
DOSSIER DE PROJET

Création et développement d'une application
web e-commerce

Titre Professionnel Développeur Web et Web
Mobile

Rattrapage de la partie front-end du référentiel

TP-01286 — RNCP n°31114

Formateur : Ainhoa RIBALLO – Michel SAFFRE

Référent formation : François BECETTE

Session du 22 Avril 2021
Locaux l'Escalier
SAINT LEONARD DE NOBLAT



TABLE DES MATIERES

- I. Liste des compétences du référentiel
- II. Objectif du projet
- III. Caractéristiques du projet
- IV. Spécifications techniques
- V. Réalisations
- VI. Présentation du jeu d'essai
- VII. Veille
- VIII. Situation de recherche
- IX. Extrait et traduction d'une source anglophone
- Conclusion
- Annexes

I - Liste des compétences du référentiel

- Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité
1. Maquetter une application
 2. Réaliser une interface utilisateur web statique et adaptable
 3. Développer une interface utilisateur web dynamique
 4. Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce

II – Objectif du projet

Mon projet, support de cet épreuve, consiste à élaborer un site vitrine pour le gérant d'un établissement de restauration rapide situé à CHAMBERET en Corrèze. Ne disposant que d'une page sur un réseau social, l'idée d'un site lui a paru séduisante.

Il a alors été convenu de réaliser essentiellement, du côté client, une page présentant les plats, et du côté administrateur, une zone dédiée à la gestion du contenu. A cela a été ajouté les pages que l'on retrouve traditionnellement sur les sites web : contact, mentions légales, plan du site.

S'agissant des technologies employées, le framework libre CodeIgniter, écrit en PHP, est utilisé pour le back-end, afin de profiter de ses atouts en matière de légèreté, de performance, de modularité et de sécurité. Côté front-end, la bibliothèque Bootstrap est utilisée tant pour son design apuré que pour la portabilité sur les versions mobile et tablette du projet ; et quelques éléments édités en Javascript sont également présents.

Pour compléter ce projet, et uniquement pour les besoins de l'examen, une version sous Wordpress a également été élaborée, incluant une partie e-commerce.

Les 2 versions sont disponibles aux adresses suivantes :

- <http://corrdis.fr/dev/jba/aupetitcreuxsympa/>
- <http://corrdis.fr/dev/jba/aupetitcreuxsympa/APCS-WP>

III – Présentation fonctionnelle

A) Contexte

Comme indiqué précédemment, le projet est apporté par un professionnel de la restauration rapide ayant ouvert assez récemment à CHAMBERET.

Bien que d'un certain âge et n'étant pas très à l'aise avec l'informatique, il a parfaitement conscience de ce que peuvent lui apporter des outils numériques pour son commerce. Avant notre accord, il avait déjà référencé son commerce sur Google Maps et ouvert une page sur Facebook.

Avoir un site vitrine en complément lui permettrait essentiellement de cibler la clientèle de passage, et de maintenir plus facilement à jour sa carte. Il souhaite également avoir la main sur la gestion du contenu du site.

B) Expression des besoins

Outre les exigences liées au référentiel pour le passage du présent examen, les besoins ont été clairement délimités par le client. Une grande partie des scénarios ont pu être d'ailleurs établis. Aussi, pour l'utilisateur, il s'agit de :

- Accéder à une page d'accueil avec une présentation du restaurant ;
- Pouvoir consulter la liste des plats et leur tarif ;
- Avoir accès à tous les moyens de contact du restaurant ;
- Localiser le restaurant et connaître ses horaires d'ouverture ;

Pour l'administrateur, il doit être possible de :

- Accéder à un espace personnel ;
- Consulter et gérer le contenu des menus ;

C) Détails des fonctionnalités

S'agissant d'un site vitrine assez simple, les fonctionnalités

sont limitées à l'essentiel.

1) Objectif

Il s'agit ici de présenter de manière exhaustive l'ensemble des informations pertinentes. Elles doivent être parfaitement lisibles, quelque soit le support, et gérable en toute autonomie pour l'administrateur.

2) Acteurs

Les personnes qui peuvent potentiellement utiliser le site sont :

- La clientèle habituelle qui souhaite se tenir informés des nouveautés ou des changements ;
- La clientèle passagère, le secteur géographique étant touristique ;
- L'administrateur pour gérer le contenu.

Les données renseignées sont importantes : sans informations à jour des prix et des horaires d'ouverture, le site perd tout son intérêt pour le visiteur.

3) Actions possibles sur le site

On cherche ici à délimiter les actions que les acteurs précédemment définis peuvent être amenés à effectuer.

• **Pour l'administrateur :**

Le restaurateur devra pouvoir se servir du site en toute autonomie, une fois la livraison faite. Pour se faire, un espace personnel sécurisé est mis en place, via un formulaire de connexion.

Une fois dans cet espace, un CRUD est mis à disposition pour permettre la création, la modification et la suppression des données qui sont affichées côté client.

- **Pour le visiteur :**

La fonction essentielle pour le visiteur est de pouvoir s'informer sur le restaurant et les plats, et éventuellement, entrer en contact avec le gérant. Le visiteur arrivera donc sur une page d'accueil à partir de laquelle il pourra naviguer vers ces différents éléments.

4) Fonctionnalités

Pour l'administrateur, il s'agit uniquement de la création d'un dashboard permettant la gestion des plats, via un système de CRUD. L'accès se fait via un formulaire de connexion avec email et mot de passe.

