



# Camagru

*Résumé : Le but de ce projet est de créer une application Web.*

*Version : 4*

# Contenu

<b>je</b>	<b>Avant-propos</b>	<b>2</b>
<b>II</b>	<b>Introduction</b>	<b>3</b>
<b>III</b>	<b>Objectifs</b>	<b>4</b>
<b>IV</b>	<b>Instructions générales</b>	<b>5</b>
<b>V</b>	<b>Partie obligatoire</b>	<b>6</b>
V.1	Caractéristiques communes . . . . .	6
V.2	Fonctionnalités utilisateur . . . . .	7
V.3	Fonctionnalités de la galerie. . . . .	7
V.4	Fonctionnalités d'édition. . . . .	8
V.5	Contraintes et choses obligatoires. . . . .	9
<b>VI</b>	<b>Partie bonus</b>	<b>dix</b>
<b>VII</b>	<b>Soumission et évaluation par les pairs</b>	<b>11</b>

# Chapitre I

## Avant-propos

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

En 1794, Claude Chappe tente de résoudre le problème des communications à longue distance, alors limité à la vitesse des chevaux. Il a mis en place un ingénieux système de communication télégraphique aérien pendant la Révolution française. Les « tours » (tours) de Chappe étaient couvertes par un mât mobile visible aux jumelles depuis la tour voisine la plus proche située à environ 10 à 15 km.

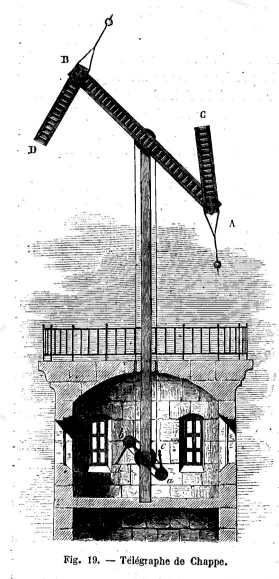


Fig. 19. — Télégraphe de Chappe.

La ligne Paris-Lille est opérationnelle la même année et permet des transmissions entre les deux villes à une vitesse de 9 minutes par lettre à travers 15 tours. Évidemment, le temps de transmission dépend de sa longueur.

En 1844, 534 tours étaient dispersées sur le territoire français reliant sur plus de 5000km, les villes les plus importantes.

Mais ce système présentait 2 gros inconvénients : il ne pouvait pas fonctionner de nuit à cause, évidemment, de la mauvaise visibilité, et du nombre d'opérateurs par tour (2 tous les 15 km).

Heureusement, nous sommes au 21ème siècle.

# Chapitre II

## Introduction

Vous êtes maintenant prêt à créer vos premières applications Web, comme des pros. Si cela ne vous dérange pas, le Web est un monde vaste et riche, qui vous permet de diffuser rapidement des données et du contenu à tous, partout dans le monde.

Maintenant que vous connaissez les bases, le moment est venu d'abandonner ces anciennes listes de tâches et ces sites Web de commerce électronique et de vous envoler vers des projets plus ambitieux.

Aussi, voici le moment pour vous de découvrir de nouvelles notions et la beauté de :

- Conception réactive
- Manipulation du DOM
- Débogage SQL
- Contrefaçon de demande intersite
- Partage de ressources entre origines
- ...

# Chapitre III

## Objectifs

Ce projet web vous met au défi de créer une petite application web vous permettant de réaliser des retouches photo et vidéo de base à l'aide de votre webcam et de quelques images prédéfinies.



Bien évidemment, ces images doivent avoir un canal alpha, sinon votre superposition n'aurait pas l'effet escompté !

Nous imaginerons par exemple le moment précis d'un lancement de chat intergalactique, en voici la preuve :



Les utilisateurs de l'application doivent pouvoir sélectionner une image dans une liste d'images superposables (par exemple un cadre photo ou d'autres objets « nous ne voulons pas savoir à quoi vous utilisez cela »), prendre une photo avec sa webcam et admirez le résultat qui devrait être en mélangeant les deux images.

Toutes les images capturées doivent être publiques, sympathiques et commentables.

