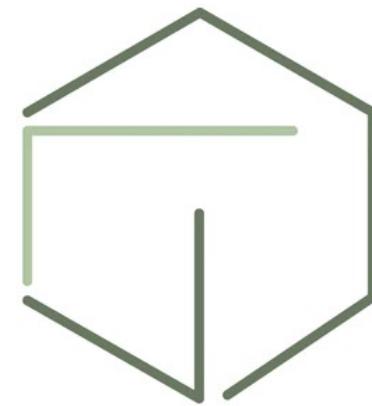


AVRIL-JUIN 2022

CONCEPTION ET RÉALISATION D'UN SITE WEB E-COMMERCE

RAPPORT DE STAGE



Grønt Lab

STAGIAIRES :

Hicham GUENDOUZ

Lindy FABIGNON

Fabrice FERREIRA

Marie CISSOKHO

Table des Matières

REMERCIEMENTS.....	2
INTRODUCTION DU PROJET	3
PRESENTATION ET RESUME DU PROJET.....	4
REUNION DE PROJET.....	5
Besoins des utilisateurs et de l'administrateur.....	6
EXPRESSIONS DES BESOINS DU CAHIER DES CHARGES.....	7
REUNION D'EQUIPE.....	8
Trello.....	9
GitHub.....	10
Figma.....	11
Visual Studio Code.....	12
Xampp(Mysql-MariaDB).....	13
Répartition des tâches.....	14
Conception maquette.....	15
BASE DE DONNEES.....	18
Modèle Physique de Données.....	19
Modèle Logique de Données.....	20
MVC.....	21
Contrôleur.....	22
EXTRAITS DE CODE LES PLUS SIGNIFICATIFS.....	23
BACK-END.....	23
FRONT-END.....	32
CONCLUSION.....	35
ANNEXE.....	36
Cahier des charges.....	36
Capture d'écran du site	41

Remerciements

Mes sincères remerciements

À la boutique GrontLab pour sa confiance en une équipe de stagiaires ;

À Fabrice FERREIRA, Lindy FABIGNON et Marie CISSOKHO pour avoir mené à terme le projet et le rapport de stage ;

À l'ensemble des stagiaires de la formation pour leur entraide et leur convivialité ;

À notre formateur Patrick ROUYER pour la qualité des cours, ainsi que sa patience et sa disponibilité ;

À l'AFPA Champs-sur-Marne pour avoir mis à disposition un espace de travail.

Introduction du projet

De nos jours, le commerce électronique est considéré comme un indispensable si l'on souhaite la prospérité de son projet de commerce. L'interface, la communication et le visuel sont primordiaux. Le développement web s'agit, essentiellement, d'assurer aux utilisateurs et aux clients, une application fiable, sécurisée, pratique et adaptée aux besoins de ces derniers.

Le projet porte sur la création de A à Z d'un site web e-commerce pour un professionnel : une boutique de bien-être, ayant eu recours à la plateforme Shopify pour la création de son site(pratique car rapide). Au niveau de l'originalité et des détails, les CMS(Content Management System) ne valent pas l'efficience de la programmation qui, elle, donne la possibilité de satisfaire les besoins au détail près du commanditaire.

Dans la première partie, nous allons présenter le cahier des charges, ainsi que les besoins des utilisateurs et de l'administrateur.

La deuxième partie sera consacrée à la gestion du projet et aux divers choix concernant les outils et répartitions des tâches.

Dans la dernière partie, on étudiera et donnera un aperçu des pages et vues réalisées tout en montrant des lignes de codes les plus importantes.

Présentation et résumé du projet

Dans le cadre de la formation développeur web et web mobile, avril et Juin ont été consacrés à un stage en entreprise. Après avoir démarché une boutique et défini les attentes de nos futurs tuteurs, nous avons été amenés à réaliser un site e-commerce, de façon théorique dans un premier temps (Cahier des charges, Conception maquette), puis ensuite concrètement en codant (langages php, javascript, html, CSS et framework « Symfony »).

Le rapport de stage présentera dans sa globalité le projet Gront Lab, les problématiques rencontrées et solutions apportées, ainsi que le retour d'expérience personnel.

Réunion de projet

La réunion de lancement de projet a eu lieu le premier jour du stage. Elle a permis d'établir la date de démarrage officielle du projet, ainsi que les autres dates clés, d'exposer les objectifs et la vision du projet.

Ainsi, en concertation avec le commanditaire, il a été convenu de mettre en place un système de télétravail (boutique étriquée) et d'un comité de pilotage pour animer le projet : réunions hebdomadaires pour en évaluer l'avancement, définir les actions à mener, suivre un planning(cf Trello).

La présentation de l'entreprise, l'exposition du contexte et des objectifs du projet, l'identification du périmètre fonctionnel du projet (technologies à utiliser, charte graphique et éditoriale, définition des contenus...), une reformulation validant les demandes du commanditaire ont rendu l'exercice de rédaction du cahier des charges (chapitre suivant) plus simple.

Besoin des utilisateurs et de l'administrateur

L'utilisateur doit pouvoir :

- Consulter les produits de la boutique ,
- Construire un panier ,
- Envoyer un message a l'entreprise via la page contact ,
- Connaitre les détails d'un produit ,
- Commenter un produit ,
- Commenter un événement ,
- Consulter le site depuis un smartphone ,une tablette ou un ordinateur.

L'administrateur doit pouvoir :

- Publier un nouveau produit,
- publier un événement ,
- Mettre à jour de la page d'accueil ,
- Présenter l'entreprise ainsi que les partenaire « A propos »,
- Modérer les commentaires sur l'ensemble du site ,
- Choisir le produit mise en valeur dans « notre sélection ».

Expressions des besoins du cahier de charges

Le cahier des charge a pour objectif d'exprimer les points suivants :

1. Les objectifs du site
2. La cible adressée par le site
3. Objectif quantitatifs après 1 an
4. Périmètre du projet
5. Charte graphique
6. Logo du site
7. Arborescence du site
8. Description fonctionnelle du site
9. Fiche produits
10. Espace client
11. Description fonctionnelle du Back-Office
12. Fonctionnalités standards
13. Gestion du front
14. Gestion du Back-Office
15. Information relatives aux contenus
16. Contraintes techniques
17. Prestations attendues
18. Planning
19. Méthodologie suivi
20. Attribution des rôles
21. Suivi du projet
22. Propriété

Reunion d'équipe

Le projet a été géré de la façon suivante :

- Les outils qui seront utilisés pour le projet ,
- Choix des langages utilisés ,
- La gestion du temps de travail et répartition des taches pour l'ensemble des développeurs stagiaires participant au projet(Cf Trello),
- La création d'un espace dédié au stockage du projet(Cf GitHub) ,
- L'éditeur de Code qui sera utilisé pour le projet(CF Visual Studio Code).

Outils utilisés pour le projet :

Logiciel utilisés pour les maquettes :

- FIGMA

Logiciel pour organiser et repartir les taches :

- TRELLO

Logiciel éditeur de code :

- Visual Studio Code

Logiciel de de versionnage de projet en ligne :

- GITHUB

Langages utilisés pour le projet :

HTML : Ce langage est utilisé pour créer des pages web. L'acronyme signifie *HyperText Markup Language*.

CSS : Acronyme de « Cascading Style Sheets » ce qui signifie « feuille de style en cascade ».Le CSS correspond à un langage informatique permettant de mettre en forme des pages web (HTML)

Javascript : Langage de script orienté objet. Il permet, entre autre, d'introduire sur une page web ou HTML des petites animations ou des effets.

Php : Pour Hypertext Preprocessor, désigne un langage informatique, ou un langage de script, utilisé principalement pour la conception de sites web dynamiques.

Symfony: Symfony est un framework qui représente un ensemble de composants (aussi appelés librairies) PHP autonomes qui peuvent être utilisés dans des projets web privés ou open source. Mais c'est également un puissant Framework PHP

Trello

Trello repose sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Les cartes sont assignables à des utilisateurs et sont mobiles d'une planche à l'autre, traduisant leur avancement.

