



# Web

## Projet Camagru

42 Staff [pedago@staff.42.fr](mailto:pedago@staff.42.fr)  
coton [coton@42.fr](mailto:coton@42.fr)

*Résumé: Ce projet a pour but de vous faire réaliser une application web un peu plus complète que les précédentes, avec un peu moins de moyens.*

# Table des matières

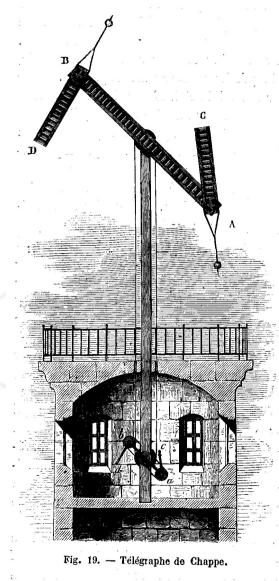
<b>I</b>	<b>Préambule</b>	<b>2</b>
<b>II</b>	<b>Introduction</b>	<b>4</b>
<b>III</b>	<b>Objectifs</b>	<b>5</b>
<b>IV</b>	<b>Consignes générales</b>	<b>6</b>
<b>V</b>	<b>Partie obligatoire</b>	<b>8</b>
V.1	Partie commune . . . . .	8
V.2	Partie utilisateur . . . . .	9
V.3	Partie galerie . . . . .	9
V.4	Partie montage . . . . .	10
V.5	Contraintes et Obligations . . . . .	11
<b>VI</b>	<b>Partie bonus</b>	<b>12</b>
<b>VII</b>	<b>Rendu et peer-évaluation</b>	<b>13</b>

# Chapitre I

## Préambule

L'histoire de la communication est aussi ancienne que l'histoire de l'humanité, et l'homme a su la faire évoluer au fil des siècles au travers d'incroyables révolutions.

En 1794, Claude Chappe se pencha sur la problématique de la communication entre de longues distances, qui était à l'époque limitée à la vitesse du cheval au galop. Il mit au point un ingénieux système de communication de télégraphe aérien pendant la Révolution. Les "tours de Chappe" étaient coiffées d'un mât mobile, visible à la jumelle de la tour voisine, distante de 10 km à 15 km.



La ligne Paris-Lille fut ainsi opérationnelle dès 1794 et permit par exemple de transmettre des messages entre ces deux villes avec une durée de neuf minutes pour transmettre une lettre via une quinzaine de tours ; le temps de transmission d'un message dépendait de sa longueur.

En 1844, 534 tours quadrillent le territoire français reliant sur plus de 5 000 km les plus importantes agglomérations.

Les gros inconvénients du système étaient qu'il ne pouvait fonctionner ni la nuit ni par mauvaise visibilité et qu'il mobilisait beaucoup d'opérateurs (deux tous les 15 kilomètres

environ).

Heureusement, nous sommes en 2018.

# Chapitre II

## Introduction

Vous voici prêt à créer vos premières applications Web, comme des grands. Si vous l'aviez pas déjà remarqué, le web est un monde vaste et riche, permettant de mettre à disposition rapidement des applications ou du contenu pour le monde entier.

Et puisque vous avez les bases, maintenant il est temps de se détacher des éternelles to-do lists et sites de eCommerce, et de voguer vers des projets de plus grandes envergures.

De même, vous allez maintenant découvrir les joies du :

- Responsive design
- DOM Manipulation
- SQL Debugging
- Cross Site Request Forgery
- Cross Origin Resource Sharing
- ...

# Chapitre III

## Objectifs

Camagru vous propose de créer une petite application web permettant de réaliser des montages basiques à l'aide de votre webcam et d'images prédéfinies.



Evidemment, ces images auront un canal alpha, sinon votre superposition n'aurait pas la prestance escomptée !

On pourra par exemple imaginer le moment fatidique d'un lancer de chat intergalactique, preuve à l'appui :



Un utilisateur de votre site devra donc être capable de sélectionner une image dans une liste d'images superposables (par exemple, des cadres ou des objets à l'utilité douteuse), prendre une photo depuis sa webcam et admirer le résultat d'un montage digne de James Cameron.

Toutes les images prises devront être publiques, like-ables et commentables.

