2021

Dossier de Projet Professionnel

Marwane Bellagha

Bleu Bois

19/07/2021

I. Table des matières

I. Liste des compétences du référentiel qui sont couvertes par le projet

A. 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

B. Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

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

II. Résumé du projet en français d'une longueur d'environ 20 lignes soit 200 à 250 mots, ou environ 1200 caractères espaces non compris

C'est installé à la manufacture MAKE ICI que Mathilde Bouchillou décide d'ouvrir son atelier (Bleu Bois) afin de pouvoir profiter aussi bien de l'infrastructure disponible sur le site ainsi que du climat d'émulation qui y règne ou l'on peut retrouver plusieurs corps de métiers. En effet, au sein de la manufacture, on peut retrouver un large panel d'artisan allant de la ferronnerie a l'ébénisterie a l'impression 3D.

Ayant déjà réussi à obtenir quelques partenariats avec d'autres artisans pour le compte de grandes sociétés de la région et réussissant à décrocher des clients par le biais du bouche-à-oreille ou en collaborant avec d'autres artisans de la manufacture ou par son réseau, Mathilde cherche à augmenter sa présence sur internet et de par la même accroitre sa visibilité grâce à un site web dans lequel on pourra retrouver ses produits phares, les collaborations réalisés avec d'autres et une sélection de commande réalisé pour des entreprises.

III. Cahier des charges, expression des besoins, ou spécifications fonctionnelles du projet

A. Cahier des charges

Élément	**Localisation / Pages Concernées**	**Fonctionnalités/ Définitions**	**Contraintes/ règles de gestion**	**Niveau de priorité (de 1à 5)**	**Temps Homme (en jour)**	**Spécifications Techniques**
MCD/ MLD	Projet Entier	Aperçu & Structure globale et simplifié de la base de données du projet		5	2	Un modèle conceptuel de données (MCD) est la représentation la plus abstraite des données d'un système d'information. Les données sont représentées sous forme d'entités et d'associations entre entité.[1] Un modèle logique de données (MLD) est la représentation des données d'un système d'information. Les données sont représentées en prenant en compte le modèle technologique qui sera utilisée pour leur gestion[2]
Maquette Fonctionnelle (Wireframe)	Projet Entier	Aperçu & Structure globale et simplifié du projet		5	2	La maquette fonctionnelle est un schéma qui montre l'agencement des parties composant une page web. Elle permet donc la visualisation des zones de texte,

[1] https://www.smartmodel.ch/home/questce/mcd
[2] https://www.smartmodel.ch/home/questce/mld

						l'emplacement des images, des vidéos, des liens, ainsi que des différents éléments graphiques.[3]
Charte Graphique	Projet Entier	Maquette plus approfondie du projet avec définition du thème		5	2	La charte graphique est un guide comprenant les recommandations d'utilisation et les caractéristiques des différents éléments graphiques (logos, couleurs, polices, typographies, symboles, calques..) qui peuvent être utilisés sur les différents supports de communication de l'entreprise. La charte graphique permet de garantir l'homogénéité et la cohérence de la communication visuelle au sein et en dehors de l'entreprise.[4]
Etablissement du cahier des charges	Projet Entier	Elaboration et accord sur les différentes fonctionnalités attendues dans le projet		5	2	Le cahier des charges formalise les besoins dans le cas d'un projet de site web, d'intranet ou d'application mobile. Il présente le contexte, les objectifs à atteindre, le niveau de qualité, les contraintes et le périmètre du projet. Il fixe la collaboration entre les différents acteurs. Le cahier des charges définit également les modalités financières et les aspects juridiques.[5]

[3] https://www.anthedesign.fr/webdesign-2/wireframe/
[4] https://www.definitions-marketing.com/definition/charte-graphique/

[5] https://yellowdolphins.com/publications/30-techniques-pour-vos-contenus/le-cahier-des-charges/

Page d'accueil	Index.php	Elaboration de la page d'accueil	Responsive	5	1	Page d'accueil réalisé en PHP à l'aide du framework Code Igniter et des media queries pour la partie responsive
Footer	Toutes les pages	Barre de navigation entre les différentes pages et les différents éléments du panel admin	Responsive	5	1	Footer réalise en HTML/CSS permettant la navigation entre les différentes pages du projet
Catégorie	Produit/catégorie	Pages de catégories présentant les produits par catégorie	Responsive	5	2	Page créé individuellement pour chaque catégorie selon leur id passé comme argument dans le Controller gérant l'affichage de la page d'accueil
Produit	uniqueproduit	Pages générées pour chaque produit avec formulaire de personnalisation et de prise de contact	Responsive	5	2	Page créé individuellement pour chaque produit selon leur id passé comme argument dans le Controller gérant l'affichage de la page d'accueil avec un formulaire de personnalisation des produits (caractéristiques stockés en bases de données) et un formulaire dans lequel le client laisse ses coordonnées.
Collaboration	Collaboration	Espace montrant les différentes collaborations avec d'autres artisans	Responsive	5	2	Affichage des différentes collaborations avec une image et un texte descriptif stockés en bases de données et pouvant être ajoutés et modifies par l'administrateur du site web depuis son panel d'administration

A propos/ Contact	Atelier	Présentation de l'entreprise et formulaire de contact	Responsive	4	2	Simples pages présentant brièvement l'atelier, l'artisane et la manufacture. Ainsi, qu'un petit de formulaire de contact permettant aux visiteurs de prendre contact directement avec l'administrateur. Le contenu de ce formulaire sera directement envoyé par mail à l'administrateur
Panel Admin	Admin	Panel de gestion des différent(e)s produits, catégories, collaboration et administrateur	Accessible uniquement par un admin verifié	5	5	Panel admin permettant la gestion des différents éléments stockés en bases de données notamment les produits, les collaborations, les administrateurs réalisé à m'aide d e différentes fonctions php en Code Igniter.

B. Expression des besoins

L'entreprise afin d'accroitre sa visibilité et de développer ses activités a exprimé le besoin d'avoir un site Web dans lequel on puisse retrouver tous ses produits que l'on peut les personnaliser selon certains critères définis au préalable mais aussi les différentes collaborations que l'entreprise réalise avec d'autres artistes. Tous ceci devant pouvoir être géré par le biais d'un panel d'administration.

C. Spécifications fonctionnelles du projet

1. Front-End

a) Description de l'existant

L'entreprise Bleu Bois n'a qu'une faible présence en ligne, elle apparait uniquement et très sporadiquement sur les réseaux. Elle utilise principalement des voies de communications plus anciennes. La grande majorité des clients sont amenés par bouche-à-oreille ou par des relations.

