



Développement d'un Site Web pour l'Eglise  
orthodoxe de la Sainte-Résurrection

Développement d'un Site de Commerce pour  
la Vente de Chaises Gaming  
"Games-Thrones"

Projets réalisés dans le cadre de la présentation au  
**Titre professionnel Développeur Web et Mobile**

présentée par

**Alina Yefimova**

## Table de matières

<b>Introduction.....</b>	<b>4</b>
Développement d'un Site Web pour l'Eglise orthodoxe de la Sainte-Résurrection.....	6
COMPÉTENCES DU RÉFÉRENTIEL COUVERTES PAR LE PROJET.....	6
ACTIVITÉ TYPE N° 1 : Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité.....	6
ACTIVITÉ TYPE N° 2 : Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité.....	6
RÉSUMÉ DU PROJET.....	7
Introduction.....	7
Objectifs et Étendue du Projet.....	7
Liste des fonctionnalités.....	7
Structure Générale du Site.....	8
Fonctionnalités Supplémentaires.....	8
Rôles des Utilisateurs sur le Site.....	8
User Stories pour le site web de l'Eglise orthodoxe de la Sainte-Résurrection.....	9
SPÉCIFICATIONS TECHNIQUES.....	10
Choix technologiques Front-End.....	10
Choix technologiques Back-end.....	10
<b>Développement d'un Site de Commerce pour la Vente de Chaises Gaming "Games-Thrones".....</b>	<b>11</b>
COMPÉTENCES DU RÉFÉRENTIEL COUVERTES PAR LE PROJET.....	11
ACTIVITÉ TYPE N° 1 : Développer la partie front-end d'une application web ou web mobile.....	11
Maquetter une application :.....	11
Réaliser une interface utilisateur web statique et adaptable.....	12
ACTIVITÉ TYPE N° 2 : Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité.....	12
Créer une base de données.....	12
Développer la partie back-end d'une application web ou web mobile.....	12
RÉSUMÉ DU PROJET.....	12
CAHIER DES CHARGES.....	13
Objectif principal.....	13
Besoins de l'Application.....	13
Objectifs de l'Application.....	13
Liste des fonctionnalités.....	14
Fonctionnalités pour les visiteurs :.....	14
Fonctionnalités pour les Utilisateurs Connectés.....	14
Fonctionnalités pour les administrateur.....	15
Mise en place de l'application.....	15
Arborescence.....	15
Maquette.....	16

Charte graphique et logo.....	19
Utilisateurs et User Stories.....	19
Spécifications techniques.....	21
Versioning (Git et Git-flow).....	21
Branches.....	21
Choix technologiques (Front-End et Back-End).....	21
Front-End.....	21
Back-End.....	22
Accessibilité (Navigateurs et types d'appareils).....	23
Architecture du projet (MVC).....	23
Dossier public/.....	24
Dossier src/.....	25
Dossier templates/.....	25
Dossier vendor/.....	25
La Base de Données.....	26
■ MLD.....	26
■ Dictionnaire de données.....	26
■ Entity-Relationship Diagram (ERD).....	32
Routes.....	32
Mon rôle dans le projet.....	33
Design :.....	34
Développement front-end :.....	34
Fonctionnalités :.....	34
Contrôleur PHP : addProductToBasketAjaxController.php.....	34
Script JavaScript : product.js.....	36
Contrôleur PHP : getCartContentsAjaxController.php.....	37
Backend de paiement : checkout_backend.php.....	38
Classe PHP : crudProduct.class.php.....	40
Fichier product.js.....	42
Fichier sendReviewFormAjaxController.php.....	44
Product.php (Affichage des commentaires).....	45
CONCLUSION.....	50

# Introduction

Apprendre à programmer peut sembler une tâche intimidante, surtout lorsqu'il s'agit d'une langue étrangère. Cependant, je suis convaincue qu'avec la bonne approche, la persévérance et la foi en ses capacités, il est possible de réussir dans ce domaine.

Mon aventure professionnelle en France a débuté par l'apprentissage de la langue française. J'étais déterminée à trouver un moyen de m'épanouir dans ce nouveau pays en mettant à profit l'expérience acquise lors de mes précédents emplois en Ukraine.

Après avoir analysé le marché du travail dans la région, évalué mes compétences et mon état d'esprit, j'ai décidé de me concentrer sur le développement web. En explorant les différentes options de formation, j'ai découvert les cours "Développeur WEB et mobile" de l'école La Plateforme. Leur programme m'a semblé intéressant et prometteur, c'est pourquoi j'ai osé postuler.

Suivre ces cours représentait pour moi un engagement majeur : investir tout mon temps et mes ressources dans une formation à temps plein pour ne pas prendre de retard sur les autres. J'étais prête à apprendre beaucoup. Cependant, je ne m'attendais pas à me retrouver à la fin de ma formation à la limite de mes capacités psychologiques, ressentant un besoin urgent de commencer à travailler et de mettre en pratique mes nouvelles compétences.

Au cours de la formation "Développeur WEB et mobile", j'ai travaillé sur plusieurs projets, dont certains seront présentés dans ma présentation d'examen. Il s'agit aussi bien de projets individuels, tels que "Développement d'un Site Web pour l'Eglise orthodoxe de la Sainte-Résurrection" et "Développement d'un Catalogue Interactif de Pokémon "Pokedex"", que de projets en équipe, comme "Développement d'un Site de Commerce pour la Vente de Chaises Gaming "Games-Thrones"" avec une équipe de 4 personnes.

Ces projets m'ont permis de comprendre dans quel environnement je pouvais travailler le mieux : en solo ou en équipe. Les deux approches se sont révélées tout aussi précieuses pour moi. J'ai réalisé que le travail en solo me permet de me perfectionner dans les détails et les aspects pratiques, tandis que le travail en équipe me donne les connaissances nécessaires pour planifier et réaliser des projets de grande envergure.

Grâce aux cours "Développeur WEB et mobile", j'ai acquis des connaissances et des compétences solides dans le domaine du développement web. J'ai appris à travailler avec différents langages de programmation, frameworks et outils, et j'ai également maîtrisé les principes de conception UX/UI et de test de sites web.

Je suis convaincue que ces connaissances et compétences me permettront de démarrer avec succès une carrière de développeur web en France.

# Développement d'un Site Web pour l'Eglise orthodoxe de la Sainte-Résurrection

## COMPÉTENCES DU RÉFÉRENTIEL COUVERTES PAR LE PROJET

### ACTIVITÉ TYPE N° 1 : Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

#### **Maquetter une application :**

Au début du projet, la conception web a été finalisée et approuvée par les parties prenantes. Ces maquettes représentent une représentation visuelle détaillée de l'interface utilisateur finale, y compris le placement des éléments, la palette de couleurs, la typographie et les interactions prévues.

Le travail du designer a permis de définir clairement les orientations esthétiques et fonctionnelles du projet. Cette étape importante a facilité le passage à la phase de développement, car elle a fourni une référence visuelle précise sur laquelle baser la programmation des différentes fonctionnalités.

#### **Réaliser une interface utilisateur web statique et adaptable :**

Ensuite, j'ai réalisé l'intégration statique et responsive du site web avec les appareils mobiles (en utilisant HTML, CSS et le framework CSS Tailwind CSS). Il était important pour moi d'avoir une base fonctionnelle avant de l'intégrer dans la structure WordPress et de la rendre dynamique.

### ACTIVITÉ TYPE N° 2 : Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Travailler avec une base de données WordPress

Développement du backend d'une application web ou d'une application web mobile

La création d'un panneau de configuration back-end pour un thème WordPress.

## RÉSUMÉ DU PROJET

### Introduction

L'église orthodoxe de la Sainte-Résurrection a besoin d'un site web moderne et attrayant pour refléter son image et mieux servir ses paroissiens. Un site web obsolète ou mal conçu peut nuire à la crédibilité de l'église et décourager les visiteurs potentiels.

Un site web bien conçu peut être un outil de communication puissant pour l'église. Il permet de diffuser des informations importantes sur la paroisse, les horaires des services religieux, les événements à venir, les opportunités de bénévolat, et bien plus encore.

Un site web accessible et informatif peut aider l'église à attirer de nouveaux fidèles et bénévoles, en particulier les jeunes adultes et les familles qui recherchent une communauté religieuse dynamique.

## Objectifs et Étendue du Projet

### Objectif principal :

Moderniser le site web de l'église orthodoxe de la Sainte-Résurrection pour fournir des informations générales sur la paroisse aux paroissiens.

### Objectifs spécifiques :

1. Déterminer les objectifs, les idées et les besoins du site web.
2. Étudier les technologies de développement web.
3. Adapter la conception technique du site (structure).
4. Élaborer le cahier des charges pour la création du site web.
5. Développer le produit logiciel.

## Liste des fonctionnalités

### Adaptation du site web aux mobiles:

Le site web doit être responsive afin d'offrir une expérience utilisateur optimale sur tous les appareils, y compris les smartphones. Cela signifie que la mise en page et le contenu du site web doivent s'ajuster automatiquement en fonction de la taille de l'écran de l'utilisateur.

### Rôle du logo:

Le logo de l'application doit servir de lien redirigeant vers la page d'accueil du site web. En cliquant sur le logo, les utilisateurs doivent être dirigés vers la page principale du site.

### Menu du site web:

Le menu accessible sur l'ensemble du site web doit être présent, y compris sur les appareils mobiles. Sur les smartphones, le menu devrait prendre la forme d'un "menu burger" pour optimiser l'espace d'affichage. Ce menu doit contenir des liens vers les différentes pages du site, à l'exception du contenu du pied de page.

### Pied de page:

Le footer doit être présent en bas de chaque page du site web. Ce pied de page doit inclure des liens vers les réseaux sociaux de l'église et ses coordonnées de contact, telles que l'adresse e-mail et le numéro de téléphone.

## Structure Générale du Site

### Page d'Accueil :

Présente un aperçu de la paroisse, sa mission, ses valeurs, et ses coordonnées.

### Horaires des Messes et Événements :

Affiche les horaires des messes hebdomadaires, des événements spéciaux, et d'autres activités paroissiales.

### À Propos :

Contient une histoire détaillée de la paroisse, sa foi, et ses membres.

### Contact :

Permet aux visiteurs de contacter la paroisse via email, téléphone, ou formulaire de contact.

#### **Nouvelles et Articles :**

Partage des nouvelles paroissiales, des événements, des articles de réflexion spirituelle, et des histoires inspirantes.

