

# DOSSIER PROJET

< Titre développeur web et web mobile />



Stage réalisé du 10/01/2022 au 04/03/2022

Au sein de l'Artisan Média

Avec Onlineformapro

Formation Access Code School

# SOMMAIRE

<b>0 . Remerciement</b>	<b>p3</b>
<b>1 . Liste des compétences du référentiel qui sont couvertes par le projet</b>	<b>p4</b>
<b>Front-end:</b>	<b>p4</b>
A. Réaliser une interface utilisateur web statique et adaptable	
B. Développer une interface utilisateur web dynamique	
<b>Back-end:</b>	<b>p6</b>
A. Créer une base de données	
B. Développer les composants d'accès au données	
C. Développer la partie back-end d'une application web ou web mobile	
<b>2 . Résumé du projet</b>	<b>p7</b>
A. Présentation de l'entreprise	
B. Le projet	
<b>3 . Cahier des charges, expression des besoins, ou spécifications fonctionnelles du projet</b>	<b>p8</b>
<b>4 . Spécification techniques du projet élaborées par le candidat, y compris pour la sécurité et le web mobile</b>	<b>p9</b>
<b>5 . Réalisations du candidat comportant les extraits de code les plus significatifs et en les argumentant, y compris pour la sécurité et le web mobile</b>	<b>p11</b>
<b>6 . Présentation du jeu d'essai élaboré de la fonctionnalité</b>	<b>p28</b>
<b>7 . Description de la veille effectuée par le candidat durant le projet, sur les vulnérabilités de sécurité</b>	<b>p30</b>
<b>8 . Description d'une situation de travail ayant nécessité une recherche effectué par le candidat durant le projet, a partir d'un site anglophone</b>	<b>p32</b>
<b>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</b>	<b>p33</b>
<b>10 . Annexe</b>	<b>p35</b>

# Remerciement

Je remercie tout d'abord mon maître de stage Monsieur Servolle Xavier de m'avoir fait confiance au sein de son entreprise L'artisan Média pour le développement de son application qui m'ont permis d'apprendre beaucoup de choses et d'acquérir de nouvelles compétences au sein de ces 2 mois de stage.

Je tiens aussi à remercier mon formateur Alain Merucci qui m'a poussé à réussir tous ces projets.

# 1. Liste des compétences du référentiel qui sont couvertes par le projets :

## Front-end :

### A . Réaliser une interface web statique et adaptable

Qu'est-ce qu'une interface web statique et adaptable ?

*“Une interface web statique et adaptable est une page web dont le contenu ne change pas d'utilisateur à utilisateur c'est à dire que chaque personne auront les mêmes données sur la page web”*

L'application possède une page web statique et adaptable. La page “index.php” récupère les contenus de la base de données avec une boucle “while”, tant qu'il y aura du contenu dans la base de données la page les affichent dans une card de la librairie Bootstrap. Peu importe l'utilisateur qui se rendra sur la page, le contenu chargé sera toujours le même et dépendra des ajouts ou des suppressions du contenu dans la base de données.

J'ai réalisé l'interface à l'aide de la librairie Bootstrap et le langage de programmation php.

**Front-end :**

**B . Développer une interface utilisateur web dynamique**

Qu'est ce qu'une interface utilisateur web dynamique ?

*“Une interface web dynamique est une page web qui est générée à la demande de l'utilisateur son contenu peut varier en fonction du contenu demander par l'utilisateur. Les données ne s'afficheront qu'une fois la page web chargée”*

Les autres pages présente de l'application sont des pages web dynamiques et s'adaptent au contenu de la base de données.

Lorsque l'on décide d'éditer une image ce sont les données et l'image sélectionnées qui vont s'afficher sur la page web, c'est information varie de l'image sélectionner dans la page précédente.

## **Back-end :**

### **A . Créer une base de données**

Qu'est ce qu'une base de données ?

*“Une base de données est un ensemble d'informations qui est organisé de manière à être facilement accessible, géré et mis à jour. Elle est utilisée par les organisations comme méthode de stockage, de gestion et de récupération de l'information.”*

L'artisan Média tourne sur un serveur interne, mais n'ayant pas travaillé sur ce serveur, j'ai donc créé une base de données MySQL indépendante à l'aide de phpMyAdmin pour le développement de l'application.

Je n'ai eu besoin que d'une seule table dans ma base de données, celle-ci stocke les images enregistrées par l'utilisateur.

### **B . Développer les composants d'accès au données**

Pour l'accès au données j'ai réalisé une connexion à la base de données avec le système de connexion PDO. J'ai donc intégré plusieurs requêtes me permettant ainsi de réaliser les actions pour communiquer avec le contenu de la base de données.

### **C . Développer la partie back-end d'une application web ou web mobile**

Pour la partie back de l'application, j'ai développé le formulaire d'enregistrement d'image, ce formulaire va servir à stocker les images sélectionnées par l'utilisateur dans un dossier. Le formulaire intercepte les données de l'image, va lui fournir un nom unique, qui n'empêche pas l'upload de la même image plusieurs fois, déplace l'image dans le dossier upload et enregistre son nom dans la base de données.

Pour la modification de l'image j'ai développé le système qui récupère l'image modifiée envoyer via une requête ajax et encodé en base64, mon fichier upload.php récupère l'image encodé avec la méthode POST, décode l'image codé en base64, lui donne un nom unique pour ne pas avoir de doublons dans les images, et ensuite déplace l'image dans le dossier upload. J'ai ensuite une requête UPDATE qui va récupérer cette image pour la modifier par l'image avec l'id sélectionné.

## 2. Résumé du projet

### Résumé du projet en Français :

#### A . Présentation de l'entreprise :

J'ai réalisé mon stage d'une durée de 2 mois chez l'Artisans Média situé à Poligny afin de valider mon titre professionnel de Développeur Web et Web Mobile. L'entreprise est une agence de communication et de studio photo et vidéo. Il propose la communication grâce à leur contenus uniques et une vision centrée sur les clients de Poligny, Arbois, Salins-les-Bains, Dole et Lons-le-Saunier. Leur entreprise regroupe la création de sites internet basés sur les CMS Wordpress et Prestashop, la création de photos, la production vidéo et le graphisme.

#### B . Le projet :

L'entreprise étant spécialisée dans la photographie, mon projet était de réaliser un module d'édition d'image interne à l'entreprise qui aura pour but de pouvoir redimensionner et ajuster les images stockées sur le serveur de l'entreprise. J'ai néanmoins travaillé sur une base de données externes pour le développement de l'application.

Le maquettage du module a été réalisé à l'aide de Figma et toutes les icônes du module ont été désignées par la graphiste de l'entreprise.

La partie front-end s'est faite sur la base des librairies CropperJs et CamanJS me permettant ainsi d'avoir une interface dynamique et ergonomique agréable pour l'utilisateur.

CropperJS est une librairie Javascript Open Source qui permet d'ajouter un cropper sur nos images pour ensuite les rogner et CamanJS qui est également une librairie Javascript permettant d'ajouter des filtres tels que la luminosité, le contraste... afin de pouvoir ajuster l'image sélectionner.

