

# Numérique et Sciences Informatiques 1<sup>re</sup>

## Projet : Traitement d'Images sur le web

### Introduction

Dans ce projet, il s'agissait de construire une page en HTML, CSS et Javascript (JS), accessible en ligne via une adresse web, et contenant une application de traitement d'images : l'utilisateur sélectionne une image de son choix depuis son ordinateur ou parmi celles proposées, et choisit différents filtres à lui appliquer (changement de couleur, flou, luminosité, etc.). L'application applique les filtres et affiche le résultat. Quand il a terminé, l'utilisateur peut télécharger l'image modifiée sur son ordinateur. L'interface sera ergonomique et facile d'utilisation ; un guide d'utilisation permettra d'informer sur le fonctionnement de l'outil.

Au départ, j'ai choisi ce sujet car, après avoir géré mon blog pendant plusieurs années, j'avais maintenant envie de coder mes propres pages web et les publier en ligne. De plus, j'avais déjà quelques bases en HTML et CSS, ce qui rendait plus facile la réalisation de ce projet.

Je commencerai par définir mon projet, puis je vous décrirai ses différentes étapes de réalisation avant de m'intéresser aux résultats et à l'analyse.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="styles.css">
    <title>HugTouch' - Retouche d'image gratuit</title>
    <link rel="icon" href="Images/Logo.png" type="image/x-icon">
  </head>
  <body>
    <header>
      <div>
        <h1>HugTouch' - Retouche d'images gratuit</h1>
        <nav id="nav">
          <ul>
            <li><a href="#" onclick="dispEditor();">Editeur</a></li>
            <li><a href="#" onclick="dispGuide();">Guide d'utilisation</a></li>
            <li><a></a></li>
          </ul>
        </nav>
      </div>
    </body>
  </html>
```

Code HTML de mon outil de traitement d'images ↑

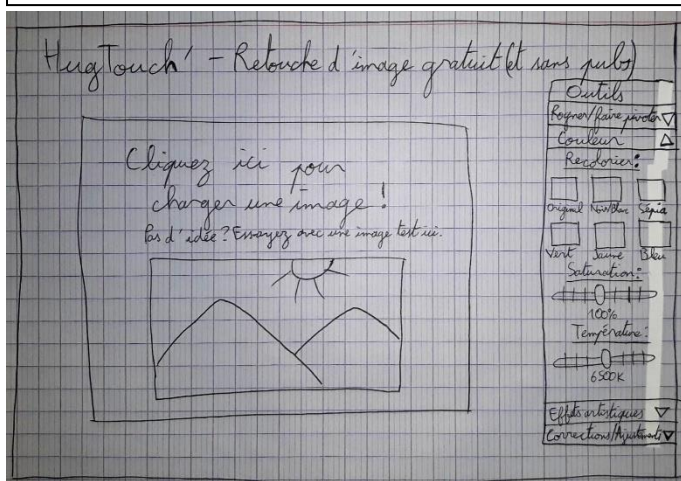
### I. Définition de mon projet

Mon ambition était de concevoir un outil proposant de nombreuses fonctionnalités de traitement d'images, telles que l'application de filtres (niveau de gris, luminosité, ...) mais aussi des transformations (rotation, inversion d'image...), voire d'autres fonctionnalités (arrondir les bords de l'image, ...). La page serait responsive, conviviale et optimisée pour l'utilisateur. Elle serait ensuite publiée sur un serveur web et accessible via une adresse web.

J'ai décidé de mettre en œuvre cette solution, en limitant les fonctionnalités à l'application de filtres, car les outils imaginés se sont révélés plus difficiles à réaliser que prévu.

J'ai utilisé Adobe Brackets comme environnement de développement afin de coder la page web, car

Premier schéma de l'interface utilisateur ↓



