



Session 2020-2021

Dossier Professionnel

*Titre Professionnel
Développeur Web Et Web Mobile*

Remerciements

Mes Premiers remerciements Irons d'abord à Monsieur **DUBURQUOI Gregory**, pour son aide précieuse qui a fait que je rédige ce dossier aujourd'hui

J'adresse également un grand merci à Monsieur **BOUCHEZ Marc** qui est pour moi un mentor dans le monde du développement Web, c'est aussi grâce à lui que j'ai développé ce fort intérêt pour le web ainsi que ces connaissances précoces en front-end, grâce à son enseignement de courte durée mais d'une efficacité effarante.

Je remercierais tout particulièrement Monsieur **STIEN Simon** qui est pour moi un deuxième mentor, je le remercie et tire mon chapeau, un homme qui en deux semaines est capable d'enseigner et de faire assimiler des mois complets de formation, il est pour moi comme un titan de l'enseignement du développement à travers ces mots je lui adresse un grand merci pour sa force de galvanisation des troupes, sa pédagogie et ses compétences qu'il nous a fait partager tout au long de ces deux semaines.

Je remercie aussi Monsieur **DESCHARLES Vincent** notre référent de formation, qui a toujours été présent et ce malgré les difficultés du quotidien c'est aussi en parti grâce à son enseignement que nous en sommes là aujourd'hui.

Je tiens à remercier Monsieur **FARISS M'hammed** d'avoir permis la mise en place des formations numériques et de nous avoir accueillis au sein du centre AIFOR.

Un merci également a toute l'équipe du centre **AIFOR** pour l'accueil ainsi que l'accompagnement au sein des locaux.

Pour finir, je Tiens à remercier tous mes collègues de formation avec qui j'ai passé six moi mémorables pour leur ouverture d'esprit, leurs esprits d'équipe ainsi que la cohésion au sein de notre groupe tout au long de la formation et des différents projets réalisés.

Table des matières

Sommaire :	2
Introduction	4
1. Compétences mises en œuvre	5
<i>Compétences couvertes par la partie front-end d'un site internet</i>	5
<i>Compétences couvertes par la partie back-end d'un site internet</i>	5
1. <i>Développer la partie front end d'une application Web ou web mobile.</i>	6
1. Le cahier des charges :	7
Objectif	8
Conception :	9
La maquette :	10
2. Front-End	11
1. <i>Partie statique : HTML, CSS</i>	12
A. Ajout des données de page d'accueil	14
B. Ajout du style de page d'accueil	15
C. Création des pages secondaires	16
D. Ajout du style des pages secondaires	17
Responsive :	21
Conclusion de la partie HTML, CSS	23
<i>Partie dynamique : JavaScript</i>	24
3. Back-End : PHP, MySQL	25
<i>Étape 1 : Mise en place de l'environnement back-end :</i>	26
<i>Optimisation de notre Template</i>	28
A. Header.php	28
B. Footer.php	29
<i>Modification du code de la page d'accueil</i>	31
<i>Interface de réservations</i>	33
<i>Validations et sécurisation des données :</i>	37

Introduction

La mise en place de ce projet vient d'une réflexion suite à une obligation de reconversion professionnelle.

N'ayant plus la possibilité d'exercer mon métier d'aide-soignant, il était pour moi indispensable de me reconvertir afin de pouvoir continuer à évoluer dans la sphère du monde professionnel.

Dans ce but, j'ai donc intégré en 2019 la formation Dynamique Vers l'Emploi au sein du centre de formation AIFOR.

Et je remercie d'ailleurs Mr DUBURQUOIS pour son aide qui nous a beaucoup apporté au cours de la formation et grâce à laquelle j'en suis là aujourd'hui.

Au cours de cette session de formation, nous avons évoqué les possibilités de réorientation propre à chacun c'est par hasard que je suis tombé sur cette proposition de formation préparatoire au numérique après avoir raté une intégration sur une formation sociale.

J'ai toujours été très attiré par le numérique et plus précisément l'informatique mais n'ayant pas les prérequis pour y accéder avant, j'ai donc suivi une autre voie.

En découvrant qu'une préparatoire aux métiers du numérique était proposée par ce même centre et qu'il m'était possible d'en intégrer une, j'ai donc sauté sur l'occasion sans même avoir à y réfléchir, et c'est ainsi que j'ai intégré la préparatoire digitale du Montreuillois qui démarrait fin 2019 et qui nous permettait de nous préparer à intégrer le titre professionnel développeur web et web mobile.

Cette préparatoire a été une succession de cours théoriques et de mise en situation sur différentes simulations de projets requérant la mise en pratique des différentes compétences acquises au fur à mesure de l'avancement de cette session.

Lors de ce dossier je présenterais donc les différents projets qui m'ont permis d'obtenir les compétences nécessaires à l'obtention du titre.

1. *Compétences mises en œuvre*

L'obtention du titre professionnel nécessite d'acquérir un certain nombre de compétences spécifiques définies dans le référentiel de formation selon deux sous-catégories :

Compétences couvertes par la partie front-end d'un site internet

- *Maquetter une application*
- *Maquetter une interface utilisateur web statique et adaptable*
- *Développer une interface utilisateur web dynamique*
- *Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce.*

Compétences couvertes par la partie back-end d'un site internet

- *Développer les composants d'accès aux données*
- *Développer la partie back-end d'une application web ou web mobile*

1. *Développer la partie front end d'une application Web ou web mobile.*

- Définir un projet d'application web
- Maquetter le projet
- Réaliser une interface web statique et adaptable.
- Développer une interface utilisateur web dynamique
- Réaliser une interface utilisateur avec solution de gestion de contenu ou e-commerce

1. Dans cette première partie nous avons fait appel à nos capacités d'**écoute**, d'**analyse** et de **compréhension** afin d'assimiler correctement la demande du prestataire et les prérequis du projet et de pouvoir monter un plan cahier des charges...
2. Pour entrer dans la phase deux du projet nous mettons en œuvre nos compétences de **structuration** ainsi que de **mise en forme** afin de pouvoir obtenir un premier aspect un peu plus concret de notre projet que nous maquettons. Ce qui nous a mené à acquérir à développer la compétence de **maquettage**.
3. Pour finir nous utilisons et optimisons nos **compétences** en **programmation** (**html** et **css**) afin de dresser un premier prototype.

Projet 1 : Le coin des bonnes auberges

1. Le cahier des charges :

Contexte du site :

- Titre du site **Le Coin Des Bonnes Auberges**
- Site ciblage public et devra donc être facilement trouvable via un référencement efficace
- Objectif concentration en un seul point des informations sur les auberges du secteur ainsi que leurs propositions gastronomiques.

L'objectif du site vise à **informer** sur la restauration traditionnelle dans le secteur du Montreuillois, notamment les établissements type auberges.

Les vacanciers, touristes auraient via cette interface, la possibilité de consulter nombreuses données sur les établissements ciblés notamment : les adresses, localisations, coordonnées, différents comptes ou pages sur les réseaux sociaux, informations sur la cuisine proposée mais aussi d'émettre leur avis sur les établissements, la cuisine, l'accueil etc....

Ce qui permettrait au nouveaux visiteurs, touristes d'avoir dans un premier temps un plan des établissements, ainsi qu'une première idée des différents types de cuisines et de plats proposés.

Exigences du projet :

- Site vitrine multilingue avec facilité d'accès aux différentes langues
- Nombre de pages -> Non défini par le client environs ... Pages selon L'organigramme.
- Charte graphique -> Couleurs Noir et Blanc
- Typographie -> Non définie par le client
- Dégagement d'une personnalité dynamique et conviviale dans une apparence épurée
- Site codé à la main (**Pas de CMS**)

Objectif

Les objectifs du site seront donc dans un premier temps :

- *Site axé sur un ciblage public, devra donc être facilement trouvable. (Référencement efficace)*
- *La génération d'un trafic conséquent.*
- *Concentration en un seul point des informations sur les auberges du secteur ainsi que leurs propositions gastronomiques.*

Public visé :

Le site dispose d'un ciblage tous publics, aussi bien un public de type touristes vacanciers ou encore habitué du ou des établissements traités par l'interface.

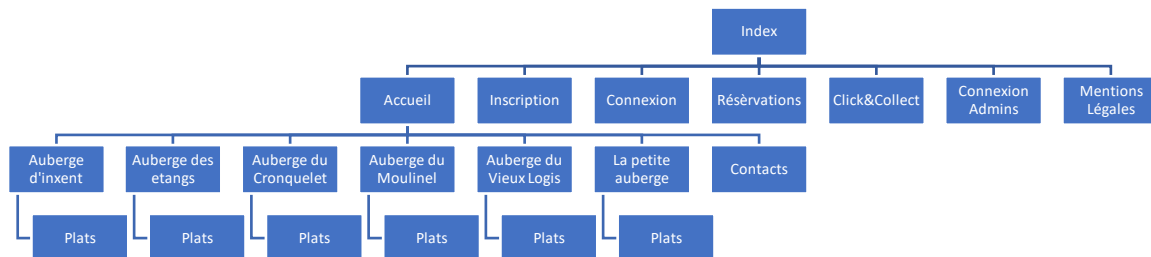
Charte graphique :

- *Noir*
- *Blanc*
- *Vert*

Contenu du site

- *Une page index comprenant une animation JavaScript faisant apparaître un bouton pour accéder au site.*
- *Une page principale contenant une galerie affichant les photos-liens de chacune des auberges répertoriées.*
- *Photos-liens redirigeant vers une page propre à chaque auberge*
- *Galleries d'images reprenant une sélection de plats*
- *Espaces avis et commentaires.*
- *Espace inscription.*

Arborescence



Conception :

En termes de conception ce site reste simple de réalisation

Ledit site reprendra un certain nombre d'auberges du secteur.

Il se composera de plusieurs pages, une page d'accueil, une page dédiée à chacune des auberges reprises dans le site, une page contacts.

On trouvera en page d'accueil une galerie d'images ayant une image cliquable pour chacune des auberges.

Le clique de chaque image redirigera vers une page dédiée à cette meme auberge.

Le site en lui-même sera réalisé manuellement à l'aide des langages :

- HTML
- CSS
- JavaScript
- PHP
- MySQL

Le projet ne possède pas de date limite si ce n'est la fin de session de formation.

La maquette :

La maquette a été réalisée sur adobe XD après finalisation du cahier des charges.

Nous avons choisi adobe XD car c'est un logiciel qui inclut de nombreuses fonctionnalités très utiles et puissantes et permet d'obtenir un rendu très réaliste fonctionnellement et graphiquement.

