Dossier de projet professionnel

Titre: Développeur web et web mobile

Date : 23/06/2021

Table des matières

Compétences du référentiel couverte par le projet

Le projet couvre les compétences énoncées ci-dessous:

Pour l'activité 1, "**Développer la partie front-end d'une application web et web mobile en intégrant les recommandations de sécurité**":

- Maquetter une application.
- Réaliser une interface web et web mobile statique.
- Réaliser une interface web utilisateurs web dynamique.
- Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce.

Pour l'activité 2, "**Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité**.":

- Créer une base de données.
- Développer les composants d'accès aux données.
- Développer la partie back-end d'une application web ou web mobile.
- Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce.

Résumé

Cette boutique en ligne se nomme Home Of Bike et a été créée en 2020. C'est une vente de vélos en ligne qui a pour but de regrouper plusieurs marques et types de vélos. Que se soit pour les femmes, les hommes, urbain, off-road et de plusieurs tailles notre e-boutique offre tous ces types de vélos. Le but de cette boutique est de proposer plusieures marques de vélos à plus petit prix qu'en boutique. Tout cela en profitant des tous les avis des clients disponibles en dessous des produits.

Périmètre du projet

Le site sera réalisé en français et sera accessible depuis l'ordinateur.

Expression des besoins

Notre site e-commerce devra être totalement fonctionnel et comporter plusieurs produits et plusieurs catégories. Un système de paiement et un espace client. Ainsi qu'un système de connexion / inscription et commentaires. Et pour finir une page de contact.

Spécifications fonctionnelles

Cibles visées par le site

Le site Home of Bike à pour but de s'adresser à des particuliers aimants le vélo que ce soit des femmes ou bien des hommes. Qu'ils soient expérimentés ou non.

Arborescence du site

L'arborescence du site est la suivante:

- Page d'accueil
- Page de connexion
- Page d'inscription
- Page a propos
- Page contact
- Page tous les produits
- Page détails produit
- Page profil
- Page commande
- Page panier
- Page paiement

Une page de plus est présente la page “admin” mais celle-ci n’est pas accessible aux utilisateurs car elle sert à la gestion du site.

Système d’inscription

Le système d’inscription va permettre à l'utilisateur de créer un compte en entrant différentes données sur lui. Comme son nom, prénom, email, username ainsi qu’un mot de passe et une confirmation de mot de passe.

Système de connexion

L’utilisateur pourra suite à son inscription se connecter directement par un formulaire de connexion et pourra se déconnecter et se reconnecter autant de fois qu’il le souhaite. Une fois connecté l’utilisateur pourra accéder à l'entièreté des fonctionnalités de la boutique.

Catalogue produit et filtre

Une page est réservée à afficher tous les produits confondus. Cette affiche devra comprendre une photo du produit, le nom et le prix du produit ainsi que deux boutons, un permettant de l’ajouter directement au panier et le deuxième d'accéder à la page du produit avec les informations détaillées.

Un système de filtre est intégré sur le catalogue produit permettant à l'utilisateur de filtrer les produits en fonction de la categorie, de la marque, de la taille, des materiaux et du sexe desiré.

Fiche produit

L’utilisateur pourra accéder à une fiche produit ou le produit déjà présent sera plus détaillé. Cette page comportera les informations suivantes:

- Le nom et la marque du produit
- Le prix du produit
- La taille du produit

- Les matériaux avec lesquels il est constitué
- La taille des roues
- La catégorie dans laquelle il se trouve
- Le sexe auquel le produit est destiné
- Une description du produits
- Une section permettant d'augmenter la quantité de produit et de l'ajouter au panier
- Une section commentaire permettant de les consulter et d'en poster si l'utilisateur est connecté.

Espace client

L'utilisateur aura accès une page dédiée à ces informations ou il pourra les modifier. Il pourra modifier son nom, son prénom, son email, son nom d'utilisateur et son mot de passe. Il pourra aussi y voir ces 5 dernières commandes passées. Il y retrouvera la date de commande, le modèle, la marque et la quantité commandée.

Fonctionnalité de recherche de produit

Le client aura accès à une search bar pour trouver le produit qu'il a en tête. Il pourra chercher en fonction des marques qu'il veut.