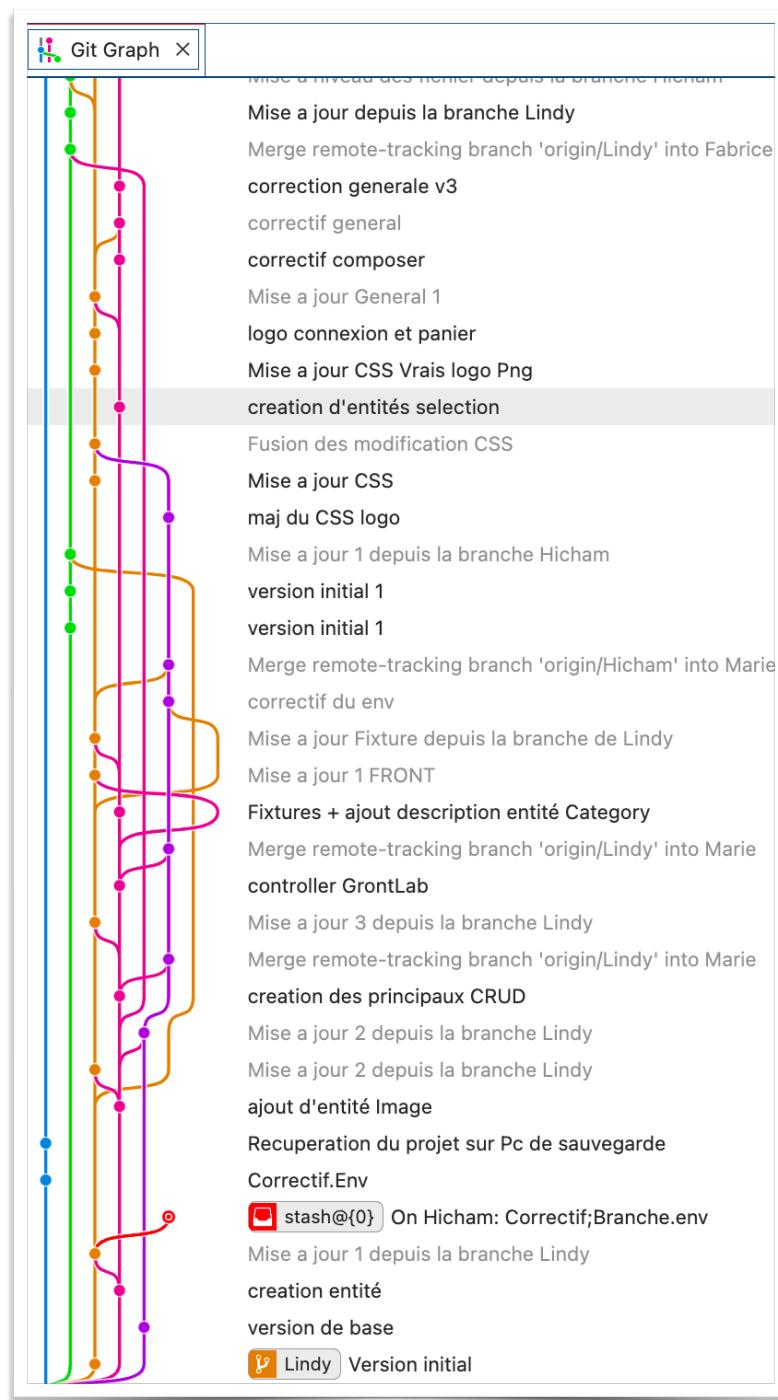
The screenshot shows a Trello project board titled "Projet Gront Lab". The board is organized into several sections:

- Work days**: Contains cards for tasks like "Doit être fait (Vital) TECHNIQUE [FRONT]Créer les vues que demande le cahier des charges" and "Doit être fait (Vital) TECHNIQUE [Front]Rendre Responsive design le site".
- Contenu du prochain compte rendu**: Contains a card for "Deadline" and a note about the next meeting date: "Date de la prochaine visite à la boutique ==> Jeudi 2".
- INFO**: Contains cards for "Doit être fait (Vital)", "INFO", and "FAQ".
- Planning**: Contains cards for "Doit être fait (Vital)", "INFO", and "FAQ".
- USER STORY**: Contains cards for "[US]", "Doit être fait (Vital)", and "INFO".
- TECHNIQUE**: Contains cards for "[TECH]", "Doit être fait (Vital)", and "INFO".
- FOCUS**: Contains cards for "Doit être fait (Vital)", "INFO", and "FAQ".
- FAQ**: Contains cards for "VERIFICATION", "TECHNIQUE", and "FAQ".
- TERMINÉ**: Contains cards for "Doit être fait (Vital)", "FOCUS", "TECHNIQUE", and "FAQ".
- Projets en cours**: Contains cards for "GRONT LAB" and "+ Ajouter une carte".

The background of the board features a scenic image of a snowy landscape with a green aurora borealis over a body of water.

GitHub

GitHub est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git. Voici ci-dessous Git Graph qui permet de visualiser les branches et les fusions entre elles ce qui permet de revenir en arrière en cas de problème.



Figma

Figma est un éditeur de graphiques et un outil de prototypage. Il est principalement basé sur le web. L'ensemble des fonctionnalités est axé sur l'utilisation dans la conception de l'interface utilisateur et de l'expérience utilisateur, en mettant l'accent sur la collaboration en temps réel

Ecran Accueil

Bienvenue dans ton CBD Shop, user.

Connexion / inscription

Favoris

MON PANIER

ACCUEIL

HUILES

FLEURS

THÉS

RESINES

PENS

ISOLATS

COSMÉTIQUES

A PROPOS

Contact

Bienvenue dans ton CBD Shop, user.

Connexion / inscription

Favoris

MON PANIER

ACCUEIL

HUILES

FLEURS

THÉS

RESINES

PENS

ISOLATS

COSMÉTIQUES

A PROPOS

CONTACTEZ VOTRE CBD SHOP

NOM PRENOM

ADRESSE EMAIL

VOTRE MESSAGE

Grontlab ©

Nos produits

Bienvenue dans ton CBD Shop, user.

Connexion / inscription

Favoris

MON PANIER

ACCUEIL

HUILES

FLEURS

THÉS

RESINES

PENS

ISOLATS

COSMÉTIQUES

A PROPOS

INFO SUR LE PRODUIT

INFO SUR LE PRODUIT

INFO SUR LE PRODUIT

INFO SUR LE PRODUIT

Grontlab ©

Espace Client

Bienvenue dans ton CBD Shop, user.

Connexion / inscription

Favoris

MON PANIER

ACCUEIL

HUILES

FLEURS

THÉS

RESINES

PENS

ISOLATS

COSMÉTIQUES

A PROPOS

VOS INFORMATIONS:

NOM PRENOM ADRESSE POSTALE

VOS COMMANDES:

commande n°1233
commande n°6789
commande n°3456

Grontlab ©

Notre catalogue

Bienvenue dans ton CBD Shop, user.

Connexion / inscription

Favoris

MON PANIER

ACCUEIL

HUILES

FLEURS

THÉS

RESINES

PENS

ISOLATS

COSMÉTIQUES

A PROPOS

NOS FLEURS IMAGE

NOS FLEURS IMAGE

NOS FLEURS IMAGE

NOS FLEURS IMAGE

Grontlab ©

Panier

Bienvenue dans ton CBD Shop, user.

