

## [Dossier de projet]

---

[Création d'un site E-commerce]

[S.Nails]

Réalisation d'une boutique en ligne, afin de présenter les prestations d'un salon esthétique, afin vendre des produits de soins & mode, suite à la crise Covid

Présenté par </-- Olivier PUCHE--/>

Réalisé avec </-- Fabio TENORIO--/>

***Diplôme de développeur web et web mobile RNCP niveau V***



# [Introduction]

---

## 1.1 Quelques mots ?

Olivier Puche, originaire de Marseille, 45 ans, 2 enfants. Après une carrière de plus de 15 ans dans l'IT, entre 1999 et 2014, en tant qu'ingénieur commercial, business manager, j'ai également créé ma société de conseil en solutions informatiques, spécialisée en SIRH. Des soucis de santé et un choix de vie personnel, m'ont incité à bifurquer vers une voie différente, dans le milieu associatif, et plus particulièrement en gérant l'activité d'une association d'art.

En 2020, après une période de réflexion, assortie à un bilan de compétences, j'ai décidé de revenir dans le digital, mais dans des fonctions, un poste plus fonctionnel.

C'est pourquoi, j'ai entrepris cette première formation au sein de La Plateforme, afin de m'initier aux bases de la programmation.

À l'issue de cette 1ère formation, je suis accepté dans une école autre informatique (après avoir passé et réussi avec succès les tests et entretiens...) Ainsi, je poursuivrais mon processus de formation-reconversion par un Master (bac+5), durant les 2 prochaines années, afin d'obtenir un diplôme en tant que « chef de projet informatique ». Je recherche une alternance en tant que chef de projet digital, logiciel...

Mon objectif serait, sera d'obtenir un poste et des responsabilités afin de piloter, manager des projets de digitalisation. Ainsi, je pourrais allier mon passé professionnel de gestion de la relation client, du management et mes futures connaissances techniques, pour piloter des projets informatiques, digitaux, au sein d'une ESN, start-up ou la DSI d'une entreprise, voir peut-être créer une nouvelle entreprise.

## 1.2 École La Plateforme.

La Plateforme, campus méditerranéen de la tech, est une école basée à Marseille, qui existe depuis 2 ans. Elle forme aux métiers du Numérique et est labélisée par la Grande École du Numérique.

Elle dispense différents cursus (développement web et d'applications, sécurité et réseaux, intelligence artificielle). D'autres formations sont à venir : logiciels... cloud, depuis cet été, à travers un nouveau partenariat avec Amazon.

C'est une école inclusive, c'est à dire ouverte à tous, dont le coût est pris en charge, soit par les financeurs publics ou par le mécénat privé, soit les OPCO des entreprises à travers des contrats en alternance.

Le principe consiste à apprendre en réalisant à 100% de projet seul, à 2 ou à plusieurs.

### 1.3 La formation de Développeur Web et Web Mobile (pédagogie de l'école)

La formation Web est ouverte à tous, sans distinction d'âge, de sexe ou de parcours. Cette formation permet de se familiariser avec la programmation de base, que sont les langages HTML, CSS, PHP, JAVASCRIPT, et autres outils nécessaires tels que : MYSQL, divers frameworks, processus...

La pédagogie est axée à 100% sur un apprentissage par projet, ainsi que quelques Runtrack, afin de découvrir les nouveaux langages et ou concepts. La formation est émaillée ponctuellement de How To, une heure, lors de laquelle un focus est fait suite aux demandes des apprenants.

Le principe consiste à ce que chacun avance à son rythme, apprenne à apprendre seul, par des recherches diverses (tutoriel, documentation...). L'entraide entre les étudiants est au coeur du système et bien sûr la réalisation des nombreux projets.

#### ***Projet professionnel : boutique e-commerce***



## [Sommaire]

---

1. Liste des compétences du référentiel qui sont couvertes par le projet
2. Résumé du projet en français d'une longueur d'environ 20 lignes, soit 200 à 250 mots, soit environ 1200 caractères
3. Cahier des charges, expression des besoins, ou spécifications fonctionnelles du projet
4. Spécifications techniques du projet, élaborées par le candidat, y compris pour la sécurité et le web mobile
5. Réalisations comportant les extraits de code les plus significatifs et en les argumentant, y compris pour la sécurité et le web mobile
6. Présentation du jeu d'essai élaboré par le candidat de la fonctionnalité la plus représentative (données en entrée, données attendues, données obtenues)
7. Description de la veille, durant le projet, sur les vulnérabilités de sécurité
8. Description d'une situation de travail ayant nécessité une recherche, effectuée par le candidat durant le projet, à partir de site anglophone
9. 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 (environ 750 signes).
10. Annexes (zoning, maquettage, schéma base de données)

# 1. [Liste des compétences du référentiel qui sont couvertes par le projet]

---

Le projet couvre les compétences énoncées ci-dessous.

## 1. Pour l'activité 1 : "Développer la partie front-end d'une application web et web mobile en intégrant les recommandations de sécurité".

- Maquetter une application

Avant de commencer le codage du site web, mon binôme et moi-même avons listé toutes les fonctionnalités du site et les différentes pages qui devraient être créées.

Nous avons par la suite réalisé une maquette sur papier pour avoir les grandes lignes du design du site.

**Nous avons ensuite créé « une maquette » sur Figma qui représentait une ébauche du design** (hors couleur définitive du site), les principales pages et les diverses informations contenues dans chaque page, ainsi que le détail des informations de la page accueil.

Le client nous a ensuite donné son avis, fait des demandes et nous avons réalisé le codage.

- Réaliser une interface utilisateur web ou mobile statique et adaptable

Le site a été réalisé afin de permettre à l'utilisateur d'accéder facilement aux informations principales dès la page accueil. L'utilisateur peut accéder à l'ensemble des informations d'une page à une autre (connexion, profil, produits, soins, panier...)

**Le menu est compris dans le header, et les informations indispensables dans le footer (« Nous contacter », RGPD, Plan Google Maps, nos coordonnées). Le header et le footer sont responsive.**

L'objectif est d'accéder en lecture et expérience utilisateur en le moins de « CLIC » possible, aux informations voulues.

- Développer une interface utilisateur web dynamique

**Tous les visiteurs peuvent accéder aux pages du site. L'historique de connexion au site sera conservé en cookie par le navigateur de l'ordinateur du visiteur.**

Toutefois, pour utiliser l'agenda, dès qu'il sera activé, il sera demandé au visiteur de créer un compte membre et pour valider un panier et passer une commande de créer un compte client, qui reprendra les informations membre en automatique, si l'utilisateur est déjà membre.

- Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce

Le visiteur peut visualiser dès la page accueil des informations synthétiques (promotions, soins, produits). De même, il est invité à se connecter au site Instagram, et un lien est présent dans chaque page. Un menu en header et footer permet l'accès facile aux informations des autres pages, **ainsi qu'un bouton recherche développé en Javascript, pour les soins et produits.**

#### ***Pour passer une commande :***

Le client se connecte au site ; renseigne son login et mot de passe ; effectue une recherche via la barre de recherche ou dans la page produit.

Il ajoute au panier, puis valide la commande et renseigne ses coordonnées bancaires.

Volontairement, nous avons conseillé à la cliente de ne pas enregistrer dans la base de données des informations trop confidentielles (coordonnées bancaires) afin de ne pas avoir la responsabilité de piratage de données.

Le prénom et le nom, afin de connaître l'identité des membres/clients.

La date anniversaire est une information « sensible », mais cependant utile pour personnaliser des offres de soins et produits.

L'email et le téléphone lui permettront de faire des campagnes marketing (e-mailing, sms, whatsaps...)

## **2. Pour l'activité 2 "Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité".**

- Créer une base de données

**Nous avons réalisé un MCD et une MPD, ce qui nous a permis de mieux comprendre les liens entre chaque table et la nécessité de créer 1 table de liaison ligne de commande et une table de liaison promotion produit.**

- Développer les composants d'accès aux données

**Le formulaire a la capacité de se pré-remplir par rapport aux données enregistrées précédemment dans la BDD SQL**, afin que l'utilisateur puisse visualiser les informations de son profil et les modifier.

Idem, en ce qui concerne l'historique des achats.

