



# PROJET LIUGO

Dossier professionnel

PRÉSENTÉE PAR  
Mathias Cabrol

MARS 2022



# Sommaire

## INTRODUCTION

- 03 Résumé
- 04 Cahier des charges
- 05 Environnement de travail
- 08 Compétences couvertes
- 09 Organisation du projet
- 11 Maquettage



## TECHNIQUE

- 12 Schéma conceptuel de BDD
- 13 Composer
- 15 Design Pattern - MVC
- 19 Requêtes AJAX
- 21 Responsive Design



## SÉCURITÉ

- 24 Failles SQL
- 25 Cross-site Request Forgery
- 26 Cross-site Scripting



## FONCTIONNALITÉS

- 27 Personalisation
- 28 Système de réservation
- 29 CRUD



## RECHERCHE

- 33 Get Element JS

## CONCLUSION

- 34 Fonctions supplémentaires
- 38 Pistes d'amélioration

## Maquettage

Imagination du design du projet et adaptation côté responsive. Recherche d'images, polices d'écriture...

## Sécurité

Documentation sur les différents types d'attaques et mise en place d'une protection.

## Personalisation

Formulaires dynamiques et personalisation de l'affichage du client par le professionnel.

## La suite

Modification et amélioration de l'existant.  
Ajout de fonctionnalités et amélioration de l'UX.

---

## IDÉE

Le but de ce projet est de palier aux besoins de communication entre les clients vacanciers et les professionnels qui les accueillent en leur fournissant un outil qui répertorie toutes les activités possibles dans leur lieu de vacance.

---

## OUTIL

Le projet doit contenir certaines fonctionnalités clés :

- Possibilité de création de trois types de comptes différents selon l'entité.
  - Possibilité de personnalisation de l'affichage présenté au client.
  - Système de réservation
- 

## USAGE

Cette application est pensée pour trois types de clientèle :

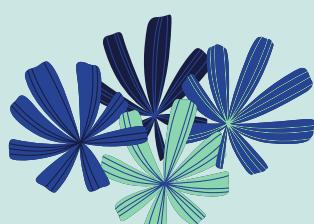
- Le personnel de réception qui aura une charge de travail réduite.
  - Les prestataires de service extérieurs, qui pourront accroître leur visibilité et proposer leurs services à l'arrivée des clients.
  - Les clients, qui pourront trouver toutes les informations nécessaires centralisées.
- 

## BUT

Le but final est permettre une meilleure gestion des réservations de leurs clients par les Hôtels et les prestataires.

Mais aussi d'avoir un suivi, et donc d'enranger des commissions basées sur le chiffre d'affaire apporté.

Cela permettra également d'orienter d'une meilleure manière les vacanciers arrivant pour la première fois dans un lieu inconnu.





## Les Docks

**143 Rue Yves le Coz,  
78000 Versailles**

**1er Janvier 2022**

**Bonjour,**

Nous souhaitons créer une application permettant aux utilisateurs de flasher un QR Code pour y accéder. Ce QR code devra être personnalisé en fonction de l'établissement auquel il réfère, en effet il sera flashé dans le cadre d'une arrivée dans un établissement Hôtelier.

En accédant à l'application, l'utilisateur devra avoir accès à un espace de bienvenue détaillant les différents services de l'Hôtel, ceux-ci devront être personnalisables par l'établissement via un accès sur un portail professionnel. Les utilisateurs pourront se connecter en tant qu'invité, mais également créer un compte et se connecter si ils souhaitent avoir accès à leur historique.

Le client devra également avoir la possibilité de réserver des activités proposées par des prestataires extérieurs à l'établissement, sur un page dédiée. Il doit pouvoir effectuer une recherche (qu'il pourra filtrer par secteur d'activité) et réserver directement via l'application ces activités. Il pourra également annuler à tout moment si nécessaire.

Ces activités seront proposées par des prestataires qui devront également avoir un accès à l'espace pro, ils pourront y créer des activités à mettre en vente sur l'application, définir les prix, les horaires, ajouter une photo d'accueil et une description.

Les établissements hôteliers et les prestataires devront avoir accès à un récapitulatif des réservations effectuées sur leur espace.

N'hésitez pas à nous contacter pour convenir d'un rendez-vous et discuter de l'éventuel design de l'application.

**Codialement,**

**John Doe**

# Environnement de travail



## Visual Studio Code

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS2.

Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code,



## Adobe XD

Adobe XD, est un logiciel qui permet de réaliser des prototypes d'applications ou de sites web.

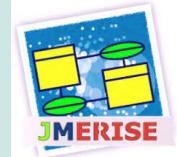
La maquette peut contenir tous les liens utiles aux tests utilisateurs ce qui favorise le travail en équipe des concepteurs d'applications et de sites.



## Git Hub

GitHub est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git.

Le site assure également un contrôle d'accès et des fonctionnalités destinées à la collaboration comme le suivi des bugs, les demandes de fonctionnalités, la gestion de tâches et un wiki pour chaque projet.



## JMerise

JMerise est un outil portable de modélisation de MCD MERISE, développé en java.

Il permet la création du MCD et la génération du MLD et du script SQL de création des tables.



## PHP My Admin

phpMyAdmin (PMA) est une application Web de gestion pour les systèmes de gestion de base de données MySQL et MariaDB, réalisée principalement en PHP et distribuée sous licence GNU GPL.

Il s'agit de l'une des plus célèbres interfaces pour gérer une base de données MySQL sur un serveur PHP..



## Maria DB

MariaDB est un système de gestion de base de données relationnelle (SGBDR) open source qui constitue une solution de remplacement compatible avec la technologie très répandue des bases de données MySQL.

MariaDB a été conçu en 2009 en réaction à l'acquisition de MySQL par Oracle Corp.