Connexion / inscription

Favoris

MON PANIER

ACCUEIL

HUILES

FLEURS

THÉS

RESINES

PENS

ISOLATS

COSMÉTIQUES

A PROPOS

ADRESSE EMAIL

NOM

PRENOM

ADRESSE D'EXPÉDITION

RECAPITULATIF DES ARTICLES MIS DANS LE PANIER DU CLIENT

OG KUSH x 1 8.50€

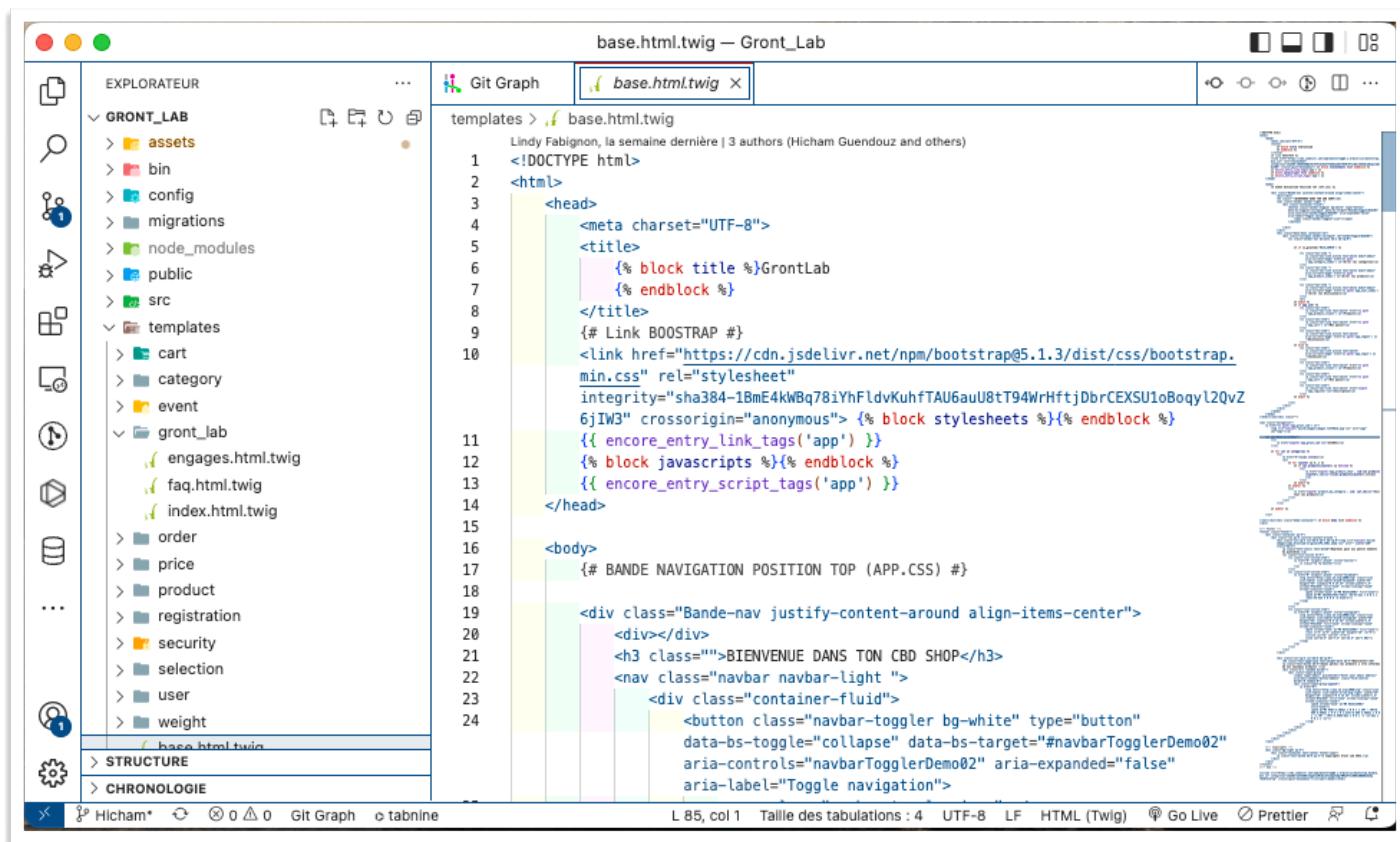
TOTAL: 8.50€

bouton continuer

Grontlab ©

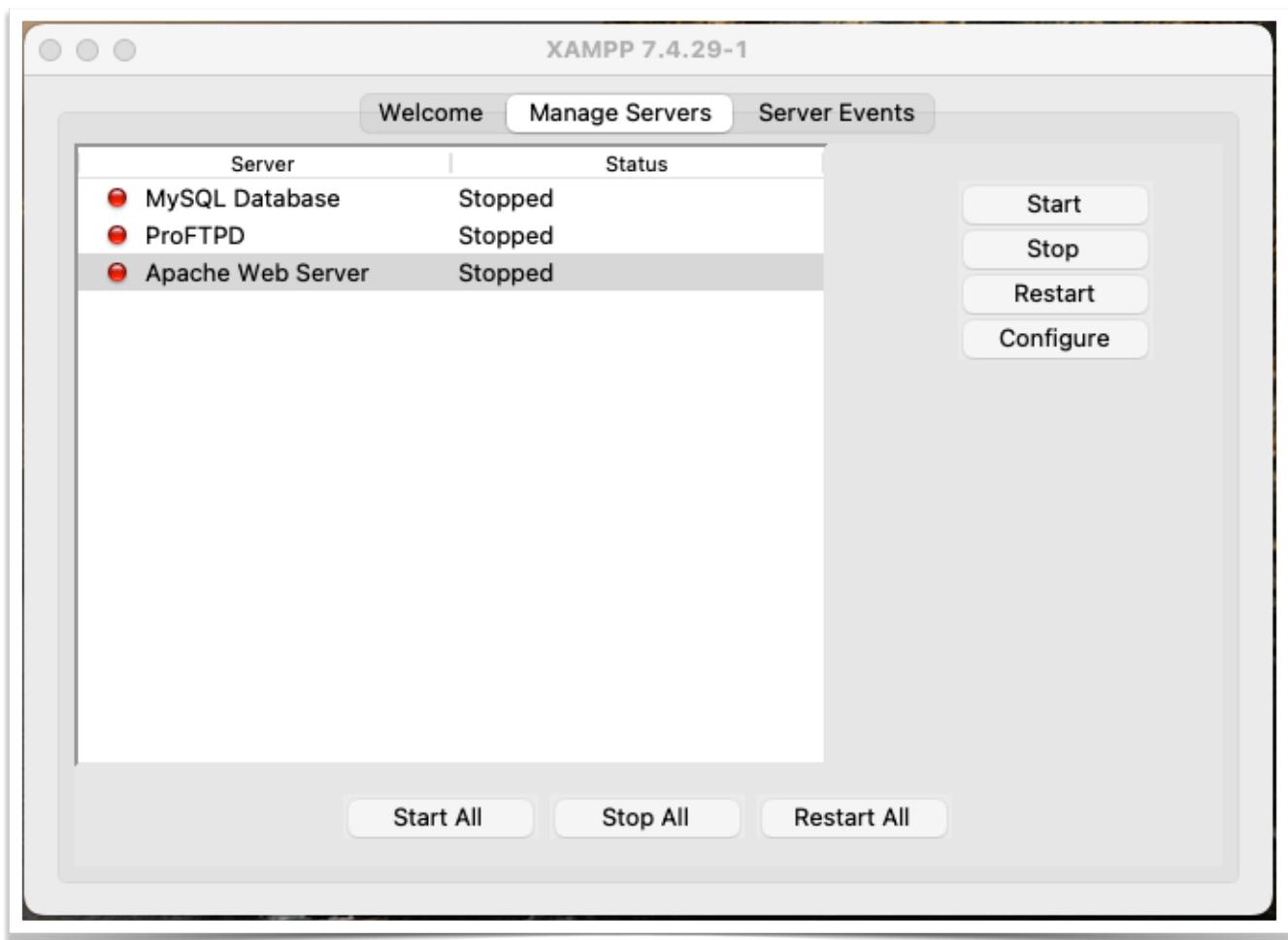
Visual Studio Code

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et MacOS.



Xampp

XAMPP permet de configurer un serveur de test local avant la mise en œuvre d'un site Web.



Répartition des tâches

Nous avons reparti les tâches de la façon suivante :

FRONT-END :

Hicham GUENDOUZ

Marie CISSOKHO

BACK-END :

Lindy FABIGNON

Fabrice FERREIRA

Pour ce qui est de la gestion des outils :

FIGMA :

Marie CISSOKHO

Lindy FABIGNON

TRELLO :

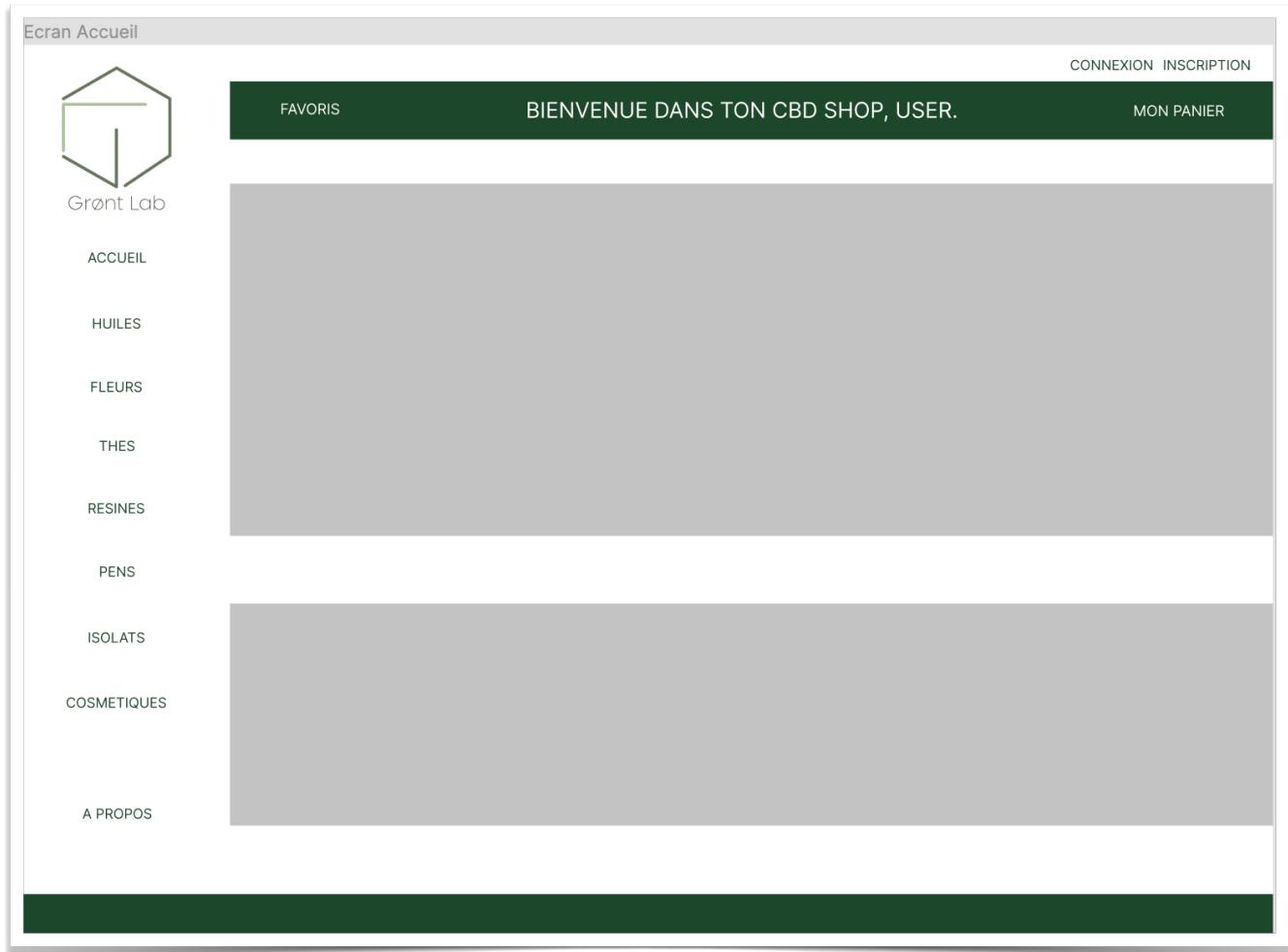
Fabrice FERREIRA

GITHUB :

Hicham GUENDOUZ

Conception de maquette

De l'élaboration du cahier des charges ressort les éléments nécessaires et incontournables à la conception des différentes maquettes.





ACCUEIL

HUILES

FLEURS

THES

RESINES

PENS

ISOLATS

COSMETIQUES

FAVORIS

BIENVENUE DANS TON CBD SHOP, USER.