Une barre de recherche permet de filtrer l'accès aux produits ou soins souhaités.

**La page administrateur synthétise l'ensemble des informations des utilisateurs, des produits, des fournisseurs...**

- Développer la partie back-end d'une application web ou web mobile

Pour se faire nous avons structuré l'ensemble de notre **site en MVC en utilisant le langage PHP**. **Les requêtes SQL dans le modèle**, les pages HTML côté utilisateur dans les vues et le pilote permettant de communiquer dans entre les vues et les modèles dans les contrôleurs. À la racine **.htaccess qui permet de définir les règles de réécriture générales**, la page **index.php (qui est également le routeur)** qui est la page sur laquelle arrive tous les utilisateurs et qui définit les différentes adresses url des pages, l'autoloader afin d'appeler les classes de n'importe quelle page sans multiplier les require de fichiers.

Enfin, nous avons également utilisé **Namespace afin de raccourcir l'écriture du code**. Un contrôleur et un modèle général qui définissent les règles d'accès des utilisateurs et à la base de manière générique.

- Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce.

Pour se faire nous avons utilisé **la bibliothèque Boootraps, installé Composer, Stripe** comme moyens de paiements, HP mail (en cas d'oubli du mot de passe) ainsi que Google Agenda pour la prise de rendez-vous, qui informe instantanément la cliente qui peut confirmer ou annuler le rendez-vous.

## 2. [Résumé du projet en français d'une longueur d'environ 20 lignes]

---

Ce dossier présente un des projets effectués au cours de ma formation au sein de La Plateforme, afin d'utiliser l'ensemble des langages, outils, process... étudiés et ainsi valider les compétences nécessaires à l'obtention du diplôme de développeur web et web mobile.

Il s'agit d'un site e-commerce, pour permettre à S.nails, un salon de soins esthétiques de promouvoir ses prestations. Par la suite, la gérante envisagerait de permettre à ses clients de prendre des rendez-vous, via un éventuel acompte à verser (afin de bloquer définitivement le créneau) Le salon a été touché par la crise Covid, la cliente souhaite réaliser des ventes de divers produits de soins et, ou accessoires de modes auprès de ses clientes, des amis/rerelations de sa clientèle, ainsi que sa communauté Instagram, afin de développer sa visibilité, un CA complémentaire à son activité principale via les ventes réalisées sur le site.

Ce projet a repris « des bases » du projet boutique, auquel de nombreuses fonctionnalités ont été rajoutées : Agenda, Composer/Stripe pour le paiement, Bootstraps pour la mise en forme et le côté responsive du site et surtout du JAVA SCRIPT (dernier langage étudié, en fin de parcours)

Nous avons également réalisé un maquetage, un MCD du projet avec les outils adéquates.

Au départ, la cliente a envisagé de déployer un véritable blog. Cependant étant très présente sur INSTAGRAM (13,5 K abonnés), n'ayant ni le temps, ni les compétences pour maintenir un blog, au final, elle a décidé de mettre un lien vers Instagram, pour créer de l'interactivité avec les utilisateurs.

Le site a été codé de mai à juin 2021, en binôme avec Fabio TENORIO, un autre stagiaire de ma promotion.

Aucun site n'existait auparavant, une simple page, une communauté Instagram. Le site a été réalisé en français et doit être accessible aux différents formats (ordinateur, tablette, téléphone), sur les différents navigateurs.



### 3. [Cahier des charges, expression des besoins, ou spécifications fonctionnelles du projet]

---

- Créer le premier site e-commerce pour le salon esthétique et la cliente. Nous avons fait le choix de Stripe pour permettre le paiement.
- Au départ, un blog était envisagé, mais la cliente est présente sur Instagram, un réseau social auquel elle est habituée, sur lequel elle a sa communauté (de nombreuses vidéos...) et qui est plus propice à son activité, des soins, la mode...
- Faire en sorte que l'utilisation du site soit la plus simple pour la gérante et sa clientèle (simple visiteur, membre, client). Utilisation de HP Mailer pour l'utilisateur, Stripe...
- Créer une base de données pour les utilisateurs, les enregistrements des formulaires, les produits, les achats.
- Créer un espace administrateur afin de faciliter la gestion des utilisateurs, la mise en ligne des produits à vendre et visualiser tous les achats d'un client, regroupe les produits par catégories et fournisseurs.
- Mettre à disposition « un agenda » qui pourrait être activé en fonction des besoins de la cliente. Un agenda a d'abord été développé, puis nous avons fait le choix de mettre une API, en lien direct avec l'agenda Google (plus intuitif, plus facile à gérer pour la gérante)
- **Arborescence du site :**
  - Page accueil (*synthèse de toutes les informations du site et lien vers la page Instagram*)
  - Page connexion & ou page d'inscription
  - Page profil (*modifier profil, visualisation du panier et historique des achats*)
  - Page soins
  - Page produits
  - Page détail produit
  - Page/liens agenda devenu Google Agenda
  - Page panier
  - Page validation de la commande & paiement
  - Page/bouton nous contacter
  - Page RGPD (mentions légales)
  - Page/liens Google Maps
  - Page administrateur (*tableaux de bord : tous les utilisateurs, produits, catégories et fournisseurs et insertion d'un nouveau produit*)
  - Page/liens (oubli du mot passe)
  - Page error 404

## 4. [Spécifications techniques du projet, élaborées par le candidat, y compris pour la sécurité et le web mobile]

---

### 1. Technologies et environnement

Les technologies utilisées pour la partie front-end sont :

- Le projet a été réalisé en langages de programmations HTML et CSS
- Boostrops, bibliothèque HTML/CSS a également été utilisée
- Javascript côté utilisateur (dynamisme et fluidité)

Les technologies utilisées pour la partie back-end :

- Le projet a été réalisé en langage de programmation PHP
- La base de données avec SQL (avec Mysql)
- Le gestionnaire de dépendances est Composer

L'environnement de développement est le suivant :

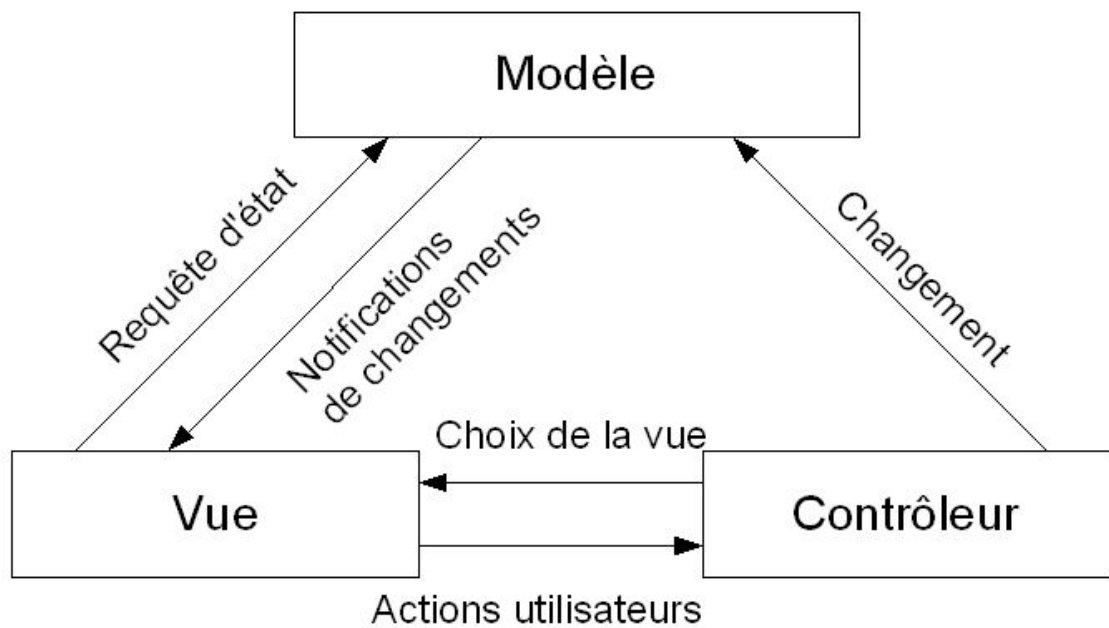
- Editeur de code : Visual Studio Code
- Outil de versioning : GIT
- Maquettage : Figma
- Schéma base de données : Lucidchart