### **Fonctionnalités Supplémentaires**

#### **Dons en Ligne :**

Offre un moyen sécurisé de faire des dons à la paroisse.

#### **Boutique en Ligne :**

Permet d'acheter des livres, de la musique, des cadeaux, et d'autres articles religieux.

### **Rôles des Utilisateurs sur le Site**

#### **Administrateurs :**

Ont un accès complet à toutes les fonctionnalités du site, y compris la possibilité d'éditer le contenu et de gérer les paramètres du site.

#### **Invités :**

Ont accès à la plupart du contenu du site, peuvent s'inscrire aux événements ou faire des dons.

Le public cible du site web de l'église orthodoxe de la Sainte-Résurrection peut être divisé en plusieurs groupes principaux

### **Mise en place de l'application**

#### **User Stories pour le site web de l'Église orthodoxe de la Sainte-Résurrection**

1. Membres actuels de la paroisse:
2. Personnes intéressées par l'orthodoxie:
3. Grand public:

Rôles:

- Utilisateur: Un membre de la paroisse ou une personne intéressée par l'église orthodoxe.
- Administrateur: Un membre du personnel de l'église responsable de la gestion du site web.

User Stories pour les utilisateurs

En tant qu'utilisateur, je souhaite :

- Pouvoir trouver facilement des informations sur l'église, telles que son histoire, ses croyances et ses pratiques.
- Consulter le calendrier des messes et des événements.
- Accéder à des sermons, des articles et d'autres ressources spirituelles.



- En savoir plus sur les programmes et les services offerts par l'église.
- Contacter l'église pour poser des questions ou demander des informations supplémentaires.
- Faire un don à l'église en ligne.
- S'inscrire pour recevoir des newsletters et des mises à jour par e-mail.
- Trouver des liens vers les réseaux sociaux de l'église.
- Naviguer facilement sur le site web, même sur un appareil mobile.
- Avoir accès à un contenu informatif et engageant qui me motive à explorer davantage ma foi.

User Stories pour les administrateurs

En tant qu'administrateur, je souhaite :

- Pouvoir me connecter au site web en utilisant un nom d'utilisateur et un mot de passe sécurisés.
- Modifier le contenu du site web, y compris les pages, les articles, les événements et les sermons.
- Ajouter de nouvelles images et vidéos au site web.
- Gérer les formulaires de contact et les demandes de renseignements.
- Suivre les statistiques du site web, telles que le nombre de visiteurs et les pages les plus consultées.
- Créer des utilisateurs et gérer leurs permissions.
- Sauvegarder et restaurer le site web en cas de problème.
- Recevoir des notifications par e-mail lorsque du nouveau contenu est soumis ou que des modifications sont apportées au site web.
- Avoir accès à une interface d'administration conviviale et intuitive.

## SPÉCIFICATIONS TECHNIQUES

Le contrôle des versions permet de suivre l'historique du code source et de collaborer efficacement. Il est crucial de conserver les versions précédentes du code, surtout en cas de problèmes majeurs avec les fonctionnalités ou de modifications importantes du code. Pour gérer les versions, j'ai choisi d'utiliser le système de contrôle de versions décentralisé Git et le service d'hébergement web GitHub.

### Choix technologiques Front-End

Pour le développement de la partie frontale du site web (visible par l'utilisateur), j'ai utilisé les langages suivants :

- HTML 5 (HyperText Markup Language): Ce langage est la base de la structure des pages web et définit le contenu et les éléments qui les composent.
- CSS 3 (Cascading Style Sheets): Ce langage permet de styliser les pages web en définissant la mise en page, les couleurs, les polices de caractères et d'autres aspects visuels.

Amélioration du dynamisme avec JavaScript et jQuery

Afin de rendre le site web plus dynamique et interactif, j'ai utilisé le langage de programmation JavaScript et la bibliothèque JavaScript open-source populaire jQuery. Cela m'a permis d'implémenter des fonctionnalités telles que :

- Des animations et des effets visuels
- Des interactions avec l'utilisateur, telles que des clics et des formulaires
- Le chargement dynamique de contenu

Utilisation de Tailwind CSS pour un design réactif

J'ai également utilisé la bibliothèque Tailwind CSS pour simplifier le développement CSS et créer un design réactif. Tailwind CSS fournit un ensemble de classes CSS pré-construites qui permettent de styliser rapidement et facilement les éléments HTML.

Fonctionnalités implémentées

Grâce à ces choix technologiques, j'ai pu implémenter les fonctionnalités suivantes sur le site web :

- Des sliders et des onglets pour une navigation intuitive
- Des boutons mis en évidence pour une meilleure interaction utilisateur
- Un design adaptatif qui s'ajuste automatiquement à la taille de l'écran de l'utilisateur

## Développement d'un Site de Commerce pour la Vente de Chaises Gaming “Games-Thrones”

### COMPÉTENCES DU RÉFÉRENTIEL COUVERTES PAR LE PROJET

En ce début de présentation, nous allons découvrir le projet de développement d'un site de commerce pour la vente de chaises gaming "Games-Thrones". Nous allons explorer les différentes étapes de ce projet, en commençant par l'analyse des besoins, puis la conception, le développement et enfin la mise en ligne et la promotion du site.

## ACTIVITÉ TYPE N° 1 : Développer la partie front-end d'une application web ou web mobile

### Maquetter une application :

Il est beaucoup plus facile de concevoir un site Internet si le concepteur connaît à l'avance la logique de disposition des éléments sur chaque page.

Etape 1:

Avant de commencer la conception du site, notre équipe a créé un schéma de composition pour définir l'organisation des éléments sur les pages.

Ensuite, nous avons élaboré l'arborescence du site, qui représente la hiérarchie des pages et leur interconnexion.

Etape 2 : Maquette du site avec Figma

- Pour réaliser la maquette du site, nous avons utilisé l'outil de design Figma.
- Le choix de Figma s'explique par plusieurs raisons :
  - Collaboration en temps réel: Figma permet à plusieurs personnes de travailler simultanément sur la maquette, ce qui était crucial pour notre projet.
  - Interface intuitive: Figma offre une interface conviviale et facile à prendre en main, même pour les débutants.
  - Fonctionnalités puissantes: Figma propose un large éventail de fonctionnalités pour créer des maquettes réalistes et interactives.
  - Version gratuite: Figma dispose d'une version gratuite suffisante pour la plupart des projets.

### Réaliser une interface utilisateur web statique et adaptable

Dans le cadre de la création du site web "Games-Thrones", nous allons nous concentrer sur la réalisation d'une interface utilisateur web statique et adaptable. Pour réaliser l'interface utilisateur de notre site web, nous allons utiliser les outils et technologies suivants :

- HTML5: Langage standard pour la création de pages web.
- CSS3: Langage pour définir le style des pages web (mise en page, couleurs, polices de caractères, etc.).
- jQuery (bibliothèque JavaScript)

## ACTIVITÉ TYPE N° 2 : Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

### Créer une base de données

Tout comme la conception de l'application elle-même, il était important pour nous de structurer nos données dès le départ. Pour ce faire, nous avons commencé par réfléchir aux

données que nous voulions intégrer à l'application, puis nous avons créé un modèle conceptuel de données (MCD).

Pour gérer la base de données de notre application, nous avons choisi d'utiliser l'interface d'administration web MyAdmin.

## Développer la partie back-end d'une application web ou web mobile

La partie backend de projet est structurée avec une répartition claire des rôles entre les différents contrôleurs et fichiers de configuration. Les routes définies dans index.php avec AltoRouter montrent une configuration robuste permettant de gérer diverses pages et fonctionnalités de l'application. Les dossiers ajax, backOffice, et classes suggèrent une organisation orientée vers des pratiques de développement modernes et modulaires.

# RÉSUMÉ DU PROJET

## Introduction

L'industrie du gaming connaît une croissance rapide et constante, entraînant une demande accrue pour des équipements spécialisés, notamment des chaises gaming. Ces chaises ne sont pas seulement un luxe mais une nécessité pour les gamers sérieux et les travailleurs à domicile qui passent de longues heures devant leur écran. Elles offrent un soutien ergonomique crucial, ce qui aide à prévenir les douleurs et les blessures liées à une mauvaise posture.

Cependant, pour capter l'attention de ce marché en pleine expansion, il est essentiel de disposer d'une plateforme de commerce électronique qui soit à la fois attrayante et fonctionnelle. Un site web moderne et bien conçu peut jouer un rôle central en attirant des clients potentiels et en les convertissant en acheteurs. En revanche, un site web obsolète ou mal conçu peut nuire à la crédibilité de l'entreprise et décourager les visiteurs, les poussant à se tourner vers des concurrents.

## Objectif principal

Développer un site web de commerce électronique pour vendre des chaises gaming de haute qualité, offrant une interface utilisateur intuitive et une gestion efficace des produits.

## Besoins de l'Application

1. Expérience Utilisateur (UX)
  - Navigation Intuitive : L'utilisateur doit pouvoir naviguer facilement et rapidement sur le site, trouver les produits qu'il recherche et passer à l'achat sans encombre.
  - Conception Responsive : Le site doit être optimisé pour tous les types d'appareils (ordinateurs de bureau, tablettes, smartphones).

- Chargement Rapide : Les pages doivent se charger rapidement pour éviter la frustration des utilisateurs et réduire le taux de rebond.
- 2. Catalogue de Produits
  - Descriptions Détaillées : Chaque chaise gaming doit avoir une description détaillée incluant les caractéristiques, les avantages, et les spécifications techniques.
  - Images et Vidéos de Haute Qualité : Les produits doivent être présentés avec des images et des vidéos de haute qualité pour donner une vue précise et attrayante aux utilisateurs.
  - Filtres et Catégories : Les utilisateurs doivent pouvoir filtrer et trier les produits par prix, marque, fonctionnalités, etc.
- 3. Processus d'Achat
  - Panier d'Achat : Une fonctionnalité de panier facile à utiliser où les utilisateurs peuvent ajouter, supprimer ou modifier les produits avant de finaliser leur achat.
  - Paiement Sécurisé : Intégration de méthodes de paiement sécurisées (cartes de crédit, PayPal, etc.) et conformité aux normes de sécurité.
  - Suivi des Commandes : Les utilisateurs doivent pouvoir suivre le statut de leur commande depuis la confirmation jusqu'à la livraison.
- 4. Gestion de Contenu
  - Interface d'Administration : Un back-office permettant de gérer les produits, les commandes, les clients, et le contenu du site (blogs, avis, etc.).