MON PANIER

VOS INFORMATIONS:

**NOM
PRENOM
ADRESSE POSTALE**

VOS COMMANDES:

**commande n°1233
commande n°6789
commande n°3456**

A PROPOS

Grontlab ©



ACCUEIL

HUILES

FLEURS

THES

RESINES

PENS

ISOLATS

COSMETIQUES

FAVORIS

BIENVENUE DANS TON CBD SHOP, USER.

CONNEXION INSCRIPTION

MON PANIER

ADRESSE EMAIL**NOM****PRENOM****ADRESSE
D'EXPEDITION****bouton continuer****RECAPITULATIF DES ARTICLES MIS
DANS LE PANIER DU CLIENT**

**OG KUSH x 1
8.50€**

TOTAL: 8.50€

A PROPOS

Grontlab ©

Nos produits

CONNEXION INSCRIPTION

MON PANIER

FAVORIS BIENVENUE DANS TON CBD SHOP, USER.



Grønt Lab

ACCUEIL

HUILES

FLEURS

THES

RESINES

PENS

ISOLATS

COSMETIQUES

A PROPOS



INFO SUR LE PRODUIT



INFO SUR LE PRODUIT



INFO SUR LE PRODUIT



INFO SUR LE PRODUIT

Grontlab ©

Notre catalogue

CONNEXION INSCRIPTION

MON PANIER

FAVORIS BIENVENUE DANS TON CBD SHOP, USER.



Grønt Lab

ACCUEIL

HUILES

FLEURS

THES

RESINES

PENS

ISOLATS

COSMETIQUES

A PROPOS

NOS FLEURS

IMAGE

NOS FLEURS

IMAGE

NOS FLEURS

IMAGE

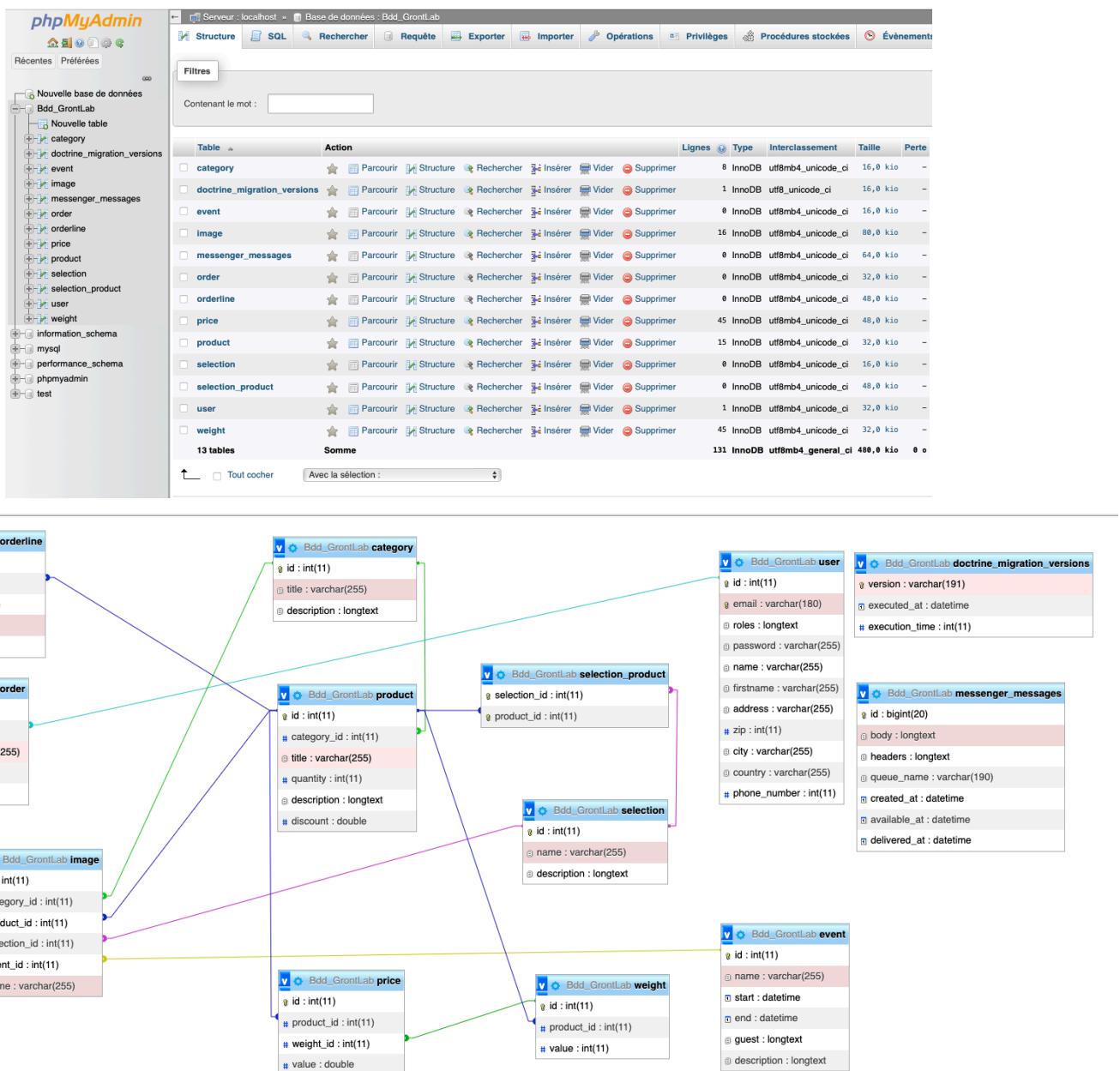
NOS FLEURS

IMAGE

Grontlab ©

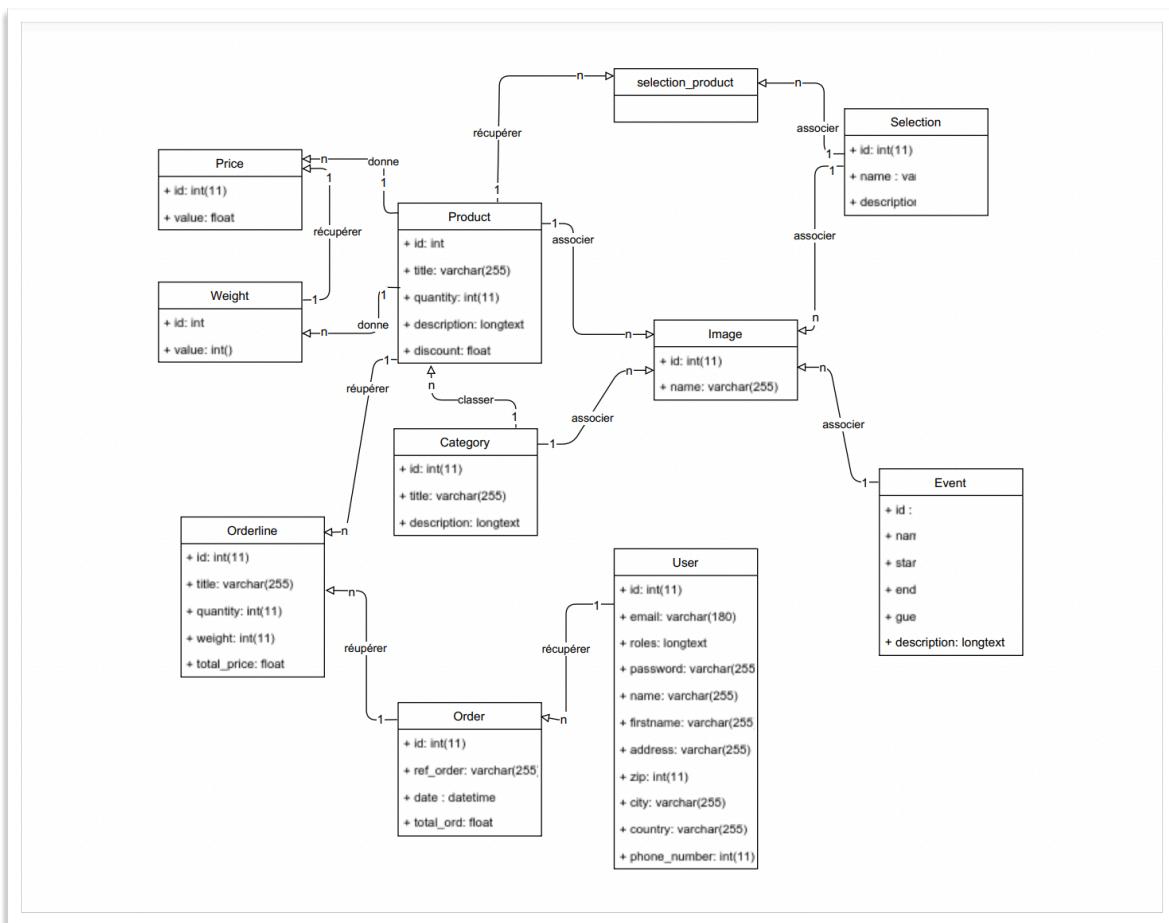
Base de Données

Capture D'écran des tables de la base de données (Avec les relations mises en valeurs)