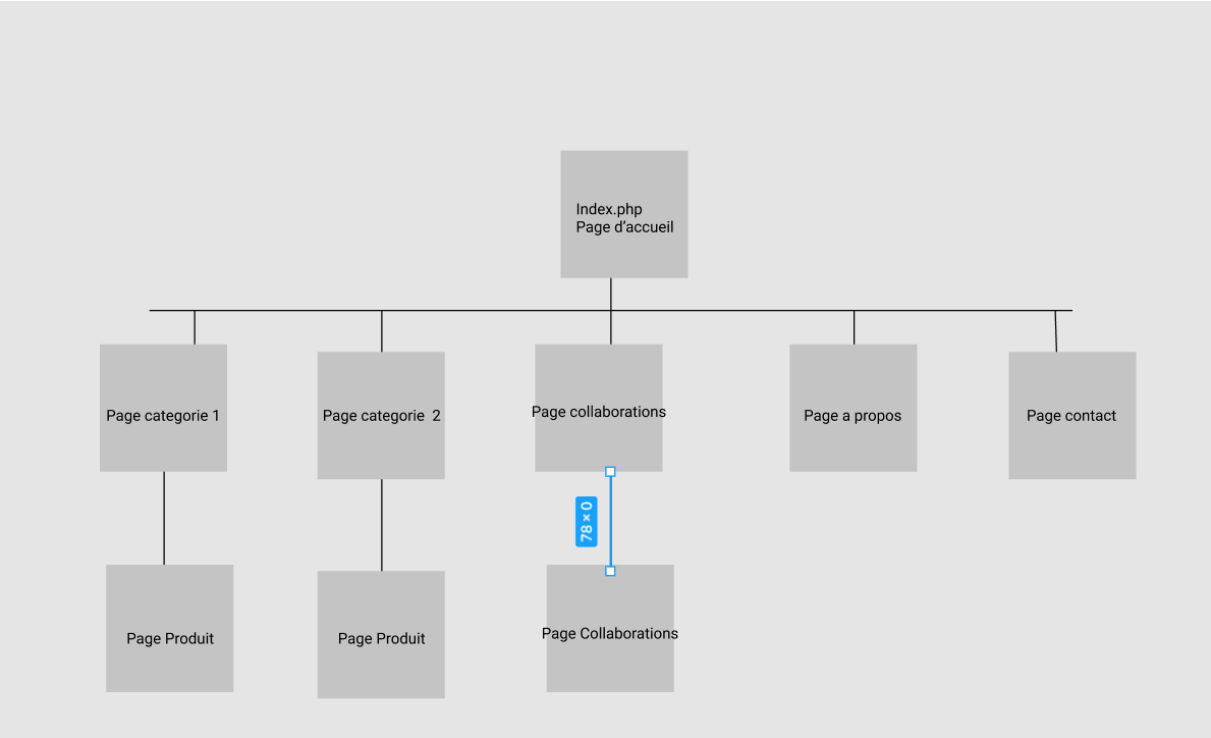
b) Périmètre du projet

Le projet consiste a effectuer, d'une part, une boutique en ligne présentant les différents produits mis en vente et fabriqués par l'entreprise, et d'autre part, de présenter les différentes collaborations réalisés avec différents artisans, enfin, d'avoir un panel d'administration permettant de gérer tous le contenu du site Web

c) Cible

La cible pour le site Web sont les potentiels nouveaux clients de Bleu Bois ainsi que les personnes intéressées par l'artisanat et ses produits, les personnes connaissant la manufacture MAKE ICI…

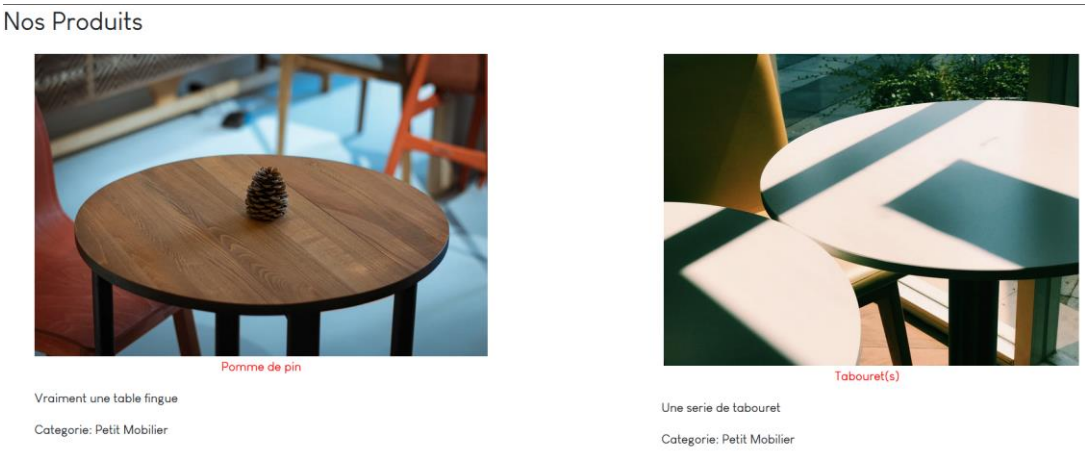
d) Arborescence



La page d'accueil est la page principale du projet, c'est par celle-ci que les visiteurs arrivent sur le site Web, elle présente brièvement sur quoi il va porter. Elle permet de naviguer entre les différentes pages. Elle relie aux pages catégories et collaborations qui permettent de se rendre sur des pages propres aux collaborations et aux produits.