## HTML 5

Le langage HTML (de l'anglais Hypertext Markup Language) est, depuis les toutes premières heures de l'Internet, le programme de base en matière de structuration, de mise en réseau et de contenu sur le World Wide Web. Cependant, le langage de balisage n'a cessé de se développer suite à la sortie de la version HTML 4.01 en décembre 1999. ..



## CSS 3

Le CSS (Cascading Style Sheet) est un langage informatique servant à décrire la présentation et le style d'un document HTML et XML. Datant des années 90, ce langage sert principalement au développement de sites web.

Le CSS se reconnaît simplement avec un point précédant ses classes, une accolade précédant ses propriétés et un double-point précédant ses valeurs.



## JavaScript

JavaScript est un langage de programmation qui permet d'implémenter des mécanismes complexes sur une page web. C'est la troisième couche des technologies standards du web, les deux premières étant HTML et CSS.

Il permet entre autres d'afficher du contenu mis à jour à des temps déterminés, des cartes interactives, des animations 2D/3D etc...



## PHP

PHP est un langage de scripts généraliste et Open Source, spécialement conçu pour le développement d'applications web. Il peut être intégré facilement au HTML. Ce qui distingue PHP des langages de script comme le Javascript, est que le code est exécuté sur le serveur, générant ainsi le HTML, qui sera ensuite envoyé au client.



## Bootstrap

Bootstrap est un framework développé par l'équipe du réseau social Twitter. Proposé en open source (sous licence MIT), ce framework utilisant les langages HTML, CSS et JavaScript fournit aux développeurs des outils pour créer un site facilement. Ce framework est pensé pour développer des sites avec un design responsive, qui s'adapte à tout type d'écran, et en priorité pour les smartphones.

# Compétences couvertes

- Maquetter une application
- Réaliser 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
- Créer une base de données
- Développer les composants d'accès aux données
- Développer la partie back-end d'une application web ou web mobile
- Elaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce

# Organisation du projet



**Elaboration d'un cahier des charges détaillé.  
Compréhension des besoins et choix des fonctionnalités.  
Création d'un Trello et repo GitHub.**



**Définition de la charte graphique (couleurs, police d'écriture).  
Elaboration d'une maquette responsive, recherche d'image et deisgn UX/UI.**



**Construction de la structure MVC.  
Ecriture code HTML et CSS et adéquation avec la maquette réalisée. Utilisation de Javascript en parallèle pour certaines fonctionnalités.**



**Création du schéma conceptuel de base de données. Génération du code SQL pour la création. Définition des cardinalités.**



**Programation de la partie Back-end de l'application en PHP.  
Création des contrôleurs et gestion du CRUD.  
Tests sur l'application et Debug.**

# Utilisation de Trello

## - Qu'est-ce-que TRELLO ?



Trello est un outil de gestion de projet en ligne, lancé en septembre 2011 et inspiré par la méthode Kanban de Toyota. Il repose sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Les cartes sont assignable à des utilisateurs et sont mobiles d'une planche à l'autre, traduisant leur avancement.

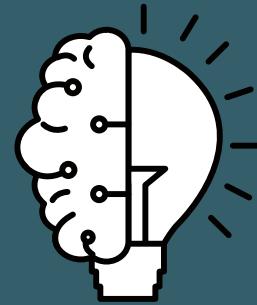
En voici un exemple illustrant mon projet.



The screenshot shows a Trello board titled "Liugo". The board has four columns: "A faire", "En cours", "A vérifier", and "Terminé".

- A faire:**
  - Vérification supplémentaires sur le code JS de création de service pour le bouton valider
  - Page "Mes réservations" contenant toutes les activités réservées par le client avec possibilité de les annuler.
  - Page de gestion du compte pour le voyageur en tenant compte du CRUD.
  - Page récapitulative des réservations effectuées par les voyageurs sur l'espace client pro.
  - Mise en ligne du projet
- En cours:**
  - Création du système de réservation sur la page du service sélectionné.
  - + Ajouter une carte
- A vérifier:**
  - + Ajouter une carte
- Terminé:**
  - Création du repo GitHub
  - Maquettage, recherche d'images et définition de la charte graphique
  - Système de création de compte pour les professionnels
  - Style CSS global
  - Création de compte et connexion de l'espace voyageur
  - Page d'accueil de l'espace voyageur avec affichage des différents services personnalisés par l'Hôtel
  - Création d'une barre de recherche sur la page des services prestataires
  - Page d'affichage pour les services prestataires

