

# BTS SIO 2022

## Support et mise à disposition de services informatiques (E4)

PAGE DE PRÉSENTATION DU DOSSIER

N° de candidat<sup>1</sup> : | 0 | 2 | 1 | 4 | 6 | 7 | 1 | 7 | 4 | 8 | | 5 |

NOM : Mutschler.....

Prénom : Sacha.....

Date de passage <sup>1</sup> : ..... / ..... /2022	Heure de passage <sup>1</sup> : .....h.....
--	---

CATEGORIE CANDIDAT <sup>2</sup> (UNE CASE A COCHER)	
<input type="checkbox"/> Scolaire <input checked="" type="checkbox"/> Apprenti <input type="checkbox"/> Formation professionnelle continue <input type="checkbox"/> Expérience professionnelle 3 ans	<input type="checkbox"/> Ex-scolaire <input type="checkbox"/> Ex-apprenti <input type="checkbox"/> Ex-formation professionnelle continue

<sup>1</sup> Informations communiquées sur votre convocation envoyée en mars-avril 2022

<sup>2</sup> Informations communiquées sur votre confirmation d'inscription

Tampon de  
L'établissement



**BTS SIO – Dossier Etudiant**  
**Justificatif d'acquisition des compétences**

**Epreuve E4**  
**Support et mise à disposition de**  
**Services informatiques**

# SOMMAIRE

- 1 INTRODUCTION ..... 4
- 2 MISSION 3 : DENDO JITENSHA ..... 5
  - 2.1 Cahier des charges..... 5
  - 2.2 Etude et conception de la solution ..... 7
  - 2.3 Gestion de projet ..... 7
  - 2.4 Mise en œuvre..... 8
  - 2.5 Bilan ..... 20

# 1 Introduction

---

L'objectif de ce document est de vous présenter les missions professionnelles que j'ai effectué dans le cadre de ma formation BTS SIO à l'école IRIS de Strasbourg.

Ces missions peuvent être de trois types :

- Effectuées en entreprise durant une alternance
- Effectuées en stage en entreprise
- Effectuées à l'école (compte-rendu de TP, projet collaboratif)

Le type de la mission sera précisé dans chaque cahier des charges.

Ce document se compose des parties suivantes :

Chapitres	Contenu
Chapitres 1 à 5	<p>Présentation des missions, avec pour chacune :</p> <ul style="list-style-type: none"><li>- Le cahier des charges</li><li>- La solution proposée</li><li>- La gestion de projet</li><li>- La mise en œuvre</li><li>- Le bilan du projet</li></ul>

## 2 Mission 3 : Dendo Jitensha

### 2.1 Cahier des charges

<b>Type de mission</b>
Mission effectuée à l'école.
<b>Contexte</b>
Dans le cadre d'un appel d'offre de l'entreprise Dendo Jitensha notre groupe répondra au cahier des charges fourni et mettra en place un site E-Commerce permettant l'achat de produits proposé par l'entreprise.
<b>Demande du client</b>
Création d'un site E-Commerce regroupant les produits (Vélos) et permettant le paiement en ligne.
<b>Expression du besoin</b>
<p>Le site livré sera un site e-commerce uniquement, composé des pages suivantes :</p> <ul style="list-style-type: none"><li>• Une page Accueil</li><li>• Une page Produits ou les produits seront triables et filtrables</li><li>• Une page Détails d'un produit lorsqu'on clique sur ce dernier</li><li>• Une bouton Mon compte en haut à droite qui permet de pour pouvoir réinitialiser le mot de passe s'il est oublié.</li><li>• Une page Mon compte où l'on peut modifier ses informations personnelles</li><li>• Un bouton de déconnexion si l'utilisateur est connecté</li><li>• Une page panier stocké en base de données. Dans ce dernier on pourra voir une liste, produit par produit, comportant sa description, son image, son prix, la quantité, et un bouton "supprimer". Le prix total sera affiché tout en dessous et un bouton "Procéder à l'achat" nous redirigera vers la page "Panier". En appuyant sur le bouton "valider le panier", on accède à la page Paiement.</li><li>• Une page Paiement.</li><li>• Une page Contact</li><li>• Une page Qui sommes-nous ?</li><li>• SGBD : MySQL via PHPMyAdmin</li><li>• Environnement de travail collaboratif : GitHub</li><li>• Environnement de développement : Visual Studio Code ou PHPStorm avec un serveur WAMP en local</li><li>• Gestion de versions : GitHub</li><li>• Tests unitaires : PHPUnit.</li></ul>

### Budget disponible

100€ Annuel en hébergement.

50 000€ (fictif) pour l'estimation de la charge de travail en jour pour l'ensemble du groupe.