## Objectifs de l'Application

1. Augmentation des Ventes
  - Créer un site attrayant et fonctionnel qui facilite le parcours d'achat de l'utilisateur, augmentant ainsi les taux de conversion et les ventes.
2. Fidélisation de la Clientèle
  - Développer des fonctionnalités et des services qui encouragent les clients à revenir et à effectuer des achats répétés.
3. Visibilité et Trafic
  - Améliorer le référencement et la visibilité du site pour attirer plus de visiteurs via les moteurs de recherche et les réseaux sociaux.
4. Satisfaction Client
  - Offrir une expérience utilisateur de haute qualité, un service client réactif, et des produits de qualité pour assurer la satisfaction des clients.
5. Optimisation des Processus Internes
  - Utiliser des outils de gestion et d'analyse pour optimiser les opérations internes, de la gestion des stocks à l'analyse des performances du site.
6. Expansion du Marché
  - Atteindre un public plus large en ajoutant des fonctionnalités multilingues et en facilitant l'expédition internationale.

## Liste des fonctionnalités

Pages accessibles aux visiteurs (utilisateurs non-connectés) :

- Accueil
- Page resultat de la recherche
- Page produit
- Formulaire de connexion
- Formulaire d'inscription
- Page de profil
- Panier d'Achat
- Mentions légale
- Conditions Générales d'Utilisation
- Nous contacter

## Fonctionnalités pour les visiteurs :

- Création de compte
- Connexion à leur compte
- Faire un achat
- Trouver facilement des produits spécifiques.
- Accès rapide aux catégories de produits.
- Consultation des avis et notes des autres clients.

## Fonctionnalités pour les Utilisateurs Connectés

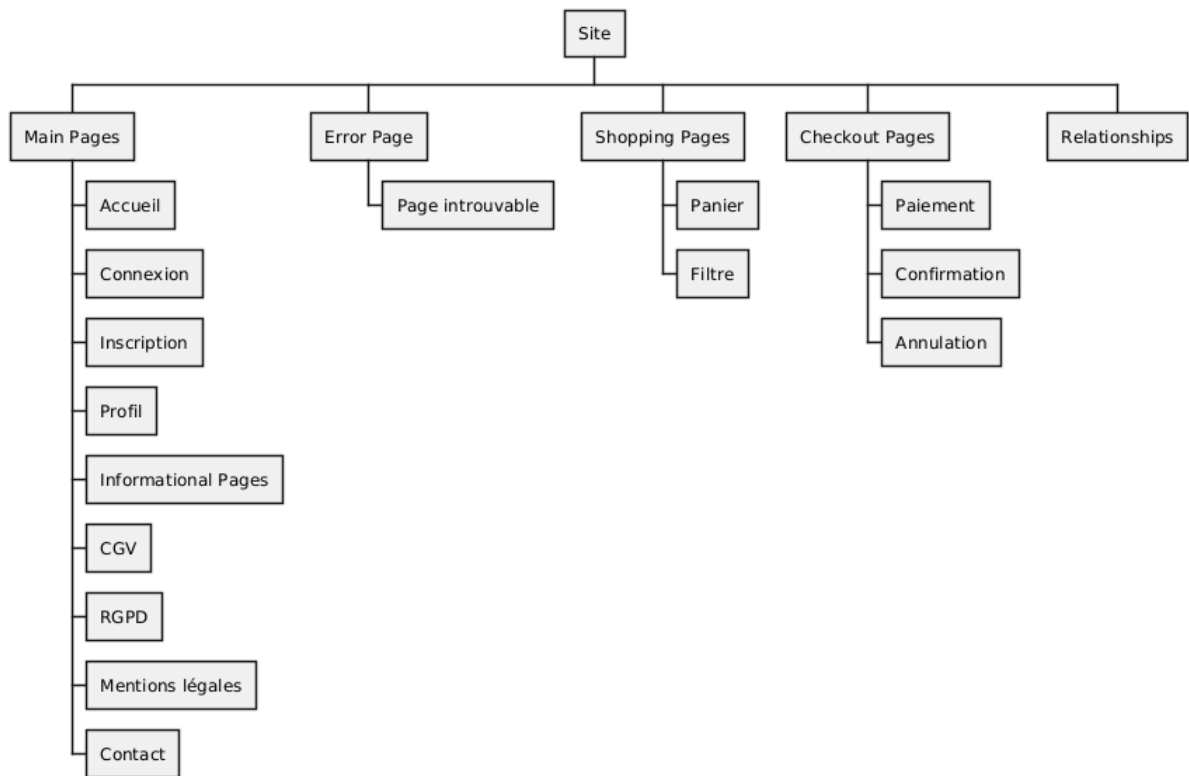
- Connexion à leur compte
- Mise à jour des informations personnelles et des préférences.
- Ajouter un commentaire
- Faire un achat

## Fonctionnalités pour les administrateur

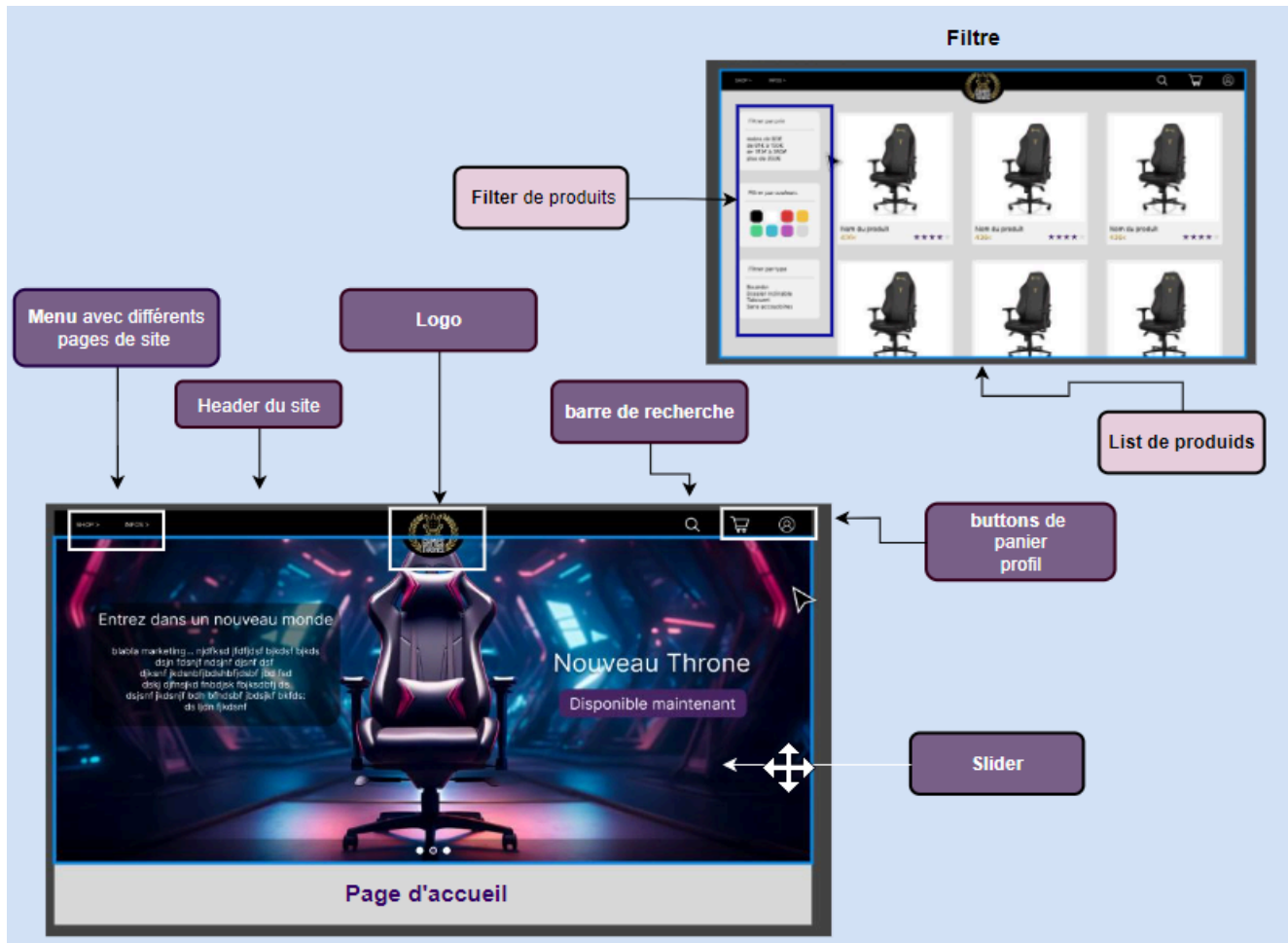
- Ajouter, éditer et supprimer des produits.
- Télécharger et gérer des médias produits.
- Ajouter des descriptions et spécifications.
- Créer et gérer des catégories.
- Configurer des filtres pour les produits.
- Voir et mettre à jour les commandes.
- Accéder aux informations des commandes.