La partie back-end du projet quant à elle s'est faite sur la base de PhpMyAdmin pour la gestion des images dans la base de données.

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

#### A . Contexte :

Le projet qui m'a été confié a pour but de répondre au besoin de l'entreprise. L'artisan Média étant spécialisé dans la photographie, mon projet était de développer un module d'édition d'image qui permet de corriger les images présentes dans le serveur.

#### B . Délais :

Le délai pour la réalisation de ce projet a été défini sur la période de stage, du 10 Janvier 2022 au 04 Mars 2022.

#### C . Environnement de travail :

L'environnement de travail était particulier, le développeur interne de l'entreprise disposait de très peu de disponibilité, j'ai eu une grande difficulté à obtenir les informations nécessaires pour la réalisation du projet.

Mon stage s'est réalisé en présentiel 2 jours sur 3 avec une petite réunion chaque jour que j'étais présent dans l'entreprise.

Le manque de communication et le peu d'aide que j'ai pu avoir était une difficulté au début mais m'a permis de chercher par moi même toutes les solutions à mes problèmes et de bien organiser mes recherches.



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

L'entreprise m'avait proposé de réaliser le projet à l'aide de la librairie PHP Imagick, mais après certains tests et de nombreuses recherches Imagick ne nous a pas permis de rendre le projet ergonomique. J'ai donc proposé à mon maître de stage de partir sur les librairies Javascript CropperJs et CamanJs afin d'obtenir l'édition d'image en temps réel et proposé à l'utilisateur une interface agréable à l'utilisation.

Ne connaissant pas les librairies CropperJs et CamanJs j'ai dû dans un premier temps les découvrir et me former à les utiliser.

### **Présentation de CropperJS :**

CropperJs est une librairie Javascript, l'un des meilleurs script JS de recadrage d'image qui fonctionne entièrement sur JQuery, mais il existe également une version vanilla. Les deux versions sont identiques mis à part la syntaxe pour leur utilisation. Le plugin prend en charge les aperçus en direct et les proportions personnalisées. La librairie possède également de nombreux outils permettant de faire pivoter l'image, recadrer automatiquement certains rapport...

### **Présentation de CamanJS :**

CamanJS est une librairie de manipulation de canvas en Javascript. C'est une combinaison d'une interface à utiliser avec des techniques d'édition d'image avancée et efficace. Il est complètement indépendant et fonctionne à la fois dans NodeJS et dans le navigateur.

N'ayant pas travaillé sur le serveur interne de l'entreprise, j'ai donc créé une page web qui représente la galerie d'image de l'entreprise. Pour la mettre en place j'ai décidé d'utiliser Bootstrap pour avoir une interface dynamique et qui s'adapte à tous les écrans. J'ai utilisé la version Bootstrap 5.

J'ai pu ainsi démarrer mon projet en utilisant les technologies suivante :

### **Front-end :**

- Javascript
- JQuery
- Bootstrap
- CropperJS
- CamanJS

**Back-end :**

- Php
- MySQL
- PhpMyAdmin

**Sécurité :**

J'ai utilisé l'extension PHP Data Object (PDO) pour la communication avec la base de données afin d'éviter toutes les injections SQL. Cependant je n'ai pas touché au serveur de l'entreprise, je n'ai donc pas pu gérer la sécurité sur le serveur de l'entreprise.

Pour me protéger des failles XSS (Cros Site Scripting) j'ai utilisé des htmlentities sur toutes les données d'entrée qui va convertir tous les caractères éligibles en entités HTML

**Documentation du code :**

J'ai pris le temps de commenter tout mon code afin que le développeur interne puisse intégrer correctement l'application au site web de l'entreprise.

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

Je vais dès à présent vous présenter toutes les réalisations effectuées pendant le développement de l'application au cours de ce stage tout en les argumentant et en vous présentant des exemples.

### 1. Création de la base de données

Pour commencer le projet j'ai décidé de créer une base de données sous PhpMyAdmin où seront stockées les images que je vais enregistrer à l'aide d'un formulaire car je travaille indépendamment du serveur de l'entreprise.

## Bases de données

### Création d'une base de données ?

Créer

J'ai ensuite créer une table file qui va comporter 2 champs

- id (AUTO-INCREMENT)
- name (VARCHAR : 255)

Cette table regroupe toutes les images stockées dans la base de données.

Structure de table

Vue relationnelle

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 id	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier  Supprimer  Plus
<input type="checkbox"/>	2 name	varchar(255)	utf8_general_ci		Non	Aucun(e)			Modifier  Supprimer  Plus

Par la suite j'ai crée le dossier où j'allais stocké tous les fichiers de l'application, j'ai d'abord commencé par créer un nouveau fichier (bdd.php) qui inclut la connexion à la base de données pour me permettre de communiquer avec le contenu de la base de données.

### bdd.php

Dans ce fichier j'ai commencé à écrire le code permettant la connexion a la base de données à l'aide de la méthode PDO.

```
<?php

try{
    $db = new PDO('mysql:host=localhost;dbname=upload_file', 'root', 'root');
}catch(PDOException $e) {
    die('Erreur de connexion : '.$e->getMessage());
}
```

J'ai rentré toutes les informations nécessaires pour la connexion (host, nom de la base de données, identifiant, mot de passe).

La connexion étant effectuée j'ai ensuite créé le fichier "index.php", cette page aura un formulaire qui nous permettra d'enregistrer les images choisies et une galerie qui affichera toutes les images présentes dans la base de données.

### index.php

On inclut ensuite la connexion a la base de données pour pouvoir passer nos requêtes.

J'ai aussi ajouté la fonction "error\_reporting(E\_ALL)" qui nous renvoie sur la page le niveau de rapport d'erreurs PHP.

```
require ('bdd.php');

error_reporting(E_ALL);
ini_set("display_errors", 1);
```

Maintenant que nous avons créé le fichier "index.php" et inséré la base de données au fichier j'ai pu créer le formulaire permettant d'insérer les images dans la base de données.

```

51 <section class="upload">
52   <div class="container col-3 d-flex justify-content-center">
53     <form action="index.php" method="post" enctype="multipart/form-data">
54       <div class="form-group col-12">
55         <p class="text-center">Sélectionner un fichier</p>
56         <input type="file" class="form-control" id="customFile" name="file">
57       </div>
58       <div class="d-flex justify-content-center">
59         <button class="enregistrer m-3" type="submit">Enregistrer</button>
60       </div>
61     </form>
62   </div>
63 </section>

```

Le formulaire utilise la méthode POST pour envoyer les données. J'ajouterai la balise form 'enctype="multipart/form-data"' car le formulaire comprend un input de type file.

J'ai décidé d'envoyer les données du formulaire sur cette même page "action="index.php"" même si l'on pourrait les renvoyer sur n'importe quelle autre page.

## Sélectionner un fichier

Choisir un fichier

Aucun fichier choisi

Enregistrer

Pour ensuite récupérer les informations du formulaire j'utilise une instruction if pour vérifier que le formulaire nous renvoi bien une valeur.

