

WEB SERVEUR I  
420-2SU-SW

ÉVALUATION FINALE - PROJET  
HIVER 2025

Pondération : 40%

## Travail à réaliser

Vous devez développer un site web qui répertorie les résultats des parties jouées sur l'arcade du département. Nous nommerons ce site : **Point Final**.

Voici l'exemple : <http://fm.shawinigan.info:98/>

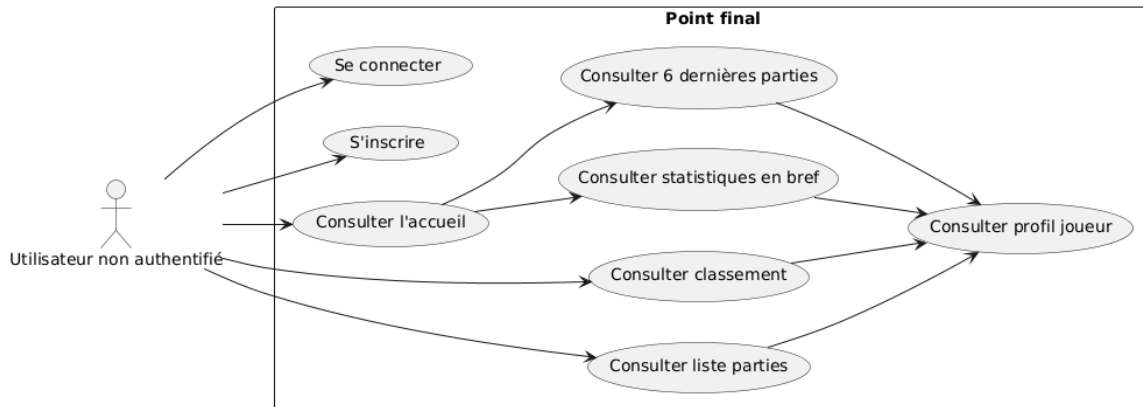
## Exigences non fonctionnelles

Exigences non liées aux fonctionnalités :

- La programmation orientée objet doit être utilisée (modèles, DAO).
- La sécurité du site est évaluée (ex. : requêtes préparées, échappement des sorties, etc.)
- Le site doit être en php.
- La base de données doit être MariaDB.
- Vous avez accès à un script de BD. Vous pouvez l'utiliser et même le modifier. Si vous le modifiez, vous devez évidemment remettre le script modifié avec votre remise!
- Vous devez mettre votre projet sur GitHub et me partager votre dépôt :
  - Vous devez faire des *commit/push* régulièrement.
- L'utilisabilité et la convivialité sont évaluées (UX → *User experience*) : rétroaction, refournir les données en cas d'erreur ou validation javascript, facilité d'utilisation, comportement conséquent, etc.
- Quoique l'essentiel de l'évaluation n'est pas centré sur le visuel, des points y sont quand même consacrés. Toutefois, focalisez-vous **dans un premier temps sur les fonctionnalités** (et non le visuel). Précisions sur le visuel :
  - Vous **NE devez PAS reproduire** mon visuel, il faut faire le vôtre.
  - Vous **NE pouvez PAS reprendre le html/css** des notes de cours (à moins que vous ayez fait le vôtre).
  - Vous pouvez partir d'un *template* utilisant une librairie CSS (ex : Bootstrap) sans problème (sauf celui que j'ai utilisé). Si vous partez d'un *template*, vous devez inclure en bas de page un lien vers l'auteur de ce dernier.
  - Le site ne doit pas nécessairement être *responsive*. Toutefois, son affichage doit être optimal sur du 1080p.

## Exigences fonctionnelles

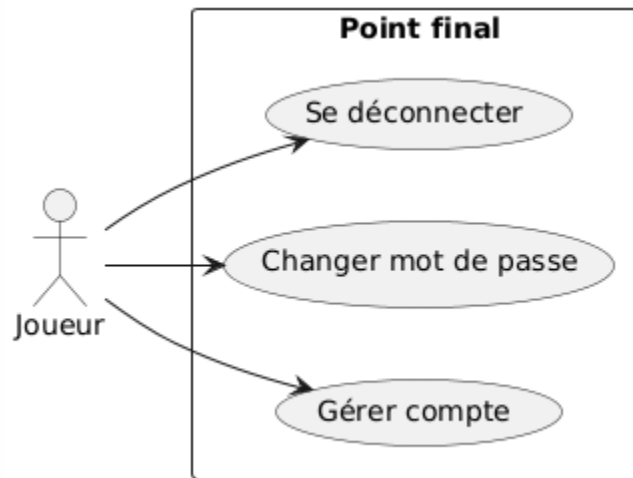
Lorsque l'utilisateur du site **n'est pas** authentifié :