c'est un logiciel facile d'utilisation et gratuit. Pour publier mon site web sur internet, j'ai choisi de mettre en place un serveur Linux avec un Raspberry Pi 4 que je possédais déjà. Afin de gérer mes échéances et les tâches à réaliser, mon choix s'est tourné vers Microsoft Outlook, qui est un logiciel de courrier, calendrier et tâches compris dans ma suite Office offerte par la région. L'utilisation de Filezilla, qui est un logiciel gratuit, m'a permis d'établir une connexion FTP avec mon serveur web distant.

Ainsi, je me suis fixé pour objectifs de réaliser des croquis schématiques de l'interface, de créer le squelette de la page web en HTML, puis de coder une page fonctionnelle en JS afin que l'utilisateur puisse

uploader, modifier et télécharger une image. Il me restait à rendre l'interface conviviale et optimisée avec du CSS, puis de mettre en place le serveur web afin de mettre le code en ligne. Pour finir, il me fallait optimiser les outils existants et ajouter de nouvelles fonctionnalités.

En respectant les délais impartis, je devais faire en sorte que le site soit complet, rapide et qu'il comporte le moins de bugs possibles.

## II. Réalisation du projet

J'ai d'abord dû mettre en place un dossier de travail dans lequel j'ai créé les différents fichiers nécessaires à la réalisation de l'application web. J'ai ainsi mis en place un environnement de développement sur mon PC.

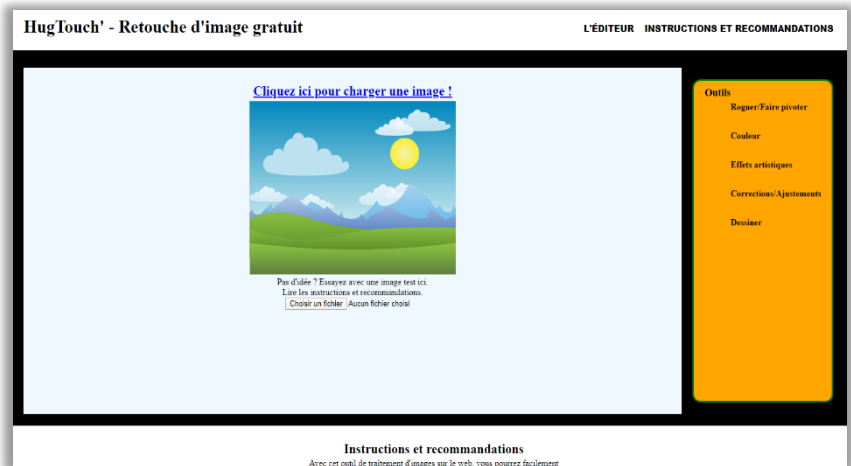
J'ai commencé par coder la page en HTML et CSS, car je ne savais pas encore programmer en JS. En effet, je n'avais pas encore pu suivre les cours correspondants. J'ai tout de même pu coder l'organisation et l'apparence globale des éléments avec Brackets, que je maîtrisais déjà.

L'un des premiers problèmes que j'ai rencontrés est l'élément <header> que je devais faire flotter en haut de la page, afin que le titre et la barre de navigation soient fixés en haut de la page, même si l'utilisateur descend. Je l'ai résolu en utilisant la déclaration CSS « position : fixed; ». Il a aussi fallu définir la superposition des éléments (afficher le header au-dessus du menu par exemple) : j'ai alors utilisé la propriété « z-index » qui permet de déterminer l'empilement des objets affichés sur la page.

Il me restait ensuite à implanter du code JS : j'ai fait de nombreuses recherches sur l'élément canvas ainsi que l'application de filtres en CSS et JS sur une image. J'ai ainsi pu comprendre le fonctionnement du canvas et son intégration au web.

Il a fallu ajouter l'interface de réglages des filtres. J'ai utilisé des boutons +/- ainsi que des curseurs réglables afin d'appliquer des filtres sur mon canvas.

J'ai continué en réalisant les services d'upload de l'image sur le canvas et de téléchargement du canvas modifié.



↑ La page web codée uniquement en HTML et CSS

