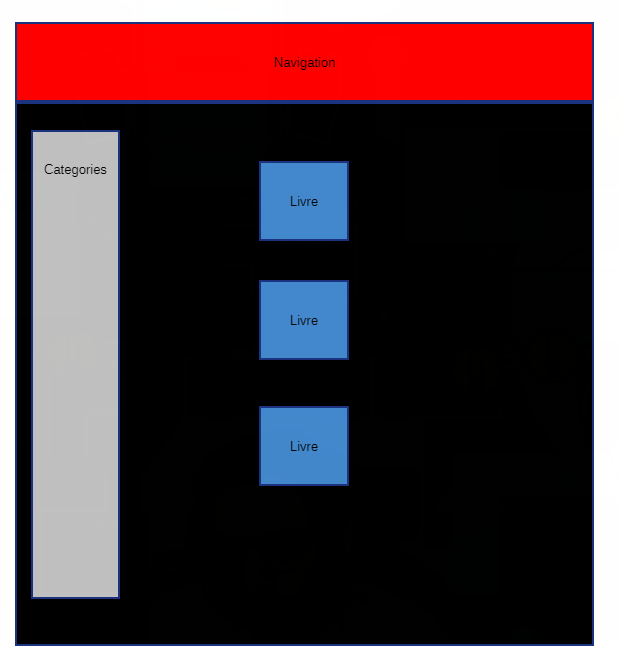


Figure 8 : Maquette Desktop

## Modélisation et conception de la base de données

### Analyse de la situation existante et des besoins

La pertinence de la base de données par rapports aux usages dépend de la qualité de l’analyse de l'existant et des besoins.

Dans ce projet de site bibliothèque les livres sont vue par les user et la bibliothécaire.

Nous avons deux types d’utilisateurs :

* **Visiteur/Utilisateur non connecté :**

Il a accès à une pages du site (Accueil) Il pourra ajouter des produits dans son panier mais ne pourra valider sa commande.

* **Administrateur/Utilisateur connecté avec le rôle « ADMIN » :**

Il est le seul à avoir accès au backoffice du site. Il pourra gérer tout le contenu du site : les ajouter des livres et tomes, les modifications etc...

**Les données**

Les livres sont classés par catégorie (Action, aventure, horreur…).

Chaque livre a un nom, une photo pour l’illustrer…

La Bibliothécaire peut modifier, ajouter, supprimer des livres, tomes ou catégories

Ces données m’ont permis de modéliser une base de données

**Le dictionnaire de données :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entité | Nom données | Type | Taille | Commentaire |
| Category | ID | Int (AUTO\_INCREMENT) |  | Clés primaire |
| Category | VARCHAR | 255 | Catégorie des livres |
| manga | ID | Int (AUTO\_INCREMENT) |  | Clés primaire |
| name | VARCHAR | 255 | Nom du livre |
| picture | VARCHAR | 255 | Image du livre |
| category | VARCHAR | 255 | Categorie du livre |
| user | id | Int (AUTO\_INCREMENT) |  | Clés primaire |
| reference | VARCHAR | 255 | Mot de passe |
| Tome | ID | Int (AUTO\_INCREMENT) |  | Clés primaire |
| name | VARCHAR | 255 | Nom du tome |
| manga | VARCHAR | 255 | Nom du livre |
| picture | VARCHAR | 255 | Image du tome |
| quantite | VARCHAR | 255 | Le nombre de tome |

Voilà ce que ça donné après avoir fait le dictionnaire de donnée

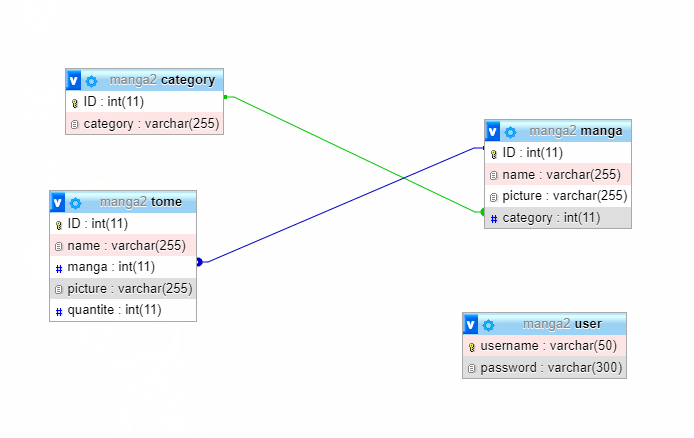


Figure 9 : Modèle physique de données ou Base de Donné

#### Le contrôleur **Controller/RegisterController.php** :

Un contrôleur est une classe qui peut contenir une ou plusieurs méthodes. Chaque méthode correspond à une route et contient toute la logique des données à envoyer au front-end à l’aide d’une vue.