Dans ce cas, php nous indique qu'il reconnaît bien un index (ici "file").

```

8  ✓ if(isset($_FILES['file'])){
9      $tmpName = $_FILES['file']['tmp_name'];
10     $name = $_FILES['file']['name'];
11     $size = $_FILES['file']['size'];
12     $error = $_FILES['file']['error'];
13
14     $tabExtension = explode('.', $name);
15     $extension = strtolower(end($tabExtension));
16
17     $extensions = ['jpg', 'png', 'jpeg', 'gif'];
18     $maxSize = 4000000;
19
20  ✓ if(in_array($extension, $extensions) && $size <= $maxSize && $error == 0){
21
22         $uniqueName = uniqid('', true);
23
24         $file = $uniqueName.".".$extension;
25
26         move_uploaded_file($tmpName, './upload/'.$file);
27
28         $req = $db->prepare('INSERT INTO file (name) VALUES (?)');
29         $req->execute([$file]);
30
31  ✓ }else{
32         echo "Une erreur est survenue";
33     }
34 }

```

On commence par stocker toutes les valeurs qui nous intéressent dans des variables pour qu'on puisse ensuite les utiliser.

Dans un premier temps j'ai fait une vérification sur l'extension du fichier. J'ai donc récupéré l'extension à partir du nom du fichier à l'aide de la fonction `explode` qui nous permet de découper une chaîne de caractères en plusieurs morceaux. Ce qui nous donne par la suite un tableau ["image", "extension"]. Pour récupérer dès à présent l'extension nous allons récupérer le dernier élément du tableau en utilisant la fonction `end()` que j'ai décidé de mettre en minuscule à l'aide de `strtolower` pour ne pas avoir de problème de comparaison par la suite.

Pour la vérification de l'extension, j'ai créé un tableau avec les extensions qui sont acceptées.

```

17     $extensions = ['jpg', 'png', 'jpeg', 'gif'];

```

Pour vérifier si l'extension est bien présente dans ce tableau on utilise la fonction `in_array()`. Si l'extension n'est pas présente alors cela nous renvoie une erreur.

L'autre vérification effectuée est la taille du fichier reçu. L'entreprise m'avait donné l'information que leur image sortant de l'appareil avec une taille d'environ 35 mb, j'ai

donc mis que la taille maximale du fichier a ne pas dépasser est 40mb.

```
18      $maxSize = 4000000;
```

Si l'on ajoute deux fois la même image, la première sera écrasée par la deuxième image. J'ai donc donné un nom unique à chaque image enregistrée à l'aide de la fonction `uniqid()`, la fonction attend 2 paramètres, une chaîne de caractère et un booléen.

Après tout ça je peux enfin déplacer les images dans un dossier que j'ai créé "upload.php".

```
> upload
```

Pour envoyer les images dans mon dossier, j'ai utilisé la fonction php "move\_uploaded\_file". Cette fonction attend 2 paramètres, (le chemin du fichier que l'on veut uploader, et le chemin vers lequel on souhaite l'uploader.)

On récupère donc nos variables, ce qui va nous donner ceci.

```
26      move_uploaded_file($tmpName, './upload/'.$file);
```

Après tout ça nous avons bien récupéré toutes nos données qui sont enregistrées dans le fichier "upload" et nous pouvons les envoyer dans la base de données à l'aide d'une requête SQL.

Pour cette requête j'utilise un 'INSERT' pour insérer les données dans la base de données.

```
28      $req = $db->prepare('INSERT INTO file (name) VALUES (?)');  
29      $req->execute([$file]);
```

Une fois la requête insérée dans le code les images sont bien enregistrées dans la base de données et nous pouvons maintenant les récupérer pour les afficher dans notre page web pour ainsi créer la galerie d'image.

					id	name
<input type="checkbox"/>	Éditer	Copier	Supprimer		33	6218d2fc6c8328.84583644.jpg
<input type="checkbox"/>	Éditer	Copier	Supprimer		34	6218d3081e6029.80632437.jpg
<input type="checkbox"/>	Éditer	Copier	Supprimer		35	6218d301947c14.77153666.jpg
<input type="checkbox"/>	Éditer	Copier	Supprimer		36	6218d30db1a6f6.76173572.jpg
<input type="checkbox"/>	Éditer	Copier	Supprimer		37	6218d31361ca37.77863325.jpg
<input type="checkbox"/>	Éditer	Copier	Supprimer		38	6218d31d5b9471.63139818.jpg
<input type="checkbox"/>	Éditer	Copier	Supprimer		39	6218d3363c4722.36357931.jpg
<input type="checkbox"/>	Éditer	Copier	Supprimer		41	6220c7b6186880.44505127.jpg



Pour afficher les images sur la page web nous allons également utiliser une requête SQL "SELECT", ici je n'ai pas utilisé de requêtes préparées car il n'y a aucun paramètre à passer.

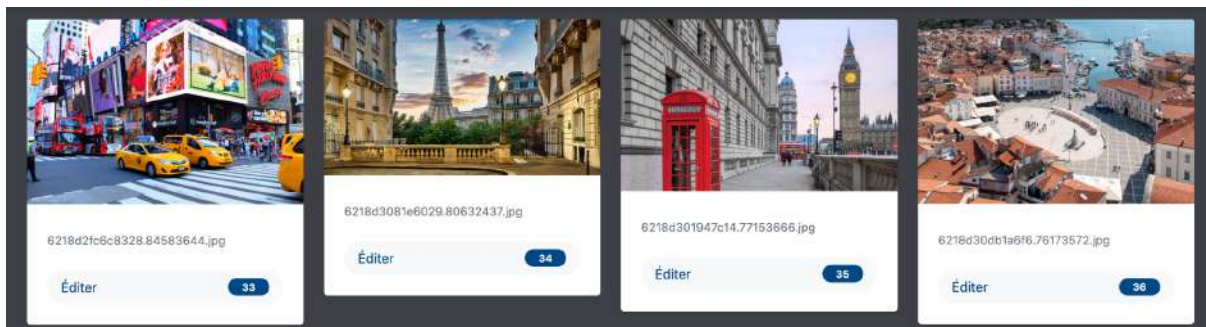
Je vais mettre une boucle(WHILE) sur le résultat pour que tant qu'il y a des résultats on continue de les afficher. Pour accéder aux résultats de la requête j'utilise la méthode (fetch) qui est une méthode PDO.

```
68 $req = $db->query('SELECT * FROM file ');
69 while($data = $req->fetch(PDO::FETCH_ASSOC)){
70     ?>
```

Après avoir testé avec un var\_dump si les données s'affichent bien à l'écran j'ai ensuite donné un peu de forme à la galerie en utilisant bootstrap pour afficher les données proprement à l'écran.

```
72 <!-- Gallery item -->
73 <div class="col-xl-3 col-lg-4 col-md-6 mb-4">
74     <div class="bg-white rounded shadow-sm"><img data-toggle="modal"
75         data-target="#exampleModal<?php echo $data['id'] ?>" src="./upload/<?= $data['name'] ?>" alt=""
76         class="img-fluid card-img-top">
77     <div class="p-4">
78         <h5> <a href="#" class="text-dark"></a></h5>
79         <p class="small text-muted mb-0"><?= $data['name'] ?></p>
80         <div class="d-flex align-items-center justify-content-between rounded-pill bg-light px-3 py-2 mt-4">
81             <a href='image.php?id=<?= $data['id'] ?>'>Éditer</a>
82             <div class="badge badge-primary px-3 rounded-pill font-weight-normal"><?= $data['id'] ?></div>
83         </div>
84     </div>
85 </div>
86 </div>
87 <!-- End -->
```

Ce qui nous a donné un résultat propre et dynamique qui s'adapte à tout type d'écran.



