

ENSEIGNEMENT TECHNIQUE DE QUALIFICATION
TECHNICIEN(NE) EN INFORMATIQUE

TRAVAIL DE FIN D'ETUDE

Sauvageland.fr

Fontaine Emmerick

ANNEE SCOLAIRE 2021-2022

Professeurs responsables

Q. CARPENTIER ET A. VANOVERVELT

Remerciements

Mes remerciements les plus sincères à toutes les personnes qui ont contribué, de près ou de loin, à l'élaboration de ce rapport, ainsi qu'à la réussite de ma formation. Ce travail de fin d'études n'aurait pas pu voir le jour sans l'aide, au combien précieux, de nombreuses personnes. C'est pourquoi, je tiens à remercier toutes celles et ceux qui m'ont aidé et encouragé tout au long de l'élaboration de ce mémoire.

Je voudrais remercier tout particulièrement :

- Monsieur **Carpentier** ainsi que Monsieur **VANOVERVELT** mes professeurs qui m'ont beaucoup appris toutes ces années et qui m'ont permis de repousser mes limites.
- Ma famille et mes amis qui m'ont toujours encouragée dans mes choix personnels. Ils ont toujours été là pour moi afin de m'aider à franchir chaque étape de ces années d'études. Ils m'ont également soutenue dans les moments heureux comme dans les moments plus difficiles.
- Monsieur **Bruyere** mon maître de stage qui m'a permis de découvrir le monde professionnel ainsi que de nouvelles connaissances informatiques qui m'ont apporté un nouveau point de vue.

Table des matières

Remerciements	2
Introduction.....	4
Logiciel utilisé.....	4
Visual Studio Code	4
GitHub	5
Matériel utilisé	8
Raspberry PI 4 B+	8
Analyse des besoin.....	10
Développement	11
Langage (Python)	11
Qu'est-ce que Python ?	11
Qu'est-ce que Flask ?.....	12
Les croquis.....	14

Introduction

Dans le but de mes études j'ai réalisé un site web pour un serveur de jeu Minecraft.

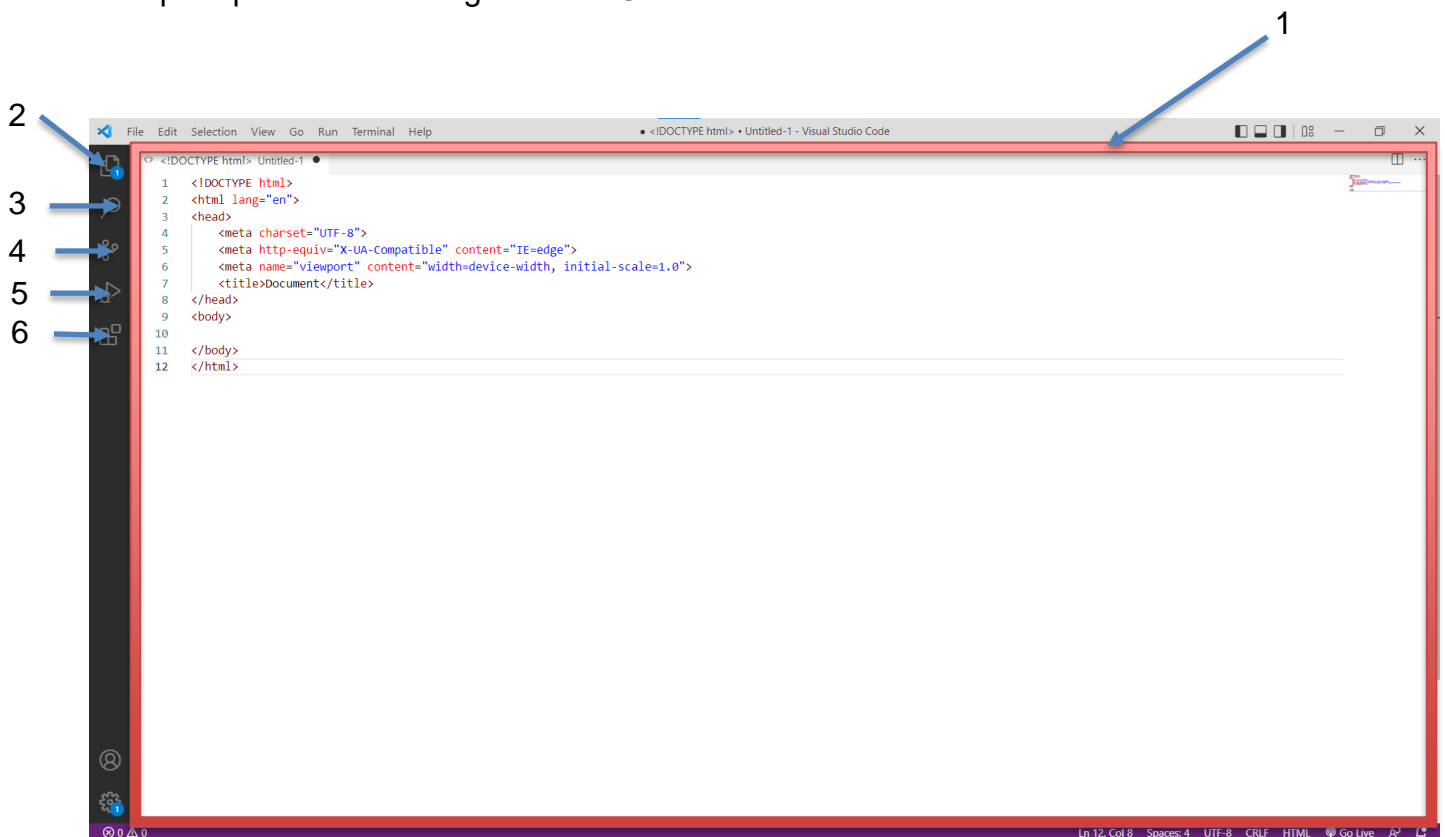
C'est un sujet qui me plait car j'aime la programmation web ainsi que la gestion de serveur et les jeux vidéo, ce projet me permet de réunir c'est trois points.

