



Édition 2023

**DOSSIER DE CANDIDATURE
PRÉSENTATION DU PROJET**



NOM DU PROJET : **OBSEM**

> **PRÉSENTATION GÉNÉRALE :**

• Le projet **OBSEM** consiste en un grand nombre d'interactions client-serveur à but ludique et éducatif. Il se décompose en deux parties :

→ La partie **Questionnaire** qui présente trois questionnaires sur un thème parmi : « Pokémon », « Culture Générale » et « NSI ».

→ La partie **Jeu** composée du jeu final (un catch-game) et du classement des autres joueurs.

L'objectif est de divertir l'utilisateur et de le tester sur une série de questions puis de le récompenser par un jeu.

• À la base, ce projet était juste un TP donné par notre professeur sur les interactions client-serveur. Puis en programmant le site, nous avons commencé à rajouter des fonctionnalités ci et là et nous avons fini par avoir un rendu assez complet. Nous nous sommes dit qu'il serait dommage de ne pas pousser le projet un peu plus loin et c'est ainsi que le serveur **OBSEM** est né.

L'intérêt majeur du projet réside en fait dans la partie codage du site : elle permet de se mettre dans la condition d'un développeur qui doit coder un site pour lui-même ou pour une autre personne, de A à Z sans utiliser d'intermédiaire et de modèle préconçu. Durant la programmation du site, on se rend alors compte de toutes les difficultés qui peuvent survenir (avoir un beau rendu graphique, récupérer les données de l'utilisateur pour les stocker et les utiliser dans d'autres parties du serveur, trouver des idées originales pour le fonctionnement, etc).

Ainsi on arrive à percevoir la complexité des sites des grandes entreprises et également la difficulté de tout programmer sois-même sans l'aide de personnes spécialisées dans tel ou tel langage de programmation.

• Dans le projet **OBSEM**, nous avons essayé de regrouper un maximum de fonctionnalités pour se rapprocher le plus possible d'un site professionnel et nous avons réussi à :

- Avoir un nombre de bugs et d'erreurs très faible.
- Développer une fonctionnalité qui permet d'envoyer un message par mail via une interface dédiée.
- Avoir un affichage qui s'adapte à peu près à tous types d'appareils (smartphones exclus).
- Créer un système de base de données (assez primitif car format csv) qui permet de traiter les données utilisateurs.
- Développer un système de classement qui permet aux joueurs de se mettre en compétition (au niveau des scores).

En procédant comme ceci, le site devient assez plaisant à visiter et l'expérience de jeu n'en est que plus agréable.

> **ORGANISATION DU TRAVAIL :**

• **Présentation de l'équipe :**

-AYVAZ Semih : Présentateur du projet.

-AUDUREAU Aubin : Chef de projet.

• Répartition des tâches :

Semih: Montage, communication et testeur.

Aubin : Codage, rédaction du dossier et chef de projet.

• Organisation du travail : *Au début du projet, on travaillait surtout durant les séances dédiées au projet en classe de NSI. Puis petit à petit, on a commencé à travailler en dehors des heures de cours en discutant du projet, par messages notamment, et tout s'est enchaîné très vite. Les vacances de février ont permis le développement intensif du projet, et lorsque l'on bloquait sur une question technique, on utilisait des forums (comme stackoverflow, openclassrooms, codeacademy, freecodecamp, ...) ou bien les nouveaux outils d'intelligence artificiel (comme chat-gpt3.5 ou encore DALL·E 2). Mais on réussissait également à surmonter les plus gros problèmes grâce au code des développeurs présents en ligne (sur Github, codeopen) ou grâce à des projets à but non-lucratif (smtpjs pour le mail par exemple). Et c'est ainsi qu'on en est arrivé au stade de production actuel.*

> LES ÉTAPES DU PROJET :

• Première étape : *Pour commencer, le rendu final était juste censé être un TP noté, on était donc parti dans l'optique de faire trois questionnaires sur les mêmes thèmes que ceux cités précédemment. On devait afficher le score des trois quizzes une fois finis puis c'était à peu près tout.*

• Deuxième étape : *Ensuite, on a pensé à faire un jeu après les questionnaires puis un classement. Et pendant le développement de ces fonctionnalités l'idée d'intégrer une interface pour reporter les bugs nous est venue.*

• Troisième étape : *À partir de ce moment là, on a commencé à sérieusement envisager de participer au concours des trophées de la NSI et on s'est attelés à des éléments plus spécifiques comme :*

→ *Un beau rendu (c'est à dire toute la grosse partie CSS).*

→ *L'adaptabilité de l'affichage et la correction des bugs gênants*

(notamment celui qui faisait que l'utilisateur pouvait modifier l'URL pour s'attribuer plus de points dans les questionnaires ou dans le catch-game).

