

CREATION DU SITE DE E-COMMERCE DE VENTE DE PC

PAGE DE SERVICE

Référence : N/A

Niveau de confidentialité : corporate

Mises à jour

Version	Date	Auteur	Description du changement
1.1	22/05/2023	NIETO Wylan	Plan et ressources
1.2	24/05/2023	NIETO Wylan	Développement
1.3	22/06/2023	NIETO Wylan	Livraison du site

Validation

Version	Date	Nom	Rôle
01	23/06/2023	NIETO Wylan	Maître d'oeuvre
		NIETO Luis	Maître d'ouvrage

SOMMAIRE

CREATION DU SITE DE E-COMMERCE DE VENTE DE PC.....	0
PAGE DE SERVICE.....	0
SOMMAIRE	0
1. INTRODUCTION :	1
2. OBJECTIF DE LA MISSION :	2
3. ANALYSE CONCEPTION :	3
3.1 INTRODUCTION	3
3.2 ANALYSE DES BESOINS	3
3.3 CONCEPTION DE LA BASE DE DONNEES	4
3.4 CONCEPTION DE L'ARCHITECTURE DJANGO	4
3.5 CONCEPTION DES INTERFACES UTILISATEUR	4
3.6 SECURITE	5
3.7 DEPLOIEMENT	5
4. TECHNOLOGIES UTILISEES :	5

Wylan

5.	VALORISATION DE L'IMAGE DE L'ORGANISATION :	6
3.8	ÉVOLUTION DU SITE WEB EXPLOITANT LES DONNÉES DE L'ORGANISATION :	13
3.9	PROCHAINES ÉTAPES :	13
6.	STRUCTURE	14
6.1	APPLICATION "STORE"	14
6.1.1 FICHIERS :	14
6.2	APPLICATION "ACCOUNTS"	14
6.2.1 FICHIERS :	14
6.3	APPLICATION "CONFIGOMATIC"	14
6.3.1 FICHIERS :	14
7.	CONCLUSION :	15
8.	TABLE DES ILLUSTRATIONS.....	15

1. Introduction :

L'objectif de ce compte-rendu est de faire le point sur les actions entreprises pour développer la présence en ligne de la société de Monsieur Luis NIETO. La société de Monsieur Luis NIETO est orientée vers la production de jeux vidéo mais il envisage également de configurer des PC et de les vendre sur internet. Cette opération pouvait être réalisée occasionnellement sur leboncoin.fr. Afin d'avoir une image plus professionnelle, il a souhaité développer un site de e-commerce.

Ce document détaillera les activités réalisées, étape par étape et les résultats obtenus.

2. Objectif de la mission :

Nous allons vous présenter le projet de création d'un site de commerce électronique axé sur la vente d'ordinateurs. En utilisant Django et Tailwind CSS, nous avons développé une plateforme conviviale et esthétiquement attrayante pour faciliter l'achat d'ordinateurs.

Dans notre concept, nous avons défini un site permettant de présenter des machines de façon détaillée. Chaque poste présenté est classé dans une gamme d'ordinateurs, distinguant les portables et les ordinateurs de bureau. Notre objectif est de fournir un site fluide et intuitif capable d'intéresser un large public.

En planifiant l'architecture du site, nous avons créé des pages clés telles que la page d'accueil, les pages de catégories de produits, les pages de produits individuels, le panier d'achat et les pages de paiement. Nous avons veillé à ce que l'interface utilisateur soit cohérente, réactive et adaptée à différentes tailles d'écran, en utilisant Tailwind CSS pour la conception.

En intégrant Django, nous avons établi une structure backend solide, permettant de gérer les utilisateurs, les authentifications, les sessions et les modèles de données liés aux ordinateurs et aux autres articles informatiques, aux commandes et aux paiements. Nous avons également développé des vues et des URL appropriées pour les différentes fonctionnalités du site, garantissant un processus d'achat sans accroc, avec la possibilité d'ajouter des produits au panier et de passer des commandes.

Nous avons mis en place des fonctionnalités essentielles du commerce électronique, telles que la recherche de produits, la filtration par catégorie ou par spécifications, l'ajout au panier, la gestion des commandes et le suivi des livraisons. Nous nous sommes assurés de mettre en place un système de paiement sécurisé.