### 2. Architecture du projet : développé avec un design pattern MVC (Model-View-Controller)

L'architecture MVC est l'une des architectures les plus utilisées pour les applications Web, elle se compose de 3 modules :

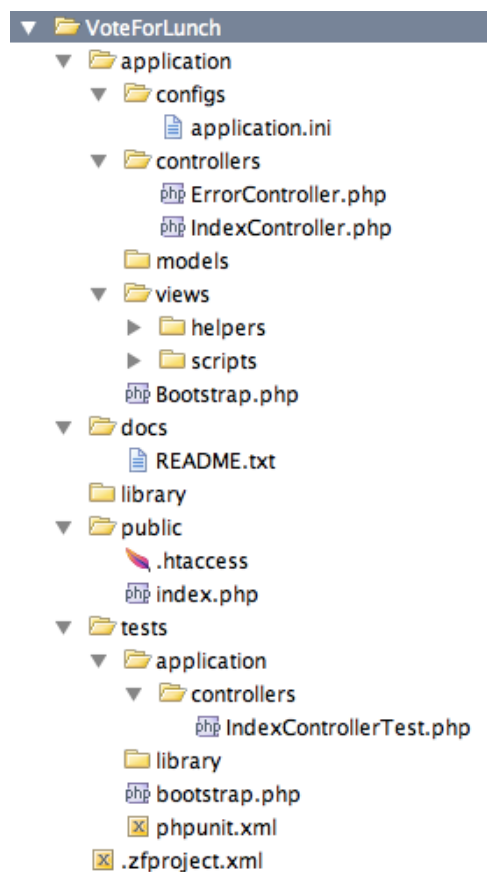
- **Modèle:** noyau de l'application qui gère les données, permet de récupérer les informations dans la base de données, de les organiser pour qu'elles puissent ensuite être traitées par le contrôleur.
- **Vue:** composant graphique de l'interface qui permet de présenter les données du modèle à l'utilisateur.
- **Contrôleur:** composant responsable des prises de décisions, gère la logique du code, il est l'intermédiaire entre le modèle et la vue.

## Schéma illustrant le design pattern MVC



Source: [developpez.com](http://developpez.com)

## Exemple d'une structure MVC



## 5. [Réalisations comportant les extraits de code les plus significatifs et en les argumentant, y compris pour la sécurité et le web mobile]

---

Nous avons fait le choix de réaliser le site en MVC, afin de structurer notre code pour plus de faciliter de lecture par des tiers, mais également par soucis de sécurité.

**Le MVC présente l'avantage de séparer les différentes parties du code. Ainsi, une page HTML visible sur les navigateurs, depuis l'ordinateur de l'utilisateur peut être analysée.**

Nos pages HTML, contiennent très peu, pour ne pas dire exceptionnellement du code PHP, et aucun code d'accès à la base de données.

Tout d'abord « .htaccess » est le 1<sup>er</sup> contrôle de sécurité côté serveur Apache, selon les règles que nous avons définies.

Ensuite, la page « index.php » est la seule page par laquelle on accède au site. Un routeur permet de la redirection vers le contrôleur général qui va jouer son rôle, à savoir (appeler le modèle général) et surtout le « bon » sous-contrôleur, qui lui-même permettra la liaison entre le bon modèle et les vues. Nous avons réécrit les « chemins réels et virtuels » des pages, afin d'afficher le nom des pages dans l'adresse « URL » et non un code « BARBARE » pour l'utilisateur, ce qui est une autre gage de sécurité.

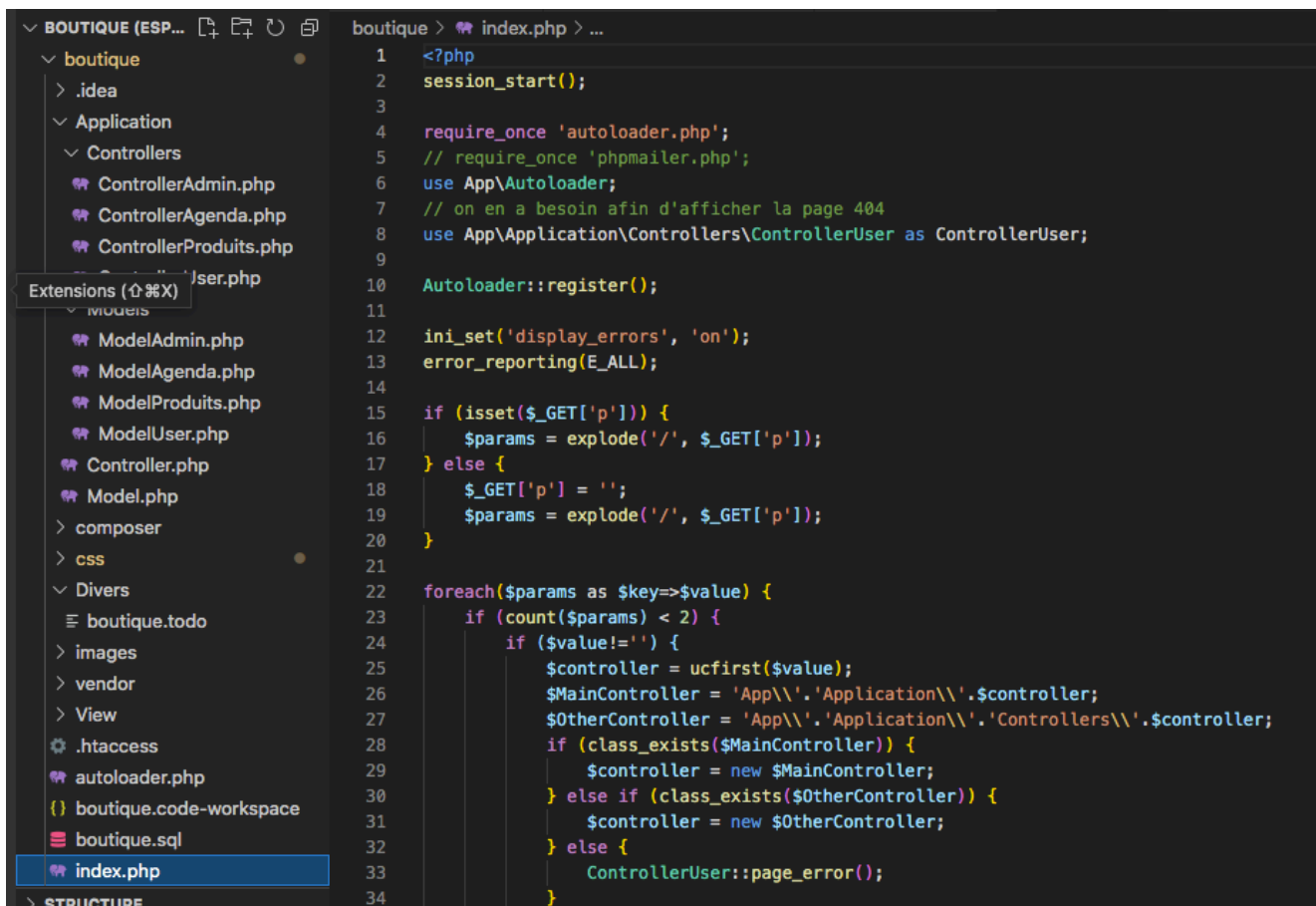
Le modèle et le contrôleur général, sont génériques, c'est à dire qu'ils contiennent des fonctions qui seront appelées par le bon modèle et le bon contrôleur et pilotent l'ensemble des éléments communs à chaque page (header, footer...)

Nous avons utilisés des « class abstract », pour la sécurité, et qui peuvent être utilisés dans les dossiers enfants.

**Au niveau du modèle, les attributs sont soit en « privé », soit en « protected ». Nous avons utilisés POO et les méthodes « prepare » et « execute », ainsi que « bind value » et « substr-replace » pour sécuriser.**

Chaque sous-contrôleur, sous modèle n'est autorisé à accéder qu'à ses propres vues. Ainsi il existe un contrôleur et modèle utilisateurs, différent de celui de l'administrateur, même si ce dernier hérite des règles de celui de l'utilisateur.