→ *La création d'un logo spécifique pour le site.*

• Quatrième et dernière étape : *Pour finir, on a voulu peaufiner au maximum le rendu et augmenter le volume de contenu disponible sur le serveur. Et c'est ainsi que l'on a ajouté de la musique et deux difficultés supplémentaires pour chaque questionnaire. (Fonctionnalités, qui au moment de la rédaction de ce dossier, sont encore en développement).*

> FONCTIONNEMENT ET OPÉRATIONNALITÉ :

• Avancement du projet :

→ **Terminé :** - Accueil et base de données

- Questionnaires (une difficulté)

- Page pour reporter les bugs

- Jeu et classement

→ **En cours** : - Deux difficultés supplémentaires pour les quizzes
- Musique
→ **Amélioration** : Méthode «post» au lieu de la méthode «get» pour les quizzes et le jeu final.

• **Approches mises en œuvre pour vérifier l'absence de bugs et s'assurer de la facilité d'utilisation du projet :**

→ **Interface dédiée aux bugs qui a permis la résolution de plusieurs d'entre eux via quelques testeurs .**
→ **Relecture du code .**
→ **Utilisation régulière du serveur pour tenter de trouver de nouveaux bugs .**
→ **Test du serveur par des personnes tiers et qui ne s'y connaissent pas en informatique pour avoir des retours sur la facilité d'utilisation.**

• **Difficultés rencontrées et solutions apportées :**

→ **Problème : Difficulté pour envoyer des mails en javascript.**
Solution : Projet libre «smtpJS».

→ **Problème : Diverses difficultés sur la compréhension de nouvelles notions informatique.**
→ **Solution : Documentation de Mozilla Firefox, forums ou chat-gpt3.5 .**

→ **Problème : Difficulté avec l'ergonomie et l'harmonie générale du site.**
→ **Solution : Codeopen.io (codes de développeurs en ligne), DALL-E2 (IA de génération d'image).**

→ **Problème : Aucune connaissances dans la création de musique et de sons.**
→ **Solution : Pixabay.com, Mixkit.com, ... (sites pour télécharger des sons et des musiques libres de droits) et 123apps.com (site pour divers outils de modification sur des fichiers audios et vidéos) .**

> **OUVERTURE :**

• **Idées d'améliorations :**

→ **Laisser les utilisateurs créer leurs quizzes et participer à ceux des autres.**

→ **Faire d'autres jeux (un pour chaque thème de quizz) qui se débloquent lorsque toutes les difficultés d'un quizz ont été faites.**

→ **Améliorer l'ergonomie et le CSS.**

• **Stratégie de diffusion :** **Faire de la pub sur les réseaux sociaux (YouTube, Instagram, Twitter, ..), poster des vidéos et des images du site qui permettent de suivre son évolution au cours des différentes mises à jours. Diversifier les secteurs d'activités du site (ajouter du contenu sur d'autres thèmes) et promouvoir l'aspect ludique du projet.**

• **Analyse critique du résultat :**

→ **Si c'était à refaire, je pense que je changerais l'architecture du site pour vraiment avoir un jeu qui suit une histoire et qui ait un peu de suspens au lieu d'avoir juste des quizzes et un jeu sans trop de relation.**

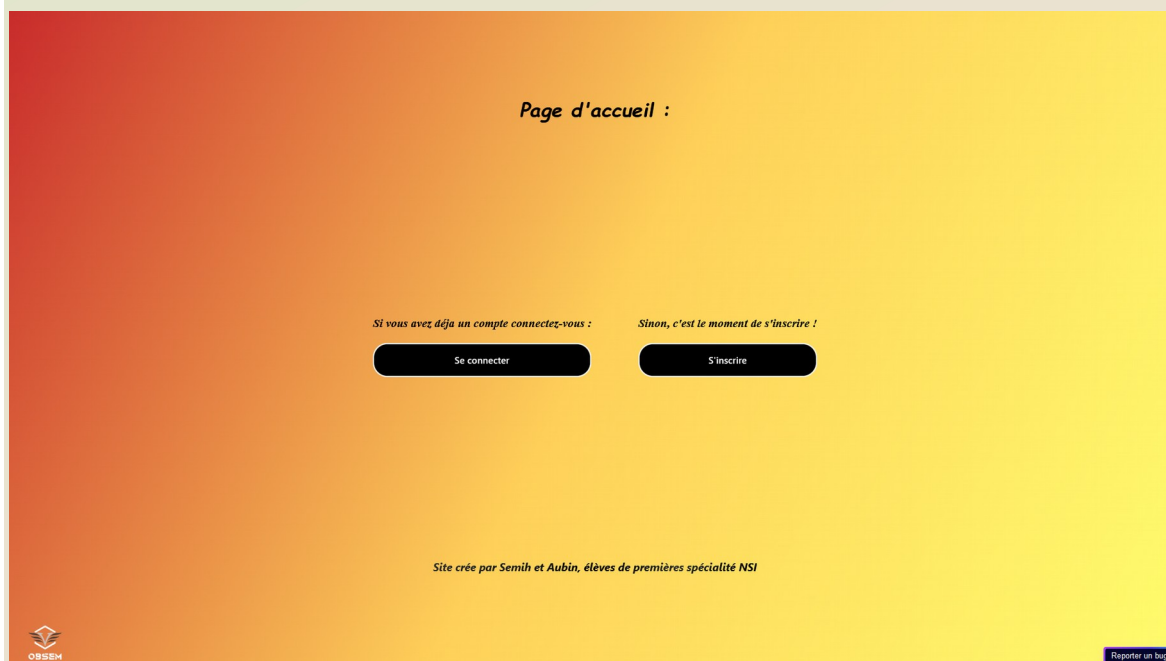
→ **Concernant les détails techniques, je pense que j'aurais préféré utiliser un modèle préconçu pour l'architecture (HTML et CSS) sur des sites spécialisés plutôt que d'avoir fait comme j'ai fait, c'est à dire tout coder de A à Z. Bien que cela permette d'avoir la main mise sur absolument TOUT ce que l'on conçoit, ça prend aussi un temps monstre, qu'on aurait pu utiliser pour autre chose (par exemple créer une histoire qui suit le jeu ...). Je pense également que j'aurais préféré une base différente qu'une base avec des scripts CGI (scripts python) qui peut poser pas mal de problèmes liés notamment au manque de doc et a la faible utilisation de ces derniers pour des projets concrets.**