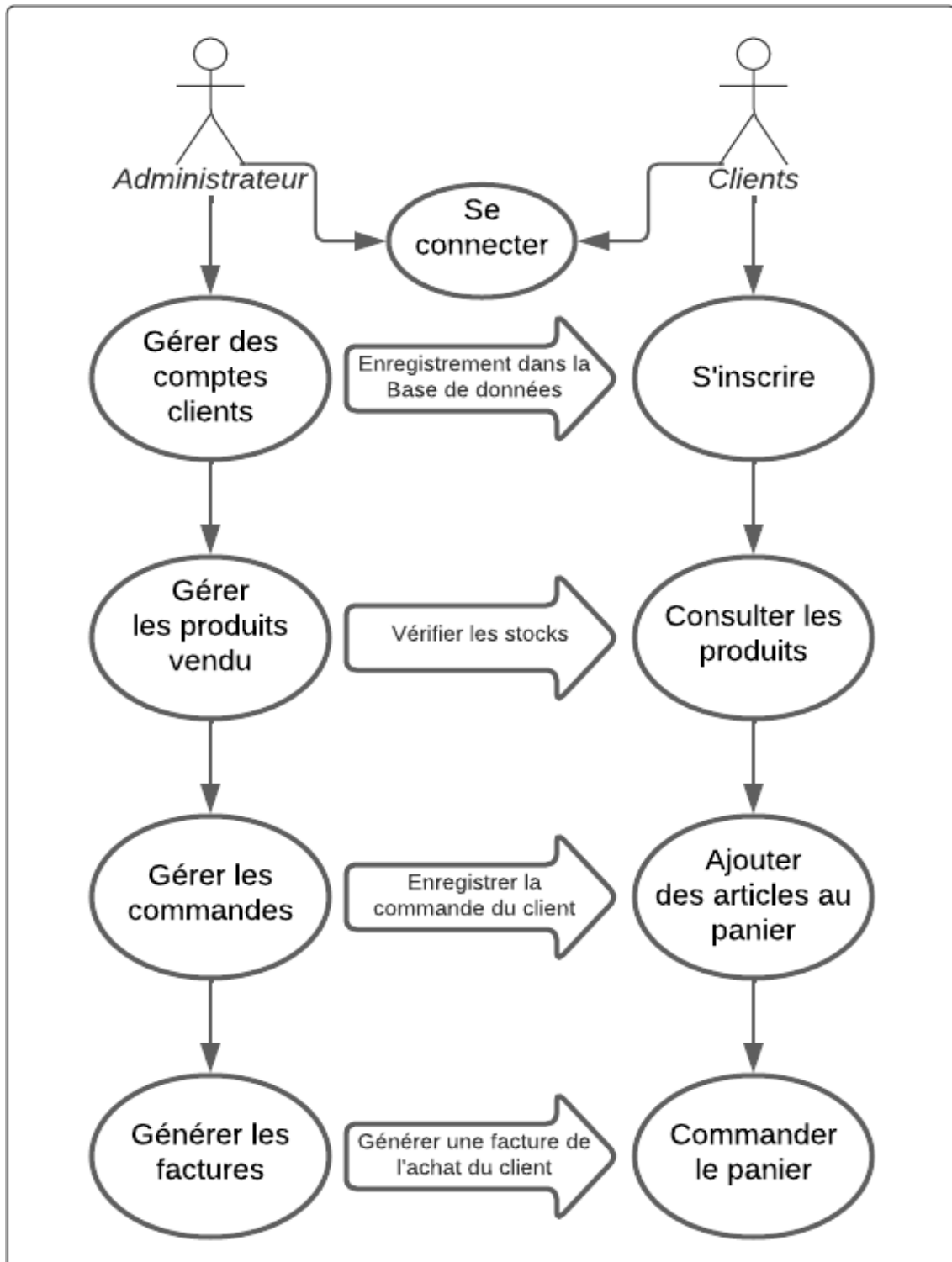
La performance et la sécurité sont des aspects clés de notre site. Nous avons effectué des tests approfondis pour optimiser les performances, identifier les éventuels problèmes et garantir la rapidité du site. Nous avons mis en œuvre des pratiques de sécurité, comme l'authentification à deux facteurs et le chiffrement des données sensibles, afin de protéger les utilisateurs et leurs informations.

3. Analyse conception :

3.1 INTRODUCTION

- Présentation du projet : un site de e-commerce spécialisé dans la vente de PC et de composants.
- Technologies utilisées : Django (un framework web Python) et Tailwind CSS (un framework CSS).
- Objectif : développer une plateforme conviviale, sécurisée et réactive pour les internautes.

3.2 ANALYSE DES BESOINS



Wylan

3.3 CONCEPTION DE LA BASE DE DONNEES

Identification des entités principales :

- Utilisateur (avec rôles : client, administrateur).
- Produit (avec catégories, marques, etc.).
- Commande (avec produits, quantité, prix, etc.).
- Méthode de paiement et d'expédition.

Modélisation des relations entre les entités :

- Relation entre Utilisateur et Commande (un utilisateur peut passer plusieurs commandes, mais une commande est propre à un seul client).
- Relation entre Commande et Produit (une commande peut contenir plusieurs produits. Chaque produit vendu est unique donc il figure sur zéro ou une commande).
- Relation entre Commande et Paiement (une commande est payée en totalité lors de l'achat. On a fait le choix de distinguer le paiement de la commande, sachant qu'ils auraient pu être confondus au niveau analytique.)

3.4 CONCEPTION DE L'ARCHITECTURE DJANGO

- Structuration du projet Django en applications logiques :
 - User : gestion des utilisateurs et des rôles.
 - Product : gestion des produits, des catégories et des marques.
 - Cart : gestion du panier d'achat.
 - Order : gestion des commandes et des états.
 - Payment : gestion des méthodes de paiement.
 - Shipping : gestion des méthodes d'expédition.
- Définition des modèles Django pour chaque application, en se basant sur la conception de la base de données.

3.5 CONCEPTION DES INTERFACES UTILISATEUR

➤ Définition des différentes pages du site :

- Page d'accueil : présentation des produits populaires et des promotions.
- Pages de liste de produits : affichage des produits avec des options de filtrage.

Wylan

- Page de détails du produit : affichage des informations détaillées d'un produit avec la possibilité de l'ajouter au panier.
- Panier d'achat : récapitulatif des produits sélectionnés avec possibilité de passer commande.
- Pages de gestion des comptes (inscription, connexion, profil utilisateur).
- Pages d'administration : gestion des produits, des commandes et des utilisateurs.
- Utilisation du framework Tailwind CSS pour le design et la mise en forme des pages.

3.6 SECURITE

- Implémentation de mesures de sécurité pour protéger le site contre les attaques courantes (injections SQL, cross-site scripting, etc.).
- Utilisation de fonctionnalités de Django telles que l'authentification utilisateur, la protection CSRF, etc.

3.7 DEPLOIEMENT

Le déploiement reste à faire à ce jour (27/06/2023) Néanmoins une réflexion sérieuse a été menée sur le sujet. Elle aborde les points suivants :

- Choix d'une plateforme d'hébergement pour déployer le site.
- Configuration du serveur chez l'hébergeur.
- Vérifier les mécanismes de sauvegarde régulière des données par l'hébergeur ou envisager une solution programmée.

