

**Rapport de Stage**

**Du 12 mai au 29 juin 2025**



**Maser Engineering  
Green Park Bâtiment 1  
57 avenue Jean Monnet  
31770 Colomiers**

# Sommaire

<b>1. Présentation de la structure d'accueil .....</b>	<b>3</b>
1.1. Présentation générale .....	3
1.2. Localisation .....	3
1.3. Statut juridique .....	4
1.4. Cœur de métier .....	5
1.5. Clients .....	5
1.6. Chiffres clés.....	6
<b>2. Présentation du contexte du stage .....</b>	<b>7</b>
2.1. Maître de stage .....	7
2.2. Positionnement dans l'organisation .....	7
2.3. Missions réalisées .....	7
<b>3. Environnement technique.....</b>	<b>8</b>
3.1. Ressources logicielles.....	8
<b>4. Création d'une boutique d'E-commerce .....</b>	<b>10</b>
4.1. Compétences mises en œuvre .....	10
4.2. Cahier des charges .....	10
4.3. Démarche / Mode opératoire .....	12
4.3.1. Contexte technique du projet.....	13
4.3.2. Présentation de la page d'accueil.....	14
4.3.2.1. Présentation du composant Product card .....	15
4.3.2.2. Présentation du Panier .....	18
4.3.3. Présentation de la page profil section adresses .....	20
4.3.3.1. Présentation du CRUD pour les adresses .....	22
<b>5. Retour d'expérience .....</b>	<b>28</b>
5.1. Remerciements .....	28
5.2. Points positifs .....	28
5.3. Pistes de progrès .....	28

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	2

## 1. Présentation de la structure d'accueil

### 1.1. Présentation générale

Maser Engineering est une société de 561 collaborateurs répartie au sein de plusieurs entités dans toute la France. Son siège social se situe à Paris. Dans la figure 1 nous pouvons voir les différentes implantations de la société en France.

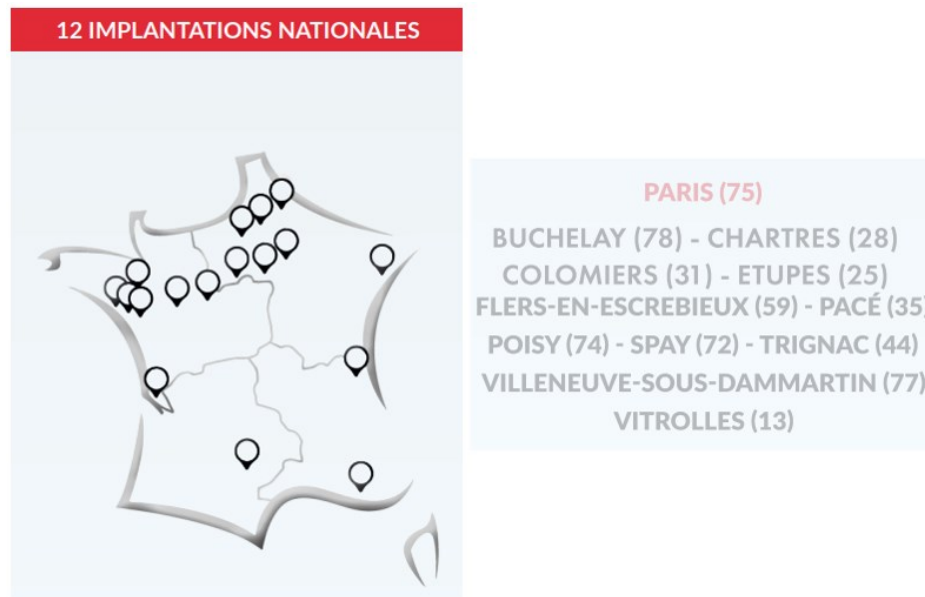


Figure 1 : Entités de Maser Engineering en France

### 1.2. Localisation

L'entité Maser Engineering où j'ai été accueilli se situe à Colomiers au 57 avenue Jean Monnet. Dans la figure 2 nous pouvons voir la localisation des locaux par rapport à l'aéroport Toulouse-Blagnac.

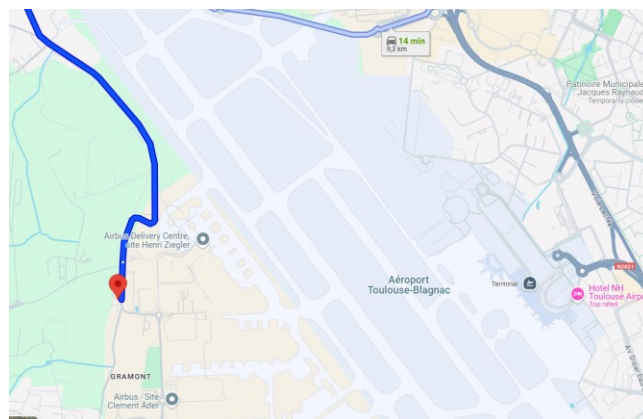


Figure 2 : Localisation de l'entreprise

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	3

Ci-dessous dans la figure 3 nous pouvons voir l'entrée du bâtiment de Colomiers.



Figure 3 : Entrée du bâtiment Maser Engineering

### 1.3. Statut juridique

La société a été créée en 1973. Maser Engineering est une société à responsabilité limitée (SARL), immatriculée sous le numéro SIREN 732050026. Maser Engineering est une filiale du Groupe CRIT (voir figure 4), acteur majeur en Ressources Humaines, Assistance Aéroportuaire, Ingénierie et Maintenance Industrielle, coté sur Euronext Paris.



Figure 4 : Maison mère de Maser Engineering

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	4

## 1.4. Cœur de métier

Maser Engineering contribue au développement de la performance de ses clients dans les secteurs Aéronautique, Aéroportuaire, Agroalimentaire, Automobile, Naval, Eolien et Pharmaceutique (Voir figure 5).

Les quatre pôles d'expertise de Maser Engineering se trouvent dans le conseil, la maintenance, les projets et la formation industrielle.

- Le conseil : Concepteur de solutions, dans une démarche d'amélioration continue, pour favoriser la transformation numérique et la productivité des entreprises.
- la maintenance industrielle : La maintenance des moyens de production, d'exploitation et leurs périphériques est l'une des expertises clés et historiques de Maser Engineering
- La gestion de projets industriel : Maser Engineering accompagne les industriels dans leurs projets globalisés, d'implantation, de transfert et de mise au point, de modernisation et d'optimisation de leurs unités automatisées de production et/ou d'exploitation
- la formation industrielle : le pôle formation de Maser Engineering accompagne les entreprises dans la montée en compétences de leurs ressources internes.