DOCUMENTATION

> Description du site :

• **Il y a tellement de fonction et de scripts qu'il serait trop long de tous les décrire un par un c'est pourquoi je vais associer ci-dessous chaque page du site au fichier/bout de code qui correspond :**

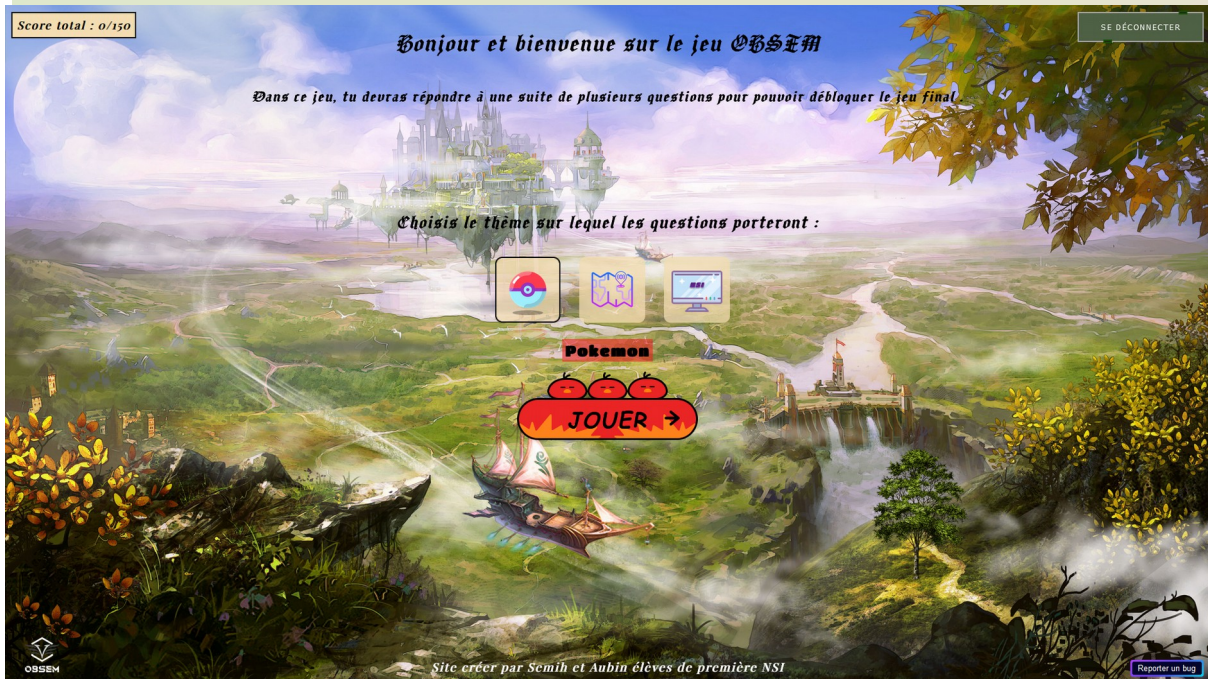
→ **L'accueil :**



Cette page est tout simplement la page d'accueil qui permet de se connecter ou de s'inscrire au site, le script qui gère son fonctionnement a pour nom «accueil.py». Le code javascript est directement intégré dedans pour que l'on puisse traiter des informations venant des fichiers csv à l'aide de ce langage. Ceci sera utile pour la page d'inscription afin de ne pas créer 2 fois le même pseudo et pour la page de connexion pour tout simplement se connecter à son compte lorsque l'on rentre les bonnes informations.