Le dashboard présente un formulaire dédié à l'enregistrement de nouveaux plats ; et un second est consacré à la modification et à la suppression de ce qui est déjà existant.

Concernant la partie visiteur :

- Une page d'accueil présentant le restaurant ;
- Une page présentant les plats et les tarifs ;
- Une page de contact.

L'ensemble de ces éléments sont accessibles par le menu. Le visiteur a également accès aux mentions légales.

IV – Choix de programmation

Bien que le client me laisse toute latitude sur les technologies à employer, j'ai cherché à approfondir mes connaissances sur CodeIgniter, notamment la version 4 sortie assez récemment. Pour le reste, je me suis cantonné aux langages appris au cours de la formation.

A) Technologies employées

1) Front-end

Les langages HTML et CSS¹ sont utilisés afin de structurer correctement non seulement le code lui-même mais aussi le style qui est appliqué à l'ensemble ou une partie de la page. En guise de soutien à cet aspect de la programmation, j'utilise jQuery pour les animations et certaines interactions, notamment via Bootstrap, que j'utilise également pour le responsive² et sa facilité à organiser l'affichage sur le navigateur.

a) HTML

Les balises HTML sont là avant tout pour bien structurer ce qui sera affiché sur le navigateur des visiteurs. Bien qu'étant incontournable et invisible pour l'utilisateur final, il est nécessaire de bien utiliser ces balises.

En effet, pour le développeur, notamment ceux qui peuvent potentiellement reprendre le projet ou ceux qui y collaborent, il permet visuellement identifier l'architecture de la page : en-tête, corps de la page, bas de page, titres, section, paragraphe, etc.

Il permet également d'appeler les fichiers utilisés pour le front-end, à l'instar des feuilles de style, les scripts, les appels à Bootstrap, ou encore les bibliothèques d'icônes comme FontAwesome.

Enfin, ces balises sont essentielles pour le référencement naturel. Bien que cela diminue avec le temps – Google mettant tout en œuvre pour mettre en avant son référencement payant – certaines balises doivent impérativement être utilisées avec une

¹ Hyper Text Markup Language et Cascading Style Sheets

² Paramètres permettant de s'adapter aux différents terminaux (écran, tablette, smartphone).

certaine attention, telles que <title> et <meta>. Les informations qui y figureront seront utilisées dans la présentation du site dans les moteurs de recherche. On peut également y faire figurer les informations à destination des robots, mais on peut également utiliser un fichier à part, comme c'est le cas dans mon projet.

b) CSS

Les feuilles de style sont des fichiers à part – du moins, il est préférable de les mettre à part plutôt que dans les fichiers structurant les pages – qui sont entièrement dédiés au style des pages.

Le style englobe non seulement le design mais également la disposition des éléments dans le navigateur. Dans le cadre de ce projet, Bootstrap allège considérablement ces fichiers par l'utilisation des class qui lui sont propres directement dans les balises HTML.

Dans le cas d'une utilisation exclusive, il est possible d'utiliser un fichier uniquement pour paramétrer l'affichage selon le terminal. On utilisera alors les media queries pour ajuster l'affichage des éléments selon si l'utilisateur est sur un ordinateur, une tablette ou un smartphone. Plus le format est petit, plus il est important de travailler sur les éléments afin de se concentrer sur les plus importants. Par exemple, dans le cadre de mon projet, la présentation des cuisiniers inclus leur photo : je fais en sorte que ces images n'apparaissent pas sur les plus petits terminaux, non seulement pour permettre une économie de données pour l'utilisateur mais également pour afficher clairement les informations.

Enfin, dans un second fichier, il est possible d'y regrouper les éléments uniquement liés au graphisme : personnalisation des polices, de la taille et des couleurs du texte, des images ou du fond, agencement au sein d'un même bloc ou entre les blocs, dimensionnement, etc. On peut même y gérer des animations tout aussi efficace et fluide qu'en Javascript, tout en gagnant en légèreté.

c) jQuery

jQuery est une bibliothèque open-source développée en JavaScript. Son but est de simplifier et d'accélérer le processus de

développement d'applications communiquant avec le serveur avec une ou plusieurs parties de la page grâce à la technologie AJAX³. Cette technologie permet ainsi d'éviter les temps de latence de création de page en effectuant des appels asynchrones. Le site web enverra donc uniquement les informations nécessaires à la mise à jour d'une zone de la page web.

Via cette bibliothèque, il est également possible d'ajouter des effets et autres transitions pour les éléments d'une page web afin d'améliorer l'expérience utilisateur.

Pour avoir plus d'informations à propos cette bibliothèque, vous pouvez vous rendre à cette [adresse](#).

d) Bootstrap

Bootstrap est une collection d'outils utiles à la création de sites web. Cette collection regroupe différents plugins jQuery pour afficher des composants comme des fenêtres modales, des tooltip et des panneaux rétractables.

Il fournit également une feuille de style CSS, qui contient les styles de base pour tous les composants HTML. Ceci a pour avantage de proposer une interface uniforme pour l'intégralité du site web. De plus, son système de grille permet de créer des applications responsives.

Une documentation plus détaillée est également disponible à cette [adresse](#).

2) Back-end

Mon choix s'est porté vers CodeIgniter car il m'a été chaudement recommandé par un de mes camarades lors de ma formation.

Pourquoi choisir un framework back-end ? En progressant dans la formation, j'ai réalisé que j'éprouvais certaines difficultés à assimiler le langage PHP et son articulation avec un SGBD, ainsi que la programmation orientée objet. Je cherchais donc un moyen pour pallier à cela, tout en respectant les exigences liées à la formation en termes d'architecture (MVC⁴) et de programmation objet.