4. Technologies utilisées :

- HTML/CSS : Les langages de balisage et de définition de style utilisés pour structurer et présenter le contenu web.
- JavaScript : Un langage de programmation polyvalent utilisé pour ajouter de l'interactivité et des fonctionnalités dynamiques côté client.
- Python : Le langage de programmation principal utilisé avec Django, apprécié pour sa simplicité et sa richesse en bibliothèques.
- Django : Un framework web Python de haut niveau qui simplifie le développement web et offre des outils puissants pour créer des applications web sécurisées et évolutives.
- PyCharm : Un IDE puissant et intuitif proposant un environnement de travail convivial spécialisé pour le Python.
- Tailwind CSS : Un framework CSS qui accélère le développement frontend grâce à ses composants prêts à l'emploi et ses classes utilitaires.

5. Valorisation de l'image de l'organisation :

Nous avons réalisé d'importants progrès pour valoriser l'image de l'organisation sur les médias numériques. Pour cela, nous avons développé un site Web attrayant et convivial utilisant le framework Django et adoptant un design moderne basé sur Tailwind CSS. Les fichiers **models.py** et **views.py** ont été créés et adaptés pour gérer les modèles de données et les vues liées aux produits de l'organisation.

Le site Web présente une interface utilisateur intuitive et esthétiquement plaisante, mettant en valeur les produits et services de l'organisation. Il utilise des fonctionnalités de navigation claires et des éléments visuels engageants pour offrir une expérience utilisateur optimale.

Voici quelques captures d'écran du site Web mettant en évidence son design attractif et convivial :

Page d'accueil :

Sur cette page d'accueil nous avons un bouton qui nous redirige vers la page produits et des boutons qui nous redirigent vers les pages de détails des produits.

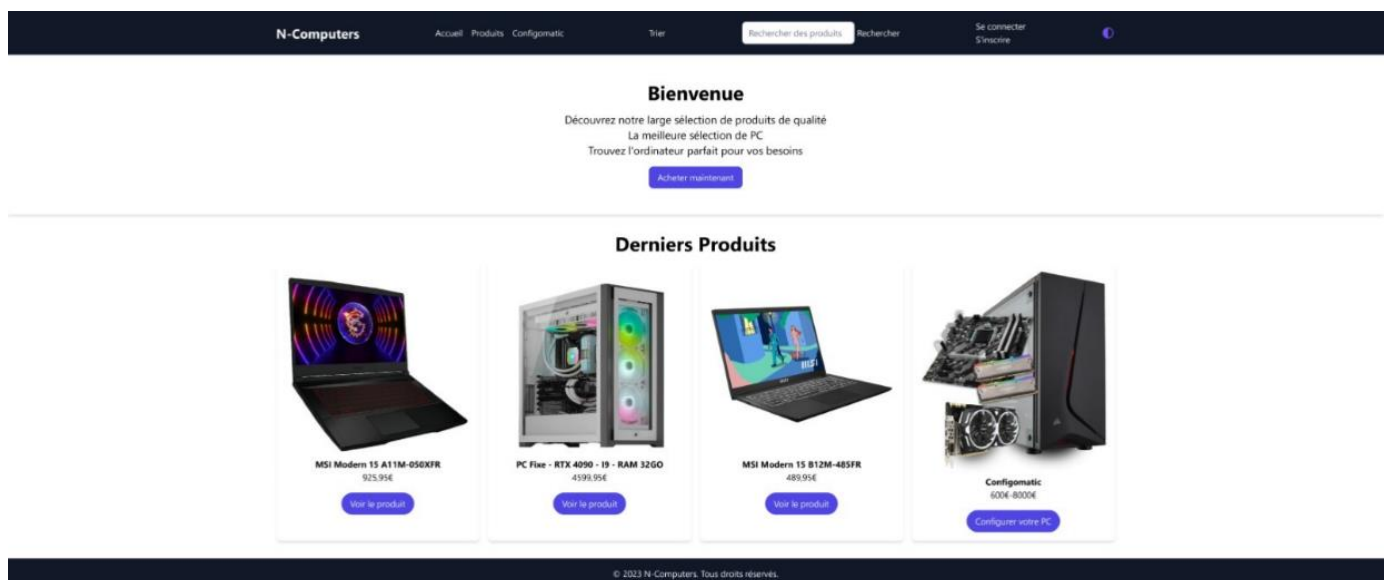


Figure 1 Page d'accueil

Code concernant cette page :

```
<section class="bg-gray-50 shadow-md py-12">
  <div class="container mx-auto text-center">
    <h1 class="text-4xl font-bold mb-4">Bienvenue</h1>
    <p class="text-xl">Découvrez notre large sélection de produits de qualité</p>
    <h1 class="text-xl">La meilleure sélection de PC</h1>
    <p class="text-xl">Trouvez l'ordinateur parfait pour vos besoins</p>
    <div class="button">
      <a class="inline-block bg-indigo-600 text-white rounded-lg px-4 py-2 mt-4 hover:bg-indigo-700"
        href="{% url 'produits' %}">Acheter maintenant</a>
    </div>
  </div>
</section>

<section class="container mx-auto text-center my-8">
  <h2 class="text-4xl font-bold mb-4">Derniers Produits</h2>
  <div class="flex justify-center">
    <div class="grid grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-4">
      {% for product in products %}
        <div class="bg-gray-50 rounded-lg shadow-md p-4">
          <a href="{% product.get_absolute_url %}" class="flex flex-col items-center">
```

