

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

présentée par
Alina YEFIMOVA

CONTENU



01

orthodoxetoulon.org

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

02

"Games-Thrones"

Développement d'un Site de Commerce pour la Vente de Chaises Gaming

03

"Pokedex"

Développement d'un Catalogue Interactif de Pokémon

01

orthodoxetoulon.org

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



EGLISE ORTHODOXE
DE LA SAINTE RÉSURRECTION
TOULON FONDÉE EN 1928

ПРАВОСЛАВНЫЙ ХРАМ
СВЯТОГО ВОСКРЕСЕНИЯ ХРИСТОВА
ТУЛОН ОСНОВАН В 1928

1. Développer la partie front-end d'une application web ou web mobile
2. Technologies utilisées
3. Description des principales étapes du projet
4. Difficultés rencontrées et solutions



! Objectif principal :

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

Modernisation du site web de l'Église Orthodoxe de la Sainte Résurrection

Objectif principal

- Actualiser le site web de l'Église Orthodoxe de la Sainte Résurrection pour offrir des informations générales à ses paroissiens.

Objectifs spécifiques

- Déterminer les besoins et objectifs du site web.
- Étudier les technologies de développement web.
- Adapter la conception technique du site.
- Élaborer un cahier des charges pour la création du site.
- Développer le produit logiciel.

Caractéristiques clés

- Adaptation du site aux mobiles pour une expérience utilisateur optimale.
- Utilisation du logo comme lien redirigeant vers la page d'accueil.
- Menu accessible sur l'ensemble du site, y compris sur les appareils mobiles.
- Présence d'un pied de page avec des liens vers les réseaux sociaux et les coordonnées de contact.



Histoires d'utilisateurs pour le site de l'Église Orthodoxe de la Sainte Résurrection

Membres actuels de la paroisse

- Accéder facilement aux informations sur l'église, 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.

Personnes intéressées par l'orthodoxie

- Découvrir des informations sur l'église, ses pratiques et ses événements.
- Consulter le calendrier des messes et des activités.
- Accéder à des ressources spirituelles telles que des sermons et des articles.
- En savoir plus sur les programmes et les services proposés par l'église.
- Contacter l'église pour obtenir des informations supplémentaires.

Grand public

- Trouver des informations générales sur l'église et ses activités.
- Consulter les horaires des messes et des événements spéciaux.
- Accéder à des ressources spirituelles telles que des sermons et des articles.
- Découvrir les programmes et services offerts par l'église.
- Contacter l'église pour des demandes d'informations.

Conception de site

site en ce moment

Accueil Horaires des offices à l'Eglise de la Sainte Résurrection L'histoire de la paroisse Dons pour soutenir notre église

Eglise Orthodoxe de la Sainte Résurrection du Patriarcat œcuménique, Toulon



L'Eglise de la Sainte Résurrection à Toulon (Var) existe depuis 1928.

Les offices sont célébrés quatre dimanches par mois et lors des grandes fêtes (en slovon et en français).

- **COUVATIONS**
- Horaires des offices à l'Eglise de la Sainte Résurrection
- L'histoire de la paroisse
- Offrandes et dons pour soutenir notre église

Eglise Orthodoxe de la Sainte Résurrection du Patriarcat œcuménique, Toulon



nouvelle version



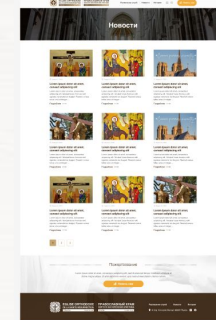
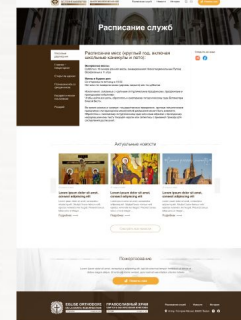
Актуальные новости

Lorem ipsum dolor sit amet, consectetur adipiscing elit



Покровительство

Lorem ipsum dolor sit amet, consectetur adipiscing elit



Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit

Актуальные новости



Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit

Frontend > Technologies utilisées

Tailwind CSS:

- Simplicité et rapidité de développement
- Flexibilité et personnalisation poussées
- Code CSS clair et facile à maintenir

```
<section id="Актуальные новости">
  <div class="mx-auto max-w-7xl px-4 my-10 md:my-16 lg:my-28">
    <div class="flex items-center gap-5 mb-14">
      <span class="flex-grow bg-gray-200 rounded h-0.5"></span>
      <h2 class="text-center text-3xl">Актуальные новости</h2>
      <span class="flex-grow bg-gray-200 rounded h-0.5"></span>
    </div>
  </div>
```

Underscoretw:

- Personnalisation Avancée :
- Intégration de Tailwind CSS :
- Méthodes Diverses d'Intégration :

La

documentation(<https://underscoretw.com/docs/custom-block>).