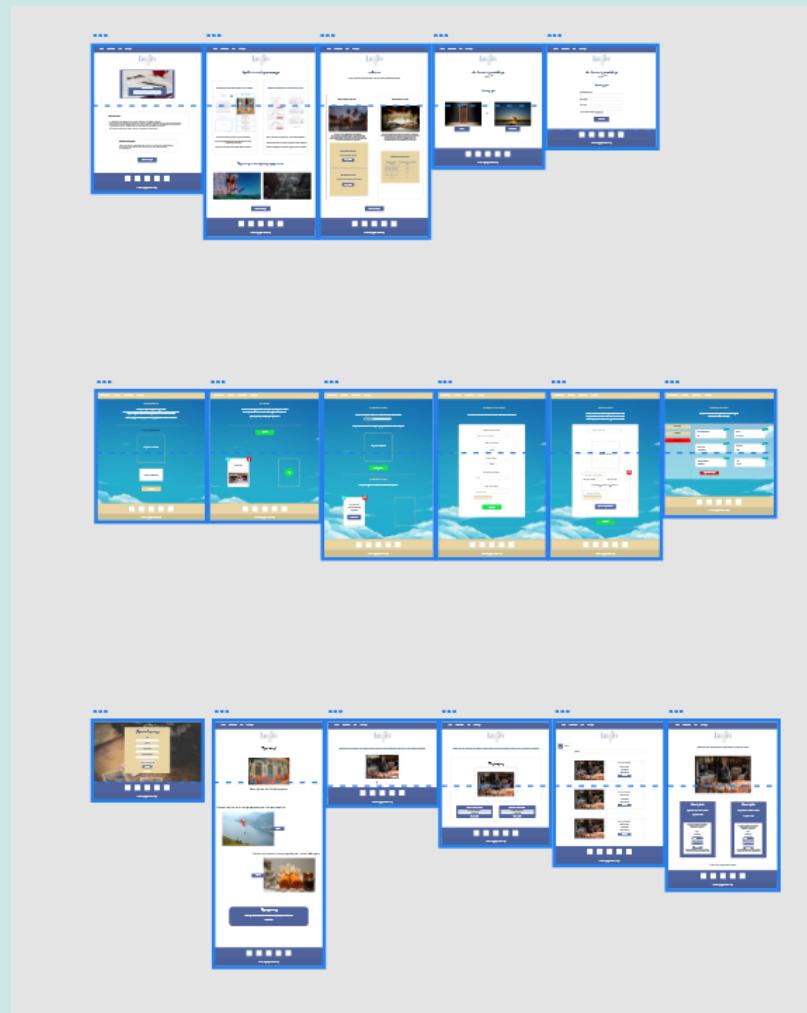
# Adobe XD MAQUETTAGE



Le maquettage est une étape importante du projet dans laquelle j'ai défini le design principal de l'application.

Elaboration d'une charte graphique, en utilisant des couleurs rappelant les vacances pour rester dans le thème et le public cible.

Création du prototype fonctionnel donnant un aperçu de l'expérience utilisateur.

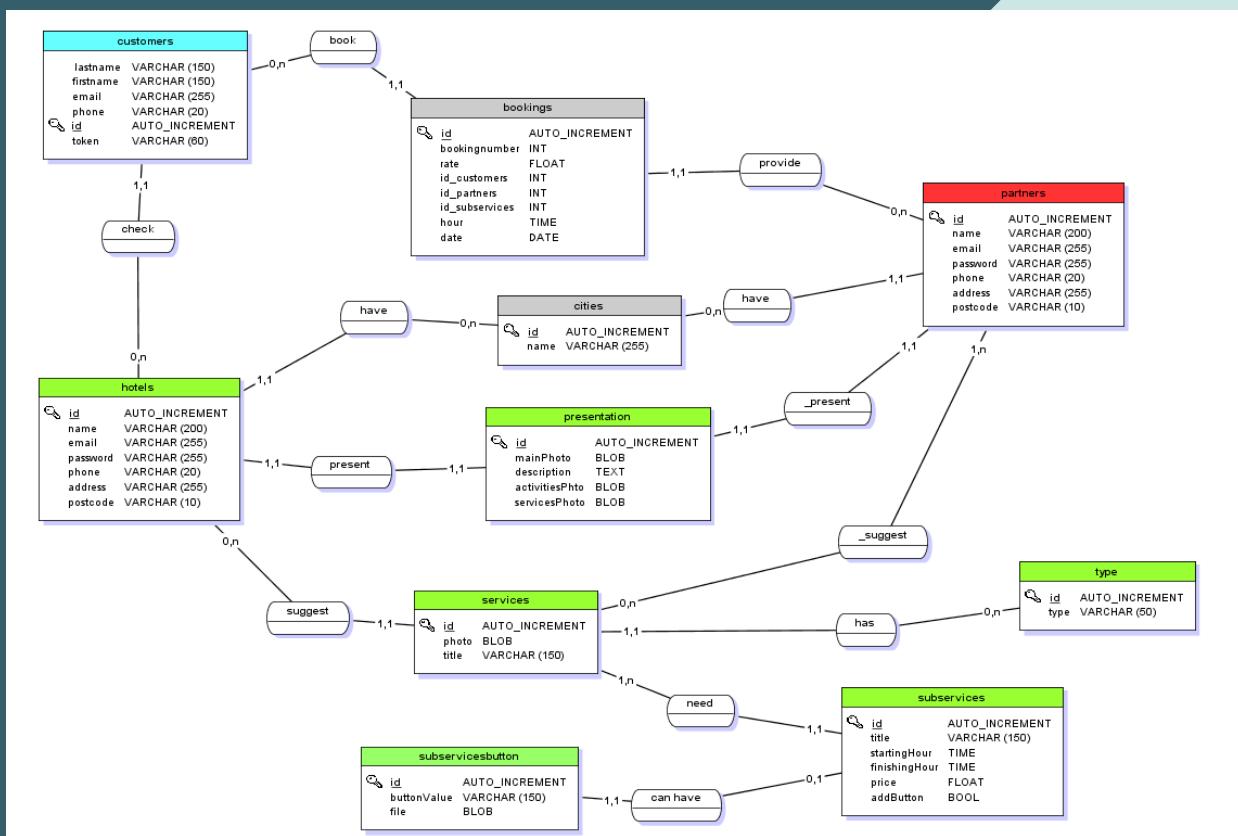


**Le maquettage est un processus de concrétisation graphique de l'interface d'une interface digitale (site web, logiciel, application...).**

# SCHÉMA CONCEPTUEL DE BDD

Le schéma conceptuel est également un étape majeure dans la réalisation du projet.

En effet, cela nous permet de définir les différentes colonnes et données qui devront être enregistrées, ainsi que les cardinalités entre chaque table pour la génération de clés étrangères.





# Composer

Composer est un logiciel gestionnaire de dépendances libre écrit en PHP.

Il permet à ses utilisateurs de déclarer et d'installer les bibliothèques dont le projet principal a besoin.

Le développement a débuté en avril 2011 et a donné lieu à une première version sortie le 1er mars 2012.

Développé au début par Nils Adermann et Jordi Boggiano<sup>6</sup> (qui continuent encore aujourd'hui à le maintenir), le projet est maintenant disponible sur la plateforme GitHub<sup>7</sup>.