## Structure Visual Studio de notre MVC et code de la page index & routeur



### 1. Exposition des données sensibles

Lorsque vous surfez sur Internet, votre navigateur utilise le protocole http (Hypertext Transfer Protocol) pour afficher les pages web, et le protocole

Transmission Control Protocol/Internet Protocol (TCP/IP) pour les transmettre.

Si le serveur web établit la connexion TCP avec le navigateur, une réponse avec le code status et le fichier demandé (généralement le fichier index.html pour la page web) sera transmise. Mais dans notre cas, les données transitent en HTTP et pas en HTTPS...

Les données qui transitent en HTTP peuvent être interceptées, car elles circulent en clair.

Comment éviter d'exposer les données sensibles en transit ?

- Utilisez le HTTPS pour l'ensemble de votre site.
- **Utilisez les requêtes GET pour récupérer et POST pour modifier les informations.**
- Sécurisez vos cookies pour qu'ils soient transmis par l'en-tête et via HTTPS.
- Sécurisez vos sessions en ajoutant une date d'expiration, en sécurisant l'ID et en ne mettant pas cet ID dans l'URL.

Les données sensibles ne sont pas seulement en transit, elles sont aussi stockées en base de données. Pour protéger certaines données stockées sur une application, il est possible d'utiliser des algorithmes de hachage.

L'intérêt des algorithmes de hachage est qu'ils permettent de calculer une empreinte (ou hash) d'une chaîne de caractères, par exemple. Cette empreinte est utile pour éviter de stocker en clair le mot de passe dans la base de données.

Comment éviter d'exposer les données stockées ?

- Sécurisez votre base de données avec le chiffrement.
- **Utilisez des algorithmes de hachage sécurisés.**
- Le masquage des données peut être utilisé pour sécuriser les données sensibles d'une base.

### Autoloader (réécriture des adresses URL et chemins d'accès aux fichiers)

```
?php
namespace App;

class Autoloader
{
    public static function register()
    {
        spl_autoload_register([
            __CLASS__,
            'autoload'
        ]);
        // pour les controllers et leurs méthodes
        $host = $_SERVER['HTTP_HOST'];
        // pour les fichiers des images et le css
        // $root = $_SERVER['DOCUMENT_ROOT'];
        $uri = rtrim(dirname($_SERVER['PHP_SELF']), '/\\');
        define('PATH', $host.$uri);
        define('IMAGES', str_replace("boutique/index.php", "boutique/images/", $_SERVER['PHP_SELF']));
        define('CSS', str_replace("boutique/index.php", "boutique/css/", $_SERVER['PHP_SELF']));
        // on a viré les constantes définies pour les chemins url
    }

    static function autoload($class_name)
    {
        $class_name = str_replace(__NAMESPACE__ . '\\', '', $class_name);
        $class_name = str_replace('\\', '/', $class_name);
        $file = __DIR__ . '/' . $class_name . '.php';
        if(file_exists($file))
        {
            require_once __DIR__ . '/' . $class_name . '.php';
        }
    }
}
```

## 2. Injection SQL

Cette vulnérabilité permet à un attaquant d'injecter des données non maîtrisées qui seront exécutées par l'application et qui permettent d'effectuer des actions qui ne sont normalement pas autorisées.

Ce type d'attaque s'effectue généralement grâce aux champs présents dans les formulaires.

Dans le cas d'une attaque par injection SQL, au lieu de mettre un nom d'utilisateur et un mot de passe sur une page de connexion, un utilisateur malveillant entrera des données directement interprétées par le moteur SQL, ce qui lui permettra de modifier le comportement de votre application.

Comment s'en prémunir ?

- Validez les entrées

**Cela consiste à limiter ce que l'utilisateur peut mettre dans la zone de texte.** Cela n'empêchera pas l'injection, mais c'est une mesure que vous pouvez mettre en place pour limiter des attaques de base. En effet, les caractères spéciaux spécifiques à certains langages ne pourront pas être utilisés : exemple en vérifiant que le mail est un mail.

- Préparez les requêtes SQL

Ce sont des requêtes dans lesquelles les paramètres sont interprétés indépendamment de la requête elle-même. De cette manière, il est impossible d'effectuer des injections.

### Modèle général pour l'appel à la base de données

```
Namespace App\Application;

abstract class Model
{
    private $db_host = 'localhost';
    private $db_login = 'root';
    private $db_password = '';
    private $db_name = 'boutique';
    private $db_charset = 'utf8mb4';
    protected $_PDO;
    // private $whitelist;
    private $whitelist = array ('utilisateurs', 'reservation', 'produit', 'panier');

    public function __construct() {
        $this->$whitelist = array ('utilisateurs', 'reservation', 'produit', 'panier');
    }

    public function connect_db()
    {
        $dsn = "mysql:host=" . $this->db_host . ";dbname=" . $this->db_name . ";charset=" . $this->db_charset;
        $options = [
            \PDO::ATTR_ERRMODE => \PDO::ERRMODE_EXCEPTION,
            \PDO::ATTR_DEFAULT_FETCH_MODE => \PDO::FETCH_OBJ
        ];
        try {
            $this->_PDO = new \PDO($dsn, $this->db_login, $this->db_password, $options);
            return $this->_PDO;
        } catch (\PDOException $e) {
            throw new \PDOException($e->getMessage(), (int)$e->getCode());
        }
    }
}
```

## Contrôleur général permettant l'affichage à partir des sous contrôleurs

```
namespace App\Application;
use App\Application\Model as Model;
use App\Application\Models;
// use App\View\Accueil;
use App\Autoloader;
use ArrayObject;

abstract class Controller
{
    public $message = '';

    public function bonne_affichage($donnee)
    {
        echo("<pre>");
        print_r($donnee);
        echo("</pre>");
    }

    // les méthodes qui permettent de charger un model
    public function load_model($model)
    {
        // voir après si on peut se passer sans attribuer null par défaut
        if ($model!=null)
        {
            $modelPath = 'App\\'. 'Application\\'. 'Models\\'. $model;
            $this->$model = new $modelPath;
            return $this->$model;
        }
    }
}
```

## Exemple de la fonction connexion d'un utilisateur de puis la Class model User

```
public function get_one_user($login, $mail=null)
{
    $stmt = $this->connect_db()->prepare("SELECT id, id_droit, login, motpasse, prenom, nom, mail, te
    $stmt->execute([':login'=>$login, ':mail'=>$mail]);
    $user = $stmt->fetch(\PDO::FETCH_OBJ);
    return $user;
}
```



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

### 1. La base de données s'articule autour de 3 tables principales.

- La table utilisateurs qui va permettre d'identifier les utilisateurs et futurs clients. Cette table est liée à différentes tables.
- La table produit qui va contenir les informations sur le produit, dont le stock.
- Enfin, il y a la table commande qui permet de recueillir les informations relatives aux commandes passées. Cette dernière est reliée à la table utilisateur de manière à pouvoir identifier qui a passé commande.
- Une table de liaison : ligne de commande pour déterminer les différents produits d'une commande

La connexion à la base de données se fait depuis le modèle général en « class abstract », et la connexion, création... d'un utilisateur se fait depuis le contrôleur user après divers contrôles, j'appelle le modèle user, afin de faire appel à la base de données depuis la fonction « connect db » et à la vue pour permettre l'affichage adéquate.