## Mise en place de l'application

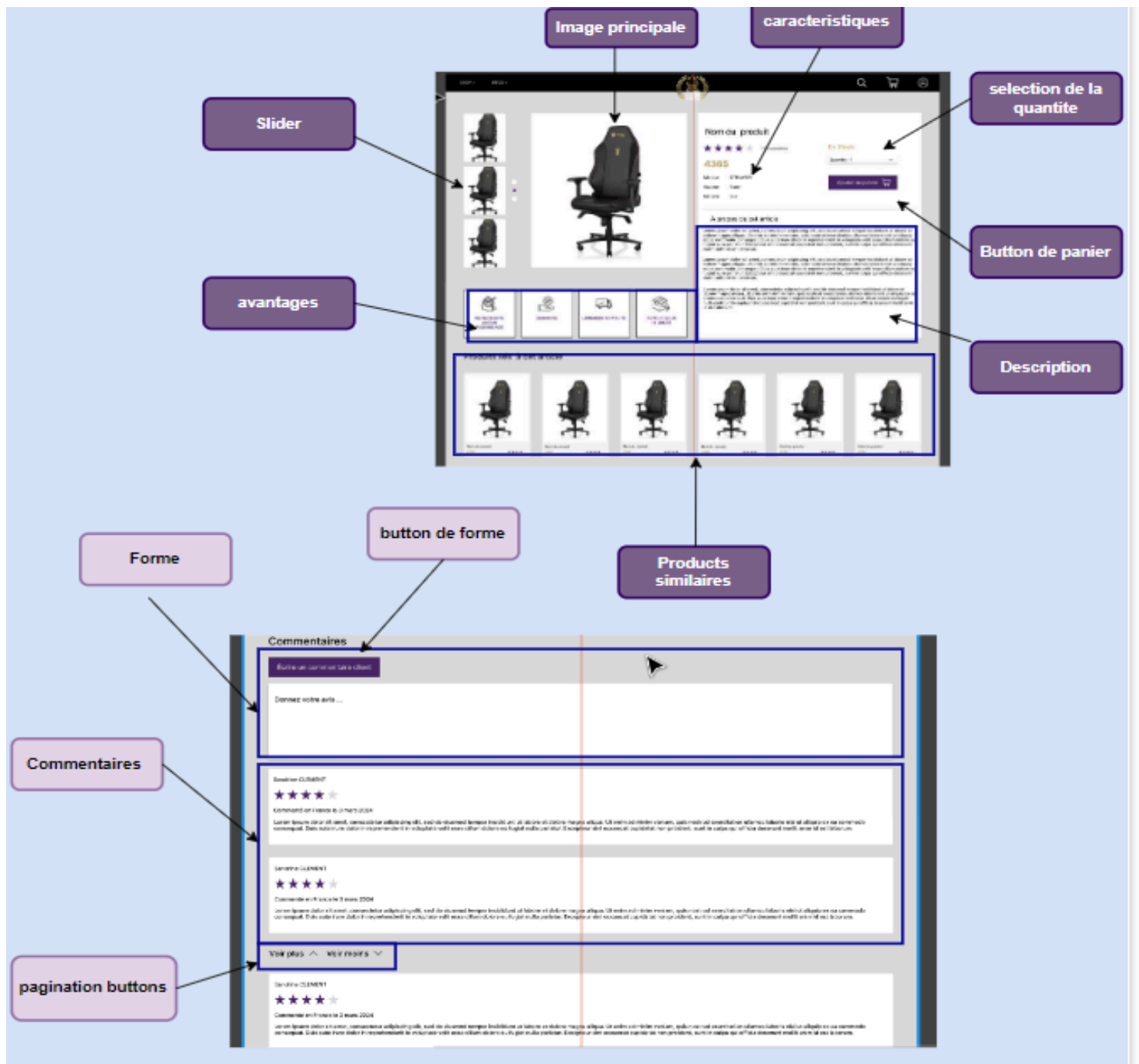
### Arborescence



## Maquette



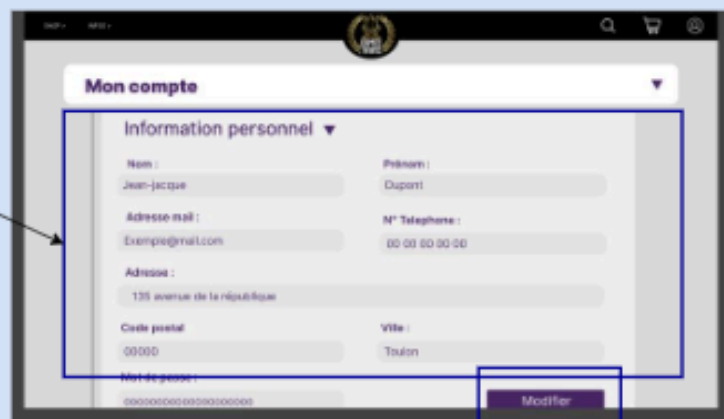






accordions

Profil de utilisateur



information personnel

button modification des données




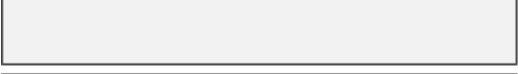



Version mobile

Menu burger

## Charte graphique et logo

### Palette de couleurs:

	#000000
	#482664
	#BAA35F
	#F2F2F2
	#BAA35F

### Le logo:

Le logo est un emblème rond de couleur dorée représentant un trône entouré d'une couronne de laurier impossible. Le trône symbolise le pouvoir et l'autorité, tandis que la couronne de laurier impossible représente la victoire et la réalisation. La couleur dorée est associée à la richesse, au luxe et au prestige.



### Typographie:

Roboto a été choisi comme police principale. Il s'agit d'une police sans empattement développée par Google. Roboto est une police polyvalente et moderne qui convient parfaitement à une variété d'applications, y compris la conception de sites Web, la conception d'impression et l'image de marque.

## Utilisateurs et User Stories

Public visé:

Gamers : Les joueurs qui recherchent des chaises ergonomiques et stylées pour améliorer leur expérience de jeu.

Professionnels de bureau : Les personnes travaillant de longues heures à un bureau et ayant besoin de chaises confortables et ergonomiques.

Parents : Les parents cherchant des chaises adaptées pour leurs enfants, avec des designs ludiques et ergonomiques.

Enthusiastes du design d'intérieur : Les individus intéressés par des meubles de qualité qui s'intègrent bien dans leur espace de vie.

Rôles:

Visiteur : Utilisateur non-connecté naviguant sur le site.

Client : Utilisateur ayant un compte et connecté.

Administrateur : Utilisateur ayant des droits complets de gestion du site.

## User Stories

En tant que	Je souhaite	Afin de
Visiteur	je veux filtrer les produits par catégorie	trouver facilement les chaises gaming qui m'intéressent.
Visiteur	je veux lire les avis des autres clients	m'assurer de la qualité et de la satisfaction des produits avant d'acheter
Visiteur	je veux pouvoir contacter le support	poser des questions sur les produits, le processus de commande ou la livraison.
Visiteur	je veux accéder aux détails complets des produits	connaître toutes les caractéristiques et les spécifications des chaises gaming qui m'intéressent.
Visiteur	je veux voir les produits en vedette	ne pas manquer les meilleures offres et nouveautés.
Visiteur	je veux pouvoir rechercher des produits	trouver facilement des articles spécifiques.
Client	je veux créer un compte pour gérer mes informations personnelles et suivre mes commandes.	faciliter mes futurs achats et gérer mes interactions avec le site.
Client	je veux voir l'historique de mes commandes	suivre mes achats passés et gérer mes réclamations si nécessaire.
Client	je veux pouvoir modifier mon panier d'achat	ajuster les quantités et les articles avant de finaliser l'achat
Client	je veux des options de paiement sécurisées pour finaliser mes achats	garantir la sécurité de mes informations financières.
Client	je veux pouvoir laisser des avis sur les produits	partager mon expérience et aider les autres clients dans leurs choix.
Administrateur	je veux ajouter, éditer et supprimer des produits pour maintenir le catalogue à jour	proposer aux clients une sélection de chaises gaming large et variée.
Administrateur	je veux gérer les images et vidéos des produits	présenter les produits de manière attrayante et informative.

Administrateur	je veux catégoriser les produits et configurer des filtres	faciliter la navigation et la recherche pour les utilisateurs.
Administrateur	je veux accéder aux détails des commandes	suivre et gérer efficacement les ventes et les livraisons.
Administrateur	je veux gérer les comptes utilisateurs	fournir un support personnalisé et résoudre les problèmes des utilisateurs.

## Spécifications techniques

### Versioning (Git et Git-flow)

La gestion de version permet de conserver l'historique du code source et de travailler plus facilement en équipe. Il était très important pour nous de garder les versions précédentes du code, notamment en cas de gros problème avec une fonctionnalité suite à des modifications dans le code. Pour cette gestion, nous avons décidé d'utiliser le logiciel de version décentralisé Git et le service web d'hébergement GitHub.

Le dépôt Git pour le projet "Games-Thrones" est hébergé sur GitHub et accessible à l'adresse suivante : <https://github.com/Jessy-Charlet/Games-Thrones>

### Branches

Les branches Git sont utilisées pour séparer les différentes fonctionnalités et les travaux en cours. Voici quelques branches typiques que vous pourriez trouver dans le dépôt :  
main : La branche principale contenant le code de production stable.

Le dépôt GitHub pour le projet "Games-Thrones" utilise les issues pour gérer les tâches et les bugs.

## Choix technologiques (Front-End et Back-End)

### Front-End

Le choix technologique concernant le Front-End (partie dite client, avec le code visible par l'utilisateur) a été fait d'un commun accord.

Nous avons bien évidemment utilisé le langage de structuration de page HTML 5 (HyperText Markup Language) et les Feuilles de Style en Cascade CSS 3 (Cascading Style Sheets). Nous avons opté, pour la dynamisation de l'application, d'utiliser le langage de programmation de scripts JavaScript et une bibliothèque JavaScript libre - Query.

Explication du choix

L'utilisation de technologies de base comme HTML, CSS et JavaScript permet aux développeurs de se concentrer sur les concepts fondamentaux du développement web sans se plonger dans les complexités d'un framework spécifique.

Cela peut être particulièrement bénéfique pour les équipes avec des membres ayant des niveaux d'expérience variés ou pour les projets nécessitant une rapidité d'exécution.

## Back-End

Le back-end du site de commerce électronique « Games-Thrones » semble être construit à l'aide des technologies suivantes :

PHP : est un langage de script à usage général largement utilisé pour le développement Web côté serveur.

Composer : est un outil de gestion des dépendances pour PHP qui simplifie le processus d'installation et de mise à jour des bibliothèques et dépendances tierces.

Altorouter : est une bibliothèque de routage légère pour PHP qui fournit un moyen simple et efficace de gérer le routage d'URL dans les applications Web. Il permet aux développeurs de définir des routes et de les associer à des actions spécifiques du contrôleur.

Architectures possibles :

Compte tenu de la présence de ces technologies, il est probable que le back-end du site Web « Games-Thrones » suive une architecture Model-View-Controller (MVC). Il s'agit d'un modèle architectural courant pour les applications Web qui sépare les préoccupations de l'application en trois couches distinctes :

Modèle : la couche modèle représente les données de l'application, y compris la logique d'accès aux données et de manipulation.

Vue : la couche de vue est chargée de présenter les données à l'utilisateur, généralement sous la forme de pages HTML.

Contrôleur : la couche contrôleur agit comme intermédiaire entre le modèle et la vue, gérant les demandes des utilisateurs, récupérant les données du modèle et mettant à jour la vue en conséquence.