Il est ainsi développé par toute une communauté<sup>8</sup>.

## Installation en ligne de commande

```
PS C:\wamp64\www\Projet-Liugo> php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
>> php -r "if (hash_file('sha384', 'composer-setup.php') === '906a84df04cea2aa72f40b5f787e49f22d4c2f19492ac310e8cba5b96ac8b64115ac402c8cd292b8a03482
574915d1a8') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
>> php composer-setup.php
>> php -r "unlink('composer-setup.php');"
```

# Exemple de paquets composer utilisés

Le but étant, pour cette formation, d'utiliser le moins de dépendances, framework... que possible.

J'ai donc installé uniquement les paquets composer qui m'étaient réellement nécessaires.

## Gestion des QR Codes

J'ai utilisé pour cette fonctionnalités le paquet bacon-qr-code, qui permet de générer les qr code qui seront flashés par les utilisateurs.



```
use BaconQrCode\Renderer\ImageRenderer;
use BaconQrCode\Renderer\Image\ImagickImageBackend;
use BaconQrCode\Renderer\RendererStyle\RendererStyle;
use BaconQrCode\Writer;

$renderer = new ImageRenderer(
    new RendererStyle(400),
    new ImagickImageBackend()
);
$writer = new Writer($renderer);
$writer->writeFile('Hello World!', 'qrcode.png');
```

Ici, on crée une variable \$renderer dans laquelle nous allons instancier trois objets inclus dans la dépendance.

La variable \$writer sera le résultat affiché.



# QU'EST-CE-QUE LE MVC ?

## MODÈLE-VUE-CONTRÔLEUR

motif d'architecture logicielle destiné aux interfaces graphiques lancé en 1978 et très populaire pour les applications web. Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs.

- Un modèle (Model) contient les données à afficher.
- Une vue (View) contient la présentation de l'interface graphique.
- Un contrôleur (Controller) contient la logique concernant les actions effectuées par l'utilisateur.

## APPLICATION DANS LE PROJET

Chaque vue du projet possède un controller, qui va contenir le code de gestion de formulaires, utilisation de méthodes du modèle ou gestion des requêtes AJAX.

Les modèles sont faits en POO(Programmation orientée Objet) et sont donc des objets.

Chaque table possède un modèle et chaque attribut correspond à une colonne de la dite table.

Le but serait, dans l'avenir, d'avoir une structure centrée sur les controllers en créant un système de routing manuel.

# CONTROLLER DE GESTION DE FORMULAIRE

```
if (isset($_POST['save'])) {
    $errorList = [];
    //Vérifications du champ de titre de service dans le formulaire
    if (!empty($_POST['ssTitle'])) {
        if (preg_match($titleRegex, $_POST['ssTitle'])) {
            $ssTitle = htmlspecialchars($_POST['ssTitle']);
        } else {
            $errorList['ssTitle'] = 'Merci d\'entrer un titre de sous-service valide(tirets et espaces acceptés)';
        }
    } else {
        $errorList['ssTitle'] = 'Merci d\'entrer un titre de sous-service';
    }
}
```

L'on vérifie que le bouton "sauvegarder" a bien été cliqué par l'utilisateur.

Dans ce cas on vient créé un tableau vide \$errorList qui va lister les erreurs potentielles.

Si la vérification de REGEX est validée, la valeur envoyée par l'utilisateur est insérée dans une variable.

L'opération est répétée pour chaque input.

Les controllers sont inclus à la vue correspondante en haut de page.

## CRÉATION DES REGEX

J'utilise le site REGEX101 pour simplifier les tests.

Exemple de REGEX créés :

```
$nameRegex = '/^[\a-zA-ZÀ-Ӯӻ-ӿ]*([\_\'\-\']*[a-zA-ZÀ-Ӯӻ-ӿ]*)?$/';
$mailRegex = '/^[\a-z0-9]([a-z0-9\-\_\.\-]*[@](\a-z0-9\.)+)[\.\-]([a-z]\{2,5\})/i';
```

Ici, pour la \$mailRegex, Nous permettons à l'utilisateur d'insérer des caractères illimités incluant tous les nombres, les lettres minuscules ainsi que certains caractères spéciaux.

Toutefois ce groupe doit impérativement commencer par une lettre ou un chiffre.

Ensuite un arobase obligatoire, puis un groupe de capture incluant lettres, chiffres et point, le plus symbolise une récurrence minimum obligatoire.

Puis le domaine, entre accolades est indiquée la récurrence autorisée, entre 2 et 5 caractères.

# EXEMPLE DE MODÈLE

```
<?php

class Account extends Database {
    private string $name;
    private string $email;
    private string $password;
    private int $id;
    private string $phone;
    private string $address;
    private string $postcode;
    private int $idCities;
    private string $sector;
    private string $token;
    private string $table;
}

public function createAccount():bool {
    $query = 'INSERT INTO ' . $this->table . '(`name`, `email`, `password`, `token`) VALUES (:name, :email, :password, :token)';
    $queryStatement = $this->db->prepare($query);
    $queryStatement->bindValue(':name', $this->name, PDO::PARAM_STR);
    $queryStatement->bindValue(':email', $this->email, PDO::PARAM_STR);
    $queryStatement->bindValue(':password', $this->password, PDO::PARAM_STR);
    $queryStatement->bindValue(':token', $this->token, PDO::PARAM_STR);
    return $queryStatement->execute();
}

public function checkifAccountExists():object {
    $query = 'SELECT COUNT(`id`) AS `check` FROM ' . $this->table . ' WHERE :email = :email';
    $queryStatement = $this->db->prepare($query);
    $queryStatement->bindValue(':email', $this->email, PDO::PARAM_STR);
    $queryStatement->execute();
    $numberOfAccounts = $queryStatement->fetch(PDO::FETCH_OBJ);
    return $numberOfAccounts;
}
```

Le code ci-dessus est segmenté en deux parties :

- La création des attributs, qui sont private et donc accessibles uniquement depuis cet objet.