Maintenant que la mise en page de la galerie d'image est réalisée je poursuis avec la page où il y aura le module d'édition. Pour ça je commence par créer un nouveau fichier "php" que je nomme (image.php). Cette page sera une page web dynamique qui ne sera générée que par la demande de l'utilisateur, elle n'aura que les données de l'image que l'on veut modifier.



Pour récupérer les données que l'utilisateur demande nous allons utiliser la variable `$_GET`, en le faisant passer par lien dans le fichier "index.php".

La variable `$_GET` est celle qui nous donne les valeurs des informations indiquées dans l'url.

Pour ce faire je crée la balise "link" qui aura (`href=" image.php?id=<?= $data['id'] ?>"`)

Le lien nous renverra les données de l'image ou l'id est le même que celui présent dans la base données.

```
81 <a href='image.php?id=<?= $data['id'] ?>'>Éditer</a>
```

Ensuite dans mon fichier "image.php" j'inclus comme précédemment le fichier "bdd.php" pour que l'on puisse communiquer avec la base de données et j'ajoute également les fonctions permettant d'afficher les erreurs php.

Avec la condition "if(isset())" je vérifie que la variable `$_GET` est bien passé et ne me renvoie pas (null) sinon une erreur s'affiche en m'indiquant qu'il n'y a aucune image. Si la variable est bien passé je vais récupérer mes données à l'aide d'une requête SQL "SELECT" en précisant que l'id récupéré est celui qui est passé dans l'url grâce à la variable `$_GET`.

Je défini à ma requête la fonction `setAttribute()` ayant comme paramètre "`PDO::ATTR_ERRMODE`" et "`PDO::ERRMODE_EXCEPTION`" afin de gérer les différentes erreurs possible.

```
1 <?php
2 require ('bdd.php');
3
4 error_reporting(E_ALL);
5 ini_set("display_errors", 1);
6
7 if(isset($_GET['id'])){
8     $req = $db->query("SELECT name FROM file WHERE id = ".$_GET['id']);
9     $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

Je rajoute une boucle(WHILE) avec la méthode `fetch` pour récupérer les résultats et pour que le contenu s'affiche sur la page web.

```
11 while($data = $req->fetch(PDO::FETCH_ASSOC)){
12     $image = "./upload/" . $data['name'];
```

Un fois tout ça effectué j'ai bien mes données qui s'affiche je peux maintenant commencer à développer le module. Les deux librairies que j'ai utilisées sont des librairies Javascript, j'ai donc créé un fichier "script.js" et ajouté les script tag de CropperJS et CamanJS.

```
177 <script src="https://cdnjs.cloudflare.com/ajax/libs/cropperjs/1.5.12/cropper.js"
178     integrity="sha512-ZK6m9vADamS15fxBPtXw6ho6A4TuX89HUbCfvxa2v2NYNT/7l8yFGJ3JlXyMN4h1NbZ0i14k6DvqbIW5CCwqkw=="
179     crossorigin="anonymous" referrerpolicy="no-referrer"></script>
180 <script src="https://cdnjs.cloudflare.com/ajax/libs/camanjs/4.1.2/caman.full.min.js"
181     integrity="sha512-JjFeUD2H//Rht+DjVf18Tuy1X5ZPtMl0svQ3RopX641DwoSilJ89LsFGq4Sw/6BSBfULqUW/CfnVopV5CfvRXA=="
182     crossorigin="anonymous" referrerpolicy="no-referrer"></script>
183 <script src="script.js"></script>
```

Les scripts étant ajoutés j'ai pu ensuite commencer à écrire le javascript, pour ajouter le cropper sur mon image je me suis référé à la documentation de CropperJS.

J'ai d'abord commencé par récupérer mon image dans une variable Javascript pour que je puisse l'appeler plus tard dans mes fonctions.

```
2 const image = document.getElementById("image");
```

La syntaxe pour ajouter un nouveau cropper est la classe Cropper:

*new Cropper(element[, options])*

- "element" représente soit une balise HTML image ou une balise HTML canvas.  
(pour ma part j'ai utilisé une balise image)
- "option" Représente les options possible dans le cropper

J'ai donc appliqué cette syntaxe sur mon code javascript, où mon élément est l'image appelée précédemment dans ma variable "image".

```
23 // Ajout de cropperJS sur nos images avec un format de 16/9
24 cropper = new Cropper(image, {
25     aspectRatio: 16 / 9,
26     movable: false,
27     zoomable: false,
28     rotatable: false,
29     scalable: false,
30     crop(event) {
31         console.log(event);
32         dataX.value = Math.round(event.detail.x);
33         dataY.value = Math.round(event.detail.y);
34         dataWidth.value = Math.round(event.detail.width);
35         dataHeight.value = Math.round(event.detail.height);
36         const canvas = this.cropper.getCroppedCanvas();
37     },
38 });
```

J'ai donc rajouté les options du cropper dans ma fonction.

- aspectRatio : Défini le format du cropper, l'entreprise ne voulant pas un cropper avec une région libre j'ai donc mis le ratio de 16/9 qui représente le ratio des écrans large avec une définition minimum de 1280 par 720

Afin d'avoir un cropper Responsive j'ai aussi ajouté l'option suivante.

- Responsive : l'option permet d'adapter le cropper a toute les tailles d'écrans, l'image ne gardera pas ses dimensions d'origine.

J'ai également ajouté la fonction crop ayant comme paramètre "event" qui va me permettre de voir dans la console tous les évènements du cropper. J'ai ici récupérer les événement suivant :

- detail.x : le début du cropper sur l'axe des abscisses par rapport au canevas du cropper
- detail.y : le début du cropper sur l'axe des ordonnées par rapport au canevas du cropper
- detail.width : largeur du cropper
- detail.height: longueur du cropper

J'ai ainsi affiché ses éléments qui sont pour moi utiles pour les membres de l'entreprise.

Nous avons maintenant un cropper ajouté sur notre image ainsi que les informations nécessaires.



Le cropper a été ajouté, j'ai pu ensuite développer la fonction qui nous permet de cropper l'image.

J'ai commencé par ajouter un bouton.