- Consultation de l'accueil
  - Consultation des statistiques « En bref » :
    - Le nombre de joueurs inscrit sur le site
    - Le nombre de jeux disponible sur le site
    - Le nombre de parties jouées total par tous les joueurs confondus
    - Le meilleur joueur actuellement (dao/modèle déjà fournis)
      - En cliquant sur le nom du meilleur joueur, nous consultons son profil
  - Consultation des six (6) dernières parties jouées.
    - Pour chaque partie, nous voyons les informations suivantes :
      - Le joueur 1 : son nom, son image et son score
      - Le jeu : son nom, son image
      - Le joueur 2 : son nom, son image et son score
      - La date de la partie
    - En cliquant sur l'image d'un des joueurs, nous consultons son profil
- Consultation du classement des joueurs :
  - Affiche un tableau du classement des joueurs. Les éléments nécessaires pour obtenir ces informations vous sont déjà fournis (DAO/modèle).
  - En cliquant sur le nom d'un joueur, nous consultons son profil.
- Consultation de la liste des parties :
  - Affiche un tableau avec la liste de toutes les parties de la plus récente à la plus ancienne.
  - En cliquant sur le nom des joueurs, nous consultons son profil

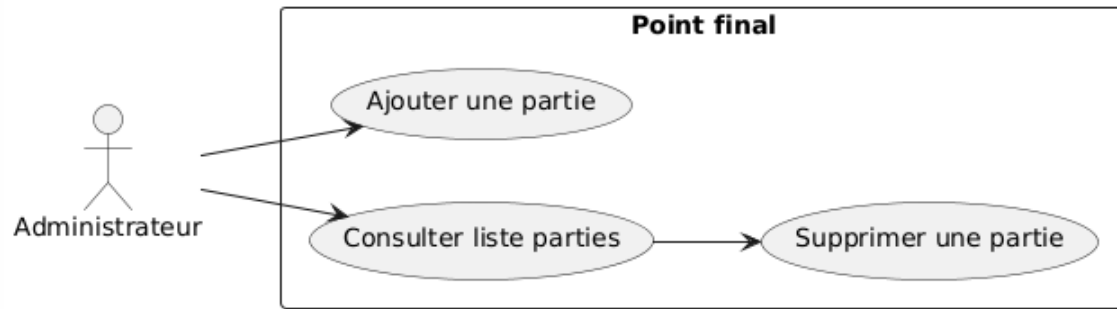
- Consultation du profil d'un joueur :
  - Affiche les informations de base du profil :
    - L'image
    - Le nom/prénom
    - Le nom d'utilisateur
    - La bio
    - Membre depuis
    - Type de compte
  - Affiche les statistiques du joueur. Les éléments nécessaires pour obtenir ces informations vous sont déjà fournis (DAO/modèle).
  - Toutes les parties jouées par ce joueur. Les informations à afficher pour chaque partie sont les mêmes que celles spécifiées pour l'accueil.
  
- L'utilisateur peut se connecter.
  - Après la connexion, l'utilisateur est dirigé vers sa page de profil.
  - Après la connexion, le menu reflète l'utilisateur connecté.
  - Une fois connecté, le seuil d'inactivité toléré est de 5 minutes. Sinon, il est redirigé vers la page de connexion.
  
- L'utilisateur peut s'inscrire (créer un compte) :
  - Le formulaire présente les champs (tous obligatoire) :
    - Nom utilisateur : Seuls les caractères alphanumériques sont acceptés. Entre 1 et 50 caractères.
    - Mot de passe : 5 caractères minimum.
    - La confirmation du mot de passe. Le mot de passe et la confirmation doivent correspondre.
    - Le prénom de l'utilisateur. Entre 1 et 50 caractères.
    - Le nom de l'utilisateur. Entre 1 et 50 caractères.
    - Bio : Entre 1 et 255 caractères.
    - Url avatar/image profil : L'URL doit être de format valide.
    - Le type de compte est systématiquement « Joueur ».
  - Lors d'une inscription réussie, message de succès.