<?php

require\_once('model/ViewManager.php');

require\_once('model/InsertManager.php');

require\_once('model/DeleteManager.php');

require\_once('model/userManager.php');

function listMangaNext($page)

{

    $viewManager = new ViewManager();

    $category = $viewManager->category();

    $pageCourante = $\_GET['page'];

    $start = ($pageCourante-1)\*5;

    $numberPage = $viewManager->numberPageManga(5);

    $manga = $viewManager->displayManga($start, 5);

    require('view/frontend/listMangaView.php');

}

function listManga()

{

    $viewManager = new ViewManager();

    $category = $viewManager->category();

    $pageCourante = 1;

    $start = ($pageCourante-1)\*5;

    $numberPage = $viewManager->numberPageManga(5);

    $manga = $viewManager->displayManga($start, 5);

    require('view/frontend/listMangaView.php');

}

function tome($id)

{

    $viewManager = new ViewManager();

    $category = $viewManager->category();

    $pageCourante = 1;

    $start = ($pageCourante-1)\*10;

    $numberPage = $viewManager->numberPageTome(10, $id);

    $tome = $viewManager->displayTome($id, $start, 10);

    $nameManga = $viewManager->selectManga($id);

    require('view/frontend/tomeView.php');

}

function tomeNext($id, $page)

{

    $viewManager = new ViewManager();

    $category = $viewManager->category();

    $pageCourante = $\_GET['pageTome'];

    $start = ($pageCourante-1)\*10;

    $numberPage = $viewManager->numberPageTome(10, $id);

    $tome = $viewManager->displayTome($id, $start, 10);

    $nameManga = $viewManager->selectManga($id);

    require('view/frontend/tomeView.php');

}

function addTome($mangaID, $nameTome, $picture, $amount, $number)

{

    if (!empty($\_POST['number'])) {

            $viewManager = new ViewManager();

            $nameManga = $viewManager->selectManga($mangaID);

            mkdir('public/image/'.$nameManga["name"]);

            $tmpname=$picture['tmp\_name'];

            $name\_file=$picture['name'];

            $chemin='public/image/'.$nameManga["name"].'/'.$name\_file;

            move\_uploaded\_file($tmpname, $chemin);

            $insertManager = new InsertManager();

            $newTome = $insertManager->insertTome($mangaID, $nameTome, $name\_file, $amount);

        header('Location: index.php?action=tome&id='.$mangaID);

    } elseif (empty($\_POST['number'])) {

        $insertManager = new InsertManager();

        $newTome = $insertManager->insertTome($mangaID, $nameTome, $picture, $amount);

        if ($newTome === false) {

            throw new Exception("Impossible d'ajouter le tome !");

        } else {

            header('Location: index.php?action=tome&id='.$mangaID);

        }

    }

}

function formTome()

{

    $viewManager = new ViewManager();

    $manga = $viewManager->getManga();

    $category = $viewManager->category();

    require('view/frontend/ajoutTome.php');

}

function addManga($name, $category, $file)

{

    $tmpname=$file['tmp\_name'];

    $name\_file=$file['name'];

    $chemin='public/image/'.$name\_file;

    move\_uploaded\_file($tmpname, $chemin);

    $insertManager = new InsertManager();

    $newManga = $insertManager->insertManga($name, $name\_file, $category);

    if ($newManga === false) {

        throw new Exception("Impossible d'ajouter le manga !");

    } else {

        header('Location: index.php');

    }

}

function formManga()

{

    $viewManager = new ViewManager();

    $category = $viewManager->category();

    $category2 = $viewManager->category();

    require('view/frontend/ajoutManga.php');

}

function addCategory($name)

{

    $insertManager = new InsertManager();

    $newCategory = $insertManager->insertCategory($name);

    if ($newCategory === false) {

        throw new Exception("Impossible d'ajouter la catégory");

    } else {

        header('Location: index.php');

    }

}

function formCategory()

{

    $viewManager = new ViewManager();

    $category = $viewManager->category();

    require('view/frontend/ajoutCategory.php');

}

function selectCategory()

{

    $viewManager = new ViewManager();

    $selectCategory = $viewManager->displayCategory($\_GET['id']);

