

Rapport projet de langages Web

Tristan GROULT

9 mai 2025

Sommaire

1	Introduction	3
2	Structure du projet	3
3	Problèmes rencontrés	4
3.1	Les images	4
3.1.1	Ajout	4
3.1.2	Accès	4
3.2	Les annotations	4
3.3	Les modifications d'images et d'annotations	5
4	Améliorations	5

1 Introduction

Ce projet vise à la création d'une application web avec des utilisateurs ayant la possibilité d'ajouter des images sur le site. Les images peuvent être annotées et taguées afin de créer des catégories et d'indiquer sur l'image des éléments. Pour ce projet, j'ai décidé de nommer l'application « Annotimage ».

2 Structure du projet

Ce projet se base sur l'architecture *Modèle-Vue-Contrôleur* (MVC) [figure 1](#). La racine du projet, pour le serveur, se situe dans le dossier `public/` qui contient le fichier `index.php` qui fait office de fichier « route » permettant la redirection des requêtes à leurs bonne destination. Le fichier `models.php` contient la partie modèle de cette architecture et permet de regrouper les fonctions étant utilisées à plusieurs endroits. Le dossier `templates/` contient quant à lui les différentes vues pour ce projet.

```
.
|-- config.php
|-- controllers.php
|-- debug.php
|-- images/
|-- models.php
|-- public/
|   |-- imageInfo.php
|   |-- image.php
|   |-- index.php
|   '-- static/
|       |-- css/
|       |-- fonts/
|       |-- images/
|       '-- js/
|-- readme.md
'-- templates/
```

FIGURE 1 – Structure du projet

Pour permettre la protection des données ainsi que d'empêcher les utilisateurs ou toute personne mal intentionnée d'accéder à des informations auxquelles elle ne sont pas censées accéder, la racine du projet se situe dans le dossier `public/`. Cela assure que seules les ressources disponibles dans ce fichier sont accessibles. Le dossier `public/` contient :

- Les fichiers `image.php` respectivement `imageInfo.php` permettent d'accéder à une image respectivement toutes les informations (au format `json`) d'une image en vérifiant au préalable les droits d'accès à cette image pour l'utilisateur courant. Cet accès se fait respectivement grâce au nom de l'image et grâce à l'ID de l'image.
- Le dossier `static/` contient toutes les ressources statiques utilisées par le site. Cela inclut les images et illustrations, le style CSS, les polices d'écriture ainsi que le code javascript. Les fichiers javascript sont regroupés au sein du dossier `js/`. `code.js` regroupe les fonctions utilitaires pouvant être utilisées par différentes pages ainsi que le

fonctionnement de base de toutes page. Cependant, chaque page ayant besoin d'un code javascript qui lui est propre a un fichier javascript associé pour cela.

Le fichier `config.php` contient la configuration nécessaire à la connexion à la base de données. Le fichier `debug.php` permet quant à lui d'activer ou de désactiver les options de débogage selon que le site soit en développement ou non. Pour finir, le fichier `controler.php` contient toutes les fonctions vers lequel `index.php` redirige les utilisateurs. Ces fonctions permettent l'affichage correct des vues ainsi que le traitement des formulaires.

L'ensemble des images des utilisateurs sont stocké dans le dossier `images/`. Toutes les images sont tout d'abord renommées avant d'être ajoutées afin d'éviter les doublons.

3 Problèmes rencontrés

3.1 Les images

3.1.1 Ajout

Lors de l'ajout des images, une attention particulière est portée sur le fait que les fichiers envoyés soient des images correctes. Cependant, ce n'est pas suffisant, il faut aussi que les fichiers images ne portent pas des noms identiques pour éviter tout problème de fichiers écrasés et donc d'images qui disparaissent. Pour cela, une requête est d'abord faite vers la base SQL pour ajouter les informations connues de l'image, l'identifiant unique qui est la clé primaire de la table image est ensuite récupérée pour renommer l'image qui vient d'être ajoutée. Cela permet de garantir un nom unique pour l'image qui vient d'être ajoutée. La base de données étant ensuite mise à jour avec le chemin vers cette image.

3.1.2 Accès

Les images pouvant être publiques comme privées, un accès spécifique doit être développé afin d'empêcher les utilisateurs d'accéder à toutes les images. Pour cela, j'utilise le fichier `image.php` qui me permet avec une requête de la forme `/image?id=nom.png` d'accéder à l'image `nom.png`. Pour cela, le fichier PHP fait une requête vers la base de données pour vérifier que l'image existe. Si elle existe, il vérifie qu'elle est soit public, soit privé et dans ce cas-là, si l'utilisateur connecté correspond à l'utilisateur qui a déposé cette image. Le code possède ensuite à l'envoi de l'image grâce à la fonction `header('Content-Type: image/' . fileType)`.