Mise en œuvre possible :

Dans une application PHP utilisant Composer, la bibliothèque Altorouter peut être utilisée pour définir des routes et les associer à des actions spécifiques du contrôleur. Les classes de contrôleur interagiraient ensuite avec la couche modèle pour récupérer et manipuler les données, et avec la couche vue pour restituer les réponses HTML appropriées.

Considérations supplémentaires:

Le back-end peut également utiliser d'autres technologies et outils pour la connectivité des bases de données, la validation des données, la gestion des erreurs, les mesures de sécurité et d'autres aspects du développement d'applications Web.

Conclusion:

Le back-end du site Web de commerce électronique « Games-Thrones » semble être construit à l'aide de PHP, Composer et Altorouter, suivant probablement un modèle architectural MVC. Cette combinaison de technologies constitue une base solide pour une application de commerce électronique évolutive et maintenable.

S'il vous plaît laissez-moi savoir si vous avez d'autres questions.

## Accessibilité (Navigateurs et types d'appareils)

Navigateurs compatibles Selon une étude de marché de 2024, les navigateurs les plus utilisés sont :

Chrome : 60,25%

Safari : 18,4%

Firefox : 8,94%

Notre application devra donc être compatible avec les versions les plus récentes des navigateurs cités ci-dessus.

Source:

<https://www.leptidigital.fr/webmarketing/parts-de-marche-navigateurs-web-10814/#:~:text=R%C3%A9sum%C3%A9%20de%20la%20r%C3%A9partition%20des%20navigateurs%20web%20en%20France%20sur,Edge%20%3A%2011%2C74%20%25>

## Architecture du projet (MVC)

En se basant sur la structure des répertoires fournie, le projet Games-Thrones essaye de suivre une architecture Model-View-Controller (MVC) avec quelques éléments supplémentaires. Voici une analyse des principaux répertoires et de leurs rôles potentiels :

- index.php : Point d'entrée principal du site web.
- boutique.sql : Fichier de base de données SQL contenant les structures de la base de données.
- composer.json et composer.lock : Fichiers de configuration pour Composer, utilisés pour la gestion des dépendances PHP.
- secrets.php : Fichier contenant des informations sensibles comme les clés API, les mots de passe, etc.

## Dossier public/

- assets/ : Contient les ressources statiques du site.
  - css/ : Feuilles de style CSS.
  - img/ : Images utilisées sur le site.
  - js/ : Scripts JavaScript.
- backOffice/ : Contient les fichiers relatifs à la gestion du back-office.
  - backOffice.php : Fichier principal pour l'administration du site.
- controller/ : Contient les contrôleurs de l'application, séparés par types et fonctionnalités.
  - js/ : Contrôleurs JavaScript.
    - backOffice/
      - backOfficeController.js : Contrôleur pour les opérations du back-office.
    - postalCodeController.js : Contrôleur pour les opérations liées aux codes postaux.
    - profilController.js : Contrôleur pour la gestion des profils utilisateurs.
    - signInController.js : Contrôleur pour la gestion de la connexion des utilisateurs.
  - php/ : Contrôleurs PHP.
    - ajax/ : Contrôleurs pour les requêtes AJAX.
      - addProductToBasketAjaxController.js : Ajout de produits au panier.
      - getCartContentsAjaxController.js : Récupération du contenu du panier.
      - getProductDataByIdAjaxController.js : Récupération des données produits par ID.
      - getReviewAjaxController.js : Récupération des avis produits.
      - sendReviewFormAjaxController.js : Envoi des formulaires d'avis.
    - backOffice/ : Contrôleurs spécifiques au back-office.
      - backOfficeController.php : Contrôleur PHP pour les opérations du back-office.
    - classes/ : Classes PHP utilisées par les contrôleurs.
      - CommandsManager.class.php : Gestion des commandes.
      - crudProduct.class.php : Opérations CRUD pour les produits.
      - crudReview.class.php : Opérations CRUD pour les avis.
      - crudUser.class.php : Opérations CRUD pour les utilisateurs.
      - Database.class.php : Gestion de la base de données.
      - Product.class.php : Classe représentant un produit.
    - deconnexion.php : Gestion de la déconnexion.
    - displayProduct.php : Affichage des produits.
    - profilController.php : Gestion des profils utilisateurs.
    - reviewHandlerController.php : Gestion des avis utilisateurs.
    - recherche.php : Recherche de produits.
    - signInController.php : Gestion de la connexion des utilisateurs.
    - signUpController.php : Gestion de l'inscription des utilisateurs.



## Dossier src/

- Contient les différents fichiers PHP pour les pages du site.
  - 404.php : Page d'erreur 404.
  - basket\_json.php : Gestion du panier en JSON.
  - basket.php : Page du panier.
  - cancel.php : Page d'annulation.
  - cgv.php : Conditions générales de vente.
  - checkout\_backend.php : Gestion du paiement.
  - contact.php : Page de contact.
  - filters.php : Gestion des filtres de produits.
  - home.php : Page d'accueil.
  - mention.php : Mentions légales.
  - product.php : Page produit.
  - profil.php : Page de profil utilisateur.
  - result.php : Page de résultats de recherche.
  - rgpd.php : Gestion des données RGPD.
  - signIn.php : Page de connexion.
  - success.php : Page de succès de commande.

## Dossier templates/

- Contient les templates PHP pour les différentes parties du site.
  - banner.php : Template de la bannière.
  - footer.php : Template du pied de page.
  - header.php : Template de l'en-tête.

## Dossier vendor/

- Contient les dépendances gérées par Composer.
  - altorouter : Dépendance pour la gestion des routes.
  - composer : Fichiers de gestion des dépendances.
  - stripe : Dépendance pour l'intégration de Stripe.
  - .gitattributes : Fichier de configuration Git.

Modèle : Implémenté à l'aide de classes PHP dans le dossier controller/php (par exemple, Database.class.php, Product.class.php). Ces classes peuvent interagir avec la base de données (représentée par boutique.sql) pour gérer les données.

Vue : Implémentée à l'aide d'une combinaison de fichiers PHP individuels (dossier src) et de modèles réutilisables (dossier templates).

Contrôleur : Géré par des contrôleurs PHP dans le dossier controller. Ces contrôleurs gèrent les requêtes des utilisateurs, interagissent avec le modèle pour récupérer ou manipuler des données, puis choisissent la vue appropriée à afficher. De plus, certains contrôleurs JavaScript peuvent exister pour des fonctionnalités spécifiques.

## La Base de Données

### ■ MLD

- Catégorie: id(identifiant unique), nom , parent\_id (identifiant de la catégorie parente)
- Client: id, prénom, nom email, address,e code\_postal, ville téléphone, rôle, mot\_de\_passe
- Image: id (identifiant unique), url, principale
- Produit: id (identifiant unique), nom , note, prix, quantité, description, couleur, matière, marque, catégorie\_id
- Commande: id (identifiant unique), stripe\_id (identifiant de la commande Stripe)
- Commande\_client: commande\_id (identifiant de la commande), client\_id (identifiant du client)
- Commande\_produit: commande\_id (identifiant de la commande), produit\_id (identifiant du produit)
- Avis: produit\_id (identifiant du produit), client\_id (identifiant du client), texte (avis du client), note (note attribuée au produit par le client)

### ■ Dictionnaire de données

category			
id	Code numérique d'un category	int	Primary key
name	Name of the category	varchar	NOT NULL
parent_id		int	Foreign key to the id column of the category table
customer			
id	Code numérique d'un category	int	Primary key
first_name	First name of the customer	varchar	NOT NULL
last_name	Last name of the customer	varchar	NOT NULL

mail	Email address of the customer	varchar	NOT NULL
adresse	Address of the customer	varchar	NOT NULL
postal_code	Postal code of the customer	varchar	NOT NULL
city	City of the customer	varchar	NOT NULL
phone	Phone number of the customer	varchar	NOT NULL
role	Role of the customer (e.g., "client", "admin")	varchar	NOT NULL
password	Password of the customer	varchar	NOT NULL
image			
id	Code numérique d'un image	int	Primary key
url	URL of the image	tinytext	NOT NULL
main	Whether the image is the main image for a product	tinyint	NOT NULL
image_product			
product_id	Foreign key to the id column of the product	int	NOT NULL
image_id	Foreign key to the id column of the image table	int	NOT NULL
order			
id	Code numérique d'un order	int	Primary key
stripe_id	Stripe ID of the order	tinytext	NOT NULL

order_cutomer			
order_id	Foreign key to the id column of the order table	int	NOT NULL
customer_id	Foreign key to the id column of the customer table	int	NOT NULL
order_product			
order_id	Foreign key to the id column of the order table	int	NOT NULL
product_id	Foreign key to the id column of the product table	int	NOT NULL
product			
id	Code numérique d'un product	int	Primary key
name	Name of the product	varchar	NOT NULL
rate	Rating of the product	float	NOT NULL
price	Price of the product	float	NOT NULL
quantity	Quantity of the product in stock	int	NOT NULL
description	Description of the product	text	NOT NULL
color	Color of the product	varchar	NOT NULL
material	Material of the product	varchar	NOT NULL
brand	Brand of the product	varchar	NOT NULL
category_id	Foreign key to the id column of the	int	NOT NULL
review			

product_id	Foreign key to the id column of the product	int	NOT NULL
customer_id	Foreign key to the id column of the customer table	int	NOT NULL
text	Text of the review	tex	NOT NULL
rating	Rating of the product		NOT NULL