Celui-ci n'est pas uniquement un projet pour l'école mais un projet personnel qui tend à évoluer dans le futur. J'ai pour objectif de réaliser un magasin en ligne (shop) ainsi qu'un système de support ce basant sur le system de ticket, une page de profile personnaliser en fonction du pseudo Minecraft et un système d'identification, le tout sous python avec l'extension web flask et hébergé temporairement sur raspberry pi.

Logiciel utilisé

Visual Studio Code

VS (Visual Studio Code) est un éditeur multicode développé par Microsoft, il possédé de multiple extension pour aider aux codages ou des fonctionnalités pratique comme l'intégration de GitHub et bien d'autres.



1. Zones de texte, VS intégré emmet, des raccourcis qui font gagner du temps lors de la programmation.
2. Explorateur de fichier, cet onglet permet de voir les dossiers et fichiers utilisés ce qui permet de faciliter la navigation.
3. La recherche, cet onglet permet de pouvoir rechercher des lettres/mots dans un fichier et de pouvoir les remplacer par d'autres.
4. Le source contrôle, cet onglet permet de pouvoir utiliser GitHub grâce à une interface graphique (nous allons y revenir plus tard).
5. Debug menu, cet onglet permet d'exécuter le code en mode debug ce qui facilite la résolution des erreurs de codage.
6. Les extensions, cet onglet permet d'installer des multitudes d'extensions diverses et variées.



GitHub est un service web qui permet d'héberger des fichiers/dossiers avec un système d'historique et de backup.

Voici ma liste des répertoires personnels

Et voici l'endroit où est stocker mon TFE

The screenshot shows a GitHub repository page for 'sauvage2603 / TFE'. The repository is public and has 1 branch and 0 tags. The file list includes folders like 'note', 'public', 'static', and 'templates', and files like '.gitignore', 'README.md', 'Rapport de stage.docx', and various Python files. The right sidebar shows repository statistics: 0 stars, 1 watching, 0 forks, and a language distribution chart for HTML (61.8%), Python (27.3%), and CSS (10.9%).

File/Folder	Description	Commit Time
note	Version final TFE	13 hours ago
public	version casi definitiva	2 days ago
static	Version final TFE	13 hours ago
templates	Version final TFE	13 hours ago
.gitignore	Initial commit	28 days ago
README.md	Initial commit	28 days ago
Rapport de stage.docx	Version final TFE	13 hours ago
__init__.py	version casi definitiva	2 days ago
auth.py	base	28 days ago
db.sqlite	fini	19 hours ago
main.py	Version final TFE	13 hours ago
models.py	fini	19 hours ago

Pour utiliser GitHub il faut avant tout le télécharger sur <https://desktop.github.com/> pour ensuite ouvrir son terminal et exécuter c'est quelques lignes de commande.

