

Dossier de projet

Projet réalisé dans le cadre de la présentation au

Titre Professionnel Développeur Web et Web Mobile
présenté par
Lebrun Ismael

Sommaire

- **Projet Principal : games-thrones**

- Contexte du dossier de projet
- Objectifs des projets présentés
- Aperçu des technologies utilisées
- Objectifs du projet
- Fonctionnalités principales
- Compétences couvertes
 - Installer et configurer son environnement de travail en fonction du projet web
 - Maquetter des interfaces utilisateur web et web mobile
 - Développer une interface utilisateur web dynamique
 - Mettre en place une base de données relationnelle
 - Développer des composants d'accès aux données
 - Développer des composants métier côté serveur
 - Développer la partie back-end d'une application web et web mobile
- Technologies utilisées
- Défis rencontrés et solutions apportées
- Résultats et impact

- **Projet Complémentaire : Voyage-blog**

- Introduction
- Objectifs du projet
- Fonctionnalités principales
- Compétences couvertes
 - Installer et configurer son environnement de travail en fonction du projet web
 - Réaliser une interface utilisateur statique web et web mobile
- Technologies utilisées
- Défis rencontrés et solutions apportées
- Résultats et impact

games-thrones

Introduction

Le projet games-thrones est une boutique en ligne spécialisée dans la vente de chaises et de fauteuils gaming. Ce projet est réalisé par une équipe de quatre développeurs web en formation dans le cadre du cursus Développeur Web et Web Mobile (DWWM). L'objectif principal est de maximiser l'utilisation de JavaScript pour terminer le projet en 6 semaines.

En travaillant sur ce projet, nous avons non seulement amélioré nos compétences techniques en développement web, mais également renforcé notre capacité à travailler en équipe et à coordonner nos efforts. Cet exercice a été conçu pour simuler un environnement de travail professionnel, nous permettant ainsi d'acquérir une expérience précieuse en gestion de projets et en collaboration.

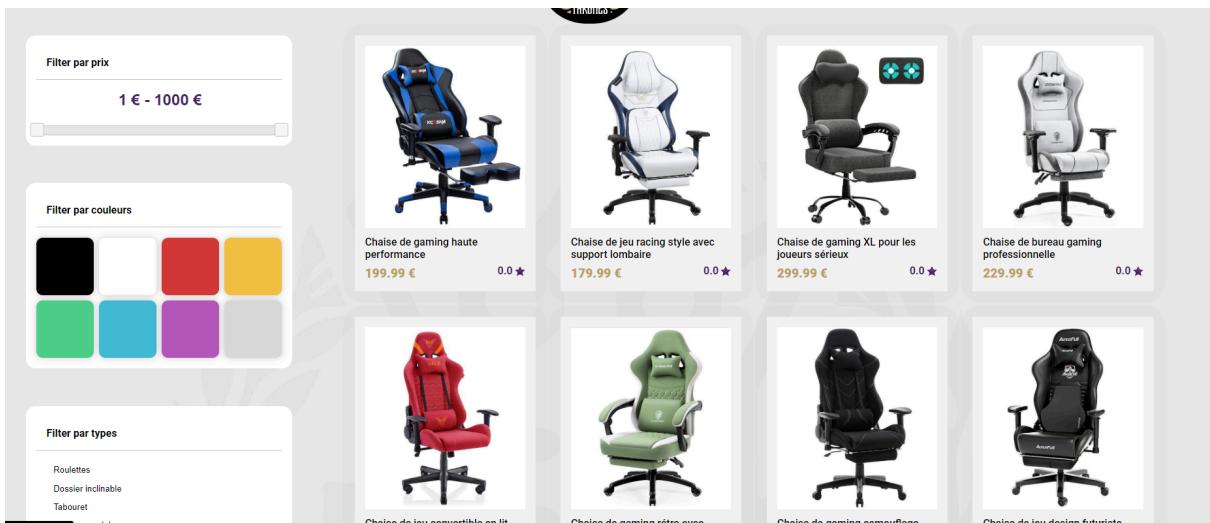
La réalisation de ce projet nous a permis d'augmenter significativement nos compétences en JavaScript, d'améliorer notre approche du développement back-end orienté objet avec PHP, et de créer des interfaces utilisateur belles et fonctionnelles grâce à CSS et JavaScript. Au final, games-thrones a servi pour notre développement professionnel, nous préparant à relever des défis encore plus complexes dans notre future carrière de développeur web tout en réalisant un projet pertinent et réutilisable.

Objectifs du projet

- **Développement technique** : Utiliser au maximum JavaScript, PHP (orienté objet), et CSS pour créer une boutique en ligne fonctionnelle.
- **Compétences pratiques** : Acquérir de l'expérience en gestion de projets et en collaboration d'équipe.
- **Fonctionnalité utilisateur** : Créer une plateforme sur laquelle les utilisateurs peuvent acheter des chaises et des fauteuils de manière sécurisée.

Fonctionnalités principales

- **Catalogue de produits** : Affichage des chaises et fauteuils, filtres de recherche.