5 développeurs.

### Outils disponibles

Postes utilisateurs avec un serveur local installé (Wamp)

Git + GitHub ([https://github.com/sachamutschler/5Bits\\_Dendo](https://github.com/sachamutschler/5Bits_Dendo))

IDE : Visual studio Code et PHP Storm

### Contraintes

Temps : environ 6 mois.

Apprentissage de Git pour certains membres du groupe.

Répartition des tâches dans le groupe.

### Confidentialité

RGPD et Mentions Légales disponibles.

## 2.2 Etude et conception de la solution

---

### 2.2.1 Les solutions possibles

---

- Développement du front avec Framework javascript (Vue JS)
- Développement avec du javascript + node + mongodb
- Développement avec un framework PHP (Laravel)
- Développement From Scratch (PHP Natif et MySQL)
- Développement avec du javascript

### 2.2.2 La solution retenue

---

- Développement From Scratch (PHP Natif et MySQL)
- Développement avec du javascript

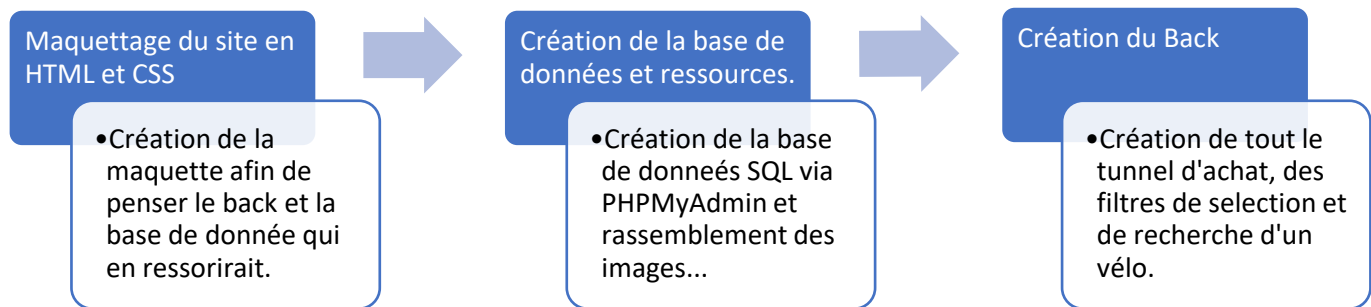
## 2.3 Gestion de projet

---

### 2.3.1 Planing de déploiement de la solution

---

Schéma de la réalisation de la mission :



### 2.3.2 Budget

---

100€ Annuel en hébergement.

50 000€ (fictif) pour l'estimation de la charge de travail en jour pour l'ensemble du groupe.

5 développeurs.

5 Postes.



## 2.4 Mise en œuvre

---

### 2.4.1 Implémentation de la solution

---

L'entreprise 5 Bits est composée de 5 personnes, ces derniers se nomment Aurélien, Fabian, Nihad, Sacha et Thomas, elle a été créée en 2021 au sein de l'école IRIS.

Le responsable d'équipe est Sacha Mutschler, il assume également le rôle de développeur.

Le développement Front end sera effectué par Nihad et ce dernier sera suppléé par Sacha.

Ils se chargeront de la mise en place de la maquette et la mise en place du GitHub, de la mise en relation vers PHP et de la partie responsive du site.

La gestion de la base de données sera assurée par Fabian.

Le développement Back sera assuré par toute l'équipe, le découpage des tâches ayant été effectué de la manière suivante :

- Nihad : Page d'accueil et Article.
- Aurélien : Page Produits et Qui sommes-nous ?
- Thomas : Inscription / Connexion et mon compte.
- Fabian : Création et administration de la base de données.
- Sacha : Page Panier, Paiement et Contact.

L'équipe mettra en place les formulaires et toutes les requêtes vers la base de données du site dans un dossier spécifique, s'inspirant du modèle « MVC ».

Nous utiliserons la technologie PHP ainsi que MySQL.

#### **Présentation du contexte :**

La société DENDO JITENSHA, actrice majeure de solutions de mobilité durables, souhaite mettre en place une solution de communication innovante qui aura pour épiscentre un site Internet original qui sera de type e-commerce.

Nous mettrons en place une solution de commercialisation en ligne performante et évolutive, et répondant aux besoins métiers. Pour cela il faudra mettre en œuvre un socle de données unique permettant d'analyser les données de la vision macro jusqu'à la maille la plus fine de manière transparente pour l'utilisateur et avec un historique satisfaisant.