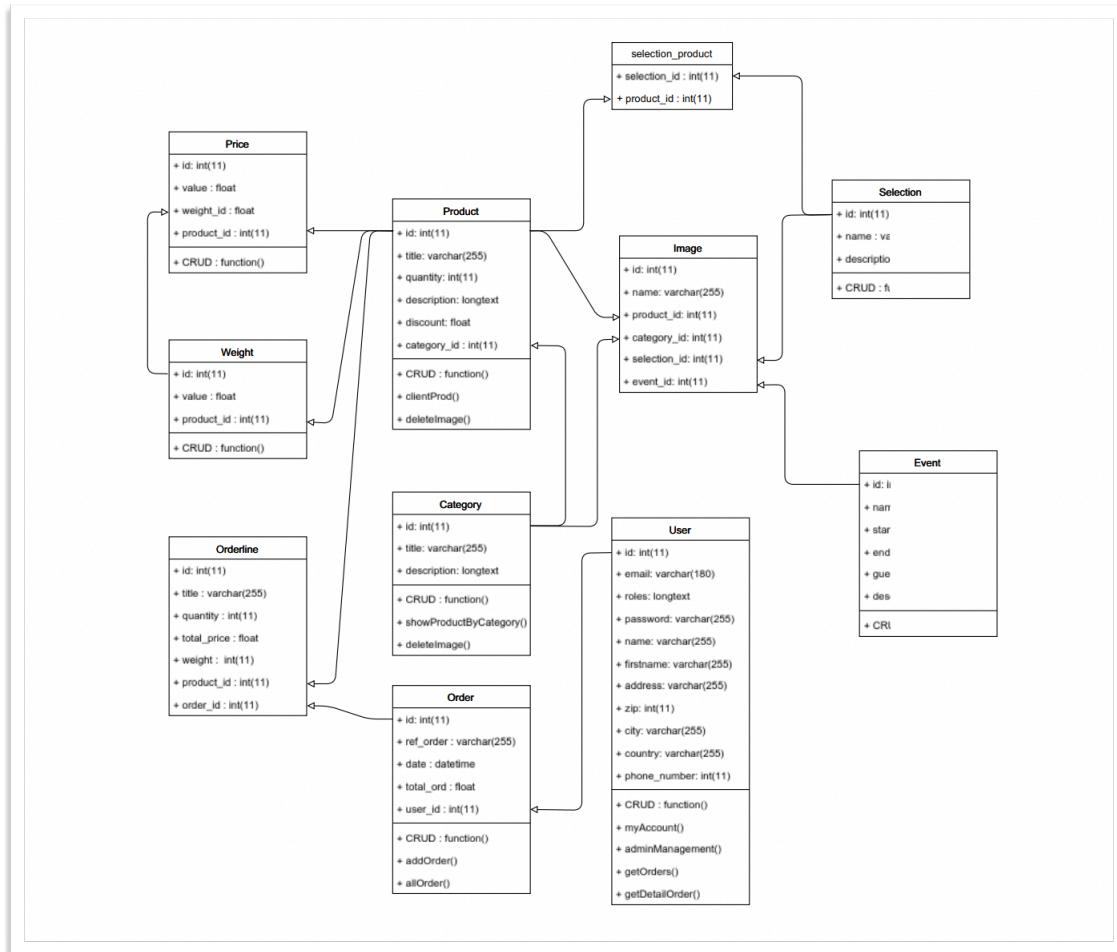
Model Conceptuel de Données

Un modèle conceptuel de données (MCD) pour une base de données permet d'identifier les principales entités à représenter, leurs relations et leurs attributs, et d'analyser la structure conceptuelle du système d'information. Contrairement à un modèle logique de données (MLD) ou à un modèle physique de données (MPD), un MCD est plus abstrait



Modèle Physique de Données

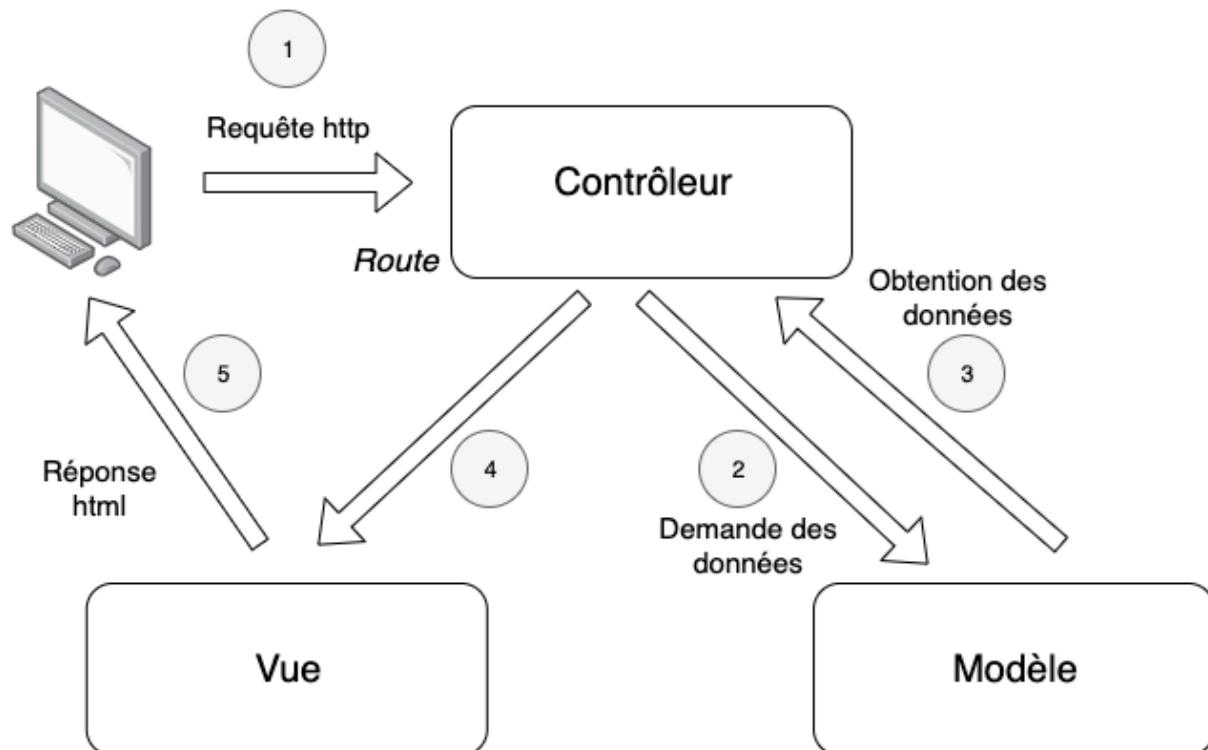
Un modèle physique de données introduit le contexte spécifique à la base de données qui manque dans les modèles conceptuels et logiques de données. Il représente les tables, colonnes, types de données, vues, contraintes, indices et procédures au sein de la base de données et/ou les informations communiquées lors des processus informatiques.



MVC

L'architecture MVC (Model-View-Controller), est l'une des architectures logicielles les plus utilisées pour les applications Web. Elle permet de créer une application web pour bien gérer la structuration d'un projet en trois parties.

Elle se compose de trois modules : modèle, vue, contrôleur



Les trois parties du MVC sont interconnectées. La vue affiche le modèle pour l'utilisateur. Le contrôleur accepte les entrées de l'utilisateur et met à jour le modèle et la vue.

Contrôleur

- composant responsable des prises de décision, gère la logique du code qui prend des décisions, il est l'intermédiaire entre le modèle et la vue.

```
PHP GrontLabController.php ×
src > Controller > PHP GrontLabController.php > ...
11  class GrontLabController extends AbstractController
12  {
13      /**
14      * @Route("/", name="app_gront_lab")
15      */
16     public function index(CategoryRepository $catRepo): Response
17     {
18         return $this->render('gront_lab/index.html.twig', [
19             'controller_name' => 'GrontLabController',
20             'categories' => $catRepo->findAll(),
21         ]);
22     }
23 }
24
25
26 /**
27 * @Route("/faq", name="app_faq")
28 */
29 public function faq(CategoryRepository $catRepo): Response
30 {
31     return $this->render('gront_lab/faq.html.twig', [
32         'controller_name' => 'GrontLabController',
33         'categories' => $catRepo->findAll(),
34     ]);
35 }
36 /**
37 * @Route("/engages", name="app_engages")
38 */
39 public function engages(CategoryRepository $catRepo): Response
40 {
41     return $this->render('gront_lab/engages.html.twig', [
42         'controller_name' => 'GrontLabController',
43         'categories' => $catRepo->findAll(),
44     ]);
45 }
46
47
48 }
```

Extraits de code les plus significatifs

Le projet a été créé avec le framework Symfony Aujourd'hui, Symfony est un Framework qui a plus de 10 ans. Depuis sa première version, il utilise un modèle de conception web classique connu sous le nom d'architecture MVC.

Voici une succession de capture commenté afin d'expliquer different fonctionnalité du site.