Deux méthodes :

- La première permet de créer un compte client en insérant dans la base de données ses informations ainsi qu'un jeton généré aléatoirement pour valider l'adresse e-mail.
- La deuxième permet de vérifier à l'aide d'un SELECT si l'adresse e-mail du client est déjà enregistrée dans la base de données.

Ensuite nous avons également les setters en fin de fichier qui vont permettre d'associer une valeur aux attributs dans le controller :

```
public function setPassword($newPassword) {
    $this->password = $newPassword;
}
```

# LES REQUÊTES AJAX

## QU'EST-CE-QUE L'AJAX ?

Ajax est une méthode utilisant différentes technologies ajoutées aux navigateurs web entre 1995 et 2005, et dont la particularité est de permettre d'effectuer des requêtes au serveur web et, en conséquence, de modifier partiellement la page web affichée sur le poste client sans avoir à afficher une nouvelle page complète.

Cette architecture informatique permet de construire des applications Web et des sites web dynamiques interactifs.

## MÉTHODES AJAX JAVASCRIPT

Il existe différentes objets permettant de faire de l'AJAX en javascript, notamment :

- XMLHttpRequest
- FormData

Pour ma part j'ai décidé de travailler avec l'objet FormData et l'API Fetch de javascript.

Sur la page suivante vous trouverez quelques exemples de code AJAX.

# EXEMPLE DE CODE EN UTILISANT FORMDATA

```
const searchRegex = /^[a-zA-ZàáääàééëëííñóööùúüýÿæÀÁÄÄÉÉËËÍÍÑÓÖÖÙÚÜÝÝÃ€\!\.\_\s-]\{2,50\}$/

searchButton.addEventListener("click", () => {
    if (searchRegex.test(search.value)) {
        const formData = new FormData()
        formData.append('search', search.value)
        // autre façon de faire de l'ajax
        fetch("./controller/searchController.php", { method: 'POST', body: formData })
            .then(response => response.json()) // si je recois du json je met .json() a la place
            .then(response => {
                noSearchRow.remove()
                if(response.length > 0){
                    response.forEach(element => {
                        console.log(element.title)
                        newHtml = `<div class="row justify-content-center" id="noSearchRow">
                            <div class="col-8 searchCol">
                                <div class="searchResult">
                                    <div class="row">
                                        <div class="col-6">
                                            
                                        </div>
                                        <div class="col-6 serviceDescriptionCol">
                                            <h3>${element.partnerName}</h3>
                                            <p>${element.title}</p>
                                            <p>A partir de ${element.serviceLowestPrice}</p>
                                            <p>Situé à ${element.cityName}</p>
                                            <a class="btn btn-outline-light customerAccountButton" href="servicePage.php?serviceId=${element.id}">Voir détails</a>
                                        </div>
                                    </div>
                                </div>
                            </div>
                        </div>`
                        headerRow.insertAdjacentHTML('afterend', newHtml)
                    })
                }
            })
    }
})
```

Le code partiel ci-dessus est utilisé pour la gestion de la recherche par l'utilisateur.

J'ai ajouté un écouteur d'évènement sur le bouton de recherche, ensuite j'effectue un test de REGEX en javascript qui permet par la suite d'afficher un message d'erreur dynamique (Les vérifications sont effectuées également en PHP).

J'instancie l'objet `FormData()` dans une constante, je me sers ensuite de la méthode `append` en indiquant en paramètres ; le nom de la donnée POST ainsi que sa valeur.

J'envoie ces données à un controller php spécifique pour ma barre de recherche, qui me renvoie une réponse XHR contenant un fichier JSON créé en PHP.

J'utilise ensuite les données incluses dans le fichier JSON pour construire le code HTML à insérer.



# RESPONSIVE DESIGN

# QU'ENTENDONS-NOUS PAR RESPONSIVE ?

Le Responsive Design ou plus précisément le Responsive Web Design (RWD) est une technique de conception d'interface digitale qui fait en sorte que l'affichage d'une quelconque page d'un site s'adapte de façon automatique à la taille de l'écran du terminal qui le lit.

Il est différent de l'Adaptative Design bien que les deux concepts aient pour but d'améliorer l'ergonomie mobile du site web.

## BOOTSTRAP

```
<div class="row justify-content-center mt-5">
    <!-- Oppening account informations col-->
    <div class="col-10 col-md-5 col-lg-5 mt-5 mt-md-0 mt-lg-0 mx-md-3 border rounded shadow">
        <h2 class="tangerine mt-2">Essayez dès maintenant :</h2>
        <p class="listMargin didot">Créez votre compte gratuitement afin d'essayer Les services
            <!-- Closing account informations col-->
    </div>
</div>
```

Bootstrap est une solution simplifiée permettant de créer des responsives layout, en effet le framework utilise le flexbox et un système de classe intégré permettant de facilement décider du comportement des pages en fonction de la largeur de l'écran. Le principe fondamental est celui des breakpoints :

- XS correspond à un écran d'une largeur maximale de 576 pixels.
- Small équivaut à une largeur d'écran située entre 576 et 768 px.

Voici quelques exemples, je ne citerai pas tous les breakpoints mais ils sont au nombre de 6.

En jouant avec ces breakpoints et d'autres propriétés CSS qui seront citées plus bas, un élément peut prendre un pourcentage différent de la largeur de l'écran.

# LES MEDIAS QUERIES

CSS 3 inclut également un système de Medias Queries, ci-dessous un exemple que nous allons analyser ensemble :

```
#slider {  
    position: relative;  
    width: 50%;  
    height: 32vw;  
    margin: 150px auto;  
    font-family: 'Helvetica Neue', sans-serif;  
    perspective: 1400px;  
    transform-style: preserve-3d;  
}  
  
@media only screen and (max-width: 768px){  
    #slider {  
        width: 60%;  
        height: 40vw;  
    }  
}
```

Ici, nous avons la balise block d'un slider, on peut voir que sa width est de 50% et sa hauteur de 32 view port width.