#### **Description fonctionnelle des besoins :**

Compte utilisateur :

- Connexion
- Création de compte
- Récupération de mot de passe
- Déconnexion

Fonction principale : Connectez l'utilisateur à son compte client

Sous-fonctions :

- Se connecter
- Mot de passe oublié
- Se créer un compte
- Se déconnecter

Fonction : Se connecter	
Objectif	Accéder au compte client, pour profiter de toutes les fonctionnalités liées (historique des commandes, accès aux informations diverses, etc. )
Description	L'accès à la page de connexion se fera par un bouton accessible sur tout le site en haut à droite.
Contraintes / règles de gestion	<p>La connexion sera gérée par les JSON token. Ainsi, il n'est pas nécessaire que l'utilisateur se connecte à chaque fois, son token sera valable une semaine avant d'avoir à entrer ses identifiants à nouveau. Le Token sera stocké sur le localStorage du navigateur.</p> <p>Au moment du chargement du site, le client se chargera de chercher si dans le localStorage se trouve un token, si oui, celui-ci sera envoyé au serveur afin de tenter une connexion et une récupération des informations client.</p>
Niveau de priorité	Priorité haute

Fonction : Se déconnecter	
Objectif	Se déconnecter du compte client sur un ordinateur
Description	L'accès à la fonction de déconnexion se fera par un bouton accessible sur tout le site en haut à droite.
Contraintes / règles de gestion	La déconnexion consistera en la suppression du JSON token du localStorage du navigateur.
Niveau de priorité	Priorité haute

Fonction : Création d'un compte	
Objectif	Permettre à un utilisateur de créer un compte client sur le site
Description	Sur la page de connexion un bouton "S'inscrire" redirigera l'utilisateur sur une page d'inscription
Contraintes / règles de gestion	Après avoir rempli les champs nécessaires à la création d'un compte, les données seront envoyées au serveur qui traitera les données afin de vérifier que, par exemple, le mail n'est pas déjà utilisé, et qui les inscrira en base de données
Niveau de priorité	Priorité haute

#### Page de Contact :

Objectif : Contacter facilement les vendeurs en remplissant un formulaire de contact.

Description : Un formulaire à remplir en complétant les informations telles qu'adresse mail, nom ainsi que le message à envoyer.

Contraintes : Service disponible uniquement avec un serveur SMTP et un hébergeur.

Niveau de priorité : Moyenne.

Fonction principale : afficher tous les produits	
Objectif	Réaliser une page qui permet à un client de consulter tous les produits du rayon
Description	La page reprend le style de la page d'accueil, avec la navbar et le footer. Le contenu est généré automatiquement en fonction des éléments de la base de données. Chaque produit permet de générer un élément "card" comprenant le nom, l'image, une description, un bouton qui permet d'accéder aux détails du produit, et un bouton qui permet d'ajouter l'article au panier.
Contraintes / règles de gestion	Il faut que lorsqu'on clique sur "ajouter l'article au panier", un modal s'ouvre afin de définir la taille du vélo et la quantité souhaitée.
Niveau de priorité	Priorité haute

Fonctions secondaires : trier par prix / trier par ordre alphabétique	
Objectif	Réaliser quatre boutons : tri prix ascendant, tri prix descendant, tri ordre alphabétique ascendant, tri ordre alphabétique descendant.
Description	La page est identique sauf que l'on souhaite effectuer un tri en fonction du bouton sur lequel l'utilisateur va cliquer.
Contraintes / règles de gestion	Il faut également un bouton "pas de tri" qui apparaît uniquement lorsqu'un tri est appliqué.
Niveau de priorité	Priorité moyenne

### Fonction : Ajouter un élément au panier

Objectif	Pouvoir ajouter au panier une unité ou plus d'une version (couleur, taille) de l'article en cours de consultation.
Description	Après choix de la version de l'article (couleur, taille), et de la quantité souhaitée, il est possible d'ajouter l'élément au panier, en un clic à l'aide d'un bouton submit.
Contraintes / règles de gestion	Pour chaque élément ajouté au panier, la somme des éléments ajoutés doit être mise à jour et apparaître en exposant dans un cercle rouge sur l'icône de panier (partie supérieure droite dans le header). En arrière-plan, le panier doit également être mis à jour. Le panier sera enregistré en base de données si le client est loggué, sinon, il sera mémorisé un certain temps à l'aide d'un cookie.
Niveau de priorité	Priorité haute

### Fonction : Consultation et du modification du panier

Objectif	Pouvoir consulter le récapitulatif des produits ajoutés au panier et mettre à jour les quantités, voire supprimer un article si nécessaire
----------	--

Description : Chaque ligne correspondra à un article, la quantité sélectionnée apparaîtra.