- **Panier d'achat** : Ajout et suppression de produits, calcul des totaux.

Chaise de gaming haute performance

199.99 €

0★

Marque: ProGamer

Couleur: Vert

Materiaux: Mesh respirant

Quantité:

Ajouter au panier

permet d'ajouter au panier le nombre de produit souhaité



voir le nombre de produit dans la panier partout grâce à ce bouton qui redirige sur le panier

Mon panier

Produit	Quantité	Total
Chaise de gaming haute performance Vert 199.99 €	<input type="button" value="-"/> 2 <input type="button" value="+"/>	399.98 €
		Total 399.98€
		Passer la commande

- **Système de paiement** : Processus de commande sécurisé à l'aide l'API stripe.

Chaise de gaming haute performance

399,98 €

Qté 2, 199,99 € chacun

Informations de livraison

E-mail

Adresse de livraison

Nom complet
France
Adresse

[Saisir l'adresse manuellement](#)

Données de paiement

Informations de la carte

1234 1234 1234 1234	
MM / AA	CVC

Les informations de facturation sont identiques aux informations de livraison

Enregistrer mes informations en toute sécurité pour le paiement en un clic

Saisissez votre numéro de téléphone pour créer un compte
Link et vérifier votre achetez plus rapidement depuis la ligne

- **Gestion utilisateur** : Inscription, connexion, et gestion de profil.



Se connecter

Email

Mot de passe

[Vous n'avez pas de compte ? Inscrivez-vous](#)



S'inscrire

Nom

Prénom

Email

Téléphone

Adresse

Code postal

Ville

Mot de passe

Confirmer le mot de passe

[S'inscrire](#)

Vous avez déjà un compte ? [Connectez-vous](#)

The screenshot shows a user profile page with the following details:

Nom	Prénom
Lebrun	Ismael

Email	Téléphone
ismael.lebrun@admin.com	11 11 11 11 11

Adresse	Ville
294 chemin de sauvebonne	Sollès-pont

Code postal	Mot de passe
83210	modifier

Buttons at the bottom:

- [Se déconnecter](#)
- [Changer le mot de passe](#)

- **Back office :** Gestion des produits et utilisateurs.

The Back Office interface displays a list of 7 products:

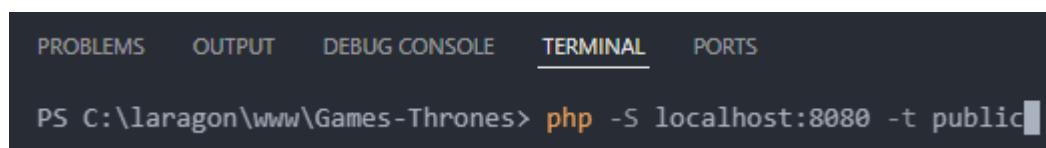
Produit	Description	Prix	Action
2 Chaise gaming en or massif	200000	€ 5,20	Modifier X
3 Chaise de gaming haute performance	199,99	€ 20	Modifier X
4 Chaise de jeu racing style avec support lombaire	179,99	€ 20	Modifier X
5 Chaise de gaming XL pour les joueurs sérieux	299,99	€ 20	Modifier X
6 Chaise de bureau gaming professionnelle	229,99	€ 20	Modifier X
7 Chaise de jeu convertible en lit	349,99	€ 20	Modifier X

Page uniquement accessible si l'utilisateur à le rôle "admin".

Compétences couvertes

1. Installer et configurer son environnement de travail en fonction du projet web

Nous avons configuré notre environnement de développement avec les outils nécessaires, notamment un serveur local (Laragon), un éditeur de code (Visual Studio Code), Github. ([voir annexe 1 et 2](#)). Une fois Laragon installé, nous avons modifié le serveur en utilisant la commande `php -S localhost:8080 -t public` dans le terminal de Visual Studio Code pour lancer notre projet sur le port 8080 et définir le dossier "public" comme étant de dossier racine du projet.