Wylan

```

        
        <h2 class="font-bold">{{ product.name }}</h2>
        {% if product.rangePrice %}
        <h3>{{ product.price }}€ - {{ product.maxPrice }}€</h3>
        {% else %}
        <h3>{{ product.price }}€</h3>
        {% endif %}
        <a href="{{ product.get_absolute_url }}"
        class="bottom-0 left-0 inline-block bg-indigo-600 text-white rounded-full px-4 py-2
mt-4 hover:bg-indigo-700">Voir
        le produit</a>
    </a>
</div>
{% endfor %}

<div class="bg-gray-50 rounded-lg shadow-md p-4">
    <a href="{% url 'configomatic' %}" class="flex flex-col items-center">
        
        <h2 class="font-bold">Configomatic</h2>
        <p>600€-8000€</p>
        <a href="{% url 'configomatic' %}"
        class="bottom-0 left-0 inline-block bg-indigo-600 text-white rounded-full px-4 py-2
mt-4 hover:bg-indigo-700">Configurer
        votre PC</a>
    </a>
</div>
</div>
</div>
</section>

```


Page de Produits :

La page produits est une page qui permet aux utilisateurs de voir la totalité des produits acheteables sur le site. Des boutons peuvent rediriger l'internaute vers les détails des produits.

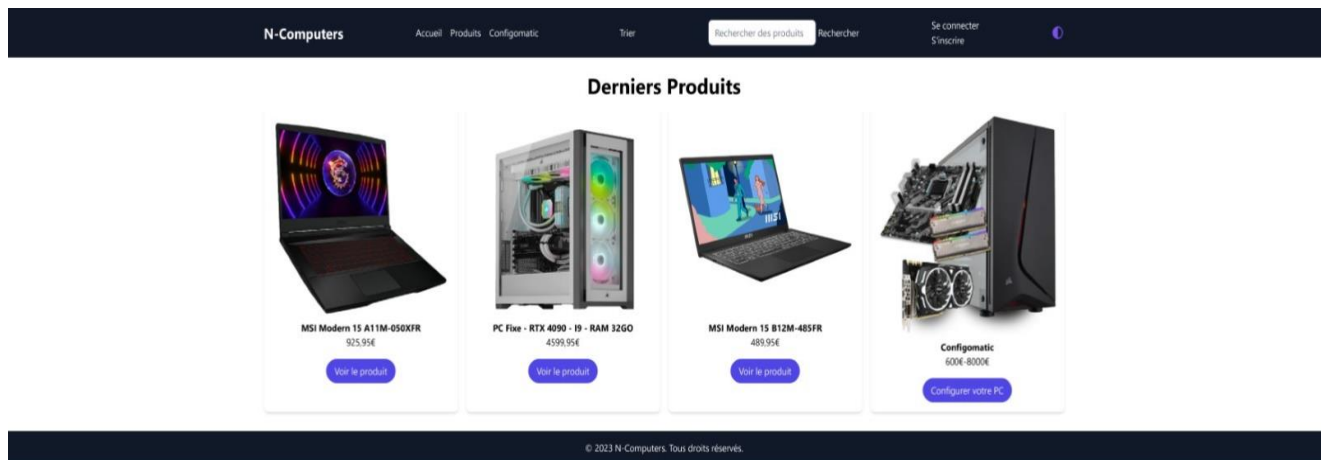


Figure 2 Page produits

Code concernant cette page :

Le fichier HTML et le CSS grâce aux classes TailwindCSS :

```
<h1 class="text-4xl font-bold mb-4">Derniers Produits</h1>
<div class="flex justify-center">
  <div class="grid grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-4">
    {% for product in products %}
      <div class="bg-white rounded-lg shadow-md p-4">
        <a href="{{ product.get_absolute_url }}" class="flex flex-col items-center">
          
          <h2 class="font-bold">{{ product.name }}</h2>
          {% if product.rangePrice %}
            <h3>{{ product.price }}€ - {{ product.maxPrice }}€</h3>
          {% else %}
            <h3>{{ product.price }}€</h3>
          {% endif %}
          <a href="{{ product.get_absolute_url }}"
            class="bottom-0 left-0 inline-block bg-indigo-600 text-white rounded-full
px-4 py-2 mt-4 hover:bg-indigo-700">Voir le produit</a>
        </a>
      </div>
    {% endfor %}

    <div class="bg-white rounded-lg shadow-md p-4">
      <a href="{% url 'configomatic' %}" class="flex flex-col items-center">
        
        <h2 class="font-bold">Configomatic</h2>
        <p>600€-8000€</p>
        <a href="{% url 'configomatic' %}"
          class="bottom-0 left-0 inline-block bg-indigo-600 text-white rounded-full
px-4 py-2 mt-4 hover:bg-indigo-700">Configurer votre PC</a>
      </a>
    </div>
  </div>
</div>
```

Wylan

Page de Détails de produits :

Cette page montre à quoi ressemble le produit, le prix et la description claire du produit.