Illustration 1 : Première version de la maquette

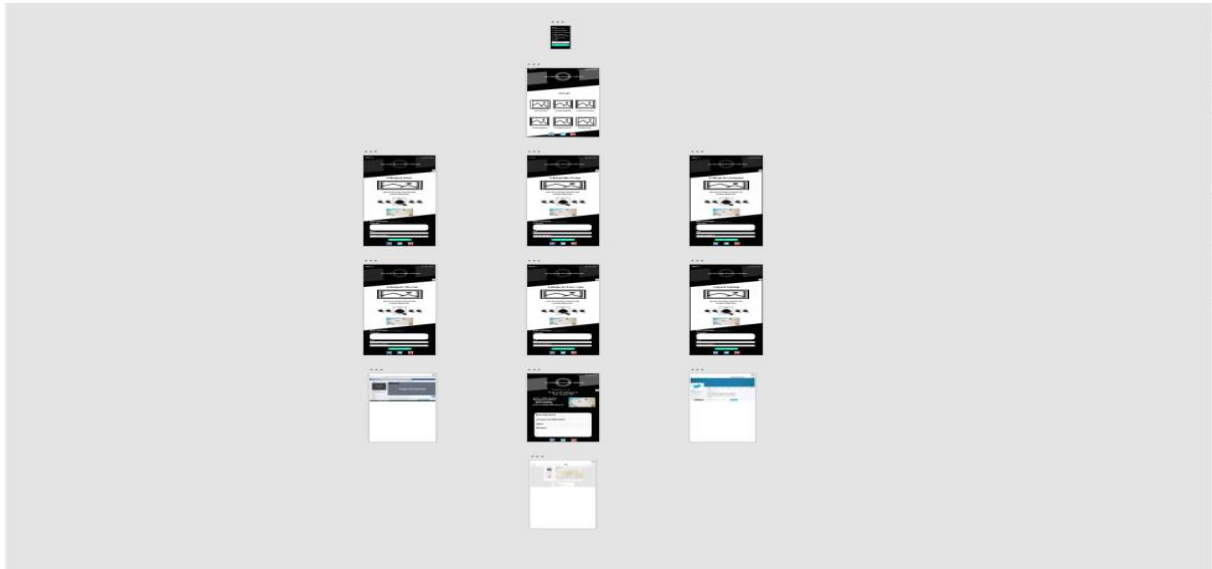
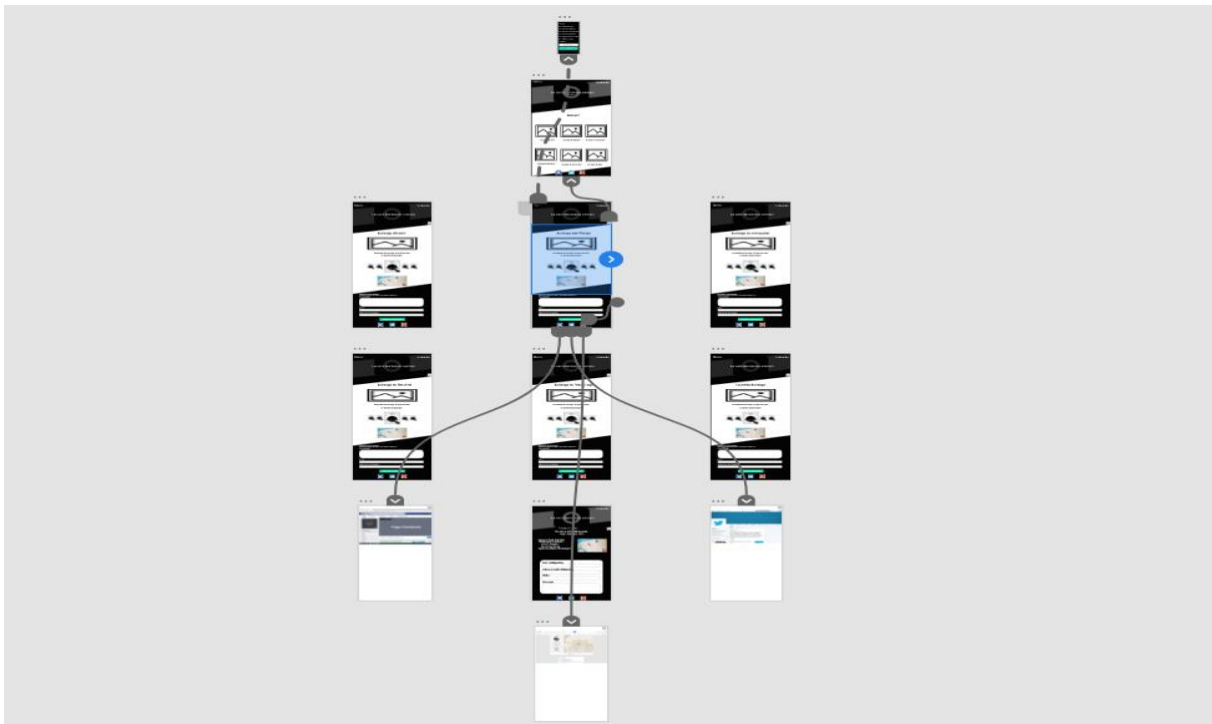


Illustration 2 : Exemple des prototypes du site



La validation de cette maquette nous permet de réaliser une première mise en place du site.

2. Front-End

*Le Front-End d'un site correspond au processus de création de l'aspect structurel du site, il est principalement réalisé grâce aux langages **HTML**, **CSS** et **JavaScript**.*

Il est la première brique de notre conception car il permet de façonner l'identité de notre site tout en ayant un visuel sur les potentielles erreurs de mise en page.

*Afin de vous exposer sa mise en place je vais vous présenter la partie dite statique (**HTML**, **CSS**).*

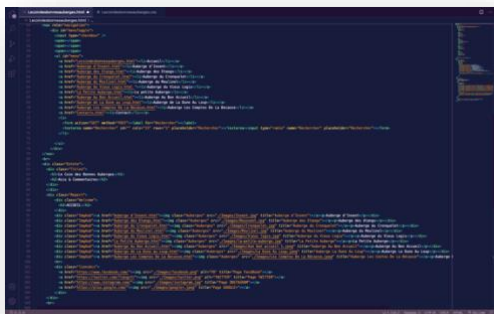
Puis j'exposerais les composants back-end ajoutées afin de créer l'accès aux données.

1. Partie statique : HTML, CSS

Structure de base (Template)

Dans cette partie, nous avons tout d'abord élaboré notre page d'accueil qui fera office de base du site en utilisant le langage html qui créera l'ossature du site ainsi que notre menu de navigation.

Illustration 1 : Html structure de base (Template)



```
</nav>
<br>
<div class="Entete">
  <div class="Titles">
    <h1>Le Coin des Bonnes Auberges</h1>
    <h2>Avis & Commentaires</h2>
  </div>
</div>
<div class="Repert">
  <div class="Welcome">
    <h2>ACCUEIL</h2>
  </div>
  <div class="ImgAub"><a href="Auberge d'Inxent.html"><a href="Auberge des Etangs.html"><a href="Auberge du Cronquelet.html"><a href="Auberge du Moulinel.html"><a href="Auberge du Vieux Logis.html"><a href="La Petite Auberge.html"><a href="Auberge Au Bon Accueil.html"><a href="Auberge de La Dune au Loup.html"><a href="Auberge Les Comptes De La Becasse.html">
    <a href="https://www.facebook.com/">
    <a href="https://twitter.com/?lang=fr"></a>
    <a href="https://plus.google.com/"></a>
  </div>
</div>
```

Puis avec le code CSS suivant qui donne des informations sur la forme, le fond, la couleur de fond (background), la couleur et style des polices.

Illustration 2 : Css définition du Style de base de notre Template

```

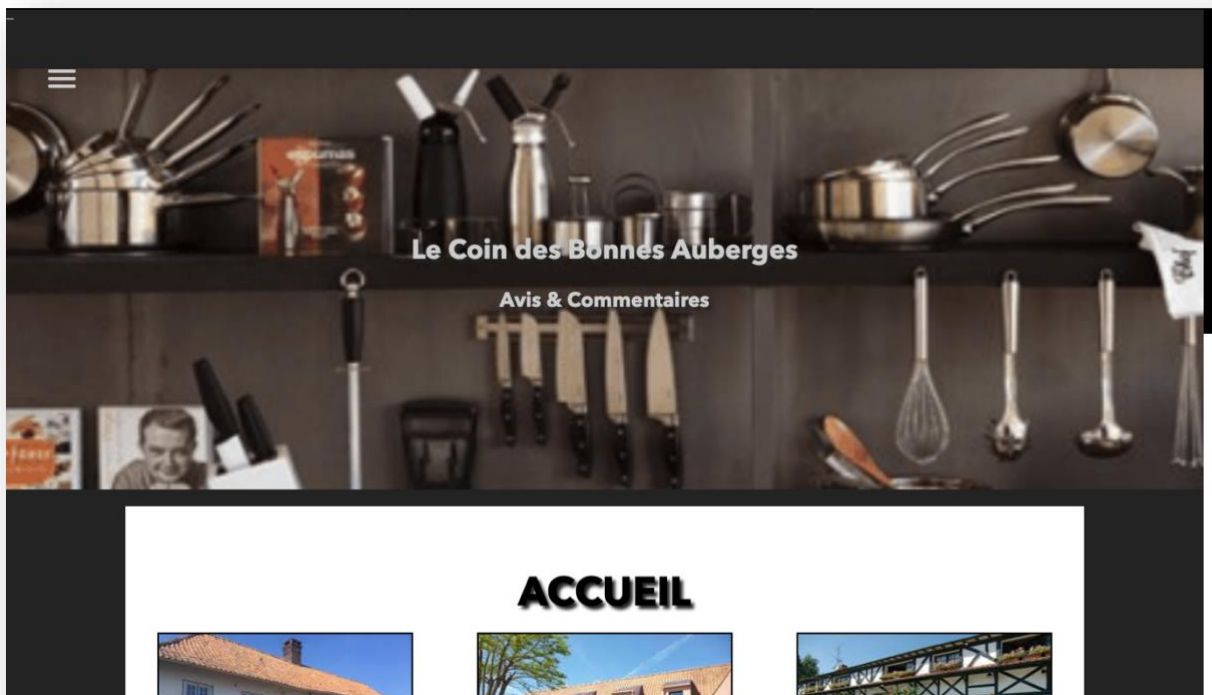
12  body{
13      margin: 0;
14      padding: 0;
15      /* make it look decent enough */
16      background: #232323;
17      color: #cdcdcd;
18      font-family: "Avenir Next", "Avenir", sans-serif;
19  }
20
21  body::-webkit-scrollbar{
22      background-color: white;
23  }
24
25  body::-webkit-scrollbar-thumb{
26      background-color: black;
27  }
28
29  .Body2{
30      background-color: white;
31      width: 80%;
32      margin: auto;
33      color: black;
34  }
35

```

Par ce code, nous déterminons la couleur en plus de déterminer le squelette de notre interface, nous lui ajoutons également une barre de défilement ainsi qu'un conteneur secondaire (<div class="Repert">).