En téléchargeant le framework, qui est maintenu et évolutif, j'ai pu constater immédiatement que les exigences que je viens

³ Asynchronous JavaScript and XML

⁴ Model-View-Controller, permettant la séparation des fichiers dans un souci de sécurité et de lisibilité.

d'évoquer sont également appliquées pour ce framework.

Ensuite, à l'usage, j'ai pu très rapidement le prendre en main, et me familiariser plus facilement à la programmation orientée objet et à m'organiser selon l'architecture MVC.

D'un point de vue plus théorique, ce framework présente de nombreux avantages.

Tout d'abord, il est très léger avec ses 17,7 Mo, comprenant un serveur fait maison pour faire tourner le projet en local, indépendamment de LAMP/WAMP. Il se démarque donc nettement des autres frameworks, mais cela se justifie par l'absence de certaines fonctionnalités que l'on peut retrouver ailleurs.

Il est également modulable : il dispose de bibliothèques propres, activables dans les fichiers de configuration. Et bien entendu, il est toujours possible de créer ses propres fonctionnalités si celles proposées ne conviennent pas avec précision à nos besoins.

La conséquence naturelle de tous ces éléments font qu'il s'agit également d'un framework rapide et fluide : puisqu'il nous permet d'utiliser que ce qui est nécessaire et qu'il a été optimisé en ce sens, il s'avère donc efficace.

Ce qui m'a également plu, c'est qu'en dehors du cœur du framework, qui est dans un dossier spécifique («vendor» par défaut), la programmation de l'application reste totalement de notre ressort. En effet, il ne s'agit pas de rentrer les informations en ligne de code pour que le framework nous prépare tous les fichiers bien comme il faut, à l'instar de Symfony. Cela correspondait totalement à mon envie de me perfectionner en PHP sans tomber dans la facilité.

A cette fin, il est mis à notre disposition tout un ensemble de classes parents à partir desquels des classes enfants pourront être créées pour les besoins de notre application. Une description de leur fonctionnement sera donnée plus bas.

Je présenterai plus bas les fichiers les plus essentiels sans pour autant entrer dans les détails, l'examen portant essentiellement sur le front-end.

3) Base de données

Ayant suivi un apprentissage essentiellement axé sur MySQL, c'est la technologie que j'ai employée dans ce projet, en utilisant le moteur InnoDB.

L'avantage est de pouvoir créer des jeux de données que l'on

peut mettre en relation entre elles via des requêtes spécifiques.

B) Contraintes/règles liées à ces choix

Comme indiqué précédemment, CodeIgniter est basé sur une architecture MVC. Il est donc important, pour assurer la maintenabilité du projet, de s'y conformer.

- **Utilisation de la base de données**

Tous les traitements et connexions avec la base de données doivent se faire dans les models. Pour accéder à la base de données les requêtes SQL habituelles fonctionnent, mais le framework met également des fonctions simplifiant leur rédaction.

- **Affichage des pages web**

Tous les affichages doivent se faire dans les views. Ces fichiers doivent regrouper que les instructions PHP renvoyant aux appels de la base de données, notamment les boucles ou la gestion des cas particuliers. Pour éviter toute répétition du code, on s'attachera à coder les parties communes à différentes vues dans un fichier commun.

- **Coordonner les données et les views avec les controllers**

Le controller est le chef d'orchestre de l'application. C'est par lui que doit passer les views et les models pour les coordonner en fonction des requêtes.

Pour en créer un, il faut créer une classe qui hérite de la classe CI_Controller. Par convention, on nomme la classe enfant avec la première lettre en majuscule.

A l'intérieur, on place toutes les fonctions nécessaires à l'affichage de la page, incluant les classes qui permettent le chargement des models nécessaires pour la page, l'appel aux bibliothèques ou helpers de CodeIgniter utiles, et l'appel aux fichiers views.

- **Paramétrer les routes**

Enfin, pour que les étapes précédentes permettent un affichage effectif du travail accompli, il faut paramétrer les routes. Dans le fichier du même nom présent dans le dossier config, il faut stocker dans la variable `$route` le nom des route qui apparaîtront dans l'URI et indiquer à quel controller et la méthode qui en dépend cela correspond. Par exemple, il est possible d'indiquer quel sera le controller par défaut :