    $category = $viewManager->category();

    require('view/frontend/categoryView.php');

}

function nameCategory($ID)

{

    $viewManager = new ViewManager();

    $name = $viewManager->nameCategory($ID);

    return $name;

}

function numberTome($id)

{

    $viewManager = new ViewManager();

    $numberTome = $viewManager->numberTome($id);

    return $numberTome;

}

function selectUpdate()

{

    $viewManager = new ViewManager();

    $category = $viewManager->category();

    require('view/frontend/modif.php');

}

function updateManga()

{

    $viewManager = new ViewManager();

    $category = $viewManager->category();

    $nameCategory = $viewManager->category();

    $nameManga = $viewManager->getManga();

    require('view/frontend/modifManga.php');

}

function selectManga()

{

    $viewManager = new ViewManager();

    $category = $viewManager->category();

    $nameManga = $viewManager->getManga();

    require('view/frontend/modifTome.php');

}

function updateTome($ID)

{

    $viewManager = new ViewManager();

    $category = $viewManager->category();

    $nameManga = $viewManager->selectManga($ID);

    $nameTome = $viewManager->listTome($ID);

    require('view/frontend/modifTome.php');

}

function editManga($ID, $newName, $newPicture, $newCategory)

{

    $insertManager = new InsertManager();

    $viewManager = new ViewManager();

    $origin = $viewManager->selectManga($ID);

    if (empty($\_POST['newName'])) {

        $newName = $origin['name'];

    }

    if (empty($\_POST['newPicture'])) {

        $newPicture = $origin['picture'];

    }

    $update = $insertManager->insertUpdateManga($ID, $newName, $newPicture, $newCategory);

    if ($newManga === false) {

        throw new Exception("Impossible de mettre a jour le manga !");

    } else {

        header('Location: index.php');

    }

}

function editTome($ID, $newName, $newPicture, $newAmount)

{

    $insertManager = new InsertManager();

    $viewManager = new ViewManager();

    $origin = $viewManager->selectTome($ID);

    if (empty($\_POST['newName'])) {

        $newName = $origin['name'];

    }

    if (empty($\_POST['newPicture'])) {

        $newPicture = $origin['picture'];

    }

    if ($\_POST['newAmount'] > '0' or $\_POST['newAmount'] == '0') {

        $newAmount = $\_POST['newAmount'];

    } else {

        $newAmount = $origin['quantite'];

    }

    $update = $insertManager->insertUpdateTome($ID, $newName, $newPicture, $newAmount);

    if ($newManga === false) {

        throw new Exception("Impossible de mettre a jour le tome !");

    } else {

        header('Location: index.php');

    }

}

function search($name)

{

    $viewManager = new ViewManager();

    $category = $viewManager->category();

    $search = $viewManager->searchManga($name);

    require('view/frontend/resultView.php');

}

function supprimerFullTome($IDmanga)

{

    $deleteManager = new DeleteManager();

    $deleteManager -> deleteTome($IDmanga);

}

function supprimerManga($IDmanga)

{

    $deleteManager = new DeleteManager();

    supprimerFullTome($IDmanga);

    $deleteManager -> deleteManga($IDmanga);

    header('Location: index.php');

}

function selectSupprimer()

{

    $viewManager = new ViewManager();

    $category = $viewManager->category();

    require('view/frontend/supprimerMangaTome.php');

}

function supprimerMangaView()

{

    $viewManager = new ViewManager();

    $category = $viewManager->category();

    $nameCategory = $viewManager->category();

    $nameManga = $viewManager->getManga();

    require('view/frontend/supprimerManga.php');

}

function user()

{

    $viewManager = new ViewManager();

    $category = $viewManager->category();

    if ($\_SERVER["REQUEST\_METHOD"]=='POST') {

        $userManager = new UserManager();

        $username = $\_POST['identifiant'];

        $user = $userManager -> getUserByUserName($username);

        if ($user) {

            $MDP = $\_POST ['mot\_de\_passe'];

            if (password\_verify($MDP, $user -> getPassword())) {

                $\_SESSION['user'] = serialize($user) ;

                header('Location: index.php');

            }

        }

    }

    require('view/frontend/ViewFormulaireLog.php');

}

    function deconnexion(){

        session\_destroy();

        session\_write\_close();

        header('Location: index.php');

    }

#### La vue **register/index.html.twig** :

La vue est le fichier final qui génère la page HTML. Elle est composée de HTML et tous les éléments qui seront interprétés par le navigateur du client.