Panier client

Le client pourra effectuer un ajout de produit à son panier pour ensuite passer commande sur le site. La page affichera les éléments suivants:

- Une photo du produit
- Le nom et la marque du produit
- Le prix du produit à l'unité
- Un bouton permettant de le supprimer du panier
- Un bouton permettant d'ajouter le produit en plus grande quantité
- Un bouton permettant d'accéder à la fiche produit
- Le prix total du nombre de produit

- Le prix total du panier
- Un bouton permettant de passer à la commande

Commentaire des clients sur les produits

Le client pourra être en mesure de commenter les produits de la boutique. Sur chaque fiche produit si l'utilisateur est connecté il aura accès à une section commentaire qui pourra l'aider dans son choix grâce à des ressenti d'autres utilisateurs. Il pourra également en poster pour partager son avis avec les autres visiteurs du site.

Back office

L'administrateur devra avoir accès à un dashboard admin pour pouvoir gérer les choses basiques présentes sur la boutique.

Gestion des produits

L'administrateur aura la capacité de gérer les produits il pourra effectivement créer de nouveau produit ou les supprimer.

Lors de la créations du produit l'administrateur pourra mettre les informations suivantes:

- Le modèle du produit
- La marque du produit
- La taille du cadre
- La couleur du produit
- Les matériaux
- Le prix du produit
- La taille des roues
- La catégorie du produit
- Le sexe auquels le produit est destiné

- Le nombre de produit en stock
- Une description du produit

Gestion des utilisateurs

L'administrateur aura la possibilité de consulter tous les utilisateurs inscrits sur son site mais aussi de les supprimer si cela est nécessaire.

Il pourra y retrouver les informations suivantes:

- L'id utilisateur
- L'email utilisateur
- Le nom d'utilisateur
- Le prénom de l'utilisateur
- Le nom de l'utilisateur

Gestion des commentaires

L'administrateur aura également la possibilité de gérer les commentaires postés par les utilisateurs. Il pourra les supprimer s' il le souhaite.

Il pourra consulter les informations suivantes:

- L'id de l'utilisateur qui a posté
- Le Username de l'utilisateur qui a posté

- Le produit que l'utilisateur a commenté
- Le contenu du commentaire

Gestion des contact

L'administrateur aura la possibilité de consulter les utilisateurs qui ont pris contact avec eux.

Il pourra consulter les infos suivantes:

- L'id de l'utilisateur ayant pris contact
- L'email de l'utilisateur ayant pris contact
- Le message de l'utilisateur

Grâce à l'email l'administrateur pourra reprendre contact avec l'utilisateur directement par mail.

Solution de paiement

La solution de paiement choisie est proposée par stripe. Elle permettra aux clients de procéder à leur paiement par carte bancaire en toute sécurité. Et bien sur en toute sécurité du cote de l'administrateur.

Spécifications techniques

Choix Techniques et environnement de travail

Technologies utilisées pour la partie back-end:

- Le site sera réalisé avec le langage PHP
- Une base de données SQL
- Gestionnaire de dépendance: Composer

Technologies utilisées pour la partie front-end:

- Le projet sera réalisé avec du HTML et CSS
- Bootstrap
- Et du Javascript pour le système de paiement

L'environnement de développement est le suivant:

- Editeur de code: Visual studio code
- Outil de versionning: GIT, Github
- Maquettage: adobe xd

Architecture

Pour ce qui est de l'architecture du projet nous avons choisi une architecture classique. Avec des dossiers organisés. Les dossier s'organisent comme ci dessous:

- index.php => à la racine
- composer.json => à la racine
- composer.lock => à la racine
- Dossier vendor (il concerne composer) => à la racine
- Dossier src => à la racine

⇒ Dossier component

→ footer.php

→ header.php

⇒ Dossier config

→ config.php

→ Dossier style

->img (toutes les images se trouvent dedans)