```
case 'sepia':
    sepia.value = parseInt(sepia.value) + val;
    sepiaTxt.innerHTML=sepia.value + "%";
    break;
case 'hueRotate':
    hueRotate.value = parseInt(hueRotate.value) + val;
    hueRotateTxt.innerHTML=hueRotate.value + "°";
    break;
}
ctx.filter = `blur(${blurring.value}px) brightness(${brightness.value}%)
contrast(${contrast.value}%) invert(${invert.value}%) saturate(${saturate.value}%)
sepia(${sepia.value}%) grayscale(${grayscale.value}%) hue-rotate(${hueRotate.value}deg)`;
ctx.drawImage(img, 0, 0, canvas.width, canvas.height);
```

Utilisation de `canvas.getContext('2d').filter` ↑

Pour coder les filtres, j'ai commencé par m'intéresser à la transformation des pixels sur le canvas avec les composantes RGBA. J'ai finalement choisi la méthode « `canvas.getContext('2d').filter` » plus simple à utiliser qui permet d'utiliser des filtres prédéfinis et de régler leur intensité (ex : `filter="blur(2px)"`) qui floute de 2 pixels l'image).

J'ai créé une fonction qui permet de visualiser les modifications d'un filtre, appelée lors d'événements (oninput, onchange et onclick) avec les éléments HTML correspondants (input, button) ; j'ai appliqué ce même principe avec les fonctions annuler et retourner à la valeur par défaut réalisées peu après.

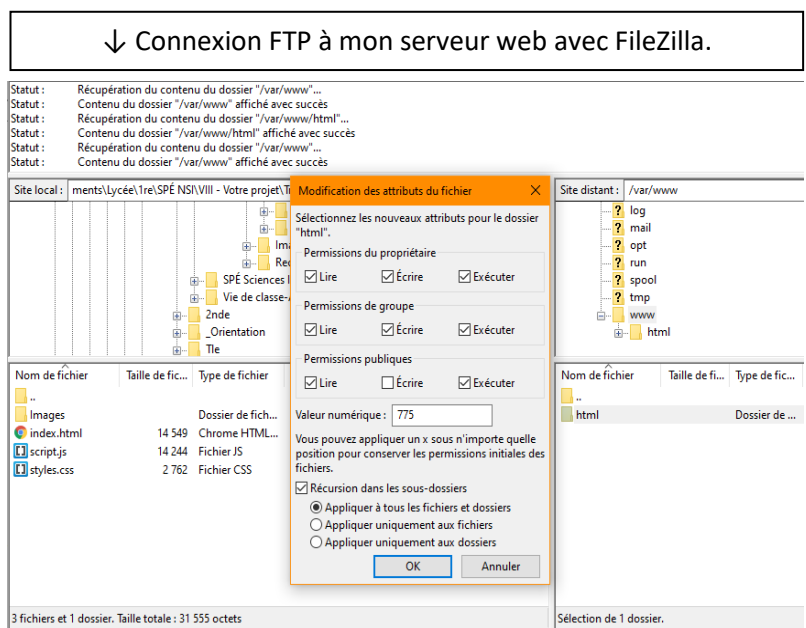
Par la suite, j'ai programmé un système de pile qui permet d'enregistrer toute action de l'utilisateur quant à l'application de filtres, et un autre système de pile pour les valeurs. Un bouton « Annuler » permet à l'utilisateur de revenir sur ses modifications grâce à une fonction JS. Une tâche qui s'est révélée être plus dure que prévu puisqu'il fallait combiner l'utilisation de boutons et de curseurs réglables pour modifier les filtres, et qu'il fallait pouvoir faire revenir les filtres à leur valeur précédente. Après de nombreuses heures de recherches et de codage, j'ai résolu ce problème en rappelant la précédente valeur d'un même filtre avant d'appliquer la suivante, puis dans le cas d'un retour, en appliquant l'ancienne valeur et en supprimant la dernière valeur du filtre précédent de la file.