Grâce à la règle "media", nous allons indiquer à CSS les propriétés à modifier, dans ce cas on indique que la modification doit être effectuée dans le cas où la largeur de l'écran serait plus petite ou égale à 768 pixels de large.

Dans ce cas le Slider est légèrement agrandit pour augmenter la visibilité de l'utilisateur.

Cette règle peut être utilisée pour tout type de propriété.

*En combinant les media queries ainsi que le framework bootstrap vu plus haut, nous pouvons facilement créer un design qui s'adapte automatiquement à l'utilisateur.*

# SÉCURITÉ

## LES FAILLES SQL

La faille SQLi, abréviation de SQL Injection, soit injection SQL en français, est un groupe de méthodes d'exploitation de faille de sécurité d'une application interagissant avec une base de données. Elle permet d'injecter dans la requête SQL en cours un morceau de requête non prévu par le système et pouvant en compromettre la sécurité.

## PROTECTION

Pour prévenir les injections SQL, il faut faire appel aux requêtes préparées. Ce sont des requêtes dans lesquels les paramètres sont interprétés indépendamment de la requête elle-même. De cette manière, il est impossible d'effectuer des injections. Dans tous les systèmes de gestion de bases de données, deux méthodes sont utilisées : `prepare()` qui prépare la requête et `execute()` qui exécute la requête avec les paramètres.

```
$queryStatement = $this->db->prepare($query);
$queryStatement->bindValue(':postcode', $this->postcode, PDO::PARAM_STR);
$queryStatement->execute();
```

# LES ATTAQUES XSS

Les attaques XSS consistent à insérer un code malveillant dans des sites Web par ailleurs fiables. Une attaque XSS se produit quand des cybercriminels injectent un script malveillant dans le contenu du site Web ciblé, qui est ensuite inclus dans le contenu dynamique reçu par le navigateur de la victime. Il est impossible pour le navigateur de différencier les balises valides de celles du hacker et il se contente donc de les exécuter.

## CONSÉQUENCES

Ces scripts malveillants peuvent accéder aux cookies, aux jetons de session ou à d'autres informations sensibles conservées par le navigateur et utilisées sur ce site. Les hackers peuvent aussi se servir du XSS pour diffuser un malware, réécrire le contenu du site, perturber des réseaux sociaux et hameçonner les identifiants d'un utilisateur.

## PROTECTION

Pour se protéger de ce type d'attaques, toutes les entrées effectuées par l'utilisateur (en HTML ou en JavaScript) doivent être filtrées, pour mon projet, j'ai utilisé les deux méthodes suivantes :

- `htmlspecialchars()` qui est une méthode PHP permettant de , et remplacer certains caractères qui ont des significations spéciales en HTML par des entités HTML pour conserver leurs significations.

```
$name = htmlspecialchars($_POST['name']);
```

- Les expressions régulières (ou REGEX) qui permettent de filtrer les entrées selon un pattern bien précis établis à l'avance.

```
if (!preg_match($nameRegex, $_POST['name'])) {
```

# LE CROSS-SITE REQUEST FORGERY

Le cross-site request forgery, souvent abrégé en CSRF (ou injection de requêtes illégitimes par rebond, en français), est un type d'attaque qui se produit quand un site Web, un blog, un e-mail, un message instantané ou une application Web malveillant(e) oblige le navigateur Web d'un utilisateur à effectuer des opérations indésirables, sur un site de confiance où l'utilisateur est actuellement authentifié. L'impact d'une attaque CSRF dépend des informations exposées dans l'application vulnérable. À leur niveau le plus élémentaire, les attaques CSRF servent à contraindre un système cible à réaliser des opérations malveillantes via le navigateur cible, à l'insu de l'utilisateur cible.

## PROTECTION

Les tokens anti-CSRF :

La méthode recommandée et la plus largement adoptée pour lutter contre les attaques cross-site request forgery consiste à utiliser un token anti-CSRF, ou token de synchronisation.

Lorsqu'un utilisateur envoie des informations ou interagit avec le site, ou entreprend quoi que ce soit d'autre qui génère un cookie, le token anti-CSRF doit être inclus dans la demande de cookie. Cette demande passe ensuite par un processus de vérification, où l'authenticité voire l'existence de ce token est vérifiée avant de traiter la demande. Si le token est manquant ou incorrect, la demande est rejetée.

# FONCTIONNALITÉS PRINCIPALES

## PERSONNALISATION

Les utilisateurs professionnels de l'application doivent pouvoir personnaliser l'entièreté des services qu'ils proposent. En effet, un prestataire peut très bien proposer des cours de parapente, des leçons de ski, ou encore un plat du jour en fonction de son secteur d'activités. Cela est implémenté par un formulaire interactif, qui réponds aux besoins des différents prestataires réalisés presque intégralement en JavaScript.

```
let numberOfServices = 1
document.addEventListener("click", event => {
  let textToAppend
  if (event.target.matches(".addPresta") && numberOfServices >= 5) {
    return alert("Vous pouvez ajouter un maximum de 5 services")
  }
  if (event.target.matches(".addPresta")) [
    parentlist = event.target.parentElement.classList[3]
    textToAppend = `<div class="presta">
      <div class="row justify-content-center">
        <div class="col-10 text-center mt-4 innerExampleCol">
          <button type="button" class="redCrossButton btn btn-outline-light deletePresta my-4">x</button>
          <input type="text" name="serviceName[]" class="mt-2 placeholder="Nom du service">
          <label for="serviceStartingHour">Heure de début</label>
          <input type="time" name="serviceStartingHour[]" class="mt-2 placeholder="heure de début">
          <input type="number" name="servicePrice[]" class="mt-2 placeholder="tarifs">
          <label for="serviceEndingHour">Heure de fin</label>
          <input type="time" name="serviceEndingHour[]" class="mt-2 placeholder="heure de fin">
          <p class="mt-2 radioQuestion">Souhaitez-vous ajouter un bouton ?</p>
          <input class="my-2 showInput" type="radio" name="buttonQuestion${numberOfServices}" value="1"><span>Oui</span>
          <input class="my-2 hideInput" type="radio" name="buttonQuestion${numberOfServices}" checked="checked" value="0"><span>Non</span>
          <div class="buttonContainer hiddenInput">
            <input type="text" name="buttonName[]" placeholder="nom du bouton" class="mt-2">
            <label>Fichier à télécharger au clic</label>
            <input type="file" name="buttonFile[]" class="my-2">
          </div>
        </div>
      </div>
    
```