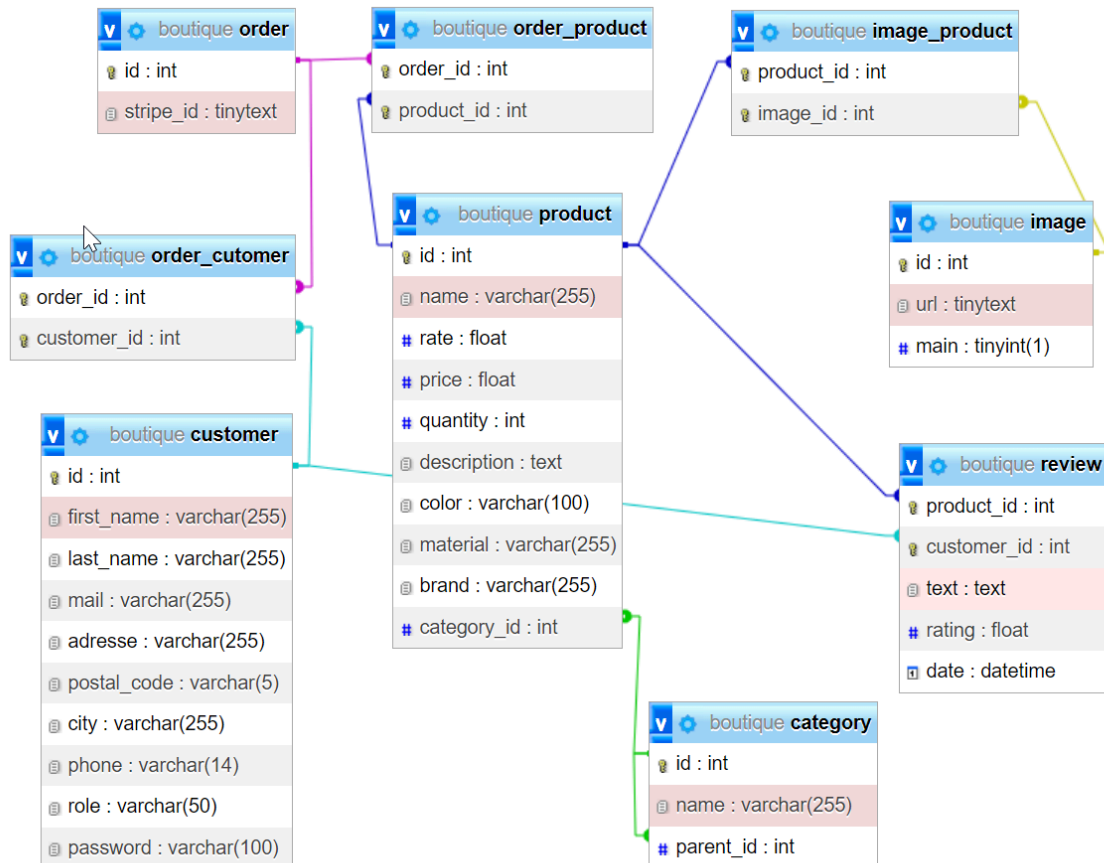
category			
id	Code numérique d'un category	int	Primary key
name	Name of the category	varchar	NOT NULL
parent_id		int	Foreign key to the id column of the category table
customer			
id	Code numérique d'un category	int	Primary key
first_name	First name of the customer	varchar	NOT NULL
last_name	Last name of the customer	varchar	NOT NULL
mail	Email address of the customer	varchar	NOT NULL
adresse	Address of the customer	varchar	NOT NULL
postal_code	Postal code of the customer	varchar	NOT NULL
city	City of the customer	varchar	NOT NULL

phone	Phone number of the customer	varchar	NOT NULL
role	Role of the customer (e.g., "client", "admin")	varchar	NOT NULL
password	Password of the customer	varchar	NOT NULL
image			
id	Code numérique d'un image	int	Primary key
url	URL of the image	tinytext	NOT NULL
main	Whether the image is the main image for a product	tinyint	NOT NULL
image_product			
product_id	Foreign key to the id column of the product	int	NOT NULL
image_id	Foreign key to the id column of the image table	int	NOT NULL
order			
id	Code numérique d'un order	int	Primary key
stripe_id	Stripe ID of the order	tinytext	NOT NULL
order_cutomer			
order_id	Foreign key to the id column of the order table	int	NOT NULL
customer_id	Foreign key to the id column of the customer table	int	NOT NULL

order_product			
order_id	Foreign key to the id column of the order table	int	NOT NULL
product_id	Foreign key to the id column of the product table	int	NOT NULL
product			
id	Code numérique d'un product	int	Primary key
name	Name of the product	varchar	NOT NULL
rate	Rating of the product	float	NOT NULL
price	Price of the product	float	NOT NULL
quantity	Quantity of the product in stock	int	NOT NULL

description	Description of the product	text	NOT NULL
color	Color of the product	varchar	NOT NULL
material	Material of the product	varchar	NOT NULL
brand	Brand of the product	varchar	NOT NULL
category_id	Foreign key to the id column of the	int	NOT NULL
review			
product_id	Foreign key to the id column of the product	int	NOT NULL
customer_id	Foreign key to the id column of the customer table	int	NOT NULL
text	Text of the review	tex	NOT NULL
rating	Rating of the product		NOT NULL

## ■ Entity-Relationship Diagram (ERD)



## Routes

## Pages principales

- \* accueil (Accueil) - La page d'accueil du site Web.
- \* connexion (Connexion) - Page permettant aux utilisateurs enregistrés de se connecter à leur compte.
- \* inscription (Inscription) - Page permettant aux nouveaux utilisateurs de créer un compte.
- \* profil (Profil) - Une page où les utilisateurs peuvent gérer les informations de leur compte, telles que l'adresse, les informations de paiement et l'historique des commandes.

## Pages d'information

- \* cgv (Conditions générales de vente) - Une page décrivant les termes et conditions qui régissent la vente de produits sur le site Web.



\* rgpd (RGPD) - Une page expliquant les pratiques du site Web en matière de confidentialité des données et leur conformité au règlement général sur la protection des données (RGPD).

\* mention (Mentions légales) - Une page contenant des informations juridiques sur la société qui possède et exploite le site Web.

\* contact (Contact) - Une page avec les coordonnées de l'entreprise, telles que l'adresse e-mail, le numéro de téléphone et l'adresse physique (si applicable).

### **Page d'erreur**

\* 404 - Une page qui s'affiche lorsque l'utilisateur tente d'accéder à une page inexistante sur le site Web.

### **Pages d'achat**

\* panier (Panier) - Une page qui montre les articles qu'un utilisateur a ajoutés à son panier, ainsi que le prix total.

\* filtre (Filtre) - Une page qui permet aux utilisateurs de filtrer la sélection de chaises selon des critères tels que le prix, le matériau, la couleur et le style.

### **Pages de paiement**

\* checkout\_backend (Paiement) - La partie du site Web qui gère le processus de paiement, où les utilisateurs saisissent leurs informations d'expédition et de paiement.

\* success (Confirmation) - Une page qui s'affiche à l'utilisateur après avoir passé une commande avec succès.

\* cancel (Annulation) - Une page qui s'affiche à l'utilisateur si sa commande est annulée.

## Mon rôle dans le projet

Dans le cadre du projet "Games-Thrones", ma participation a été essentielle à plusieurs niveaux, couvrant à la fois le design, le développement front-end et l'implémentation de fonctionnalités clés. Voici les détails de mes contributions :

### Design :

- **Conception sur Figma** : J'ai conçu la page produit sur Figma, en m'assurant que l'interface utilisateur soit intuitive et visuellement attrayante.

### Développement front-end :

- **HTML et CSS de la page produit** : J'ai réalisé la mise en page (HTML) et le style (CSS) de la page produit, garantissant une présentation claire et cohérente des informations sur les chaises proposées.
- **HTML et CSS de la page panier** : J'ai également développé la structure HTML et le design CSS de la page panier, assurant une expérience utilisateur fluide pour la gestion des articles ajoutés.

### Fonctionnalités :

- **Ajout au panier** : J'ai implémenté la fonctionnalité du bouton "Ajouter au panier", permettant aux utilisateurs d'ajouter facilement des produits à leur panier d'achat.

Dans le cadre de notre projet "Games-Thrones", j'ai développé une fonctionnalité permettant aux utilisateurs d'ajouter des produits à leur panier via une interaction AJAX. Cette fonctionnalité est composée de deux éléments principaux : le contrôleur PHP `addProductToBasketAjaxController.php` et le script JavaScript `product.js`.

**Contrôleur PHP** : `addProductToBasketAjaxController.php`

Démarrage de la session :

```
if (session_status() === PHP_SESSION_NONE) {  
    session_start();  
}
```

Si aucune session n'est démarrée, le script démarre une nouvelle session. Cela permet de stocker les informations du panier dans la session de l'utilisateur.

#### Récupération des paramètres :

Le script récupère l'ID du produit (**id**) et la quantité (**quantity**) des paramètres de la requête GET.

#### Validation des paramètres :

```
if (empty($productId) || $quantity < 1) {  
    echo json_encode(['error' => 'Invalid product ID or quantity']);  
    exit;  
}
```

Il vérifie si l'ID du produit est valide et si la quantité est supérieure ou égale à 1. Si l'un des paramètres est invalide, il renvoie un message d'erreur au format JSON.

#### Initialisation du panier :

```
if (!isset($_SESSION['cart'])) {  
    $_SESSION['cart'] = [];  
}
```

Si le panier n'existe pas encore dans la session, il est initialisé en tant que tableau vide.

#### Ajout ou mise à jour du produit dans le panier :

```
$productInCart = array_key_exists($productId, $_SESSION['cart']);  
  
if ($productInCart) {  
    $_SESSION['cart'][$productId]['quantity'] += $quantity;  
} else {  
    $_SESSION['cart'][$productId] = [  
        'id' => $productId,  
        'quantity' => $quantity,  
    ];  
}
```

Le script vérifie si le produit est déjà dans le panier. Si oui, il met à jour la quantité. Sinon, il ajoute le produit avec la quantité spécifiée.

#### Renvoi de la réponse :

```
echo json_encode(['success' => 'Product added to cart', 'cart' => $_SESSION['cart']]);
exit;
```

Enfin, le script renvoie une réponse JSON indiquant que le produit a été ajouté au panier avec succès, ainsi que le contenu actuel du panier.

#### Script JavaScript : **product.js**

##### Attente du chargement du document :

```
jQuery(document).ready(function () {
```

Le script attend que le document soit entièrement chargé avant de s'exécuter.

##### Gestionnaire de clic pour le bouton "Ajouter au panier" :

```
jQuery("#product_basketButton").click(function (event) {
    event.preventDefault();
```

Lorsque l'utilisateur clique sur le bouton "Ajouter au panier", le comportement par défaut de l'événement est annulé.

##### Récupération de l'ID du produit et de la quantité :

```
var url = window.location.href;
var params = new URLSearchParams(url.split("?")[1]);
var id = params.get("id");
const quantity = jQuery("#product_quantity").val();
```

Le script extrait l'ID du produit de l'URL et récupère la quantité sélectionnée par l'utilisateur.