Lorsqu'un utilisateur est authentifié (type de compte « Joueur »)



- Pour tester sur le site de référence : monchampf / bonjour
  - **Ne changer pas son mot de passe svp!**
- Il peut évidemment faire tout ce qu'un utilisateur non connecté peut faire.
- Il peut se déconnecter. Il est redirigé vers la page de connexion.
- Vous devez faire l'une ou l'autre des deux fonctionnalités suivantes :
  1. Il peut gérer son compte :
    - Changement du prénom
    - Changement du nom
    - Changement de la bio
    - Changement de l'image d'url
  2. Il peut changer son mot de passe :
    - Il doit fournir son mot de passe actuel pour changer son mot de passe

Lorsqu'un utilisateur est authentifié (type de compte « Administrateur »)



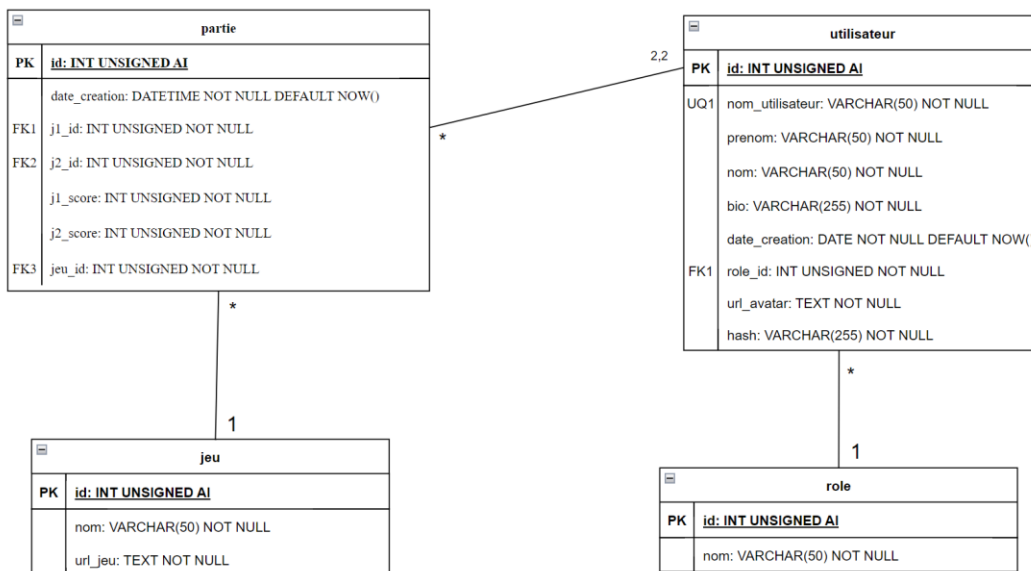
- Pour tester sur le site de référence : root / bonjour
  - Ne changer pas son mot de passe svp!
- Il peut faire tout ce que l'utilisateur connecté de type « Joueur » peut faire.
- Il peut ajouter une partie (tous les champs sont obligatoires) :
  - Choisir le joueur 1 (type « Joueur » seulement)
  - Fournir le score  $\geq 0$  du joueur 1
  - Choisir le joueur 2 (type « Joueur » seulement). Il doit être différent du joueur 1
  - Fournir le score  $\geq 0$  du joueur 2
  - Choisir le jeu
- À partir de la liste des parties, il peut supprimer une partie.

## Projet de départ

Un projet de départ vous est fourni. Il contient :