## NOS PRINCIPAUX SECTEURS D'ACTIVITÉ



Figure 5 : Les différents secteurs d'activités

## 1.5. Clients

Cette société travaille pour plus de 280 grands groupes industriels tels qu'Airbus, Airbus, Total Energies, etc... Quelques grands clients sont cités dans la figure 6 ci-dessous.

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	5

## ILS NOUS FONT CONFIANCE



Figure 6 : Quelques gros clients

### 1.6. Chiffres clés

La société Maser Engineering est composée de 561 collaborateurs répartis dans toute la France. Elle travaille pour plus de 300 clients dans le monde et son chiffre d'affaires en 2024 s'élevait à plus de 57 millions d'euros.

#### NOS CHIFFRES CLÉS

Une croissance continue et maîtrisée.

<b>50</b>	<b>57.1</b>	<b>561</b>	<b>300</b>
ans d'expertise	millions d'euros de CA réalisé en 2024	collaborateurs	clients actifs

Figure 7 : Chiffres clés

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	6

## 2. Présentation du contexte du stage

### 2.1. Maître de stage

Mon maître de stage monsieur Valentin Jauze est chef d'équipe au sein de la société Maser Engineering. De plus j'ai été formé par Monsieur Robert Hoffmann, développeur full stack, dans le service informatique.

### 2.2. Positionnement dans l'organisation

L'organisation du pôle conseil dans lequel j'ai effectué mon stage est donné dans la figure 8. J'ai effectué mon stage au sein de l'équipe Business développement Ingénierie de maintenance et méthodes, avec mes maîtres de stage.

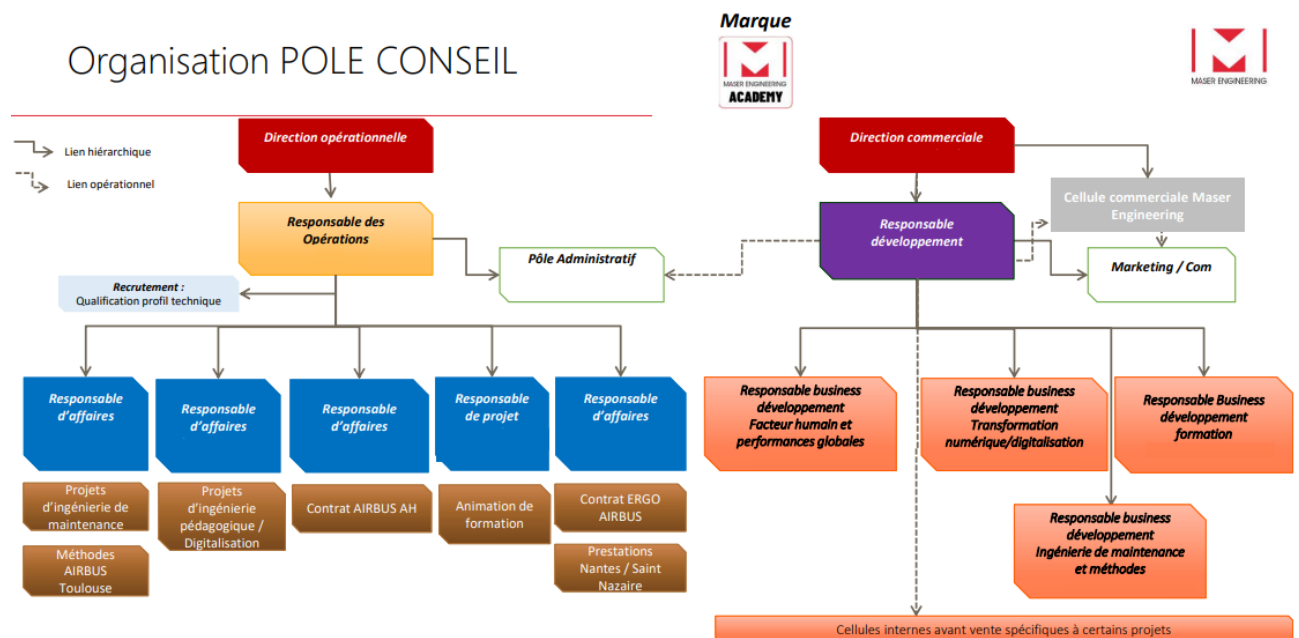


Figure 8 : Organisation du pôle Conseil

### 2.3. Missions réalisées

Le stage a consisté à créer une boutique de E-commerce. Cette boutique sera similaire à la boutique que l'on peut trouver sur le site de la FNAC. Nous pourrions y retrouver les différents produits que l'on peut acheter. Pour chaque produit nous pourrions également trouver une page pour avoir une description détaillée du produit. Il y aura également un panier pour la gestion des produits et une autre partie pour la gestion des données utilisateur.

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	7



### 3. Environnement technique

#### 3.1. Ressources logicielles

Durant mon stage j'ai utilisé différentes technologies.

Dans un premier temps, pour définir la structure de mon site ainsi que les différentes pages et interfaces du site, j'ai utilisé le logiciel Balsamiq (Figure 9). C'est un logiciel de conception de maquettes fonctionnelles (wireframes), facile à utiliser. Les utilisateurs n'ont pas besoin d'avoir de connaissances particulières en webdesign.

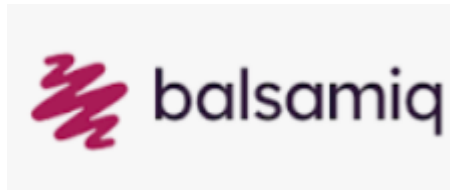


Figure 9 : Outil de maquettage

Pour créer le projet j'ai utilisé la commande NPM de NodeJS.

Avec NPM (Node Package Manager) nous pouvons installer des packages de code Node.js dans l'application. Un paquet node js est un répertoire contenant un ou plusieurs modules ou bibliothèques JavaScript utilisés pour ajouter diverses fonctionnalités aux applications ou aux scripts. Sans paquets, un développeur ou un ingénieur logiciel doit écrire un nouveau code pour chaque fonctionnalité dont son projet a besoin.



Figure 10 : Bibliothèques

Pour permettre de construire des sites dynamiques en liant les données au HTML et pour créer des composants d'interface web nous utilisons VueJS. Ce dernier utilise le TypeScript (TS), c'est une version améliorée de JavaScript qui ajoute des typages de variables pour éviter les erreurs dès l'écriture du code.