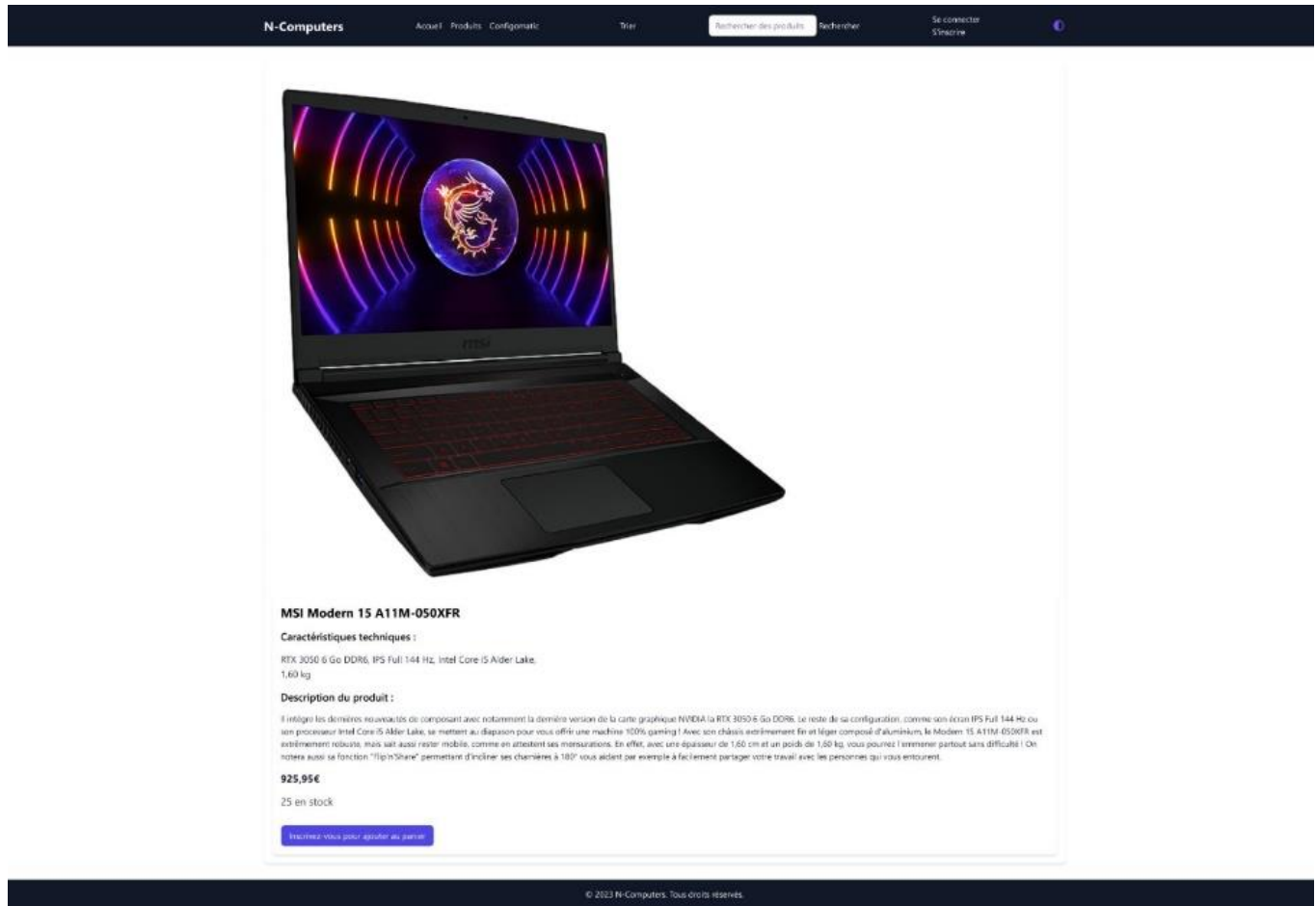


Figure 3 Page détails de produits

Code concernant cette page :

Le modèle concernant les produits :

```
class Product(models.Model):
    name = models.CharField(max_length=128)
    slug = models.SlugField(max_length=128)
    price = models.FloatField(default=0.0)
    rangePrice = models.BooleanField(default=False)
    maxPrice = models.FloatField(default=0.0)
    stock = models.IntegerField(default=0)
    description = models.TextField(blank=True)
    caracteristiques = models.TextField(blank=True)
    thumbnail = models.ImageField(upload_to="products", blank=True, null=True)

    def __str__(self):
        return f"{self.name} ({self.stock})"

    def get_absolute_url(self):
        return reverse("product", kwargs={"slug": self.slug})
```

Page de Configomatic (configuration de pc sur mesure) :

Dans cette page l'internaute peut configurer un PC sur-mesure sans faire attention à la compatibilité des composants.