```
84 <button id="cropImage">Crop</button>
```

J'ai ensuite appelé mon bouton dans mon script Javascript et je lui ai ajouté un `addEventListener()` pour que à chaque clique au bouton celui-ci appelle la fonction qui va permettre de cropper l'image.

```
4  const crop = document.getElementById("cropImage");  
  
20 // Au clique du bouton on lance la fonction cropImage  
21 crop.addEventListener("click", cropImage);
```

Pour cette fonction je me suis aussi référé à la documentation de CropperJs.

```
37 // Fonction qui permet de recuperer l'image cropper  
38 function cropImage() {  
39     let editImage = document.getElementById("editImage")  
40     // Convertit le contenu du canvas en données utilisable dans le src d'une image  
41     const imgUrl = cropper.getCroppedCanvas().toDataURL();  
42     // Création de l'élément image comportant son id et sa source  
43     const img = document.createElement("img");  
44     img.id = "editImage";  
45     img.src = imgUrl;  
46     // Supression de l'image si l'image est cropper une nouvelle fois  
47     result.removeChild(editImage);  
48     // Appel de l'image au noeud parent donné  
49     result.appendChild(img);
```

La fonction qui me permet d'extraire l'image croppée est "`getCroppedCanvas().toDataURL()`", qui va nous envoyer une URL de données contenant une représentation de l'image dessinée.

Le format de fichier et la qualité de l'image peuvent être spécifiés dans les paramètres de "`toDataURL()`" mais comme je n'ai rien spécifié les données seront exportées au format (image/png)

Je crée ensuite une variable, avec la fonction "`createElement()`" en lui passant en paramètre (img) pour qu'une nouvelle image s'affiche sur ma page web.

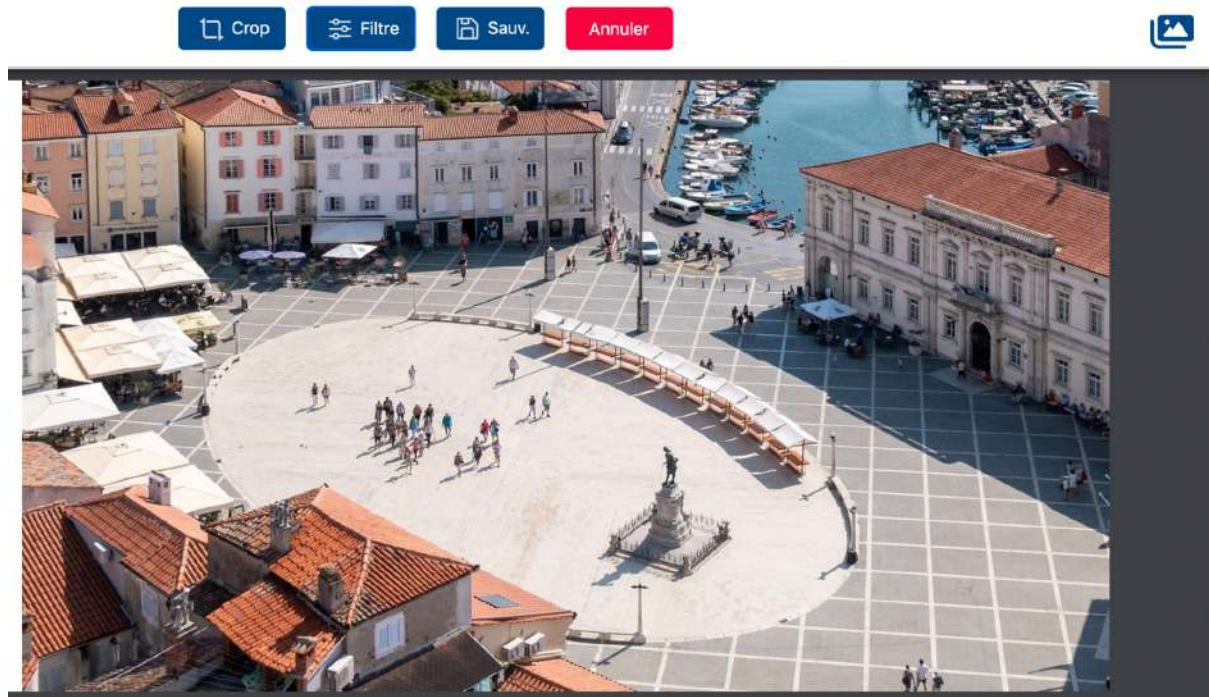
Je défini l'id et la source de l'image.

- `img.id = "editImage"`, qui représente l'id de l'image.
- `img.src = imgUrl`, la source de l'image est URL de données que nous avons récupéré précédemment qui est donc une image au format png.



Avant d'appeler l'image au noeud parent donnée, je décide de mettre la fonction "removeChild()", ce qui va me permettre de supprimer l'instance précédente a chaque fois qu'une nouvelle image est créée, ceci évite qu'il y ait une infinité d'image qui se crée sur notre page.

Ensuite j'appelle donc mon image au noeud parent donnée pour qu'elle s'affiche sur notre page web et nous avons bien l'image recadrer qui apparaît sur notre écran.



Une fois l'image récupérée, j'ai pu commencer à développer la partie des filtres pour que l'on puisse ajuster l'image avec des filtres.

Après une discussion sur l'ajout des filtres avec mon maître de stage, nous avons décidé de ne mettre que quelques filtres nous permettant d'ajuster l'image basiquement.

Nous sommes donc parti sur ses quatre filtres :

- Luminosité
- Contraste
- Saturation
- Vibrance

Pour la partie développement j'ai commencé par ajouter 4 inputs de type range sur mon fichier image.php, ses input vont permettre à l'utilisateur de régler le taux de filtres qu'il veut appliquer sur son image.

```

108 <div class="col p-3">
109   <div class="label-group">
110     
111     <label for="contrast">Contrast</label>
112   </div>
113   <div class="range-group">
114     <input class="filter-range" type="range" id="contrast" min="-50" max="50" value="0"
115       oninput="ctrval.value=contrast.value">
116     <output name="ctrval" id="ctrval" for="contrast">0</output>
117   </div>
118 </div>
119 <div class="col p-3">
120   <div class="label-group">
121     
122     <label for="saturation">Saturation</label>
123   </div>
124   <div class="range-group">
125     <input class="filter-range" type="range" id="saturation" min="-100" max="100" value="0"
126       oninput="satval.value=saturation.value">
127     <output name="satval" id="satval" for="saturation">0</output>
128   </div>
129 </div>
130 <div class="col p-3">
131   <div class="label-group">
132     
133     <label for="vibrance">Vibrance</label>
134   </div>
135   <div class="range-group">
136     <input class="filter-range" type="range" id="vibrance" min="-100" max="100" value="0"
137       oninput="vibval.value=vibrance.value">
138     <output name="vibval" id="vibval" for="vibrance">0</output>
139   </div>
140 </div>

```

J'ai ensuite appelé tous les inputs dans mon script Javascript.

Pour aller plus vite dans le développement de cette partie j'ai décidé d'utiliser jQuery.

Pour ne pas être confus je vais tout de même vous expliquer ce que j'ai utilisé.