# Chapitre IV

## Consignes générales

- Ce projet ne sera corrigé que par des humains. Vous êtes donc libres d'organiser et de nommer vos fichiers comme vous le désirez, en respectant néanmoins les contraintes listées ici.
- Vous devez rendre, à la racine de votre dépôt de rendu, un fichier **auteur** contenant votre login suivi d'un retour à la ligne :

```
$> cat -e auteur  
xlogin$
```

- Votre application Web ne doit produire aucune erreur, warning ou notice, coté serveur et coté client, dans la console web. Toutefois, en raison de l'absence d'HTTPS, les erreurs relatives à `getUserMedia()` sur la console web seront tolérées.
- Vous devrez seulement utiliser le langage PHP pour concevoir votre application coté serveur, avec seulement la librairie standard.
- Coté client, vos pages devront utiliser HTML, CSS et JavaScript.
- Aucun Framework, Micro-Framework ou librairie qui n'est pas de votre conception n'est autorisée. Vous pourrez utiliser Javascript avec les API natives des navigateurs uniquement. Les Frameworks CSS seront tolérés, tant qu'ils n'ajoutent pas du JavaScript interdit.
- Vous utiliserez interface d'abstraction PDO de PHP pour communiquer avec votre Base de Données, qui devra être requêtable en SQL. Le mode d'erreur de ce driver sera obligatoirement défini sur `PDO::ERRMODE_EXCEPTION`
- Votre application ne devra comporter aucune faille de sécurité. Vous devez au minimum gérer ce qui est indiqué dans la partie obligatoire, mais nous vous engageons à aller plus loin dans la sécurité de votre application, la confidentialité de vos données en dépendent !

- Vous êtes libres d'utiliser le serveur web de votre choix, que ce soit **Apache**, **Nginx** ou même un **built-in web server**<sup>1</sup>.
- L'ensemble de votre application devra être au minimum compatible sur **Firefox** ( $\geq 41$ ) et **Chrome** ( $\geq 46$ )

---

1. <http://php.net/manual/en/features.commandline.webserver.php>



# Chapitre V

## Partie obligatoire

### V.1 Partie commune

Vous devez donc réaliser une application web. Bien que ce n'est pas obligatoire, nous vous conseillons de structurer votre application (en MVC, par exemple).

Votre site devra avoir une mise en page décente (c'est à dire au moins un header, une section principale et un footer), être présentable sur mobile, et avoir un comportement et un layout adaptés sur de petites résolutions (votre navigateur peut simuler ce mode, oui oui...).

Tous vos formulaires doivent avoir des validations correctes, et l'ensemble de votre site devra être sécurisé. Ce point est OBLIGATOIRE et sera vérifié longuement en soutenance. Pour vous faire une petite idée, voici quelques éléments qui ne sont pas considérés comme sécurisés :

- Avoir des mots de passe “en clair” dans une base de données.
- Pouvoir injecter du code HTML ou JavaScript “utilisateur” dans des variables mal protégées.
- Pouvoir uploader du contenu indésirable.
- Pouvoir modifier une requête SQL.
- Utiliser un formulaire externe au site pour modifier du contenu dit-protégé.
- etc...

## V.2 Partie utilisateur

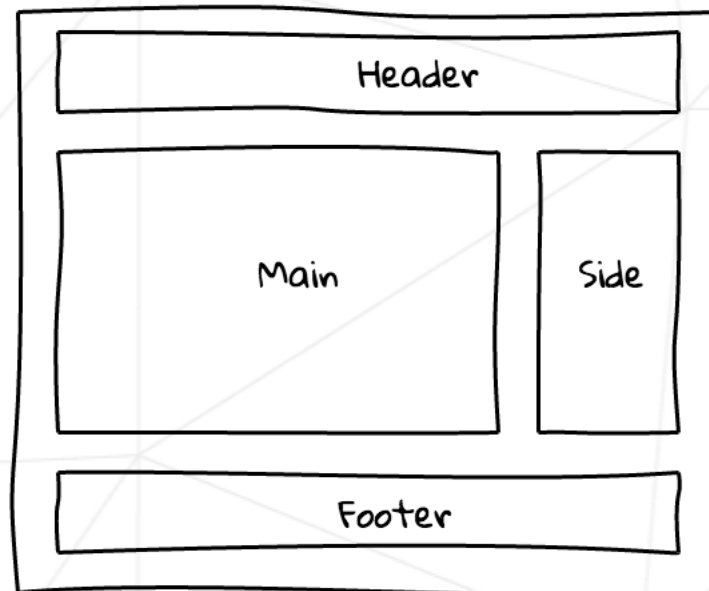
- L'application doit permettre à un utilisateur de s'inscrire, en demandant au minimum une adresse email, un nom d'utilisateur et un mot de passe un tant soit peu sécurisé.
- L'utilisateur doit confirmer son compte via un lien unique envoyé par mail, à l'adresse renseigné dans le formulaire d'inscription.
- L'utilisateur doit ensuite être capable de se connecter avec son nom d'utilisateur et son mot de passe. Il doit également pouvoir recevoir un mail de réinitialisation de son mot de passe en cas d'oubli.
- L'utilisateur doit pouvoir se déconnecter en un seul clic depuis n'importe quelle page du site.
- Une fois connecté, l'utilisateur aura possibilité de modifier son nom d'utilisateur, son adresse mail et son mot de passe.