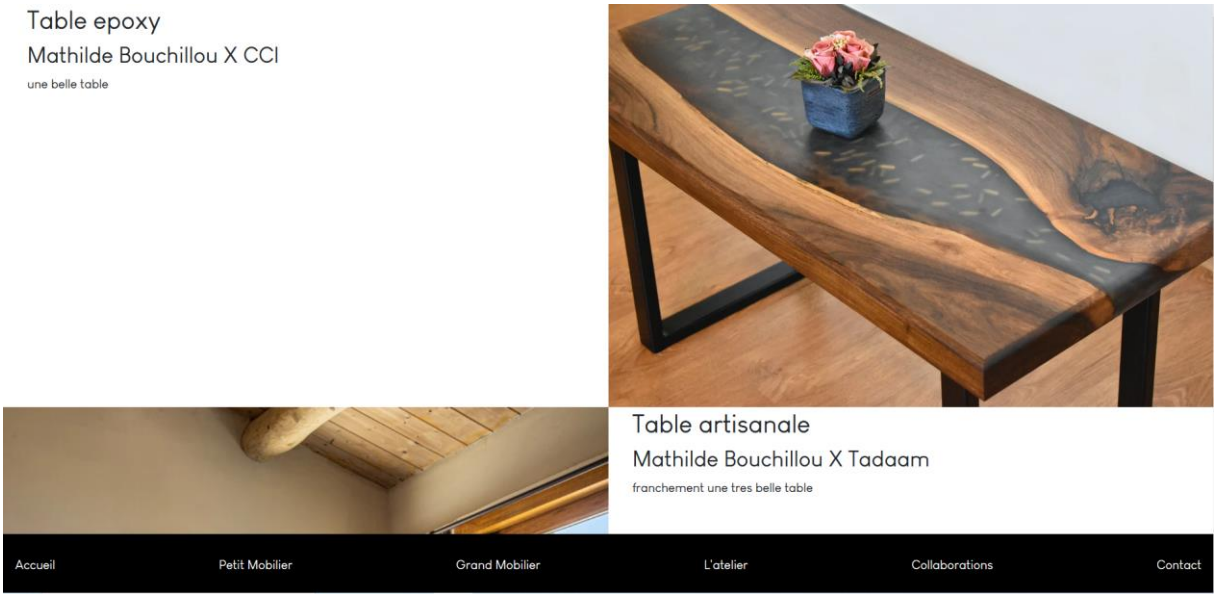
e) Description des fonctionnalités

(1) Catalogue produit



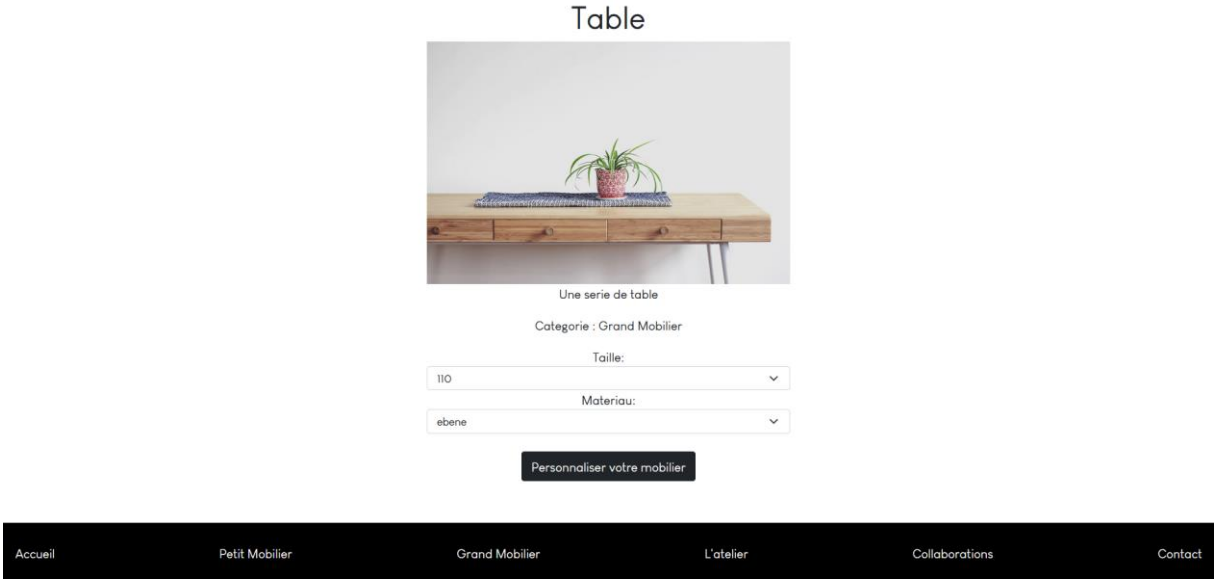
Le catalogue produit est divisé en deux sous catégories. Chacun étant créé [2]dynamiquement selon l'id de la catégorie. Chaque page permet de se rendre sur des pages de produit.

(2) Catalogue Collaborations



Ici, on peut voir les collaborations réalisées par l'artisane avec d'autres artisans. On peut y retrouver une image présentant la collaboration ainsi que le nom du collaborateur et une petite description pour chaque travail collaboratif

(3) Fiche produit



Ici, on a une page pour chaque produit reprenant la même structure de base avec une image, le nom du produit, sa catégorie, un formulaire permettant de le personnaliser et

le cas échéant un formulaire permettant de laisser ses coordonnées pour pouvoir être recontactés par l'entreprise.

(4) Back Office

IV. Spécifications techniques du projet, élaborées par le candidat, y compris pour la sécurité et le web mobile

A. Spécifications techniques

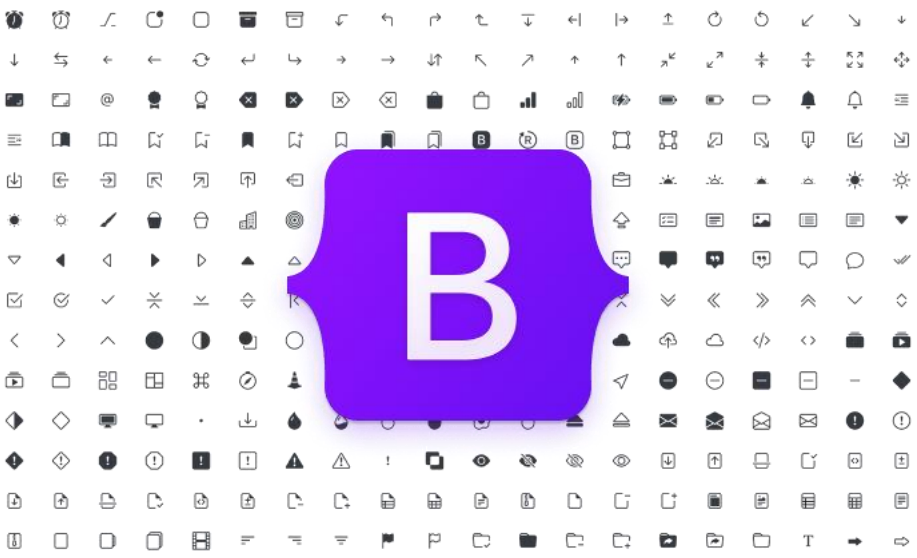
1. Front-End

a) Bootstrap

Bootstrap est un cadre de développement frontal gratuit et open source pour la création de sites et d'applications Web. Le cadre Bootstrap s'appuie sur HTML, CSS et JavaScript (JS) pour faciliter le développement de sites et d'applications réactifs et axés sur le mobile

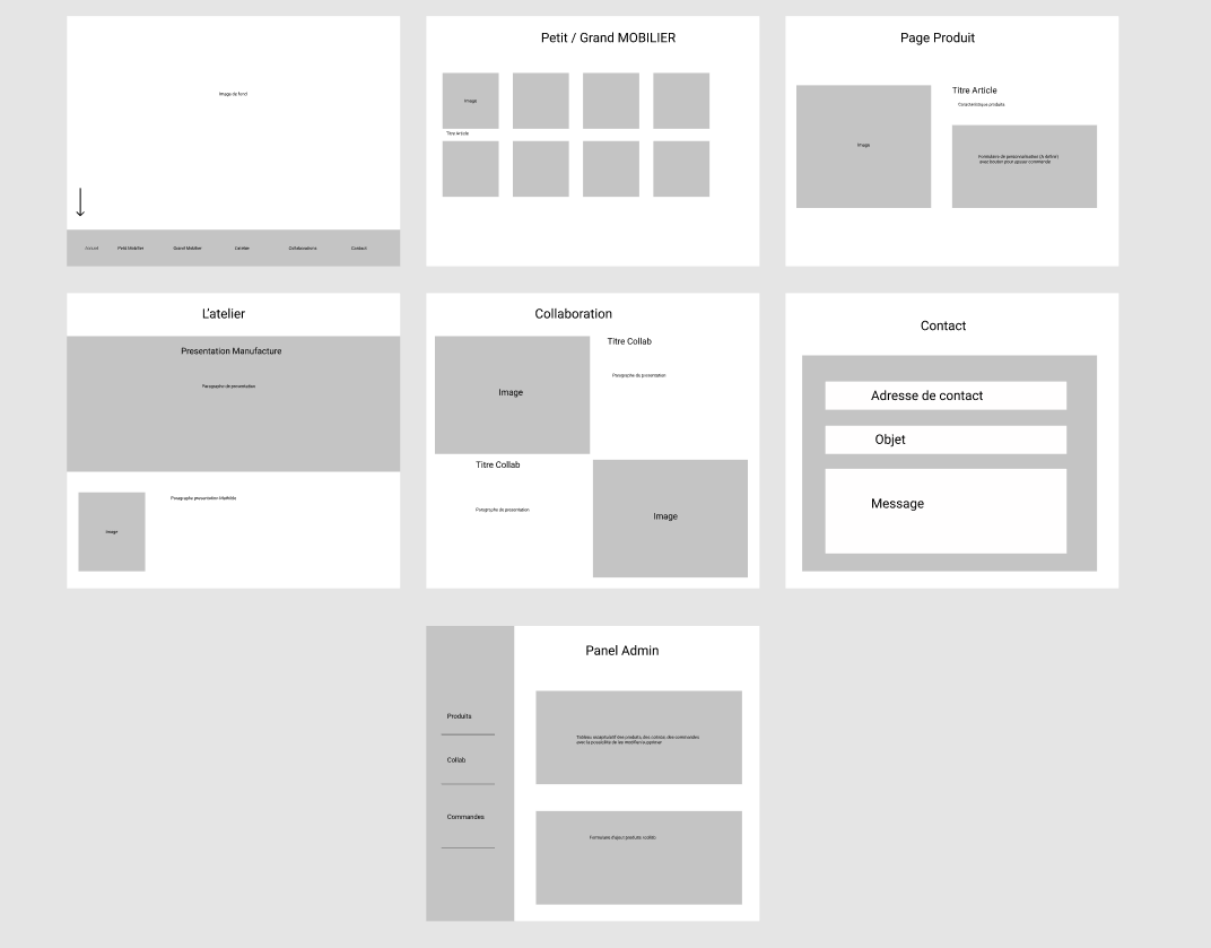
Le responsive design permet à une page web ou à une application de détecter la taille et l'orientation de l'écran du visiteur et d'adapter automatiquement l'affichage en conséquence ; l'approche "mobile first" part du principe que les smartphones, les tablettes et les applications mobiles spécifiques à une tâche sont les principaux outils des employés pour accomplir leur travail et répond aux exigences de ces technologies dans la conception.