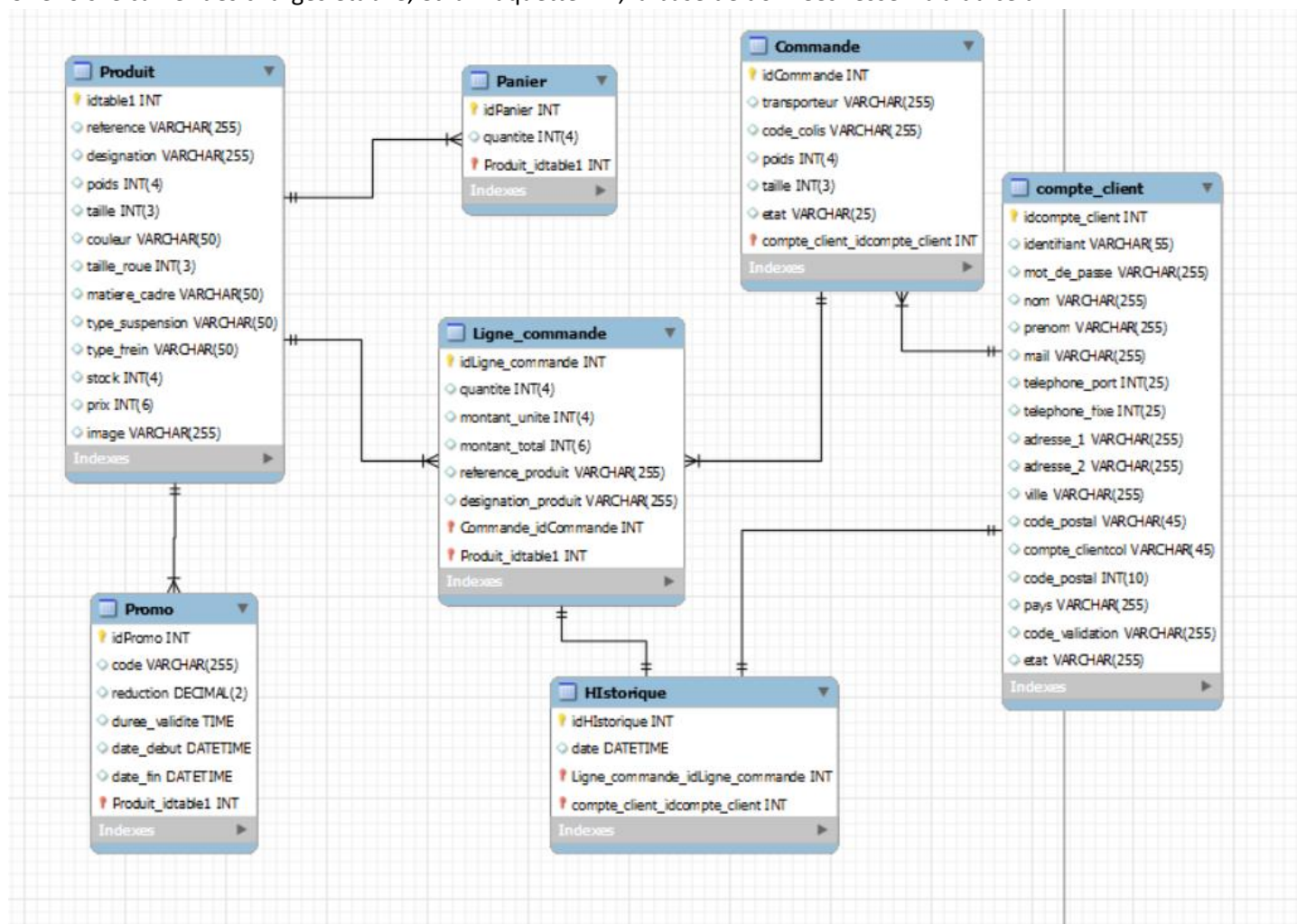
Il y aura la possibilité de supprimer l'article du panier.

Contraintes / règles de gestion	Chaque fois que la quantité article est modifiée ou que l'article est supprimé, le prix total de la ligne est également recalculé. Un recalcul du prix total du panier sera effectué, tenant compte de la présence ou non d'un coupon de réduction.
---------------------------------	--

Priorité Haute.