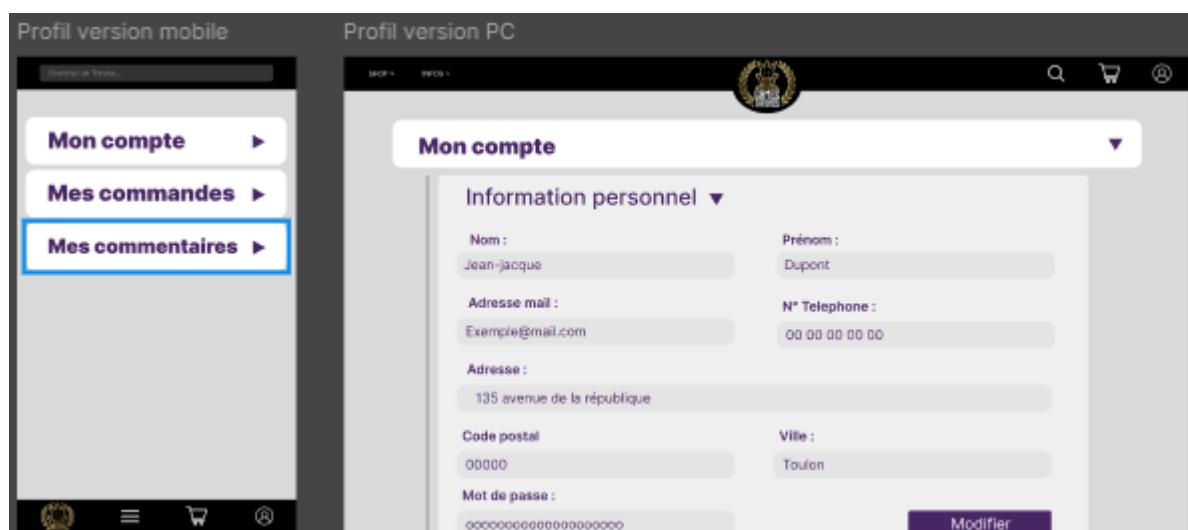


```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS  
PS C:\laragon\www\Games-Thrones> php -S localhost:8080 -t public
```

Chaque membre de l'équipe a configuré son environnement pour garantir la compatibilité et l'efficacité du travail collaboratif.

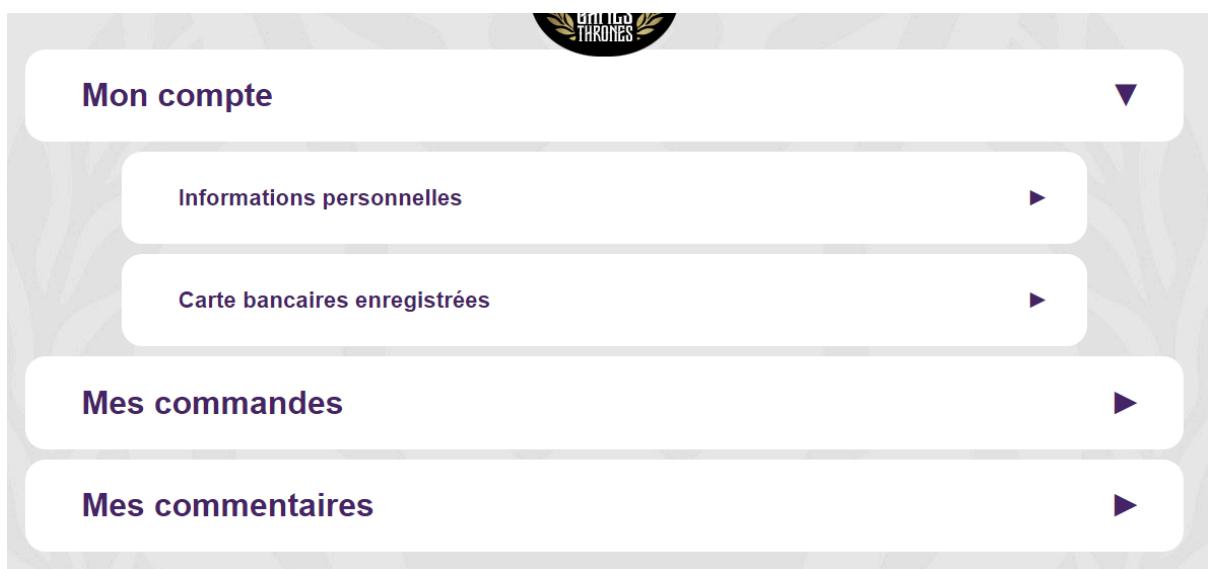
2. Maquetter des interfaces utilisateur web et web mobile

Avant de commencer le développement, nous avons créé des maquettes des interfaces utilisateur à l'aide de Figma. Ces maquettes ont servi de guide visuel tout au long du projet, assurant une expérience utilisateur cohérente et intuitive.



3. Réaliser une interface statique web et web mobile

Lorsque la maquette est terminée, et qu'on s'est réparti les tâches. On a commencé à travailler sur nos pages respectives. J'ai fait la page de profil. J'y ai créé les formulaire, les sections et div nécessaire pour correspondre à la maquette. Ensuite j'ai utilisé du Javascript pour les accordéons.



3. Développer une interface utilisateur web dynamique

Nous avons développé une interface utilisateur dynamique en utilisant HTML, CSS, et JavaScript. L'utilisation de JavaScript a permis de créer des interactions dynamiques telles que le remplissage automatique des informations de l'utilisateur et la possibilité de le modifier directement depuis sa page de profil.