# Chapitre IV

## Instructions générales

- Ce projet sera corrigé par des humains uniquement. Vous êtes autorisé à organiser et à nommer vos fichiers comme bon vous semble, mais vous devez suivre les règles suivantes.
- Votre application Web ne doit produire aucune erreur, aucun avertissement ou ligne de journal dans aucune console, côté serveur et côté client. Néanmoins, en raison de l'absence de HTTPS, toute erreur liée à `getUserMedia()` sont tolérées.
- Vous êtes libre d'utiliser n'importe quel langage pour créer votre application côté serveur, mais, pour chaque fonction que vous utilisez, vous devez vérifier qu'un équivalent existe dans la bibliothèque standard PHP.
- Côté client, vos pages doivent utiliser HTML, CSS et JavaScript.
- Une conteneurisation à jour est indispensable.
- N'oubliez pas que certains choix peuvent vous rendre plus attractif sur le marché du travail.
- Tout framework, micro-framework ou bibliothèque que vous ne créez pas et sans équivalent dans la bibliothèque standard PHP est totalement interdit, à l'exception des frameworks CSS qui n'ont pas besoin d'JavaScript.
- Votre application doit être exempte de toute faille de sécurité. Vous devez traiter au minimum les cas mentionnés dans la partie obligatoire. Néanmoins, nous vous encourageons à approfondir la sécurité de votre application, pensez à la confidentialité de vos données !
- Vous êtes libre d'utiliser le serveur Web de votre choix, comme Apache, Nginx ou même le serveur Web intégré<sup>1</sup>.
- Votre application Web doit être au moins compatible avec Firefox ( $\geq 41$ ) et Chrome ( $\geq 46$ ).



Pour des raisons de sécurité évidentes, toutes les informations d'identification, clés API, variables d'environnement, etc... doivent être enregistrées localement dans un fichier `.env` et ignorées par git. Les informations d'identification stockées publiquement vous mèneront directement à l'échec du projet.

---

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

# Chapitre V

## Partie obligatoire

### V.1 Caractéristiques communes

Vous développerez une application web. Même si cela n'est pas obligatoire, essayez de structurer votre application (en application MVC, par exemple).

Votre site Web doit avoir une mise en page décente (c'est-à-dire au moins un en-tête, une section principale et un pied de page), pouvoir s'afficher correctement sur les appareils mobiles et avoir une mise en page adaptée sur les petites résolutions.

Tous vos formulaires doivent avoir des validations correctes et l'ensemble du site doit être sécurisé. Ce point est OBLIGATOIRE et sera vérifié lors de l'évaluation de votre candidature. Pour avoir une idée, voici quelques éléments qui ne sont PAS considérés comme SÉCURISÉS :

- Stockez les mots de passe simples ou non chiffrés dans la base de données.
- Offrent la possibilité d'injecter du HTML ou du JavaScript « utilisateur » dans des variables mal protégées.
- Offrez la possibilité de télécharger du contenu indésirable sur le serveur.
- Offre la possibilité de modifier une requête SQL.
- Utiliser un formulaire externe pour manipuler des données dites privées

## **V.2 Fonctionnalités utilisateur**

- L'application doit permettre à un utilisateur de s'inscrire en demandant au moins une adresse e-mail valide, un nom d'utilisateur et un mot de passe avec au moins un niveau minimum de complexité.
- A la fin du processus d'inscription, l'utilisateur doit confirmer son compte via un lien unique envoyé à l'adresse email renseignée dans le formulaire d'inscription.
- L'utilisateur doit alors pouvoir se connecter à votre application, grâce à son identifiant et son mot de passe. Il devrait également pouvoir demander à l'application d'envoyer un mail de réinitialisation de mot de passe, s'il oublie son mot de passe.
- L'utilisateur doit pouvoir se déconnecter en un clic à tout moment sur n'importe quelle page.
- Une fois connecté, un utilisateur doit modifier son identifiant, son adresse mail ou son mot de passe.

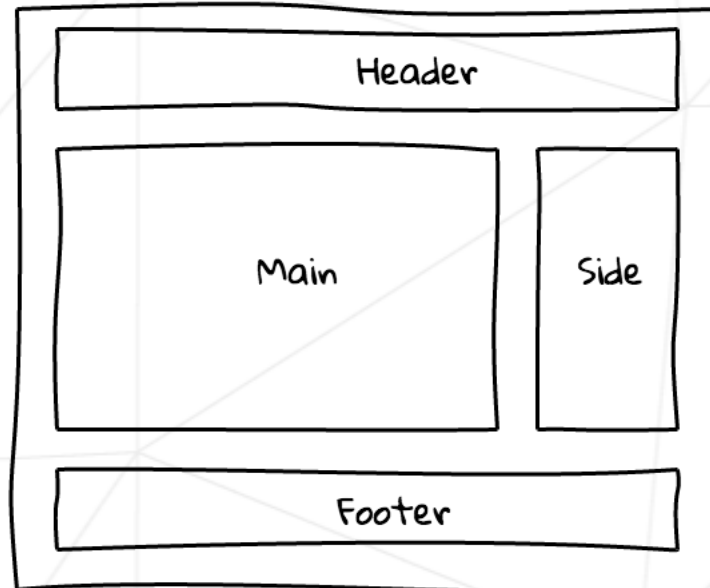
## **Fonctionnalités de la galerie V.3**