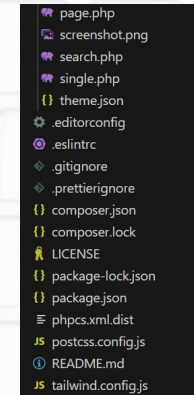
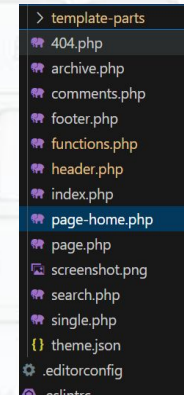
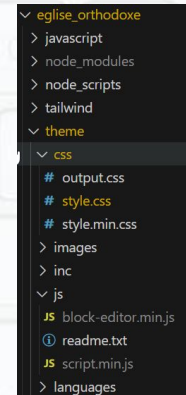
Swiper et Flowbite

Swiper

- Sliders Responsifs : Idéal pour carrousels et sliders interactifs.
- Personnalisation : Options de configuration étendues.
- Performance : Fluide et optimisé pour les mobiles.

Flowbite

- Composants UI : Collection de composants UI avec Tailwind CSS.
- Accessibilité : Composants conçus pour être accessibles.
- Documentation : Documentation détaillée pour une intégration facile.



Difficultés rencontrées et solutions

Intégrer des blocs personnalisés dans un thème Underscores : Mon expérience

Intégrer des blocs personnalisés dans un thème Underscores s'est révélé être une expérience enrichissante, mais non sans défis. En tant que développeur relativement novice en JavaScript, j'ai rencontré plusieurs obstacles au cours de ce processus.

Comprendre la structure des blocs personnalisés:

- Apprendre les concepts de fonctionnalités de bloc et de logique d'éditeur a été un défi majeur.

Maîtriser le rôle de `block.json`:

- Naviguer dans les différentes propriétés de `block.json` et les utiliser efficacement a nécessité des efforts et de la pratique.

Exploiter les ressources externes:

- La documentation officielle d'Underscores n'offre pas toujours un guide détaillé sur le développement de blocs personnalisés.



02

"Games-Thrones"

Développement d'un Site de Commerce pour
la Vente de Chaises Gaming



- 1.Objectifs
- 2.Interface utilisateur
- 3.Maquette
- 4.Base de données
- 5.Front-end
- 6.Back-end



Objectifs

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.

Objectifs spécifiques :

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

Frontend :

- HTML5
- CSS3
- Bibliothèques
JavaScript : jQuery

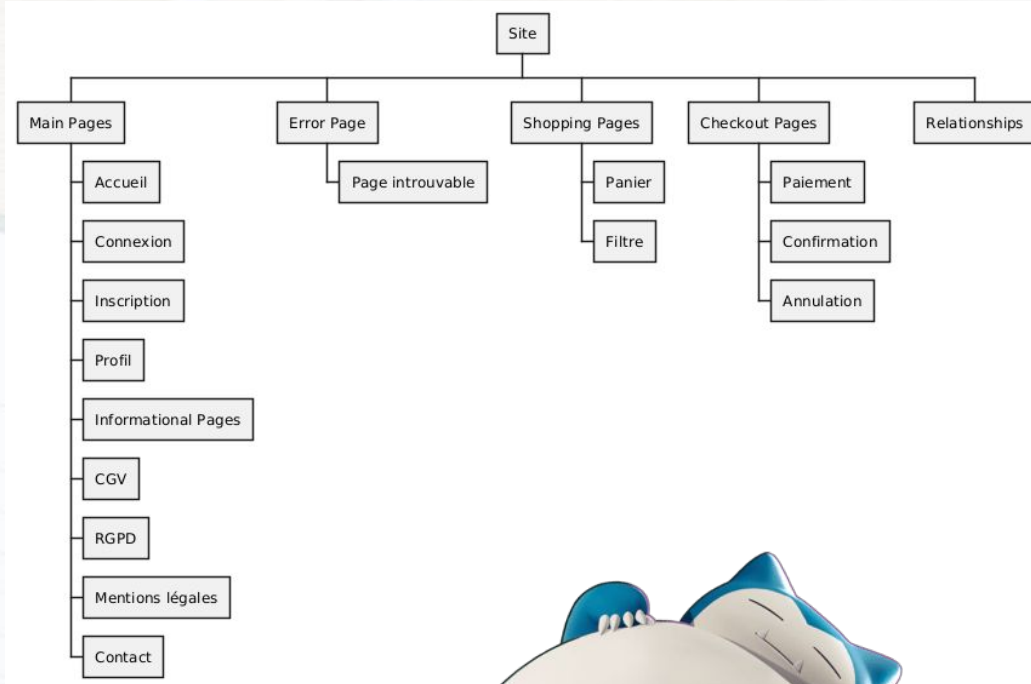
Backend :