->bootstrap.css

->bootstrap.min.css

->Style.css

→ Dossier class

-> categorieclass.php

-> commandeclass.php

-> commentaireclass.php

-> contactclass.php

-> panierclass.php

-> productclass.php

-> userclass.php

- Dossier pages => à la racine

→ about.php

→ admin.php

→ boutique.php

→ catégorie.php

→ commande.php

→ connexion.php

→ contact.php

→ inscription.php

→ paiement.php

→ panier.php

→ produit.php

→ profil.php

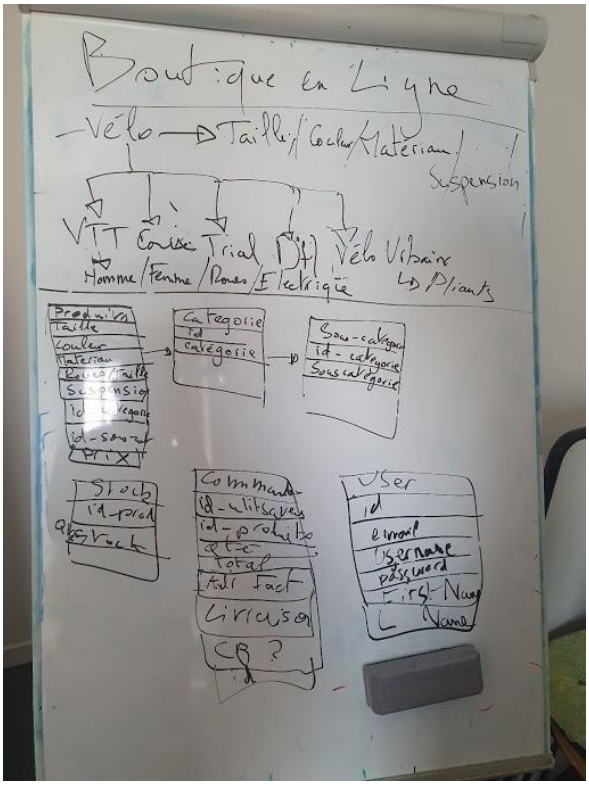
→ script.js (pour le système de paiement)

Réalisation

Organisation avant le commencement:

Avant toute chose nous nous sommes concertés et avons fait un wireframe des différentes pages et de la base de données pour évaluer l'étendue du travail. Pour ensuite faire la maquette.


Profil
Commerce
Inscription
Header
Footer
Index.php
Panier
A propos
Connexion
Paiement