Bootstrap comprend des composants d'interface utilisateur, des mises en page et des outils JS ainsi que le cadre de mise en œuvre. Le logiciel est disponible précompilé ou sous forme de code source.

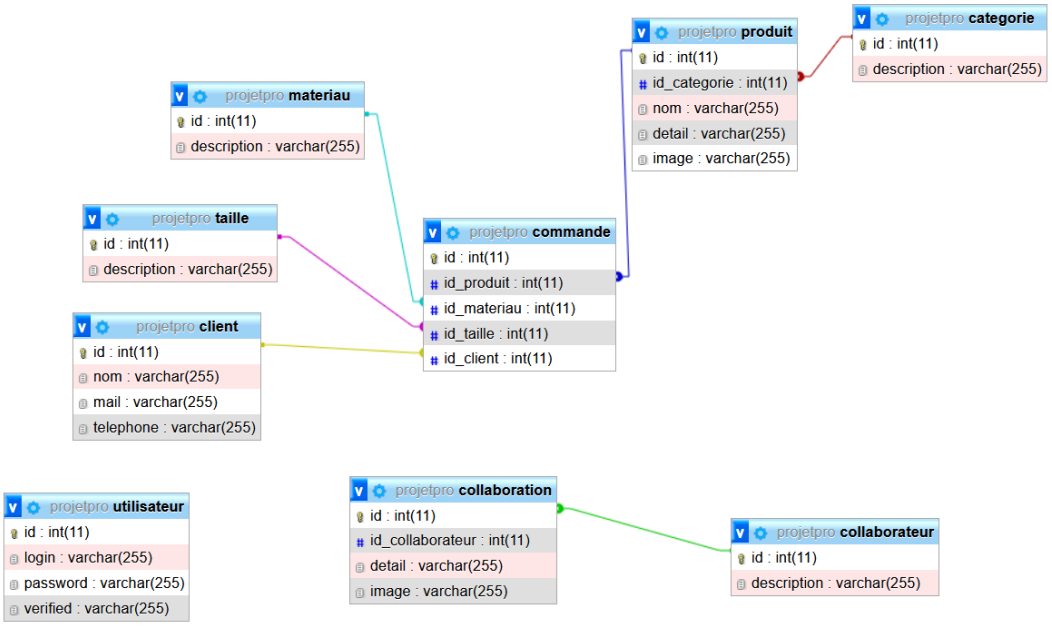


2. Back-End

a) Maquette fonctionnel / Wireframe



b) Modèle Conceptuel de Données (MCD)/ Modèle physique de données (MPD)



c) Code

d) Architecture

L'architecture *Modèle/Vue/Contrôleur* (MVC) est une façon d'organiser une interface graphique d'un programme. Elle consiste à distinguer trois entités distinctes qui sont, le *modèle*, la *vue* et le *contrôleur* ayant chacun un rôle précis dans l'interface.

L'organisation globale d'une interface graphique est souvent délicate. Bien que la façon MVC d'organiser une interface ne soit pas la solution miracle, elle fournit souvent une première approche qui peut ensuite être adaptée. Elle offre aussi un cadre pour structurer une application.

Dans l'architecture MVC, les rôles des trois entités sont les suivants.

- Modèle : données (accès et mise à jour)
- Vue : interface utilisateur (entrées et sorties)
- Contrôleur : gestion des événements et synchronisation

(1) Rôle du modèle

Le modèle contient les données manipulées par le programme. Il assure la gestion de ces données et garantit leur intégrité. Dans le cas typique d'une base de données, c'est le modèle qui la contient.

Le modèle offre des méthodes pour mettre à jour ces données (insertion suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ses données. Dans le cas de données importantes, le modèle peut autoriser plusieurs vues partielles des données. Si par exemple le programme manipule une base de données pour les emplois du temps, le modèle peut avoir des méthodes pour avoir, tous les cours d'une salle, tous les cours d'une personne ou tous les cours d'un groupe de Td.

(2) Rôle de la vue

La vue fait l'interface avec l'utilisateur. Sa première tâche est d'afficher les données qu'elle a récupérées auprès du modèle. Sa seconde tâche est de recevoir toutes les actions de l'utilisateur (clic de souris, sélection d'une entrée, boutons, …). Ses différents événements sont envoyés au contrôleur.

La vue peut aussi donner plusieurs vues, partielles ou non, des mêmes données. Par exemple, l'application de conversion de bases a un entier comme unique donnée. Ce même

entier est affiché de multiples façons (en texte dans différentes bases, bit par bit avec des boutons à cocher, avec des curseurs). La vue peut aussi offrir la possibilité à l'utilisateur de changer de vue.

(3) Rôle du contrôleur

Le contrôleur est chargé de la synchronisation du modèle et de la vue. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle et ensuite avertit la vue que les données ont changé pour que celle-ci se mette à jour. Certains événements de l'utilisateur ne concerne pas les données mais la vue. Dans ce cas, le contrôleur demande à la vue de se modifier.

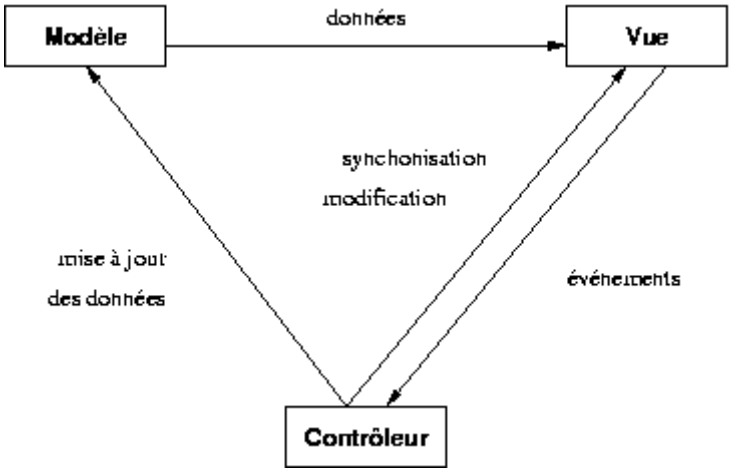
Dans le cas d'une base de données des emplois du temps. Une action de l'utilisateur peut être l'entrée (saisie) d'un nouveau cours. Le contrôleur ajoute ce cours au modèle et demande sa prise en compte par la vue. Une action de l'utilisateur peut aussi être de sélectionner une nouvelle personne pour visualiser tous ses cours. Ceci ne modifie pas la base des cours mais nécessite simplement que la vue s'adapte et offre à l'utilisateur une vision des cours de cette personne.