### Contrôleur user avec la fonction « construct » pour ouvrir la session utilisateur

```
<?php
Namespace App\Application\Controllers;
Use App\Application\Controller;
Use App\Application\Models\ModelUser;
Use App\Application\Controllers\ControllerProduits;

class ControllerUser extends Controller
{
    private $id;
    private $id_droit;
    protected $prenom;
    protected $nom;
    protected $motpasse;
    protected $mail;
    protected $login;
    protected $telephone;
    protected $dateanniversaire;
    // il n'y a pas de colonne $adresse dans la table utilisateur;

    public function __construct()
    {
        $this->user = new ModelUser();
        $this->produits = new ControllerProduits();
        if (isset($_SESSION['user']))
        {
            $this->id = $_SESSION['user']->id;
            $this->login = $_SESSION['user']->login;
            $this->message = '';
        }
    }
}
```

Contrôleur admin, enfant de user, permettant l'accès à diverses informations,  
avec un contrôle d'accès

```
class ControllerAdmin extends ControllerUser
{
    private $id;
    private $id_droit;
    protected $categories;

    public function __construct()
    {
        $this->admin = new ModelUser();
        $this->adminProducts = new ModelAdmin();
        $this->droit = $this->adminProducts->get_all_droits();
        $this->categories = $this->selectCategories();
        $this->commandes = $this->adminCommandes();
        if(isset($_SESSION['user']->login) AND $_SESSION['user']->id_droit == 200 )
        {
            $this->id = $_SESSION['user']->id;
            $this->login = $_SESSION['user']->login;
            $this->id_droit = $_SESSION['user']->id_droit;
            $this->message = '';
            $this->stock = $this->sommeStock();
            $this->valeurStock = $this->valeurStock();
        }
    }
}
```

Création d'un nouveau utilisateur depuis le modèle user

```
public function new_user_data($data)
{
    $insert_values = [];
    // je crée un array avec les noms des colonnes
    $column_names = $this->columns_names('utilisateurs');
    // je parcours l'array associatif $data qui contient les données renseignés par ControllerUser
    foreach ($data as $key => $value)
    {
        // je parcours aussi les noms des colonnes
        for ($i=0;isset($column_names[$i]);$i++)
        {
            //afin de vérifier, pour chaque element de $data,
            //si la clé correspond au nom d'une colonne
            if ($column_names[$i]==$key)
            {
                //si oui, j'ajoute $value à l'array insert_values
                //pour l'envoyer à la méthode insert_values
                $insert_values[$key]=$value;
            }
        }
    }
    return $insert_values;
}

//j'insère les infos d'un nouveau utilisateur dans la bdd
public function insert_user($data)
{
    $data = $this->new_user_data($data);
    return $this->insert('utilisateurs', $data);
}

//je mets à jour les informations de l'utilisateur selon les données fournis par ControllerUser
public function update_user($data, $id)
{
    return $this->update('utilisateurs', $data, $id);
}
```

## 7. [Description de la veille, durant le projet, sur les vulnérabilités de sécurité]

---

### 1. Il existe deux types de vulnérabilités.

- Les vulnérabilités côté serveur.
- Les vulnérabilités côté site web (codage et vérification).

Au niveau des vulnérabilités côté serveur, le client achètera un hébergement chez Plesk, OVH, AWS... qui dispose d'une très bonne sécurité. L'hébergeur étant un hébergeur externe je ne peux pas intervenir sur la sécurité des serveurs web proposés par OVH.

En ce qui concerne les vulnérabilités côté site, dans un premier lieu, nous avons utilisé les variables « sessions pour sécuriser les données de l'utilisateur lorsqu'il est connecté. »

**Pour sécuriser nos différentes variables nous avons utilisé « str\_replace » pour supprimer certains caractères spéciaux mais aussi pour enlever les espaces, mais plus fréquemment « htmlspecialchars ».**

Nous avons également sécurisé l'inscription et la connexion en utilisant des **vérifications grâce à des « if », « else », « empty » ...**

J'envoie en premier mes variables en méthode « post » et non en méthode « get » pour plus de sécurité.

Je vérifie en premier que chaque variable est remplie, puis je vérifie que chaque variable soit correctement remplie.

Dans le nom et le prénom, il n'y a pas de caractères spéciaux, pour l'adresse mail, il y a des caractères, des chiffres, certains caractères puis un « @ » puis des caractères, des chiffres, certains caractères puis un « . » puis des caractères.

Exemple d'un contrôle que nous rajouterons :

```
<?php
$email_a = 'joe@example.com';
$email_b = 'bogus';

if (filter_var($email_a, FILTER_VALIDATE_EMAIL)) {
    echo "L'adresse email '$email_a' est considérée comme valide.";
}
if (filter_var($email_b, FILTER_VALIDATE_EMAIL)) {
    echo "L'adresse email '$email_b' est considérée comme valide.";
} else {
    echo "L'adresse email '$email_b' est considérée comme invalide.";
}
?>
```

Par la suite, je sécurise le mot de passe en vérifiant dans un premier temps que le mot de passe et le mot de passe de vérification sont les mêmes

**Puis je crypte le mot de passe en utilisant « password\_hash »** avant de l'enregistrer dans la base de données.

Pour réaliser le projet j'ai utilisé du HTML, CSS, PHP, JS, MySQL. Je sécurise les systèmes de connexion et d'inscription.

Dans un premier temps, je crypte le mot de passe lors de l'enregistrement, pour cela j'utilise la fonctionnalité « password\_hash » de PHP.

Dans un second temps, je compare le mot de passe crypté enregistré dans ma base de données avec le mot de passe tapée lors de **la connexion grâce à la fonctionnalité « password\_verify » de PHP, en demandant au nouvel utilisateur de se connecter.**

Je vérifie bien évidemment que le mail et le nom d'utilisateur ne soient pas déjà utilisés et enregistrés dans ma base de données avec une comparaison avec ce qui est marqué sur le formulaire d'inscription et les données récupérées dans ma base de données.

Pour la connexion, je vérifie que le login existe bien dans ma base de données et je vérifie ensuite le mot de passe, si les deux valeurs correspondent aux données enregistrées dans ma base de données, je connecte l'utilisateur et crée en même temps les valeurs \$\_SESSION dans session\_start(), qui permettent de garder l'utilisateur connecté sur toutes les pages du site web.

***Le tout étant piloté par notre MVC, qui fera appel au bon contrôleur, modèle et vue.***

## Fonction inscription depuis le contrôleur user avec divers contrôles et messages

```
// METHODES LIES AUX VIEWS
public function inscription()
{
    if (isset($_POST['login']))
    {
        if (!empty($_POST['login']) AND !empty($_POST['mail']) AND !empty($_POST['motpasse'])
        AND !empty($_POST['confirmer_motpasse'])
        AND !empty($_POST['prenom']) AND !empty($_POST['nom'])
        AND !empty($_POST['telephone']) AND !empty($_POST['dateanniversaire'])
        AND !empty($_POST['subscribeRGPD']))
        {
            $user = $this->user_exists($_POST['login'], $_POST['mail']);
            // $user génère un objet, si pas d'objet, renvoi d'un booléen : false
            if ($user==false)
            {
                $login = htmlspecialchars($_POST['login']);
                /*$password = sha1($_POST['motpasse']);
                $password2 = sha1($_POST['confirmer_motpasse']);*/
                $loginlength = strlen($login);

                if ($loginlength <= 55)
                {
                    $user = $this->user_exists($_POST['login'], $_POST['mail']);

                    if($user == 0)
                    {
                        if($_POST['motpasse'] ==$_POST['confirmer_motpasse'])
                        {
                            $password = password_hash($_POST['motpasse'], PASSWORD_DEFAULT);
                            $_POST['id_droit']=1;
                            $this->new_user($_POST);
                            $this->message = "Votre compte a bien été créé !";
                            $this->render('connexion', $this->message);
                        }
                    }
                    else
                    {
                        $this->message = 'Vos mots de passes ne correspondent pas';
                        return $this->render('inscription', $this->message);
                    }
                }
                else
                {
                    $this->message = 'Votre login ne doit pas dépasser 55 caractères';
                    return $this->render('inscription', $this->message);
                }
            }
            else
            {
                $this->message = 'Le mail existe déjà';
                return $this->render('connexion', $this->message);
            }
        }
        else
        {
            $this->message = 'Vous devez renseigner tous les champs et confirmer avoir lu les mentio
            return $this->render('inscription', $this->message);
        }
    }
    else
    {
        $this->message = '';
        return $this->render('inscription', $this->message);
    }
    $this->render('inscription');
}
```

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

Durant notre formation et la réalisation du projet, nous avons effectué de nombreuses recherches (Architecture MVC, installation de Composer, Stripe, PHP mailer...) et plus particulièrement, afin de mettre en œuvre, en pratique le langage Javascript, et entre autre pour le bouton recherche, afin de faciliter à l'utilisateur les recherches des produits en vente sur le site.

Ainsi, j'ai visionné plusieurs tutoriels, lu de la documentation w3schools et j'ai apprécié la qualité et la facilité des informations de MDN.