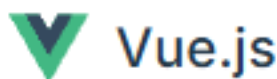


Figure 11 : Framework utilisé (VueJS)

Quasar permet de créer et surtout d'importer des composants d'interface web préfabriqués : c'est une surcouche à Vue JS.

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	8





Figure 12 : Framework utilisé (Quasar)

Pour éditer le code j'ai utilisé Visual Studio Code, ainsi que différentes extensions comme VueJS-officiel, essentielles pour gérer l'auto-complétions et pour interpréter le code.



Figure 13 : Editeur de code

Enfin pour gérer les différentes versions du code, j'ai utilisé le logiciel Git dont on peut voir le logo dans la figure 14.



Figure 14 : Gestionnaire de versions

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	9

## 4. Création d'une boutique d'E-commerce

### 4.1. Compétences mises en œuvre

Compétences issues du BLOC 1 du référentiel - Support et mise à disposition de services informatiques :

- 1) Gérer le patrimoine informatique
- 2) Répondre aux incidents et aux demandes d'assistance et d'évolution
- 3) Travailler en mode projet

### 4.2. Cahier des charges

L'objectif principal du stage est de créer une boutique en ligne.

Description globale :

Le système doit permettre d'accéder librement à la boutique (en tant qu'invité ou utilisateur), de parcourir et d'afficher les produits (nom, prix, catégorie), de rechercher et de filtrer par catégorie, puis de sélectionner et d'ajouter les articles désirés à un panier modifiable (avec retrait individuel possible). Enfin, pour valider la commande et procéder au paiement, l'utilisateur doit se connecter ou créer un compte.

Description détaillée :

Ci-dessous nous pouvons voir la description détaillée du site subdivisé en cinq grandes parties, ainsi que chaque actions et/ou informations disponibles.

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	10

## La Boutique :

### ☐ Page d'accueil

- Listing des produits
  - Pagination
  - Recherche
  - Filtre par catégorie

### ☐ Page produit

- Nom
- Détail
- Prix
- Bouton ajout au panier
- Commentaire
- Proposition de produits de même catégorie

### ☐ Gestion du panier

- Ajout
- Retrait
- Total prix panier

### ☐ Client

- Création du compte
  - Identifiant
  - Vérification email
  - Mot de passe
  - Confirmation du mot de passe
- Connexion au compte
  - Vérification de l'identifiant ou email
  - Vérification mot de passe
  - Mot de passe oublié
- Gestion du compte
  - Changement coordonnées
    - ✓ Identifiant
    - ✓ Mail
    - ✓ Mot de passe
  - Changement des informations bancaire
    - ✓ Listing
      - Numéro carte
      - Date expiration
      - CVV
    - ✓ Ajouter
    - ✓ Modifier
    - ✓ Supprimer
  - Gestion des adresses
    - ✓ Listing
      - Pays
      - Ville
      - Code postal
    - ✓ Création
    - ✓ Modification
    - ✓ Suppression
  - Historique des commandes
    - ✓ Listing

### ☐ Page de checkout (4 étapes)

- Panier
  - Listing
  - Retrait
  - Total des prix
  - Modification quantité
  - Ajout
    - ✓ Listing d'autre produit de même catégorie que le panier
- Livraison
  - Gestion des adresses
    - ✓ Listing
    - ✓ Création
      - Pays
      - Ville
      - Code postal
    - ✓ Total des prix
- Paiement
  - Gestion du paiement
    - ✓ Listing
    - ✓ Création
      - Numéro carte
      - Date expiration
      - CVV
    - ✓ Total des prix
- Récapitulatif

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	11

### 4.3. Démarche / Mode opératoire

A partir du cahier des charges et en utilisant le logiciel Basamiq j'ai ainsi pu maquetter chaque page du site. Grâce à cette approche j'ai obtenu une vue sommaire de l'ensemble du site, qui permet également de visualiser la position des boutons, les icônes, etc... Cela permet aussi de définir les actions qui seront possibles pour l'utilisateur. Le cahier des charges est assez grand, ce qui engendre une grande quantité de page et d'actions à définir. Dans ce rapport nous allons se concentrer sur la page d'accueil afin de pouvoir les détailler le plus précisément possible. Dans une seconde étape nous verrons les différents composants web que j'ai utilisé afin de pouvoir développer le site proprement dit.

La figure 15 représente une vue schématique de la page d'accueil. Sa mise en forme et les différents composants à l'intérieur permettent de respecter le cahier des charges des fonctionnalités pour l'utilisateur, comme le listing des produits, la possibilité d'effectuer la recherche d'un produit à l'aide d'un mot clé, ou bien la recherche à partir de sa catégorie.

Les autres fonctionnalités inclus dans cette page sont : un clic sur l'icône "Panier" qui permet d'ouvrir un volet (drawer) pour gérer le panier ; un clic sur l'icône "Profil" qui permet de se rendre à la page de connexion ; un clic sur le bouton "+Panier" qui permet d'ajouter le produit sélectionné au panier ; enfin un clic sur le bouton "Voir+" qui permet d'accéder à la page de détail du produit sélectionné.

