

# Projet Personnel Encadré Travail attendu



**Par Mathis Laversin et Faissel Friouchen**

## TABLE DES MATIERES

<b>Σελτιον 1</b>	<b>Plan type du document Projet</b> .....	<b>3</b>
<b>1.1</b>	<b>But du document</b> .....	<b>3</b>
<b>1.2</b>	<b>Cahier des charges</b> .....	<b>3</b>
<b>1.3</b>	<b>Spécification fonctionnelle</b> .....	<b>4</b>
<b>1.4</b>	<b>Planning</b> .....	<b>6</b>
<b>1.5</b>	<b>Conception</b> .....	<b>7</b>
<b>1.6</b>	<b>Déploiement du site</b> ..... Error! Bookmark not defined.	
<b>1.7</b>	<b>Realisation des tests</b> .....	<b>9</b>
<b>1.8</b>	<b>Annexes</b> .....	<b>10</b>



## Σεχτιον 1 Plan type du document Projet

---

### 1.1 But du document

Site de e-commerce – Vente de coques de smartphones.

Dans ce document nous allons vous présenter le déroulement du projet.

Il contient les maquettes de l'application ainsi que les spécifications fonctionnelles.

Ce document s'adresse à des clients ayant des connaissances techniques en développement afin de réaliser un projet de memory contenant une base donnée.

Ce projet est développé par Mathis Laversin et Faïssel Friouchen.

### 1.2 Cahier des charges

De nos jours avec l'essor des boutiques en ligne. Il est important de se créer une présence sur internet. En tant qu'entreprise, ayant une boutique physique ou non, posséder une boutique fonctionnelle et accessible sur internet n'est pas négligeable.

#### Fonctionnalités attendues :

- Site fonctionnel et hébergé.
- Le site possède une base de données pour permettre aux clients de se créer un compte.
- Le site permet aux utilisateurs de se créer un panier.
- Pour commander il faut être connecté.

Le fait de s'identifier permet d'associer une base de données aux utilisateurs, afin d'enregistrer leurs paniers et leurs listes de souhaits.

#### Délais :

12 mai 2023.

## 1.3 Spécification fonctionnelle

L'application sera accessible à tous sur internet. Les utilisateurs non connectés pourront accéder et visualiser le site. Une fois connectés les utilisateurs pourront enregistrer des produits dans leurs paniers et procéder au paiement.

### 1.3.1 Maquette du projet

Les maquettes ont été effectuées sur Figma.

#### Page d'accueil.

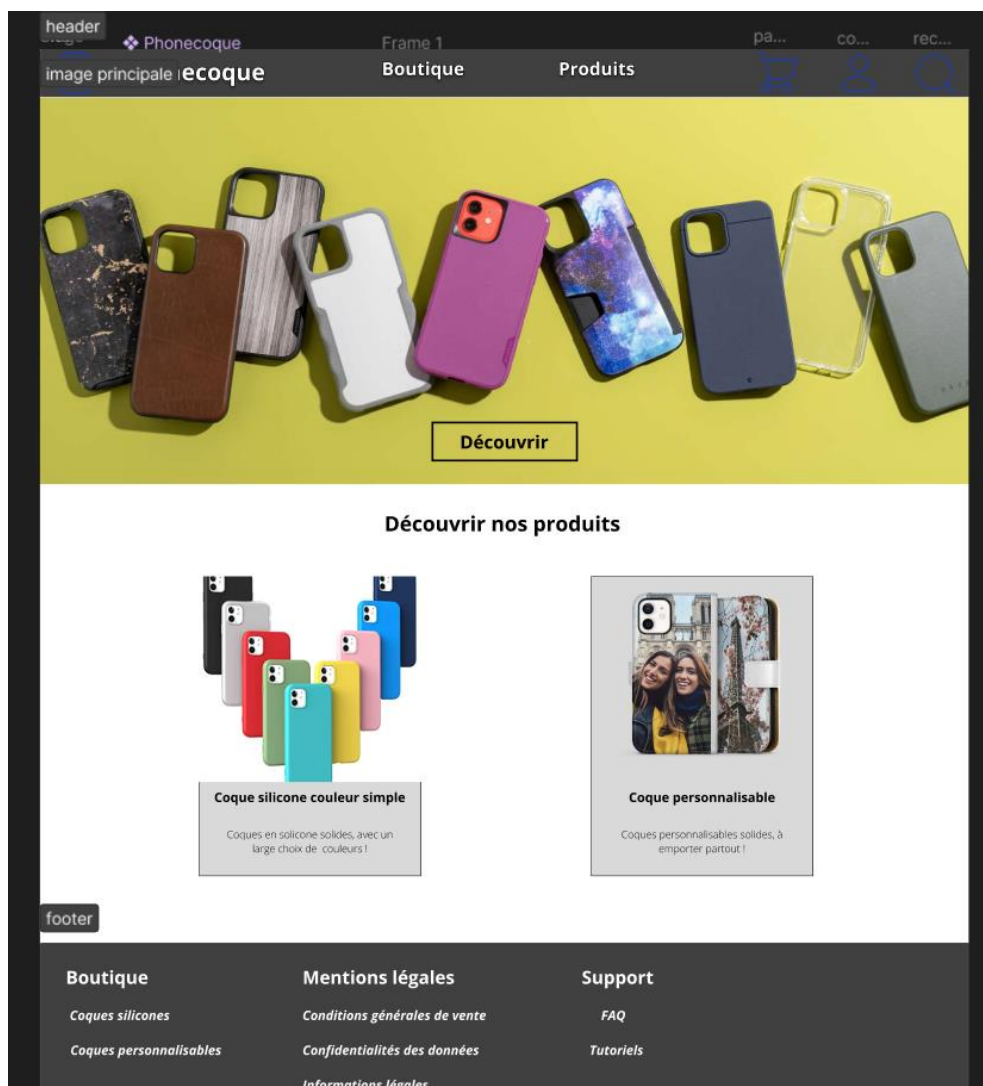
Lors de l'entrée sur le site, l'utilisateur arrivera sur la page d'accueil, avec des produits mis en avant. En haut à droite, dans le header, un accès à la page de connexion.