```

51 // Selection de tout le inputs type range
52 $('input[type=range]').change(applyFilters);

```

Le code en Javascript Vanilla se définit par:

- document.querySelectorAll('input[type=range]').addEventListener("change", applyFilters);

A chaque changement des valeurs de l'input, ils vont appeler la fonction applyFilter qui est ma fonction pour appliquer les filtres.

```

54 function applyFilters() {
55   // Récupération des valeurs des inputs
56   var brit = parseInt($('#brightness').val());
57   var cntrst = parseInt($('#contrast').val());
58   var vibr = parseInt($('#vibrance').val());
59   var sat = parseInt($('#saturation').val());
60
61   //Fonction de CamanJS qui applique les filtres sur l'image sélectionnée
62   Caman('#editImage', img, function () {
63     // Permettant à l'objet de revenir en arrière
64     this.revert(false);
65     this.brightness(brit).contrast(cntrst).vibrance(vibr).saturation(sat).render();
66   });
67 }

```

J'ai récupéré toutes les valeurs de chaque inputs.

Le code en Javascript Vanilla se définit par:

- `var filtre = parseInt(document.getElementById("filtre").value);`

La fonction `parseInt()` permet de nous envoyer sous forme d'entier les arguments analysés dans une chaîne de caractères.

Pour la suite, je me suis référé à la documentation de CamanJs.

A l'aide de la class Caman nous avons remplacé l'image par un canvas en récupérant l'id de l'image.

Pour l'ajout des filtres je récupère la valeur des inputs qui vont être appliqués grâce au fonction de CamanJs.

La fonction `"render()"` commence le processus de rendu. Cela exécutera toutes les fonctions de filtrage appelées depuis l'initialisation ou depuis le rendu précédent.

```
65      this.brightness(brit).contrast(ctrst).vibrance(vibr).saturation(sat).render();
```

Pour que l'on puisse retourner à la valeur par défaut de l'image on utilise la fonction `"revert()"` ayant comme paramètre `"false"`, grâce à cette fonction nous pouvons aller dans les valeurs positives et retourner dans les valeurs négatives sans que la librairie nous rajoute des calques supplémentaires.

```
64      this.revert(false);
```

Pour éviter de mettre toutes les valeurs à 0 pour récupérer l'image par défaut, j'ai inséré un bouton qui supprime tous les filtres appliqués sur l'image. Ce bouton nous permet de retoucher à zéro la photo si elle ne nous correspond pas.

```
80      revertBtn.addEventListener("click", e => {  
81          Caman("#editImage", function () {  
82              this.revert();  
83          });  
84      });
```

Comme la fonction précédente, celle-ci s'applique sur le canvas qui le rétablit à son état d'origine tout en conservant les dimensions recadrées ou redimensionnées.

Nous avons maintenant une image qui est redimensionnée et sur laquelle on peut appliquer des filtres.



Ayant à présent toutes les fonctionnalités, il m'a fallu à présent remplacer l'image présente dans la base de données par celle modifiée.

Pour ça j'ai d'abord dû extraire l'image du canvas. J'ai donc fait un bouton qui au clic va appeler ma fonction. Pour l'envoi du fichier dans la base de données j'ai encodé mon image en base64.

Base64 est un algorithme d'encodage qui permet de convertir n'importe quel caractère en un alphabet qui se compose de lettres Latin, chiffres, plus, et slash. Grâce à cet algorithme, même des caractères chinois, emojis, et images peuvent être convertis en une chaîne "lisible".

```
86 // Extraction de l'image du canvas
87 upload.addEventListener("click", e => {
88     Caman("#editImage", function () {
89         // Conversion de l'image en Base64
90         var imageFinal = this.toBase64();
91         // Envoi de l'image a la fonction saveToServer
92         saveToServer(imageFinal);
93     })
94 }
95 }
```

Après le rendu de l'image j'envoi mon image vers ma fonction "saveToServer()" en lui passant comme paramètre le rendu de mon image.

Ma fonction saveToServer contient une requête ajax, grâce à cette requête il est alors possible d'appeler un fichier php distant et donc de récupérer des données ou d'en envoyer. Pour ma part je vais envoyer mon image afin de l'enregistrer dans la base de données à l'aide d'une requête SQL.



Pour la requête j'ai utilisé la méthode de jQuery, l'objet XMLHttpRequest permet de répondre à ce besoin, cependant son utilisation en JavaScript natif est relativement lourde à mettre en place. JQuery permet de réaliser des appels Ajax de manière beaucoup plus simple.

```
97 function saveToServer(imageFinal) {
98     // Utilisation d'une requête ajax vers le fichier upload.php
99     $.ajax({
100         type: "POST",
101         url: "upload/upload.php",
102         data: {
103             // La donnée à envoyer est l'image du canvas et l'id de l'image
104             imgBase64: imageFinal,
105             idUp: idUpdate
106         }
107     }).done(function (o) {
108         console.log('Upload success');
109     });
110 }
```

J'envoie ma requête avec la méthode "POST" vers mon fichier upload.php. J'ai ici 2 données que j'envoie vers mon fichier.

Mon image et l'ID de l'image présente dans la base de données.

Ayant eu beaucoup de problèmes pour récupérer l'id à l'aide de la méthode \$\_GET, j'ai donc décidé de la récupérer d'une manière différente.

J'ai créé dans mon fichier image.php un input de type hidden qui prend comme valeur le \$\_GET[id] de l'image sélectionnée.

```
56 <input id="udp" type="hidden" value="<?= $_GET['id']?>">
```

Je lui attribue un "id" pour que je puisse récupérer mon élément dans mon script JS. J'appelle donc mon élément en précisant que je récupère sa valeur.

```
10 const idUpdate = document.getElementById("udp").value;
```

Je vais ensuite envoyer ses informations via ma requête ajax vers mon fichier "upload.php".

En m'aidant de la console je vérifie que tout la requête s'est effectuée sans aucun problème.

J'ai ensuite développé la partie back-end de l'enregistrement de l'image modifiée.