Nous avons maintenant notre structure de base pour notre site

Nous reprendrons le style de cette base afin de créer les pages suivantes, ce qui nous permettra d'avoir un visuel identique sur chaque page.

Illustration :

Ainsi nous avons obtenu une base solide pour notre site, à partir de cet instant nous pouvons donc ajouter et modeler très facilement le contenu de notre première page (accueil).

A. Ajout des données de page d'accueil

Afin de respecter la demande de notre client fictif, (voir cahier des charges), nous allons ajouter à notre conteneur secondaire plusieurs petits conteneurs qui accueilleront les données (photo et liens de redirection) de chacune de nos auberges.

Illustration :

```

<div class="Repert">
  <div class="Welcome">
    <h2>ACCUEIL</h2>
  </div>
  <div class="ImgAub"><a href="Auberge d'Inxent.html"></a>
  <div class="ImgAub"><a href="Auberge des Etangs.html"></a>
  <div class="ImgAub"><a href="Auberge du Cronquelet.html"></a>
  <div class="ImgAub"><a href="Auberge du Moulinel.html"></a>
  <div class="ImgAub"><a href="Auberge du Vieux Logis.html"></a>
  <div class="ImgAub"><a href="La Petite Auberge.html"></a>
  <div class="ImgAub"><a href="Auberge Au Bon Accueil.html"></a>
  <div class="ImgAub"><a href="Auberge de La Dune au Loup.html"></a>
  <div class="ImgAub"><a href="Auberge Les Comptes De La Becasse.html"></a>
  </div>
  <div class="liensAcc">
    <a href="https://www.facebook.com/"></a>
    <a href="https://twitter.com/?lang=fr"></a>
    <a href="https://www.instagram.com/"></a>
    <a href="https://plus.google.com/"></a>
  </div>
</div>

```

B. Ajout du style de page d'accueil

Maintenant que nous avons ajoutés nos conteneurs avec nos données, nous allons en définir le style en utilisant de nouveau des propriétés CSS.

Nous modifierons dans un premier temps le conteneur secondaire de notre Template de façon à ce que celui-ci dispose les données en trois colonnes de trois lignes chacune en utilisant aux propriété display (flex) et (flex-wrap).

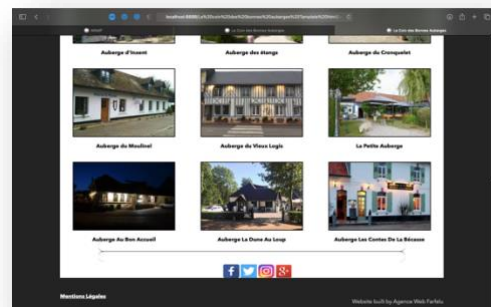
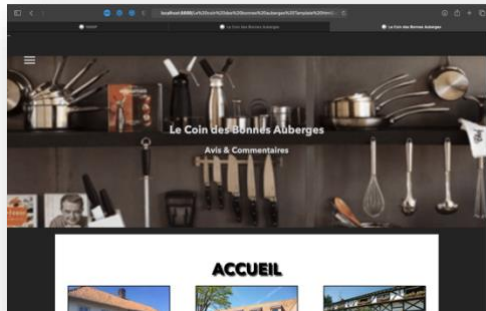
```
.Reprt{
  width: 80%;
  display: flex;
  flex-wrap: wrap;
  justify-content: space-around;
  background-color: white;
  margin: auto;
  padding-top: 30px;
}
```

Illustration 3 : Css propre au contenu de la page d'accueil

```
6 <link rel="stylesheet" href="/Lecoindesbonnesauberges.css">
7 <style type="text/css">
8 .Titles{
9   display: block;
10  text-align: center;
11  padding-top: 10px;
12 }
13
14 .Auberges{
15   height: 200px;
16   width: 300px;
17   display: block;
18   border: 2px solid black;
19   margin: auto;
20   margin-top: 20px;
21 }
22
23 .Reprt{
24   width: 80%;
25   display: flex;
26   flex-wrap: wrap;
27   justify-content: space-around;
28   background-color: white;
29   margin: auto;
30   padding-top: 30px;
31 }
32
33 .ImgAub{
34   display: block;
35   width: 300px;
36   text-align: center;
37   color: black;
38   font-weight: bold;
39 }
40
41 .Welcome{
42   width: 80%;
43   height: 100px;
44   color: black;
45   text-align: center;
46   font-size: xx-large;
47   font-weight: bold;
48 }
49
```

Avec l'ajout de quelques données notamment l'insertion de conteneurs, d'images et de champs de texte avec les balises `<div>`, `` et `<h2>` on obtient notre page complète.

Illustration : rendu visuel de l'accueil



C. Création des pages secondaires

Maintenant que nous avons notre page d'accueil, nous allons pouvoir procéder à la création des pages secondaires qui renseigneront les données propres à chaque établissement.

Pour ceci nous allons également créer un design de base que l'on retrouvera sur chacune des pages auberges.

Nous commencerons donc par reprendre notre Template de base et y apporter quelques modifications de contenu et de style CSS.

```
<div class="Body">
<div class="ETANGS">
<div class="Etablissements">

<div class="Présentations">
<p>Au cœur de la Vallée de la Courne, dans une bâtisse du XVIIIème siècle et ancien presbytère, La
<p>Venez apprécier, au restaurant, les saveurs de nos produits locaux accompagnés de beaux vins.</p>
<p>L'auberge d'Isen dispose de 5 chambres de charme où vous passerez une agréable nuit à la camp
</div>
<div class="CookLine">
<div id="Slide">
<figure>

</div>
</div>
```

```
<iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1s142544.78913314410!2s773195515842087!3s58.3784866794645!2s"
</div>
<div class="form">
<form class="Comments" action="ma-page-de-traitement" method="POST">
<div>
<label for="Name">Nom:</label>
<input type="text" id="Name" required name="user_name">
</div>
<div>
<label for="Message">Commentaire:</label>
<div class="textarea">
<textare id="Mess" type="text" id="Commentaire" required name="User_comment">
</div>
<div class="button">
<button type="submit"><strong>Envoyer le message</strong></button>
</div>
</form>
</div>
```

Nous commencerons par ajouter au conteneur secondaire trois conteneurs qui engloberont les différentes données des auberges :

`<div class="établissements">` qui englobera la photo principale de l'auberge en question

`<div class="présentations">` qui contiendra le texte de présentation

`<div class="Cookslide">` qui englobera un slider d'images des plats proposés

Grace à la balise `<iframes>` nous afficherons un plan routier interactif qui permet d'obtenir un itinéraire pour se rendre dans l'établissement depuis l'endroit où on se trouve.

Enfin nous ajouterons une balise `<form>`, qui permettra aux visiteurs de laisser un commentaire sur la page.

D. Ajout du style des pages secondaires

Tout d'abord nous allons grâce à une fonction CSS (@keyframes Zoom) créer une animation qui fera apparaître notre image de l'établissement sur notre premier conteneur (`<div class="Aub">`) qui contiennent les photos des auberges.

```
.Aub{
  height: 400px;
  width: 70%;
  margin-left: 15%;
  box-shadow: 0px 0px 0px black 5px 5px 5px;
  animation: linear 5s zoom;
}
```

Identification du conteneur qui recevra l'animation en le récupérant par son nom de classe

Animation keyframe et définition de l'état par défaut (from) à l'état secondaire to

```
@keyframes zoom {
  from {
    transform: scale(0);
  }
  to {
    transform: scale(1);
  }
}
```

Ensuite nous allons ajouter notre zone de texte dans le conteneur (`<div class="Présentations">`) et en définir le style.

```

</div>
<div class="Présentations">
  <p>Au coeur de la Vallée de la Course, dans une bâtisse du XVIIIème siècle et ancien presbytère, Laurence et Jean-Marc Six
  <p>Venez apprécier, au restaurant, les saveurs de nos produits locaux accompagnés de beaux vins.</p>
  <p>L'auberge d'Inxent dispose de 5 chambres de charme où vous passerez une agréable nuit à la campagne, à deux pas de Mont
</div>
</div>

```

```

.Présentations{
  display: block;
  width: 90%;
  margin: auto;
  padding: 10px;
  box-shadow: 5px 5px 5px black;
}

```

Style CSS du conteneur
Présentations

Puis nous poursuivons en ajoutant à notre conteneur (`<div class="CookSlide">`) un second conteneur (`div class="slider"`), qui fera défiler les photos des différents plats proposés, nous donnons à ce conteneur un `[id]` qui nous permettra de l'identifier dans le **CSS** pour lui attribuer les paramètres nécessaires à l'animation de slide, ou encore si on souhaite le sélectionner en php.

```

<div class="CookSlide">
  <div id="slider">
    <figure>
      
      
      
      
      
      
    </figure>
  </div>
</div>

```

Ajout de style et de la fonction CSS du slider :

```

201     opacity: 0.5;
202   }
203
204   div#slider {
205     width: 100%;
206     max-width: 1000px;
207   }
208
209   div#slider figure {
210     position: relative;
211     width: 600%;
212     margin: 0;
213     padding: 0;
214     font-size: 0;
215     text-align: left;
216   }
217
218   div#slider figure img {
219     width: 10%;
220     height: 100%;
221     float: left;
222     height: 300px;
223   }
224
225   div#slider {
226     margin: auto;
227     width: 100%;
228     max-width: 600px;
229     overflow: hidden;
230     border: 1px solid black;
231     box-shadow: 4px 4px 4px black;
232   }
233
234   @keyframes slidy {
235     0% { left: 0%; }
236     20% { left: 0%; }
237     25% { left: -100%; }
238     45% { left: -100%; }
239     50% { left: -200%; }
240     70% { left: -200%; }
241     80% { left: -200%; }
242   }
243
244   div#slider figure {
245     position: center;
246     width: 500%;
247     margin: 0;
248     padding: 0;
249     font-size: 0;
250     left: 0;
251     text-align: center;
252     animation: 15s slidy infinite;
253   }