## V.3 Partie galerie

- La galerie devra être publique, donc accessible sans authentification. Elle doit afficher l'intégralité des images prises par les membres du site, triées par date de création. Elle doit pouvoir permettre à l'utilisateur de les commenter et de les liker si celui-ci est authentifié.
- Lorsque une image reçoit un nouveau commentaire, l'auteur de cette image doit en être informé par mail. Cette préférence est activée par défaut, mais peut être désactivée dans les préférences de l'utilisateur.
- La liste des images doit être paginée, avec au moins 5 éléments par page.

## V.4 Partie montage

FIGURE V.1 – Une vague idée du layout de la page de montage



Cette partie ne doit être accessible qu'aux utilisateurs connectés, et rejeter poliment l'internaute<sup>1</sup> dans le cas contraire.

Cette page devra être composée de deux sections :

- Une section principale, contenant l'aperçu de votre webcam, la liste des images superposables disponibles et un bouton permettant de prendre la photo.
- Une section latérale, affichant les miniatures de toutes les photos prises précédemment.

Votre mise en page doit donc normalement ressembler à la Figure V.1.

- Les images superposables doivent être sélectionnables, et le bouton permettant de prendre la photo ne doit pas être cliquable tant qu'aucune image n'est sélectionnée.
- Le traitement de l'image finale (donc entre autres la superposition des deux images) doit être fait côté serveur, en PHP.
- Parce que tout le monde n'a pas de webcam, vous devez laisser la possibilité d'uploader une image au lieu de la prendre depuis la caméra.
- L'utilisateur doit pouvoir supprimer ses montages, et uniquement les siens.

---

1. oui, le mot internaute vient d'être cité

## V.5 Contraintes et Obligations

Pour résumer, votre super application devra respecter les choix technologiques suivants :

- Langages autorisés :
  - [Serveur] PHP
  - [Client] HTML - CSS - JavaScript (restreints aux API natives des navigateurs)
- Framework autorisés :
  - [Serveur] Aucun
  - [Client] Frameworks CSS tolérés, tant que ça rajoute pas du JavaScript.

En plus de cela, votre projet doit comporter impérativement :

- Un fichier `index.php`, contenant le point d'entrée de votre site, et situé à la racine de votre arborescence.
- Un fichier `config/setup.php`, capable de créer ou recréer le schéma de la base de données, en utilisant les infos contenues dans le fichier `config/database.php`.
- Un fichier `config/database.php`, contenant la configuration de votre base de données qui sera instanciée via PDO au format suivant :

```
$> cat config/database.php
<?php
    $DB_DSN = ...;
    $DB_USER = ...;
    $DB_PASSWORD = ...;
```

Le DSN (Data Source Name) contient les informations requises pour se connecter à la base, par exemple `'mysql:dbname=testdb;host=127.0.0.1'`. En général, un DSN est constitué du nom du pilote PDO, suivi d'une syntaxe spécifique au pilote. Plus de détails sont disponibles dans la documentation PDO de chaque pilote<sup>2</sup>.

---

2. Pour plus d'informations, voir [la documentation du constructeur de PDO](#)

# Chapitre VI

## Partie bonus

Si la partie obligatoire a été réalisée entièrement et parfaitement, vous pouvez ajouter les bonus que vous souhaitez ; ils seront évalués à la discrétion de vos correcteurs. Vous devez néanmoins toujours respecter les contraintes de base. Par exemple, le traitement de l'image doit impérativement se faire côté serveur.

Si l'inspiration vous manque, voici quelques pistes :

- “AJAXifier” les échanges avec le serveur.
- Faire un aperçu du rendu final en live, directement sur l'aperçu de la caméra. On notera que c'est bien plus simple qu'il n'y paraît.
- Faire une pagination infinie sur la partie galerie.
- Pouvoir partager ses images sur les réseaux sociaux.
- Pouvoir faire un rendu d'un GIF animé.

# Chapitre VII

## Rendu et peer-évaluation

Les consignes suivantes seront présentes dans le barème de soutenance. Soyez très attentifs lors de l'application de ces dernières car elles seront sanctionnées par un 0 sans appel.

Vous pouvez poser vos questions sur le forum, Slack, etc...  
Bon courage à tous !