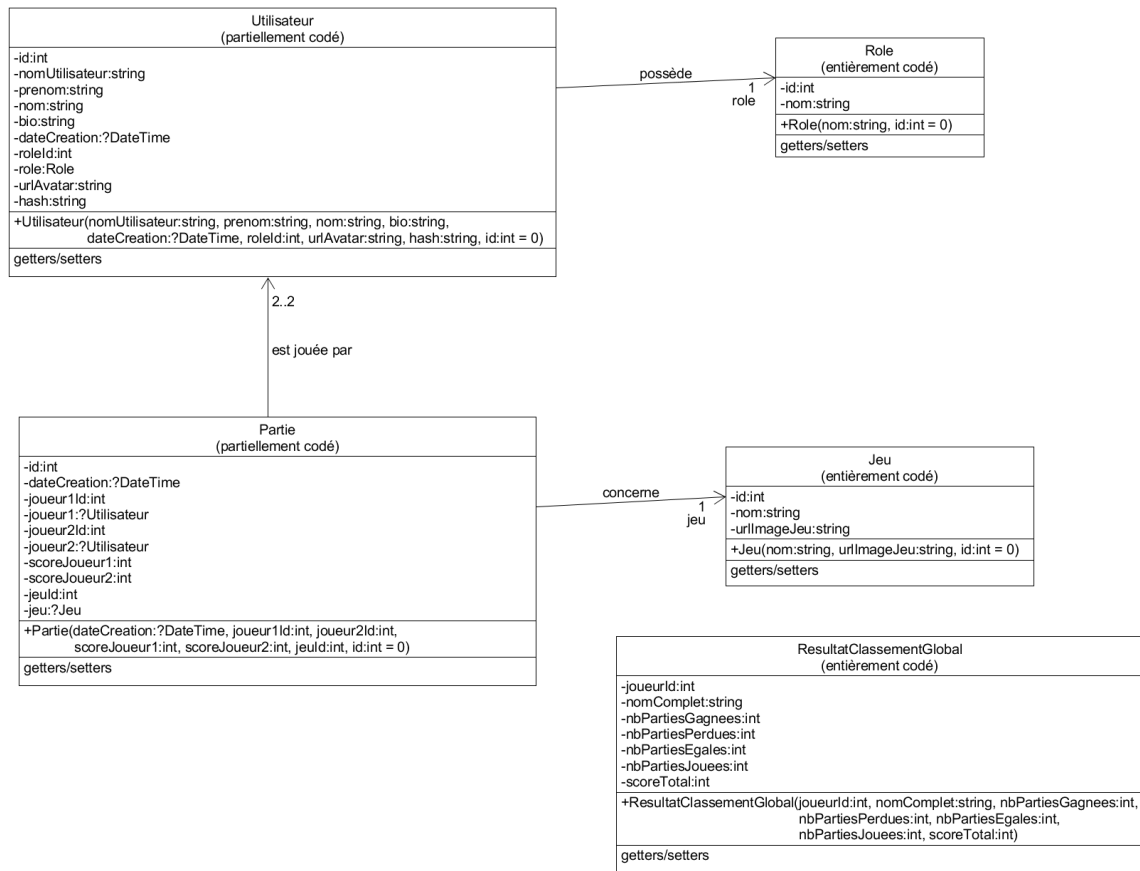
- Un script sql qui permet de créer la base de données et qui insère déjà des données de tests.
- Les classes de bases pour les DAO
- Certains DAO entièrement codés ou partiellement codés.
- Les modèles (mais ils n'ont pas tous les validations nécessaires dans les setters).
- Les fichiers php suivants :
  - autoloader.php
  - config.php
  - session.php
- Une page html de départ
- Le logo du site et son favicon.ico

## Schéma de la base de données



## Diagramme de classes des modèles

Note : Les modèles sont déjà faits, mais les *setters* d'Utilisateur et Partie ne possèdent pas les vérifications nécessaires afin d'assurer l'intégrité des données.