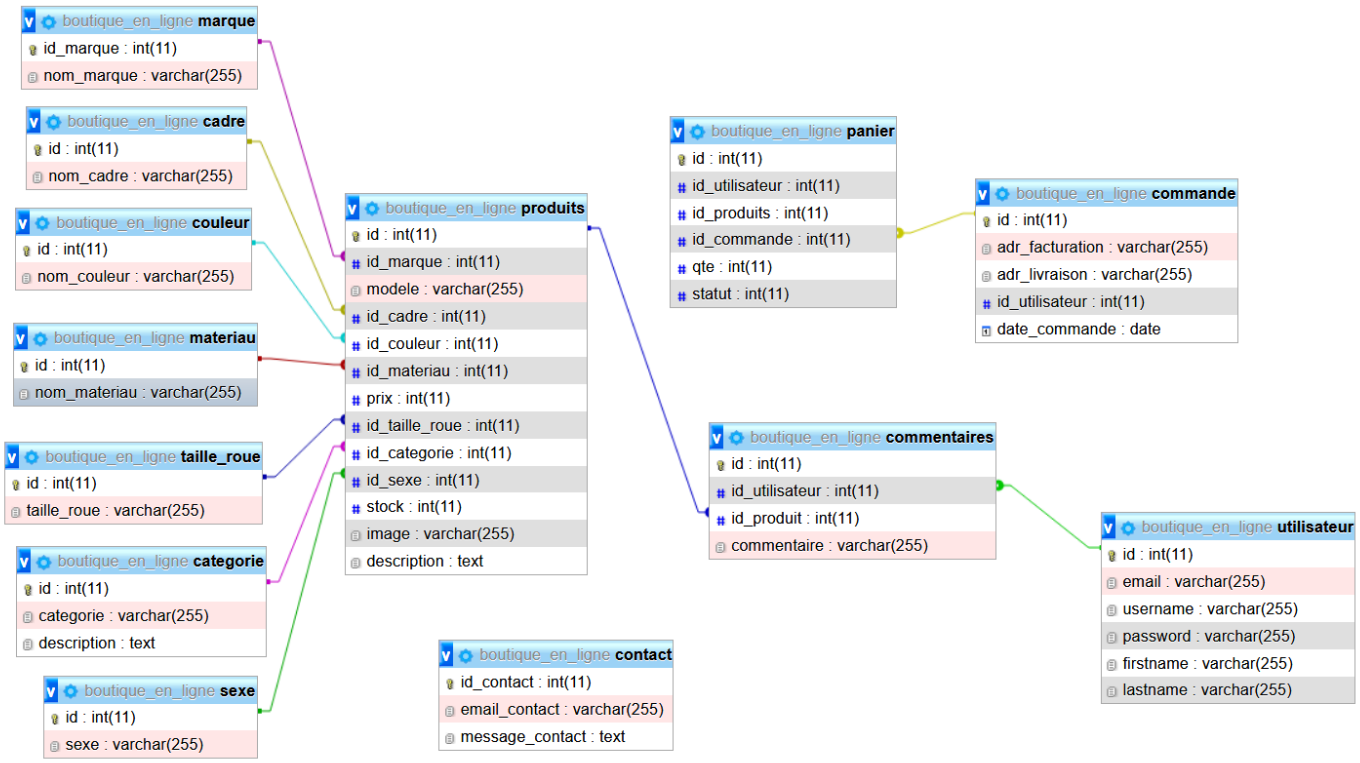

Boutique en Ligne
Vélo
Taille
Couleur
Matériau
Suspension
VTT
Course
Trial
DH
Vélo Vitesse
Homme
Femme
Enfant
Electrique
Pliants
Produits
Catégorie
Sous-catégorie
Stock
Commande
User
Email
Password
Nom
Prénom

Maquettage:

La maquette du site a été faite sur le logiciel adobe xd. N'ayant aucun point de départ, l'entièreté du design a été faite par mes soins. Vous pourrez retrouver la maquette dans l'annexe de ce dossier.

Conception base de données:

J'ai conçue la base de données de la manière suivante:



Vous trouverez également dans les annexes le modèle conceptuel de données. Comme on peut le voir ci- dessus l'essentiel de la base de données s'articule autour de la table produits. Elle est reliée à toutes les petites tables qui caractérisent le produit. Nous avons la table utilisateur en relation avec la table des commentaires et la table panier en relation avec la table commande.

Extrait de code significatif

Panier client:

```php
public function addtocart($id_produit,$qte,$id_utilisateur,$id_commande){
    $db=$this->connectdb();
    $id_produit=htmlspecialchars(trim($id_produit));
    $id_commande=htmlspecialchars(trim($id_commande));
    $qte=htmlspecialchars(trim($qte));
    $query=$db->prepare("SELECT * FROM `panier` WHERE `id_utilisateur`='{$id_u
    $query->execute();
    $produitinpanier=$query->fetchAll();
    if($produitinpanier==null){
        $query=$db->prepare("INSERT INTO `panier`(`id_utilisateur`, `id_produi
        $query->execute();
    }else{
        $newqte=$produitinpanier[0]['qte']+$qte;
        $query=$db->prepare("UPDATE `panier` SET `qte`='{$newqte}' WHERE `id_u
        $query->execute();
    }
```

Tout d'abord j'ai créé une fonction qui permet d'ajouter le produit en base de données. La fonction addtocart permet de vérifier si le produit que l'utilisateur n'est pas déjà dans son panier au quel cas il ne le mettra pas une seconde fois. Si le produit n'est pas en base de données il sera alors ajouter ainsi que toute les informations importantes comme l'id utilisateur, l'id produit, la quantité et l'id commande. Cette fonction sert aussi à updater la quantité si l'utilisateur l'a modifiée.

```php
public function getpanierbyidutilisateur($id_utilisateur){
    $db=$this->connectdb();
    $query=$db->prepare("SELECT produits.modele, produits.prix, produits.i
    materiau.nom_materiau, taille_roue.taille_roue, categorie.categorie,se
    JOIN `marque` ON produits.id_marque=marque.id_marque
    JOIN `cadre` ON produits.id_cadre=cadre.id
    JOIN `couleur` ON produits.id_couleur=couleur.id
    JOIN `materiau` ON produits.id_materiau=materiau.id
    JOIN `taille_roue` ON produits.id_taille_roue=taille_roue.id
    JOIN `categorie` ON produits.id_categorie=categorie.id
    JOIN `sexe` ON produits.id_sexe=sexe.id
    JOIN `panier` ON produits.id=panier.id_produits WHERE panier.id_utilis
    $query->execute();
    $allresults=$query->fetchAll();
    return $allresults;
```

Ensuite j'ai créé une fonction nommée getpanierbyutilisateur ou je récupère toutes les informations qui vont nous servir à l'affichage de ce produit dans le panier. Comme vous pouvez le voir ci-dessus dans cette requête SQL je récupère des informations de différentes tables de notre base de données grâce aux JOIN. Puis je recupere toutes ces informations grâce à un fetchAll(). Je n'aurai plus qu'à appeler ma fonction dans la page voulue et afficher les informations dans l'ordre désiré.