```

Nous poursuivons en ajoutant les deux derniers éléments de notre page à savoir l'iframe et le formulaire de dépôt de commentaires.

```

</div>
<iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d2544.789113314416!2d1.773195615842887!3d50.3704866794645!2m3!1
<br>
<br>
<hr>
<br>
<div class="form">
  <form class="Comments" action="ma-page-de-traitement" method="POST">
    <div>
      <label for="Name">Nom:</label>
      <br>
      <input type="text" id="Name" required name="user_name">
    </div>
    <div>
      <label for="Message">Commentaire:</label>
      <br>
      <textarea id="Mess" type="text" id="Commentaire" required name="User_comment"></textarea>
    </div>
    <div class="butonform">
      <button type="submit"><strong>Envoyer le message</strong></button>
    </div>
  </form>
</div>
<br>

```

```

iframe {
  width: 90%;
  height: 300px;
  display: block;
  margin-left: 5%;
  box-shadow: 0 0 0px 0px 0px 0px;
}

```

Attributs de style de l'iframe

Style du formulaire de dépôt de commentaires

```

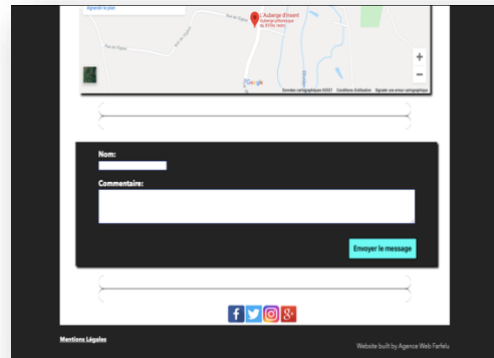
310
311 .form {
312   width: 90%;
313   padding: 10px;
314   margin-left: 4%;
315   background-color: #f2f2f2;
316   box-shadow: 0 0 0px 0px 0px 0px;
317   color: #333;
318 }
319
320 form div + div {
321   margin-top: 10px;
322 }
323
324 label { /div nom et message/
325   display: inline-block;
326   width: 100px;
327   border-radius: 10px;
328   font-size: larger;
329   font-weight: bold;
330 }
331
332 input, textarea { /text area name/
333   font: 1em sans-serif;
334   width: 200px;
335   box-sizing: border-box;
336   border: 1px solid #ccc;
337 }
338
339 input:focus, textarea:focus { /zone de texte focus/
340   border-color: #000;
341   background-color: #fff;
342   opacity: 0.9;
343 }
344
345 textarea {
346   vertical-align: top;

```

Nous avons maintenant notre base pour les pages secondaires.

Illustration : Pages secondaires





Responsive :

Un des enjeux de la mise en place d'un site internet est sa facilité d'accès et sa lisibilité sur les appareils mobiles.

Dans ce but nous avons commencés à travailler le CSS en utilisant la commande spécifique :

(@media queries)

```

402
403 @media all and (min-width: 320px) and (max-width: 670px){/*Responsive.....
404     .form{
405         width: 380px;
406     }
407
408     .butonform{
409         position: relative;
410         right: 120px;
411     }
412
413     .Contacts{
414         flex-flow: column-reverse;
415         align-items: center;
416     }
417
418     iframe{
419         width: 90%;
420     }
421
422     .liens{
423         display: flex;
424         justify-content: center;
425         width: 15%;
426     }
427
428     .liensAcc{
429         width: 210px;
430     }
431

```

L'utilisation de @media nous permet de définir les tailles maximales et minimales d'écrans (en pixels), en dessous ou au-dessus de ces tailles, le style utilisé sera le style contenu dans la balise @media ce qui nous permet l'adaptation des différents attributs à une taille d'écran spécifique.

Pour chacun des attributs de @media on obtient un visuel différent :

Illustration : Visuel du site lors de l'utilisation sur une tablette format portrait et sur mobile, on peut voir que le contenu est adapté au besoin de l'écran.



Conclusion de la partie HTML, CSS

1. Compétences mises en œuvre

Pour pouvoir réaliser les différentes parties j'ai dû utiliser différentes connaissances HTML, CSS que nous avons vues en cours afin de pouvoir réaliser un site qui soit complet sans animation ou dynamisme mais adaptable.

J'ai utilisé mes connaissances en CSS afin de parfaire le design du site et d'assurer la mise en place des différents modules de manière claire et précise.

2. Compétences validées dans cette partie

Cette partie du projet me permet de valider les compétences suivantes :

- *Maquetter une application*
- *Réaliser une interface utilisateurs web statique et adaptable.*

Ces compétences sont validées par la création du cahier des charges, de la maquette et de la mise en place des langages HTML, CSS.

3. Avantages de cette partie

Ce morceau de site nous constitue une base solide pour la mise en place d'animations dynamiques et de back-end sur le site sans avoir besoin de modifier tous les visuels.

Qui plus est l'utilisation du CSS afin de rendre le site responsive apporte la possibilité d'être consulté sur différents types d'appareils.

Partie dynamique : JavaScript

Une animation simple slider des pages auberges

N'étant pas satisfait de notre slider d'image en CSS nous avons décidé de le remplacer par un slider qui utilisera une fonction JavaScript afin d'optimiser l'animation mais aussi le rendu visuel.

Nous avons donc modifié le carrousel d'image présent sur chaque page d'auberge

Figures ci-dessous :

```

<div class="slideshow-container">
  <div class="mySlides fade">
    
    <div class="text">Hors d'Oeuvre</div>
  </div>
  <div class="mySlides fade">
    
    <div class="text">Hors d'Oeuvre</div>
  </div>
  <div class="mySlides fade">
    
    <div class="text">Hors d'Oeuvre</div>
  </div>
  <div class="mySlides fade">
    
    <div class="text">Plat</div>
  </div>
  <div class="mySlides fade">
    
    <div class="text">Plat</div>
  </div>
  <div class="mySlides fade">
    
    <div class="text">Plat</div>
  </div>
  <div class="mySlides fade">
    
    <div class="text">Dessert</div>
  </div>
  <div class="mySlides fade">
    
    <div class="text">Dessert</div>
  </div>
  <div class="mySlides fade">
    
    <div class="text">Dessert</div>
  </div>
</div>
<br>
<div style="text-align:center">
  <span class="dot" onclick="currentSlide(1)"></span>
  <span class="dot" onclick="currentSlide(2)"></span>
  <span class="dot" onclick="currentSlide(3)"></span>
  <span class="dot" onclick="currentSlide(4)"></span>
  <span class="dot" onclick="currentSlide(5)"></span>
  <span class="dot" onclick="currentSlide(6)"></span>
  <span class="dot" onclick="currentSlide(7)"></span>
  <span class="dot" onclick="currentSlide(8)"></span>
  <span class="dot" onclick="currentSlide(9)"></span>
</div>

```

Puis nous y avons créé un fichier JavaScript pour la gestion de l'animation slide

Figure ci-dessous :

```

1  var slideIndex = 1;
2  showSlides(slideIndex);
3
4  function plusSlides(n) {
5    showSlides(slideIndex += n);
6  }
7
8  function currentSlide(n) {
9    showSlides(slideIndex = n);
10  }
11
12 function showSlides(n) {
13   var i;
14   var slides = document.getElementsByClassName("mySlides");
15   var dots = document.getElementsByClassName("dot");
16   if (n > slides.length) {slideIndex = 1}
17   if (n < 1) {slideIndex = slides.length}
18   for (i = 0; i < slides.length; i++) {
19     slides[i].style.display = "none";
20   }
21   for (i = 0; i < dots.length; i++) {
22     dots[i].className = (property) Element.className: string
23   }
24   slides[slideIndex - 1].style.display = "block";
25   dots[slideIndex - 1].className += " active";
26   slideIndex++;
27   if (slideIndex > slides.length) {
28     slideIndex = 1
29   }
30   slides[slideIndex - 1].style.display = "block";
31   setTimeout(showSlides, 4000);
32 }

```


3. Back-End : PHP, MySQL

Partie 1 : PHP

Dans cette partie, nous allons tout d'abord revu chacun de nos fichiers HTML vers PHP afin de pouvoir rendre notre site apte à recevoir du back-end, puis nous avons retiré les doctypes des pages pour y importer le fichier header qui contient les métadonnées et notre menu importé par une fonction [getPart()], ainsi que la liaison au fichier de style de nos pages puis le footer qui lui contient les différents liens et scripts contenu dans le pied de page (footer.php).

Une fois nos extensions modifiées nous allons maintenant séparer notre Template de base en deux fichiers distincts (header.php et footer.php) que nous pourrions inclure dans chaque page du site en les appelant via la fonction PHP (Include).

Mais avant toute chose il nous a fallu procéder à la création d'un environnement back-end sans quoi nous ne serions plus en mesure d'afficher notre site j'expliquerais cette partie dans une annexe.

(Cf : Voir annexe installation BDD)

Ensuite nous allons opérer quelques modifications afin de mieux adapter le site au back-end qu'on y ajoutera.

Pour commencer nous ajouterons à notre Template de base un second menu que l'on disposera en haut à droite du document afin d'optimiser l'ergonomie du site, puis en haut à gauche, une icône de toque qui servira de lien de retour vers l'accueil et nous ajoutons des icônes fa fa afin d'illustrer notre menu secondaire.

Nous modifions le style conteneur secondaire (div .repert) pour remplacer le background de couleur blanche d'origine par une couleur noire.

Nous allons également créer et connecter une base de données du site afin de pouvoir stocker et gérer les commentaires et différentes données du site puis nous procéderons à la connexion du site à cette base de données en utilisant une fonction PHP (détails annexe installation de PBM)

Nous ajouterons une interface de réservations et de commandes de plats qui permettront aux visiteurs de réserver une table pour un repas ou encore de commander et venir chercher leur commande via un système de click and collect.