C'est à partir de là que j'ai mis en place le serveur web : sur mon Raspberry, j'ai installé Raspbian, avant de configurer une adresse IP fixe, la connexion SSH, et d'installer le serveur Apache. Ensuite, la box internet a été configurée pour rediriger les connexions entrantes vers le Raspberry, et les noms de domaines [www.hugjournal.fr](http://www.hugjournal.fr) et [www.cracahouetes.fr](http://www.cracahouetes.fr) ont été associées à l'adresse IP de ma box. Je pouvais ainsi utiliser Filezilla pour copier mes fichiers directement sur le serveur, et le site était accessible par tous ceux connaissant son adresse.

Pour finir, j'ai amélioré l'interface en ajoutant des boutons pour appliquer la valeur par défaut du filtre à l'aide d'une fonction JS, et en rendant l'interface utilisateur plus ergonomique.

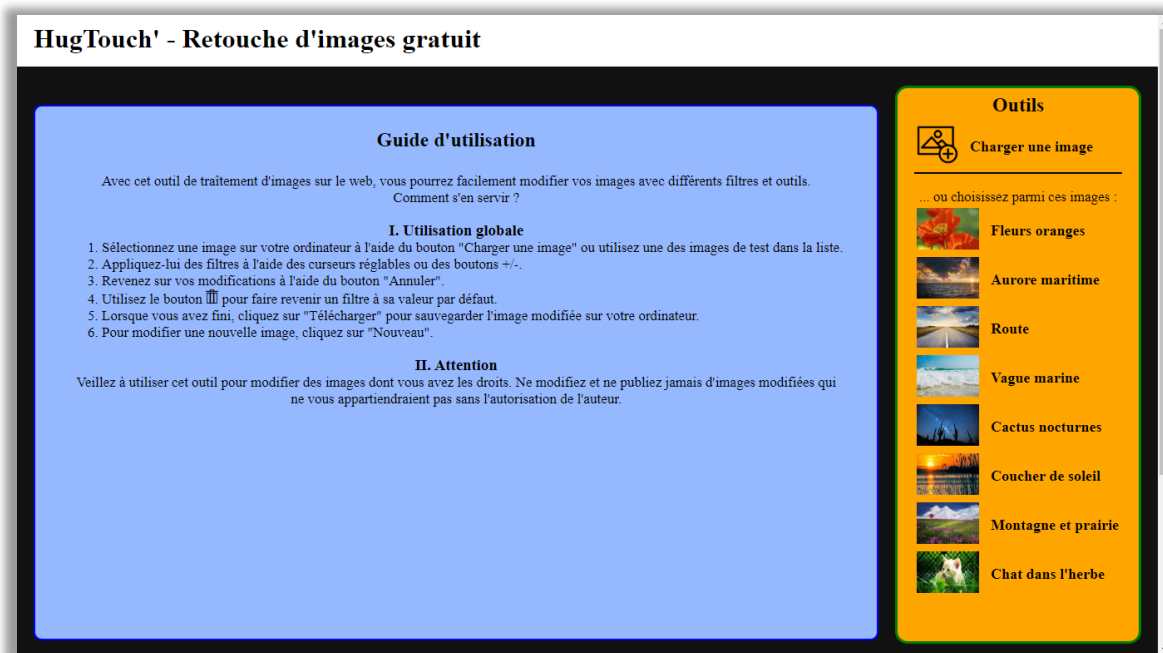
Je me suis connecté sur la page depuis Google Chrome et Mozilla Firefox sur PC et constaté qu'elle fonctionnait correctement.

Le site était désormais achevé.



### III. Résultats et analyse

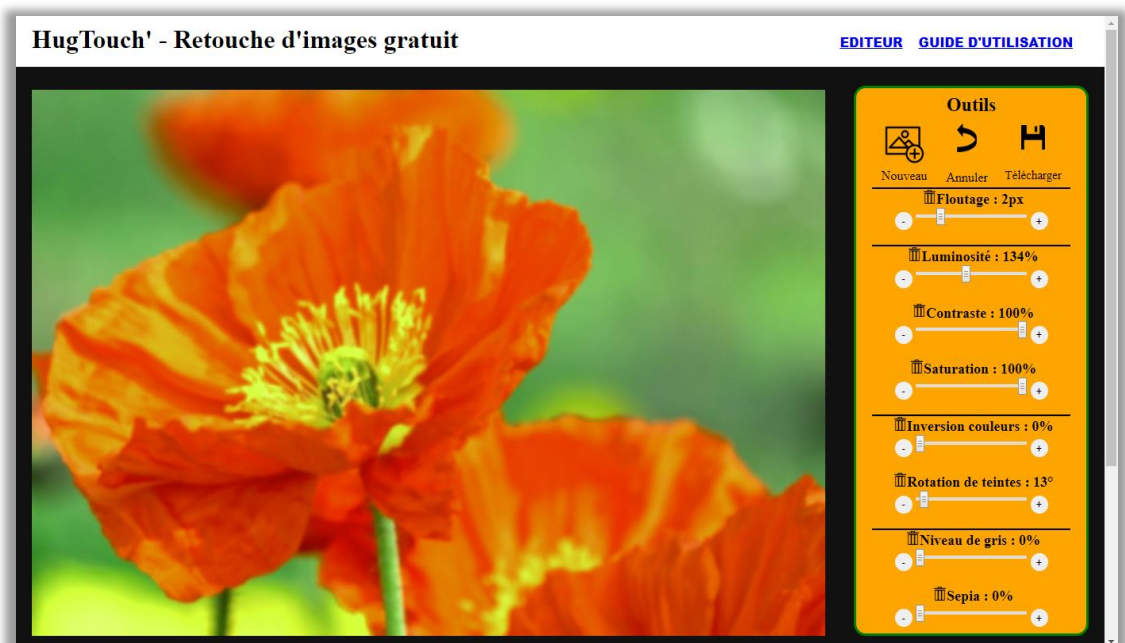
Finalement, lorsqu'on tape [www.cracahouetes.fr](http://www.cracahouetes.fr) ou [www.hugjournal.fr](http://www.hugjournal.fr), on arrive sur une page web fonctionnelle. La première chose que voit l'utilisateur est un guide permettant l'utilisation correcte de l'outil de traitement d'image ; à droite, un menu pour sélectionner une image depuis son ordinateur ou parmi celles proposées ; en haut, la barre de titre avec le nom de l'outil « HugTouch' – retouche d'images gratuit » ; en bas, le pied de page avec la date et le nom du créateur du site (moi !).



← Page d'accueil  
de mon outil de  
traitement  
d'images sur le  
web