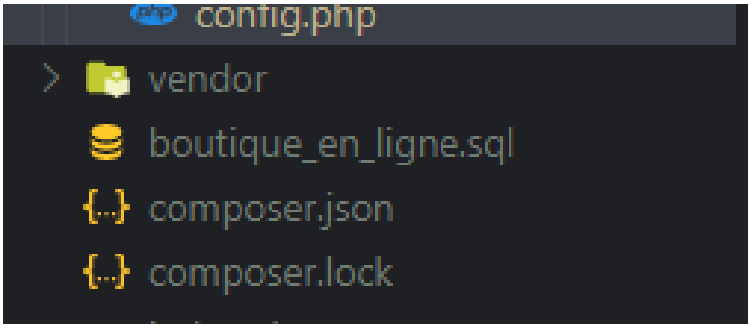
```php
public function updateqtepanier($id_produit,$qte,$id_utilisateur){
    $id_produit=htmlspecialchars(trim($id_produit));
    $qte=htmlspecialchars(trim($qte));
    $id_utilisateur=htmlspecialchars(trim($id_utilisateur));
    $db=$this->connectdb();
    $query=$db->prepare("UPDATE `panier` SET `qte`='{$qte}' WHERE `id_utilisa
    $query->execute();
}
public function dropproduitpanier($id_produit, $id_utilisateur){
    $id_produit=htmlspecialchars(trim($id_produit));
    $id_utilisateur=htmlspecialchars(trim($id_utilisateur));
    $db=$this->connectdb();
    $query=$db->prepare("DELETE FROM `panier` WHERE `id_utilisateur`='{$id_u
    $query->execute();
}
```

Pour finir cette page panier j'ai fais deux dernières fonctions. La première fonction qui se nomme updateqtepanier va comme bien dit dans son nom servir à updater la quantité de produits. Directement sur la page panier si l'utilisateur va pouvoir grâce à cette fonction augmenter ou baisser la quantité de produits dans son panier. Tout cela grâce à un UPDATE dans notre requête sql.

La dernière fonction de cette class panier se nomme dropproduitpanier et va être utile grâce à un simple DELETE dans notre requête à supprimer le produit du panier avec une bouton sur la page panier.

Integration du systeme de paiement:

Stripe est une solution de paiement qui va permettre aux clients d'effectuer leur paiement en carte bancaire mastercard ou visa. Son intégration est organisée autour d'une API.



J'ai tout d'abord télécharger composer. Composer est un logiciel gestionnaire de dépendance. Il permet à ces utilisateurs de déclarer et d'installer les bibliothèques dont le projet a besoin. Composer contient tous les fichiers nécessaires au fonctionnement de stripe. Mais une légère configuration est tout de même nécessaire.

```
window.onload = () => {
    // definitions des variables
    let stripe = Stripe('pk_test_51ILmbsCfQp9Oksz4D3XXOjxhiXp5SkOnweOo1aPouOW56DEq
    let elements = stripe.elements()
```

Tout d'abord je défini les variables dont j'aurais besoins. Je définis d'abord la variable stripe qui est la clé API accessible à tout le monde.

```
// objet de la page
let cardHolderName = document.getElementById("cardholder-name")
let cardButton = document.getElementById("card-button")
let clientSecret = cardButton.dataset.secret;
```

Je récupère et définit les éléments de ma page HTML grâce à un getelementbyid().

```
// on gere la saisie
card.addEventListener("change", (event)=>{
    let displayError = document.getElementById("card-errors")
    if(event.error){
        displayError.textContent = event.error.message;
    }else{
        displayError.textContent = "";
    }
})
```