Un problème c'est aussi posé sur les formats des images. Nous prenons donc en charge 3 formats d'images, le `jpeg`, le `png` et le `webp` car les images doivent être ouvert avec une fonction spécifique à leur format et la fonction `header` doit être appelée avec le `fileType` de l'image envoyé.

3.2 Les annotations

Les annotations sont des rectangles qui sont spécifiées grâce à 2 points avec leurs coordonnées X et Y représentant 2 coins opposés du rectangle. Les images pouvant avoir des dimensions variées et l'affichage des images pouvant se faire sur des périphériques de taille différentes. Une translation des points doit être appliquée afin que les annotations apparaissent toujours au même endroit. Dans les faits, Lorsqu'un utilisateur annote une image, il sélectionne une zone sur l'image affichée sur son périphérique. Connaissant la taille de l'image affichée ainsi que la taille réelle de l'image, Nous pouvons convertir les coordonnées des 2 points sélectionnés par l'utilisateur en coordonnées équivalentes sur l'image d'origine. Pour l'affichage, il suffit donc de

reconvertir les coordonnées données sur la taille d'images originales en coordonnées sur l'image affichée sur le périphérique du client. L'entiereté de ces actions est gérée par javascript pour que l'affichage chez le client soit dynamique.

3.3 Les modifications d'images et d'annotations

La gestion des tags des images ainsi que des annotations étant dynamiques et gérée par javascript, le chargement d'images déjà ajouté pour les modifier doit pouvoir garder le fonctionnement dynamique. Il est donc impossible de charger les annotations et tags directement en PHP. Pour cela, lorsqu'une image doit être modifiée, le code javascript fait une requête vers le serveur pour demander les informations d'une image grâce à la requête `/imageIngo.php?id=nb` où `nb` représente l'identifiant de l'image. La réponse du serveur est alors du json contenant les informations de l'image qui peut être ensuite analysé en javascript. Après cela, comme pour l'ajout manuel des tags et des annotations qui auraient pu être faits par l'utilisateur, les annotations et tags sont ajoutées permettant la gestion dynamique de ces éléments.

4 Améliorations

Les améliorations suivantes ont été ajoutées au projet :

- Pour l'ensemble des images, l'utilisateur déposant une image a l'opportunité de déposer avec des tags permettant d'associer autant de catégories qu'ils veulent à une image et donc de regrouper les images aussi entre elle.
- Toutes les entrées utilisateurs sont sanitarisées afin qu'aucun code PHP ou HTML ne puisse être exécuté.
- Les images déposées sont rigoureusement vérifiées pour être sûr qu'elles vérifient bien le statut d'image.
- Les mots de passe sont stockés sous forme de hashes pour plus de sécurité.
- Un système de notification permettant dans PHP d'ajouter des informations à afficher à leurs utilisateurs afin que celui ci puisse savoir les erreurs ou les informations sur ce qui vient d'arriver. Les notifications s'affichent quelques secondes, puis disparaissent où l'utilisateur peut les fermer de lui même.
- Lorsque c'est possible, un système de « callback » est utilisé permettant à l'utilisateur lorsqu'il ferme une page ou après avoir ajouté ou modifié une image d'être redirigé vers la page depuis laquelle il a accédé à cette ressource.
- Une page paramètres dans laquelle les utilisateurs peuvent :
 - Modifier les informations de leur compte en modifiant l'adresse e mail ou le nom d'utilisateur qui leur est associé. Les utilisateurs peuvent aussi modifier leur mot de passe.
 - Supprimer leur compte en supprimant toutes les images et annotations associées au compte
 - Se déconnecter.
- Une page de mise à jour permettant de mettre à jour les informations d'une image incluant la description, les tags qui leur est associés ainsi que sa visibilité.
- Une amélioration à la page annotation qui permet aussi de continuer l'annotation d'une image ayant déjà été annotée ainsi que de supprimer les notations ayant déjà été ajoutées.
- Une page permettant l'affichage de toutes les images présentant un tag spécifié.
- Une option permettant de supprimer une image ayant été ajoutée par un utilisateur.

- Une option permettant d'afficher et de cacher les annotations présentes sur une image.
- Une option de recherche permettant de rechercher un utilisateur en préfixant le mot d'un arobase permettant aussi de rechercher des tags en le préférant d'un dièse et permettant de rechercher des mots présents dans la description en ne le préfixant pas. L'ensemble de ces éléments peuvent être recherchés en même temps et seront traités comme une conjonction de disjonction selon leur type.
- L'entièreté du site est « responsive » et peut donc être utilisée à la fois sur PC et sur mobile.