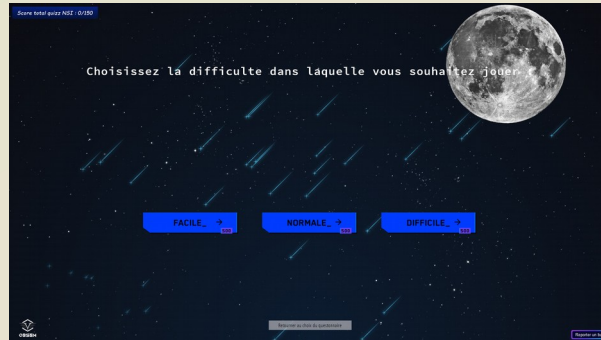
Lorsqu'une connexion ou une inscription réussie, le formulaire est envoyé à un script nommé «connectionScript.py» qui se charge d'écrire les bonnes informations dans les fichiers csv et de rediriger l'utilisateur où il faut (sur le jeu si il a réussi tous les quizzes sinon à la page de choix des questionnaires).

→ **La page de choix du questionnaire :**



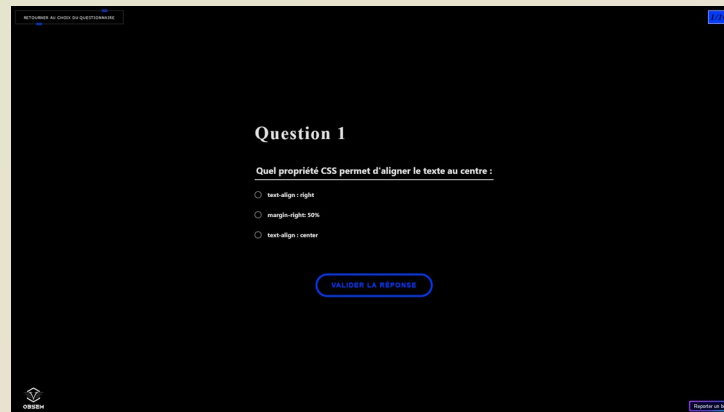
Cette page est le centre du site. Elle permet de choisir parmi les trois thèmes de quizz proposés et de voir le score total obtenu. Elle permet aussi d'accéder au jeu final lorsque tous les quizzs ont été faits. Le script en charge de cette page se nomme «pageChoixQuestionnaire.py» et est accompagné d'un script javascript externe («pageChoixQuestionnaire.js») pour afficher le score et gérer diverses actions.

→ La page de sélection de la difficulté :



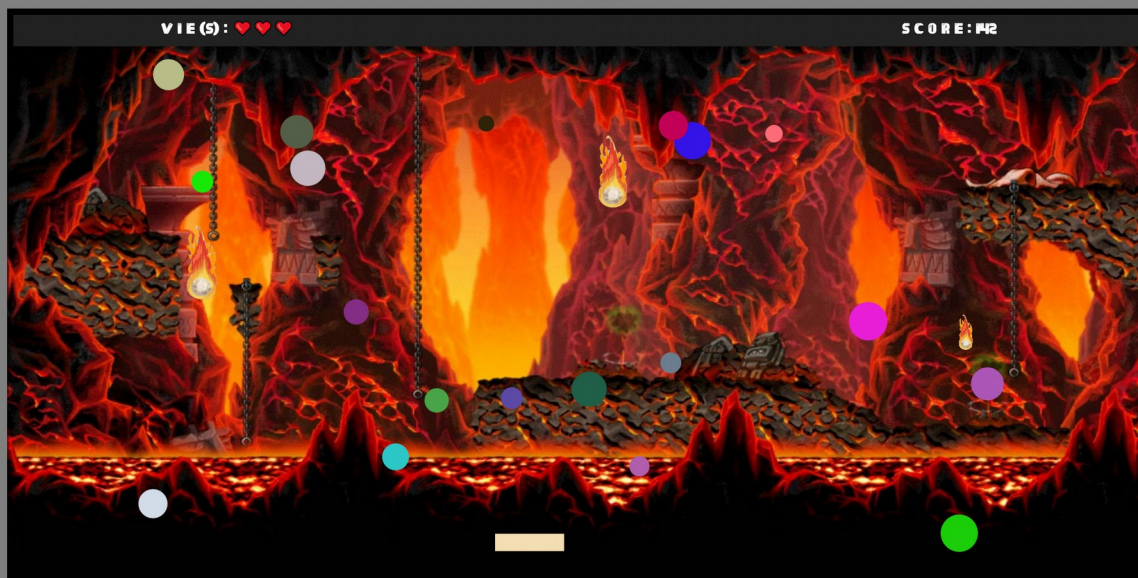
Avant de pouvoir faire un questionnaire, l'utilisateur doit d'abord choisir la difficulté dans laquelle il souhaite le faire. Chaque questionnaire possède trois difficultés qui peuvent rapporter au maximum 50 points chacune. La page de sélection de la difficulté est gérée par le script «selectDifficulty.py» et génère une page différente selon le thème donné.

→ Les quizzes :



Toutes les informations relatives aux quizzes sont traitées par un script nommé «`traitementReponseFormulaires.py`». Chaque questionnaire est différencié grâce à la variable «`theme`» récupérée dans l'URL. Chaque quizz possède également un fichier javascript qui lui est propre. De plus le script «`progressionQuiz.js`» commun à tous les questionnaires permet comme son nom l'inique de suivre la progression sur le quizz mais également de gérer des actions comme les sons et l'affichage. Les quizzes constituent la base du site et sont là avant tout pour divertir en même temps d'instruire l'utilisateur.