Lorsqu'une image est sélectionnée, le guide disparaît pour laisser place à l'image et le menu se transforme afin d'afficher les options d'éditations et les filtres. Le bouton « Nouveau » permet de modifier une nouvelle image, « Annuler » de revenir sur l'application d'un filtre et « Télécharger » de sauvegarder l'image modifiée sur son ordinateur. En dessous sont présents différents filtres avec pour chacun un curseur réglable et des boutons +/- permettant de régler l'intensité du filtre. Une icône « corbeille » à côté de chaque filtre permettra de le ramener à sa valeur par défaut. L'utilisateur peut donc modifier les valeurs des filtres, voir l'aperçu en direct, et annuler ses modifications. Dans la barre de titre est également apparu un lien « Guide d'utilisation » qui permet d'afficher de nouveau le guide, qui peut ensuite être fermé par clic sur la croix rouge en haut à droite du guide.

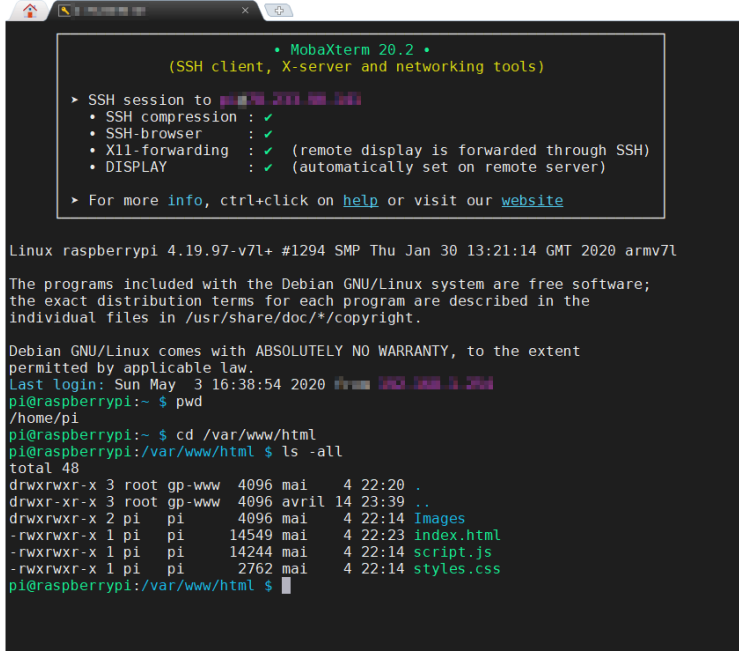
Edition d'une  
image grâce à ma  
page web. Ici,  
utilisation d'une  
des images de  
test avec  
l'application de  
plusieurs filtres.  
→



Je suis ravi d'avoir codé un outil fonctionnel à l'allure quasi-professionnelle, publié sur un serveur web que j'ai mis en place, le tout en quelques mois seulement. Je respecte les objectifs décrits dans le sujet, à savoir que l'utilisateur peut modifier une image en superposant différents filtres avant de la télécharger. J'aurais aimé avoir le temps d'optimiser la page afin qu'elle soit complètement responsive et donc consultable sur des tailles d'écrans très variables, ce que je n'ai pas pu faire entièrement en raison des délais impartis.

La programmation de cette page web m'a permis d'apprendre à manipuler HTML, CSS et JS, et m'a donné l'occasion de découvrir différentes techniques de créations d'une page web fonctionnelle, interactive, ergonomique et agréable à consulter et à utiliser. J'ai notamment appris les différentes techniques pour utiliser l'élément Canvas : y implanter une image, modifier les pixels, télécharger le rendu en image, ...

J'ai également pu manipuler l'OS Linux, en ligne de commande et en interface graphique, afin de mettre en place un serveur web où stocker ma page ; ainsi que le « réseau » en paramétrant ma box internet afin de rediriger les connexions entrantes vers mon serveur.