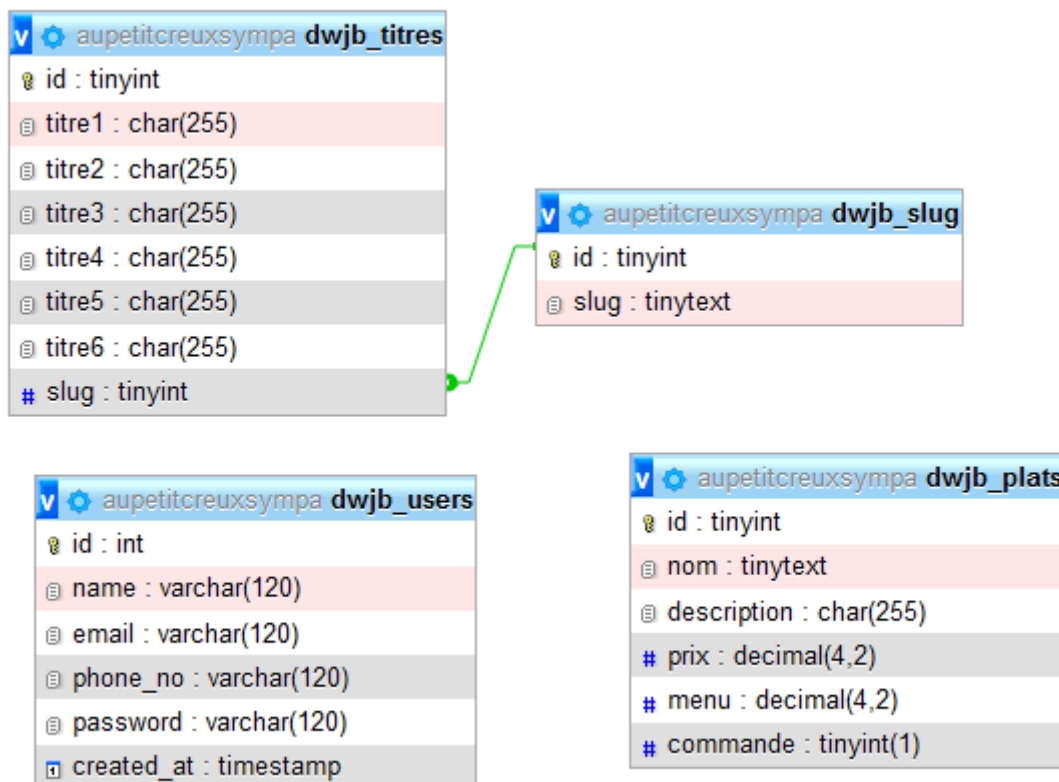
```
$routes->get('/', 'GlobalController::view/home');
```

Cela signifie qu'à l'arrivée sur le site, le controller «GlobalController» sera utilisé, sa méthode « view » et en paramètre « home ».

V – Présentation structurelle du projet

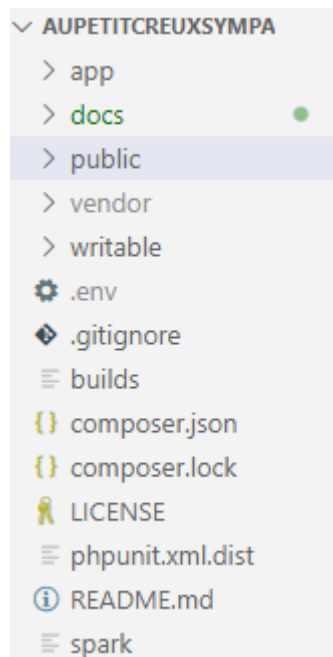
A) Modèle logique de données

Encore sujet à perfectionnement, les contenus ont été dissociés des uns des autres, notamment les données servant à s'identifier pour accéder à la zone administrateur :



B) Architecture physique

Comme indiqué précédemment, il s'agit de télécharger la dernière version de CodeIgniter pour ensuite la manipuler. Les dossiers et fichiers sont ainsi organisés :



S'agissant d'un framework back-end, je n'évoquerai que les fichiers les plus importants et qui sont mis à disposition des développeurs.

1) Le fichier Index.php

Tout d'abord, le fichier index.php présent dans le dossier Public. Ce fichier a pour objectif de traiter toutes les requêtes des navigateurs web. Il initialise le cœur de CodeIgniter ainsi que les constantes essentielles au système. Ensuite, il instancie la classe du contrôleur qui est appelée par l'adresse et lance la méthode spécifiée dans le 2ème segment de l'URL, si indiqué. Le format par défaut de l'URI est sous cette forme :

/index.php/nom_controleur/nom_methode

2) Le dossier App

Ensuite vient le dossier App, là où tout le travail doit s'opérer. On y trouve les dossiers suivants :

- **« config »**

Ce dossier contient plusieurs fichiers importants et certains sont des passages obligatoires pour le bon fonctionnement de l'application.

- Autoload.php : Contient la liste des contrôleurs, modèles et bibliothèques à charger automatiquement lors du chargement de CodeIgniter. Très utile pour charger par défaut des fonctionnalités natives de CodeIgniter qui sont utiles à l'application mais l'impact sur les performances peut se discuter ;
- Config.php : Contient les réglages de base pour le site (URL de base du site, encodage de base, caractères autorisés dans l'URL,...) ;
- Constants.php : Définit les constantes de base utiles à l'application ;
- Database.php : Contient la configuration pour l'accès à la base de données ;
- Doctypes.php : Définit les différents doctypes HTML supportés ;
- Email.php : Configure l'envoi des mails ;
- ForeignCharacters.php : Contient la table de conversion des caractères accentués vers des caractères ascii ;
- Migration.php : Ce fichier configure la migration de CodeIgniter. Ceci permet, par exemple, de mettre à jour une base de données d'une installation existante de manière organisée ;
- Mimes.php : Définit les différents types MIME⁵ de fichiers gérés par CodeIgniter ;
- Routes.php : Permet de modifier le routage du site web, c'est-à-dire le contrôleur à appeler en fonction des différentes adresses. Il définit également le contrôleur à appeler par défaut ;
- UserAgents.php : Définit les tableaux de « user-agent »⁶ qui permettent de distinguer les navigateurs web (détections des plateformes mobiles,...).

⁵ Multipurpose Internet Mail Extensions : est un standard d'Internet qui étend le format de données des courriels.

⁶ Chaîne de caractères qui définit une application cliente qui se connecte à Internet.