A. Back-End :

Ci-dessous se trouve les captures d'écran pour la gestion d'image

Product/_form.html.twig

```
9  {# Si le produit a une ou des images #}
10 {# pour chaque image du produit #}
11 {# On ajoute un lien permettant de supprimer une image #}
12 {# On ajoute un lien permettant de supprimer une image #}
13 {# On ajoute un lien permettant de supprimer une image #}
14 {# On ajoute un lien permettant de supprimer une image #}
15 {# On ajoute un lien permettant de supprimer une image #}
16 {# On ajoute un lien permettant de supprimer une image #}
17 {# On ajoute un lien permettant de supprimer une image #}
18 {# On ajoute un lien permettant de supprimer une image #}
19 {# On ajoute un lien permettant de supprimer une image #}
```

- Création d'un lien qui sera récupéré dans editForm.js pour le traiter

EditForm.js

```
20 // Gestion des boutons "Supprimer"
21 let links = document.querySelectorAll("[data-delete]")
22 // On boucle sur links
23 for(let link of links){
24     // On écoute le clic
25     link.addEventListener("click", function(e){
26         // On empêche la navigation
27         e.preventDefault()
28         // On demande confirmation
29         if(confirm("Voulez-vous supprimer cette image ?")){
30             // On envoie une requête Ajax vers le href du lien avec la méthode DELETE
31             fetch(this.getAttribute("href"), {
32                 method: "DELETE",
33                 headers: {
34                     "X-Requested-With": "XMLHttpRequest",
35                     "Content-Type": "application/json"
36                 },
37                 body: JSON.stringify({_token: this.dataset.token})
38             }).then(
39                 // On récupère la réponse en JSON
40                 response => response.json()
41             ).then(data => {
42                 if(data.success)
43                     // On supprime la div dans laquelle était stocké le lien
44                     this.parentElement.remove()
45                 else
46                     alert(data.error)
47             }).catch(e => alert(e))
48         }
49     })
50 }
51 }
52 }
```

- Après avoir récupéré la réponse en JSON qui sera transmise au ProductController on supprime la div dans laquelle était stockée le lien.

```
6 parameters:
7   images_directory: '%kernel.project_dir%/public/uploads'
```

routes/services.yaml

- On crée une nouvelle Image dans laquelle pour chaque image envoyée on stock le nom créé de l'image
Puis on crée la relation entre le produit et le nom de l'image stockée.

ProductController new() & edit()

À l'envoi du formulaire:

```
src > Controller > ProductController.php > ...
98 /**
99  * @Route("/{id}/edit", name="app_product_edit", methods={"GET", "POST"})
100 */
101 public function edit(Request $request, Product $product, ProductRepository $productRepository, CategoryRepository $catRepo): Response
102 {
103     $form = $this->createForm(ProductType::class, $product);
104     $form->handleRequest($request);
105
106     if ($form->isSubmitted() && $form->isValid()) {
107         // On récupère les images transmises
108         $images= $form->get('images')->getData();
109
110         // On boucle sur les images transmises
111         foreach ($images as $image) {
112             // générer nouveau nom de fichier
113             $file=md5(uniqid()). '.' . $image->guessExtension();
114
115             // On copie le fichier dans le dossier upload
116             $image->move(
117                 $this->getParameter('images_directory'), $file
118             );
119
120             // On stock le nom de l'image dans la bdd
121             $img = new Image();
122             $img->setName($file);
123             $product->addImage($img);
124         }
125         $productRepository->add($product);
126         return $this->redirectToRoute('app_product_index', [], Response::HTTP_SEE_OTHER);
127     }
128
129     return $this->renderForm('product/edit.html.twig', [
130         'product' => $product,
131         'form' => $form,
132         'price' => $product->getPrices(),
133         'categories' => $catRepo->findAll(),
134     ]);
135 }
```

On récupère les images transmises dans le formulaire grâce au paramètre 'images-directory' ajouté dans le fichier routes/services.yaml indiquant le chemin vers le dossier uploads créé manuellement.

Ci-dessous se trouve les captures d'écran pour l'affichage des prix / poids

product/clientProd.html.twig

```
9      {# Initialiser le compteur de produit pour recuperation du prix/poids dans getPrice.js #}
10     {% set selectId = 0 %}
```

- Initialiser le compteur qui servira d'identifiant au `<select>` d'un produit.
Il est incrémenter pour chaque produit à afficher.

```
141    {# Si le produit a un poids #}
142    {% if product.weights is not empty %}
143        <li>
144            <strong>Poids:</strong>
145            <select class="weights" id={{selectId}}>
146                {% set counter = 0 %}
147                {% for weight in product.weights %}
148                    <option value="{{counter}}>{{weight.value}}</option>
149                    {% set counter = counter + 1 %}
150                {% endfor %}
151            </select>
152        </li>
153    {% endif %}
```

- Si le produit a un poids

Créer un `<select>` qui affiche chacun des poids du produit avec le même identifiant / compteur que le `<select>` des prix.

```

49          <!--      the Price -->
50      <span class="price">
51          {# Si le produit a un poids #}
52          {% if product.weights is not empty %}
53              {# Parcourir les prix + initialiser le compteur de valeur#}
54              <select class="prices" id={{selectId}} disabled>
55                  {% set counter = 0 %}
56                  {% for price in product.prices %}
57                      <option value="{{counter}}">{{price.value}}</option>
58                      {% set counter = counter + 1 %}
59                  {% endfor %}
60              <select>
61          {% else %}
62              {# Si le produit n'a pas de poids #}
63              {% for price in product.prices %}
64                  <select class="prices" id={{selectId}} disabled>
65                      <option value=""{{price.value}}</option>
66                  <select>
67                  {% endfor %}
68  {% endif %}

```

- Si le produit a plusieurs poids

On stock chaque élément Prix du produit dans un `<select>`.

Le `<select>` va servir à afficher la valeur de l'élément poids sélectionné sans faire une recherche dans toute la table *Price*

- Si il n'y a pas de poids on affiche tout simplement le prix

captures d'écran pour l'affichage des prix / poids coté Javascript

getPrice.js

```
public > js > js getPrice.js > ...
Lindy Fabignon, il y a 1 heure | 1 author (Lindy Fabignon)
1  document.addEventListener("DOMContentLoaded", function () {      Lindy Fabignon,
2    // Selectionner toutes les <div> de PRICE et WEIGHT
3    let priceElement = document.querySelectorAll(".prices");
4    let selectElement = document.querySelectorAll(".weights");
5
6    // Initialiser le compteur on click pour modifier ou ajouter un innerHTML
7    let counter = 0;
```

- Récupération de chaque <select> Prix et Poids de la page ***Product/ClientProd.html***
- Crédation d'un compteur car l'action à effectuer diffère selon le nombre de click déjà fait sur le <select> Poids.

```

9   for (let selectElem of selectElement) {
10    selectElem.addEventListener("click", function () {
11      // Quand une nouvelle <option> est sélectionnée
12      selectElem.addEventListener("change", function () {
13        // Recuperer l'id du select WEIGHT
14        let selectId = selectElem.getAttribute("id");
15        // Parcourir les <div> select PRICE
16
17        for (let priceElem of priceElement) {
18          // Recuperer l'id du select PRICE
19          let selectedPriceId = priceElem.getAttribute("id");

```

- Pour chaque élément de la div contenant les <select> Poids écouter si click
- Ecouter si changement de valeur:
On parcours le <select> contenant les prix.
On récupère l'id du <select> Weight
- Pour chaque élément de la <div> contenant les Prix:
On récupère l'id du <select> Price

```

20       // Si l'id du select PRICE == id du select ELEM
21       if (selectedPriceId == selectId) {
22         // Parcourir chaque element <select> PRICE
23         for (var i = 0; i < priceElem.options.length; i++) {
24           // Recuperer l'id du weight sélectionné
25           let weightId = selectElem.selectedIndex;
26           // SI l'<option value> de PRICE = <option value> WEIGHT sélectionné
27           if (priceElem.options[i].value == weightId) {

```

- Si l'id du <select> Price est le même que celui du Prix alors
On récupère l'id <option value> du poids

```

27   if (priceElem.options[i].value == weightId) {
28     // Si premier click
29     if (counter == 0){
30       // Ajouter la valeur <selected> a <select> PRICE
31       priceElem.innerHTML += "<option selected=" + priceElem.options[i].value + ">" + priceElem.options[i].
32       innerHTML + "</option>";
33       let priceVal = priceElem.options[i].innerHTML;
34       let weightVal = weightId + 1;
35       // modifier le <a href > ajout panier
36       getRoute(weightVal, priceVal, selectId);
37       // Incrementer le compteur de click
38       counter++;
39     }

```

- Si c'est la première fois que l'utilisateur clic sur le bouton
On insère une ligne dans le `<select>` Prix qui sera l'`<option selected>` Price affiché correspondant au Poids sélectionné.
- Appel de la fonction `getRoute()` permettant de modifier le lien d'ajout au panier du produit avec le poids sélectionné par l'utilisateur.

```

40   // Sinon (si second click (counter>0))
41   else {
42     // Ajouter la valeur <selected> a <select> PRICE
43     priceElem.options[priceElem.selectedIndex].innerHTML = priceElem.options[i].innerHTML;
44     let priceVal = priceElem.options[i].innerHTML;
45     let weightVal = weightId + 1;
46     // modifier le lien vers ajout panier
47     getRoute(weightVal, priceVal, selectId)
48   }
49

```

- Si l'utilisateur change de poids a nouveau
On récupère le dernier prix affiché, donc l'élément sélectionné dans le `<select>` Price. Et on modifie son contenu html
- Récupérer la valeur du prix sélectionné = `<option value>`
Récupérer la valeur du `<select value>` poids (+1 pcq y'a un doublon pas encore expliqué)
- Appel de la fonction `getRoute()`

getPrice.js getRoute()

```
70  function getRoute(weightId, priceVal, selectId){  
71      //Recuperer le ou les liens <a href> vers panier  
72      let linkContainer = document.querySelectorAll(".link_container");  
73  
74      // Si il y a un ou des liens  
75      if(linkContainer != null){  
76          // Parcourir les liens  
77          for (let link of linkContainer) {  
78              // Recuperer la route  
79              let route = link.getAttribute('href');  
80              // init regExp pour recuperer les bonnes valeurs  
81              const regExpP = /([0-9]|[aA-zZ])+$/;  
82              const regExpW = /(?=product.weight[0].value).....|([0-9]{1,})(!.....)/gm;  
83  
84              if (link.getAttribute("id") == selectId) {  
85                  let newRoute = route.replace(regExpW, weightId).replace(regExpP, priceVal);  
86                  link.setAttribute("href", newRoute);  
87              }  
88          }  
89      }  
90  }  
91 }
```