##### Envoi de la requête AJAX :

```

fetch(`/addProductToBasketAjaxController?id=${id}&quantity=${quantity}`)
  .then((response) => response.json())
  .then((data) => {
    console.log("Product added to cart:", data);
    window.location.reload();
  })
  .catch((error) => {
    console.error("Error adding product to cart:", error);
  });

```

Une requête fetch est envoyée au contrôleur PHP avec les paramètres ID et quantité. La réponse est traitée en JSON. Si l'ajout au panier est réussi, la page est rechargée pour refléter les changements. En cas d'erreur, elle est affichée dans la console.

Cette fonctionnalité permet une expérience utilisateur fluide pour l'ajout de produits au panier sans rechargement initial de la page, améliorant ainsi l'interactivité et la réactivité de l'application.

- **Calcul du panier** : J'ai développé les fonctionnalités nécessaires pour calculer automatiquement le total du panier, offrant aux utilisateurs une vue précise de leurs achats en temps réel.
- **Intégration Stripe** : J'ai intégré Stripe pour le traitement des paiements en ligne, sécurisant ainsi les transactions et facilitant les paiements pour les utilisateurs.

**Contrôleur PHP : `getCartContentsAjaxController.php`**

**Vérification et récupération du contenu du panier :**

```

if (!isset($_SESSION['cart'])) {
    echo json_encode(['cart' => []]);
    exit;
}

$cartContents = $_SESSION['cart'];
$cartJson = json_encode(['cart' => $cartContents]);

```

Le script vérifie si le panier existe dans la session. Si ce n'est pas le cas, il renvoie un panier vide. Sinon, il récupère le contenu du panier et le convertit en JSON.

#### Connexion à la base de données et récupération des détails des produits :

```
$conn = Database::connect();
$product = new crudProduct();
$cartContent = $product->getProductsByCartJson($conn, $cartJson);
echo json_encode($cartContent);
exit;
```

Le script se connecte à la base de données, crée une instance de `crudProduct`, et utilise la méthode `getProductsByCartJson` pour récupérer les détails des produits dans le panier, puis renvoie ces détails sous forme de JSON.

#### Backend de paiement : `checkout_backend.php`

##### Initialisation de Stripe et configuration des en-têtes :

```
\Stripe\Stripe::setApiKey($stripeSecretKey);
header('Content-Type: application/json');
```

Le script initialise Stripe avec la clé API secrète et configure les en-têtes de la réponse pour indiquer qu'elle sera au format JSON.

##### Démarrage de la session et récupération du contenu du panier :

```
if (session_status() === PHP_SESSION_NONE) {
    session_start();
}

$cartContents = $_SESSION['cart'];
$cartJson = json_encode(['cart' => $cartContents]);
$conn = Database::connect();
$product = new crudProduct();
$cartContent = $product->getProductsByCartJson($conn, $cartJson);
```

Le script démarre une session si nécessaire, récupère le contenu du panier, le convertit en JSON, et utilise `crudProduct` pour obtenir les détails des produits.

### Configuration de la session de paiement Stripe :

```
$cartId = uniqid();
$_SESSION['cart_id'] = $cartId;

$stripeArray = [
    'line_items' => [],
    'mode' => 'payment',
    'success_url' => 'http://localhost:8080/success?cart_id=' . $cartId,
    'cancel_url' => 'http://localhost:8080/cancel',
    'billing_address_collection' => 'required',
    'shipping_address_collection' => [
        'allowed_countries' => ['FR']
    ],
    'metadata' => [
        'cart_id' => $cartId
    ]
];
```

Le script génère un ID de panier unique et le stocke dans la session, puis configure les paramètres de la session de paiement Stripe, incluant les URLs de succès et d'annulation, ainsi que les informations de facturation et de livraison.

### Ajout des produits à la session Stripe

```
foreach ($cartContent['products'] as $key => $value) {
    $stripeArray["line_items"][$key] = [
        'price_data' => [
            'currency' => 'eur',
            'product_data' => [
                'name' => $cartContent['products'][$key]["name"],
            ],
            'unit_amount' => $cartContent['products'][$key]["price"] * 100,
        ],
        'quantity' => $cartContent['products'][$key]["quantity"],
    ];
}
```

Le script parcourt les produits du panier et les ajoute à la session Stripe avec les détails nécessaires, tels que le nom, le prix (en centimes) et la quantité.

**Création de la session de paiement et redirection :**

```
$checkout_session = \Stripe\Checkout\Session::create($stripeArray);  
  
header("HTTP/1.1 303 See Other");  
header("Location: " . $checkout_session->url);
```

Le script crée une session de paiement Stripe et redirige l'utilisateur vers l'URL de la session de paiement.

**Classe PHP : `crudProduct.class.php`**

La classe `crudProduct` contient des méthodes pour gérer les produits dans la base de données, notamment pour récupérer les détails des produits dans le panier.

**Méthode `getProductsByCartJson` :**



```

public function getProductsByCartJson($conn, $jsonData) {
    $productIds = array();
    $productQuantities = array();
    $totalProducts = 0;
    $data = json_decode($jsonData, true);

    if (isset($data['cart'])) {
        foreach ($data['cart'] as $productId => $details) {
            $productIds[] = $productId;
            $productQuantities[$productId] = $details['quantity'];
            $totalProducts += $details['quantity'];
        }
    }

    if (count($productIds) > 0) {
        $productList = array();
        $placeholders = implode(',', array_fill(0, count($productIds), '?'));
        $sql = $conn->prepare("SELECT * FROM product WHERE id IN ($placeholders)");
        $sql->execute($productIds);

        while ($row = $sql->fetch(PDO::FETCH_ASSOC)) {
            $product = new crudProduct();
            $product->readProduct($conn, $row['id']);
            $productData = $product->getProductData();
            $productData['quantity'] = $productQuantities[$product->getId()];
            $productList[] = $productData;
        }
        return array('products' => $productList, 'totalProducts' => $totalProducts);
    } else {
        return array('products' => array(), 'totalProducts' => 0);
    }
}

```

Cette méthode prend les IDs des produits du panier, les récupère de la base de données, et les retourne avec leurs quantités et d'autres détails.

En conclusion, cette fonctionnalité de paiement assure que les utilisateurs puissent facilement ajouter des produits à leur panier, voir les détails de leur panier, et procéder au paiement en utilisant Stripe. Elle intègre une gestion efficace des sessions, une interaction avec la base de données pour obtenir les détails des produits, et une intégration sécurisée avec l'API Stripe.

- **Système de commentaires** : J'ai mis en place la fonctionnalité de commentaire, permettant aux utilisateurs de laisser des avis sur les produits.

```
<section id="Commentaires">
  <div class="container">
    <h3 class="reviewTitle">Commentaires</h3>
    <form class="reviewForm" method="POST" action="">
      <div class="ratingContent">
        <div class="ratingContentTop">
          <span>Votre avis</span>
          <div class="rating-area">
            <!-- Input radio pour la notation -->
            <!-- Textarea pour le commentaire -->
            <!-- Bouton pour soumettre le formulaire -->
          </div>
        </div>
      </div>
    </form>
```

#### Fichier **product.js**

Ce fichier JavaScript gère la soumission du formulaire de manière asynchrone via AJAX.

```

jQuery(document).ready(function () {
    jQuery(".reviewForm").submit(function (event) {
        event.preventDefault();

        const rating = jQuery('input[name="rating"]:checked').val();
        const text = jQuery('textarea[name="text"]').val();

        if (!rating || !text) {
            alert("Merci de remplir tous les champs.");
            return;
        }
    })

```

```

const formData = new FormData();
formData.append("rating", rating);
formData.append("text", text);
formData.append("product_id", productId);

fetch("/sendReviewFormAjaxController", {
    method: "POST",
    body: formData,
})
    .then(response => response.json())
    .then(data => {
        if (data.success) {
            alert("Votre avis a été soumis avec succès!");
            jQuery(".reviewForm").trigger("reset");
        } else {

```

```

            alert(data.error);
        }
    })
    .catch(error => {
        console.error("Erreur lors de la soumission du formulaire:", error);
        alert("Une erreur est survenue. Veuillez réessayer plus tard.");
    })

```

### Fichier `sendReviewFormAjaxController.php`

Ce fichier PHP traite les soumissions de formulaire, insère les avis dans la base de données et renvoie une réponse JSON.

```
$rating = htmlspecialchars(filter_input(INPUT_POST, 'rating', FILTER_UNSAFE_RAW));
$text = htmlspecialchars(filter_input(INPUT_POST, 'text', FILTER_UNSAFE_RAW));
$productId = htmlspecialchars(filter_input(INPUT_POST, 'product_id', FILTER_UNSAFE_RAW));

if ($rating !== false && $text !== false && $productId !== false) {

    $review = new crudReview();

    if ($review->readReview($conn, $productId, $customerId) !== false) {
        $response['success'] = false;
        $response['error'] = 'Commentaire déjà existant.';
    } else {
        $review->createReview($conn, $productId, $customerId, $text, $rating, date('Y-m-d'));
        $response['success'] = true;
        $response['error'] = '';
    }
}
```

```
require_once $_SERVER['DOCUMENT_ROOT'] . "/controller/php/classes/Database.class.php";
require_once $_SERVER['DOCUMENT_ROOT'] . "/controller/php/classes/crudReview.class.php";

session_start();
$customerId = $_SESSION['user'];

$conn = Database::connect();

$response = [
    'success' => false,
    'error' => 'Une erreur est survenue.'
];

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
```

```

    } else {
        $response['error'] = 'Données invalides.';
    }
} else {
    $response['error'] = 'Méthode de requête non autorisée.';
}

header('Content-Type: application/json');
echo json_encode($response);
?>

```

- **Pagination des commentaires** : J'ai ajouté une pagination pour les commentaires, assurant une navigation fluide et une lecture aisée des avis des utilisateurs.

### Product.php (Affichage des commentaires)

Ce fichier est responsable de l'affichage initial des commentaires et de la gestion de la pagination.