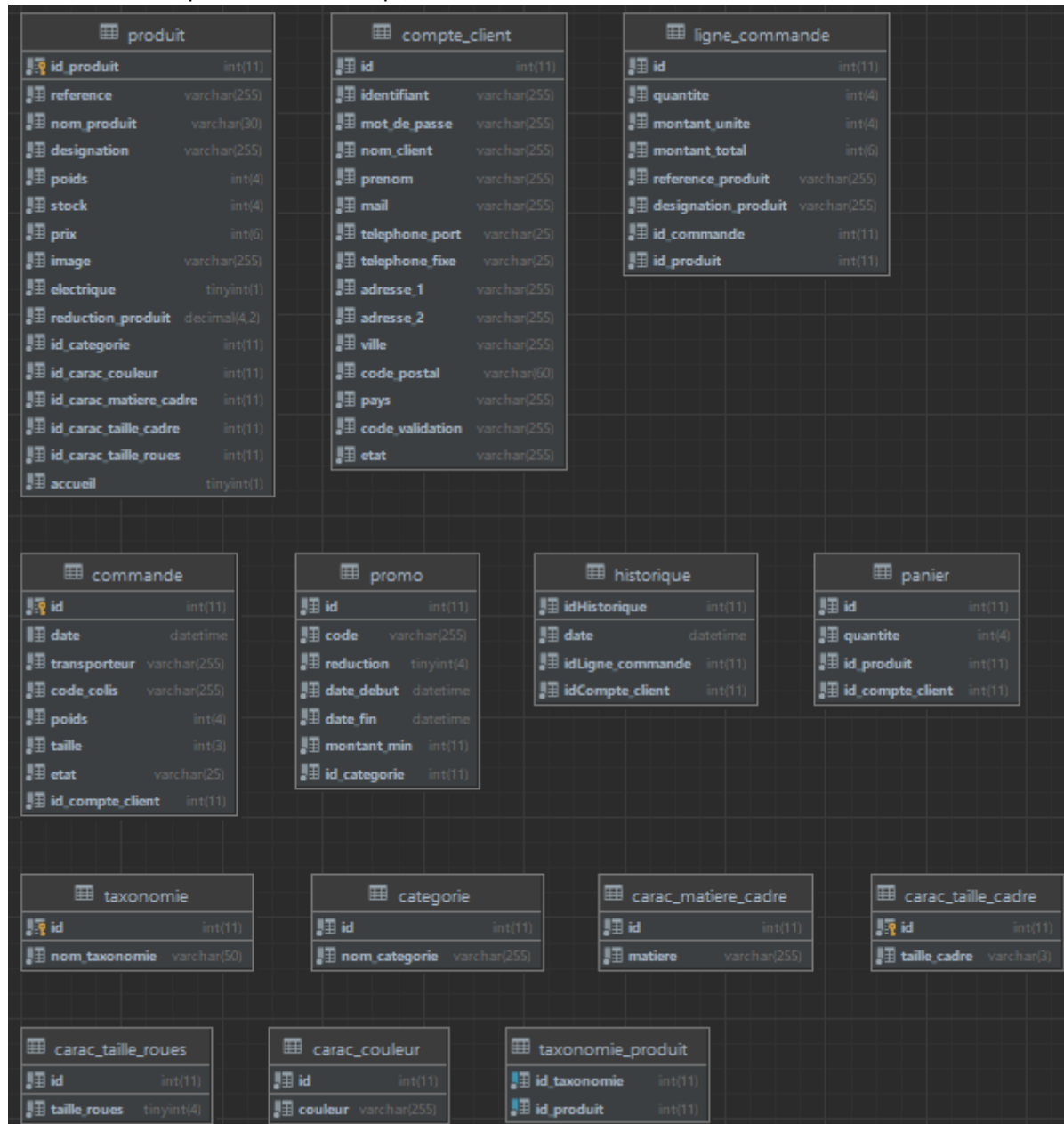
Paiement : Création d'une page from scratch en utilisant l'API de Stripe et l'environnement test de ce dernier.

Une fois le cahier des charges établie, et la maquette fini, la base de données ressemblait à cela :