<?php ob\_start(); ?>

<?php

while ($data = $category->fetch()) {

    ?>

<li><a href="index.php?action=selectCategory&amp;id=<?= $data['ID'] ?>"><?= $data['category'] ?></a></li>

<?php

}

?>

<?php $genre = ob\_get\_clean(); ?>

<?php ob\_start();?>

<h1> connection </h1>

<form action="index.php?action=login " method="post">

    <label for="username">nom d'utilisateur</label>

    <input type="text" name="identifiant" id="username">

    <label for="password">mot de passe</label>

    <input type="password" name="mot\_de\_passe" id="password">

    <button class="connection" type="submit" id="connection">

        connection

    </button>

</form>

<?php $body = ob\_get\_clean();

include('template.php') ?>

La vue génère l’écran suivant :

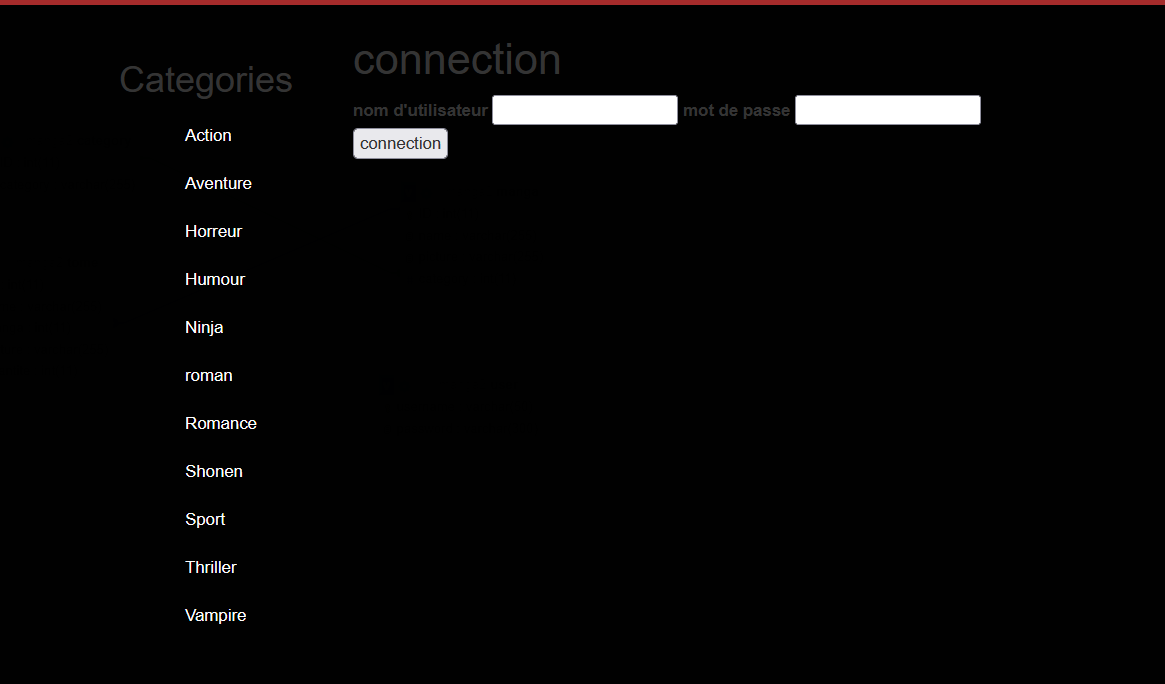


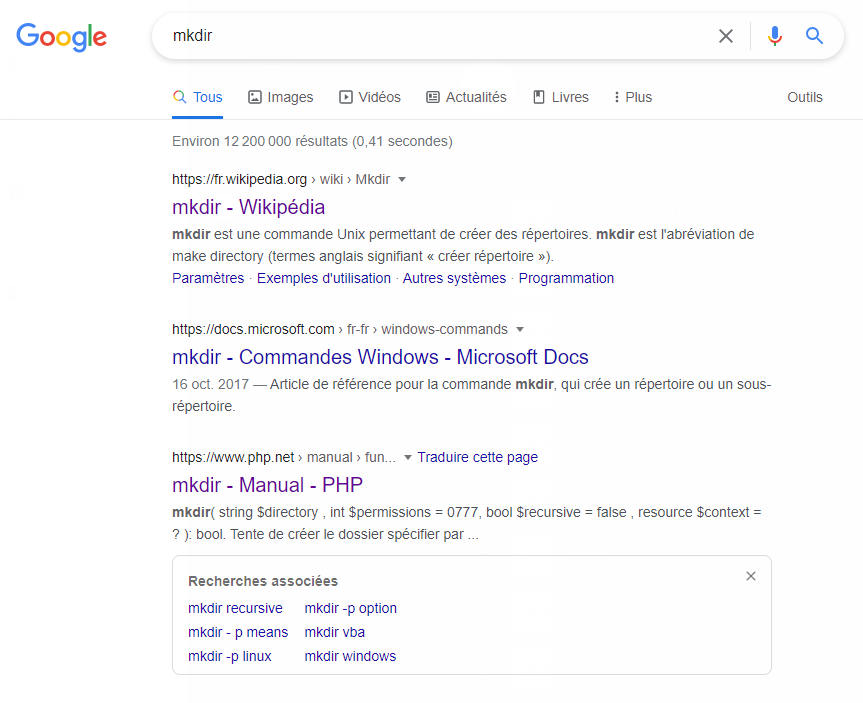
Figure 10 : Formulaire de connexion

1. Situation de travail

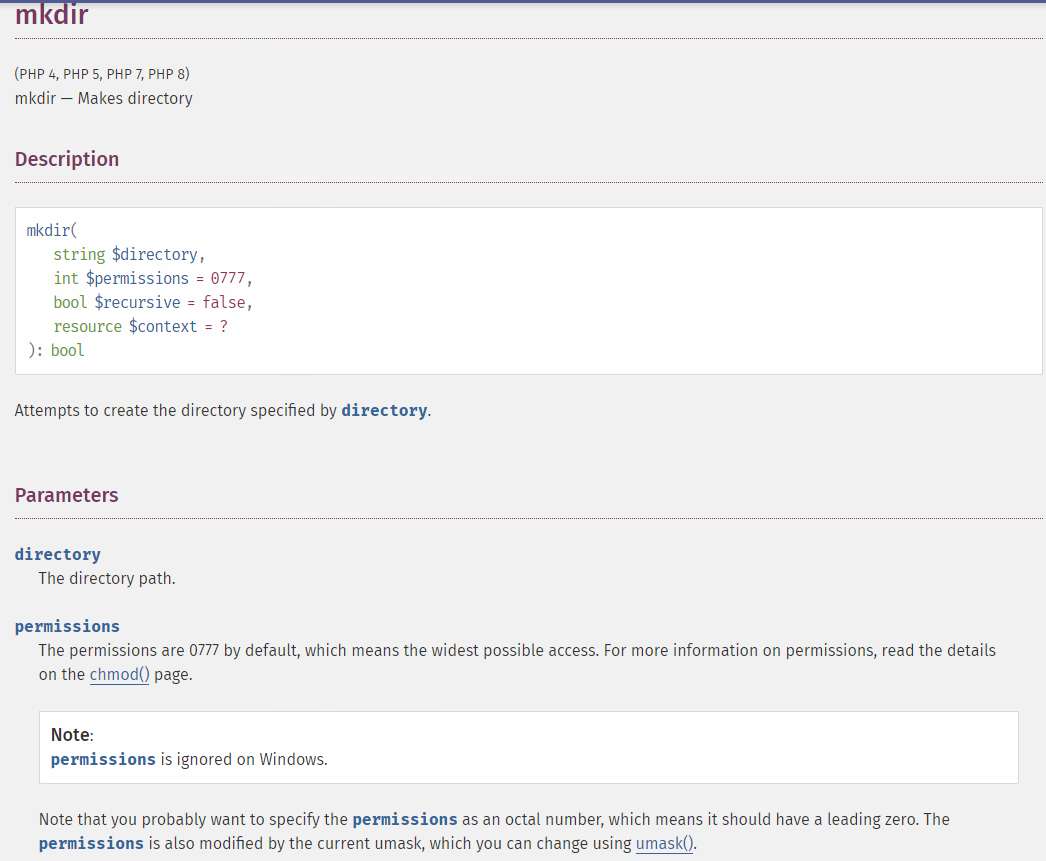
J’avais un problème car je n’arrivais pas avoir l’image dans les tomes des diffèrent livre j’ai donc cherché un moyen d’avoir ses images.