- php
- Altorouter

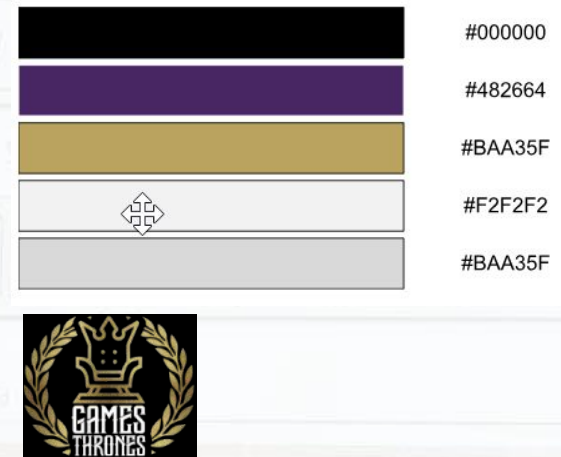


Développement de l'interface utilisateur

Arborescence

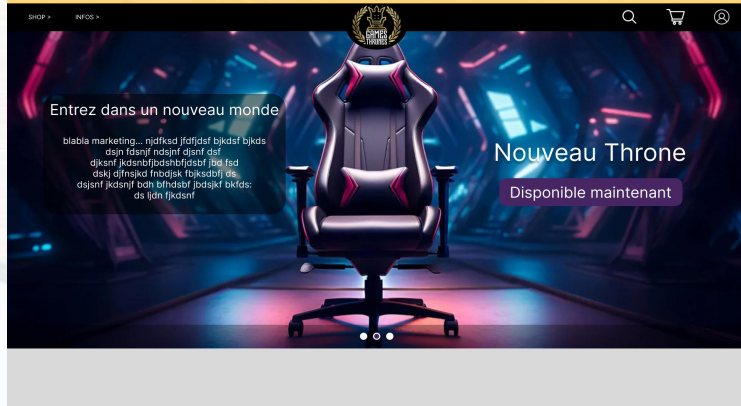


Charte graphique et logo

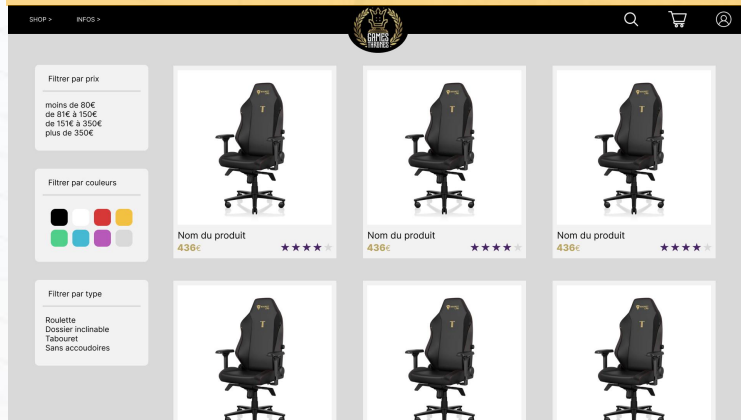


Maquette version PS

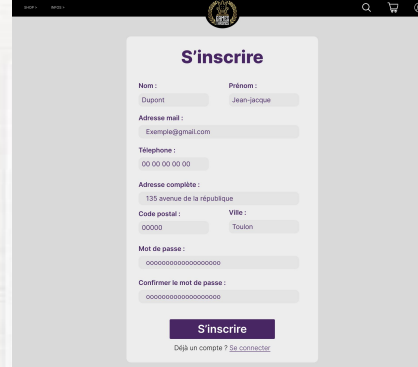
Page d'Accueil



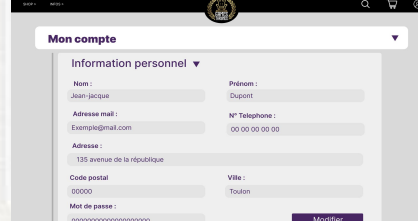
Filtre de Produits



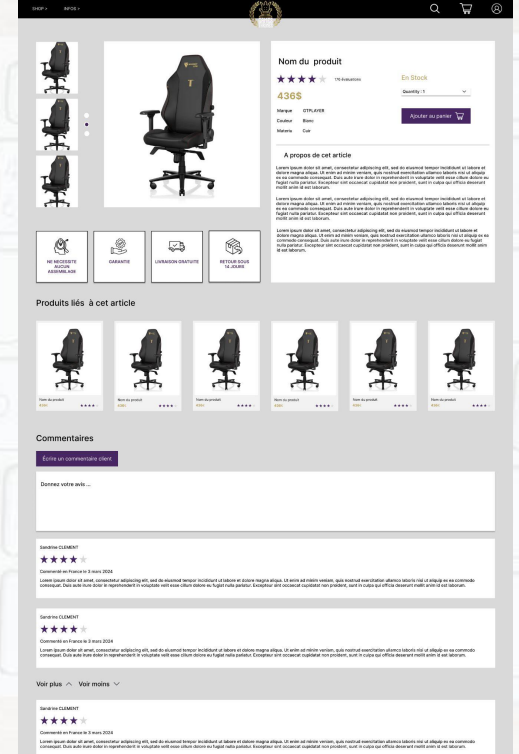
Inscription



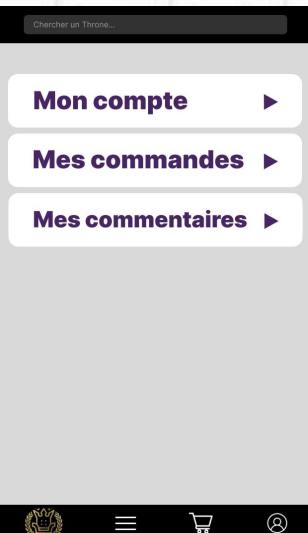
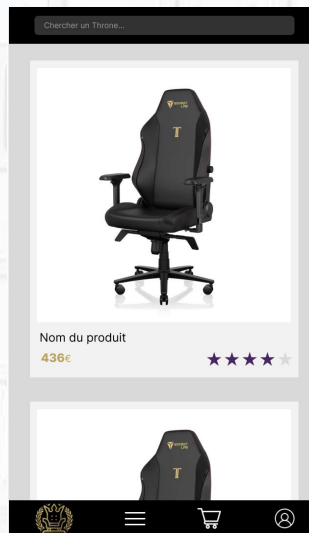
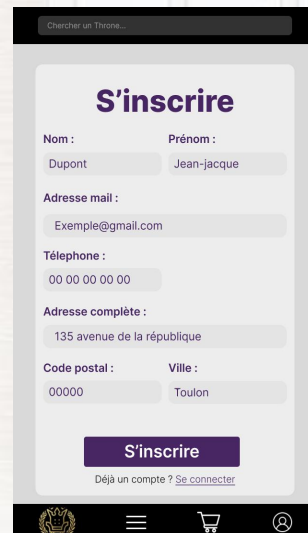
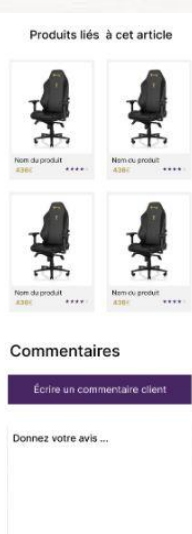
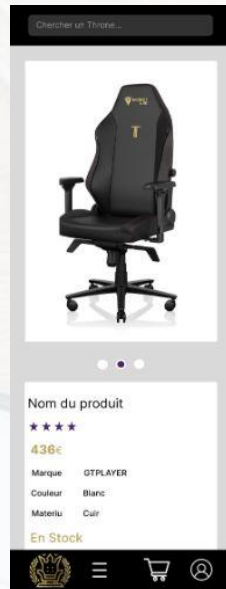
Page de profil



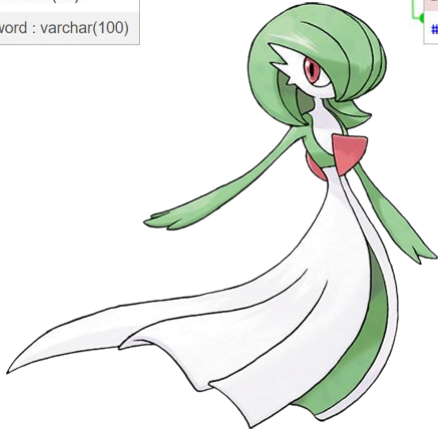
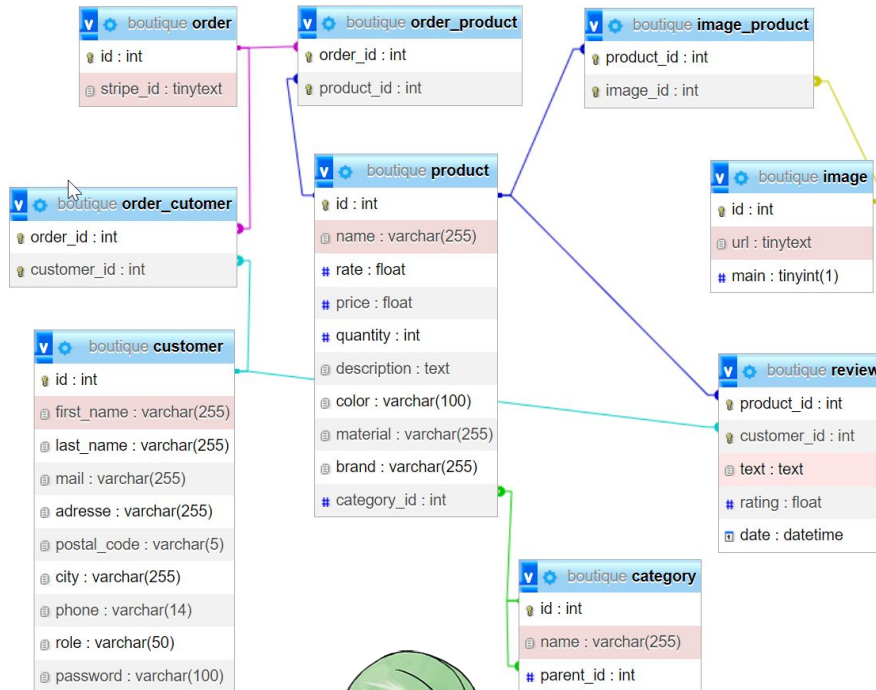
Page de Produit



Moquette mobile version



Base de données



Client-Commande (1:N): Un client peut passer plusieurs commandes, mais une commande est associée à un seul client (clé étrangère **customer_id** dans **Commande** référençant la clé primaire **customer_id** dans **Client**).

Commande-Produit (M:N): Une commande peut contenir plusieurs produits, et un produit peut apparaître dans plusieurs commandes (table relationnelle **Commande_produit** stockant les **order_id** de **Commande** et les **product_id** de **Produit**).

Produit-Image (1:N): Un produit peut avoir plusieurs images associées, mais une image appartient à un seul produit (table relationnelle **Image_produit** stockant les **product_id** de **Produit** et les **image_id** de **Image**).