Je gère maintenant la saisie avec un addEventListener qui va nous servir à "écouter ce qu'il se passe a chaque changement". Grâce au javascript nous allons pouvoir afficher une erreur sans rafraichir la page de l'utilisateur.

```
// on gere maintenant le paiement
cardButton.addEventListener("click", ()=> {
    stripe.handleCardPayment(
          clientSecret, card, {
              payment_method_data: {
                  billing_details: {name: cardHolderName.value}
              }
          }
    ).then((result) => {
        if(result.error){
            document.getElementById("errors");innerText = result.error.message
        }else{
            document.location.href = "../index.php";
        }
    })
})
```

Pour finir on va s'occuper du paiement en lui-même. Lorsque l'utilisateur va cliquer sur le bouton de confirmation de paiement on va vérifier si le paiement va bien se faire ou non.Si le paiement échoue je vais afficher un message d'erreurs et si le paiement réussi l'utilisateur sera renvoyé sur la page d'accueil. Pour essayer des paiement stripe nous fournit des numéros de carte fait pour les test. Ensuite pour vérifier si le paiement est bien effectué on peut le vérifier sur notre dashboard stripe que j'ai créé au commencement.

MONTANT		DESCRIPTION	CLIENT
1 040,00 €	Réussi ✓	pi_1IwCbMCfQp9Oksz4LzpVkhB2	Jean Michel
520,00 €	Réussi ✓	pi_1IfPXPCfQp9Oksz4xDfZSsxJ	rachid
30 000,00 €	Réussi ✓	pi_1If6TyCfQp9Oksz4T7KJF50O	merliche
3 000,00 €	Réussi ✓	pi_1If689CfQp9Oksz41d6Dac5w	michmich
867,00 €	Réussi ✓	pi_1If5XTCfQp9Oksz4wk7YeEBb	manoo
867,00 €	Réussi ✓	pi_1If5VxCfQp9Oksz4DIQrr62t	nerner

Sécurité mise en oeuvre

```
$username = htmlspecialchars(trim($username));
$password = htmlspecialchars(trim($password));
$cpassword = htmlspecialchars(trim($cpassword));
$firstname = htmlspecialchars(trim($firstname));
$lastname = htmlspecialchars(trim($lastname));
$email = htmlspecialchars(trim($email));
```

Tout d'abord je sécurise les champs contre les injections SQL. Puis Je rajoute la fonction trim() elle retourne la chaîne string, après avoir supprimé les caractères invisibles en début et fin de chaîne.

```
$query = $db->prepare("SELECT id FROM utilisateur WHERE username= '$username'");
$query->execute();
```

J'effectue des requêtes préparées avec la fonction prepare(). Lors de la préparation, un template de requête est envoyé au serveur de base de données. Le serveur effectue une vérification de la syntaxe, et initialise les ressources internes du serveur pour une utilisation ultérieure.

```
$password = password_hash($password, PASSWORD_BCRYPT);
```

Je hash les passwords pour une bonne sécurité des données utilisateurs.