- Pour chaque lien d'ajout au panier récupérer la route par défaut
- En regEx on récupère l'élément *Price* et *Weight* dans le href du lien
- Si l'id du lien correspond à l'id de l'élément weight sélectionné (les compteur id Price, Weight et link ont été initialisé en même donc chaque élément n est lié a l'élément n de l'autre):
on modifie la route en replaçant les regEx par la valeur du Poids et Prix

A. Front-End :

Mise en place de la librairie Bootstrap pour donner forme aux boutons, ainsi qu'aux différents éléments sur le site.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <title>
6              {% block title %}GrontLab
7                  {% endblock %}
8          </title>
9          {% Link BOOTSTRAP %}
10         <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrfhtjDbrCEXSU1oBoqyl20vZ6jIW3" crossorigin="anonymous"> {% block stylesheets %}{% endblock %}
11         {{ encore_entry_link_tags('app') }}
12         {% block javascripts %}{% endblock %}
13         {{ encore_entry_script_tags('app') }}
14     </head>    racine du dossier, le mois dernier + Mise à jour 1 FRONT ..
```

Lien Bootstrap (CSS)

```
177     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka75k0Gln4gmtz2MlQnikT1wXgYs0g+OMhuP+IlRH9sENB00Lrn5q+8nbTov4+p" crossorigin="anonymous"></script></body></html>
```

Lien Bootstrap (Javascript)

Code html de la barre de navigation

```
<ul id="menu-accordeon"> Hicham Guendouz, le mois dernier + Mise à jour CSS FAQ/Engagés ...
<li>
    <a href="{{path('app_gront_lab')}}">ACCUEIL</a>
</li>

{&gt; for cat in categories %}
    <li>
        <a href="#">{{cat.title}}</a>
        <ul>
            {&gt; for counter in 0..2 %}
                {&gt; if cat.products[counter] is defined %}
                    <li>
                        <a href="{{path('app_product_show', {id:cat.products[counter].id})}}>{{cat.products[counter].title}}</a>
                    </li>
                {&gt; endif %}
            {&gt; endfor %}
        </ul>
        <a href="{{path('product_by_category', {id: cat.id})}}>Voir tous les produits</a>
    </li>
{&gt; endfor %}

</ul>
```

Code CSS de la barre de navigation

```
19 /* Menu accordeon barre de nav principal */
20 #menu-accordeon {
21     list-style: none;
22     text-align: center;
23     width: 185px;
24     position: relative;
25 }
26 #menu-accordeon ul {
27     padding: 0;
28     margin: 0;
29     list-style: none;
30     text-align: center;
31 }
32 #menu-accordeon li {
33     background-color: whites;
34     border-radius: 6px;
35     margin-bottom: 2px;
36 }
37 #menu-accordeon li li {
38     max-height: 0;
39     overflow: hidden;
40     transition: all 0.5s;
41     border-radius: 0;
42     background: □white;
43     box-shadow: none;
44     border: none;
45     margin: 0;
46 }
47 #menu-accordeon a {
48     display: block;
49     text-decoration: none;
50     color: ■#20492d;
51     padding: 8px 0;
52     font-family: verdana;
53     font-size: 1.2em;
54 }
55 #menu-accordeon ul li a,
56 #menu-accordeon li:hover li a {
57     font-size: 0.8em;
58 }
59 #menu-accordeon li:hover {
60     background: □white;
61 }
62 #menu-accordeon li li:hover {
63     background: ■#999;
64 }
65 #menu-accordeon ul li:last-child {
66     border: none;
67 }
68 #menu-accordeon li:hover li {
69     max-height: 15em;
70 }
```

Carousel dans l'index réalisé en bootstrap

```
14 <div id="carouselExampleDark" class="carousel carousel-dark slide" data-bs-ride="carousel">
15   <div class="carousel-indicators">
16     <button type="button" data-bs-target="#carouselExampleDark" data-bs-slide-to="0" class="active" aria-current="true"
17       aria-label="Slide 1"></button>
18     <button type="button" data-bs-target="#carouselExampleDark" data-bs-slide-to="1" aria-label="Slide 2"></button>
19     <button type="button" data-bs-target="#carouselExampleDark" data-bs-slide-to="2" aria-label="Slide 3"></button>
20   </div>
21   <div class="carousel-inner">
22     <div class="carousel-item active" data-bs-interval="10000">
23       <a href="#">
24         
25       </a>
26     </div>
27     <div class="carousel-item" data-bs-interval="20000">
28       <a href="#">
29         
30       </a>
31     </div>
32     <div class="carousel-item">
33       <a href="#">
34         </a>
35     </div>
36     <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleDark" data-bs-slide="prev">
37       <span class="carousel-control-prev-icon" aria-hidden="true"></span>
38       <span class="visually-hidden">Previous</span>
39     </button>
40     <button class="carousel-control-next" type="button" data-bs-target="#carouselExampleDark" data-bs-slide="next">
41       <span class="carousel-control-next-icon" aria-hidden="true"></span>
42       <span class="visually-hidden">Next</span>
43     </button>
44   </div>
</div>
```

Ci-dessous ce trouve le CSS du carousel