```

1  <?php
2  require "../bdd.php";
3
4      $img = $_POST['imgBase64'];
5      $img = str_replace('data:image/png;base64,', '', $img);
6      $img = str_replace(' ', '+', $img);
7      $data = base64_decode($img);
8      $file = uniqid('', true) . '.png';
9      move_uploaded_file($img, './upload/'.$file);
10     $success = file_put_contents($file, $data);
11     print $success ? $file : 'Unable to save the file.';

```

Dans un premier temps j'ai inclus le fichier avec la connexion a ma base de données a mon fichier. Sachant que je vais effectuer une requête SQL, il est important pour moi d'avoir cette connexion pour communiquer avec le contenu de ma table.

Les données ont été envoyées via la requête AJAX avec la méthode POST. Je vais donc les appeler dans mon fichier php. J'appelle d'abord mon image que je stocke dans une variable.

- `$img = $_POST['imgBase64']`

Sur cette même variable j'utilise la fonction `str_replace()` qui va remplacer toutes les occurrences de ma chaîne de caractère. Rappelons que notre image est encodée en base64. Ici la fonction va nous générer une image en png. L'image pour l'instant n'a pas de nom.

Je crée ensuite une nouvelle variable permettant de décoder l'image en base64:

- `$data = base64_decode($img)`

La fonction `base64_decode` décode une chaîne en MIME base64. La fonction retourne les données décodées ou `false` si une erreur survient. Les informations initiales peuvent être binaires.

Je donne ensuite un nom unique au fichier.

- `$file = uniqid('', true) . ".png"`

On utilise `uniqid` qui nous génère un identifiant unique.

On ajoute un opérateur de concaténation qui retourne la concaténation de ses deux arguments, et à la fin une chaîne de caractère `".png"` qui représente l'extension du fichier sauvegardé.

On peut enfin enregistrer le fichier dans un dossier, pour ma part ce sera le même dossier où sont enregistrés les fichiers du formulaire présent dans la page `index.php`.

- `move_uploaded_file($img, './upload/' . $file)`

La fonction `move_uploaded_file()` nous déplace un fichier téléchargé.

On utilise ensuite la fonction `file_put_contents()`, qui va écrire les données de l'image dans le fichier.

- `$success = file_put_contents($file, $data)`

Dans cette situation notre fonction va écrire les données récupérées par `$data` pour les écrire dans `$file`.

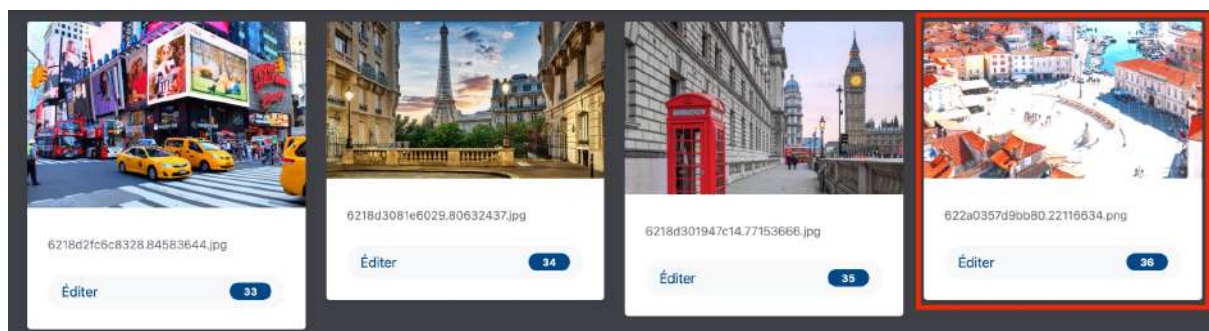
Une fois toutes ces étapes réalisées j'ai bien l'image qui est enregistrée dans mon dossier, j'ai pu ensuite préparer ma requête SQL afin de modifier mon fichier dans la base de données.

Pour ma requête SQL j'utilise la requête `UPDATE` qui va mettre à jour mes éléments.

```
13 $req = $db->prepare('UPDATE file SET name=? WHERE id='.$_POST['idUp'] );
14 $req->execute([$file]);
```

Pour l'id je récupère bien ma donnée envoyée via ma requête AJAX en méthode `POST`. J'ai donc bien l'ID de l'image que je veux mettre à jour dans ma base de données.

Nous pouvons maintenant vérifier si notre image a bien été mise à jour.



Notre image a bien été mise à jour et toutes les fonctionnalités de l'application ont donc bien été respectées.

Après avoir validé les fonctionnalités avec mon maître de stage, nous avons mis en place l'interface avec le graphiste interne à l'entreprise en respectant la palette de couleurs qui m'a été transmise.









## 6. Présentation du jeu d'essai élaboré de la fonctionnalité.

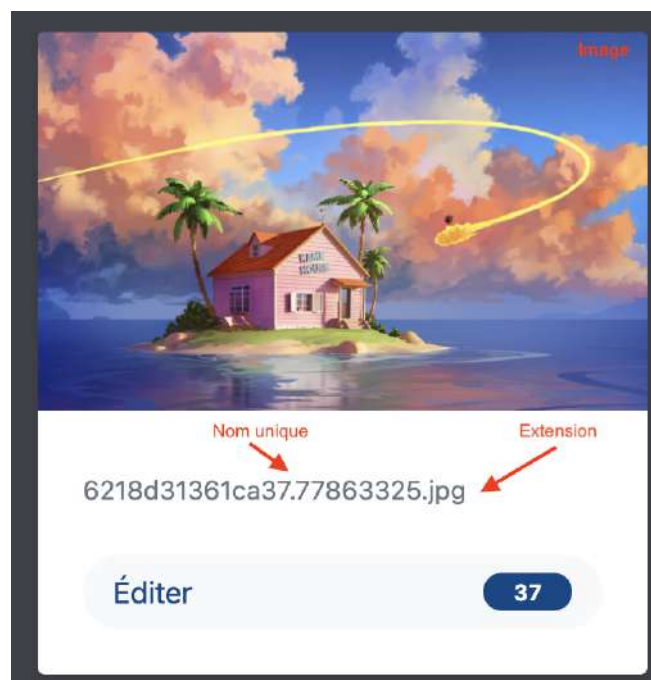
### 1. Formulaire d'ajout de fichier dans la base de données

Après la mise en place du traitement du formulaire j'ai vérifié:

- Que les fichiers se sont bien insérés et qu'il comporte tous un nom unique ainsi que leur extension d'origine.

					id	name
<input type="checkbox"/>		Éditer		Copier		Supprimer
33	6218d2fc6c8328.84583644.jpg					
<input type="checkbox"/>		Éditer		Copier		Supprimer
34	6218d3081e6029.80632437.jpg					
<input type="checkbox"/>		Éditer		Copier		Supprimer
35	6218d301947c14.77153666.jpg					
<input type="checkbox"/>		Éditer		Copier		Supprimer
36	622a0357d9bb80.22116634.png					

- Et que toutes les informations sont bien affichées sur ma page web.



## 2. Mise à jour des images dans la base données

Après avoir mis en place le traitement des mise a jour des images de la base de données.

- J'ai vérifié que l'image s'est bien remplacée dans la base de données et qu'elle possède bien l'extension png



- Et j'ai vérifié que l'image s'est bien mise à jour et s'affiche correctement avec toutes ses informations sur notre page web.



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

### 1. Éviter les injections SQL

- Qu'est ce qu'une injection SQL ?

Une injection SQL est une forme de cyberattaque lors de laquelle un pirate utilise un morceau de code SQL (« Structured Query Language », langage de requête structurée) pour manipuler une base de données et accéder à des informations potentiellement importantes.

Pour éviter les injections SQL il faut filtrer et nettoyer les données servant d'entrée aux instructions SQL (comme par exemple les entrées de type texte d'un formulaire). Un autre moyen plus efficace pour se protéger des injections SQL est l'utilisation des requêtes préparées (prepared statements).

Pour ma part toutes les requêtes présentes dans mon code utilisent la méthode PDO, je peux donc créer les instructions de préparation et avoir des requêtes préparées.

J'ai aussi enlevé toutes les fonctions qui me montraient les messages d'erreurs pour ainsi éviter de divulguer des informations au cybercriminel.