Le contrôleur est souvent scindé en plusieurs parties dont chacune reçoit les événements d'une partie des composants. En effet si un même objet reçoit les événements de tous les composants, il lui faut déterminer quelle est l'origine de chaque événement. Ce tri des événements peut s'avérer fastidieuse et peut conduire à un code pas très élégant (un énorme switch). C'est pour éviter ce problème que le contrôleur est réparti en plusieurs objets.

3. Interactions

Les différentes interactions entre le modèle, la vue et le contrôleur sont résumées par le schéma de la figure suivante.[6]

[6] https://www.irif.fr/~carton/Enseignement/InterfacesGraphiques/Cours/Swing/mvc.html



a) Framework

CodeIgniter est un cadre de développement d'applications - une boîte à outils - destiné aux personnes qui créent des sites Web en PHP. Son objectif est de vous permettre de développer des projets beaucoup plus rapidement que si vous écriviez du code à partir de zéro, en fournissant un riche ensemble de bibliothèques pour les tâches les plus courantes, ainsi qu'une interface simple et une structure logique pour accéder à ces bibliothèques. CodeIgniter vous permet de vous concentrer de manière créative sur votre projet en minimisant la quantité de code nécessaire pour une tâche donnée.

B. Sécurité

Une faille XSS consiste à injecter du code directement interprétable par le navigateur Web, comme du JavaScript ou du HTML. Cette attaque ne vise pas directement le site comme le ferait une injection SQL mais concerne plutôt la partie client c'est-à-dire vous (ou plutôt votre navigateur). Ce dernier ne fera aucune différence entre le code du site et celui

injecté par le pirate, il va donc l'exécuter sans broncher. Les possibilités sont nombreuses : redirection vers un autre site, vol de cookies, modification du code HTML de la page, exécution d'exploits contre le navigateur : en bref, tout ce que ces langages de script vous permettent de faire.[7]

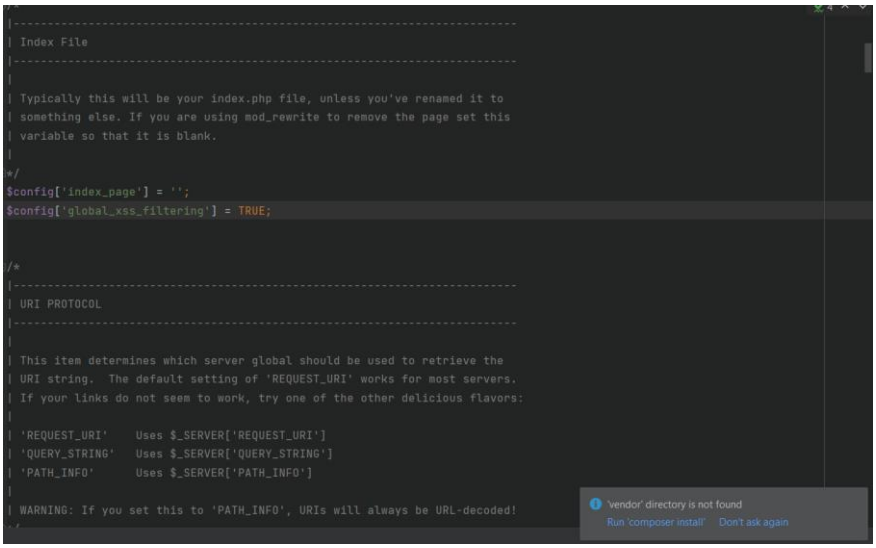
CodeIgniter est livré avec un filtre de prévention contre le Cross Site Scripting, qui recherche les techniques couramment utilisées pour déclencher le JavaScript ou d'autres types de code qui tentent de détourner les cookies ou de faire d'autres choses malveillantes. Si un élément non autorisé est rencontré, il est rendu sûr en convertissant les données en entités de caractères.

Un deuxième paramètre facultatif, is_image, permet d'utiliser cette fonction pour tester les images en vue d'éventuelles attaques XSS, ce qui est utile pour la sécurité du téléchargement de fichiers. Lorsque ce deuxième paramètre vaut VRAI, au lieu de renvoyer une chaîne altérée, la fonction renvoie VRAI si l'image est sûre, et FAUX si elle contient des informations potentiellement malveillantes qu'un navigateur peut tenter d'exécuter[8]

V. Réalisations du candidat comportant les extraits de code les plus significatifs et en les argumentant, y compris pour la sécurité et le web mobile

A. Sécurité

On peut voir ci-dessous, l'activation de l'option de Code Igniter 3 permettant la protection contre les failles XSS.



[7] https://zestedesavoir.com/articles/232/les-failles-xss/
[8] https://codeigniter.com/userguide3/libraries/security.html