Mon compte



Informations personnelles

Nom

Lebrun

Prénom

Ismael

Email

ismael.lebrun@admin.com

Téléphone

11 11 11 11 11

Adresse

294 chemin de sauvebonne

Code postal

83210

Ville

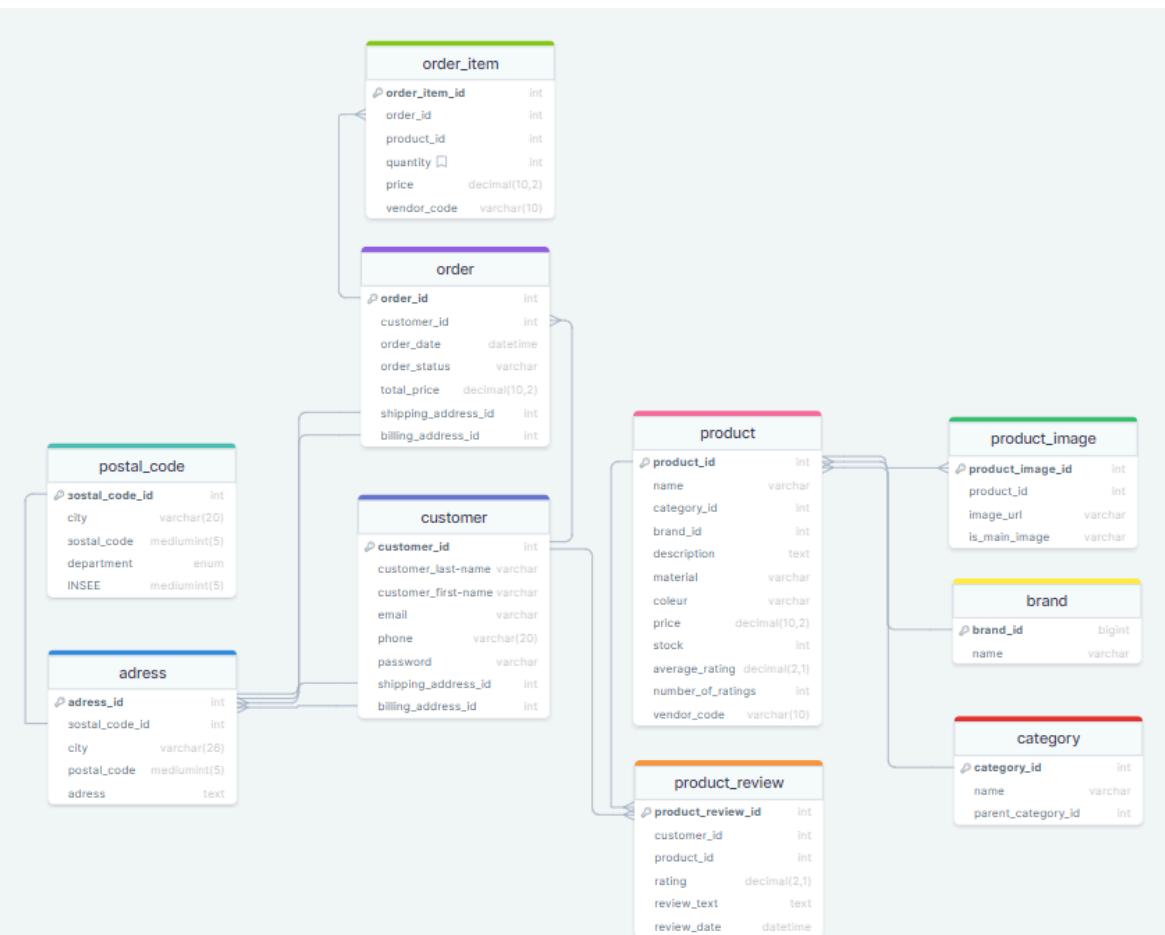
Sollies-pont

Mot de passe

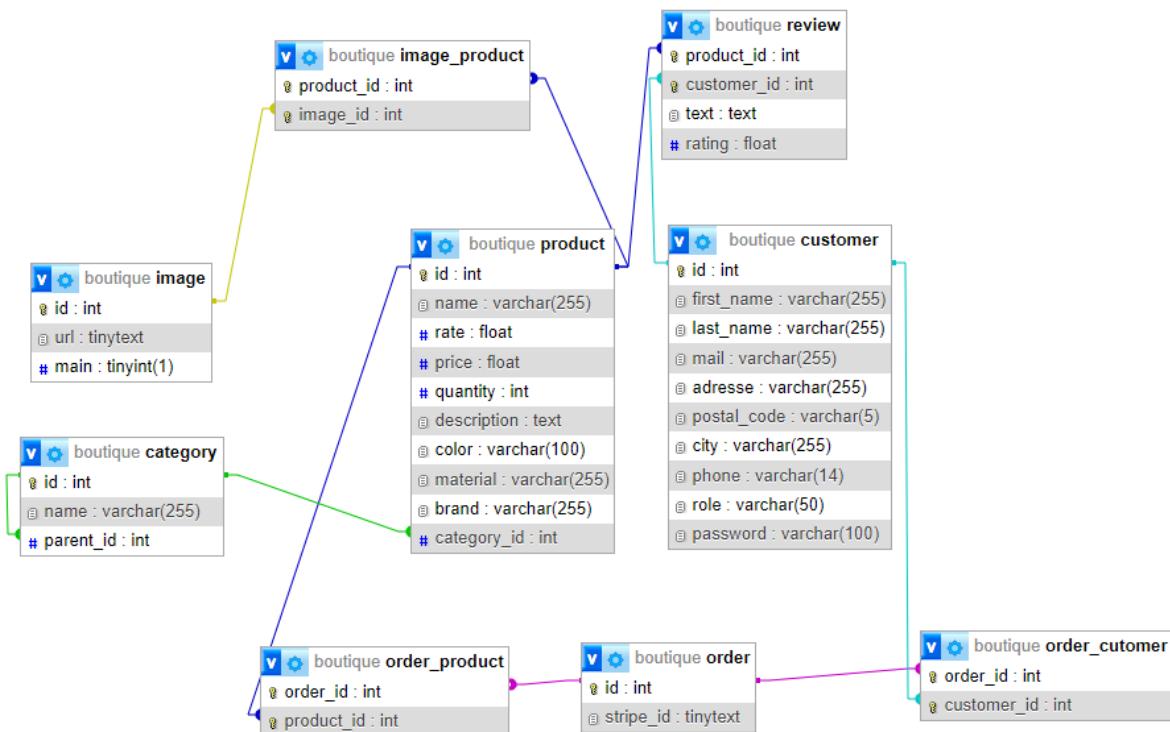
modifier

4. Mettre en place une base de données relationnelle

Nous avons conçu et mis en place une base de données MySQL pour gérer les produits, les utilisateurs et les commandes. La structure de la base de données a été pensée pour optimiser les performances et la sécurité des données. D'abord nous nous sommes mis d'accord pour faire une maquette de la base de données, et on y a assigné une personne :



Puis on a créé la base de données. Avec le temps et l'utilisation, on a découvert des bugs ou des améliorations à faire. En voici le résultat final :



5. Développer des composants d'accès aux données

Des classes PHP ont été créées pour gérer l'accès aux données, permettant d'effectuer des opérations CRUD (Create, Read, Update, Delete) sur les différentes entités de la base de données. Ces classes assurent une séparation claire entre la logique de l'application et les opérations de base de données.

```
35  public function getAll($id)
36  {
37      // connection to database
38      $conn = Database::connect();
39
40      // prepare the SQL request
41      $sql = $conn->prepare("SELECT * FROM customer WHERE id = :customer_id");
42      $sql->execute(
43          array(
44              'customer_id' => $id
45          )
46      );
47      // Stock the data in $userData
48      $userData = $sql->fetch(PDO::FETCH_ASSOC);
49      $sql->closeCursor();
50
51      // Return the data
52      return $userData;
53 }
```

6. Développer des composants métier côté serveur

Nous avons développé des composants métiers pour gérer les règles de l'application, comme la vérification de l'unicité des adresses email et des numéros de téléphone lors de l'inscription des utilisateurs. Cela assure l'intégrité des données et une meilleure expérience utilisateur.

```

135 |     public function testInsertUser($mail, $phone)
136 |     {
137 |         // Connection to database
138 |         $conn = Database::connect();
139 |
140 |         // Set $error to "none"
141 |         $error = "none";
142 |
143 |         // Check if the mail is valid