Configurer son nom pour l'inscrire dans les fichiers lors de l'exportage des fichiers.

```
git config --global user.name "aquelito"
```

Configurer son adresse mail.

```
git config --global user.email "axel@aquelito.fr"
```

Initialiser un répertoire sur l'ordinateur.

```
git init <noms du répertoire>
```

Pour ajouter tout fichier du répertoire au projet.

```
git add *
```

Permet d'enregistrer les changements effectués du code dans l'historique git.

```
git commit
```

Pour push les commits à votre dépôt distant sur de la branche spécifiée.

```
git push <remote> <branche>
```

Pour chercher et merger les modifications d'une branche du serveur git et l'applique à la branche sur laquelle vous vous trouvez.

```
git pull <branche>
```

Matériel utilisé

Raspberry Pi 4 B+

Le Raspberry pi est un nano-ordinateur de la taille d'une carte de crédit conçu par des professeurs de l'université de Cambridge (école anglaise) l'avantage du Raspberry pi est sa faible consommation, son prix attractif (du moins à l'époque) et ces pin GPIO qui ne me seront d'aucune utilité pour ce projet.

Comme tout ordinateur le Raspberry a besoin d'un os et pour installer celui-ci on utilise Pi Imager





Operating system : on sélectionne l'os dans notre cas Raspbian, c'est l'os fait exprès pour le Raspberry sous base de linux Debian.

SD Card : On sélectionné le périphérique sur le quelle l'os sera installer.

Après avoir sélectionné tout ceci on a puis qu'a appuyé sur Write.

Analyse des besoin.

- La première chose a pensé pour un serveur Minecraft et la facilité de connexion au serveur, Minecraft utilise un système d'IP pour se connecter au serveur. Cette IP doit être visible dès la première page.
- La deuxième chose est la rentabilité du serveur, la première chose qui engendre des fonds est le shop en ligne. Il nous faut donc un shop qui permet d'acheter des bonus comme des grades sur Minecraft avec une méthode sécurisé.
- La troisième chose est un support qui permet de résoudre les problèmes des utilisateurs ou de les renseigné sur divers choses. Je vais donc créer un système de ticket pour que les administrateurs puissent pouvoir répondre aux demandes.
- Le quatrième besoin est un système d'identification qui accompagnera le support et le shop.

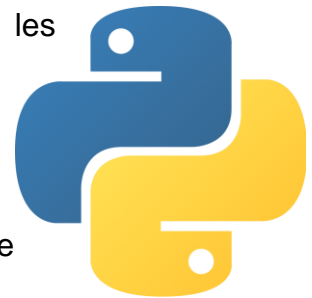
Développement

Langage

Pour la création d'un site web cela va de soi l'utilisation d'HTML est primordial ainsi que le css pour le designer des pages, j'ai rajouté à sa l'utilisation de python pour l'exécution des commandes serveur. Pourquoi avoir choisi Python ? J'aurai très bien pu choisir un langage comme PHP, mais mon choix c'est porté sur un langage un peu plus moderne, j'ai utilisé une extension se nommant Flask. Je voulais découvrir de nouvelle manière de programmer des sites web et deux plus améliorer ma connaissance de Python celui -ci fut le bon compromis.

Qu'est-ce que Python ?

Python est un langage de programmation créée en 1991, les multiples plates-formes il fonctionne autant sur ordinateur, que sur smartphone, que ce soit sur Windows ou Linux. Celui-ci est de plus libres de droits. Il a encore porté également de nombreuses extensions élargissant les possibilités de création. Une phrase résume bien Python « il ne faut pas dire que peut-on faire avec Python mais plutôt que ne peut-on pas à faire avec Python ».



Qu'est-ce que Flask ?

Flask est une extension de python créée en 2010 concurrent

PHP on dit de lui qu'il est un microframework car il est très



léger. Cette extension possède également ses propres librairies, voici des plus connus et utilisés pour mon TFE.

- Flask-Login :

Cette library permet de gérer les inscriptions ainsi que des connexions et de fournir des raccourcis pour la programmation.

- Flask-mail :

Cette librairie permet d'envoyer des e-mails comme indiqué dans son nom, je n'ai malheureusement pas pu l'utiliser car j'ai rencontré des problèmes liés aux adresses email Google mais sera implanté dans un futur proche.

- Flask sqlalchemy :

Cette librairie permet d'utiliser un ORM pour la gestion de mes bases de données ce qui facilite grandement l'utilisation de celle-ci. Elle permet également de pouvoir changer de langage de base de données rapidement en quelques lignes de commandes.

Leur utilisation sera expliquée plus en détail plus tard.

Qu'est-ce que HTML ?

HTML (HyperText Markup Language) a été créée en 1993. Celui-ci permet de décrire de l'hypertexte (qui signifie un document incorporant un système de renvois vers un autre document se nommant les liens), de structurer schématiquement la page web ainsi que de créer des formulaires de saisie se nommant input qui permet l'acquisition des données que l'utilisateur aura enregistré et pour finir celui-ci permet l'utilisation des multimédias tels que des images et des vidéos.