Depuis le header l'utilisateur pourra accéder aux produits.

En scrollant sur le site, l'utilisateur pourra voir des photos des produits.

Ensuite, des cartes de produits seront affichées.

Enfin, le footer sera visible avec les mentions légales et liens rapides.



## Page de connexion.

Le client pourra se créer un compte avec son mail, nom, prénom et mot de passe.  
Il y aura un moyen d'afficher/cacher le mot de passe.

## Page de produit.

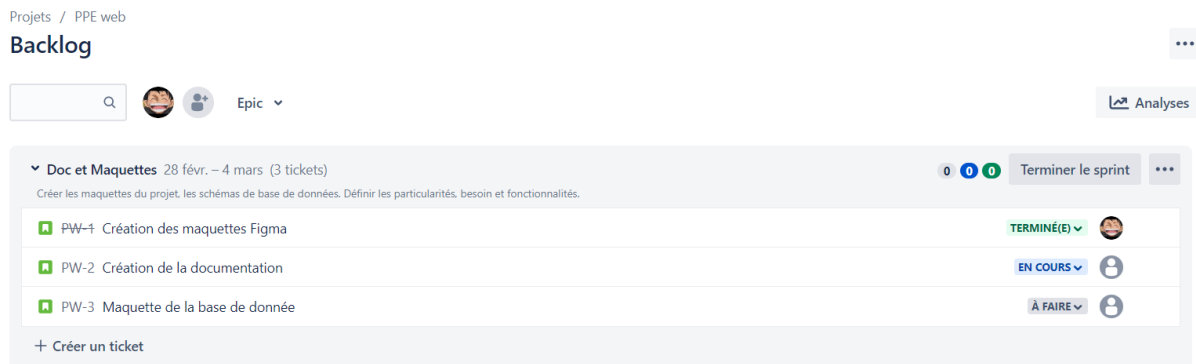
Sur la page du produit, il y aura à gauche l'image de la coque qui sera modifiée selon les personnalisations de l'utilisateur. A droite les personnalisations proposées. La couleur par exemple, le logo sur la coque. Cliquer sur ces modifications changera dynamiquement l'image du produit en prenant en compte les modifications.



## 1.4 Planning

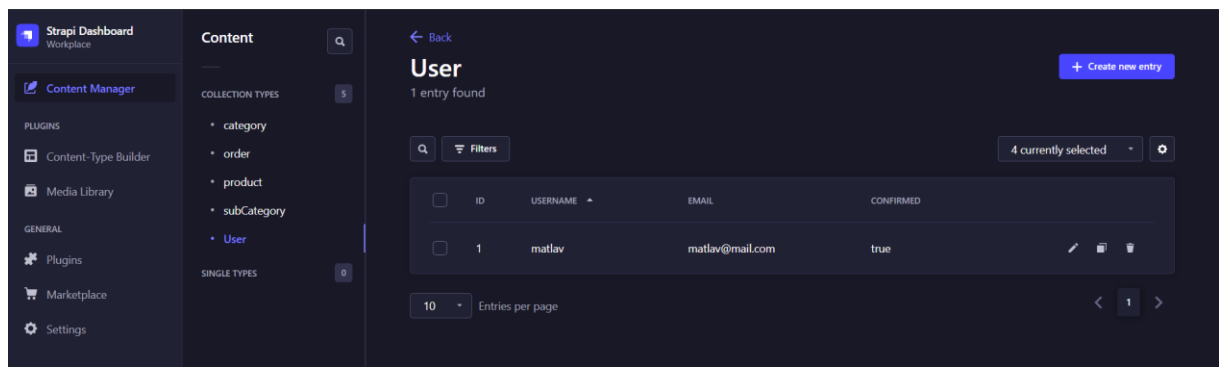
	IV.	FÉV.	MARS	AVR.
Sprints			Doc...	
<ul style="list-style-type: none"> <li>PW-4 Apprentissage React et Node.js</li> <li>PW-5 Coder le projet</li> </ul>				