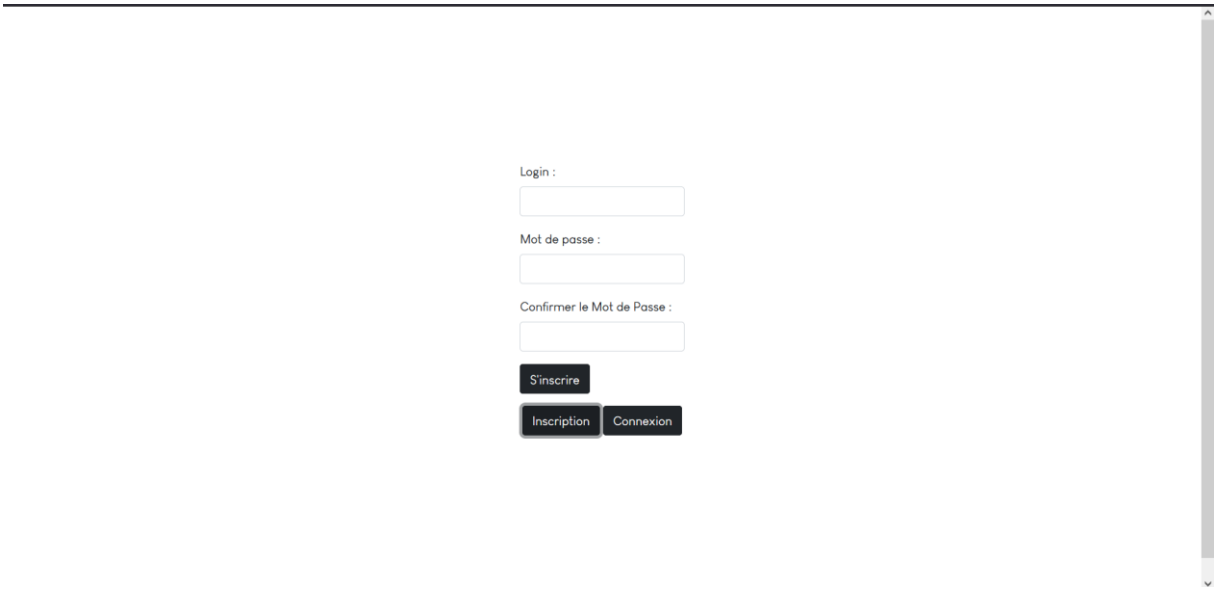
B. Web Mobile

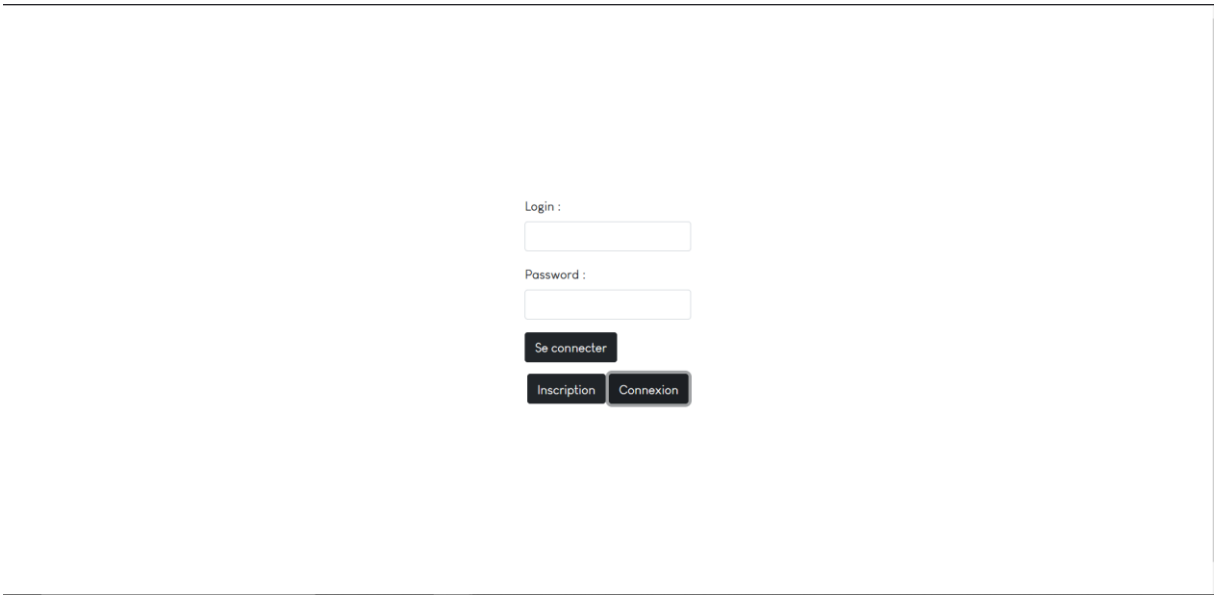
Ici, une media queries pour gérer la resonsivité du site Web

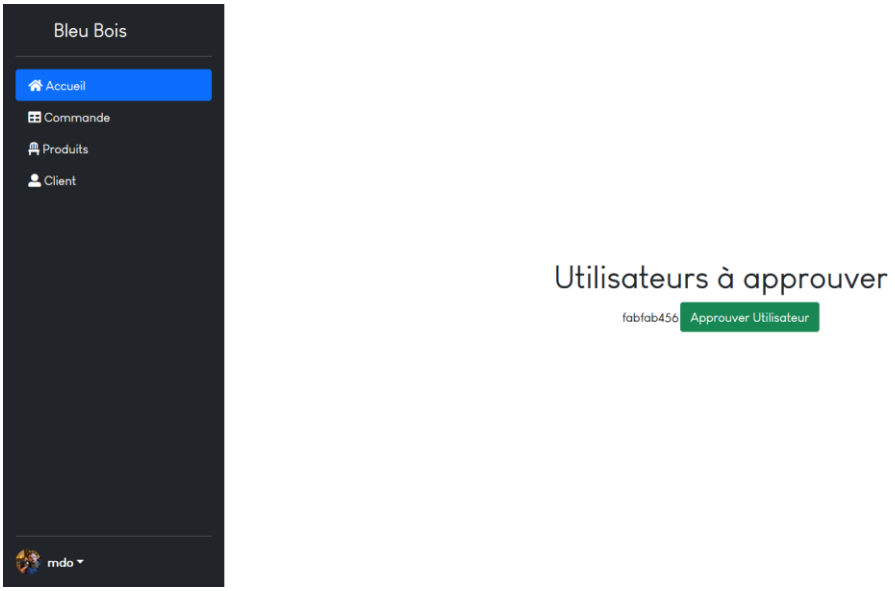
```
@media (max-width: 800px) {
    .herocontainer h1{
        letter-spacing: 15px;
    }
    .herocontainer h2 {
        letter-spacing: 10px;
    }
    }
```

VI. Présentation du jeu d'essai élaboré par le candidat de la fonctionnalité la plus représentative (données en entrée, données attendues, données obtenues)

A. Panel Admin







VII. Description de la veille, effectuée par le candidat durant le projet, sur les vulnérabilités de sécurité

A. Faille XSS

Les attaques de type Cross-Site Scripting (notée parfois XSS ou CSS) sont des attaques visant les sites web affichant dynamiquement du contenu utilisateur sans effectuer de contrôle et d'encodage des informations saisies par les utilisateurs. Les attaques Cross-Site Scripting consistent ainsi à forcer un site web à afficher du code HTML ou des scripts saisis par les utilisateurs. Le code ainsi inclus (le terme « injecté » est habituellement utilisé) dans un site web vulnérable est dit «malicieux».

Il est courant que les sites affichent des messages d'information reprenant directement un paramètre entré par l'utilisateur. L'exemple le plus classique est celui des «

pages d'erreur 404 ». Certains sites web modifient le comportement du site web, afin d'afficher un message d'erreur personnalisée lorsque la page demandée par le visiteur n'existe pas. Il est ainsi possible de faire afficher ce que l'on souhaite au site web en remplaçant par toute autre chaîne de caractère.

Ainsi, si aucun contrôle n'est effectué sur le contenu fourni par l'utilisateur, il est possible d'afficher du code HTML arbitraire sur une page web, afin d'en changer l'aspect, le contenu ou bien le comportement.

Grâce aux vulnérabilités Cross-Site Scripting, il est possible à un pirate de récupérer par ce biais les données échangées entre l'utilisateur et le site web concerné. Le code injecté dans la page web peut ainsi servir à afficher un formulaire afin de tromper l'utilisateur et lui faire saisir par exemple des informations d'authentification.

Par ailleurs, le script injecté peut permettre de rediriger l'utilisateur vers une page sous le contrôle du pirate, possédant éventuellement la même interface graphique que le site compromis afin de tromper l'usager.

Dans un tel contexte, la relation de confiance existant entre l'utilisateur et le site web est complètement compromise.

Lorsque les données saisies par l'utilisateur sont stockées sur le serveur pendant un certain temps (cas d'un forum de discussion par exemple), l'attaque est dite « persistante ». En effet, tous les utilisateurs du site web accès à la page dans laquelle le code nuisible a été introduit.

Les attaques dites « non persistantes » concernent les pages web dynamiques dans lesquelles une variable entrée par l'utilisateur est affichée telle quelle (par exemple l'affichage du nom de l'utilisateur, de la page en cours ou du mot saisie dans un champ de formulaire). Pour pouvoir exploiter cette vulnérabilité, l'attaquant doit fournir à la victime une URL modifiée, passant le code à insérer en paramètre. Néanmoins, une URL contenant des éléments de code Javascript pourra paraître suspecte à la victime, c'est la raison pour laquelle cette attaque est la plupart réalisée en encodant les données dans l'URL, afin qu'elle masque le code injecté à l'utilisateur.