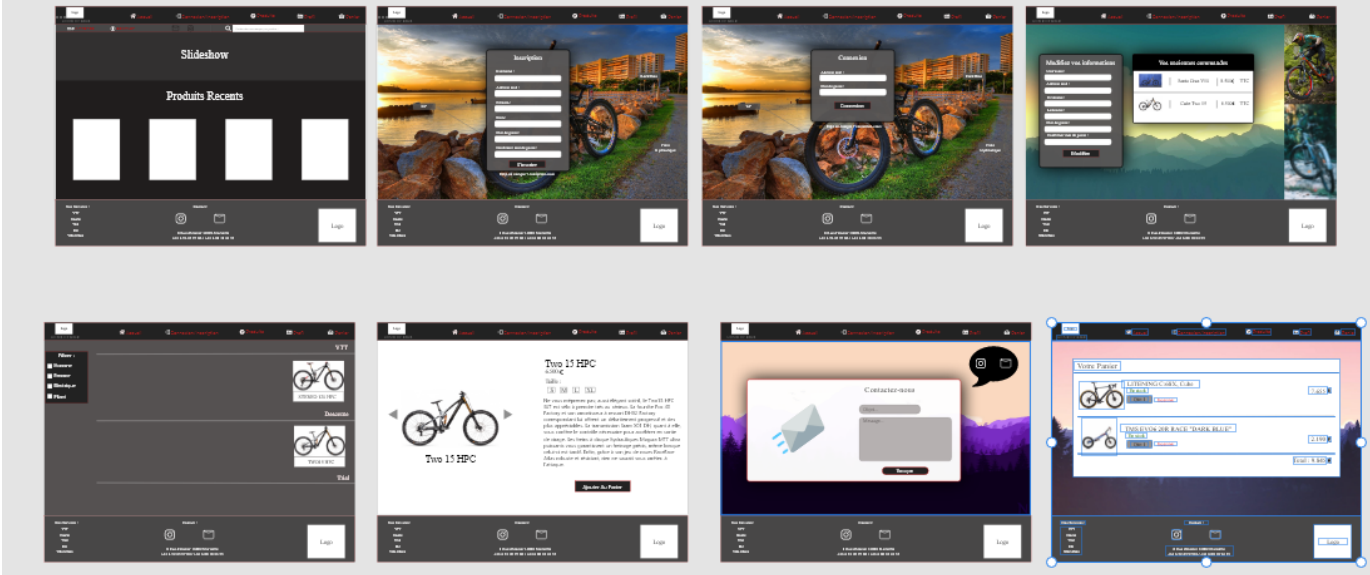
Mes sources d’aides

Pour ce projet je me suis aidé des sources suivantes:

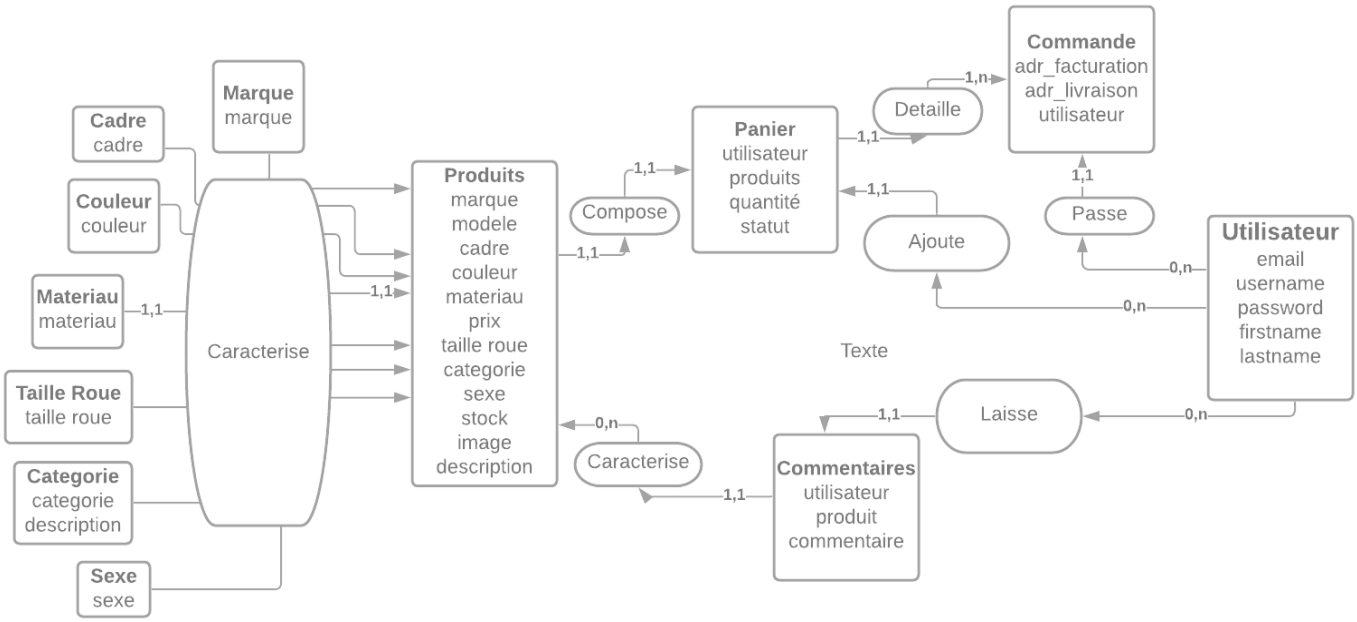
- https://www.php.net/
- https://stackoverflow.com/
- https://developer.mozilla.org/fr/docs/Web/JavaScript
- D’autres forum et sites

Annexe

Maquette



Modèle conceptuel des données



Modèle logique des données