→ Le jeu final (Catch-Game) :



Cette page se débloque lorsque tous les quizzes ont été terminés et on peut y accéder par l'accueil lorsque l'on se connecte ou par la page de choix du questionnaire.

Le jeu disponible sur la page consiste en une plateforme rectangulaire dirigée par l'utilisateur et devant attraper le maximum de ronds colorés (chaque rond rapporte un certain score qui dépend de la taille de ce dernier) sans se faire toucher par les boules de feu (si cela arrive, le joueur perd une vie).

Le jeu peut se mettre en pause à tout instant lorsque le joueur appuie sur échap, à partir de là il a le choix entre reprendre le jeu, recommencer ou retourner à l'accueil (les deux dernières options entraînent une perte du score).

Le jeu se termine quand le joueur a perdu ses trois vies. Lorsque cela arrive il a alors le choix entre recommencer pour tenter d'améliorer son score ou bien aller à la page de classement.

Le jeu est géré principalement en javascript par le fichier «jeuFinal.js» et la page est générée par le script «finalGame.py».

→ **Page de classement :**

Se déconnecter Rejouer au Jeu Final

VOUS ÊTES PREMIER !

QUIZZS SCORE:		
Rank : 1	Username : test	Score : 150
Rank : 2	Username : tifi	Score : 136

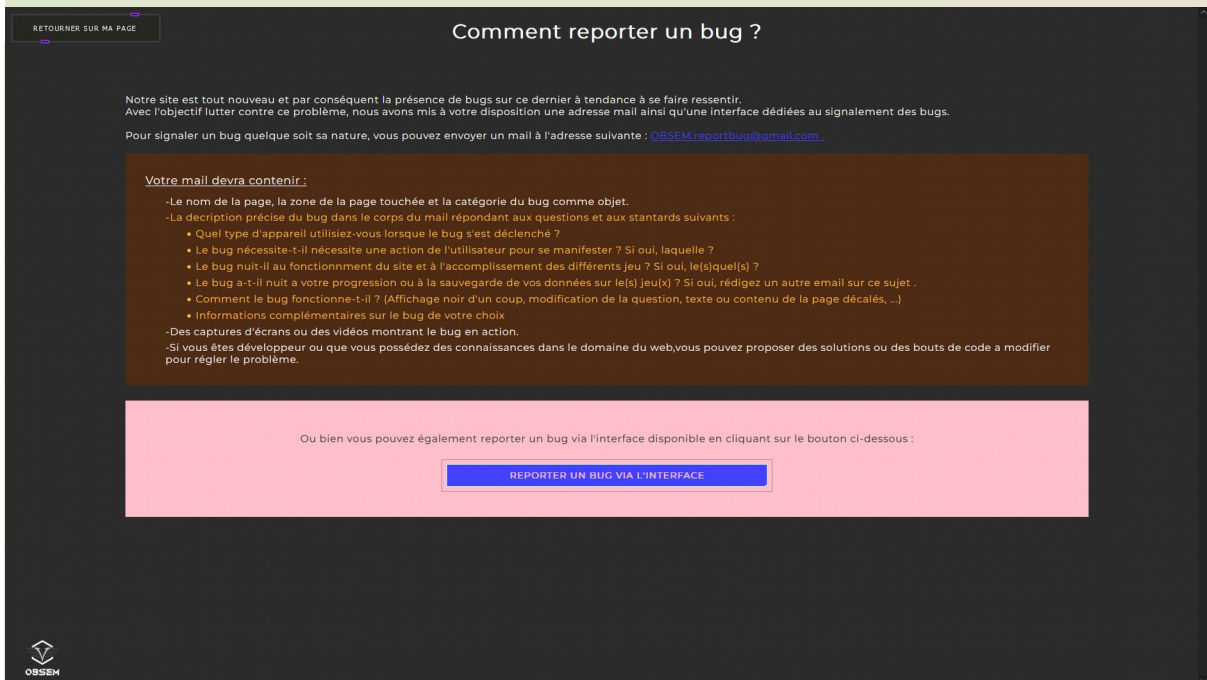
VOUS ÊTES DEUXIÈME !

GAME SCORE:		
Rank : 1	Username : tifi	Score : 4252
Rank : 2	Username : test	Score : 154