- **« controllers »**

On crée ici les controllers dont on a besoin. En l'espèce, je dispose :

- d'un fichier « BaseController.php », fichier pré-configuré par le framework, servant de super controller, dont tous les autres controllers dépendent en tant qu'enfants ;
- d'un fichier « GlobalController.php » permettant l'affichage de n'importe quelle page selon l'URL demandée ;
- d'un fichier « ContactController.php » regroupant les méthodes permettant l'envoi d'un mail de contact ;
- d'un fichier « AdminController » permettant la connexion à la zone administrateur et le paramétrage d'une session ;
- d'un fichier « DashboardController.php » regroupant les méthodes liées au CRUD.

- **« models »**

On regroupe dans ce dossier les fichiers qui interagissent avec la base de données. A ce jour, le projet dispose :

- d'un fichier « AdminModel.php » qui prépare la table Users utilisée pour l'identification de l'administrateur ;
- d'un fichier « PlatsModel.php » qui prépare la table des Plats pour le CRUD ;
- d'un fichier « ContentModel.php » qui permet de recueillir les données autres que celles relatives aux plats.

- **« views »**

Ce dossier contient les fichiers des différentes vues utilisées par CodeIgniter. Ces vues peuvent être organisées très librement, comme pour mon projet avec une partie dédiée à la zone administrateur, une autre pour la partie visiteur et une dernière pour la personnalisation des pages d'erreurs. Y est également présent les templates pour la partie visiteur et celui pour la partie administrateur.

On retrouvera dans ces fichiers le balisage HTML précédemment décrit ainsi que les variables PHP dans lesquelles

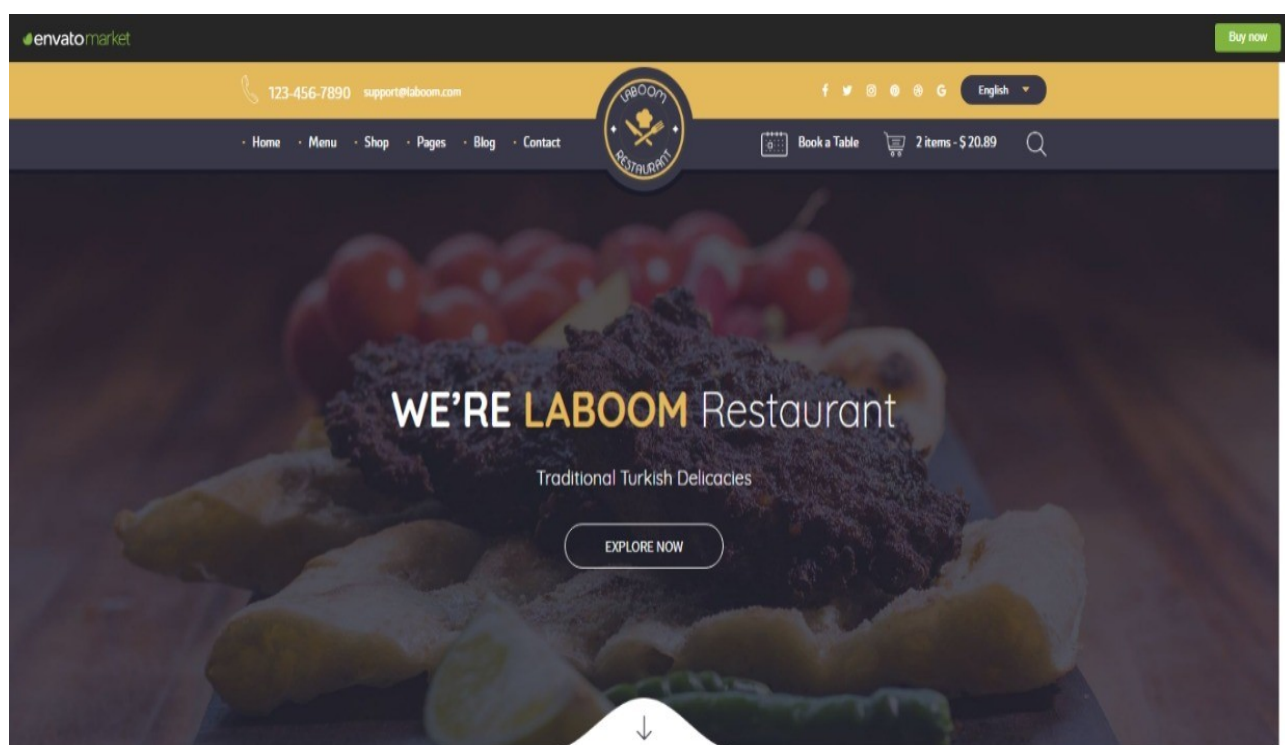
sont injectées les données. Il y aura également toutes les class Bootstrap et, selon les pages, les formulaires.

3) Le dossier Public

Ici figurent les fichiers liés au front-end : CSS, JS, images, favicon ; ainsi que le fichier index.php précédemment décrit, le htaccess et le fichier à destination des robots des moteurs de recherches.