```

MobaXterm 20.2
(SSH client, X-server and networking tools)

> SSH session to [redacted]
  • SSH compression : ✓
  • SSH-browser      : ✓
  • X11-forwarding   : ✓ (remote display is forwarded through SSH)
  • DISPLAY          : ✓ (automatically set on remote server)

> For more info, ctrl+click on help or visit our website

Linux raspberrypi 4.19.97-v7l+ #1294 SMP Thu Jan 30 13:21:14 GMT 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun May 3 16:38:54 2020 from [redacted]
pi@raspberrypi:~$ pwd
/home/pi
pi@raspberrypi:~$ cd /var/www/html
pi@raspberrypi:/var/www/html$ ls -all
total 48
drwxrwxr-x 3 root gp-ww 4096 mai 4 22:20 .
drwxr-xr-x 3 root gp-ww 4096 avril 14 23:39 ..
drwxrwxr-x 2 pi pi 4096 mai 4 22:14 Images
-rwxrwxr-x 1 pi pi 14549 mai 4 22:23 index.html
-rwxrwxr-x 1 pi pi 14244 mai 4 22:14 script.js
-rwxrwxr-x 1 pi pi 2762 mai 4 22:14 styles.css
pi@raspberrypi:/var/www/html$

```

↑ Connexion au Raspberry Pi en SSH via MobaXterm.

## Conclusion

J'ai vraiment aimé réaliser cette page web, car j'ai pu faire un outil fonctionnel et esthétique. De plus, ce projet était vraiment très formateur puisque cela m'a permis de coder de nombreuses heures en HTML, CSS et Javascript.

J'envisage de devenir développeur web, ce projet m'a donc permis de bien étendre et consolider mes connaissances en programmation web.

## Bibliographie / Sitographie / Logithèque

J'ai utilisé plusieurs sites de manière récurrente afin de trouver les réponses à mes recherches. Parmi eux : <https://stackoverflow.com/> ; <https://developer.mozilla.org/fr/> ; <https://www.w3schools.com/> ; <https://www.developpez.net/> ; <https://openclassrooms.com/> ; <https://www.alsacreations.com/> .



Mon Raspberry Pi 4 Model B ↑

J'ai utilisé Adobe Brackets comme environnement de développement sur PC, Filezilla afin d'établir une connexion FTP avec mon serveur, MobaXterm pour établir une connexion SSH avec mon Raspberry, ainsi que plusieurs navigateurs (Chrome, Firefox, ...) pour visualiser ma page pendant le développement.

Je voudrais remercier ma famille et mes proches qui m'ont soutenu et conseillé durant la réalisation de ce projet.