144 |         $sql = $conn->prepare("SELECT mail FROM customer WHERE mail = :email");
145 |         $sql->execute(
146 |             array(
147 |                 'email' => $mail
148 |             )
149 |         );
150 |         if ($sql->rowCount() > 0) {
151 |             // set $error to "mailAlreadyUsed" in case the mail is already taken
152 |             $error = "mailAlreadyUsed";
153 |             // return the error
154 |             return $error;
155 |         }
156 |         $sql->closeCursor();
157 |
158 |         // Check if the phone is valid
159 |         $sql = $conn->prepare("SELECT phone FROM customer WHERE phone = :phone");
160 |         $sql->execute(
161 |             array(
162 |                 'phone' => $phone
163 |             )
164 |         );
165 |         if ($sql->rowCount() > 0) {
166 |             // set $error to "phoneAlreadyUsed" in case the phone number is already taken
167 |             $error = "phoneAlreadyUsed";
168 |             // return the error
169 |             return $error;
170 |         }
171 |         $sql->closeCursor();
172 |
173 |         // Return "none" if there is no error
174 |         return $error;
175 |
176 |     }

```

7. Développer la partie back-end d'une application web et web mobile

Le back-end de l'application a été développé en PHP orienté objet, assurant une structure de code propre et modulaire. Le back-end gère les requêtes des utilisateurs, les transactions avec la base de données, et les réponses JSON pour les opérations AJAX côté client.

```

162 |         fetch("http://localhost:8080/controller/php/profilController.php", requestOptions)
163 |             .then(response => {
164 |                 return response.json();
165 |             }
166 |             )
167 |             .then(data => {
168 |                 switch (data.status) {
169 |                     case "error":
170 |                         self.location = '/profil?error=' + data.error;
171 |                         break;
172 |                     case "success":
173 |                         self.location = '/profil?success=success';
174 |                         break;
175 |                     default:
176 |                         break;
177 |                 }
178 |             })
179 |             .catch(error => {
180 |                 self.location = '/profil?error=UnexpectedError'
181 |             });