Front-end

HTML - Page de produit

une répartition des principales sections de page de produit:

- section "description": cette section affiche des détails sur le produit, y compris des images, des avantages, une description et des options d'achat.

```
<section id="description">
</section>
```
- **Côté gauche** : affiche des photos du produit, des avantages et des informations sur le produit.
 - Les photos de produits peuvent utiliser une fonctionnalité de curseur
 - Les avantages sont affichés sous forme d'icônes avec des descriptions textuelles.
 - Les informations sur le produit incluent le prix, la note, la marque, la couleur, le matériau et la quantité.

```
<div class="leftSide">
  <div class="productPhotos">
    <div class="slider">
      <ul ...
      </ul>
    </div>
    <div class="imageActuelle">
      
    </div>
    <div class="sliderDots"></div>
  </div>
  <div class="productBenefits">...
</div>
```



```
<div class="rightSide">
  <div class="descriptionTop">...
  </div>
  <div class="descriptionBottom">
    <h3 class="descriptionTitle">En savoir plus :</h3>
    <div>
      <p><?= $product["description"] ?></p>
    </div>
  </div>
</div>
```

- **Côté droit** : affiche le titre du produit, la description et le bouton « Ajouter au panier ».
- **Commentaires**: Cette section permet aux utilisateurs de laisser des avis et affiche les avis existants sur le produit.

```
<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 type="radio" id="star-5" name="rating" value="5">
            <label for="star-5" title="Evaluation «5»"></label>
            <input type="radio" id="star-4" name="rating" value="4">
            <label for="star-4" title="Evaluation «4»"></label>
            <input type="radio" id="star-3" name="rating" value="3">
            <label for="star-3" title="Evaluation «3»"></label>
            <input type="radio" id="star-2" name="rating" value="2">
            <label for="star-2" title="Evaluation «2»"></label>
            <input type="radio" id="star-1" name="rating" value="1">
            <label for="star-1" title="Evaluation «1»"></label>
          </div>
        </div>
        <textarea name="text"></textarea>
        <button class="reviewFormBtn" type="submit">Écrire un commentaire</button>
      </div>
    </form>
  </div>
```

Front-end

HTML - Page de produit

Éléments et fonctionnalités clés :

<h1> : désignent le titre du produit

```
<h1 class="productTitle"><?= $product["name"] ?></h1>
```

**** : affichez des photos de produits et des icônes d'avantages.

```

```

<p> : présentent des descriptions de produits, des textes sur les avantages et d'autres informations textuelles.

```
<p class="benefitsItemText">GARANTIE</p>
```

<input> : permettent aux utilisateurs de sélectionner une quantité et éventuellement de soumettre des avis.

```
<label for="quantity">Quantité</label><br>
<input id="product_quantity" type="number" class="quantity" name="quantity" min="1"
  max="<?= $product["quantity"] ?>" value="1">
```

<textarea> : fournit une zone de texte permettant aux utilisateurs de rédiger leur avis.

```
<textarea name="text"></textarea>
```

<form> : définit la structure de soumission des données d'évaluation.

```
<form class="reviewForm" method="POST" action="">
```

**** : consiste à regrouper les éléments en ligne (texte, images, liens, etc.)

```
<span>Ajouter au panier</span> 
```

<div> : créer des boîtes invisibles qui maintiennent les éléments liés ensemble.

```
<div class="rating-area">|...
</div>
```

<button> : déclenchez des actions telles que l'ajout du produit au panier, la soumission d'un avis et potentiellement l'affichage de plus d'avis.

```
<button class="reviewFormBtn" type="submit">Écrire un commentaire</button>
```

Front-end

CSS - Page de produit

```
::after,  
::before {  
  box-sizing: border-box;  
}  
ul,  
li,  
a {  
  list-style: none;  
  padding: 0;  
  margin: 0;  
  text-decoration: none;  
}  
body {  
  margin: 0;  
  padding: 0;  
  background-color: #rgb(230, 230, 230);  
  position: relative;  
  min-height: 100vh;  
  max-width: 100%;  
  font-family: "Roboto", sans-serif;  
}
```

```
button {  
  color: #ffff;  
  background-color: #482664;  
}  
  
.container {  
  max-width: 90vw;  
  margin: 3em auto;  
}
```

Styles globaux :

::after, ::before : Ceux-ci ciblent tous les pseudo-éléments (::after et ::before) de n'importe quel élément (*).

box-sizing: border-box pour eux, garantissant que la largeur et la hauteur de l'élément incluent le remplissage et la bordure.

ul, li, a : réinitialise les styles des listes non ordonnées.

stylisé l'élément **body** sans marges, sans remplissage, avec un arrière-plan gris clair et définit la famille de polices sur "Roboto", sans empattement.

Product Benefits

```
.benefitsItem:hover {  
  transform: scale(1.05);  
}  
  
.benefitsItemImg {  
  width: 40%;  
  padding-bottom: 1em;  
  height: auto;  
}  
  
.benefitsItemText {  
  object-fit: cover;  
  font-size: 1em;  
  font-weight: 500;  
}  
/*** End Benefits ***/
```

```
/*** Start Benefits ***/  
  
.productBenefits {  
  display: grid;  
  gap: 2em;  
  grid-template-columns: repeat(3, minmax(0, 1fr));  
  padding: 2em 0 0 0;  
}  
  
.benefitsItem {  
  background-color: #ffffff;  
  padding: 10px;  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
  justify-content: space-around;  
  text-align: center;  
  cursor: pointer;  
  transition: transform 0.5s;  
}
```