Nous avons aussi utilisé la méthode Agile SCRUM sur Jira afin d'organiser les tâches à faire et avoir une vision globale.

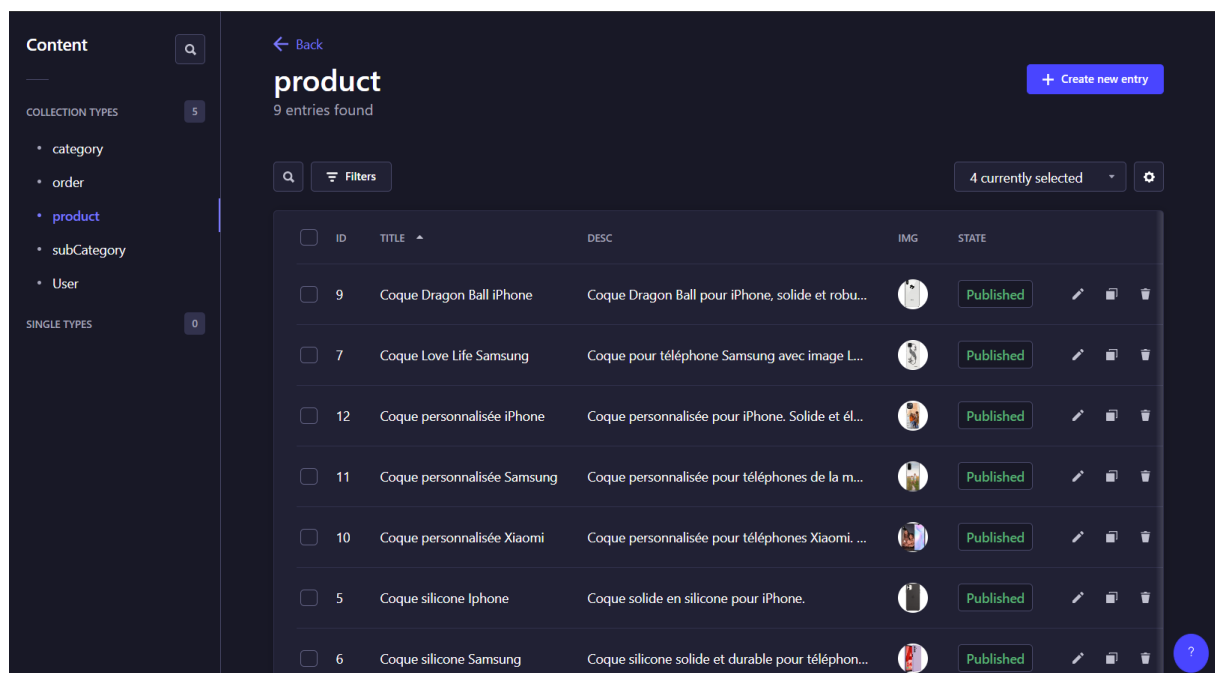


## 1.5 Conception

### 1. Structuration de la base de données avec le CMS Strapi.



Les produits sont aussi enregistrés dans la base Strapi.



Chaque produit peut être personnalisé et il est possible de lui attribuer une catégorie et des sous-catégories. Ainsi qu'un Type, permettant de les regrouper.