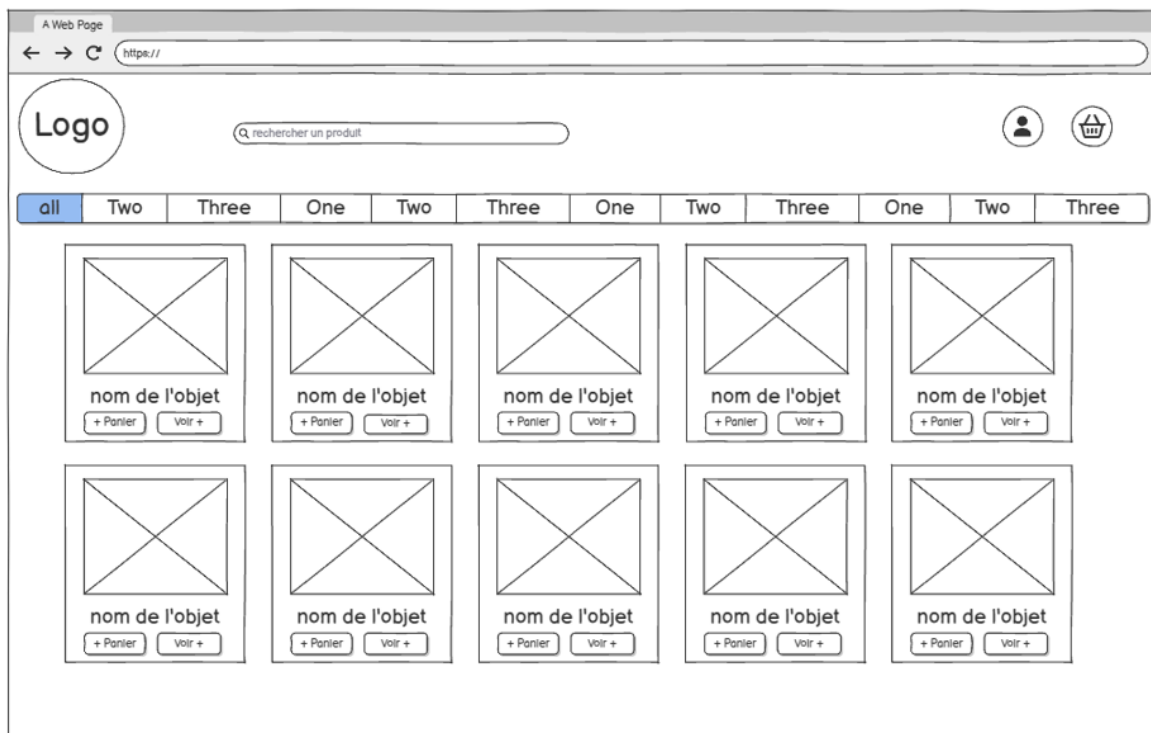


Figure 15 : Page d'accueil

De la même manière via l'outil basalmiq on peut voir dans la figure 16 la page du profil utilisateur avec notamment la disposition de l'onglet adresses. Sa mise en forme et les

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	12

différents composants à l'intérieur permettent de respecter le cahier des charges des fonctionnalités pour l'utilisateur, comme la liste des sections pour chaque onglet du site.

Les autres fonctionnalités inclus dans cette page sont : un clic sur l'icône "Modifier" qui permet d'ouvrir un formulaire pour mettre à jour des informations sur les adresses ; un clic sur l'icône "Supprimer" qui permet de supprimer l'adresse sélectionnée ; un clic sur le bouton "Ajouter" qui permet d'afficher un formulaire pour d'intégrer une nouvelle adresse dans la liste.




Figure 16 : Gestion des adresses du profil utilisateur

#### 4.3.1. Contexte technique du projet

Tout d'abord à travers cette partie du rapport nous allons présenter plusieurs pages de la boutique. Nous en profiterons pour visualiser une partie de mon travail. Dans un souci de clarté je vais expliquer ce qui a été nouveau pour moi mais également par rapport à la formation SIO SLAM de première année.

A partir des différentes pages créées avec le logiciel Basalmiq et du cahier des charges j'ai pu trouver comment me prendre pour mener à bien ce projet. Tout d'abord j'ai créé une petite base de données, mais sur les conseils de Monsieur Robert Hoffmann, cette création a été faite avec des fichiers JSON (voir figure 17). Cela m'a permis de partir d'un brouillon facile à mettre à jour et bien plus simple à mettre en place, étant donné que je ne ferais que la partie front-end du site.

Il est important de souligner le fait qu'idéalement un fichier JSON correspond à une table de la base de données. Or dans un souci de contrainte cela ne s'applique pas systématiquement. De plus le site repose sur le local store : c'est un stockage interne au

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	13

navigateur. Ceci permet de conserver les informations même si la page est rafraichie. Tout cela a pour but de donner l'expérience utilisateur la plus fidèle possible d'une boutique E-commerce mais sans la lourdeur du côté back-end.

```
products.json ×
src-ssr > custom > mock-data > products.json > ...
1
2 { "id": 1, "name": "clavier", "price": 89, "category": "périphérique" },
3 { "id": 2, "name": "souris", "price": 49, "category": "périphérique" },
4 { "id": 3, "name": "écran", "price": 320, "category": "affichage" },
5 { "id": 4, "name": "imprimante", "price": 215, "category": "impression" },
6 { "id": 5, "name": "routeur", "price": 130, "category": "réseau" },
7 { "id": 6, "name": "casque", "price": 78, "category": "audio" },
8 { "id": 7, "name": "caméra", "price": 145, "category": "vidéo" },
9 { "id": 8, "name": "téléphone", "price": 999, "category": "mobile" },
10 { "id": 9, "name": "scanner", "price": 170, "category": "impression" },
11 { "id": 10, "name": "webcam", "price": 65, "category": "vidéo" },
12 { "id": 11, "name": "micro", "price": 59, "category": "audio" },
13 { "id": 12, "name": "enceinte", "price": 110, "category": "audio" },
14 { "id": 13, "name": "chargeur", "price": 29, "category": "accessoire" },
15 { "id": 14, "name": "carte mémoire", "price": 25, "category": "stockage" },
16 { "id": 15, "name": "hdd externe", "price": 140, "category": "stockage" },
17 { "id": 16, "name": "tablette", "price": 1152, "category": "mobile" },
18 { "id": 17, "name": "projecteur", "price": 540, "category": "affichage" },
19 { "id": 18, "name": "switch", "price": 300, "category": "réseau" },
20 { "id": 19, "name": "ssd", "price": 160, "category": "stockage" },
21 { "id": 20, "name": "stylius", "price": 45, "category": "accessoire" },
22 { "id": 21, "name": "moniteur", "price": 400, "category": "affichage" },
23 { "id": 22, "name": "boîtier PC", "price": 105, "category": "composant" },
24 { "id": 23, "name": "ventilateur", "price": 35, "category": "composant" },
25 { "id": 24, "name": "câble HDMI", "price": 18, "category": "accessoire" },
26 { "id": 25, "name": "adaptateur", "price": 22, "category": "accessoire" },
27 { "id": 26, "name": "batterie externe", "price": 80, "category": "accessoire" },
28 { "id": 27, "name": "station d'accueil", "price": 150, "category": "accessoire" },
29 { "id": 28, "name": "logiciel antivirus", "price": 60, "category": "logiciel" },
30 { "id": 29, "name": "clef USB", "price": 15, "category": "stockage" },
31 { "id": 30, "name": "support écran", "price": 55, "category": "accessoire" }
32
```

Figure 17 : Base de données des produits

Enfin nous verrons uniquement la page d'accueil et la page profil. Les autres pages du site sont similaires en termes de codes. Il n'y a que le contexte qui change. C'est pourquoi je ne présente pas tout le projet dans son intégralité.

#### 4.3.2. Présentation de la page d'accueil

Une fois la base de données des produits établis, j'ai réalisé la page d'accueil dont on peut voir la version finale vue par l'utilisateur ci-dessous dans la figure 18.