Media queries - CSS

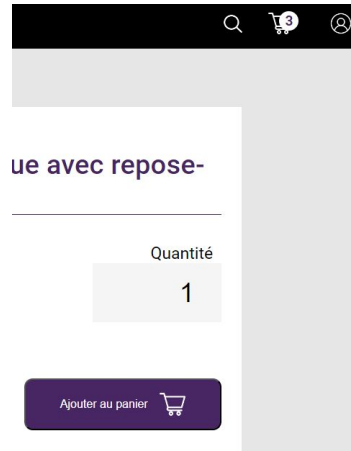
```
@media (max-width: 768px) { ...  
}  
  
@media (max-width: 425px) { ...  
}
```


Front-end

JS - Page de produit

Notre site utilise **jQuery** pour ajouter des fonctionnalités interactives à une pag web

jQuery est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web2.



```
/****** add to the cart *****/
jQuery(document).ready(function () {
  jQuery("#product_basketButton").click(function (event) {
    event.preventDefault();

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

    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);
      });
  });
});
```

```
/****** "Voir plus" Button *****/
jQuery(document).ready(function () {
  var currentPage = 2;

  function fetchReviews() {
    fetch(
      `/getReviewAjaxController?product_id=${productId}&page=${currentPage}`
    )
      .then((response) => response.text())
      .then((data) => {
        jQuery(".reviews").append(data);
        console.log(data.length);
        if (data.length < 3) {
          jQuery("#toggleButton").hide();
        }
      })
      .catch((error) => console.error("Error fetching reviews", error));
    currentPage++;
  }

  jQuery("#toggleButton").click(fetchReviews);
});
```



Front-end

JS - Page de produit

Commentaires

Commentaires

Votre avis ★ ★ ★ ★ ★

Lequel

Écrire un commentaire

ALINA alina
#5
Commenté en France le 22/05/2024
Alina User 7

Name_Test Last_Name_Test
#5
Commenté en France le 15/05/2024
Super chaise très confortable facile à monter, les coussins de tête et des lombaires sont très agréables, très jolie rendu et j'adore la matière effet simili cuir ! Inclinaison et le repose pied est également agréable ! Je recommande sans hésiter !



```
/****** form to send a post request *****/

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("Error submitting form:", error);
        alert("Une erreur est survenue. Veuillez réessayer plus tard.");
      });
  });
});
```

Back-end

Routing



```
$router = new AltoRouter();
```

Initialisation du routeur :

```
require '../vendor/autoload.php';
```

Définition des routes :

```
$router->map('GET', '/', '/home', 'accueil');
$router->map('GET', '/connexion', '/signIn', 'connexion');
$router->map('GET', '/inscription', '/signUp', 'inscription');
$router->map('GET', '/profil', '/profil', 'profil');
$router->map('GET', '/cgv', '/cgv', 'cgv');
$router->map('GET', '/rgpd', '/rgpd', 'rgpd');
$router->map('GET', '/mention', '/mention', 'mention');
$router->map('GET', '/contact', '/contact', 'contact');
$router->map('GET', '/produit', '/product', 'produit');
$router->map('GET', '/404', '/404', '404');
$router->map('GET', '/panier', '/basket', 'panier');
$router->map('GET', '/filtre', '/filters', 'filtre');
$router->map('GET', '/checkout', '/checkout_backend', 'checkout_backend');
$router->map('GET', '/success', '/success', 'success');
$router->map('GET', '/cancel', '/cancel', 'cancel');
```

Autoload des classes :

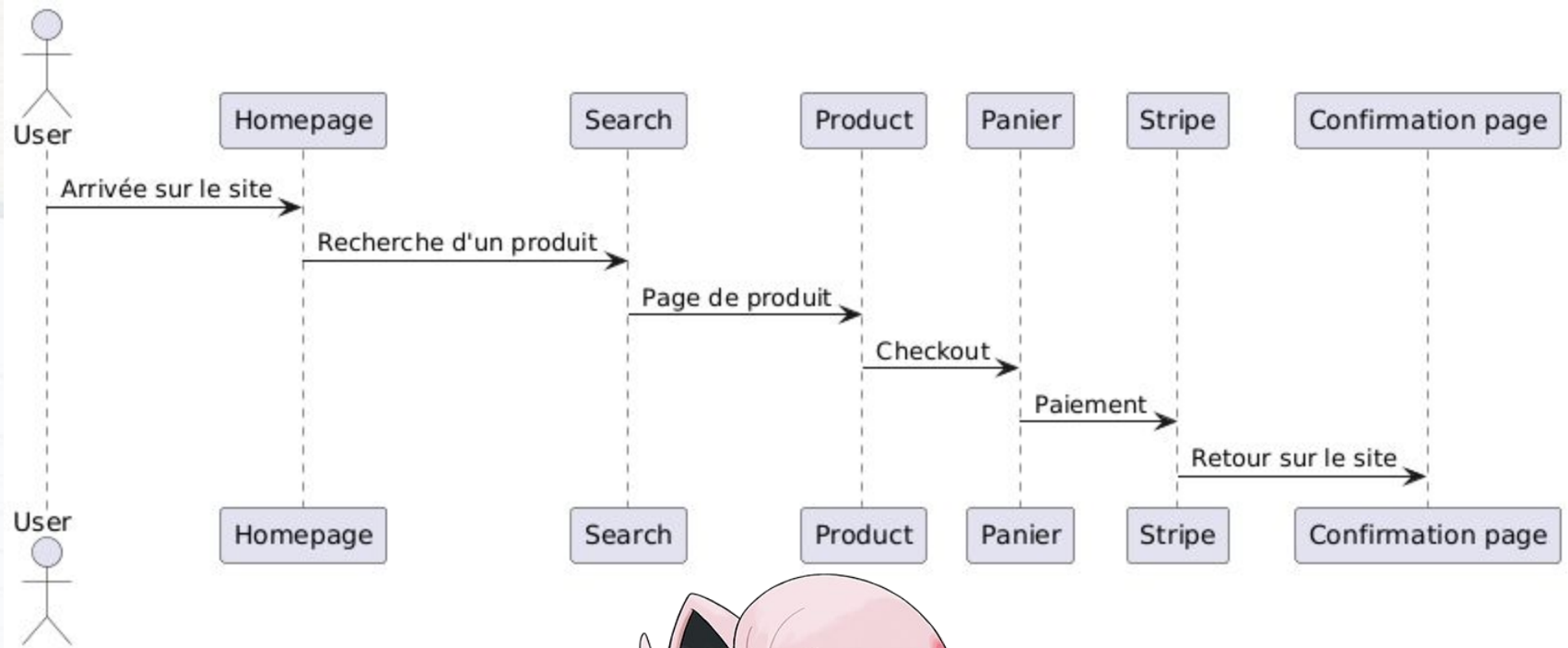
```
function my_autoloader($class)
{
    include 'controller/php/classes/' . $class . '.class.php';
}
```