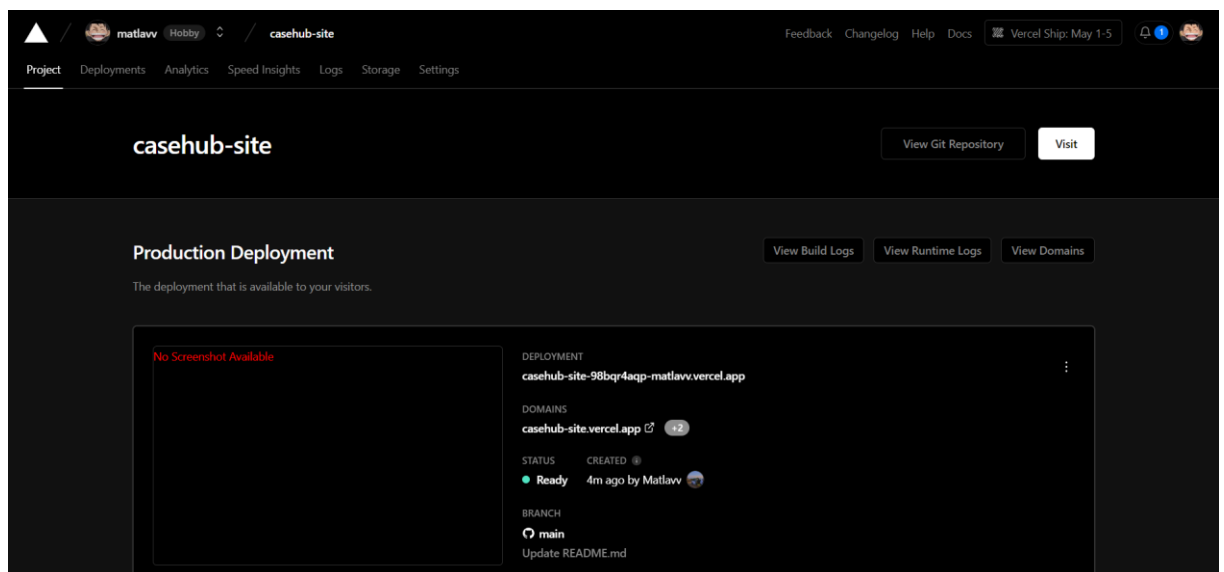
The screenshot shows a product editing form for 'Coque Dragon Ball iPhone'. The form includes fields for title, description, image (img), and a second image placeholder (img2). The price is set to 35. The 'isNew' toggle is set to FALSE. The categories are 'Images' and 'iPhone', both marked as 'Published'. The type is 'Meilleures ventes'. On the right, there is a sidebar with 'Editing published version' and 'INFORMATION' section showing creation and update details by Mathis Laversin. Action buttons include 'Edit the model', 'Configure the view', and 'Delete this entry'.

## 1.6 Déploiement du site

Pour le déploiement du site nous avons utilisés Vercel.

Vercel permet depuis un dépôt Github de déployer une application React en s'occupant du build.

Lorsque le dépôt Git est mis à jour, vercel met automatiquement à jour le site.





## 1.7 Réalisation des test

Nous avons réalisé des tests pour le paiement avec Stripe, qui propose un mode test avec une carte de test.

Coque Xiaomi one piece

**70,00 \$US**

Qté 2, 35,00 \$US chacun

Propulsé par **stripe** | [Conditions d'utilisation](#) [Confidentialité](#)

mathis.laversin92@gmail.com

Adresse de livraison **OBLIGATOIRE**

Mathis Laversin

France

Adresse

Ligne d'adresse n°2

Code postal Ville

Données de paiement

Informations de la carte


4242 4242 4242 4242 **VISA**

04 / 24 424

☒ L'adresse de facturation et l'adresse de livraison sont identiques

Enregistrer mes informations en toute sécurité pour le paiement en un clic

Réglez vos achats sur ce site plus rapidement ainsi que partout où Link est accepté.

 06 12 34 56 78 **Facultatif**

[link](#) · [En savoir plus](#)

## 1.8 Annexe

### Code de connexion à l'API de Strapi :

```
("use strict");
const stripe = require("stripe")(process.env.STRIPE_KEY);
/**
 * order controller
 */

const { createCoreController } = require("@strapi/strapi").factories;

module.exports = createCoreController("api::order.order", ({ strapi }) => ({
  async create(ctx) {
    const { products } = ctx.request.body;
    try {
      const lineItems = await Promise.all(
        products.map(async (product) => {
          const item = await strapi
            .service("api::product.product")
            .findOne(product.id);

          return {
            price_data: {
              currency: "eur",
              product_data: {
                name: item.title,
              },
              unit_amount: Math.round(item.price * 100),
            },
            quantity: product.quantity,
          };
        })
      );

      // You, il y a 2 semaines • adding code ...
    } catch (error) {
      // ...
    }
  }
});
```

```
const session = await stripe.checkout.sessions.create({
  shipping_address_collection: {allowed_countries: ['FR']},
  payment_method_types: ["card"],
  mode: "payment",
  success_url: process.env.CLIENT_URL+"?success=true",
  cancel_url: process.env.CLIENT_URL+"?success=false",
  line_items: lineItems,
});

await strapi
  .service("api::order.order")
  .create({ data: { products, stripeId: session.id } });

return { stripeSession: session };
} catch (error) {
  ctx.response.status = 500;
  return { error };
}
},
));
```