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	14

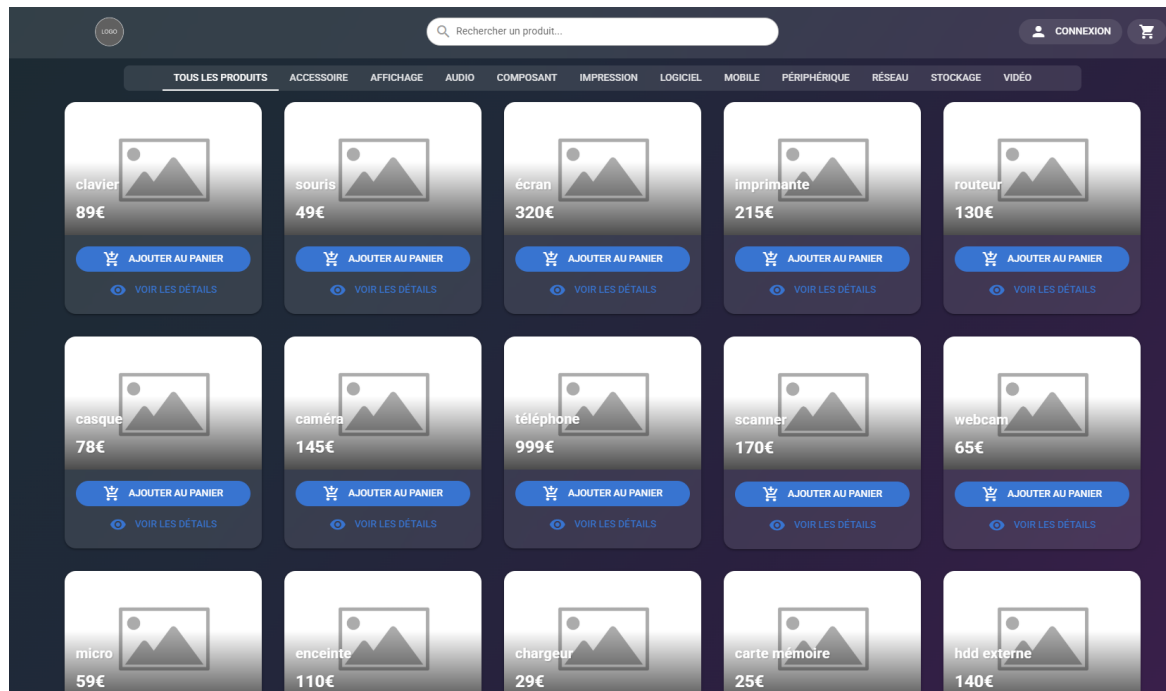


Figure 18 : Page d'accueil

#### 4.3.2.1. Présentation du composant Product card

Le premier composant que nous allons détailler est le “product card”. Il est utilisé dans la page d'accueil pour permettre de lister chaque produit présent dans la base de données tout en évitant une répétition de code. Vous pouvez le visualiser dans la figure 19.



Figure 19 : Composant Product card

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	15



Dans la figure 20 on peut voir la structure du “product card”.

```
<template>
  <q-card class="product-card q-ma-md" style="max-width: 280px">
    <q-img src="../../public/Placeholder.png">
      <div class="absolute-bottom product-overlay">
        <div class="text-h6 text-white product-name">{{ props.product.name }}</div>
        <div class="text-h5 text-white price">{{ props.product.price }}€</div>
      </div>
    </q-img>
    <q-card-actions class="q-pa-md">
      <q-btn @click="addToBasket" color="primary" unelevated rounded icon="add_shopping_cart" label="Ajou">
      <q-btn @click="goToProductPage" flat color="primary" icon="visibility" label="Voir les détails" cla
    </q-card-actions>
  </q-card>
</template>
```

Figure 20 : Structure du “product card”

Dans cette structure nous allons détailler les fonctionnalités suivantes :

1. L’affichage d’un “product card” pour chaque produit dans la base de données.
2. La possibilité d’aller sur la page du produit qui nous intéresse.
3. Un exemple de CSS du “product card”.

Le Product card est structuré de façon à ce qu’il se répète (Voir figure 21) grâce à un v-for dans la page d’accueil pour chaque produit de la base de données grâce à son identifiant (ID). Tout cela grâce à Vue JS.

```
<template v-for="product in results" :key="product.id">
  <ProductCardComponent :product="product"></ProductCardComponent>
</template>
```

Figure 21 : v-for du “product card”

Dans la figure 22, nous pouvons voir la définition du type “product” grâce au Type Script. Ce type est essentiel pour pouvoir manipuler les produits et leurs caractéristiques.

```
export interface Product {
  id: number;
  name?: string | null;
  price?: number | null;
  category?: string | null;
}
```

Figure 22 : Définition du type “Product”

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	16

Le nom et le prix du composant sont dit “databind”, c’est à dire que leur nom et leur prix sont récupérés en pointant sur la propriété de l’objet concerné (voir figure 23). Le bouton “Voir les détails” permet d’aller à la page de l’article grâce à son ID.

```
<div class="text-h6 text-white product-name">{{ props.product.name }}</div>
```

Figure 23 : Databind du nom du produit dans une balise HTML

Dans la structure du “product card” nous pouvons y trouver des composants qui sont précédé de leur appellation par un “q-”. Ce sont des composants Quasar comme dans la figure 24 ci-dessous. Nous pouvons y voir un bouton Quasar avec une class CSS Quasar qui appelle la fonction “go to product page”.

```
<q-btn @click="goToProductPage" class="full-width" />
```

Figure 24 : Bouton Quasar appelant la fonction “go to product page”

Dans la figure 25, nous retrouvons la fonction “go to product page” appelée par le bouton.

```
const api = useApi();
const router = useRouter();

const props = defineProps<{
  product: Product;
}>();

const goToProductPage = async () =>
{
  await router.push(`/product/${props.product.id}`);
};
```

Figure 25 : Fonction “go to product page”

Enfin nous pouvons retrouver un exemple de CSS du composant Product card dans la figure 26.

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	17

```

style scoped>
/* #region Card Styles */
.product-card {
  transition: all 0.3s ease;
  border-radius: 16px;
  overflow: hidden;
  background: rgba(255, 255, 255, 0.1);
  backdrop-filter: blur(10px);
}

.product-card:hover {
  transform: translateY(-4px);
  background: rgba(255, 255, 255, 0.15);
}
/* #endregion */

/* #region Image Styles */
.product-image {
  transition: transform 0.3s ease;
}

.product-card:hover .product-image {
  transform: scale(1.05);
}
/* #endregion */

```

Figure 26 : CSS du “product card”

#### 4.3.2.2. Présentation du Panier

Le second composant que nous allons étudier est un bouton panier (voir figure 27) présent dans la page d'accueil. Il affiche le nombre d'article dans le panier et lorsqu'on clique dessus un volet s'ouvre à droite de la page.