“**Games - Thrones**” code PHP configure un routeur (**AltoRouter**) pour gérer les différentes URL et actions sur un site web. Il utilise des fichiers de templates pour structurer les pages HTML et peut également gérer des appels **AJAX** et des actions de formulaire.

Traitement des routes :

```
if (is_array($match)) {
    // Handle routes that send JSON
    if (str_contains($match["name"], "Ajax")) {
        if (is_callable($match['target'])) {
            call_user_func_array($match['target'], $match['params']);
        } else {
            $params = $match['params'];
            require "../src/{$match['target']}.php";
        }
    } else if (str_contains($match["name"], "_backend")) {
        if (is_callable($match['target'])) {
            call_user_func_array($match['target'], $match['params']);
        } else {
            $params = $match['params'];
            require "../src/{$match['target']}.php";
        }
    } else {
        // Handle routes that send HTML
        require '../templates/header.php';
        if (is_callable($match['target'])) {
            call_user_func_array($match['target'], $match['params']);
        } else {
            $params = $match['params'];
            require "../src/{$match['target']}.php";
        }
        require '../templates/footer.php';
        echo "</html>";
    }
}
```

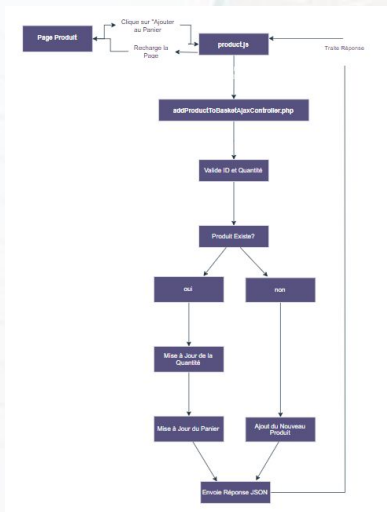

Back-end > Processus d'achat



Back-end

L'Ajout d'un Produit au Panier

- Reçoit les paramètres de la requête (ID du produit et quantité).
- Valide les paramètres.
- Vérifie si le produit est déjà dans le panier.
- Met à jour la quantité si le produit est déjà présent.
- Ajoute un nouveau produit si ce n'est pas le cas.
- Renvoie une réponse JSON avec le statut de l'opération.



```
// Start a session if it hasn't already been started
if (session_status() === PHP_SESSION_NONE) {
    session_start();
}

// Get product ID and quantity from the request parameters
$productId = isset($_GET['id']) ? intval($_GET['id']) : null;
$quantity = isset($_GET['quantity']) ? intval($_GET['quantity']) : 1;

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

// Initialize the cart if it doesn't exist in the session
if (!isset($_SESSION['cart'])) {
    $_SESSION['cart'] = [];
}

// Check if the product already exists in the cart
$productInCart = array_key_exists($productId, $_SESSION['cart']);

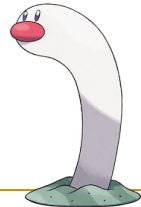
// If the product exists, update the quantity
if ($productInCart) {
    $_SESSION['cart'][$productId]['quantity'] += $quantity;
    // Otherwise, add the product to the cart
} else {
    $_SESSION['cart'][$productId] = [
        'id' => $productId,
        'quantity' => $quantity,
    ];
}

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

Back-end

Paiement

stripe



Les étapes de l'exécution du script PHP

- Chargement des Dépendances
- Configuration de Stripe
- Démarrage de la Session
- Récupération du Panier
- Connexion à la Base de Données.
- Création de l'Objet Produit
- Récupération des Produits du Panier
- Création d'un ID de Panier Unique
- Préparation des Données Stripe
- Création de la Session de Paiement Stripe
- Redirection vers Stripe