```

Technologies utilisées

- **Front-end** : HTML, CSS, JavaScript, Jquery
- **Back-end** : PHP (orienté objet), fetch API (Javascript)
- **Base de données** : MySQL
- **Outils de gestion de projet** : Trello, GitHub
- **Conception d'interface** : Figma

Défis rencontrés et solutions apportées

- **Accéder et interagir avec la base de données**

L'une des consignes de l'exercice était : "Faire un maximum en JS". Cependant, effectuer des interactions directes avec la base de données en JavaScript n'est pas sécurisé. Nous avons donc dû trouver une autre méthode pour y parvenir de manière sécurisée.

C'est là que l'API fetch de JavaScript est intervenue. Grâce à fetch, nous pouvons envoyer des requêtes HTTP pour transmettre des données de manière sécurisée à des scripts PHP. Ces scripts PHP, à leurs tours, interagissent avec la base de données pour effectuer les modifications nécessaires ou envoyer les informations demandées. Ainsi, fetch permet de séparer la logique côté client et serveur, garantissant une meilleure sécurité et organisation du code.

```
162   fetch("http://localhost:8080/controller/php/profilController.php", requestOptions)
163     .then(response => {
164       return response.json();
165     }
166     )
167     .then(data => {
168       switch (data.status) {
169         case "error":
170           self.location = '/profil?error=' + data.error;
171           break;
172         case "success":
173           self.location = '/profil?success=success';
174           break;
175         default:
176           break;
177       }
178     })
179     .catch(error => {
180       self.location = '/profil?error=UnexpectedError'
181     });

```

La page “profilController.php” :

```
19     $user = new CrudUser();
20
21     $testInsert = $user->testUpdateUser($id, $mail, $phone, $password);
22
23
24     if ($testInsert == 'mailAlreadyUsed') {
25         echo json_encode(
26             array(
27                 'status' => 'error',
28                 'error' => 'mailAlreadyUsed'
29             )
30         );
31     } elseif ($testInsert == 'wrongPassword') {
32         echo json_encode(
33             array(
34                 'status' => 'error',
35                 'error' => 'wrongPassword'
36             )
37         );
38     } elseif ($testInsert == 'phoneAlreadyUsed') {
39         echo json_encode(
40             array(
41                 'status' => 'error',
42                 'error' => 'phoneAlreadyUsed'
43             )
44         );
45     } elseif ($testInsert == 'UserNotFound') {
46         echo json_encode(
47             array(
48                 'status' => 'error',
49                 'error' => 'UserNotFound'
50             )
51         );
52     } else {
53         $user->updateUser($id, $name, $firstname, $mail, $phone, $adresse, $postalCode, $city, $password);
54         session_start();
55         $_SESSION['userFirstName'] = $user->getFirstname();
56         echo json_encode(
57             array(
58                 'status' => 'success',
59                 'error' => 'none'
60             )
61         );
62     }
63 }
```

- **Rendre lisible et sécurisé le lien du site**

Lors de l'utilisation d'un site internet, on peut remarquer dans la barre de recherche le lien du site. Ce lien est généralement propre et lisible pour permettre à l'utilisateur de comprendre où il se situe sur le site et comment il est organisé.

Pour ne pas laisser accès aux pirates, ces liens ne représentent pas comment est vraiment organisé le site, son arborescence. Car si les pirates le connaissent alors c'est la voix ouverte aux piratage du site.

Pour pallier ce problème, on a choisi d'utiliser Alto routeur comme routeur pré-monté afin de modifier les liens, mapper les URL et contrôler l'accès des utilisateurs.

Voici à quoi ressemblerait le lien de la page profil sans routeur :

[“localhost/games-thrones/src/profil.php”](http://localhost/games-thrones/src/profil.php)

Ici on peut facilement voir l'arborescence du site :

- localhost : Adresse du serveur web de votre machine
- games-thrones : Le dossier racine du site
- src : Un dossier dans lequel se trouve les pages du site, on peut le dire car juste après ce dossier se trouve une des pages principales du site.

Dans ce cas, on utilise un routeur pour mapper les routes. Comme vu un peu plus haut, on utilise la commande ‘`php -S localhost:8080 -t public`’ pour définir le dossier “public” comme étant le dossier racine du projet. Quand on ouvrira la page “localhost:8080” alors le code de l’index (vu un peu plus tard) ira chercher dans le lien ce qui se trouve après le “8080” et affichera la page demandée.

Ensuite le mappage, dans l’index, à la racine du site, on créer des routes :

```
10 $router->map('GET', '/', '/home', 'accueil');
```

Avec cette ligne, on dit que si après le “8080” il y a “/” (= rien), alors il ira chercher la page “home” et le lien apparaîtra comme “localhost:8080”. Le “accueil” ici est là pour donner un nom à la route et le “GET” est la méthode de requête HTTP de la page.