B. Faille CSRF

La falsification de requêtes intersites (CSRF) est une attaque qui force un utilisateur final à exécuter des actions indésirables sur une application Web dans laquelle il est actuellement authentifié. Avec une petite aide d'ingénierie sociale (comme l'envoi d'un lien par courriel ou par chat), un attaquant peut tromper les utilisateurs d'une application Web pour qu'ils exécutent les actions de son choix. Si la victime est un utilisateur normal, une attaque CSRF réussie peut forcer l'utilisateur à effectuer des demandes de changement d'état telles que le transfert de fonds, la modification de son adresse électronique, etc. Si la victime est un compte administratif, CSRF peut compromettre l'ensemble de l'application Web.

C. SSRF

A partir d'une application web vulnérable, les SSRF permettent d'interagir avec le serveur, afin d'en extraire des fichiers et de trouver ses autres services actifs. Mais cela ne s'arrête pas là. Il est également possible de scanner le réseau interne afin d'en cartographier les IP et Ports ouverts.

L'idée générale est la suivante : si une fonctionnalité permet d'interagir avec des ressources (exemple : chargement d'une image dans l'application ou redirection vers une page), alors l'attaquant pourra tenter de soumettre une requête au serveur de telle sorte que la ressource recherchée soit interne au serveur (fichiers, services, ressources disponibles en localhost seulement) ou à son réseau.

D. SqLi

Les attaques par injection de commandes SQL sont des attaques visant les sites web s'appuyant sur des bases de données relationnelles.

Dans ce type de sites, des paramètres sont passés à la base de données sous forme d'une requête SQL. Ainsi, si le concepteur n'effectue aucun contrôle sur les paramètres passés dans la requête SQL, il est possible à un pirate de modifier la requête afin d'accéder à l'ensemble de la base de données, voire à en modifier le contenu.

En effet, certains caractères permettent d'enchaîner plusieurs requêtes SQL ou bien ignorer la suite de la requête. Ainsi, en insérant ce type de caractères dans la requête, un pirate peut potentiellement exécuter la requête de son choix.

Un certain nombre de règles permettent de se prémunir des attaques par injection de commandes SQL :

- Vérifier le format des données saisies et notamment la présence de caractères spéciaux ;
- Ne pas afficher de messages d'erreur explicites affichant la requête ou une partie de la requête SQL.
- Supprimer les comptes utilisateurs non utilisées, notamment les comptes par défaut ;
- Eviter les comptes sans mot de passe ;
- Restreindre au minimum les privilèges des comptes utilisés ;
- Supprimer les procédures stockées ;

E. LFI/RFI

L'objet de l'attaque, comme son nom l'indique, est d'inclure un fichier local (LFI) ou distant (RFI) au sein d'une ressource accessible depuis un SI. L'intérêt est multiple :

Dans le cas d'une LFI, cela permet par exemple :

- D'accéder au code source de fichiers privés stockés sur le serveur ciblé par l'attaque
- D'exécuter un script disponible sur le serveur dans un contexte non conventionnel (non prévu par le SI)

Dans le cas d'une RFI, cela permet par exemple :

- De faire exécuter par l'application un script stocké sur un serveur distant et construit sûr-mesure par le pirate
- De défigurer le site

Ces types d'attaques sont de moins en moins présentes dans les applications qui sont basées majoritairement sur des framework robustes. Mais ces vulnérabilités existent bel et bien, il est donc intéressant de connaître des méthodes d'attaques pour en mesurer la gravité.

Cette vulnérabilité est aussi couramment appelée “faille d’include” (en rapport avec le nom de la fonction PHP utilisée pour inclure un flux).

F. XXE

Elle se caractérise par la possibilité de lire des fichiers sur le serveur cible. Elle peut ainsi mettre en danger celui-ci, en accédant, par exemple, à un fichier de configuration contenant des mots de passe, en copiant les fichiers de la base de données ou en récupérant le code source d’une application.

Il est commun que certains sites web se servent de données utilisateurs pour créer des fichiers XML (ou reposant sur du XML, tel que des PDF), avant de renvoyer ceux-ci à l’utilisateur.

Il est donc possible pour un utilisateur malveillant d’essayer d’injecter dans le code XML des références à des entités externes, notamment des fichiers sensibles se situant sur le serveur. Puisque c’est le serveur qui va lire le XML, il va essayer de résoudre les entités externes avant de les incorporer au document final, qui sera ainsi renvoyé au client en contenant des données sensibles.

Pour être vulnérable, un service doit inclure trois comportements :

- Il doit être possible pour un utilisateur de manipuler du contenu XML qui sera ensuite parsé par le serveur. Cela peut se faire via un upload de fichier, un éditeur de texte (autorisant des formats XML) ou la réutilisation de données provenant du client dans un champ XML (par exemple les noms et prénoms, une adresse, ou un email).
- Le parseur XML autorise la définition et l’utilisation d’entité.
- Le parseur XML doit parser et interpréter les références externes dans les entités.

G. Attaque Serveur Web

Les premières attaques réseau exploitaient des vulnérabilités liées à l'implémentation des protocoles de la suite TCP/IP. Avec la correction progressive de ces vulnérabilités les attaques se sont décalées vers les couches applicatives et en particulier le web, dans la

mesure où la plupart des entreprises ouvrent leur système pare-feu pour le traffic destiné au web.

Le protocole HTTP (ou HTTPS) est le standard permettant de véhiculer les pages web par un mécanisme de requêtes et de réponses. Utilisé essentiellement pour transporter des pages web informationnelles (pages web statiques), le web est rapidement devenu un support interactif permettant de fournir des services en ligne. Le terme d'« application web » désigne ainsi toute application dont l'interface est accessible à travers le web à l'aide d'un simple navigateur. Devenu le support d'un certain nombre de technologies (SOAP, Javascript, XML RPC, etc.), le protocole HTTP possède désormais un rôle stratégique certain dans la sécurité des systèmes d'information.

Dans la mesure où les serveurs web sont de plus en plus sécurisés, les attaques se sont progressivement décalées vers l'exploitation des failles des applications web.

Ainsi, la sécurité des services web doit être un élément pris en compte dès leur conception et leur développement.

VIII. Description d'une situation de travail ayant nécessité une recherche, effectuée par le candidat durant le projet, à partir de site anglophone

Dès le début de l'utilisation du Framework PHP CodeIgniter, il a fallut dans un souci de lisibilité, de cohérence et de style, faire en sorte que la page index.php servant de routeur ne s'affiche dans toutes les URLs des pages du projet

IX. Extrait du site anglophone, utilisé dans le cadre de la recherche décrite précédemment, accompagné de la traduction en français effectuée par le candidat sans traducteur automatique

A. Extrait du site anglophone

CodeIgniter URLs

By default, URLs in CodeIgniter are designed to be search-engine and human friendly. Rather than using the standard "query string" approach to URLs that is synonymous with dynamic systems, CodeIgniter uses a segment-based approach:

example.com/news/article/my_article

Note

Query string URLs can be optionally enabled, as described below.

URI Segments

The segments in the URL, in following with the Model-View-Controller approach, usually represent:

example.com/class/function/ID

- The first segment represents the controller class that should be invoked.
- The second segment represents the class function, or method, that should be called.
- The third, and any additional segments, represent the ID and any variables that will be passed to the controller.

The URI Library and the URL Helper contain functions that make it easy to work with your URI data. In addition, your URLs can be remapped using the URI Routing feature for more flexibility.