- Cette partie est publique et doit afficher toutes les images éditées par tous les utilisateurs, classées par date de création. Il doit également permettre (uniquement) à un utilisateur connecté de les aimer et/ou de les commenter.
- Lorsqu'une image reçoit un nouveau commentaire, l'auteur de l'image doit être averti par email. Cette préférence doit être définie comme vraie 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 Fonctionnalités d'édition

Figure V.1 : Juste une idée de mise en page pour la page d'édition



Cette partie doit être accessible uniquement aux utilisateurs authentifiés/connectés et rejeter doucement tous les autres utilisateurs qui tentent d'y accéder sans être connectés avec succès.

Cette page doit contenir 2 sections :

- Une section principale contenant l'aperçu de la webcam de l'utilisateur, la liste des images superposables et un bouton permettant de capturer une image.
- Une section latérale affichant des vignettes de toutes les photos prises précédemment.

Votre mise en page devrait normalement ressembler à celle de la figure V.1.

- Les images superposables doivent être sélectionnables et le bouton permettant de prendre la photo doit être inactif (non cliquable) tant qu'aucune image superposable n'a été sélectionnée.
- La création de l'image finale (donc entre autres la superposition des deux images) doit se faire côté serveur.
- Étant donné que tout le monde ne possède pas de webcam, vous devez autoriser le téléchargement d'une image utilisateur au lieu d'en capturer une avec la webcam.
- L'utilisateur doit pouvoir supprimer ses images modifiées, mais uniquement ses créations, pas celles des autres utilisateurs.

## V.5 Contraintes et éléments obligatoires

En résumé, votre application Über doit respecter les choix technologiques suivants :

- Langues autorisées :
  - **[Serveur]** Tout (limité à la bibliothèque standard PHP)
  - **[Client]** HTML - CSS - JavaScript (uniquement avec l'API native du navigateur)
- Frameworks autorisés :
  - **[Serveur]** N'importe lequel (jusqu'à la bibliothèque standard PHP)
  - **[Client]** Frameworks CSS tolérés, sauf s'ils ajoutent du JavaScript interdit.

Votre projet doit contenir impérativement :

- Un (ou plusieurs) conteneur pour déployer votre site avec une seule commande. tout ce qui est équivalent à docker-compose est ok.

# Chapitre VI

## Partie bonus

Si la partie requise est entièrement et parfaitement réalisée, vous pouvez ajouter le bonus que vous souhaitez ; Ils seront évalués par vos évaluateurs. Vous devez cependant toujours respecter les exigences des parties bonus (c'est-à-dire que le traitement des images doit être effectué côté serveur).

Si vous manquez d'inspiration, voici quelques pistes :

- "AJAXify » échange avec le serveur.
- Proposez un aperçu en direct du résultat édité, directement sur l'aperçu de la webcam. Il convient de noter que c'est beaucoup plus simple qu'il n'y paraît.
- Faites une pagination infinie de la partie galerie du site.
- Offrir la possibilité à un utilisateur de partager ses images sur les réseaux sociaux.
- Créez un GIF animé.



La partie bonus ne sera évaluée que si la partie obligatoire est PARFAITE. Parfait signifie que la partie obligatoire a été intégralement réalisée et fonctionne sans dysfonctionnement. Si vous n'avez pas réussi TOUS les conditions obligatoires, votre partie bonus ne sera pas du tout évaluée.

# Chapitre VII

## Soumission et évaluation par les pairs

Remettez votre devoir dans votre Git référentiel comme d'habitude. Seuls les travaux réalisés à l'intérieur de votre référentiel seront évalués lors de la soutenance. N'hésitez pas à vérifier les noms de vos dossiers et fichiers pour vous assurer qu'ils sont corrects.