Qu'est-ce que CSS ?

CSS (Cascading style sheets) est un langage informatique utilisé dans le design créé en 1996. Celui-ci permet d'effectuer des actions comme définir un fond d'écran, changer la police, la taille, voire même la couleur d'un texte mais il permet aussi de définir la taille et la position de tous les éléments affichés sur l'écran. Je ne pourrai pas citer toutes les fonctionnalités de celui-ci tellement il y en a.

Les croquis

Après avoir fait une analyse des besoins ainsi que réfléchit au langage de programmation, nous pouvons nous attaquer au premier croquis du designer de notre site web, pour ce faire j'ai utilisé le site draw.io.

Les voici.



Support

[Jouer](#)[Nouveauté](#)[Support](#)[Boutique](#)[Forum](#)[Profile](#)

Support

N° ticket	Dernier réponse	Etat	Sujet	Créer un ticket
00001	modérateur	Résoluit	Problème connexion	Voir
00002	Vous	En attente	Aide	Voir
0003	modérateur	Fermer	Troll	Voir

◀ 1 , 2,...,15 ▶

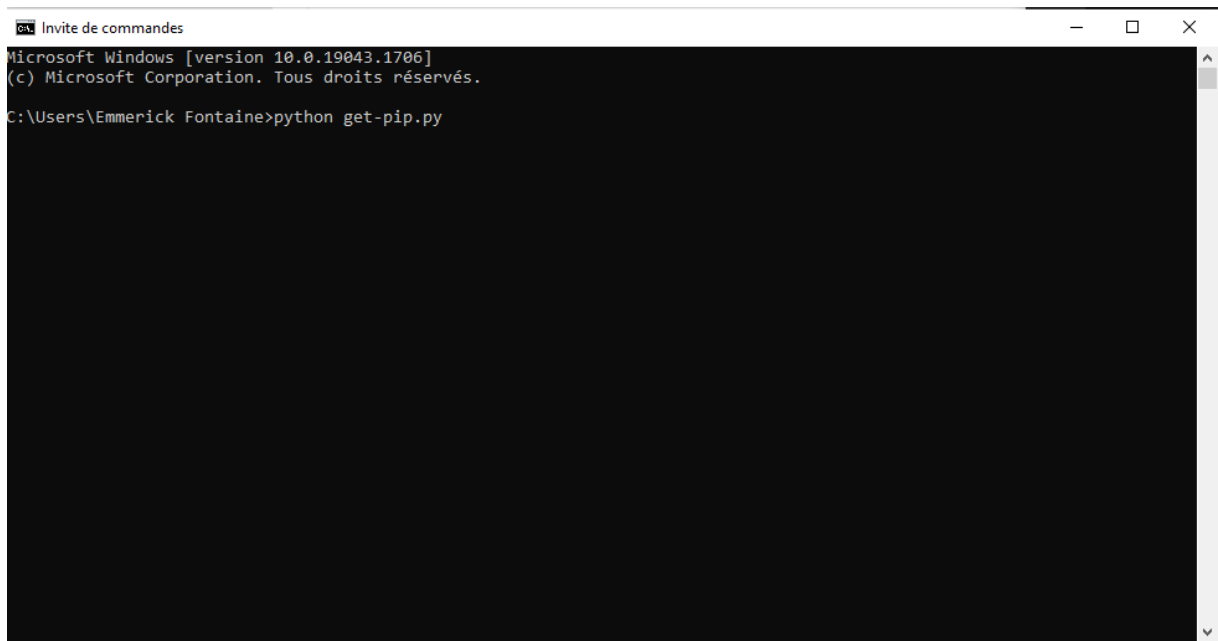


Les schémas n'étaient bien sûr pas le design définitif mais celui-ci m'apporta une direction à suivre pour la création de ces pages.

Mise en place de l'environnement

Dans un premier temps j'installe python en me rendant sur python.org, c'est un simple exécutable à lancé.

Une fois python installé j'installe une extension se nommant PIP, celui-ci permet d'installer d'autres extensions très rapidement et facilement et s'installe via le cmd (l'invite de commande) de Windows.



```
Invite de commandes
Microsoft Windows [version 10.0.19043.1706]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Emmerick Fontaine>python get-pip.py
```

PIP étant installé je peux maintenant installer Flask et les extensions utilisées dans le projet grâce à ces commandes aussi utilisées dans le cmd.

```
pip install Flask
```

```
pip install flask-login
```

```
pip install Flask-SQLAlchemy
```

```
pip install Flask-Mail
```