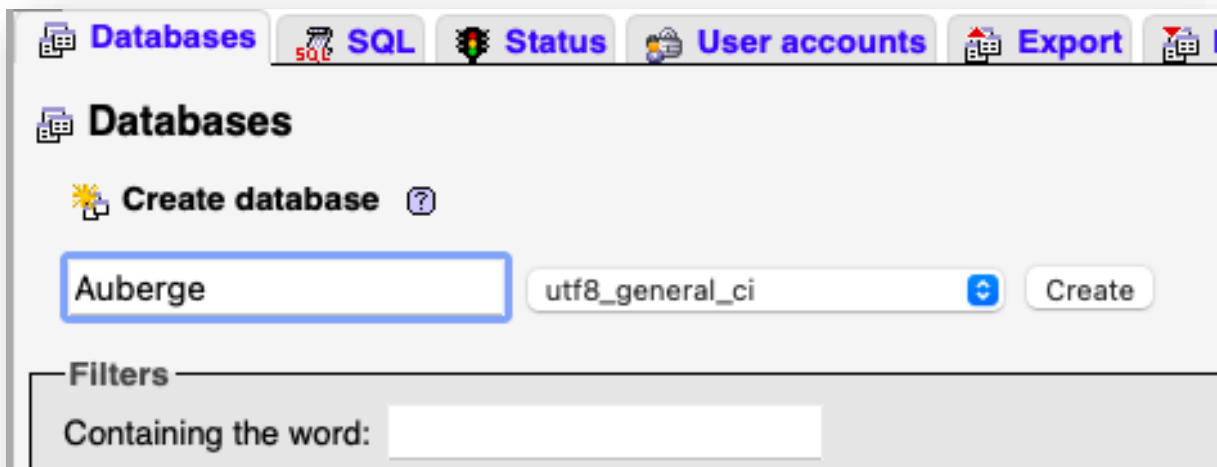
Puis afin de réduire le nombre de lignes de codes dans nos documents nous allons enregistrer les données d'établissements dans la base de données.

Nous allons maintenant modifier notre page d'accueil afin d'en alléger le code, en retirant tous les conteneurs ajoutés lors de la partie front-end et nous les remplacerons par un seul conteneur, imbriqué dans une requête préparée afin de pouvoir répéter le conteneur autant de fois qu'il n'y a de résultat dans la base de données.

PHP ira récupérer les données dans la base de données et clonera le conteneur en fonctions du nombre de résultats obtenus lors de l'appel de données

Étape 1 : Mise en place de l'environnement back-end :

A. Création de la base de données



B. Connexion base de données

```

Include > connect_mysql.php > ...
1  <?php
2      try{
3
4          $pdo = new PDO("mysql:host=localhost;dbname=auberge",'Nguma','nguma');
5          $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
6          $pdo->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_OBJ);
7      }catch(Exception $e){
8          echo 'erreur : ' . $e->getMessage(). '<br>';
9          echo 'N° : ' . $e->getCode();
10     }
11     ?>

```

Ici, nous allons travailler en utilisant une méthode orientée objet nous utilisons donc un try de connexion qui contient les données de connexion à la base de données, puis nous ajoutons en fin de code un script catch qui demande à PHP de récupérer et d'afficher les erreurs en cas de non-fonctionnement.

C. Appel du fichier de connexion Base de données

Ici avec ce code PHP, nous pouvons appeler nos différents fichiers on appelle le header avec **la fonction include**, puis on requiert le fichier de connexion base de données avec la fonction **require**.

```

Lecoindesbonnesauberges.php > html > head >
1  <?php
2  include "../Include/header.php";
3  require "../Include/connect_mysql.php";
4  ?>

```

*Ces deux fonctions d'apparence semblables le sont sur presque tous les points à la différence que l'une d'elle (**require**) mettra fin à l'exécution du code en cas de non-fonctionnement, tandis que l'autre (**include**) ne bloquera pas l'exécution de ce code.*

Optimisation de notre Template

A. Header.php

Le fichier header.php, est un fichier qui contient la partie initiale de la balise HTML, les fameuses métadonnées auxquelles j'ai fait allusion plus haut, ainsi que la balise `head` et la balise `body` et le code nécessaire à la mise en place du menu.

Illustrations : Les codes du fichier header.php

```

1 <?php
2 ?>
3 <!DOCTYPE html>
4 <html lang="en">
5 <head>
6 <meta charset="UTF-8">
7 <meta name="viewport" content="width=device-width, initial-scale=1.0">
8 <link rel="stylesheet" href="/Lecoindesbonnesauberges.css">
9 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
10 <style type="text/css">
11 .Title{
12 display: block;
13 text-align: center;
14 padding-top: 10px;
15 }
16
17 .Report{
18 width: 100%;
19 display: flex;
20 flex-wrap: wrap;
21 justify-content: space-around;
22 background-color: white;
23 margin: auto;
24 padding-top: 30px;
25 }
26
27 .Welcome{
28 width: 100%;
29 height: 100px;
30 color: black;
31 text-align: center;
32 font-size: xx-large;
33 font-weight: bolder;
34 }
35
36 .LiensAcc{
37 width: 400px;

```

```

92 <body>
93 <div class="navbar1">
94 <a href="/Lecoindesbonnesauberges.php">
95 <a class="active1" href="Connexion.php"><i class="fa fa-fw fa-user"></i> Connexion</a>
96 <a class="active1" href="#"><i class="fa fa-fw fa-search"></i> Recherche</a>
97 <a class="active1" href="Contacts.php"><i class="fa fa-fw fa-envelope"></i> Contact</a>
98 <a class="active1" href="/Reserv/reservations.php"><i class="fa fa-fw fa-home"></i> Reservations</a>
99 </div>
100 <nav role="navigation">
101 <div id="menuToggle">
102 <input type="checkbox" />
103 <span></span>
104 <span></span>
105 <span></span>
106 <ul id="menu">
107 <a href="/Lecoindesbonnesauberges.php"><li>Accueil</li></a>
108 <a href="aubergeincent.php"><li>Auberge d'Incent</li></a>
109 <a href="aubergedesetangs.php"><li>Auberge des Etangs</li></a>
110 <a href="aubergeducronquelet.php"><li>Auberge du Cronquelet</li></a>
111 <a href="aubergedumoulinet.php"><li>Auberge du Moulinet</li></a>
112 <a href="aubergeduvieuxlogis.php"><li>Auberge du Vieux Logis</li></a>
113 <a href="lapetiteauberge.php"><li>La petite Auberge</li></a>
114 <a href="aubergeaubonaccueil.php"><li>Auberge Au Bon Accueil</li></a>
115 <a href="aubergedeladuneauloup.php"><li>Auberge de La Dune Au Loup</li></a>
116 <a href="aubergelescomptesdelabecasse.php"><li>Auberge Les Comptes De La Becasse</li></a>
117 <a href="Contacts.php"><li>Contact</li></a>
118 <form action="" method="POST"><label for="Rechercher"></label>
119 <textarea name="Rechercher" id="" cols="15" rows="1" placeholder="Rechercher"></textarea><input type="submit" value="Rechercher"></form>
120 </li>
121 </ul>
122 </div>
123 <div class="LiensAcc">
124 <a href="#" class="facebook"><i class="fa fa-facebook"></i></a>
125 <a href="#" class="twitter"><i class="fa fa-twitter"></i></a>
126 <a href="#" class="google"><i class="fa fa-google"></i></a>
127 <a href="#" class="instagram"><i class="fa fa-instagram"></i></a>
128

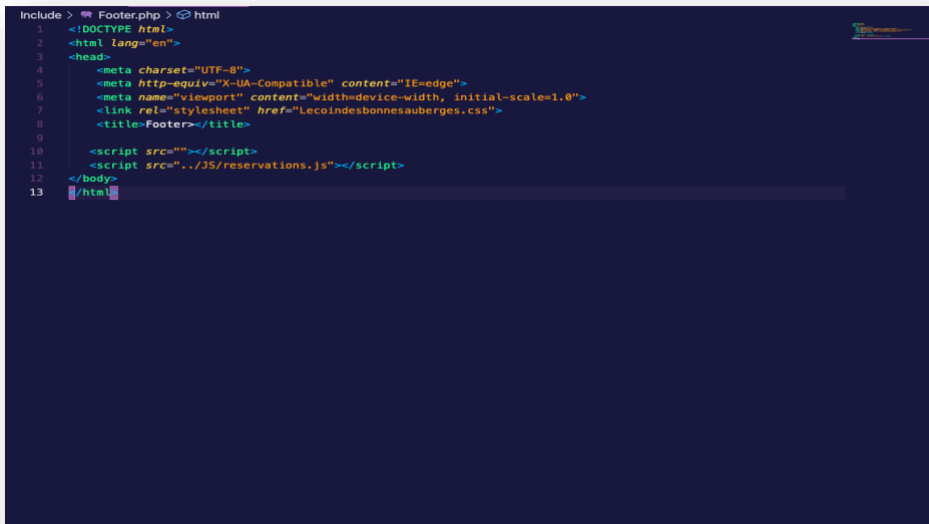
```

Code comprenant le menu qui sera présent sur chaque page.

B. Footer.php

Footer.php, est la partie qui contient les balises de fermeture des tags HTML et body c'est aussi la partie où seront appelées les différents scripts nécessaires au fonctionnement du JavaScript.

Illustrations :

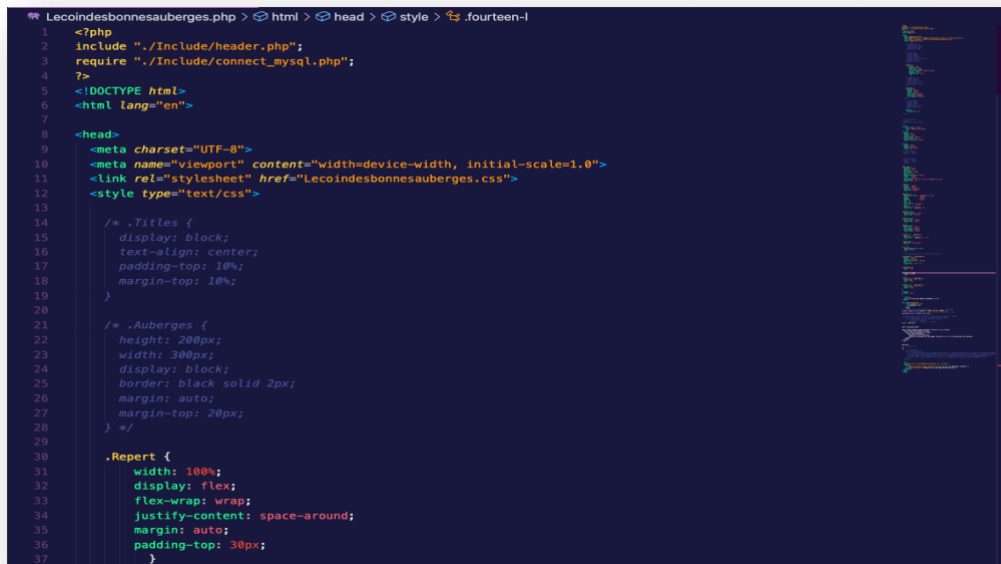


```

1 </html>
2 </body>
3
4 <script src=""></script>
5 <script src=""></script>
6
7 </html>
8
9
10
11
12
13