Génération du code HTML à insérer

# SYSTÈME DE RÉSERVATION

Les utilisateurs peuvent également réserver les activités qu'ils souhaitent.

Pour cela ils peuvent utiliser la barre de recherche, et ensuite indiquer une date dans un formulaire.

Une requête AJAX est alors effectuée et le controller vérifie si une réservation a déjà été effectuée sur la date indiquée.

Ensuite le controller renvoie en réponse les créneaux horaires encore disponibles, qui sont insérés dans le formulaire sous forme de SELECT pour que le client puisse sélectionner l'horaire qui lui convient.

```
$jsonTable = array();

if (! $booking->checkIfBookingsOnSameDate($subServiceId)) {

    array_push($jsonTable, array(
        'id' => $selectedSubServices->subServiceId,
        'title' => $selectedSubServices->subServiceTitle,
        'startingHour' => substr($selectedSubServices->subServiceStartingHour, 0, 2),
        'finishingHour' => substr($selectedSubServices->subServiceFinishingHour, 0, 2),
        'price' => $selectedSubServices->subServicePrice
    ));

} elseif ($booking->checkIfBookingsOnSameDate($subServiceId)) {
    $bookedHours = $booking->getBookingHourOnSameDate($subServiceId);
    foreach ($bookedHours as $hour) {
        $bookedHoursArray[] = substr($hour, 0, 2);
    }
    array_push($jsonTable, array(
        'id' => $selectedSubServices->subServiceId,
        'title' => $selectedSubServices->subServiceTitle,
        'startingHour' => substr($selectedSubServices->subServiceStartingHour, 0, 2),
        'bookedHours' => $bookedHoursArray,
        'finishingHour' => substr($selectedSubServices->subServiceFinishingHour, 0, 2),
        'price' => $selectedSubServices->subServicePrice
    ));
}

$json = json_encode($jsonTable);
echo $json;
```

*Création du JSON en insérant les données récupérées*

# CRUD

*Create, Read, Update and Delete*

## CREATE

Il existe deux grands types de création dans ce projet :

- La création de compte, qui permet à l'utilisateur de pouvoir se connecter dans un espace dédié.

- La création de service, qui permet aux professionnels de personnaliser leur offre.

```
//Insertion des données
if (count($errorList) == 0) {
    //Hashage du mdp
    $hashedPassword = password_hash($password, PASSWORD_DEFAULT);
    $account = new Account;
    //Selon le type d'utilisateur, configurer la table de création de compte
    if (isset($_GET['type'])) {
        if ($_GET['type'] == 'hotels') {
            $account->setTable('hotels');
        } else if ($_GET['type'] == 'partners') {
            $account->setTable('partners');
        }
        //Création d'une nouvelle instance Hotels
        //Setters
        $account->setName($name);
        $account->setEmail($mail);
        $account->setPassword($hashedPassword);
        $checkAccountExists = $account->checkIfAccountExists();
        //Si l'adresse e-mail ne correspond à aucun compte existant dans la bdd, insérer les données
        if (!$checkAccountExists->check) {
            $token = new Token;
            $generatedToken = $token->createToken();
            $account->setToken($generatedToken);
            if ($account->createAccount()) {
                ...
            }
        }
    }
}
```

```
//On compte le nombre d'erreurs liées au formulaire
if (count($errorList) == 0) {
    //Ajout du service
    $service->setServiceTitle($serviceTitle);
    //Instance de la classe permettant la création du slug
    $slugify = new Slug;
    //Transformation du titre du service en slug
    $serviceSlug = $slugify->slugify($serviceTitle);
    $service->setSlug($serviceSlug);
    try {
        //Début de la transaction
        $service->beginTransaction();
        $service->addService();
        //Récupération de son id
        $serviceId = $service->getServiceId();
    }
}
```

A noter

J'ai également créé un système de gestion de fichier, permettant aux professionnels d'ajouter des photos ou des fichiers en pdf comme des menus à afficher. En voici un exemple de méthode :

```
public function registerCategoryFile($oldPath, $oldFileName, $id, $directory)
{
    $path = $directory . '/' . $_SESSION['login'] . '/category/';
    $temp = explode(".", $oldFileName);
    return rename($oldPath, $path . 'categoryPhoto' . $id . '.' . end($temp));
}
```

# CRUD

*Create, Read, Update and Delete*

## READ

Le but est de permettre aux clients individuels d'avoir accès aux informations personnalisées par les professionnels.