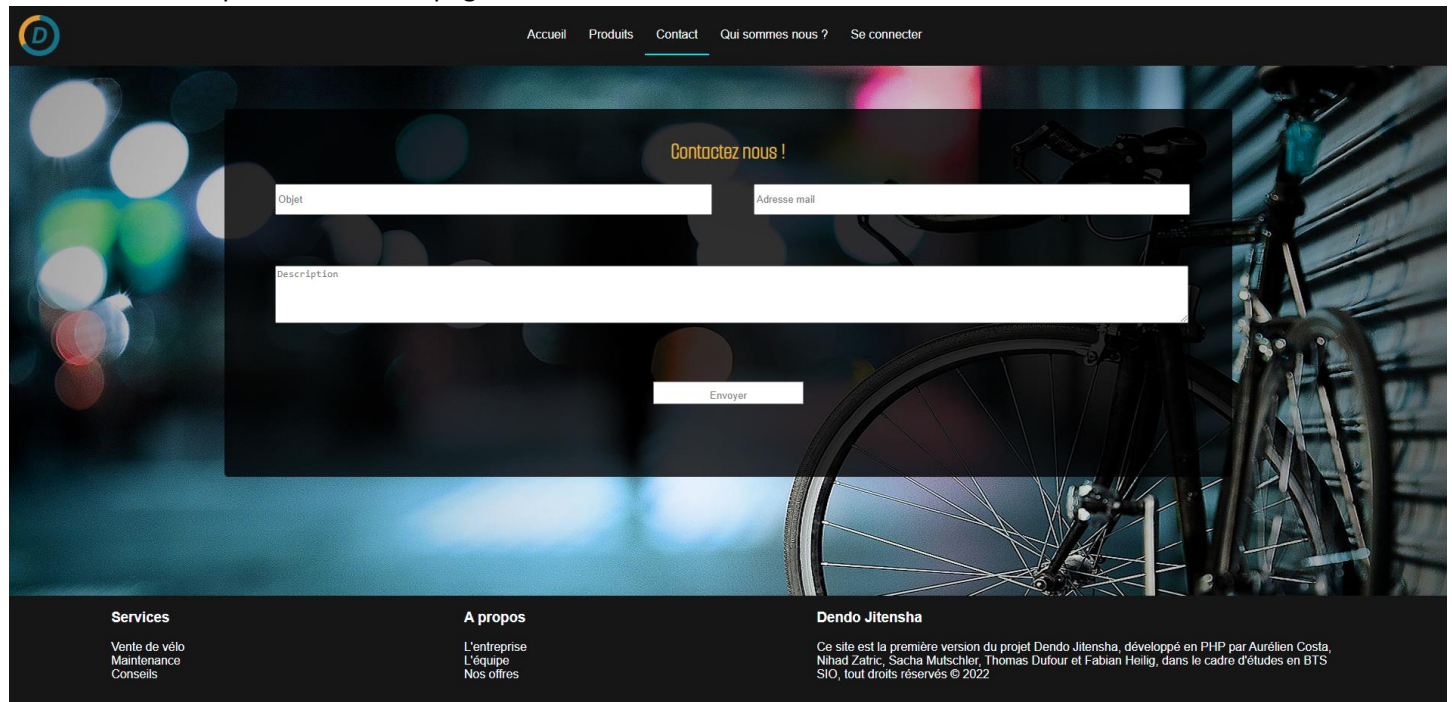




Cette base fut cependant modifiée par la suite :



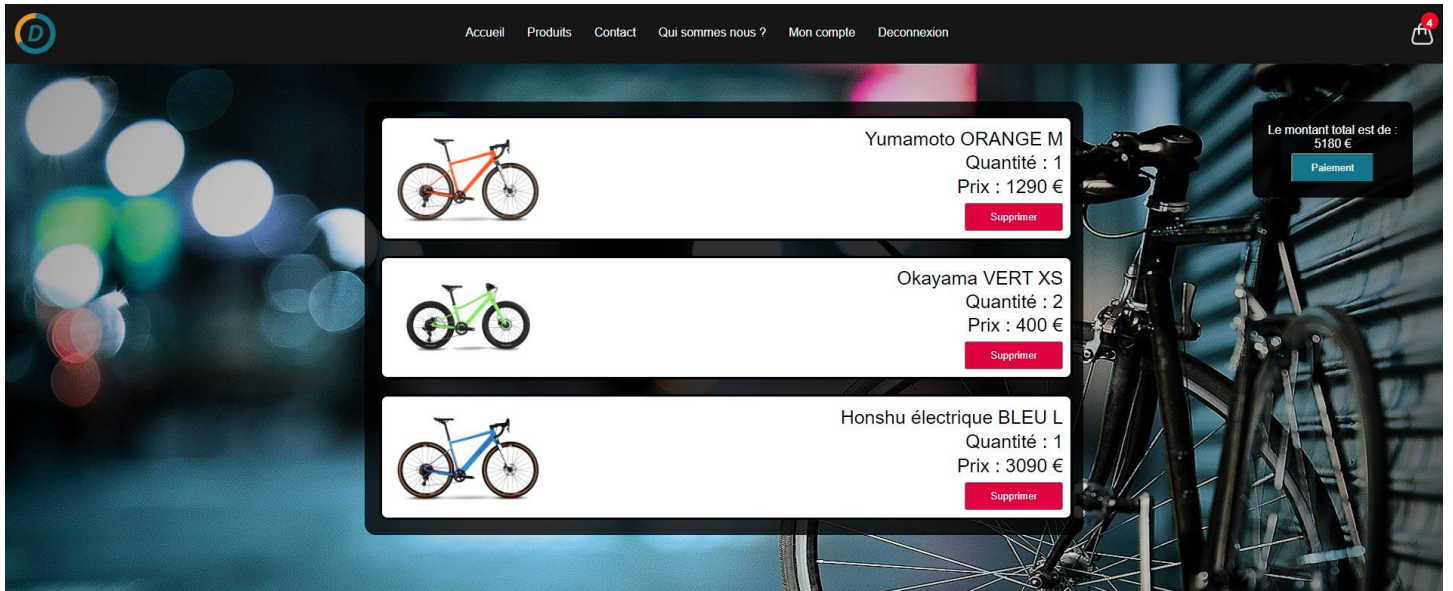
J'ai dans mon cas pu donc créer les pages suivantes :



```
View > 🐞 post_contact.php > ...
1  <?php
2
3
4  if(isset($_POST['button'])) {
5      $objet = $_POST['objet'];
6      $message = $_POST['description'];
7      $mail = $_POST['email'];
8      $headers = 'FROM: '. $mail;
9      mail('contactdendo@gmail.com', $objet, $message, $headers);
10 }
11 elseif(isset($_POST['button'])) {
12     echo('Une erreur est survenue');
13 }
```