```

En appelant ces fichiers on obtient notre Template de base pour toutes nos pages.



```

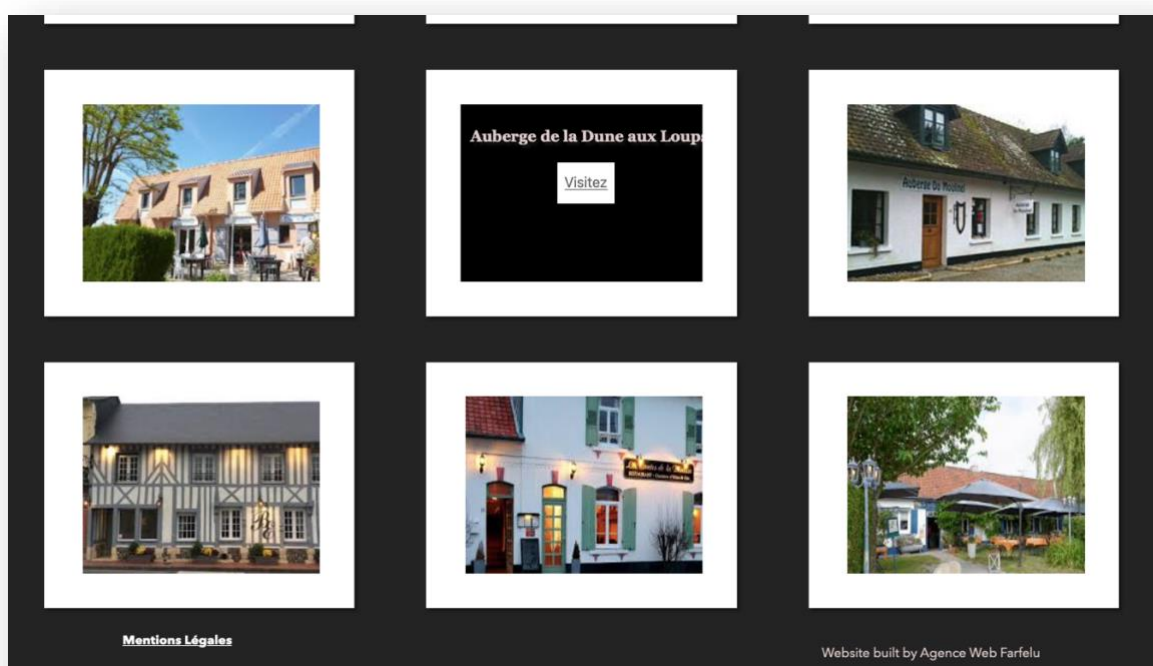
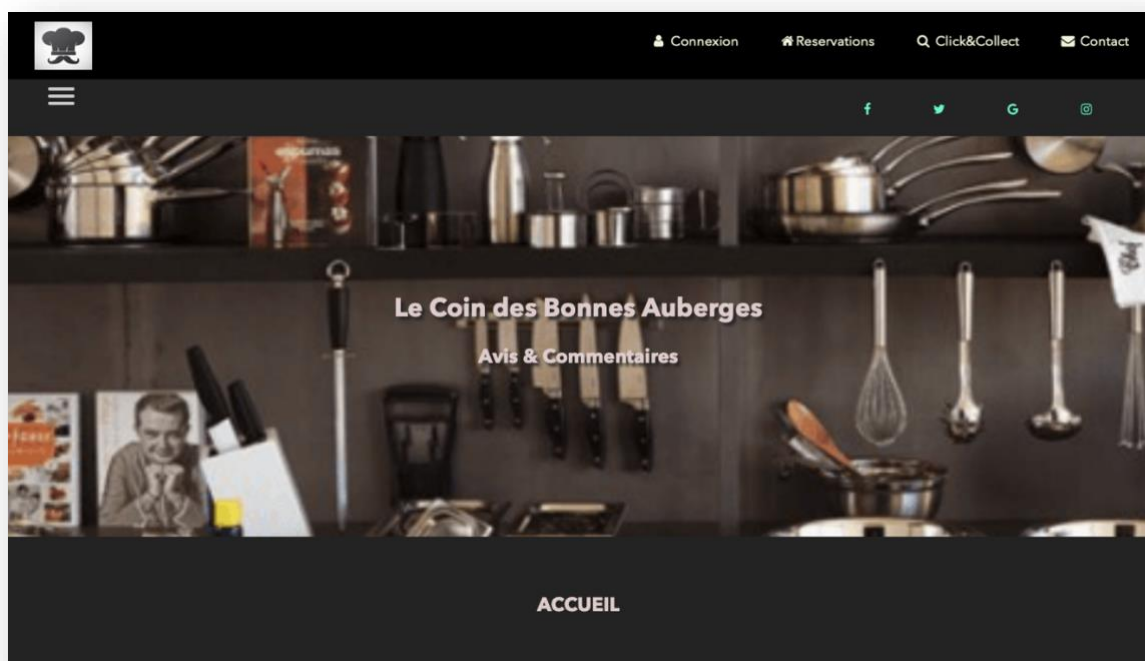
1 <?php
2 include "../Include/header.php";
3 require "../Include/connect_mysql.php";
4 ?>
5 <DOCTYPE html>
6 <html lang="en">
7
8 <head>
9
10 <meta charset="UTF-8">
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
12 <link rel="stylesheet" href="Lecoindesbonnesauberges.css">
13 <style type="text/css">
14
15 /* .Titles {
16     display: block;
17     text-align: center;
18     padding-top: 10px;
19     margin-top: 10px;
20 }
21
22 /* .Auberges {
23     height: 200px;
24     width: 300px;
25     display: block;
26     border: black solid 2px;
27     margin: auto;
28     margin-top: 20px;
29 } */
30
31 .Report {
32     width: 100%;
33     display: flex;
34     flex-wrap: wrap;
35     justify-content: space-around;
36     margin: auto;
37     padding-top: 30px;
38 }
39

```

De plus ce Template nous permet d'avoir des pages plus claires et d'éviter une surcharge de balises lors du codage.

On obtient donc un visuel identique sur chaque page de site ce qui nous permet donc une homogénéité au niveau graphique.

Illustration : Rendu de la nouvelle Template



La séparation de l'architecture du site avec les données contenues nous a permis de faciliter l'insertion de différentes balises de style en CSS et de surcroit de faciliter la gestion des contenus.

Modification du code de la page d'accueil

Comme je l'explique plus haut nous avons retiré tous les conteneurs dans lesquels on retrouvait nos images etc, pour les remplacer par un seul conteneur imbriqué dans un code PHP qui envoie une requête au serveur afin de pouvoir retourner les données, puis créer une boucle qui exécutera la requête jusqu'à ce qu'elle ait trouvé toutes les données existantes.

Illustration :

Dans la figure ci-contre on ouvre une balise PHP dans laquelle on commence par déclarer une variable qui a pour paramètre une requête SQL.

Puis nous créons une boucle qui affichera grâce à la fonction PHP fetch ces données sous forme de tableau et s'arrêtera seule une fois qu'elle aura affiché toutes les données disponibles dans la table.

```

204
205 <div class="Repert">
206   <div class="Welcome">
207     <h2>ACCUEIL</h2>
208   </div>
209   <?php
210   // requête de la table Auberge pour avoir le nom et l'image
211   $req3 = $pdo->query("SELECT * FROM auberges WHERE 1");
212   // Boucle de la requête pour avoir une seule auberge par items
213   while($article = $req3->fetch()){
214
215   // class items est le recto de la carte avec l'image de l'auberge
216   // dans le background on met la variable article image
217   // class=details est le verso de la carte
218   // h3 nom de l'auberge
219   // renvoie à la page de l'auberge
220   echo <<<ARTICLE
221
222
223   <div class="bg-items">
224
225   <div class="items"><img id="anim" src={$article->image}>
226   <div class="fourteen-l"></div>
227   <div class="fourteen-r"></div>
228   <div class="details">
229     <h3>{$article->nom}</h3>
230     <button id="anim1"><a id="page" href={$article->lien2}>Visitez</a></button>
231   </div>
232   </div>
233   </div>
234
235   ARTICLE;
236
237   // fin de boucle
238   };
239   ?>
240

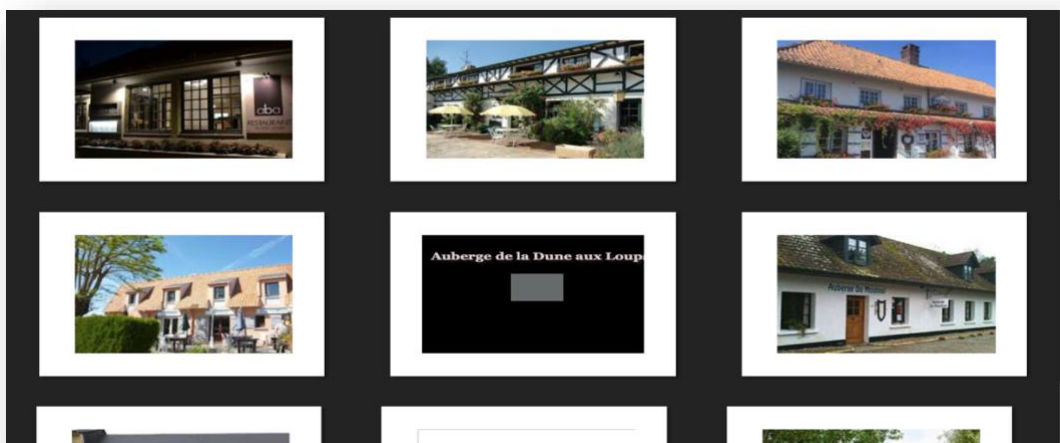
```

```

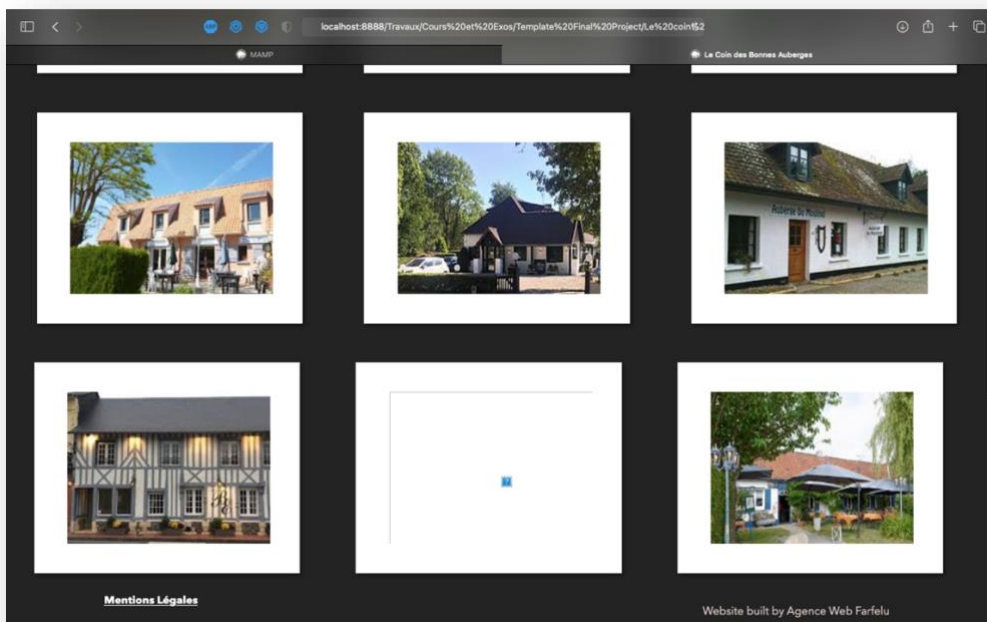
<div class="Repert">
  <div class="Welcome">
    <h2>ACCUEIL</h2>
  </div>
  <div class="ImgAub"><a href="Auberge d'Inxent.html"></a>
  <div class="ImgAub"><a href="Auberge des Etangs.html"></a>
  <div class="ImgAub"><a href="Auberge du Cronquelet.html"></a>
  <div class="ImgAub"><a href="Auberge du Moulinel.html"></a>
  <div class="ImgAub"><a href="Auberge du Vieux Logis.html"></a>
  <div class="ImgAub"><a href="La Petite Auberge.html"></a>
  <div class="ImgAub"><a href="Auberge Au Bon Accueil.html"></a>
  <div class="ImgAub"><a href="Auberge de La Dune au Loup.html"></a>
  <div class="ImgAub"><a href="Auberge Les Comptes De La Becasse.html"></a>
  <hr>
  <div class="liensAcc">
    <a href="https://www.facebook.com/"></a>
    <a href="https://twitter.com/?lang=fr"></a>
    <a href="https://www.instagram.com/"></a>
    <a href="https://plus.google.com/"></a>
  </div>
</div>