```
assets > styles > accueil.css > ...
Hicham Guendouz, le mois dernier | 1 author (Hicham Guendouz)
1  /*Carousel modification*/ Hicham Guendouz,
2  .carousel {
3    display: flex;
4    justify-content: center;
5    align-items: center;
6    width: 50%;
7    margin-left: 25%;
8  }
9  .carousel-indicators {
10  position: absolute;
11  bottom: 0;
12  margin: 0;
13  left: 0;
14  right: 0;
15  width: auto;
16  }
17  .carousel-indicators li,
18  .carousel-indicators li.active {
19    float: left;
20    width: 33%;
21    height: 10px;
22    margin: 0;
23    border-radius: 0;
24    border: 0;
25    background: #ccc;
26  }
27  .carousel-indicators li.active {
28    background: orange;
29  }
```

Conclusion

Le site web e-commerce répond aujourd’hui aux attentes fixées lors de la réunion de projet : un site personnalisé et fonctionnel.

L’administrateur peut créer des catégories, enregistrer des produits, procéder aux modifications d’usage ou à leurs suppressions, les agrémenter de photos, assigner des droits à ses collaborateurs.

L’utilisateur a accès à l’ensemble des catégories des produits proposés par la boutique, aux produits, aux différents conditionnements, passer sa commande via un panier.

La mise en production du site nécessitera d’ajouter des fonctionnalités supplémentaires (commenter un produit, un évènement, ajouter en favori, pages administrateur et utilisateur , API de paiement, etc....). Cette partie sera mise en œuvre à la fin de la formation suite à la demande du commanditaire.

D’un point de vue personnel, ce projet a permis de développer de nombreuses compétences dont l’assimilation des langages abordés en formation, une amélioration dans la recherche d’informations & le travail en équipe.

CAHIER DES CHARGES Gront Lab

1. Présentation d'ensemble du projet

1.1 Présentation de l'entreprise :

- « Gront lab » est une entreprise de vente de produits à base de CBD.

1.2 Les objectifs du site :

- Le site de « Gront Lab » sera la seule plateforme de vente de l'entreprise, et doit être en mesure de proposer une expérience de qualité aux visiteurs du site. Le back-office du site doit également permettre une gestion quotidienne des activités e-commerce (suivi des commandes, mise à jour du catalogue produit).

1.3 La cible adressée par le site :

- « Gront Lab » cible les particuliers. Le discours marketing, les modes de tarification pratiqués ou l'image de marque qui sera développée ciblera principalement une clientèle relativement variée (18+) utilisant les supports digitaux de manière intensive.

1.4 Objectifs quantitatifs après 1 an :

- Nombre de produits vendus : ?
- Visiteurs par jour : 50 à 100
- Taux de conversion : 5% minimum
- Nombre de comptes clients : 150+

1.5 Périmètre du projet :

- L'activité vise uniquement la France métropolitaine.
- Le site sera disponible en français uniquement.
- Le site sera intégralement "Responsive Design".
 - L'ensemble des fonctionnalités détaillées dans ce document seront accessibles depuis un mobile.

2. Description graphique et ergonomique

2.1 Charte graphique :

La charte graphique doit être proposée par le prestataire, en respectant les consignes suivantes.

- La charte graphique doit être moderne et épurée. Utiliser les inspirations ci-dessous
- La couleur dominante du site sera le blanc : #ffffff
- La couleur secondaire, utilisée pour les titres, les boutons, et autres éléments de navigation sera le vert pistache : #d1e7dd
- Autre couleur vert foncée : #20492D

Vert clair : #AFC5A1

Vert taupe : #667860

2.2 Inspirations

Inspirations graphiques

-

-

2.3 Logo



Gront Lab

« Gront Lab » se réserve le droit de fournir directement le logo à utiliser.

3. Description fonctionnelle et technique

3.1 Arborescence du site :

Le site se décompose en 6 pages principales, toutes accessibles depuis le menu du site, lui-même présent à l'identique sur l'ensemble des pages du site. Ces pages principales sont :

- Home (<http://GrontLab>)
- Nos produits (<http://GrontLab/>)
- Notre catalogue (<http://GrontLab.fr/vitrine>)
- Panier (<http://GrontLab/panier>)
- Gestion de commande (Http://GrontLab/gestion_commande)
- Espace Client (<http://GrontLab/profil>)

En plus de ces pages principales, les pages suivantes doivent être accessibles depuis le FOOTER :

- Contact (<http://GrontLab/feedback>)

3.2 Description fonctionnelle du site :

Boutique en ligne n°1 : Notre collection

Le site de « Gront Lab » doit bien évidemment comporter une boutique en ligne, accessible depuis l'onglet "Notre Catalogue". Cette boutique en ligne doit par défaut afficher l'ensemble des produits mis en ligne depuis le back-office, et disponibles en stocks.

Depuis la liste de chaque produit, les utilisateurs doivent pouvoir :

- Ajouter le produit au panier
- Consulter la fiche du produit

Inspiration : https://boutique.deli-hemp.fr/fleurs-transformees/10413-11324-crystal-amnesia-sativa-l.html#/7-grammage-1_gramme

tout les sites se ressemblent à ce niveau-là.

Fiches produits

Chaque fiche produit est composée des informations structurées suivantes :

- Nom du produit
- Photo du produit
- Description du produit
- Prix du produit
- Sélection du poids désiré pour visualiser le prix

L'utilisateur pourra depuis ces pages produits effectuer un ajout panier.

Espace Client

Un utilisateur sera dans l'obligation de créer un compte client pour compléter un achat. Afin de créer un compte, l'utilisateur devra seulement fournir son adresse email et définir un mot de passe.

La création d'un compte est indispensable afin de pouvoir finaliser une commande.

3.3 Description fonctionnelle du Back-office :

Depuis le back-office, l'équipe de « Gront Lab » doit être en mesure d'effectuer en autonomie l'ensemble des tâches quotidiennes nécessaires au bon fonctionnement du site e-commerce.

Fonctionnalités standards

- Publication de nouveaux produits : Mise en ligne de nouveaux produits en toute autonomie. Possibilité d'utiliser du HTML aussi bien qu'un éditeur simplifié.

Fonctionnalités E-Commerce

1. Gestion du front

- Gestion de catégorie de produits : Ajout et retrait des produits affichés dans chacune des collections.
- Gestion des promotions et codes promos : Création et suppression de codes promo et offres temporaires sur des produits ou groupes de produits.
- Mise à jour de la page d'accueil : Possibilité de mettre en avant des produits, promotions et articles

2. Gestion du Back-office

- Suivi des commandes : Affichage et modification de l'état (en cours de livraison, etc.)
- Gestion des stocks : Manuelle dans un premier temps. Il doit être possible de suivre depuis le Back-office le niveau des stocks pour chacun des produits.

3.4 Informations relatives aux contenus :

- Type de contenus : Le site utilisera des types de contenus différents : images, ressources téléchargeables, qu'il doit être possible de télécharger facilement sur le site.

3.5 Contraintes techniques :

- Le site doit être compatible avec l'ensemble des navigateurs standards : Google Chrome, Android, Mozilla Firefox, Internet Explorer, Safari & Opéra. Les versions compatibles doivent être définies explicitement par le prestataire).
- Le développement du site doit en tenir compte et rendre des intégrations futures facilement réalisables.

4. Prestations attendues et modalités de sélection des prestataires

4.1 Prestations attendues :

- Design : Réalisation de maquettes pour chacune des pages principales listées dans l'arborescence du site
- Intégration : Intégration de ces maquettes après validation
- Développement du site et du back-office
- Création et paramétrage de la base de données du site

4.2 Planning :

Le projet sera découpé en 3 phases :

1. Phase de prototypage : Construction des maquettes, validation des choix d'architectures et de technologies.
2. Phase de développement du pilote : Développement de l'ensemble des fonctionnalités, intégration des maquettes
3. Phase d'industrialisation : Intégration du catalogue de produits et des contenus.

L'objectif est donc un passage en production (livraison finale) le 30/05/2022.

4.3 Méthodologie de suivi :

L'ensemble du projet sera basé sur une méthodologie agile type "scrum".

Attribution des rôles

Product Owner : Un membre de l'équipe de « Gront Lab ». Il sera en charge de définir les priorités et de prendre les décisions lors de potentiels arbitrages.

Scrum Master : Un membre de l'équipe du prestataire qui sera en charge du respect de la méthodologie et des relations avec le Product Owner.

Organisation du projet :

- Construction du back log. Pour chacune des 3 phases, un Product back log découplant l'ensemble des fonctionnalités et éléments à réaliser sera définit.

- Sprints : Le projet sera organisé sur la forme de sprints hebdomadaires. Ces sprints se décomposeront toujours de la manière suivante :

- Jour 0 – Le Product owner et l'équipe du prestataire fixent l'objectif général du sprint à partir du Product Backlog.

- Jour 1 – L'équipe définit la liste des tâches à réaliser au cours du sprint. Chaque tâche est évaluée en temps afin d'ajuster le nombre de tâche à réaliser en fonction de la capacité de développement disponible sur le sprint.

- Jour 3 – Le Product Owner et le Scrum Master retravaillent le Product Backlog et définissent l'objectif principal du prochain sprint.

- Jour 4 – Démo – l'équipe du prestataire présente au Product Owner les résultats du travail réalisé au cours du dernier sprint. La démo se fera sur un environnement de pré-production

- Jour 5 – L'équipe définit la liste des tâches du prochain sprint (Sprint Backlog).

Dans le projet <>Gront Lab>> c'est Romain au titre de Président Général le Product Owner !

Suivi du projet

L'ensemble du product Backlog et des sprints seront organisés et suivis sur un board Trello, afin de garantir une bonne compréhension et lisibilité de l'avancement du projet par toutes les parties prenantes.

4.4 Propriété :

« Gront Lab » sera propriétaire de l'ensemble des images, graphismes, icônes et autres contenus créés pour le site. Le prestataire s'engage également à transmettre ses droits de propriété, d'exploitation, de reproduction, d'adaptation, de distribution et de traduction sur l'ensemble du site, ainsi que le code source et l'ensemble des accès à « Gront Lab » lors de la mise en production initiale.

Capture d'écran du site

Vue page accueil utilisateur

BIENVENUE DANS TON CBD SHOP



Grønt Lab

ACCUEIL

ab
cumque
ea
cum
libero
saepe
pariatur
nihil

< >

Grønt Lab

A propos

BIEN ETRE

Le cannabidiol est un composant vertueux du cannabis, il est une substance utilisée pour soulager différents troubles ou maladies.

FAQ

Comment doser ? Vers quels produits me diriger ? Besoin d'un conseil ?

ENGAGÉ

GRONT LAB s'engage à toujours privilégier la qualité et la traçabilité.

Connexion
Produits
Mon panier
Inscription

Vue page produit utilisateur

BIENVENUE DANS TON CBD SHOP



Grønt Lab

ACCUEIL

quam
et
deleniti
similique
sint
quas
delectus
a

amet Nouveau

• Titre amet
• Quantités: 100
• Poids: 1
• Promotion: 5

Detail du produit
Ajouter au panier

veritatis Nouveau

Detail du produit
Ajouter au panier

Produits

Grønt Lab

NEWSLETTER

Soyez parmis les premiers à être informez de nos nouveaux produits !

Enter your email address

Reprenez gout aux petits moments du quotidien.

Vue page produit Administrateur

Bienvenue dans ton CBD Shop

Produits

Identifiant	Titre	Prix	Quantités	Description	Poids	Promotion	Images	actions
16	ipsum	10	100	Optio labore aliquid quae deserunt consequatur ab. Occaecati voluptas exercitationem sunt molestiae sit distinctio harum. Est impedit aut sapiente.	1	5		<button>Aperçu</button> <button>Modifier</button>
17	labore	10	100	Eum nobis aut sint non aut aliquam. Ut sed sit aut fugiat. Omnis ex facilis aut assumenda magni voluptate molestiae qui.	1	5		<button>Aperçu</button> <button>Modifier</button>
18	doloribus	10	100	Doloribus amet accusamus beatae maiores deserunt officia. Ut aspernatur aut ratione eaque veritatis repellendus doloremque non. Autem corrupti non ut et quod. Minus voluptatem vel nostrum reprehenderit iusto est autem aut.	1	5		<button>Aperçu</button> <button>Modifier</button>
19	cupiditate	10	100	Ut est est ipsum nihil. Ut velit tempore natus. Ut quia quis libero et.	1	5		<button>Aperçu</button> <button>Modifier</button>
20	amet	10	100	Hic illum fuga facere qui et quae molestias. Voluptatem architecto voluptas ut optio ipsam repudiandae provident. Mollitia asperiores autem nisi distinctio est autem cumque.	1	5		<button>Aperçu</button>

Vue page détail produit administrateur

Bienvenue dans ton CBD Shop

Produits

Identifiant	18
Titre	doloribus
Prix	10
Quantités	100
Description	Doloribus amet accusamus beatae maiores deserunt officia. Ut aspernatur aut ratione eaque veritatis repellendus doloremque non. Autem corrupti non ut et quod. Minus voluptatem vel nostrum reprehenderit iusto est autem aut.
Poids	1
Promotion	5
Images	

[Gérer les catégories](#)
[Gérer les produits](#)
[Espace de gestion](#)
[Gérer les Utilisateurs](#)

[Produits](#)
[Mon panier](#)
[Espace Client](#)
[Déconnexion](#)

[Retour à la liste des produits](#) [Modifier](#) [Supprimer](#)

Nouveau

doloribus