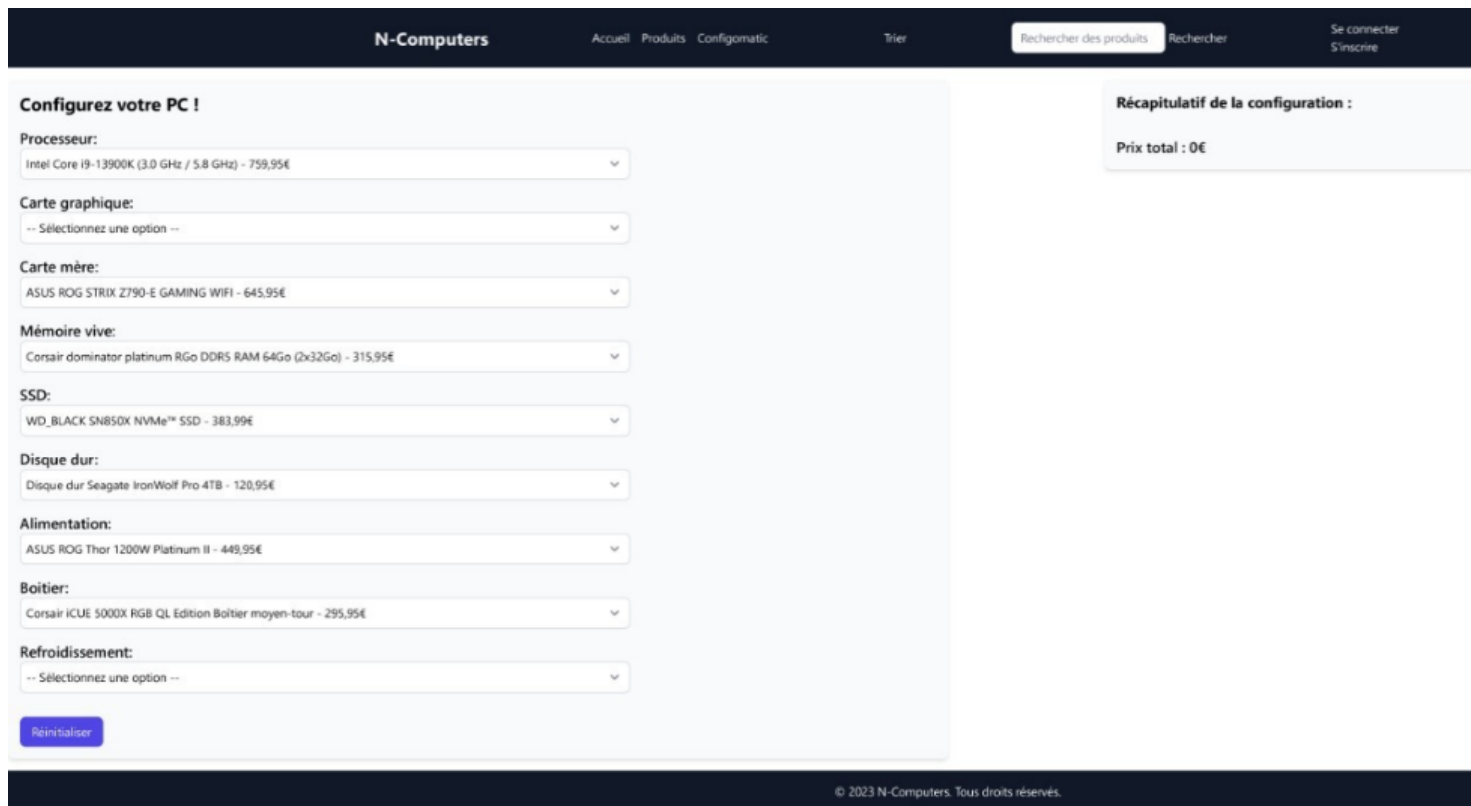


Figure 4 Page configomatic (configuration de pc sur-mesure)

Code concernant cette page(seulement la partie HTML/CSS du processeur) :

```
<div class="mb-4">
  <label for="processeur" class="font-semibold text-xl block">Processeur:</label>
  <select name="processeur" id="processeur" class="border border-gray-300 rounded-lg p-2 w-2/3">
    <option value="">-- Sélectionnez une option --</option>
    {% for processeur in processeurs %}
    <option value="{{ processeur.id }}" data-prix="{{ processeur.price }}"
      data-thumbnail="{{ processeur.thumbnail.url }}" data-socket="{{ processeur.socket }}">
      {{ processeur.name }} - {{ processeur.price }}€
    </option>
    {% endfor %}
  </select>
</div>
```

Page Trier (similaire à la page Rechercher) :

Dans cette page nous pouvons trier les différents produits par : Prix par ordre croissant, Prix par ordre décroissant, Nom de A-Z et Nom de Z-A.

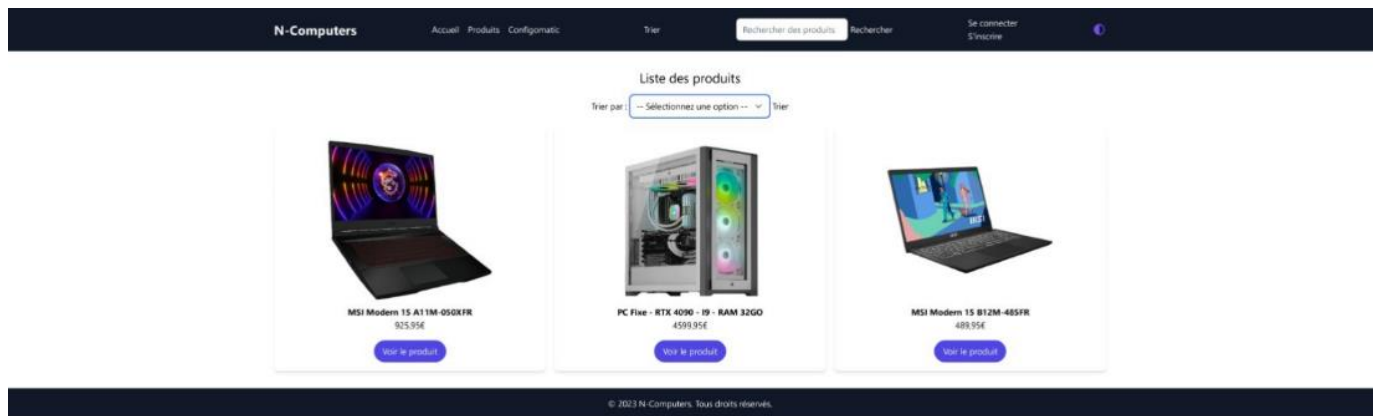


Figure 5 Page trier

Code concernant cette page :

```
<main class="container mx-auto text-center my-8">
  <h2 class="text-2xl mb-4">Liste des produits</h2>

  <form method="GET">
    <label for="sort_by">Trier par :</label>
    <select id="sort_by" name="sort_by" class="rounded-lg">
      <option value="">-- Sélectionnez une option --</option>
      <option value="name_asc">Nom (A-Z)</option>
      <option value="name_desc">Nom (Z-A)</option>
      <option value="price_asc">Prix croissant</option>
      <option value="price_desc">Prix décroissant</option>
    </select>
    <button type="submit">Trier</button>
  </form>

  <div class="grid grid-cols-1 md:grid-cols-3 gap-4 mt-4">
    {% for product in products %}
      <div class="bg-white rounded-lg shadow-md p-4">
        <a href="{{ product.get_absolute_url }}" class="flex flex-col items-center">
          <div class="w-2/3 h-2/3">
            
          </div>
          <h2 class="font-bold">{{ product.name }}</h2>
          <h3>{{ product.price }}€</h3>
          <a href="{{ product.get_absolute_url }}" class="inline-block bg-indigo-600 text-white rounded-
full px-4 py-2 mt-4 hover:bg-indigo-700">Voir le produit</a>
        </a>
      </div>
    {% endfor %}
  </div>
</main>
```

Wylan

Page pour Se connecter (similaire à la page S'inscrire) :

Cette page est simplement une page de connexion. Si l'internaute est nouveau sur le site, un bouton le redirige vers la page d'inscription.