On voit ici l'utilisation de la fonction mail de PHP, les informations sont ensuite transmises par mail grâce au serveur SMTP.





Le panier a été relativement difficile à mettre en place comme vous pourrez le voir :

```
<head>
    <?php include ('head.php'); ?>
</head>
<div id="header">
    <?php include ('navbar.php'); ?>
    <link href="public/css/panier.css" rel="stylesheet">
</div>
<div class="cont_panier">
    <div class="cont_panier2">
        <?php
        $prix = 0;
        include('Model/model_panier.php');
        $total = 0;
        for($i=0; $i<count($tableau_id_produit); $i++) {
            ?>
            <div class="panier">
                
                <div class="text_panier">
                    <!-- Création d'un formulaire -->
                    <form action="" class="quantity" method="post">
                        <!-- Affichage du nom du produit avec la ligne $i puis la colonne -->
                        <label for="title_panier" class="t_panier"><?php echo $tableau_produit[$i]['nom_produit']; ?></label>
                        <!-- Affichage de la quantité de vélo sélectionnée avec la ligne $i puis la colonne -->
                        <label class="t_panier">Quantité : <?php echo $tableau_id_produit[$i]['quantite']; ?> </label>
                        <!-- Si la reduction est différente de 0 alors on calcule la réduction -->
                        <?php if($tableau_produit[$i]['reduction_produit'] != 0) {
                            $prix = $tableau_produit[$i]['prix'] * (1 - ($tableau_produit[$i]['reduction_produit'] / 100) );
                        }
                        else {
                            /* Sinon on affiche le prix */
                            $prix = $tableau_produit[$i]['prix'];
                        }
                        $total = $total + ($prix * $tableau_id_produit[$i]['quantite']);
                    ?>

                    <label class="t_panier">Prix : <?php echo $prix; ?> €</label>

                    <button type="submit" value="<?php echo($tableau_produit[$i]['id_produit']) ?>" name="deleteItem" class="bouton delete_panier">Supprimer</button>

                    </form>
                </div>
            </div>
        </div>
    </div>
    <?php
```

On voit au-dessus que j'effectue un include du fichier effectuant les requêtes SQL que voici :

```
//requete prix vélo
$id_compte_client = intval($_SESSION['identifiant']);

$id_produit = "SELECT * FROM panier WHERE id_compte_client = $id_compte_client";
$req_id_produit = $conn->query($id_produit);
$req_id_produit->execute();
$req_id_produit->setFetchMode(PDO::FETCH_CLASS, 'id_produit');
$tableau_id_produit = $req_id_produit->fetchAll(PDO::FETCH_ASSOC);
/* var_dump($tableau_id_produit); */
if(count($tableau_id_produit) !== 0) {

    $requete = "SELECT * FROM produit WHERE id_produit IN (";
    for($i=0; $i < count($tableau_id_produit); $i++) {
        if($i == count($tableau_id_produit) - 1) {
            $requete.= $tableau_id_produit[$i]['id_produit'] . ')';
        }
        else {
            $requete.= $tableau_id_produit[$i]['id_produit'] . ', ';
        }
    }

    $produit = $conn->query($requete);
    $produit->execute();

    $tableau_produit = $produit->fetchAll(PDO::FETCH_ASSOC);
```

Ces requêtes sont ensuite sécurisées avec des « BindParam » .