Removing the index.php file

By default, the index.php file will be included in your URLs:

example.com/index.php/news/article/my_article

If your Apache server has mod_rewrite enabled, you can easily remove this file by using a .htaccess file with some simple rules. Here is an example of such a file, using the “negative” method in which everything is redirected except the specified items:

RewriteEngine On

RewriteCond %{REQUEST_FILENAME} !-f

RewriteCond %{REQUEST_FILENAME} !-d

RewriteRule ^(.*)$ index.php/$1 [L]

In the above example, any HTTP request other than those for existing directories and existing files is treated as a request for your index.php file.

Note

These specific rules might not work for all server configurations.

Note

Make sure to also exclude from the above rule any assets that you might need to accessible from the outside world.

Adding a URL Suffix

In your config/config.php file you can specify a suffix that will be added to all URLs generated by CodeIgniter. For example, if a URL is this:

example.com/index.php/products/view/shoes

You can optionally add a suffix, like .html, making the page appear to be of a certain type:

example.com/index.php/products/view/shoes.html

Enabling Query Strings

In some cases you might prefer to use query strings URLs:

index.php?c=products&m=view&id=345

CodeIgniter optionally supports this capability, which can be enabled in your application/config.php file. If you open your config file you'll see these items:

$config['enable_query_strings'] = FALSE;

$config['controller_trigger'] = 'c';

$config['function_trigger'] = 'm';

If you change "enable_query_strings" to TRUE this feature will become active. Your controllers and functions will then be accessible using the "trigger" words you've set to invoke your controllers and methods:

index.php?c=controller&m=method

Note

If you are using query strings you will have to build your own URLs, rather than utilizing the URL helpers (and other helpers that generate URLs, like some of the form helpers) as these are designed to work with segment based URLs.

B. Traduction

URLs de CodeIgniter

Par défaut, les URLs de CodeIgniter sont conçues pour être conviviales pour les moteurs de recherche et les humains. Plutôt que d'utiliser l'approche standard "query string" pour les URLs qui est synonyme de systèmes dynamiques, CodeIgniter utilise une approche basée sur les segments :

example.com/news/article/mon_article

Note

Les URLs de type query string peuvent être activées de manière optionnelle, comme décrit ci-dessous.

Segments d'URI

Les segments dans l'URL, en suivant l'approche Modèle-Vue-Contrôleur, représentent habituellement :

example.com/classe/fonction/ID

Le premier segment représente la classe de contrôleur qui doit être invoquée.

Le deuxième segment représente la fonction de classe, ou méthode, qui doit être appelée.

Le troisième segment, et tout autre segment supplémentaire, représente l'ID et les variables qui seront transmises au contrôleur.

La bibliothèque URI et l'URL Helper contiennent des fonctions qui facilitent le travail avec vos données URI. En outre, vos URL peuvent être remappées à l'aide de la fonction de routage des URI pour plus de flexibilité.

Suppression du fichier index.php

Par défaut, le fichier index.php sera inclus dans vos URL :

example.com/index.php/news/article/mon_article

Si le mod_rewrite est activé sur votre serveur Apache, vous pouvez facilement supprimer ce fichier en utilisant un fichier .htaccess contenant quelques règles simples. Voici un exemple d'un tel fichier, utilisant la méthode "négative" dans laquelle tout est redirigé sauf les éléments spécifiés :

RewriteEngine On

RewriteCond %{REQUEST_FILENAME} !-f

RewriteCond %{REQUEST_FILENAME} !-f !-d

RewriteRule ^(.*)$ index.php/$1 [L]

Dans l'exemple ci-dessus, toute requête HTTP autre que celles concernant les répertoires et les fichiers existants est traitée comme une requête pour votre fichier index.php.

Note

Ces règles spécifiques peuvent ne pas fonctionner pour toutes les configurations de serveur.

Note

Veillez à exclure de la règle ci-dessus toutes les ressources qui doivent être accessibles depuis le monde extérieur.

Ajout d'un suffixe d'URL

Dans votre fichier config/config.php, vous pouvez spécifier un suffixe qui sera ajouté à toutes les URLs générées par CodeIgniter. Par exemple, si une URL est la suivante

example.com/index.php/produits/vue/chaussures

Vous pouvez éventuellement ajouter un suffixe, comme .html, pour faire apparaître la page comme étant d'un certain type :

example.com/index.php/produits/vue/chaussures.html

Activation des chaînes de requête

Dans certains cas, vous préférerez utiliser des URL de type query strings :

index.php?c=produits&m=view&id=345

CodeIgniter supporte optionnellement cette capacitÃ©, qui peut Ãªtre activÃ©e dans votre fichier application/config.php. Si vous ouvrez votre fichier de configuration, vous verrez ces éléments :

$config['enable_query_strings'] = FALSE ;

$config['controller_trigger'] = 'c' ;

$config['function_trigger'] = 'm' ;

Si vous changez "enable_query_strings" en TRUE, cette fonctionnalité deviendra active. Vos contrôleurs et fonctions seront alors accessibles en utilisant les mots "déclencheurs" que vous avez définis pour invoquer vos contrôleurs et méthodes :

index.php?c=controller&m=method

Remarque

Si vous utilisez des chaînes de requête, vous devrez créer vos propres URL, plutôt que d'utiliser les aides URL (et d'autres aides qui génèrent des URL, comme certaines des aides de formulaire), car celles-ci sont conçues pour fonctionner avec des URL basées sur des segments.

X. Glossaire

Arborescence : organisation du contenu et des pages d'un site internet et les liens entre chaque page. Un site Web est constitué de contenu sur une variété de sujets et présenté sous la forme d'articles ou de pages. L'arborescence est vraiment le squelette ou la structure du site et montre la manière dont son contenu est groupé, lié et présenté au visiteur.[9]

MCD : outil de conception de base de données qui permet de définir la mise en oeuvre de structures physiques et de requêtes portant sur des données.[10]

MLD : structure logique globale d'une base de données, indépendamment du logiciel ou de la structure de stockage des données. Il constitue une représentation formelle des données

[9] https://www.1ere-position.fr/blog/comment-creer-arborescence-site/
[10] http://infocenter-archive.sybase.com/help/index.jsp?topic=/com.sybase.stf.poweramc.docs_12.5.0/html/dogu/dogup3.htm

nécessaires au fonctionnement d'une entreprise ou d'une activité commerciale, et contient le plus souvent des données qui ne sont pas encore mises en oeuvre dans la base de données physique.[11]

Maquette fonctionnelle ou Wireframe : schéma qui montre l'agencement des parties composant une page web. Permet donc la visualisation des zones de texte, l'emplacement des images, des vidéos, des liens, ainsi que des différents éléments graphiques.[12]

Framework : désigne en programmation informatique un ensemble d'outils et de composants logiciels à la base d'un logiciel ou d'une application.

Media Query : désigne un module, une spécification CSS3, qui permet d'adapter le contenu d'une page Internet à des conditions particulières. La plupart du temps, il est mis à contribution pour modifier l'affichage d'une page Web, selon la hauteur et la largeur de l'écran utilisé pour la navigation (écran d'ordinateur, écran de smartphone, écran de tablette numérique, etc.).

[11] http://infocenter-archive.sybase.com/help/index.jsp?topic=/com.sybase.stf.poweramc.docs_12.5.0/html/dogu/dogup3.htm
[12] https://www.anthedesign.fr/webdesign-2/wireframe/