```
error_reporting(E_ALL);  
ini_set("display_errors", 1);
```

### 2. Éviter les attaques XSS

- Qu'est ce qu'une attaque XSS ?

Le cross-site scripting est un type de faille de sécurité des sites web permettant d'injecter du contenu dans une page, provoquant ainsi des actions sur les navigateurs web visitant la page.

Pour éviter ses attaques, on utilise le principe "filter on input" et "escape on output". Nous devons appliquer ces vérifications sur tous les paramètres, y compris les paramètres cachés, les paramètres GET/POST, cookies et headers HTTP.

### 3. Éviter la falsification de demandes intersites

- Qu'est ce que la falsification de demandes intersites ?



Cross Site Request Forgery (CSRF) est l'une des vulnérabilités les plus graves qui peut être exploitée de diverses manières, de la modification des informations de l'utilisateur à son insu à l'accès complet au compte de l'utilisateur.

Pour éviter de telles attaques, il faut utiliser dans les formulaires qui le permette utiliser la méthode POST plutôt que la méthode GET.

L'utilisation de HTTPS qui est devenue la nouvelle norme a juste titre. HTTPS crypte la connexion entre votre site Web et l'utilisateur.

Dans mon application j'ai donc utilisé la méthode POST dans mes formulaires lorsque cela était possible.

N'ayant pas intégré l'application sur le serveur de l'entreprise, je n'ai pas pu procéder à l'utilisation du protocole HTTPS.

#### 4. Avoid Vulnerable JavaScript Libraries

Notre site web peut utiliser des bibliothèques JavaScript qui imposent des risques de sécurité. On considère généralement que 20% de ces bibliothèques rendent un site Web plus vulnérable.

J'ai donc à l'aide du service de Vulnerability DB vérifier si les librairies que j'ai utilisées étaient bien sûres.

- CropperJS

**Security** NO KNOWN SECURITY ISSUES ?

La librairie est bien sécurisée et ne comporte aucun problème de sécurité connu.

- CamanJS

**Security** NO KNOWN SECURITY ISSUES ?

La librairie est également sécurisée et ne comporte aucun problème de sécurité connu.

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

Durant mon stage, j'ai dû m'adapter à 2 nouvelles librairies javascript que je ne connaissais pas. La plupart des sites étant en anglais et il y avait très peu de ressources en français j'ai donc demandé toutes mes recherches en anglais à google. Il était ainsi plus simple de trouver des solutions à mes problèmes. Je vais donc pour cette section vous parler de mon travail de recherche qui consiste à envoyer une image du canvas javascript à la base de données php. J'ai donc tapé ma recherche dans google "How to save an image canvas to a server ?". Pour résoudre mon problème je suis allé sur le site [geeksforgeeks.com](https://www.geeksforgeeks.org) qui répond exactement à ma situation.



How to save an image canvas to a server ?



<https://www.geeksforgeeks.org> > ho... ▾ Traduire cette page

### How to save an HTML 5 Canvas as an image on the server

20 août 2019 — Step 1: HTML code to convert **canvas** into **image**.

<https://usefulangle.com> > post > java... ▾ Traduire cette page

### Uploading Canvas Image to a Server - UsefulAngle

4 janv. 2021 — Uploading as Data URL String ... The data URL of the **canvas image** can be retrieved using the `toDataURL()` method. ... // PNG data url let ...

<https://gist.github.com> > kylejson ▾ Traduire cette page

### Send canvas as image to server/save form input in localStorage

Send **canvas** as **image** to **server/save** form input in localStorage - test.html.

<https://levelup.gitconnected.com> > h... ▾ Traduire cette page

### How to Save HTML5 Canvas Drawing as an Image - Level Up ...

13 oct. 2020 — As for the second method, I am going to convert the **canvas** as blob data and send it via `FormData` to a remote **server**. After that, you can easily ...



## 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

**Site anglophone utilisé :**

- <https://www.geeksforgeeks.org/how-to-save-an-html-5-canvas-as-an-image-on-the-server/>

**Extrait anglophone :**

How to save an HTML 5 Canvas as an image on the server ?

Last Updated : 20 Aug, 2019

Saving HTML canvas as an image is pretty easy, it can be done by just right-clicking on the canvas and save it as an image. But saving canvas as an image on the server is quite different. This article will show you how to achieve that. Sometimes it is required to save canvas image after doing some server processing and this article will help in sending canvas image over the server for processing.

Here you will see, how to convert an HTML content into an image using JQuery. We know that there is a plugins html2canvas, by using that plugin we can easily convert the HTML content into image content after that we can save that file by right-clicking and choosing the save image option.

After that, we will convert that CANVAS image into the URL format and that to the server by using ajax after that the main part will be done by the PHP code. The PHP code will save that image on your server.

Below steps will illustrate the approach clearly.

Step 1: HTML code to convert canvas into image.

Step 2: Display the output to check that the canvas is successfully converted into an image.

Step 3: Converting the image into URL format using canvas.toDataURL() method.

Step 4: Sending the converted URL format into your server via Ajax.

Step 5: This php code will save the image into the server.

## **Extrait traduit en Français :**

### Comment enregistrer un Canvas HTML 5 en image sur le serveur ?

Dernière mise à jour : 20 août 2019

Sauvegarder le canvas HTML en image est vraiment simple. Cela peut être effectué juste en cliquant avec le bouton droit de la souris sur le canvas et l'enregistrer en tant qu'image. Mais pour enregistrer le canvas en image sur le serveur, la méthode est assez différente. Cet article vous montre comment l'effectuer. Parfois il est recommandé de sauvegarder l'image du canvas après avoir fait un traitement sur le serveur, cet article vous aidera pour le traitement pour l'envoi de l'image du canvas sur le serveur. Ici vous allez voir comment convertir un contenu HTML en image en utilisant JQuery. Nous savons qu'il y a un plugin html2canvas, à l'aide de ce plugin nous pouvons facilement convertir le contenu HTML en image, même si nous pouvons enregistrer l'image à l'aide du bouton droit de la souris en choisissant l'option enregistrer l'image.

Après ça nous allons convertir l'image du canvas au format URL et l'envoyer au serveur en utilisant AJAX. Après sa partie principale se fera en PHP, le PHP va nous permettre d'enregistrer l'image sur votre serveur.

Les étapes suivantes illustreront clairement l'approche.

Étape 1: Le code HTML pour convertir le canvas en image.

Étape 2 : Afficher la sortie pour vérifier que le canvas s'est correctement converti en image.

Étape 3 : Convertir l'image au format URL en utilisant la méthode `canvas.toDataURL()`.

Étape 4 : Envoi de l'URL converti sur le serveur à l'aide de la méthode AJAX

Étape 5 : Le code PHP enregistrera l'image dans le serveur.

# 10. Annexes