 Reporteur un bug

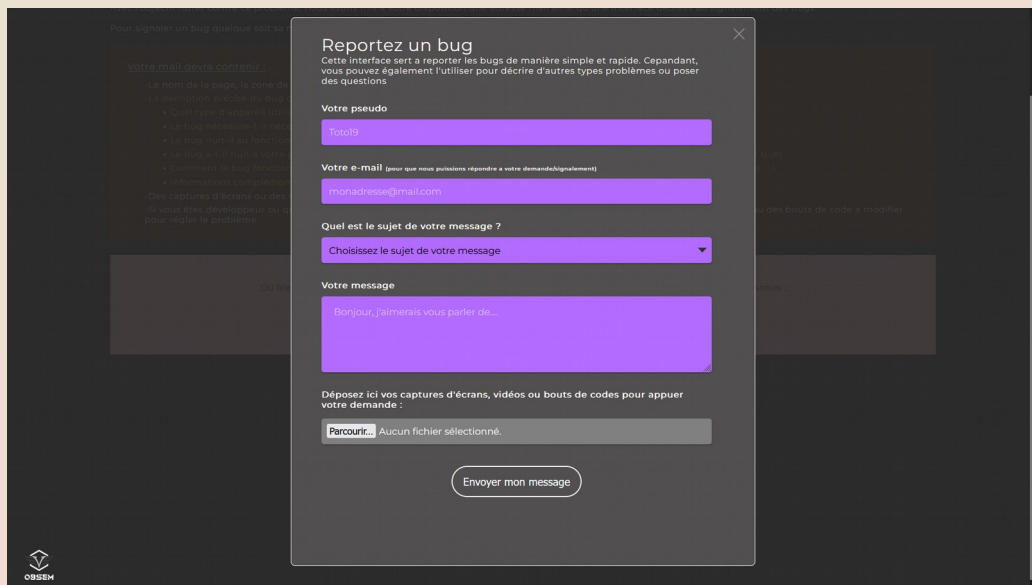
Cette page montre le classement de tous les joueurs ayant fini le jeu. On arrive à afficher le classement en triant les données du fichier «classement.csv» en python avec le script «classement.py». Le score du jeu peut être amélioré par n'importe quel joueur à n'importe quel moment. Cependant celui des quizzes ne peut pas changer car on ne peut faire les quizzes qu'une seule fois. Grâce à un script open source trouvé sur Internet et quelques modifications, on peut obtenir son rang dans le classement en toutes lettres. Cette page constitue normalement la fin du site.

→ **Page pour reporter les bugs :**



Comme vous avez pu le remarquer sur les images précédentes, un bouton «reporter un bug» est présent sur toutes les pages du site, il mène en fait vers le fichier «reportBug.html». Cette page donne des précisions sur comment reporter un bug et l'adresse mail à laquelle il peut être reporté. Elle propose également une interface pour reporter directement un bug sans s'embêter à rédiger un mail (voir ci-dessous):

→ L'interface en question :



Cette interface fonctionne avec le projet libre smtpJS mentionné plus tôt, l'utilisateur rentre les informations nécessaires et si elles remplissent certaines conditions (validité du mail, taille du message, champ rempli, ...) elles sont envoyées à l'adresse mail dédiée, écrite sur la page.

→ Page d'erreur d'IP :

Vous n'avez pas accès a cette page !

Notre site fonctionne par le biais des adresses IP et nous voyons que vous n'avez pas créer de compte avec une IP associée sur notre site.
Si nous vous laissons parcourir notre jeu tel quel, cela provoquerait de nombreuses erreurs, par conséquent votre accès est restreint pour le moment.

Le fait que votre adresse IP ne soit pas renseignée peut venir de deux choses:

- Vous ne vous êtes pas encore crée de compte, dans ce cas nous vous invitons à le faire sans tarder en cliquant sur le lien suivant : [Accueil](#)
- Vous vous êtes connecté sur un autre appareil et vous avez laissé votre appareil actuel allumé, puis vous avez tenté de continuer à jouer sans repasser par la page d'accueil. Dans ce cas, nous vous recommandons de vous connecter de nouveau via ce lien : [Accueil](#)

Si vous n'arrivez toujours pas à accéder à notre site, tenter de redémarrer votre appareil ou signaler nous un bug à : OBSEM.reportbug@gmail.com.
Plus d'informations pour savoir comment signaler un bug en cliquant sur le bouton en bas à droite.



Site crée par Semih et Aubin, élèves de premières spécialité NSI

[Reporteur un bug](#)

Cette page s'affiche lorsqu'une erreur avec l'adresse IP se produit (lorsqu'elle n'est pas trouvée dans la base de données). Un script pour tester si l'adresse IP existe est présent à chaque début de page et permet l'affichage d'un message d'erreur. Cela évite de pouvoir se connecter sans s'être inscrit et cela évite également d'autre problèmes liés aux données des fichiers csv. Le site reconnaissant les utilisateurs grâce à leur adresse IP et gérant les données en conséquence, on ne peut pas se permettre que quelqu'un se connecte sans que son adresse ne soit enregistrée.

Cette page est reliée à deux fichiers externes (un CSS et un javascript). Le code pour faire fonctionner la page doit être réécrit à chaque début de fichier du site faute d'avoir trouvé comment utiliser des fonctions venant de fichiers externes et générant du HTML dans des scripts CGI.

> Spécifications techniques :

→ **Architecture** : Le site fonctionne grâce à des scripts CGI gérés par le module natif «http.server» de python. Ces scripts python viennent générer les pages nécessaires au fonctionnement du site et viennent constituer la structure du projet.