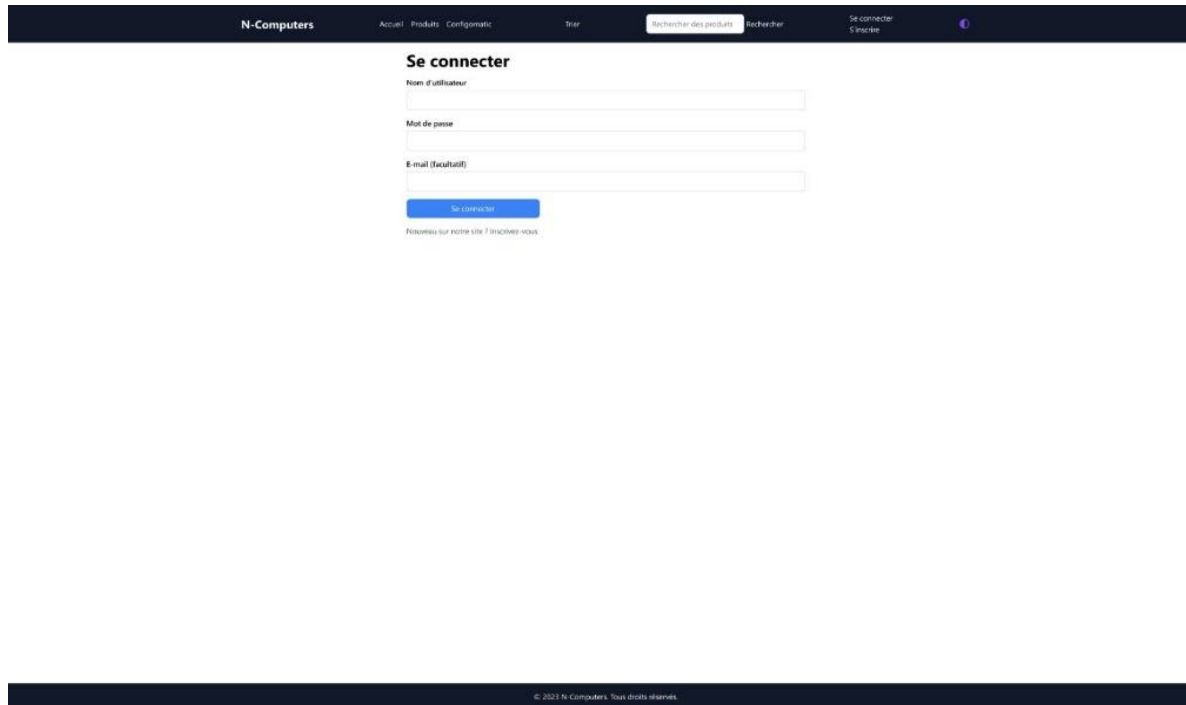


Figure 6 Page Se connecter

Code concernant cette page :

```
<div class="flex flex-col items-center justify-top mt-100 h-screen">
  <div class="w-1/3">
    <h1 class="text-4xl font-bold mt-4 mb-4">Se connecter</h1>
    <form method="POST">
      {% csrf_token %}
      <div class="mb-4">
        <label for="username" class="block font-medium">Nom d'utilisateur</label>
        <input id="username" type="text" name="username" required class="mt-1 px-4 py-2 border border-
gray-300 rounded-md w-full">
      </div>
      <div class="mb-4">
        <label for="password" class="block font-medium">Mot de passe</label>
        <input id="password" type="password" name="password" required class="mt-1 px-4 py-2 border border-
gray-300 rounded-md w-full">
      </div>
      <div class="mb-4">
        <label for="email" class="block font-medium">E-mail (facultatif)</label>
        <input id="email" type="email" name="email" class="mt-1 px-4 py-2 border border-gray-300 rounded-
md w-full">
      </div>
      <button type="submit" class="bottom-0 left-0 inline-block bg-indigo-600 text-white rounded-xl px-4
py-2 mt-4 hover:bg-indigo-700">Se connecter</button>
    </form>
    <p class="mt-4 text-gray-600">
      Nouveau sur notre site ?
      <a href="{% url 'signup' %}" class="nav-link nav-link text-white hover:text-gray-400 transition-
colors duration-200">
        Inscrivez-vous
      </a>
    </p>
  </div>
</div>
```

Wylan

3.8 ÉVOLUTION DU SITE WEB EXPLOITANT LES DONNEES DE L'ORGANISATION :

Le site Web a été continuellement amélioré pour exploiter au mieux les données de l'organisation et offrir une expérience utilisateur enrichie. Nous avons développé des modèles de données supplémentaires pour gérer des entités telles que les cartes mères, les mémoires vives, les SSD, les disques durs, les alimentations, les boîtiers et les refroidissements. Ces entités ont été intégrées dans les fichiers **models.py** existants.