The image shows a payment modal window. It contains four input fields: 'Adresse e-mail' with an envelope icon, 'Numéro de carte' with a card icon, 'MM / AA' with a calendar icon, and 'CVV' with a lock icon. Below these fields is a large blue button labeled 'Payer'. The modal has a close button (X) in the top right corner.

La page paiement est accessible et fonctionnelle en utilisant l'API de Stripe.

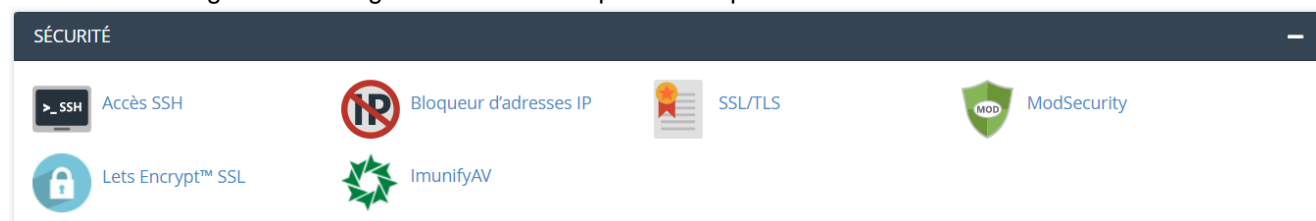
## Événements

ÉVÉNEMENT	ID	DATE
Votre solde a des nouvelles opérations disponibles	evt_1Kfw5sHi3NyOGrbX0XUK11DR	22/03/2022 02:01:04
Un montant de 23,70 € a été débité à ch_3KdbS3HiJNyOGrbX1k5ErQOH	evt_3KdbS3HiJNyOGrbX1c9cbQ6m	15/03/2022 15:34:20
Un montant de 41,70 € a été débité à ch_3KdWKBHiJNyOGrbX0wzXBvlu	evt_3KdWKBHiJNyOGrbX0ocKq019	15/03/2022 10:05:52
Votre solde a des nouvelles opérations disponibles	evt_1KdhuHi3NyOGrbXFB05Rtr	15/03/2022 01:57:50

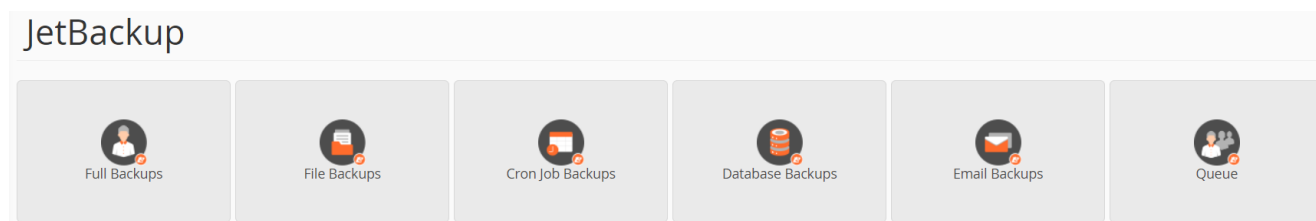
L'API reprend les informations communiquées avec les variables de paiement qui seront fournies par le panier.

```
<script
  src="https://checkout.stripe.com/checkout.js" class="stripe-button"
  data-key="pk_test_51KYVVEHiJNy0GrbX5zfIbX0z1IoPqmalGX73qmirWtx2NzeU8VC
  data-amount="<?php $_SESSION['total'];?>"
  data-name="<?php echo $_GET["item_name"];?>"
  data-description="<?php echo $_GET["item_name"];?>"
  data-currency="inr"
  data-locale="auto">
</script>
</form>
</div>
</div>
<div class="col-md-6">
  <div class="checkout-container">
    <h4>Product Name&nbsp;:&nbsp;<?php //echo $_GET["item_name"]?></h4>
    "/>
    <span>Price &nbsp;:&nbsp;<?php echo $_SESSION['total'];?></span>
  </div>
</div>
</div>
```

Dendo est hébergé sur l'hébergeur « o2Switch » qui offre un panel de sécurité :



Ainsi qu'un outil de backup :



## 2.5 Bilan

---

### 2.5.1 Validation des exigences point par point

---

- ☐ Application disponible et fonctionnelle.
- ☐ Délai respecté.
- ☐ Cahier des charges respecté.

### 2.5.2 Compétences acquises

---

- Approfondissement de PHP et MySQL.
- Utilisation d'une API.
- Management d'un groupe de projet.
- Répartition des tâches.
- Concaténation des éléments dans une requête SQL.
- Création d'un cahier des charges.
- Estimation de couts.