4) Charte graphique

Au regard de la carte proposée par le restaurateur, un thème a été trouvé selon un exemple pris sur le web :



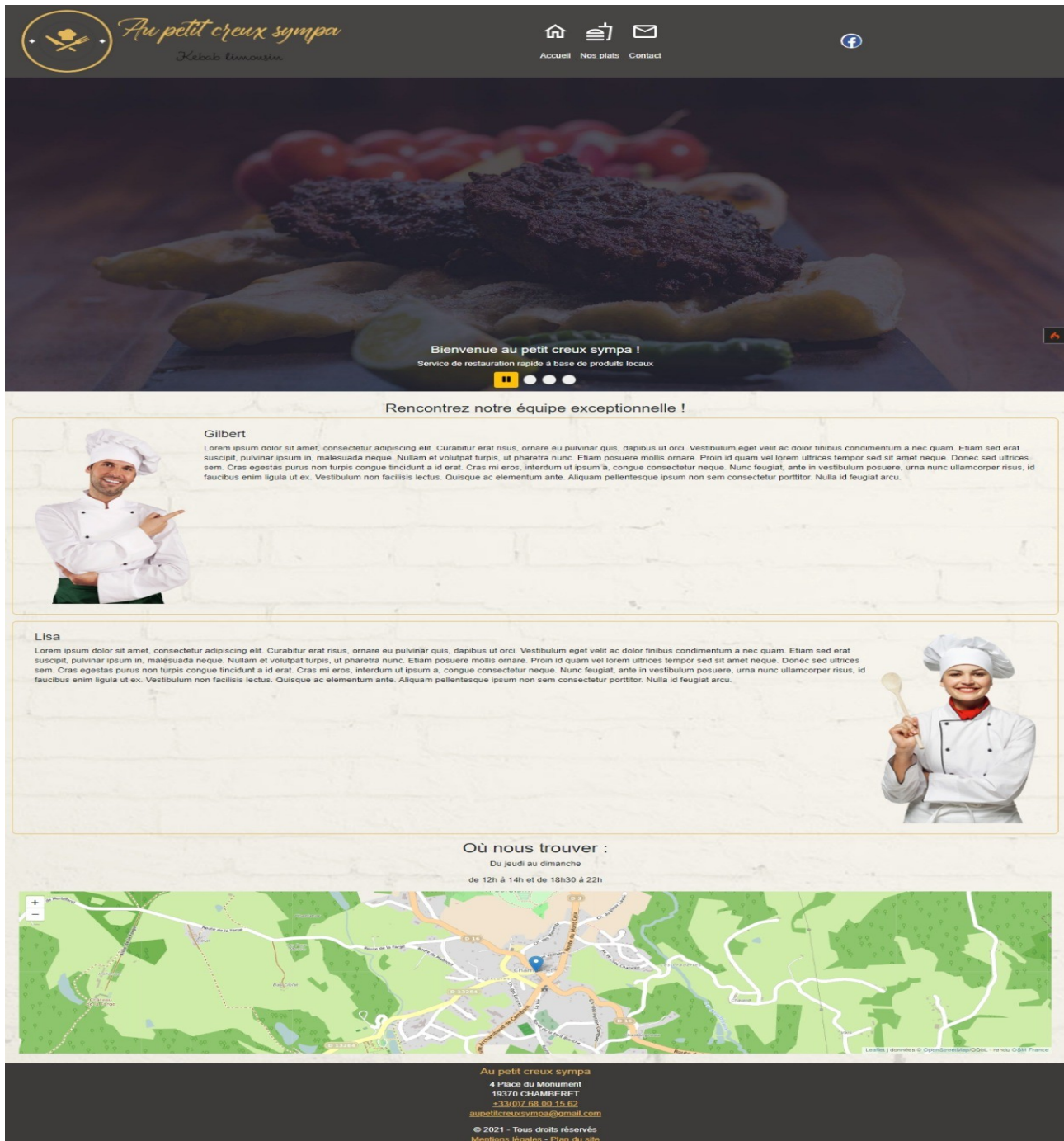
Le jaune est la couleur dominante du restaurant, la teinte retenue est la #e4b95b, et pour la mettre en valeur tout en conservant un certain contraste, une teinte de gris foncé est utilisé en background : #474444.

couleur : E4B95B


couleur : 474444

VI – Jeu d'essais




Page d'accueil :




Page des plats :



Au petit creux sympa
Kebab limousin

[Accueil](#) [Nos plats](#) [Contact](#)




Menus à emporter

Nos plats	Tarif du plat	Tarif du menu (plat + frites + boisson)
Kebab limousin dinde et veau Salade, tomate, oignon, sauce blanche, 150g de viandes (70/30)	6.00 €	8.00 €
Kebab limousin mouton et veau Salade, tomate, oignon, sauce blanche, 150g de viandes (70/30)	7.00 €	9.00 €
Sandwich boeuf Salade, tomate, oignon, fromage, sauce, 180g de viande	7.00 €	9.00 €
Sandwich merguez Salade, tomate, oignon, sauce, 3 merguez	6.00 €	8.00 €
Cheeseburger Salade, tomate, oignon, cheddar, sauce, 125g de viande	6.00 €	8.00 €
Cheeseburger enfant Steak haché, fromage fondant, sauce	4.00 €	6.00 €
Burger végétarien Salade, tomate, oignon, cheddar, burger végé	7.00 €	9.00 €
Fish and chips	8.00 €	
Petite Frites	1.00 €	
Grande Frites	2.00 €	
Boisson	2.00 €	
Supplément sauce	0.50 €	
Sur commande :		
Couscous Merguez	8.00 €	
Couscous Poulet et merguez	10.00 €	
Couscous Royal Merguez, poulet et mouton	12.00 €	
Pizza américaine	8.00 €	

Au petit creux sympa
4 Place du Monument
19370 CHAMBERET
+33(0)7 68 00 15 62
aupetitcreuxsympa@gmail.com


© 2021 - Tous droits réservés
[Mentions légales](#) - [Plan du site](#)

Page de contact :




Au petit creux sympa
Kebab limousin

[Accueil](#) [Nos plats](#) [Contact](#)



Contact

Civilité : 

*Nom :

Téléphone :


*Email :