Grâce aux fonctionnalités du framework Django, nous avons réalisé des tests d'intégration approfondis pour garantir le bon fonctionnement du site Web et la cohérence des données. Des tests d'acceptation ont également été effectués pour valider les fonctionnalités clés du site, notamment l'ajout de produits au panier, le passage de commandes et la navigation fluide entre les différentes pages.

Ce projet est séparé en différentes applications telles que : store, configomatic, accounts, theme et.djangotests.

Séparer le projet en différentes applications permet de mieux gérer ces données et d'être plus organisé

3.9 PROCHAINES ETAPES :

Bien que nous ayons accompli des progrès significatifs dans le développement de la présence en ligne de l'organisation, il reste encore des opportunités d'amélioration. Voici quelques-unes des prochaines étapes que nous envisageons :

- Élargir les efforts de marketing en ligne : Nous allons renforcer nos activités de marketing en ligne pour atteindre un public plus large. Cela comprendra des campagnes publicitaires ciblées, une optimisation du référencement, une présence active sur les réseaux sociaux et des partenariats stratégiques.
- Améliorer la conversion des ventes : Nous allons analyser les données d'analyse pour identifier les points faibles du processus de conversion des ventes et mettre en œuvre des améliorations pour augmenter le taux de conversion. Cela pourrait inclure des améliorations de l'ergonomie du site, des incitations à l'achat, des offres spéciales, etc.
- Optimiser l'expérience utilisateur : Nous allons continuer à optimiser l'expérience utilisateur en prenant en compte les commentaires des utilisateurs, en effectuant des tests d'utilisabilité et en mettant en place des fonctionnalités améliorées. Nous nous engageons à offrir une interface conviviale, une navigation fluide et des temps de chargement rapides.
- Maintenir une présence active sur les médias numériques : Nous allons rester engagés et actifs sur les médias numériques en partageant régulièrement du contenu pertinent, en interagissant avec les utilisateurs, en répondant aux commentaires et aux questions, et en surveillant les tendances du marché.

6. Structure

6.1 APPLICATION "STORE"

6.1.1 FICHIERS :

- `models.py` : Ce fichier définit les modèles de données utilisés dans l'application "store". Le modèle principal est `Product`, qui représente un produit avec des attributs tels que le nom, le prix, la description et l'image.
- `views.py` : Ce fichier contient les vues de l'application "store". Les vues implémentées incluent la page d'accueil (`index`), la liste des produits (`produits`), les détails d'un produit spécifique (`product_detail`), l'ajout d'un produit au panier (`add_to_cart`), l'affichage du panier (`cart`) et la suppression du panier (`delete_cart`).
- `urls.py` : Ce fichier définit les URL (chemins) correspondant aux vues de l'application "store". Les URL sont associées aux vues correspondantes.

6.2 APPLICATION "ACCOUNTS"

6.2.1 FICHIERS :

- `models.py` : Ce fichier définit les modèles de données utilisés dans l'application "accounts". Le modèle `User` fourni par Django est utilisé pour gérer les utilisateurs du système. Aucun modèle personnalisé n'a été ajouté.
- `views.py` : Ce fichier contient les vues de l'application "accounts". Les vues implémentées incluent l'inscription d'un utilisateur (`signup`), la connexion d'un utilisateur (`login_user`) et la déconnexion d'un utilisateur (`logout_user`).
- `urls.py` : Ce fichier définit les URL correspondant aux vues de l'application "accounts". Les URL sont associées aux vues correspondantes.

6.3 APPLICATION "CONFIGOMATIC"

6.3.1 FICHIERS :

- `models.py` : Ce fichier définit les modèles de données utilisés dans l'application "configomatic". Les modèles incluent `Processeur`, `CarteGraphique`, `CarteMere`, `MemoireVive`, `SSD`, `DisqueDur`, `Alimentation`, `Boitier` et `Refroidissement`. Chacun de ces modèles représente un composant d'une configuration d'ordinateur avec des attributs tels que le nom, le prix, la quantité en stock, la description et l'image.
- `views.py` : Ce fichier contient les vues de l'application "configomatic". La vue principale est `configomatic`, qui affiche les composants disponibles pour la configuration d'un ordinateur. Les vues supplémentaires incluent la filtration des composants en fonction de la carte mère sélectionnée.
- `urls.py` : Ce fichier définit les URL correspondant aux vues de l'application "configomatic". Les URL sont associées aux vues correspondantes.

Le projet utilise également le framework CSS Tailwind pour la mise en page et le style des pages HTML. Tailwind est intégré dans les fichiers HTML des templates, permettant ainsi de styliser les différents éléments de l'interface utilisateur.

Ce compte-rendu résume les fichiers et fonctionnalités principales du projet Django avec Tailwind CSS. Le projet comporte des applications distinctes pour gérer les produits, les comptes utilisateur et la configuration d'un ordinateur. Chaque application possède ses propres modèles, vues et URL. Tailwind CSS est utilisé pour la conception de l'interface utilisateur.

7. Conclusion :

La présence en ligne de l'organisation devrait considérablement évoluer grâce aux actions entreprises dans le cadre de ce projet. Nous devons encore publier le site, assurer un référencement afin d'être visible facilement dans les moteurs de recherche.

Le site Web est fonctionnel en local . Les prochaines étapes identifiées consiste à choisir un hébergeur, réserver un nom de domaine, publier le site et commencer à l'administrer

8. Table des illustrations

<i>Figure 1 Page d'accueil.....</i>	<i>6</i>
<i>Figure 2 Page produits.....</i>	<i>8</i>
<i>Figure 3 Page détails de produits</i>	<i>9</i>
<i>Figure 4 Page configomatic (configuration de pc sur-mesure.....</i>	<i>10</i>
<i>Figure 5 Page trier</i>	<i>11</i>
<i>Figure 6 Page Se connecter.....</i>	<i>12</i>