```

Dans la figure ci-contre on peut voir le code qui affichait précédemment les établissements en page d'accueil.

Illustration : Rendu du nouvel affichage de données

Grâce à notre fichier `footer.php` qui est appelé sur la page nous avons la possibilité d'ajouter sur chacune de nos pages différents liens comme les mentions légales mais aussi un petit texte indiquant le développeur du site.

Illustration : Rendu de l'affichage des données bis

Pour compléter la mise en page du HTML et donner un visuel graphique au site on utilise le CSS, langage de style qui permet de modifier l'apparence des différentes balises HTML et de leur donner des informations comme la taille, la couleur etc.

Interface de réservations

Dans cette partie, nous allons créer une interface de réservations pour nos auberges, cet espace offre aux clients la possibilité de réserver une table pour un repas ou encore de commander différentes formules et venir les récupérer sur place.

Pour cela nous avons eu besoin de créer des nouveaux fichiers, reservation.php, controlreserv.php, et function.php.

- Le fichier controlreserv.php sera le fichier qui vérifiera et exécutera le formulaire si aucune erreur n'est détectée lors de la validation du formulaire par le bouton submit.
- Le fichier function.php lui se chargera de valider et sécuriser les données.

Une fois ces étapes remplies, nous commencerons par y appeler nos différents fichiers (header et connectMySQL) via les fonctions include :

```
<?php
include "../Include/header.php";
?>
```

Puis nous pourrions insérer notre formulaire de réservations :

```
<?php
include "../Reserv/controlreserv.php";
?>
<div style="overflow-x:auto;">
<fieldset>
  <form action="../controlreserv.php" method="POST">
    <label for="nom">Nom</label>
    <input type="text" name="nom" id="nom" required>
    <br>
    <label for="prenom">Prénom</label>
    <input type="text" name="prenom" id="prenom" required>
    <br>
    <label for="email">Adresse mail</label>
    <input type="text" name="email" id="mail" required>
    <br>
    <label for="phone">Numero de téléphone</label>
    <input type="num" name="phone" id="phone" required>
    <br>
    <label for="adresse">Adresse</label>
    <input type="text" name="adresse" id="adresse" required>
    <br>
    <label for="ville">Ville</label>
    <input type="text" name="ville" id="ville" required>
    <br>
    <label for="cp">Code postal</label>
    <input type="num" name="cp" id="cp" required>
    <br>
    <label for="pays">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;& Pays</label>
    <select name="pays" id="firstname" style="width:185px;" >
      <option value="French">France</option>
      <option value="Belgium">Belgique</option>
      <option value="Holland">Pays-bas</option>
      <option value="England">Angleterre</option>
      <option value="Deutschland">Allemagne</option>
    </select>
  </form>
```

Comme vous pouvez le constater notre formulaire de réservation nécessite de nombreux champs compte tenu des différents contextes possibles. Pour le choix de la date de la réservation nous avons utilisé un objet jquery qui nous permet d'afficher un calendrier avec sélection de l'heure

Mais celui-ci récupère également des données un peu plus précises sur le client notamment les coordonnées postales, le pays d'origine.

```
<label for="auberge">&nbsp;&nbsp;&nbsp;&nbsp;& Choisir une auberge <span class="requis">*</span></label>
<select name="auberge" id="room" class="fl-select" style="width:295px;" required >
  <option selected="selected" value="">Choisir une auberge</option>
  <option value="aubergeinvent">Auberge d'Inxent</option>
  <option value="aubergedesetangs">Auberge des Etangs</option>
  <option value="aubergevieuxlogis">Auberge du Moulinet</option>
  <option value="aubergevieuxlogis">Auberge du Vieux Logis</option>
  <option value="petiteauberge">La Petite Auberge</option>
  <option value="aubergebonaccueil">L'Auberge du Bon Accueil</option>
  <option value="aubergeduneauxlous">Auberge de la Dune Aux Loups</option>
  <option value="aubergecomtesbecasse">Auberge Les Comtes de La Bécasse</option>
</select>
<br>
<label for="nbredepersonnes">Nombre de personnes</label>
<input type="number" min="1" max="10" class="fl-date" name="nbredepersonnes" id="nbredepersonnes" place
<label for="datetimepicker">&nbsp;&nbsp;&nbsp;&nbsp;& Date <span class="requis">*</span></label>
<input type="text" id="datetimepicker" name="datetimepicker">
<br>
<label for="formule">&nbsp;&nbsp;&nbsp;&nbsp;& Formule</label>
<select name="formule" id="formule" style="width:185px;" >
  <option selected="selected" value="">Choisir une formule</option>
  <option value="aemporter">A Emporter</option>
  <option value="surplace">Sur Place</option>
</select>
<br>
<label for="resmessage">&nbsp;&nbsp;&nbsp;&nbsp;& Message</label>
<textarea class="fl-textarea" rows="10" cols="40" placeholder="Votre message" name="resmessage" id="res
<br>
<input type="submit" name="envoyer" id="item-15" class="fl-submit" style="width:150px;" value="RÉSERVER
<br>
<input type="reset" id="item-16" class="fl-reset" style="width:150px;" value="Annuler" />
<br>
</form>
```

La bibliothèque jquery est une application qui permet de générer et d'utiliser rapidement et très facilement des fonctions Javascript, pour cela il nous faut nous rendre sur le site jquery et y rechercher l'objet requis, puis on télécharge l'objet et récupérons les liens de fichiers de style une fois ces fichiers récupérés il ne reste qu'à les intégrer au site et à lier les différents scripts nécessaires au fonctionnement.

Ce procédé nous a permis d'intégrer rapidement un datetimepicker qui apparaît lors du clic sur le champ date.

Illustration : fonction datetimepicker :

```
<label for="datetimepicker">&nbsp;&nbsp; Date <span class="requis">*</span></label>
<input type="text" id="datetimepicker" name="datetimepicker">
```

Ci-dessus on ajoute les label et input datetimepicker.

```
<?php
include "../Include/header.php";
?>
<link rel="stylesheet" href="../Lecoindesbonnesauberges.css">
<script src="../../datetimepicker-jquery-plugin/js/jquery.min.js"></script>
<link rel="stylesheet" type="text/css" href="../../datetimepicker-jquery-plugin/css/j
<script src="../../datetimepicker-jquery-plugin/js/jquery.datetimepicker.js"></script>
```

Ici nous lions les fichiers de styles de plugin et les scripts du datetimepicker

Ici notre propre fichier JavaScript dans lequel on sélectionnera l'id du champ qui recevra la fonction.

```
<script src=""></script>
<script src="../../JS/reservations.js"></script>
</body>
</html>
```

JS > JS reservations.js > ...

```
1 $(document).ready(function() {
2     $('#datetimepicker').datetimepicker();
3 });
4 $('#datetimepicker').datetimepicker({
5     format:'Y-m-d H:i',
6     inline:true
7 });
8
```

Ci-contre le fichier JavaScript qui récupère l'élément sur lequel il va appliquer la fonction par son ID et donne le style de l'objet à afficher.

Maintenant que nous avons notre page de réservation nous allons pouvoir éditer le fichier d'exécution du formulaire.

Ce fichier vérifiera un à un les champs du formulaire afin de s'assurer que ceux-ci ont tous bien été remplis.

Illustration :

```

1  <?php
2
3  <?php
4  if(!empty($_POST)){
5      $errors = array();
6
7      if(empty($_POST['nom'])){
8          $errors = 'Remplir le champ nom';
9      }
10     elseif(empty($_POST['prenom'])){
11         $errors = 'Remplir le champ prenom';
12     }
13     elseif(empty($_POST['email'])){
14         $errors = 'Remplir le champ email';
15     }
16     elseif(empty($_POST['phone'])){
17         $errors = 'Remplir le champ téléphone';
18     }
19     elseif(empty($_POST['adresse'])){
20         $errors = 'Remplir le champ adresse';
21     }
22     elseif(empty($_POST['ville'])){
23         $errors = 'Remplir le champ ville';
24     }
25     elseif(empty($_POST['cp'])){
26         $errors = 'Remplir le champ code postal';
27     }
28     if(empty($_POST['pays'])){
29         $errors = 'Remplir le champ pays';
30     }
31     elseif(empty($_POST['cp'])){
32         $errors = 'Remplir le champ code postal';
33     }
34     if(empty($_POST['pays'])){
35         $errors = 'Remplir le champ pays';
36     }
37     elseif(empty($_POST['auberge'])){
38         $errors = 'veuillez choisir une auberge';
39     }
40     elseif(empty($_POST['nbredepersonnes'])){
41         $errors = 'Choisir un nombre de personnes';
42     }
43     elseif(empty($_POST['datetimepicker'])){
44         $errors = 'Veuillez choisir une date';
45     }
46     elseif(empty($_POST['formule'])){
47         $errors = 'Veuillez choisir une formule';
48     }
49     elseif(empty($_POST['resmessage'])){
50         $errors = '';
51     }
52 }

```

Dans le cas où un champ n'était pas rempli l'utilisateur sera invité à remplir le champ en question.