### *Tutoriels (vidéos Youtube Nouvelle Techno sur Javascript)*

The image shows a YouTube interface for a playlist named "Javascript" by the channel "Nouvelle Techno". The playlist has 42 videos and 4,630 views, with the last update 6 days ago. The main video player shows a thumbnail for "Tutoriel : La portée des variables en javascript". To the right, a list of videos is displayed, each with a number, a thumbnail, a title, and a duration. The videos are:

- 9. On joue au Morpion ??? Création d'un Tic Tac Toe en HTML, CSS et JS (29:49)
- 10. Introduction à ChartJS (19:56)
- 11. Le stockage local et de session en Javascript (15:11)
- 12. Utiliser les datasets en Javascript (13:22)
- 13. Créer un diaporama en HTML/CSS/JS (34:44)
- 14. Créer une « barre de progression » horizontale en haut de page (11:30)

# 9. [Extrait de site anglophone, utilise dans le cadre de la recherche décrite précédemment, accompagné de la traduction en français effectué part le candidat sans traducteur automatique]

## Documentation générale sur les langages de programmation et Mysql

**w3schools.com** THE WORLD'S LARGEST WEB DEVELOPER SITE

TUTORIALS • REFERENCES • EXAMPLES • EXERCISES • CERTIFICATES

**HTML and CSS**  
Learn HTML  
Learn CSS  
Learn Bootstrap  
Learn W3.CSS  
Learn Colors  
Learn Icons  
Learn Graphics  
Learn How To  
Learn Sass

**JavaScript**  
Learn JavaScript  
Learn jQuery  
Learn React  
Learn AngularJS  
Learn JSOJ  
Learn AJAX  
Learn W3.JS

**Server Side**  
Learn SQL  
Learn PHP  
Learn ASP  
Learn Node.js  
Learn Raspberry Pi

**Programming**  
Learn Python  
Learn Java  
Learn C++  
Learn C#  
Learn Machine Learning

**Web Building**  
Web Templates  
Web Statistics  
Web Certificates  
Web Editor  
Web Development  
Test Your Typing Speed

**XML Tutorials**  
Learn XML  
Learn XML AJAX  
Learn XML DOM  
Learn XML DTD  
Learn XML Schema  
Learn XSLT  
Learn XPath  
Learn XQuery

**References**  
HTML Reference  
CSS Reference  
JS Reference  
SQL Reference  
PHP Reference  
jQuery Reference  
Python Reference  
Bootstrap 3 Reference  
Bootstrap 4 Reference  
W3.CSS Reference  
Java Reference  
Angular Reference  
Sass Reference  
UTF-8 Reference

**Exercises**  
HTML Exercises  
CSS Exercises  
JavaScript Exercises  
SQL Exercises  
Python Exercises  
PHP Exercises  
jQuery Exercises  
Java Exercises  
C++ Exercises  
C# Exercises  
Bootstrap Exercises

### HTML

The language for building web pages

LEARN HTML HTML REFERENCE

**HTML Example:**

```
<!DOCTYPE html>
<html>
<title>HTML Tutorial</title>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Try It Yourself »

### CSS

The language for styling web pages

LEARN CSS CSS REFERENCE

**CSS Example:**

```
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
```

Try It Yourself »

### JavaScript

The language for programming web pages

LEARN JAVASCRIPT JAVASCRIPT REFERENCE

**JavaScript Example:**

```
<button onclick="myFunction()">Click Me</button>

<script>
function myFunction() {
  var x =
    document.getElementById("demo");
  x.style.fontSize = "20px";
  x.style.color = "red";
}
</script>
```

Try It Yourself »

### SQL

A language for accessing databases

LEARN SQL SQL REFERENCE

**SQL Example:**

```
SELECT * FROM Customers
WHERE Country="Mexico";
```

Try It Yourself »

### Python

A programming language

LEARN PYTHON

### PHP

A web server programming language

LEARN PHP

### jQuery

A JavaScript library for developing web pages

LEARN JQUERY

### Java

A programming language

LEARN JAVA

### W3.CSS

A modern CSS framework for faster and better responsive

### Color Picker

### Bootstrap

A CSS framework for designing better web pages

web pages

LEARN W3.CSS

LEARN BOOTSTRAP 3

LEARN BOOTSTRAP 4

### Exercises

Test yourself with exercises

HTML Exercises CSS Exercises JavaScript Exercises

SQL Python PHP jQuery Java C++ C#

### Web Templates

Browse our selection of free responsive HTML Templates

BROWSE TEMPLATES

### How To Section

Code snippets for HTML, CSS and JavaScript

**About Services Clients Contact**

**Full Sidebar**  
This sidebar is of full height (100%) and always shown.  
Scroll this window to see the "fixed" effect.  
Some text to enable scrolling. Lorem ipsum dolor sit amet, illum definitions no quo, malisset concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ut la, id agam omnis eventur eum. After laboramus reputandae nec et, forendent officium et la ut, Eum no.

**About Services Clients Contact**

**Auto Sidebar**  
This sidebar is as tall as its content (the links), and is always shown.  
Scroll this window to see the "fixed" effect.  
Some text to enable scrolling. Lorem ipsum dolor sit amet, illum definitions no quo, malisset concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ut la, id agam omnis eventur eum. After laboramus reputandae nec et, forendent officium et la ut, Eum no.

LEARN HOW TO

### Web Certificates

Get Certified in HTML, CSS, and JavaScript

Get Certified »

WEB CERTIFICATES QUIZZES EXERCISES FORUM | ABOUT

W3Schools is optimized for learning, testing, and training. Examples might be simplified to improve reading and basic understanding. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using this site, you agree to have read and accepted our terms of use, cookie and privacy policy.  
Copyright 1999-2020 by Refnes Data. All Rights Reserved.

👍

# Traduction en français

## HTML et CSS

Apprenez le HTML  
Apprenez le CSS  
Apprenez Bootstrap  
Apprenez W3.CSS  
Apprenez les couleurs  
Apprenez les icônes  
Apprenez les graphiques  
Apprenez à réagir  
Apprenez Sass

## JavaScript

Apprenez JavaScript  
Apprenez jQuery  
Apprenez à réagir  
Apprenez AngularJS  
Apprenez JSON  
Apprenez AJAX  
Apprenez W3.JS

## Du côté serveur

Apprenez SQL  
Apprenez PHP  
Apprenez l'ASP  
Apprenez Node.js  
Apprenez Raspberry Pi

## Programmation

Apprenez Python  
Apprenez Java  
Apprenez le C ++  
Apprenez le C #  
Apprenez la machine learning

## Construction Web

Modèles Web  
Statistiques Web  
Certificats Web  
Editeur Web  
Développement web  
Testez votre vitesse de frappe

## Tutoriels XML

Apprenez XML  
Apprenez XML AJAX  
Apprenez XML DOM  
Apprenez XML DTD  
Apprenez le schéma XML  
Apprenez XSLT  
Apprenez XPath  
Apprenez XQuery

## Références

Référence HTML  
Référence CSS  
Référence JS  
Référence SQL  
Référence PHP  
Référence jQuery  
Référence Python  
Référence Bootstrap 3  
Référence Bootstrap 4  
Référence W3.CSS  
Référence Java  
Référence angular  
Référence Sass  
Référence UTF-8

## Des exercices

Exercices HTML  
Exercices CSS  
Exercices JavaScript  
Exercices SQL  
Exercices PHP  
Exercices Python  
Exercices Java  
Exercices C ++  
Exercices C #  
Exercices Bootstrap

## HTML

Le langage pour créer des pages Web

APPRENDRE LE HTML RÉFÉRENCE HTML

### Exemple HTML:

```
<!DOCTYPE html>
<html>
<title>HTML Tutorial</title>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Essayez vous-même »

### Exemple CSS:

```
body {
background-color: lightblue;
}

h1 {
color: white;
text-align: center;
}

p {
font-family: verdana;
font-size: 20px;
}
```

Essayez vous-même »