→ **Langages et bibliothèques utilisés :**

- Python : interactions client-serveur et traitement de données, bibliothèques : http.server, cgi, cgi, csv, os, sys, datetime .
- HTML : structure web avec laquelle sont écrites les pages via python .
- CSS : style du site et ergonomie, bibliothèque : feather (icônes).
- Javascript : gère les actions du client sur la page, bibliothèques :

jQuery, smtpJS .

→ **Matériel et choix techniques :**

-Le projet OBSEM à été développé sous Visual Studio Code, mais, pour lancer le serveur (fichier «OBSEM-Serv.py») l'éditeur de code spyder à été utilisé et le fonctionnement sous d'autres éditeurs de code n'est absolument pas garanti et risque de provoquer des erreurs fatales telles que le malfonctionnement du module csv ou encore des problèmes d'encodages.

-Le projet OBSEM a principalement été testé sous Firefox et des problèmes peuvent survenir sous d'autres navigateurs.

→ **Format de stockage des données** : Les données de l'utilisateur sont stockées au format csv et traitées grâce au module python du même nom. Il existe en tout et pour tout trois fichiers csv nécessaires au bon fonctionnement du site :

- «activityOfTheSite.csv» qui contient le pseudo, le mot de passe, l'adresse IP, la dernière date de connexion et le score de l'utilisateur.

- «pointsOfGamer.csv» qui contient les données relatives aux questionnaires effectués par l'utilisateur.

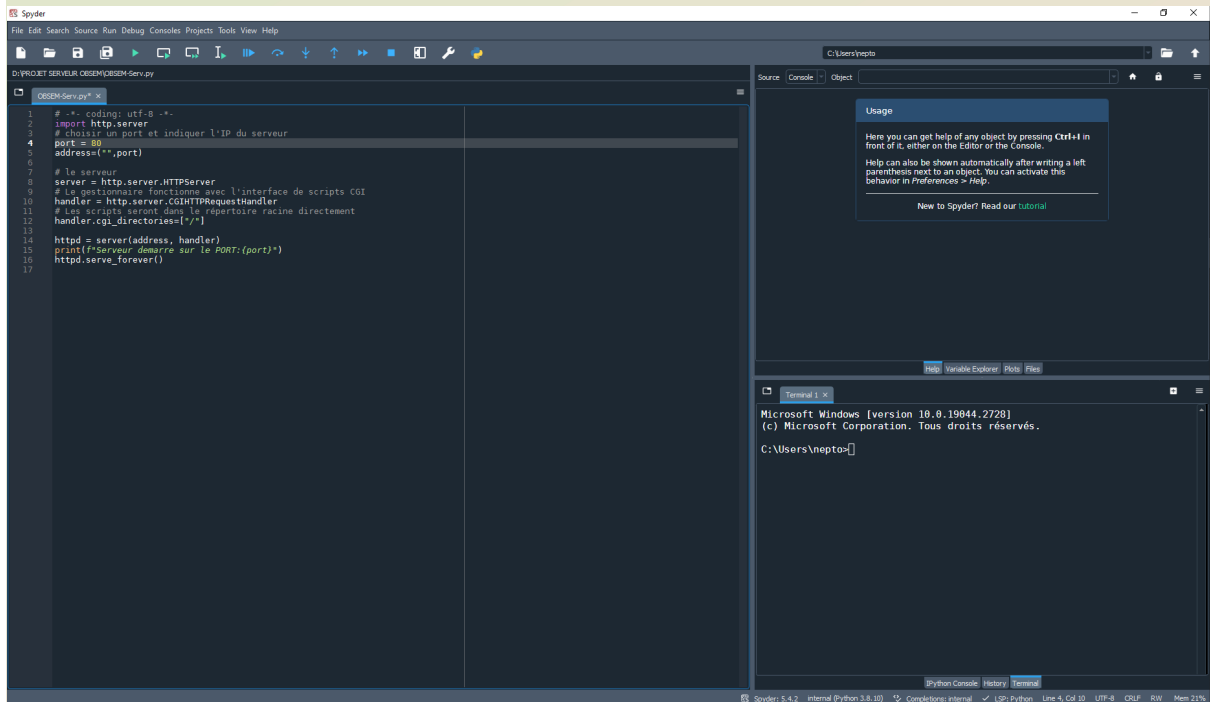
- «classement.csv» qui contient le score total obtenu au quiz et le score obtenu au jeu final de l'utilisateur.

> Guide d'utilisation et informations complémentaires :