```
<?php

require_once $_SERVER['DOCUMENT_ROOT'] . '/../vendor/autoload.php';
require_once $_SERVER['DOCUMENT_ROOT'] . '/../secrets.php';
require_once $_SERVER['DOCUMENT_ROOT'] . '/controller/php/classes/Database.class.php';
require_once $_SERVER['DOCUMENT_ROOT'] . '/controller/php/classes/crudProduct.class.php';

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

$YOUR_DOMAIN = 'http://localhost:8080';

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

// Get the cart contents from the session
$cartContents = $_SESSION['cart'];

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

$conn = Database::connect();

$product = new crudProduct();
$cartContent = $product->getProductsByCartJson($conn, $cartJson);

/*var_dump($_SESSION);
die();*/

$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']
    ]
];
```

03

pokedex

Développement d'un Catalogue Interactif de Pokémon

1. Développer la partie front-end d'une application web ou web mobile
2. Structure Générale
3. Description des principales étapes du projet



Objectifs

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).

Structure Générale

Technologies Utilisées

Frontend :

Vue.js
CSS
JavaScript

Pokemons

- 1 Bulbizarre
- 2 Herbizarre
- 3 Florizarre
- 4 Salamèche
- 5 Reptincel
- 6 Dracaufeu
- 7 Carapuce
- 8 Carabaffe
- 9 Tortank
- 10 Chenipan
- 11 Chrysacier
- 12 Papilusion
- 13 Aspicot
- 14 Coconfort

bul

- 1 Bulbizarre



Bulbizarre

HP: 45
attack: 49
defense: 49
special_attack: 65
special_defense: 65
speed: 45



Poison



Plante

Pokemons

- 1 Bulbizarre
- 2 Herbizarre
- 3 Florizarre
- 4 Salamèche
- 5 Reptincel
- 6 Dracaufeu
- 7 Carapuce
- 8 Carabaffe
- 9 Tortank
- 10 Chenipan
- 11 Chrysacier

Search for a Pokemon

- 1 Bulbizarre
- 2 Herbizarre
- 3 Florizarre
- 4 Salamèche
- 5 Reptincel
- 6 Dracaufeu
- 7 Carapuce
- 8 Carabaffe
- 9 Tortank
- 10 Chenipan
- 11 Chrysacier
- 12 Papilusion

Pourquoi Vue.JS ?

Vue.js a été choisi pour ce projet en raison de sa simplicité, de sa flexibilité et de son adoption croissante par la communauté. Ces qualités le rendent idéal pour développer un catalogue interactif de Pokémon qui est à la fois performant et agréable à utiliser.

Simplicité et Courbe d'Apprentissage :

- Plus facile à apprendre pour les développeurs débutants et intermédiaires.
- Documentation claire et bien structurée, avec de nombreux exemples.

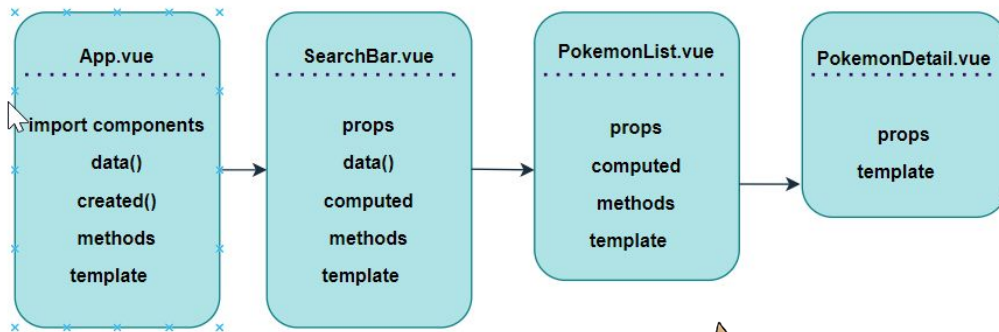
Performance et Flexibilité :

- Léger et performant, particulièrement adapté pour des projets de taille moyenne.
- Flexible et facile à intégrer avec d'autres bibliothèques ou projets existants.

Adoption et Écosystème :

- Forte adoption par la communauté, particulièrement pour les startups et les projets individuels.
- Écosystème riche avec des plugins et des outils communautaires.

Vue.JS > Code



```
<script setup>
import SearchBar from "../components/SearchBar.vue";
import PokemonList from "../components/PokemonList.vue";
import PokemonDetail from "../components/PokemonDetail.vue";
</script>

<script>
export default {
  components: {
    SearchBar,
    PokemonList,
    PokemonDetail,
  },
  data() {
    return {
      apiPokemons: "https://pokebuildapi.fr/api/v1/pokemon/limit/100",
      searchTerm: "",
      selectedPokemon: null,
    };
  }
};
```

```
<template>
<div class="container">
  <div class="left">
    <h1>Pokemons</h1>

    <PokemonList
      :apiPokemons="apiPokemons"
      :searchTerm="searchTerm"
      :pokemonData="pokemonData"
      @open-detail="openDetail"
    />
  </div>
  <div class="right">
    <div>
```



```
<style>
.container {
  display: flex;
  flex-direction: row;
  gap: 9.625rem;
  width: 80%;
}
.left {
  width: 40%;
}
.right {
  width: 60%;
}
@media (min-width: 1024px) {
}
</style>
```


**Merci de votre
temps 😊**