**Figure 27 : Panier “drawer”**

La structure du “drawer” (volet du panier) est visible dans la figure 28.

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	18

```
<q-drawer side="right" v-model="drawerRight" show-if-above :width="300" :breakpoint="500" class="cart-drawer">
  <q-scroll-area class="fit">
    <template v-for="product in basket" :key="product.id">
      <div class="q-ma-md">
        <q-card class="cart-card q-mx-xl">
          <q-card-section class="q-pt-none">
            <div class="text-h6 text-white">{{ product.name }}</div>
            <div class="text-subtitle2 text-white">{{ product.price }}€</div>
            <div class="row items-center">
              
            </q-card-section>
          </q-card>
        </div>
      </template>
      <div class="column items-center q-mt-md q-mb-xl">
        <div class="text-h6 q-mb-md text-white">Total à payer : {{ local.basketPrice }}€</div>
        <q-btn color="primary" label="Acheter" @click="handleCheckout" size="lg" style="width: 200px" :disabled="lc
      </div>
    </q-scroll-area>
  </q-drawer>
```

Figure 28 : Structure du drawer

Dans cette structure nous allons détailler les fonctionnalités suivantes :

1. L’affichage des produits enregistrés dans le panier par l’utilisateur via le local store.
2. La suppression éventuelle de produits dans le panier via le local store.
3. Un exemple de CSS du “drawer”.

Dans la structure du drawer je fais appel aux variables globales du store Pinia.

```
getters: {
  basketCount: (state) => state.data.basket.length,
  basketPrice: (state) => state.data.basket.reduce((total, item) => total + (item.price ?? 0), 0),
  isLoggedIn: (state) => state.data.currentUser !== null,
  userName: (state) => state.data.currentUser?.name || 'Invité',
},
```

Figure 29 : Déclaration des variables globales avec Pinia (bibliothèque de gestion d’état)

Je récupère les produits dans le panier en les répétant avec leur ID dans des composants “cart-card” grâce à un v-for, comme indiqué dans la figure 30.

```
<template v-for="product in basket" :key="product.id">
```

Figure 30 : v-for du composant “cart-card” pour chaque ID

Pour la suppression des produits un bouton Quasar est à disposition de l’utilisateur. Ce bouton appelle une fonction “delete product in basket”.

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	19

```
<q-btn class="q-ml-sm" @click="api.deleteProductInBasket(product)" icon="cancel" color="red" />
```

Figure 31 : Bouton Quasar appelant la fonction “delete product in basket”

Dans la figure 32 nous retrouvons la fonction “delete product in basket” appelée par le bouton.

```
const deleteProductInBasket = (product: Product): boolean =>
{
  local.deleteProductFromBasket(product);
  return true;
};
```

Figure 32 : Fonction “delete product in basket” dans le local store

Dans la figure 33 on peut voir un exemple du code CSS du composant drawer.

```
/* #region Cart Drawer Styles */
.cart-drawer
{
  background: rgba(0, 0, 0, 0.3);
  backdrop-filter: blur(15px);
}

.cart-card
{
  background: rgba(255, 255, 255, 0.1);
  backdrop-filter: blur(10px);
  transition: all 0.3s ease;
}

.cart-card:hover
{
  background: rgba(255, 255, 255, 0.15);
  transform: translateY(-2px);
}
/* #endregion Cart Drawer Styles */
```

Figure 33 : CSS du drawer

#### 4.3.3. Présentation de la page profil section adresses

Une fois la page d'accueil terminé, il est assez simple d'établir la page profile. Dans un premier lieu je structure une base de données pour mes utilisateurs. Par la suite nous verrons la section adresses de la page et les différentes fonctionnalités que l'on peut y retrouver.

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	20

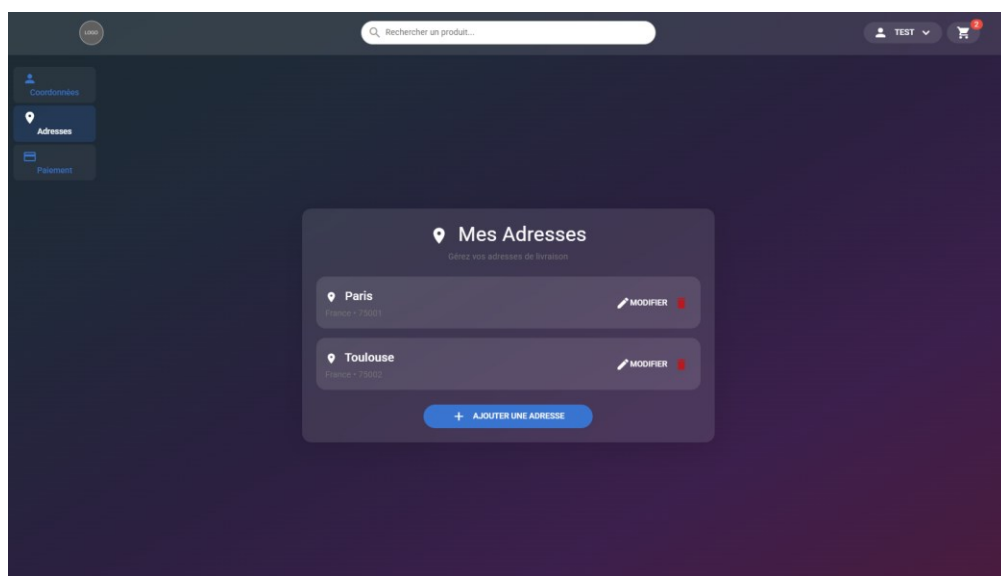
```

users.json ×
src-ssr > custom > mock-data > users.json > {} 1
1  [
2    {
3      "id": 1,
4      "name": "test",
5      "email": "test@gmail.com",
6      "password": "123",
7      "adresses": [
8        {
9          "country": "France",
10         "city": "Paris",
11         "postalCode": "75001"
12       },
13       {
14         "country": "France",
15         "city": "Toulouse",
16         "postalCode": "75002"
17       }
18     ],
19     "basketCards": [
20       {
21         "cardNumber": "1234567890123456",
22         "expirationDate": "12/25",
23         "cryptogram": "123"
24       }
25     ]
26   },

```

**Figure 34 : Base de données des utilisateurs**

Une fois la base de données des utilisateurs établis, j'ai réalisé la page profil section adresses dont on peut voir la version finale vue par l'utilisateur ci-dessous. Nous allons voir cette page comme un gros composant qui offre la possibilité de gérer ces adresses entièrement.