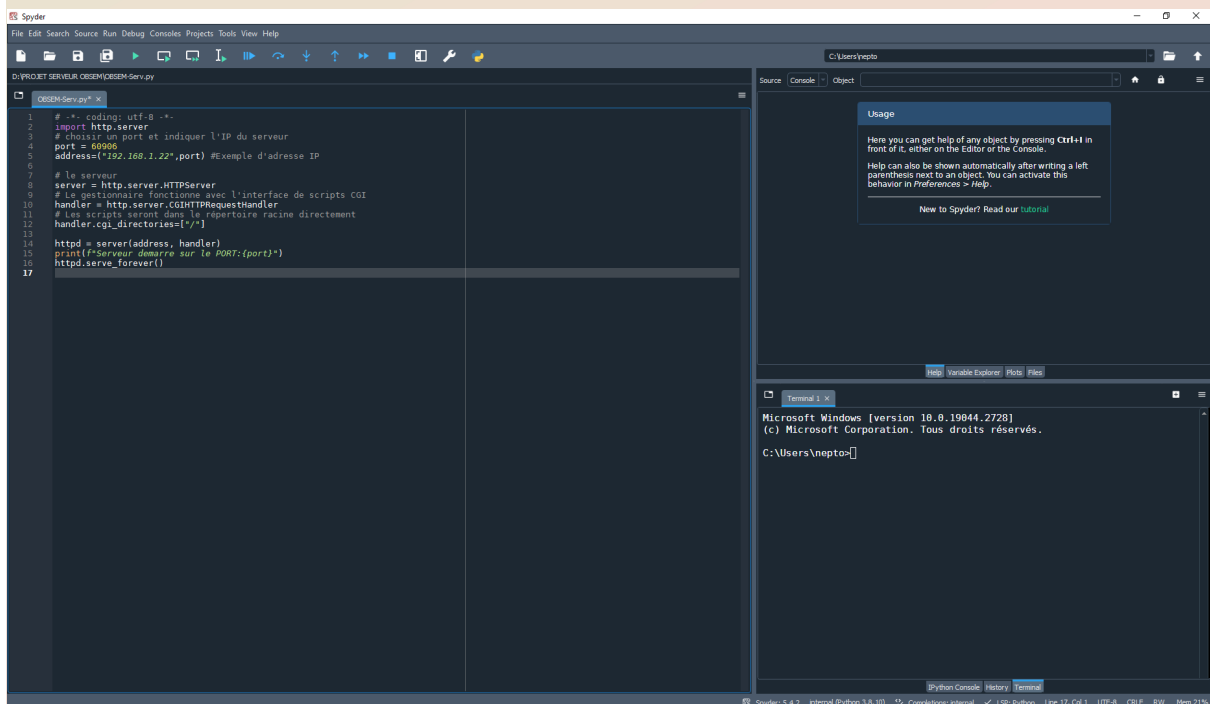
Pour lancer le serveur, ouvrez le fichier «OBSEM-Serv.py» dans spyder, puis dans la variable adresse, modifiez "VOTRE ADRESSE IP" par l'adresse de la

machine sur laquelle le serveur va être hébergé, sans modifier le port. Faites ceci uniquement si vous souhaitez accéder au serveur via une autre machine sinon remplacer simplement le champ "VOTRE ADRESSE IP" par "" et changez la valeur la variable port à 80.

→ Serveur en local :



→ Serveur distant :



Une fois cette étape achevée, allez sur Firefox et tapez:

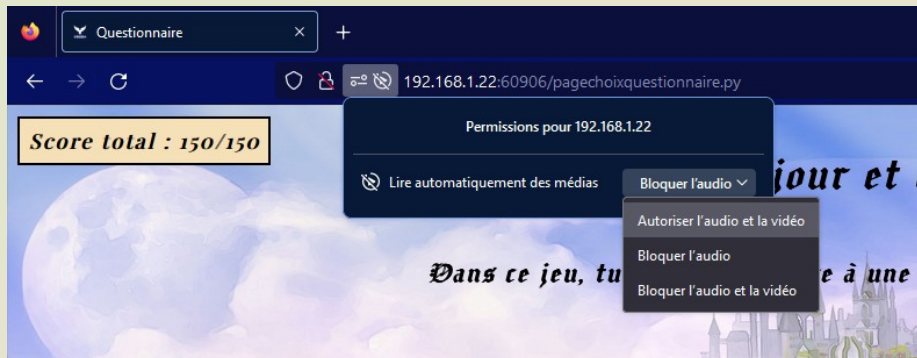
→ localhost/accueil.py (pour les serveurs locaux)

→ VOTRE ADRESSE IP:60906/accueil.py (pour les serveurs distants)

Vous arriverez normalement sur la page d'accueil. À partir de là, je vous laisse découvrir le site !

→ Informations complémentaires :

-Veillez à autoriser l'audio et les vidéos sur Firefox quand vous utilisez le site (voir capture d'écran ci-dessous).



-Si le jeu final semble être lent ou semble bugger, rafraîchissez la page.

-Pour consulter les messages envoyés via l'interface de report de bugs, les informations de connexion au compte dédié sont disponibles dans le dossier reportBugGuide sous le nom de «Log account Bug.txt» .

-Toutes les images ou autres productions artistiques ne sont pas forcément libres de droit, mais ce projet étant normalement censé rester à un statut privé, nous avons estimé que ce ne serait pas un problème pour le moment. Mais nous nous excusons d'avance si cela venait à engendrer des complications.

Un grand merci à tous ceux qui ont lu ce dossier jusqu'ici, la seule chose que nous puissions vous souhaiter c'est de profiter de notre projet. Si vous avez des questions passez par l'interface de report de bugs et sélectionner le sujet autre, ou envoyez un mail à l'adresse dédiée.

Signé : L'équipe de développement du projet OBSEM