On peut aussi faire en sorte que si l'utilisateur met dans la barre de recherche un lien, une route n'existant pas, le faire arriver sur une page 404 :

```

53  if (is_array($match)) {
54      // Handle routes that send JSON
55      if (str_contains($match["name"], "Ajax")) {
56          if (is_callable($match['target'])) {
57              call_user_func_array($match['target'], $match['params']);
58          } else {
59              $params = $match['params'];
60              require "../src/{$match['target']}.php";
61          }
62      } else if (str_contains($match["name"], "_backend")) {
63          if (is_callable($match['target'])) {
64              call_user_func_array($match['target'], $match['params']);
65          } else {
66              $params = $match['params'];
67              require "../src/{$match['target']}.php";
68          }
69      } else {
70          // Handle routes that send HTML
71          require '../templates/header.php';
72          if (is_callable($match['target'])) {
73              call_user_func_array($match['target'], $match['params']);
74          } else {
75              $params = $match['params'];
76              require "../src/{$match['target']}.php";
77          }
78          require '../templates/footer.php';
79          echo "</html>";
80      }
81  } else {
82      // 404 error
83      header("location:" . $router->generate('404') . "");
84 }

```

Sur ce code, on récupère le lien actuel demandé, on va chercher le fichier associé,

pour l'appeler dans l'index (la page de ce code).

Si la page n'est pas trouvée, alors on redirige l'utilisateur sur la page 404 avec

"header("location:" . \$router->generate('404') . "");".

De cette manière, l'utilisateur verra toujours un lien simple et sécurisé, qui Cela empêchera de voir la vraie arborescence du projet.

Exemple de la page profil :

route : 13 \$router->map('GET', '/profil', '/profil', 'profil');

lien sur le navigateur : <http://localhost:8080/profil>

Résultats et impact

Le projet games-thrones, bien que non terminé, m'a apporté plus que des "bénéfices", j'ai pu découvrir une nouvelle manière de coder, en orienté objet. Bien que l'ayant appris plus tôt, je n'avais jamais vraiment coder en orienté objet. Ça m'a aussi grandement amélioré, que ce soit en PHP ou en Javascript ou l'utilisation de github.

J'ai pu aussi améliorer ma gestion du temps, de mes tâches. Apprendre à programmer en équipe :

- mettre des commentaires
- expliquer comment fonctionne les parties
- programmer de façon propre
- bien nommer ses variables, ses classes, méthodes, fonctions.

J'ai aussi amélioré mes capacités sociales, je suis très timide de base, mais en m'efforçant de discuter avec mes collègues, que ce soit pour parler d'un problème sur le projet ou autre.

Voyage-blog

Introduction

Le projet Voyage-blog est un site vitrine réalisé en une semaine par une équipe de deux développeurs web en formation dans le cadre du cursus Développeur Web et Web Mobile (DWWM). Ce projet avait pour objectif principal de créer une interface web et web mobile statique, tout en mettant l'accent sur le responsive design pour assurer une expérience utilisateur optimale sur tous les appareils.

En travaillant sur ce projet, nous avons non seulement amélioré nos compétences techniques en développement web notamment le responsive, mais également appris à collaborer efficacement en petit groupe. Cet exercice a été conçu pour simuler un environnement de travail professionnel, nous permettant ainsi d'acquérir une expérience précieuse en gestion de projets et en collaboration.

Objectifs du projet

- **Développement technique** : Utiliser HTML et CSS pour créer un site vitrine statique, et mettre en œuvre des techniques de responsive design pour assurer une compatibilité sur tous les appareils.



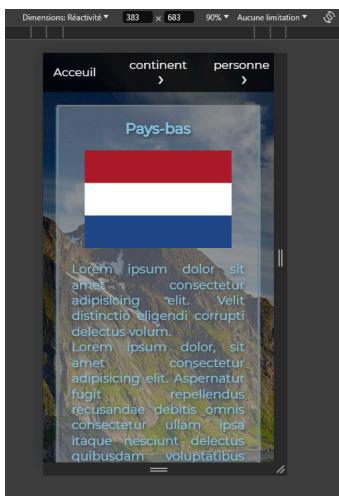
- **Compétences pratiques** : Acquérir de l'expérience en gestion de projets et en collaboration d'équipe.
- **Fonctionnalité utilisateur** : Créer une plateforme où les utilisateurs peuvent consulter des informations sur deux pays.