**Figure 35 : Page profil section adresses**

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	21

#### 4.3.3.1. Présentation du CRUD pour les adresses

Dans cette page nous allons détailler les fonctionnalités suivantes :

1. La possibilité de créer une nouvelle adresse.
2. L’affichage des adresses existante.
3. La possibilité de modifier une adresse.
4. La possibilité de supprimer une adresse.

Les fonctionnalités que nous venons de citer correspond à un CRUD (pour Create, Read, Update, Delete) nous allons voir cela dans le cadre des adresses de l’utilisateur “test”.

Ici le Create correspond au bouton “ajouter une adresse”, qui ouvre un formulaire pour ajouter une nouvelle adresse à la liste.

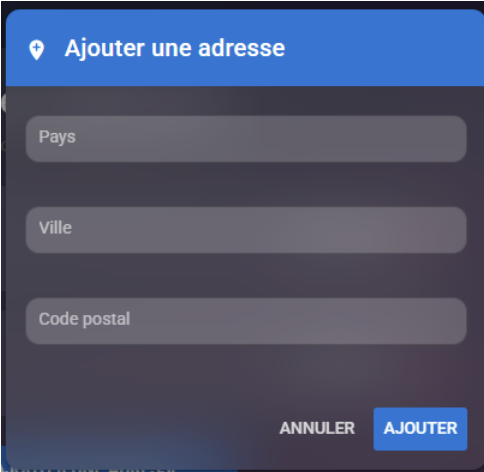


Figure 36 : Formulaire d’ajout d’une adresse

Ci-dessous la structure du formulaire permettant à un utilisateur d’ajouter une nouvelle adresse.

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	22



```
<!-- #region Add Dialog -->
<q-dialog v-model="showAddAddressForm">
  <q-card style="min-width: 400px" class="rounded-borders">
    <q-card-section class="bg-primary text-white">
      <div class="text-h6">
        <q-icon name="add_location" class="q-mr-sm" />
        Ajouter une adresse
      </div>
    </q-card-section>

    <q-card-section class="q-pt-lg">
      <div class="q-gutter-md">
        <q-input v-model="editAddressCountry" label="Pays" outlined dense prepend-icon="public" :rules="[(val) => !!val || 'Le']">
        <q-input v-model="editAddressCity" label="Ville" outlined dense prepend-icon="location_city" :rules="[(val) => !!val |
        <q-input v-model="editAddressPostalCode" label="Code postal" outlined dense prepend-icon="markunread_mailbox" :rules="
      </div>
    </q-card-section>

    <q-card-actions align="right" class="q-pa-md">
      <q-btn flat label="Annuler" class="btn-cancel" v-close-popup />
      <q-btn label="Ajouter" color="primary" unelevated v-close-popup @click="addAddress" />
    </q-card-actions>
  </q-card>
</q-dialog>
<!-- #endregion -->
```

Figure 37 : Structure du formulaire d'ajout d'une adresse

Dans la figure 38, nous pouvons voir la définition du type "Adress" grâce au Type Script. Ce type est essentiel pour pouvoir manipuler les adresses et leurs caractéristiques.

```
export interface Address {
  country?: string | null;
  city?: string | null;
  postalCode?: string | null;
}
```

Figure 38 : Définition du type "adress"

Ici nous retrouvons un bouton Quasar à disposition de l'utilisateur. Ce bouton appelle une fonction "add address".

```
<q-btn label="Ajouter" color="primary" unelevated v-close-popup @click="addAddress" />
```

Figure 39 : Bouton de validation du formulaire appelant la fonction "add address"

Voici la fonction "add address" appelée par le bouton correspondant au Create du CRUD.

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	23

```
// #region Variables
const api = useApi();
const $q = useQuasar();
const editAddressCountry = ref('');
const editAddressCity = ref('');
const editAddressPostalCode = ref('');
const showEditAddressForm = ref(false);
const showAddAddressForm = ref(false);
const addresses = ref<Address[]>([]);
const editingAddress = ref<Address | null>(null);
// #endregion

const addAddress = () => {
  if (!editAddressCountry.value || !editAddressCity.value || !editAddressPostalCode.value) {
    notify('Veuillez remplir tous les champs', 'negative');
    return;
  }

  addresses.value.push({
    country: editAddressCountry.value,
    city: editAddressCity.value,
    postalCode: editAddressPostalCode.value,
  });

  const result = api.updateUserAddresses(addresses.value);
  if (result === true) {
    notify('Adresse ajoutée avec succès', 'positive');
    editAddressCountry.value = editAddressCity.value = editAddressPostalCode.value = '';
  } else {
    notify(typeof result === 'object' ? result.message : 'Erreur lors de l\'ajout', 'negative');
  }
};
```

**Figure 40 : Fonction “add address”**

Le Read correspond au listing des adresses de l'utilisateur test. Cela se passe via le composant adresse banner ci-dessous.



**Figure 41 : Composant “adresses banner”**

Ci-dessous la structure du “adresses banner”.

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	24

```
<div v-if="adresses.length > 0" class="q-mb-lg">
  <div v-for="(address, index) in adresses" :key="index" class="q-mb-md">
    <q-card flat bordered class="address-card address-card-hover">
      <q-card-section class="q-pa-md">
        <div class="row items-center">
          <div class="col">
            <div class="text-h6 text-primary q-mb-xs">
              <q-icon name="place" class="q-mr-sm" />
              {{ address.city }}
            </div>
            <div class="text-body2 text-grey-7">
              {{ address.country }} • {{ address.postalCode }}
            </div>
          </div>
          <div class="col-auto">
            <q-btn-group flat>
              <q-btn flat dense icon="edit" color="primary" class="q-mr-xs" @click="showEditAddi">
                <q-tooltip>Modifier</q-tooltip>
              </q-btn>
              <q-btn flat dense icon="delete" color="negative" @click="deleteAddress(address)">
                <q-tooltip>Supprimer</q-tooltip>
              </q-btn>
            </q-btn-group>
          </div>
        </div>
      </q-card-section>
    </q-card>
  </div>
</div>
```

Figure 42 : Structure du composant “adresses banner”

Je récupère les adresses dans la base de données et dans le local store si l'utilisateur a ajouté des adresses auparavant sur le site. Puis je parcours avec un v-for un tableau regroupant toutes les d'adresses pour les répéter dans le composant “adresses banner”. Tout cela correspond au Read du CRUD.

```
<div v-for="(address, index) in adresses" :key="index" class="q-mb-md">
```

Figure 43 : v-for du “adresses banner” pour chaque adresse dans un tableau

Update correspond au bouton “modifier”, qui ouvre un formulaire affichant les informations de l'adresse existante tout en laissant la possibilité de les modifier (Figure 44).