*Objet de votre demande :

*Message :

Indisponible

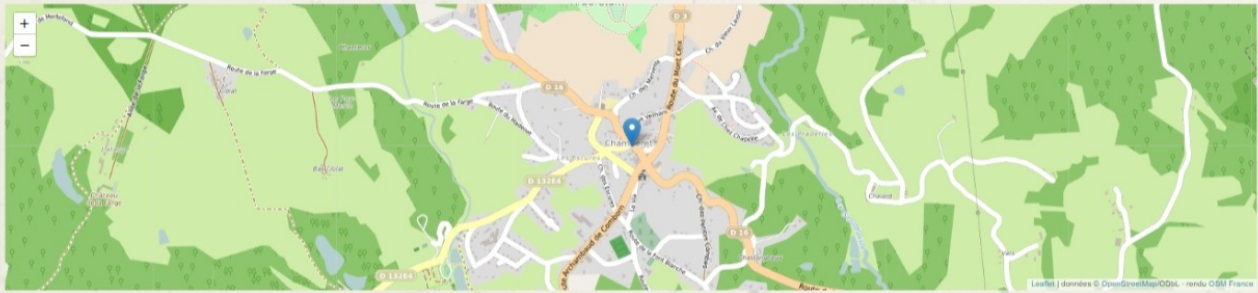
Les champs précédés d'une étoile sont obligatoires.



PAGE EN CONSTRUCTION / ENVOI INDISPONIBLE

Où nous trouver :

Du jeudi au dimanche
de 12h à 14h et de 18h30 à 22h



Au petit creux sympa
4 Place du Monument
19370 CHAMBERET
+33(0)7 68 00 15 62
aupeitcreuxsympa@gmail.com

© 2021 - Tous droits réservés
[Mentions légales](#) - [Plan du site](#)

Page de connexion admin et ajout de plat :

The screenshot shows the login page for 'Au petit creux sympa'. The header features the logo on the left, navigation links (Accueil, Nos plats, Contact) in the center, and a Facebook icon on the right. The main content area, titled 'Connexion', contains two input fields for 'Email' and 'Mot de passe', followed by a 'Valider' button. The footer displays the restaurant's name, address (4 Place du Monument, 19370 CHAMBERET), phone number (+33(0)7 68 00 15 62), email (aupetitcreuxsympa@gmail.com), and copyright notice (© 2021 - Tous droits réservés). It also includes links for 'Mentions légales' and 'Plan du site'.

Connexion

Email

Mot de passe

Valider

Au petit creux sympa
4 Place du Monument
19370 CHAMBERET
+33(0)7 68 00 15 62
aupetitcreuxsympa@gmail.com
© 2021 - Tous droits réservés
[Mentions légales](#) - [Plan du site](#)

The screenshot shows the 'Ajouter un plat' (Add dish) page. The header is identical to the login page but includes an additional link 'Ajouter un plat' in the navigation menu. The main content area contains four input fields for 'Nom du plat', 'Description', 'Prix', and 'Prix du menu'. Below these is a section for 'Sur commande ?' with radio buttons for 'oui' and 'non'. A 'Valider' button is at the bottom. The footer is the same as the login page.

Ajouter un plat

Nom du plat :

Description :

Prix :

Prix du menu :

Sur commande ?

☐ oui

☐ non

Valider

Au petit creux sympa
4 Place du Monument
19370 CHAMBERET
+33(0)7 68 00 15 62
aupetitcreuxsympa@gmail.com
© 2021 - Tous droits réservés
[Mentions légales](#) - [Plan du site](#)

23

VII – Veille sur les vulnérabilités de sécurité : l'OWASP

L'Open Web Application Security (OWASP) est un organisme à but non lucratif mondial qui milite pour l'amélioration de la sécurité des logiciels. L'objectif est d'informer les individus ainsi que les entreprises sur les risques liés à la sécurité des systèmes d'information. L'organisation fonctionne comme une communauté de professionnels qui partagent la même vision des choses. Tout le monde est libre de rejoindre la communauté qui compte aujourd'hui plus de 45 000 membres.

L'OWASP propose un guide de développement pour les applications web dans lequel se trouve les bonnes pratiques à adopter lors de la phase de développement d'un projet web. Des outils d'audits sont aussi mis à disposition des internautes par cette même organisation.

Ces professionnels délivrent régulièrement un top 10 des attaques les plus fréquentes dans le monde. Le dernier en date remonte à 2017 et concerne les vulnérabilités suivantes :

- Injection,
- Piratage de session,
- Exposition de données sensibles,
- Entités externes XML (XXE),
- Contournement du contrôle d'accès,
- Mauvaise configuration de sécurité,
- Cross-Site Scripting (XSS),
- Désérialisation non sécurisée (Insecure Deserialisation),
- Utilisation de composants présentant des vulnérabilités connues,
- Manque de surveillance et de monitoring.

Toutes ces failles sont finalement liées à un manque de vigilance et de rigueur dans le travail accompli. Il existe des bonnes pratiques que l'OWASP prône afin d'éviter, surtout pour les entreprises, des conséquences très graves non seulement pour leur réputation que pour leurs finances. Selon cette organisation, 80% des applications présentent au moins une faille ; même les grands groupes ne sont pas à l'abri, à l'instar de la fuite de données qu'a subi Facebook en 2018.

Dans le cas concret du présent projet, les développeurs ont bien entendu pris des précautions également par rapport aux problèmes de sécurité. Certaines de ces problématiques seront décrites plus bas, dans la partie relative à l'extrait anglophone à traduire.