Informations  
bouclées  
dans la vue.



```
<?php foreach ($servicesByPage as $service) { ?>
    <div class="col-8 searchCol">
        <div class="searchResult">
            <div class="row">
                <div class="col-6">
                    title ?>">
                </div>
                <div class="col-6 serviceDescriptionCol">
                    <h3><?= $service->partnerName ?></h3>
                    <p><?= $service->title ?></p>
                    <p>A partir de <?= $serviceLowestPrice[$service->id] ?>€</p>
                    <p>Situé à <?= $cityName[$service->id] ?></p>
                    <a class="btn btn-outline-light customerAccountButton" href="servicePage.php?serviceId=<?= $service->id ?>">Découvrir</a>
                </div>
            </div>
        </div>
    </div>
<?php } ?>
```

```
$service = new Service;
$city = new City;

$servicesByPage = $service->getAllPartnersServices(1);

foreach($servicesByPage as $selectedService) {
    $service->setServiceId($selectedService->id);
    $serviceLowestPrice[$selectedService->id] = $service->getSubServiceLowerPriceFromService()->lowestPrice;
    $city->setCityId($selectedService->cityId);
    $cityName[$selectedService->id] = $city->getCityNameFromCityId();
```



Appels à la  
base de  
données.

# CRUD

*Create, Read, Update and Delete*

## UPDATE

Les utilisateurs peuvent mettre à jour les informations de leur comptes dans une section dédiées.

```
if (isset($_POST['email'])) {
    if (preg_match($regexMail, $_POST['email'])) {
        $email = htmlspecialchars($_POST['email']);
        $account->setEmail($email);
        $account->updateAccountEmail();
    } else {
        $errorList['email'] = 'Merci d\'entrer un email valide';
    }
} else {
    $errorList['email'] = 'Merci d\'entrer un email';
}
```

*A noter*

Cette méthode existe pour chaque information renseignée.  
Elle est couplée à une requête AJAX pour une modification dynamique.

# CRUD

*Create, Read, Update and Delete*

## DELETE

Les utilisateurs peuvent supprimer leur compte si nécessaire, les fichiers qu'ils avaient enregistrés seront également supprimés dans le processus.

Les professionnels peuvent également supprimer des services ajoutés.

```
if(isset($_POST['deleteConfirm'])) {
    $account->deleteAccount();
    $fileCheck = new Files;
    $fileCheck->rmdir($dirName . '/' . $_SESSION['login']);
    session_destroy();
    header('Location: ../proside/homepage.php');
    exit;
}
```

*A noter*

*Si un service est supprimé, les sous-services associés ainsi que les fichiers sont également supprimés.*

*L'utilisateur peut choisir de supprimer uniquement le sous-service sans impacter le service si nécessaire.*



# EXEMPLE DE RECHERCHE

## GET ELEMENT IN JAVASCRIPT AFTER:2020

<https://javascript.info/searching-elements-dom>

De loin la méthode la plus versatile, `elem.querySelectorAll(css)` retourne tous les éléments à l'intérieur de "elem" qui correspondent au sélecteur CSS donné.

Ici, nous récupérons tous les éléments `<li>` qui sont les derniers enfants.

Cette méthode est sans nul doute puissante car nimporte quel sélecteur CSS peut être utilisé.

### querySelectorAll

By far, the most versatile method, `elem.querySelectorAll(css)` returns all elements inside `elem` matching the given CSS selector.

Here we look for all `<li>` elements that are last children:

```
1 <ul>
2   <li>The</li>
3   <li>test</li>
4 </ul>
5 <ul>
6   <li>has</li>
7   <li>passed</li>
8 </ul>
9 <script>
10  let elements = document.querySelectorAll('ul > li:last-child');
11
12  for (let elem of elements) {
13    alert(elem.innerHTML); // "test", "passed"
14  }
15 </script>
```

# **CONCLUSION**



## FONCTIONS SUPPLÉMENTAIRES

Création d'un chat permettant aux prestataires de communiquer avec les clients individuels directement depuis l'application.

Avec système de messagerie.

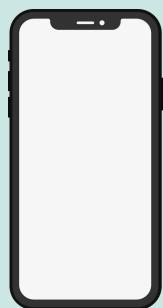
Incorporation de notifications envoyées aux prestataires lorsqu'une réservation est effectuée.



Création d'un espace client séparé pour les prestataires de services et pour les Hôtels, afin de personnaliser dn'avantages les options de création à leur disposition.



Création d'une application disponible sur les stores (Play store, Apple store) permettant aux utilisateurs d'accéder à leur espace membre et de gérer leurs réservations.



Ajout d'un système de transactions et de paiement (type paybox), les utilisateurs pourraient régler leur réservation directement sur l'application.  
Ajouter également un système de redistribution des commissions.

Refonte du design et du logo pour les rendres plus orientés UX.

Réécriture du code en utilisant un framework JS tel que REACT, permettant ainsi de développer l'application en REACT NATIVE.



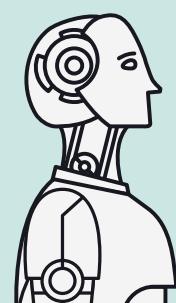


Modification de la partie back-end afin qu'elle puisse convenir à la partie application mobile et web.



Ajout d'un système de tutoriel à la création du compte, pour la partie particuliers et professionnels. Permettant ainsi d'appréhender plus facilement l'application en tant que nouvel utilisateur.

Création d'un chatbot répondant aux questions ciblées des utilisateurs.  
Création de réponses préparées.



## PISTES D'AMELIORATION



## **AMÉLIORATION DE LA CHARTE GRAPHIQUE**

S'intéresser d'avantage au design UI/UX, comprendre les problématiques de l'utilisateur et réaliser une interface plaisante et attrayante.

## **OPTIMISATION DE L'ACCESSIBILITÉ**

Développer de nouvelles fonctionnalités et adapter d'une meilleure façon le projet afin qu'il soit entièrement accessible, et ce sans aucune difficulté, aux personnes en situation d'handicap.

## **MISE EN PLACE D'UN SYSTÈME DE ROUTING**

Création d'un système de routing manuel permettant aux différents controllers d'inclure les fichiers nécessaires, création d'un fichier htaccess permettant la réécriture des URL.

## **PROTECTIONS SUPPLÉMENTAIRES**

Ajout d'un système de tokens anti-CSRF permettant de se protéger de cette attaque éventuelle.

Création d'un système de captcha évitant ainsi la présence de boting.