Une fois les données vérifiées il exécutera la préparation des données en vue de leur enregistrement, enregistrera les données dans la table définie dans la requête php puis l'utilisateur sera renvoyé vers une page de confirmation sur laquelle on affichera un message **"réservation effectuée"** grâce à un echo et afficherons les détails de la réservation avec des liens permettant de la modifier.

Un message d'erreur s'affichera pour indiquer ce qui n'a pas fonctionné.

Figure : Enregistrement des données (Requête prépar

```

require "../Include/connect_mysql.php";
$req = $pdo->prepare("INSERT INTO reservations SET nom = ?, prenom = ?, email = ?, phone = ?, adresse = ?, ville = ?, cp = ?, pays = ?, au
$req->execute([$POST['nom'], $POST['prenom'], $POST['email'], $POST['phone'], $POST['adresse'], $POST['ville'], $POST['cp'], $POST
exit(header('location: ../confreserv.php'));
}

```

Figure : Affichage des erreurs

```

if(isset($errors)){
    echo'<font color=red>'. $errors. '</font>';
    header('location: reservations.php');
    exit;
}
?>

```

Aperçu de l'exécution d'une réservation :

Figure 1 : Page de réservations

Code postal : 24256 CHOISISSEZ LE CLICK & COLLECT

Pays : France

Choisir une auberge : * Auberge des Etangs

Horaires des services : 21h

Nombre de personnes : 2

Date*

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Message : Soirée d'anniversaire

RÉSERVER Annuler

Fig. 2 : Confirmation de la réservation

Le Coin des Bonnes Auberges
Avis & Commentaires

Gérez votre Réservation

Tableau de vos reéservations

ID :	Nom :	Date :	Auberge :	Service :	Table :	Modifier :
4	Dupont	2016-11-18 21:00:00	aubergedesetangs	6	Table N° :	Modifier :

[Mentions Légales](#)

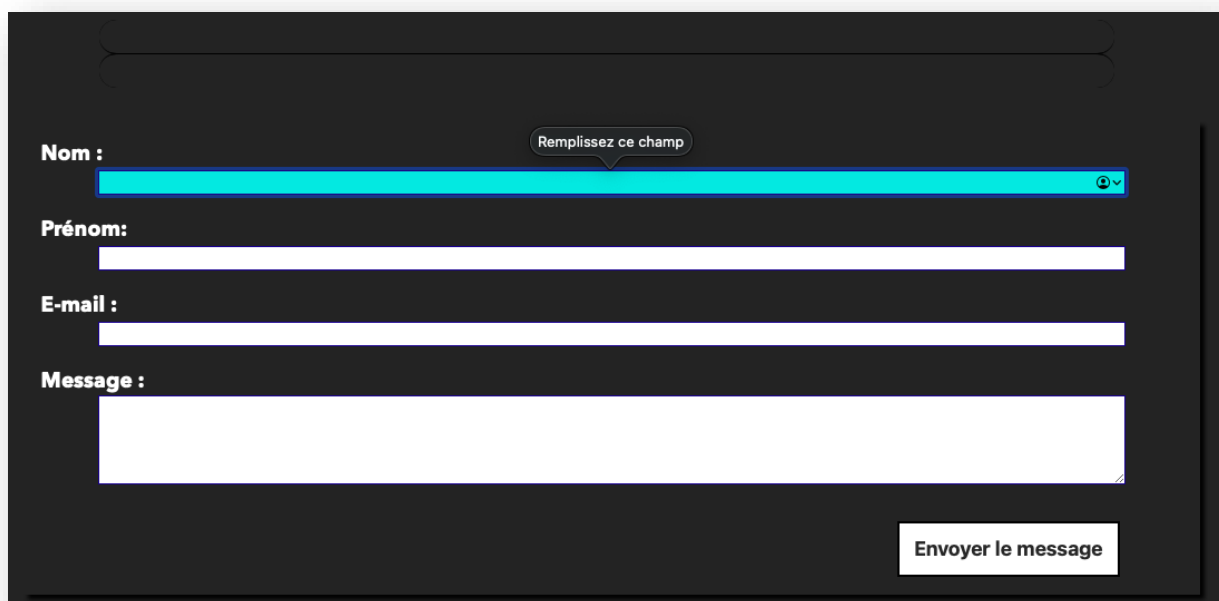
Website built by Agence Web Farfelu

Validations et sécurisation des données :

Afin d'optimiser la sécurisation des données reçues comme nous le détaillons plus haut, nous procédons à deux vérifications, nous effectuons d'abord une première vérification directement sur le formulaire en utilisant le paramètre **required** sur chacun des **inputs**.

Cette fonction empêche l'envoi du formulaire si l'un des champs n'était pas rempli et affiche un message qui demande à l'utilisateur de remplir le champ en question.

Figure :



The image shows a contact form on a dark background. The form contains four input fields: 'Nom :', 'Prénom:', 'E-mail :', and 'Message :'. The 'Nom :' field is highlighted in red, indicating it is required and has not been filled. A red tooltip with the text 'Remplissez ce champ' (Fill this field) points to the 'Nom :' label. The 'Prénom:' field is empty. The 'E-mail :' field is empty. The 'Message :' field is empty. At the bottom right of the form is a button labeled 'Envoyer le message' (Send the message).

Modélisation de la base de données

Un schéma de base de données représente la configuration logique d'une base de données relationnelle.

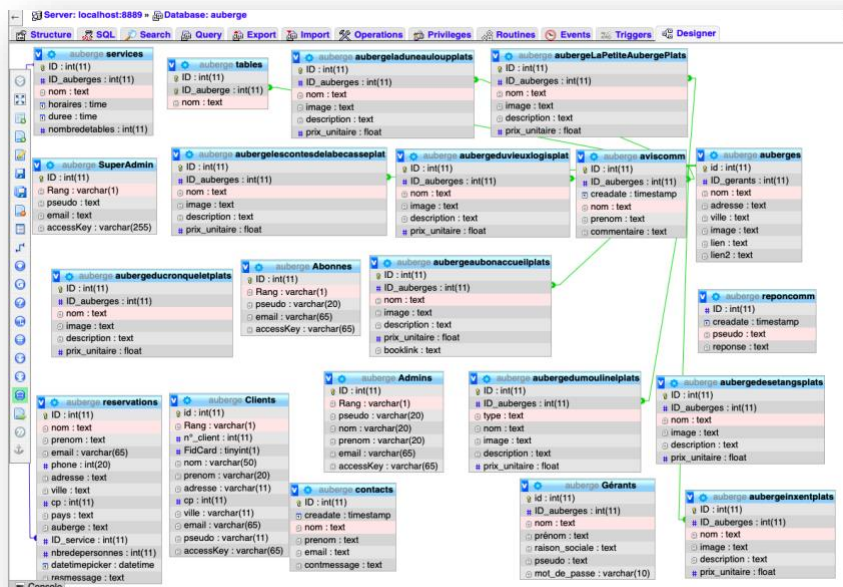
Il peut se présenter à la fois sous forme d'une représentation visuelle d'un ensemble de formules (**appelées contraintes d'intégrité**) qui régissent une base de données.

Ces formules sont exprimées dans un langage de définition des données, comme le langage SQL.

Il peut être simple, sous forme de diagramme **UML** ou encore de **MCD** (modèle conceptuel de données)

Le diagramme **UML** et le **MCD** peuvent ressembler de près au schéma simple à la différence qu'eux sont légendés des notions de cardinalités.

Figure : Schéma de la base de données Le coin des bonnes auberges



Fg 1 : Ci-contre, on peut voir un schéma simple des relations entre les tables de la base de données.

Fg 2 : Exemple de contrainte d'intégrité

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change X Drop More
2	ID_gerants	int(11)			Yes	NULL			Change X Drop More
3	nom	text	utf8_general_ci		Yes				Change X Drop More
4	adresse	text	utf8_general_ci		No				Change X Drop More
5	ville	text	utf8_general_ci		No				Change X Drop More
6	image	text	utf8_general_ci		No				Change X Drop More
7	lien	text	utf8_general_ci		No				Change X Drop More
8	lien2	text	utf8_general_ci		No				Change X Drop More

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY		BTREE	Yes	No	id	9	A	No	
owner		BTREE	No	No	ID_gerants	1	A	Yes	

Les cardinalités sont des couples de valeurs que l'on trouve entre chaque entité et ses associations liées.

Donc, pour une association de deux entités, il y a quatre cardinalités à indiquer (2 de chaque côtés).

Il y a pour ça 3 valeurs : 0, 1 et N qui signifie plusieurs.

Pour les associations de 2 entités, ce sont des valeurs qui permettent d'indiquer combien de fois au minimum et au maximum le champ d'une entité peut être lié à un autre champ.

Exemple :

Dans le cas de notre base de données, on a une liaison entre les tables auberges et Gérants.

- Dans la table auberges on trouve un champ [ID_gérants]
- Dans la table Gérant on trouve [ID_auberge]

Leurs cardinalités respectives seraient :

Auberges		Gérants	
Minimum	Maximum	Minimum	Maximum
1	N	1	N

Ce qui signifie clairement que les tables auberges et Gérants peuvent à l'avoir eu minimum une relation et au maximum plusieurs relations l'une avec l'autre.

Plus précisément une auberge peut avoir au minimum un gérant (1) et au maximum plusieurs gérants (N), comme un gérant peut gérer une auberge au minimum (1) et plusieurs au maximum (N).

N'ayant pas de logiciel comme Workbench pour je n'ai pu joindre un aperçu de diagramme UML ou encore de représentations MCD j'en prendrai donc une vue en cours.

Annexe 1 : Cahier des charges le coin des bonnes auberges