Fonctionnalités principales

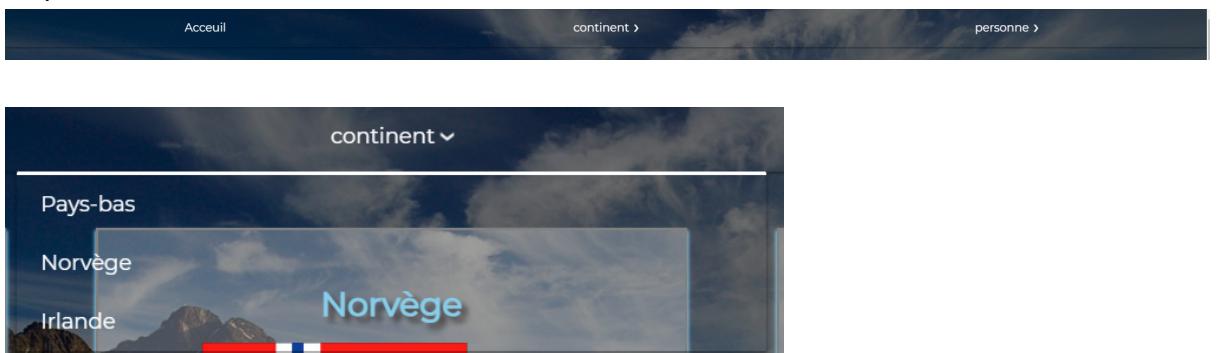
- **Présentation de pays** : Pages dédiées à la présentation de deux pays différents.



- **Responsive design** : Assurer une expérience utilisateur optimale sur tous les appareils.



- **Navigation fluide** : Mise en place d'un menu de navigation intuitif pour faciliter l'exploration du site.



Compétences couvertes

1. **Installer et configurer son environnement de travail en fonction du projet web :**
Utilisation d'outils comme un serveur local (Laragon), un éditeur de code (Visual Studio Code), et GitHub.
2. **Développer une interface utilisateur web statique**
Structure HTML :
 - Construction de la structure des pages web en utilisant HTML5. Cela inclut la création des différents éléments de la page tels que les en-têtes, les paragraphes, les listes, les formulaires et les images. L'objectif est de créer une structure sémantiquement correcte pour améliorer l'accessibilité et le référencement.

```
1  <main>
2    <article class="article">
3      <section class="article1">
4        <div class="fond_article1"></div>
5        <h1 class="titre_section1">Pays-Bas</h1>
6        
7        <p class="text_section1">Lorem ipsum dolor sit amet consectetur adipisicing elit. Velit distinctio
8          eligendi corrupti delectus volum.<br> Lorem ipsum dolor, sit amet consectetur adipisicing elit.
9          Aspernatur fugit repellendus recusandae debitis omnis consectetur ullam ipsa itaque nesciunt
10         delectus quibusdam voluptatibus tenetur, consequuntur tempora. Minima reiciendis aut corporis
11         doloribus.</p>
12    </section>
13    <section class="article2">
14      <h1 class="titre_section2">Norvège</h1>
15      
16      <p class="text_section2">Lorem ipsum dolor sit amet consectetur adipisicing elit. Velit distinctio
17        eligendi corrupti delectus volum.<br> Lorem ipsum dolor, sit amet consectetur adipisicing elit.
18        Aspernatur fugit repellendus recusandae debitis omnis consectetur ullam ipsa itaque nesciunt
19        delectus quibusdam voluptatibus tenetur, consequuntur tempora. Minima reiciendis aut corporis
20        doloribus.</p>
21    </section>
22    <section class="article3">
23      
24    </section>
25  </article>
26 </main>
```

Ici, nous avons fait en sorte que chaque balise soit à sa place et qu'elle corresponde à ce qu'elle contient, par exemple on ne va pas mettre du texte brut dans la balise `<main>` mais dans une balise `<p>` qui pourrait être dans le `<main>`.

Mise en Forme avec CSS :

- Application de styles CSS pour définir l'apparence des éléments HTML.
Utilisation de propriétés CSS pour contrôler la disposition (layout), les couleurs, les polices, les espacements, les bordures et les effets visuels.S

Responsive Design :

- Implémentation du responsive design pour garantir que l'interface utilisateur s'adapte à différents types d'appareils et tailles d'écran. Utilisation des media queries CSS pour ajuster les styles en fonction de la largeur de l'écran et garantir une expérience utilisateur optimale sur les mobiles, tablettes et ordinateurs.

Technologies utilisées

- **Front-end :** HTML, CSS, PHP
 - Le php est utilisé pour permettre d'appeler le menu et le footer sur chaque page pour ne pas répéter le code

```
<?php
    include('nav.html');
    include('main_index.html');
    include('footer.html');
?>
```

- **Outils de gestion de projet :** Trello, GitHub

Défis rencontrés et solutions apportées

- **Responsive design :** Utilisation de media queries en CSS pour garantir que le site s'adapte à toutes les tailles d'écran.

```
136 @media screen and (min-width: 768px) {
137     .article {
138         flex-direction: row;
139         flex-wrap: wrap;
140         justify-content: space-between;
141     }
142
143     .article1, .article2, .article3 {
144         width: calc(33% - 40px);
145     }
}
```

- **Coordination de l'équipe :** Collaboration étroite et communication efficace pour garantir la cohérence du design et du développement.

Résultats et impact

Le projet voyage-blog est celui qui m'a plongé dans le responsive. C'est là où j'ai pu mettre en pratique les théories apprises lors des petits exercices fait un peu avant ce projet.