## CSS

Le langage pour styliser les pages Web

APPRENDRE CSS RÉFÉRENCE CSS

## JavaScript

Le langage de programmation des pages Web

APPRENDRE JAVASCRIPT

RÉFÉRENCE JAVASCRIPT

### Exemple JavaScript:

```
<button onclick="myFunction()" *Click Me</button>

<script>
function myFunction() {
var x =
document.getElementById("demo");
x.style.fontSize = "20px";
x.style.color = "red";
}
</script>
```

Essayez vous-même »

### Exemple SQL:

```
SELECT * FROM Customers
WHERE Country='Mexico';
```

Essayez vous-même »

## SQL

Un langage pour accéder aux bases de données

APPRENDRE SQL RÉFÉRENCE SQL

### Python

Un langage de programmation

APPRENDRE PYTHON

### PHP

Un langage de programmation de serveur Web

APPRENDRE PHP

### jQuery

Une bibliothèque JavaScript pour développer des pages Web

APPRENDRE JQUERY

### Java

Un langage de programmation

APPRENDRE JAVA

### W3.CSS

Un framework CSS moderne pour des pages Web plus rapides et plus réactives

### Pipette à couleurs



### Amorcer

Un framework CSS pour concevoir de meilleures pages Web



## Des exercices

Testez-vous avec des exercices

Exercices HTML

Exercices CSS

Exercices JavaScript

SQL

Python

PHP

jQuery

Java

C ++

C #

## Modèles Web

Parcourez notre sélection de modèles HTML réactifs gratuits



PARCOURIR LES MODÈLES

## Section Comment

Extraits de code pour HTML, CSS et JavaScript

À propos  
Prestations de service  
Clients  
Contact

### Barre latérale complète

Cette barre latérale est de pleine hauteur (100%) et toujours affichée. Faites défiler cette fenêtre pour voir l'effet «float».

Du texte pour activer le défilement. Lorem ipsum dolor sit amet, illum definitiorem ne qui, malissae concludaturque et eum, altera fabulas ut quo. Aliqui causae gloritur sus la, id agam ornis evertitur eum. Aliet habebamus aperit fundibus non est, Societatem

À propos  
Prestations de service  
Clients  
Contact

### Barre latérale automatique

Cette barre latérale est aussi haute que son contenu (les liens) et est toujours affichée. Faites défiler cette fenêtre pour voir l'effet «float».

Du texte pour activer le défilement. Lorem ipsum dolor sit amet, illum definitiorem ne qui, malissae concludaturque et eum, altera fabulas

APPRENDRE À

## Certificats Web

Obtenez la certification HTML, CSS et JavaScript

Obtenir une certification \*

CERTIFICATS WEB

QUIZ

DES EXERCICES

FORUM | À PROPOS

W3Schools est optimisé pour l'apprentissage, les tests et la formation. Les exemples pourraient être simplifiés pour améliorer la lecture et la compréhension de base. Les distracteurs, références et exemples sont constamment examinés pour éliminer les erreurs, mais nous ne pouvons garantir l'exactitude totale de tout le contenu. En utilisant ce site, vous acceptez d'avoir lu et accepté nos conditions d'utilisation, cookies et politique de confidentialité.

Copyright 1999-2020 par Refnnes Data. Tous les droits sont réservés.





## Focus sur JavaScript

► Complete beginners

► JavaScript Guide

► Intermediate

► Advanced

**References:**

► Built-in objects

► Expressions & operators

► Statements & declarations

► Functions

► Classes

► Errors

► Misc

This section is dedicated to the JavaScript language itself, and not the parts that are specific to Web pages or other host environments. For information about [API](#) specifics to Web pages, please see [Web APIs](#) and [DOM](#).

The standards for JavaScript are the [ECMAScript Language Specification](#) <sup>↗</sup> (ECMA-262) and the [ECMAScript Internationalization API specification](#) <sup>↗</sup> (ECMA-402). The JavaScript documentation throughout MDN is based on the latest draft versions of ECMA-262 and ECMA-402. And in cases where some [proposals for new ECMAScript features](#) <sup>↗</sup> have already been implemented in browsers, documentation and examples in MDN articles may use some of those new features.

Do not confuse JavaScript with the [Java programming language](#) <sup>↗</sup>. Both "Java" and "JavaScript" are trademarks or registered trademarks of Oracle in the U.S. and other countries. However, the two programming languages have very different syntax, semantic, and use.

### Looking to become a front-end web developer?

We have put together a course that includes all the essential information you need to work towards your goal.

[Get started](#)

## Tutorials

► Intermédiaire

► Avancé

**Références :**

► Objets natifs

► Expressions & opérateurs

► Instructions & déclarations

► Fonctions

► Classes

► Errors

► Plus

JavaScript avec les [API](#) spécifiques des navigateurs web pour les pages web, veuillez consulter les sections sur les [API Web](#) (*Web API* en anglais) et le [DOM](#).

Le standard qui spécifie JavaScript est [ECMAScript](#). En 2012, tous les navigateurs modernes supportent complètement ECMAScript 5.1. Les anciens navigateurs supportent au minimum ECMAScript 3. Une sixième version majeure du standard a été finalisée et publiée le 17 juin 2015. Cette version s'intitule officiellement ECMAScript 2015 mais est encore fréquemment appelée ECMAScript 6 ou ES6. Étant donné que les standards ECMAScript sont désormais édités sur un rythme annuel, cette documentation fait référence à la dernière version en cours de rédaction, actuellement c'est [ECMAScript 2020](#) <sup>↗</sup>.

JavaScript ne doit pas être confondu avec le [langage de programmation Java](#) <sup>↗</sup>. Java et JavaScript sont deux marques déposées par Oracle dans de nombreux pays mais ces deux langages de programmation ont chacun une syntaxe, une sémantique et des usages différents.

### Vous cherchez à devenir un développeur web front-end ?

Nous avons élaboré un cours qui comprend toutes les informations essentielles dont vous avez besoin pour atteindre votre objectif.

[Commencer](#)

## Tutoriels

# Traduction français JavaScript

How to create a DOM tree

Introduction to events

How whitespace is handled by HTML, CSS, and in the DOM

Examples of web and XML development using the DOM

▼ Interfaces
AbortController
AbortSignal
AbstractRange
Attr
ByteString
CDATASection
CharacterData
ChildNode
CSSPrimitiveValue
CSSValue
CSSValueList

## Interfaces du DOM

- [Attr](#)
- [CharacterData](#)
- [ChildNode](#) 🔗
- [Comment](#)
- [CustomEvent](#)
- [Document](#)
- [DocumentFragment](#)
- [DocumentType](#)
- [DOMError](#) 🗑️
- [DOMException](#)
- [DOMImplementation](#)
- [DOMString](#)
- [DOMTimeStamp](#)
- [DOMSettableTokenList](#)
- [DOMStringList](#)
- [DOMTokenList](#)
- [Element](#)
- [EventTarget](#)
- [HTMLCollection](#)
- [MutationObserver](#)

- [MutationRecord](#) (en-US)
- [NamedNodeMap](#)
- [Node](#)
- [NodeFilter](#)
- [NodeIterator](#)
- [NodeList](#)
- [NonDocumentTypeChildNode](#) (en-US)
- [ParentNode](#)
- [ProcessingInstruction](#)
- [Selection](#) 🔗
- [Range](#)
- [Text](#)
- [TextDecoder](#) (en-US) 🔗
- [TextEncoder](#) 🔗
- [TimeRanges](#)
- [TreeWalker](#)
- [URL](#)
- [Window](#)
- [Worker](#)
- [XMLDocument](#) 🔗

How to create a DOM tree

Introduction to events

How whitespace is handled by HTML, CSS, and in the DOM

Examples of web and XML development using the DOM

▼ Interfaces
AbortController
AbortSignal
AbstractRange
Attr
ByteString
CDATASection
CharacterData
ChildNode
CSSPrimitiveValue
CSSValue