VIII – Situation de recherches

Dans le cadre de ce projet, je devais chercher une alternative à Google Maps. Pour se faire, je me suis tourné vers OpenStreetMap, moteur de rendu propulsé par LeafletJS.

Leaflet est une librairie Javascript open-source et mobile-friendly. OpenStreetMap regroupe des cartographes bénévoles du monde entier autour de cet outil afin de maintenir à jour les routes et les structures locales.

Pour l'utiliser, il faut configurer les latitude et longitude du lieu que l'on souhaite cibler. Ensuite, il faut créer l'objet (intitulé map par exemple) et le faire pointer vers l'élément HTML qui l'accueillera. Cet objet est paramétrable : on y fait figurer les coordonnées GPS sur lesquelles la carte sera centrée, le niveau de zoom par défaut, la désactivation du zoom via le scroll de la souris, etc.

Leaflet ne fournissant pas les cartes, il faut préciser le moteur de rendu que l'on souhaite utiliser directement dans le code.

Enfin, on peut y faire figurer et personnaliser des marqueurs, des figures géométriques, des pop-ups, des images, etc.

Voici comment j'ai procédé pour ce projet, dont le rendu est visible plus haut :

```
// SCRIPT MAP

var lat = 45.5832236;
var lon = 1.7206023;
var macarte = null;
// Fonction d'initialisation de la carte
function initMap() {
// Créer l'objet "macarte" et l'insérer dans l'élément HTML qui a l'ID "map"
macarte = L.map('map', {
  center: [lat, lon],
  zoom: 15,
  scrollWheelZoom: false
});
// Leaflet ne récupère pas les cartes (tiles) sur un serveur par défaut. Nous devons lui préciser où nous souhaitons les récupérer. Ici, openstreetmap.fr
L.tileLayer('https://{s}.tile.openstreetmap.fr/osmfr/{z}/{x}/{y}.png', {
  // Il est toujours bien de laisser le lien vers la source des données
  attribution: 'données © <a href="//osm.org/copyright">OpenStreetMap</a>/ODBL - rendu <a href="//openstreetmap.fr">OSM France</a>',
  minZoom: 1,
  maxZoom: 20
}).addTo(macarte);
var marker = L.marker([lat, lon]).addTo(macarte);
}
window.onload = function(){
// Fonction d'initialisation qui s'exécute lorsque le DOM est chargé
  initMap();
};
```

IX – Extrait et traduction d'une documentation anglophone

Comme indiqué précédemment, je vais utilisé un extrait de la documentation de CodeIgniter relatif aux mesures de sécurité présentes au sein du framework.

Extraits :

« **XSS Filtering**

CodeIgniter comes with a Cross Site Scripting filter. This filter looks for commonly used techniques to embed malicious JavaScript into your data, or other types of code that attempt to hijack cookies or do other malicious things. [...]

Validate input data

CodeIgniter has a Form Validation Library that assists you in validating, filtering, and prepping your data.

Even if that doesn't work for your use case however, be sure to always validate and sanitize all input data. For example, if you expect a numeric string for an input variable, you can check for that with `is_numeric()` or `ctype_digit()`. Always try to narrow down your checks to a certain pattern.

Have it in mind that this includes not only `$_POST` and `$_GET` variables, but also cookies, the user-agent string and basically all data that is not created directly by your own code.[...] ».

Traduction :

Filtre XSS

CodeIgniter est fourni avec un filtre anti-Cross-Site Scripting. Ce filtre recherche les techniques habituellement utilisées pour injecter malicieusement du JavaScript dans vos données ou tout autre type de code qu tenterait de pirater les cookies ou tout autre chose malveillante.[...]

Validation des données entrantes

CodeIgniter dispose d'une librairie de validation des

formulaire qui vous accompagne afin de valider, filtrer et insérer vos données.

Même si elle ne fonctionne pas dans votre cas en particulier, veuillez bien à toujours valider et assainir toutes données entrantes. Par exemple, si vous attendez une donnée numérique dans une variable entrante, vous pouvez la vérifier avec `is_numeric()` ou `ctype_digit()`. Essayez toujours d'affiner vos vérifications suivant un certain modèle.

Gardez à l'esprit que cela n'inclut pas seulement les variables `$_POST` et `$_GET`, mais également les cookies, les chaînes de caractères des agents utilisateurs et plus généralement, toutes les données qui ne sont pas créées directement par votre propre code.
[...]

Conclusion

Ce projet permet une véritable mise en situation avec les attentes de ce secteur professionnel : véritable vitrine du client sur le web, il faut l'accompagner afin de créer un site à son image. S'ils se tournent vers des développeurs pour nous confier un projet, c'est qu'ils ont conscience de l'opportunité de posséder un site décrivant leurs prestations.

Dans le cadre de ce projet en particulier, la simplicité et l'efficacité étaient de mise : seules les informations les plus pertinentes pour la clientèle de ce restaurateur doivent y figurer. Cela vaut également pour la gestion en toute autonomie pour ce dernier. Bien que partageant une certaine proximité, il nous tenait à coeur de pouvoir délivrer un site facile d'accès et à prendre en main.

D'un point de vue purement personnel, cela me permet d'avoir le squelette d'un site réutilisable dans le cas où cette situation se représente ; et également, de m'en servir de laboratoire de recherches et de test afin de continuer à avancer vers un objectif d'amélioration de mes compétences. J'attends beaucoup notamment des retours de la clientèle une fois le site livré.

Annexes

Maquettage effectué avec Balsamiq