Figure 44 : Formulaire de modification

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	25

Dans la figure 45, on retrouve la structure du formulaire de modification d'une nouvelle adresse.

```
<q-dialog v-model="showEditAddressForm">
  <q-card style="min-width: 400px" class="rounded-borders">
    <q-card-section class="bg-primary text-white">
      <div class="text-h6">
        <q-icon name="edit_location" class="q-mr-sm" />
        Modifier l'adresse
      </div>
    </q-card-section>

    <q-card-section class="q-pt-lg">
      <div class="q-gutter-md">
        <q-input v-model="editAddressCountry" label="Pays" outlined dense prepend-icon="public" :rule
        <q-input v-model="editAddressCity" label="Ville" outlined dense prepend-icon="location_city"
        <q-input v-model="editAddressPostalCode" label="Code postal" outlined dense prepend-icon="mar
      </div>
    </q-card-section>

    <q-card-actions align="right" class="q-pa-md">
      <q-btn flat label="Annuler" class="btn-cancel" v-close-popup />
      <q-btn label="Enregistrer" color="primary" unelevated v-close-popup @click="updateAddress" />
    </q-card-actions>
  </q-card>
</q-dialog>
```

Figure 45 : Structure du formulaire de modification adresse

Ici nous retrouvons un bouton Quasar à disposition de l'utilisateur. Ce bouton appelle une fonction "update address".

```
<q-btn label="Enregistrer" color="primary" unelevated v-close-popup @click="updateAddress" />
```

Figure 46 : Bouton de validation du formulaire appelant la fonction "update address"

Voici la fonction "update address" appelée par le bouton correspondant à l'Update du CRUD.

```
// #region CRUD Functions
const updateAddress = () => {
  const index = addresses.value.findIndex((addr) => addr === editingAddress.value);

  // si index n'est pas vide
  if (index !== -1) {
    addresses.value[index] = {
      country: editAddressCountry.value,
      city: editAddressCity.value,
      postalCode: editAddressPostalCode.value,
    };

    const result = api.updateUserAddresses(addresses.value);
    if (result === true) {
      notify('Adresse modifiée avec succès', 'positive');
    } else {
      notify(typeof result === 'object' ? result.message : 'Erreur lors de la modification', 'negative');
    }
  } else {
    notify('Erreur lors de la modification', 'negative');
  }
};
```

Figure 47 : Fonction "update address"

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	26

Enfin le Delete correspond au bouton supprimer, qui supprime de la liste et du local store l'adresse sélectionnée.

```
<q-btn flat dense icon="delete" color="negative" @click="deleteAddress(address)">
```

Figure 48 : Bouton de suppression d'adresse appelant la fonction "delete address"

Voici la fonction "delete address" appelée par le bouton correspondant au Delete du CRUD.

```
const deleteAddress = (address: Address) => {
  const index = addresses.value.findIndex(
    (addr) =>
      addr.country === address.country &&
      addr.city === address.city &&
      addr.postalCode === address.postalCode,
  );

  if (index !== -1) {
    addresses.value.splice(index, 1);
    const result = api.updateUserAddresses(addresses.value);
    if (result === true) {
      notify('Adresse supprimée avec succès', 'positive');
    } else {
      notify(typeof result === 'object' ? result.message : 'Erreur lors de la suppression', 'negative');
    }
  } else {
    notify('Erreur lors de la suppression', 'negative');
  }
};
```

Figure 49 : Fonction "delete address"

La fonction "delete address" appelle aussi une fonction "delete product in basket" qui a dû être obligatoirement défini dans le local store.

```
const deleteProductInBasket = (product: Product): boolean =>
{
  local.deleteProductFromBasket(product);
  return true;
};
```

Figure 50 : Fonction "delete product in basket"

Nous avons fini de présenter la page d'accueil et la page profil. Pour rappel, ce n'est qu'une partie de la boutique que je vous présente, car le fonctionnement technique des autres pages est identique, il y a uniquement que le contexte d'usage qui change.

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	27

## 5. Retour d'expérience

### 5.1. Remerciements

Je voudrais ici remercier monsieur Valentin Jauze et monsieur Robert Hoffmann, mes maitres de stage qui m'ont d'abord accepté au sein de leur équipe de la société Maser Engineering pour effectuer mon stage. Leur accueil chaleureux et leur bienveillance m'ont permis de m'épanouir.

Je tiens à remercier monsieur Valentin Jauze de m'avoir donné ma chance et de m'avoir accompagné tout au long de cette expérience professionnelle avec beaucoup de patience et de pédagogie.

Je remercie également monsieur Robert Hoffmann pour ses conseils au quotidien, le partage de son expérience professionnelle et la transmission de ses méthodes de travail.

Enfin, je tiens à remercier toutes les personnes qui m'ont conseillé et relu lors de la rédaction de ce rapport de stage : ma famille et mes amis.

### 5.2. Points positifs

Parmi les points positifs de ce stage le premier qui me vient à l'esprit est l'accueil et l'environnement de travail qui est vraiment excellent au sein de Maser Engineering. Que ce soit au niveau de l'open-space où j'ai effectué mon stage, ou au niveau des mes maitres de stage, toutes les personnes ont vraiment été patiente et bienveillante à mon égard. J'en retient un excellent souvenir.

Ensuite le sujet sur lequel j'ai travaillé a été très intéressant et représentatif de la réelle charge de travail pour réaliser une boutique en ligne. De plus j'ai appliqué certaine partie des cours de base de données et de développement.

### 5.3. Pistes de progrès

Il aurait été intéressant d'avoir plus de temps afin de mettre en place une véritable base de données.

Une autre partie qui aurait été aussi intéressante d'aborder est la gestion du site web par les administrateurs.

<b>Etudiant</b>	Hontans Sylvain	<b>Tuteur</b>	Jauze Valentin	<b>Entreprise</b>	Maser Engineering
<b>Année Scolaire</b>	2024/2025	<b>Section</b>	1SIO	<b>Page</b>	28