#### Inclusion des fichiers nécessaires et récupération des commentaires :

```

<?php
require ('../public/controller/php/classes/crudReview.class.php');
require ('../public/controller/php/ajax/getReviewAjaxController.php');

$productId = $_GET["id"];
$conn = Database::connect();
$reviews = (new crudReview)->readReviewsByProductId($conn, $productId);

```

**Affichage des commentaires** : Si aucun commentaire n'est trouvé, un message approprié est affiché. Sinon, les commentaires sont affichés dans une boucle foreach.

```

if (empty($reviews)) {
    echo "<div id='noReviews'>
        <p>Il n'y pas encore de commentaire pour ce produit.</p>
    </div>";
} else {
    echo '<div class="reviews">';
    foreach ($reviews as $review) {
        // Affichage des détails du commentaire
    }
    echo '</div>';
}

```

**Bouton "Voir plus"** : Un bouton est affiché à la fin des commentaires pour permettre à l'utilisateur de charger plus de commentaires.

```

echo '<div class="toggleContButton">
    <button id="toggleButton">Voir plus</button>
</div>';

```

Product.js (Gestion du bouton "Voir plus")

Ce fichier gère le chargement des commentaires supplémentaires lorsque l'utilisateur clique sur le bouton "Voir plus".

**Initialisation de la pagination :**

jQuery

```

jQuery(document).ready(function () {
    var currentPage = 2; // Commence à partir de la deuxième page

    function fetchReviews() {
        fetch(`/getReviewAjaxController?product_id=${productId}&page=${currentPage}`)
            .then((response) => response.text())
            .then((data) => {
                jQuery(".reviews").append(data); // Ajoute les nouveaux commentaires à la fin
                if (data.length < 3) {
                    jQuery("#toggleButton").hide(); // Cache le bouton s'il n'y a plus de commentaires
                }
            })
            .catch((error) => console.error("Error fetching reviews", error));

        currentPage++; // Passe à la page suivante pour le prochain chargement
    }

    jQuery("#toggleButton").click(fetchReviews); // Écouteur de clic sur le bouton "Voir plus"
});

```

getReviewAjaxController.php (Script de récupération des commentaires via AJAX)

Ce script récupère les commentaires en fonction de l'identifiant du produit et du numéro de page.

#### Traitement PHP :

```

$product_id = intval(@$_GET["product_id"]);
$page = intval(@$_GET["page"]);
$limit = 3; // Limite de commentaires par page

$offset = ($page - 1) * $limit;

// Utilisation de la classe CRUD pour obtenir les commentaires de la base de données
$review = new crudReview();
$getReviewData = $review->getReviewsByProductId($conn, $product_id, $limit, $offset);

foreach ($getReviewData as $reviewData) {
    // Affichage des détails du commentaire
}

```

Fonctionnement global :

- Lorsque la page `product.php` est chargée, les premiers commentaires sont récupérés et affichés.
- Un bouton "Voir plus" est affiché pour permettre à l'utilisateur de charger plus de commentaires.
- Lorsque l'utilisateur clique sur ce bouton, une requête AJAX est envoyée à `getReviewAjaxController.php` avec l'identifiant du produit et le numéro de page.
- `getReviewAjaxController.php` récupère et retourne les commentaires correspondant à la page demandée.
- Les commentaires reçus via AJAX sont ajoutés à la fin de la liste existante dans `product.php`.
- Le bouton "Voir plus" est caché lorsque toutes les pages de commentaires ont été chargées.

Grâce à ces contributions, j'ai pu apporter une valeur ajoutée significative au projet "Games-Thrones", améliorant à la fois l'expérience utilisateur et les fonctionnalités de l'application.



# Développement d'un Catalogue Interactif de Pokémon “Pokedex”

## Introduction

L'importance de ce projet réside dans la création d'une plateforme interactive et ludique pour explorer les caractéristiques des Pokémon. Un tel site web permet aux utilisateurs de rechercher des Pokémon, de consulter leurs détails, et d'interagir avec une base de données riche en informations.

## Objectifs et Étendue du Projet

Objectif principal : Développer un site web interactif pour cataloguer et explorer les Pokémon, en utilisant Vue.js pour une expérience utilisateur dynamique et CSS pour une interface élégante et réactive.

## Objectifs spécifiques

1. Déterminer les objectifs, les idées et les besoins du site web.
2. Étudier les technologies de développement web adaptées à un catalogue interactif.
3. Adapter la conception technique du site (structure et fonctionnalités).
4. Développer le produit logiciel final.

## Structure Générale du Site

### Page d'Accueil :

Présente un aperçu des Pokémon avec des illustrations attractives et une barre de recherche interactive.

### Catalogue de Pokémon :

Affiche une liste de tous les Pokémon disponibles avec des filtres par type, génération, et d'autres caractéristiques spécifiques.

### Fiche Pokémon :

Contient des descriptions détaillées des Pokémon, leurs statistiques, évolutions, et autres informations pertinentes.

### Recherche :

Offre des filtres et une barre de recherche pour trouver rapidement des Pokémon spécifiques.

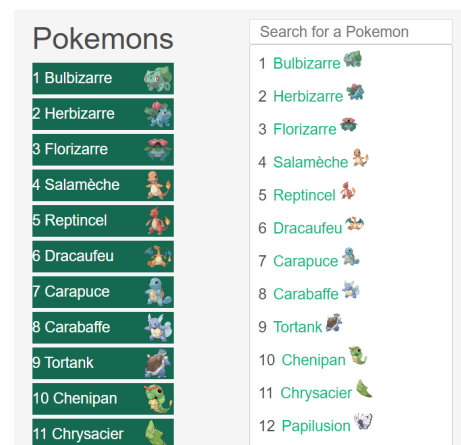
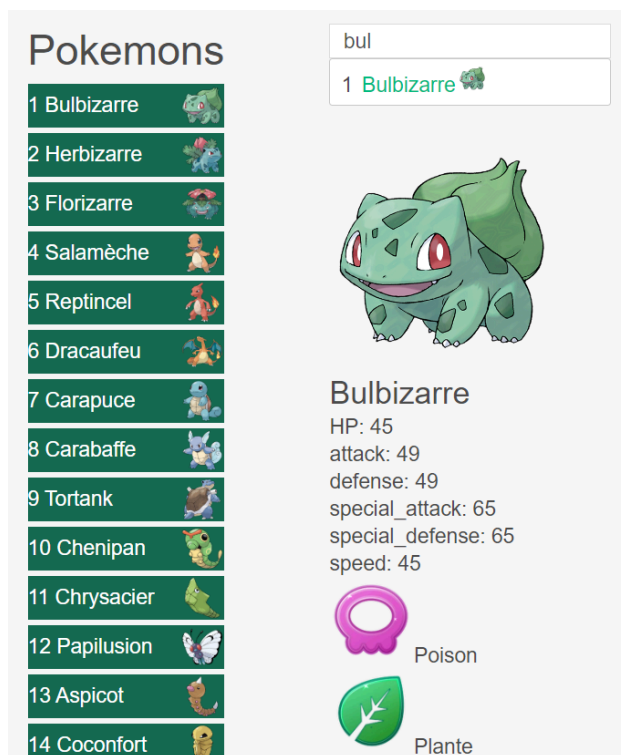
## Utilisateurs

**Visiteurs :** Peuvent accéder à la majorité des informations

## Technologies Utilisées pour le Site de Catalogue Interactif de Pokémon

### Frontend :

- ✓ Vue.js
- ✓ CSS
- ✓ JavaScript





- **Right Section**
  - Contient le composant **SearchBar**.
  - Contient le composant **PokemonDetail**.

### SearchBar.vue

- **props**
  - **pokemonData**: Array
- **data()**
  - **searchTerm**: ""
  - **selectedPokemon**: null
- **computed**
  - **filteredPokemon()**: Filtre les Pokémon en fonction du terme de recherche..
- **methods**
  - **emitOpenDetail(pokemon)**: Émet un événement vers **App.vue** pour définir le Pokémon sélectionné..
  - **selectPokemon(pokemon)**: Sélectionne un Pokémon et émet un événement.
  - **searchPokemon(term)**: Réinitialise le terme de recherche.
- **template**
  - Contient un champ de saisie pour la recherche.
  - Affiche la liste des Pokémon filtrés.
  - Gère les événements de clic pour émettre des événements de détail.

### PokemonList.vue

- **props**
  - **apiPokemons**: String
  - **searchTerm**: String
  - **pokemonData**: Array
- **methods**
  - **emitOpenDetail(pokemon)**: Émet un événement vers **App.vue** pour définir le Pokémon sélectionné.
- **computed**
  - **filteredPokemonList()**: Filtre la liste des Pokémon en fonction du terme de recherche.
- **template**
  - Displays list of Pokemon.
  - Handles click events to emit detail events.

### PokemonDetail.vue

- **props**
  - **selectedPokemon**: Array
- **template**
  - Affiche la liste des Pokémon.
  - Gère les événements de clic pour émettre des événements de détail.

## CONCLUSION

Ce projet de groupe, réalisé en un mois, a été une expérience à la fois enrichissante et formatrice. Il nous a donné un aperçu concret du quotidien d'une développeuse web.

En rétrospective, l'un des principaux défis de notre application e-commerce "Games-Thrones", spécialisée dans la vente de chaises, a été la gestion du temps. Nous avons consacré trop d'heures aux réunions, parfois au détriment de la résolution rapide des problèmes. Malgré cela, je suis fière de présenter "Games-Thrones" avec son MVP finalisé. Ce projet a été exigeant, mais nous l'avons mené à bien en équipe.

Ce projet m'a permis de travailler sur le front-end et le back-end, conformément au rôle d'une développeuse web full-stack, et de soutenir mes collègues. Pendant cette période, j'ai beaucoup appris, tant sur le plan technique que personnel. J'ai constaté de réels progrès en matière de pédagogie et de transmission de connaissances. Le travail sur moi-même, entrepris ces dernières années, a été bénéfique, m'aidant à gérer efficacement de nombreuses situations stressantes.

Je suis satisfaite de mes progrès et de ma contribution significative à la réalisation de "Games-Thrones". Toutefois, je ressens une pointe de frustration car j'aurais aimé aller plus loin en quatre semaines, une période très courte. Si possible, je continuerai à travailler sur ce projet, ayant de nombreuses idées de fonctionnalités à ajouter, ainsi que des améliorations à apporter à l'intégration.

En repensant à mes débuts en formation, où je n'avais aucune connaissance en code, et en voyant mon niveau actuel, je ressens une grande fierté. Bien que loin d'être parfaite, je suis au début de mon apprentissage. Il reste encore beaucoup à apprendre et à approfondir.

Après la formation, je continuerai à apprendre et coder. J'ai d'ailleurs plusieurs projets personnels en tête. Je vais également postuler pour des stages, voire des emplois, dans diverses entreprises, afin de débiter ma nouvelle vie professionnelle.