J’ai effectué une recherche avec le moteur de recherche de Google les mots clés suivants : « mkdir »

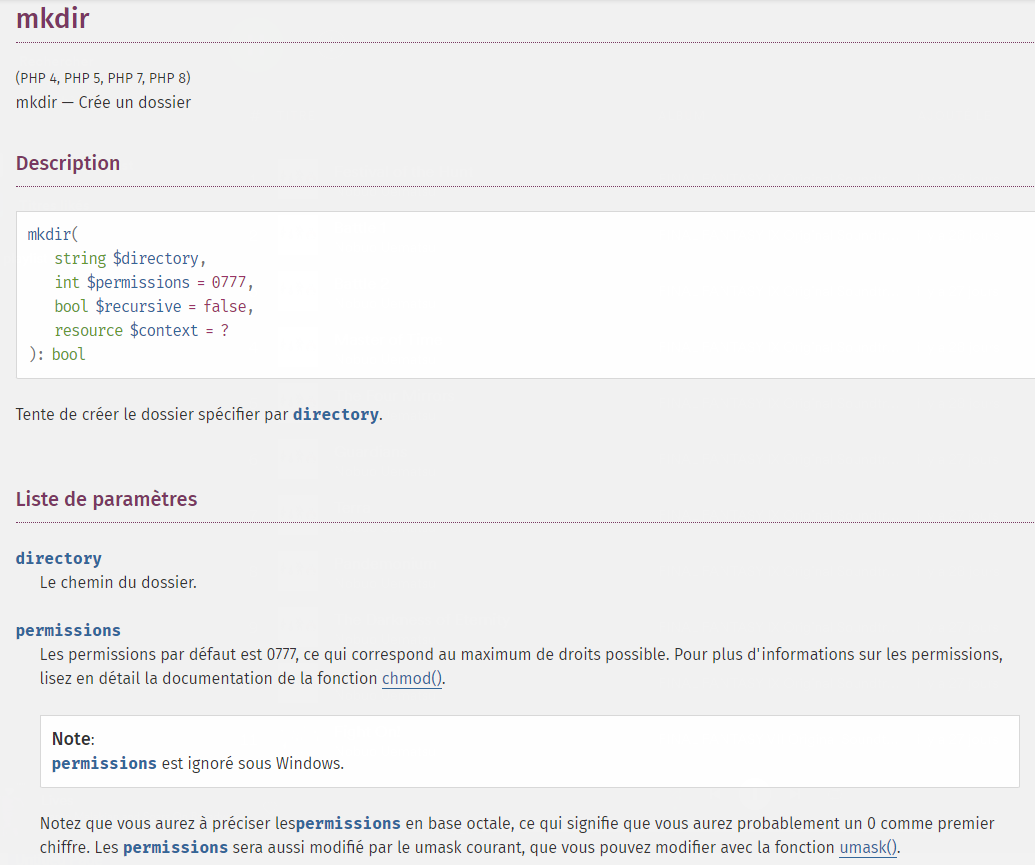
J’ai suivi le troisième lien proposé par Google.



Ce qui m’a permis d’avoir la documentation sur tout le processus. (site en Anglais )



Voici l’extrais en français :



1. Bilan et Conclusion

## Bilan

Les exigences fonctionnelles fixées dès le début par le cahier des charges ont été menées à bien. Les principaux modules caractérisant un site web ont été intégrés.

Les exigences non fonctionnelles qui assurent la qualité et la performance du site ont toutes été développées. Son ergonomie est satisfaisante pour les utilisateurs. Il est compatible avec les navigateurs Microsoft Edge, Firefox et Chrome.

Cependant, certains points doivent être améliorés ou n’ont pu être développés par manque de temps. Par exemple :

L’ajout d’un système d’inscription pour les utilisateurs pour pouvoir réserver leurs livres.

Néanmoins, je trouve que le projet « **BIBLIOTHEQUE** » est conforme aux objectifs visés qui consistent à intégrer les modules de base caractérisant un site web **responsive design**.

## Conclusion

Le projet de site web « BIBLIOTHEQUE » permet de mettre en relation plusieurs enseignements.

Travailler sur ce projet a été particulièrement enrichissant. Il m’a permis de mettre en pratique les compétences acquises durant ma formation tels que le maquettage, la modélisation, les différents langages de programmation, ainsi que d’améliorer mes connaissances liées au développement d’un site web.

Travailler en autonomie m’a permis de développer mon sens de l’organisation, ainsi que la logique de conception.

J’y ai appris aussi à m’auto-former, ce qui me permettra de progresser continuellement.

Cette formation a été riche d’enseignement pour développer mon expérience et consolider mes compétences.

Les difficultés rencontrées durant le développement du projet m’ont donné un aperçu de l’organisation dans le monde professionnel. Je me sens prêt à pouvoir m’adapter à une équipe de développement.