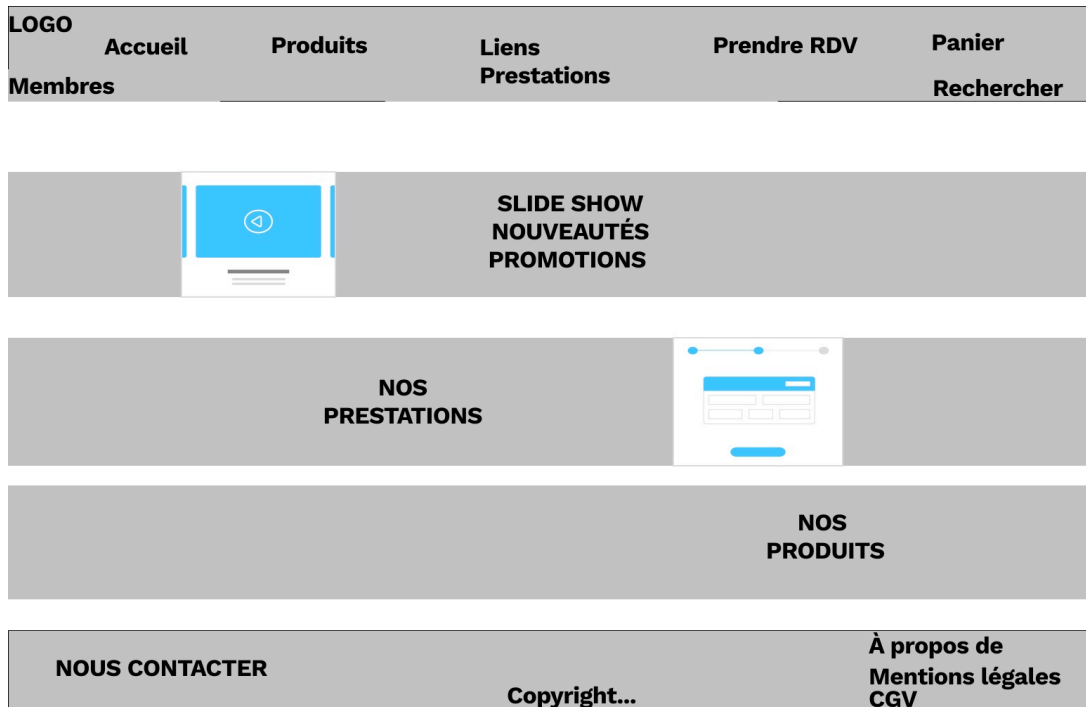
## Interfaces du DOM

- [Attr](#)
- [CharacterData](#)
- [ChildNode](#) 🔗
- [Comment](#)
- [CustomEvent](#)
- [Document](#)
- [DocumentFragment](#)
- [DocumentType](#)
- [DOMError](#) 🗑️
- [DOMException](#)
- [DOMImplementation](#)
- [DOMString](#)
- [DOMTimeStamp](#)
- [DOMSettableTokenList](#)
- [DOMStringList](#)
- [DOMTokenList](#)
- [Element](#)
- [EventTarget](#)
- [HTMLCollection](#)
- [MutationObserver](#)

- [MutationRecord](#) (en-US)
- [NamedNodeMap](#)
- [Node](#)
- [NodeFilter](#)
- [NodeIterator](#)
- [NodeList](#)
- [NonDocumentTypeChildNode](#) (en-US)
- [ParentNode](#)
- [ProcessingInstruction](#)
- [Selection](#) 🔗
- [Range](#)
- [Text](#)
- [TextDecoder](#) (en-US) 🔗
- [TextEncoder](#) 🔗
- [TimeRanges](#)
- [TreeWalker](#)
- [URL](#)
- [Window](#)
- [Worker](#)
- [XMLDocument](#) 🔗

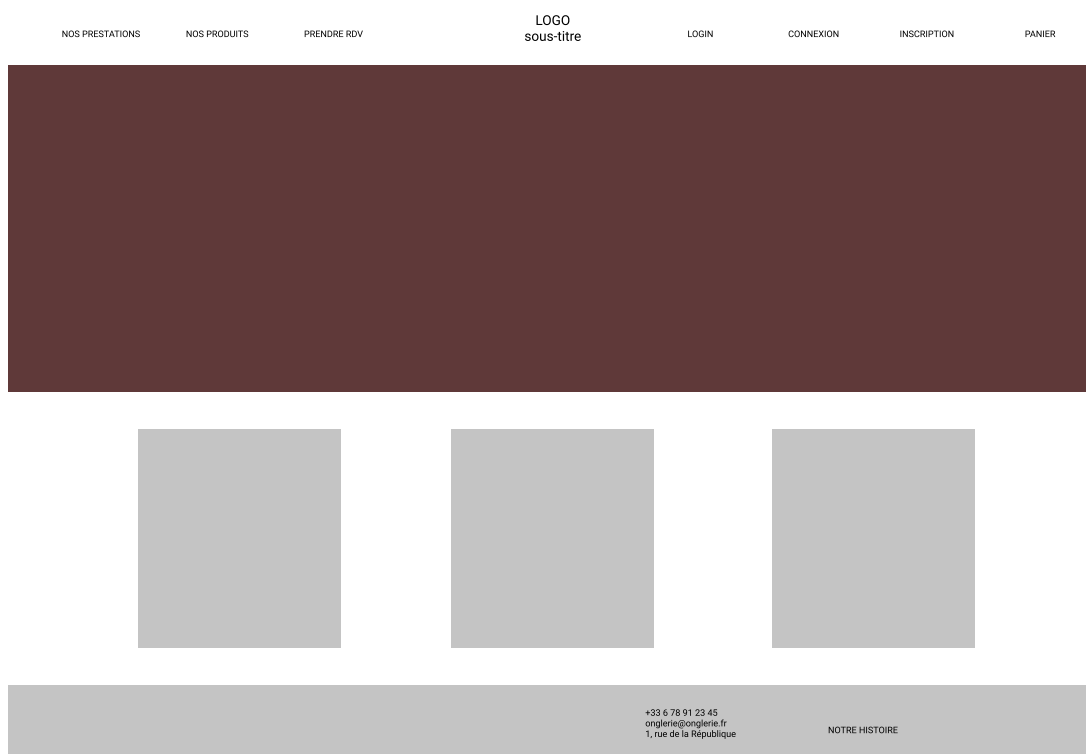
## 10. [Annexes : zoning, maquettage, schéma base de données]

### 1. Zoning page accueil








### 2. Maquettage des principales pages

#### *Page accueil*



**Page produit**

rechercher un produit

	<p>Nom du produit</p> <p>Description du produit: lorem ipsum lorem ipsum lorem ipsum lorem ipsum</p>	<p>Prix en euros</p>	<p>ajouter au panier</p>	<p>commander</p>
	<p>Nom du produit</p> <p>Description du produit: lorem ipsum lorem ipsum lorem ipsum lorem ipsum</p>	<p>Prix en euros</p>	<p>ajouter au panier</p>	<p>commander</p>
	<p>Nom du produit</p> <p>Description du produit: lorem ipsum lorem ipsum lorem ipsum lorem ipsum</p>	<p>Prix en euros</p>	<p>ajouter au panier</p>	<p>commander</p>
	<p>Nom du produit</p> <p>Description du produit: lorem ipsum lorem ipsum lorem ipsum lorem ipsum</p>	<p>Prix en euros</p>	<p>ajouter au panier</p>	<p>commander</p>
	<p>Nom du produit</p> <p>Description du produit: lorem ipsum lorem ipsum lorem ipsum lorem ipsum</p>	<p>Prix en euros</p>	<p>ajouter au panier</p>	<p>commander</p>


**Page administrateur**

The diagram illustrates the 3M model structure, which is composed of three main components arranged horizontally:

- Somme commande**: This component is further divided into two sub-components:
  - Mois précédent** (Previous month)
  - Mois en cours** (Current month)
- Valeur du stock** (Stock value)
- Quantité totale du stock** (Total stock quantity)

Liste des produits disponibles

lorem ipsum lorem ipsum lorem ipsum  
lorem ipsum lorem ipsum lorem ipsum  
lorem ipsum lorem ipsum lorem ipsum  
lorem ipsum lorem ipsum lorem ipsum  
lorem ipsum lorem ipsum lorem ipsum  
lorem ipsum lorem ipsum lorem ipsum  
lorem ipsum lorem ipsum lorem ipsum  
lorem ipsum lorem ipsum lorem ipsum  
lorem ipsum lorem ipsum lorem ipsum

Three yellow rectangular placeholders for product images, arranged horizontally at the bottom of the page.

[illegible]

# MPD