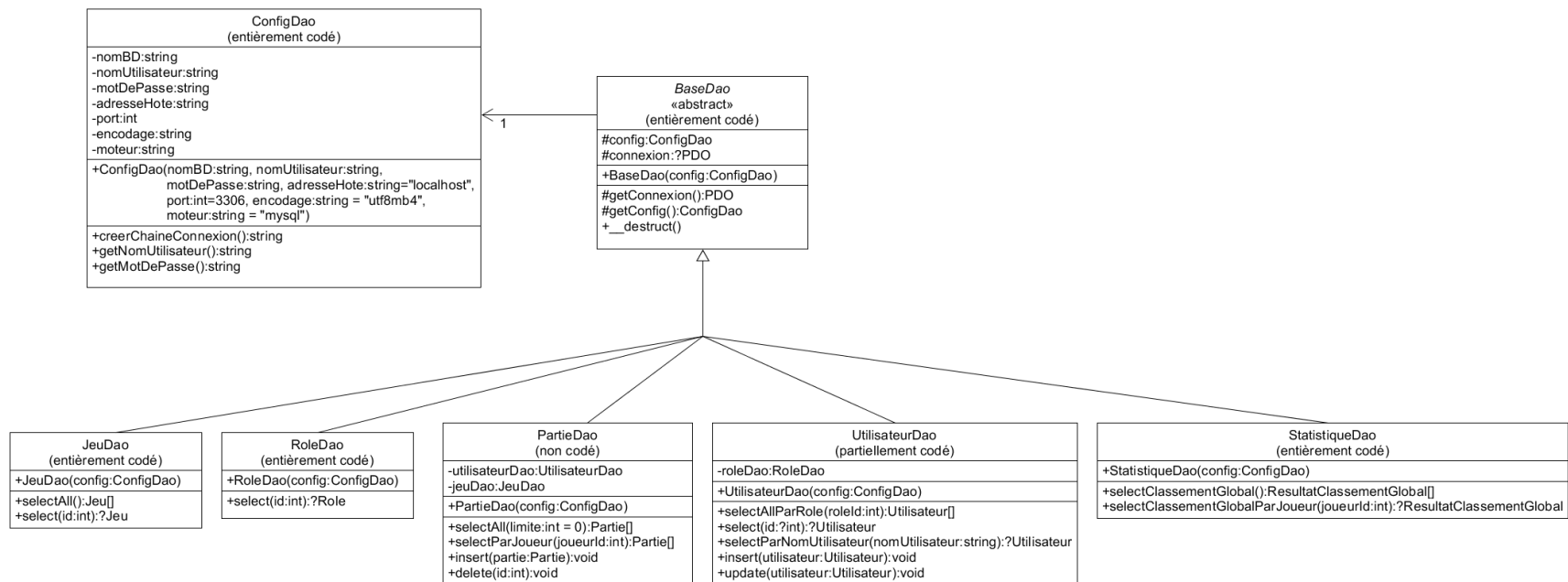




## Diagramme de classes des DAO

Note 1 : Certains DAO sont déjà complètement ou partiellement codés pour vous.

Note 2 : Vous n'êtes pas obligés de suivre à la lettre les méthodes proposées dans les DAO. Vous pouvez faire vos propres méthodes. J'ai mis celles dont j'ai eu besoin, mais il n'y a pas qu'une seule façon de faire les choses...



## Github (obligatoire)

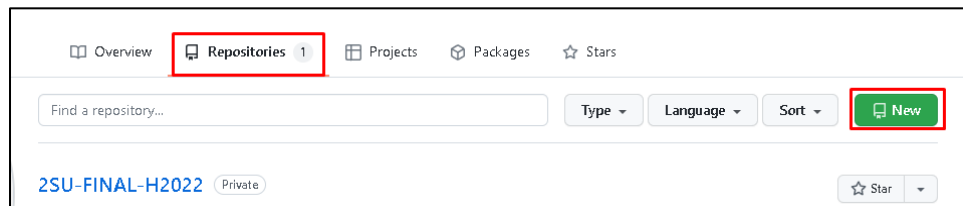
Puisque la remise se fait via Github et qu'il y a des points associés à la régularité des *push*, voici la marche à suivre pour mettre en place votre projet. Je vous encourage fortement à ce que ça soit la première chose que vous faites, avant même de coder une ligne.

### Étape #1

Créez-vous un compte sur Github si ce n'est pas déjà fait : <https://github.com/>

### Étape #2

Créez-vous un dépôt **privé** sur Github à l'aide de l'interface web.



### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

*Required fields are marked with an asterisk (\*).*

**Repository template**

Start your repository with a template repository's contents.

**Owner \***  / **Repository name \***

✓ 2SU-FINAL-H2024 is available.

Great repository names are short and memorable. Need inspiration? How about [cautious-spoon](#) ?

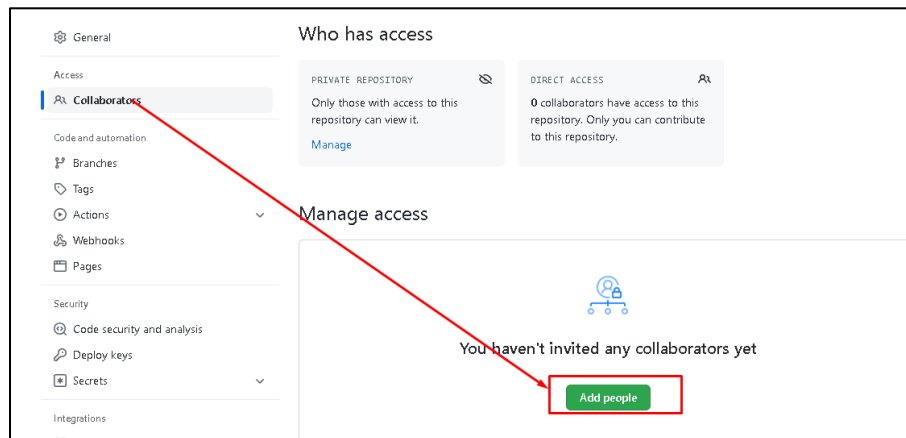
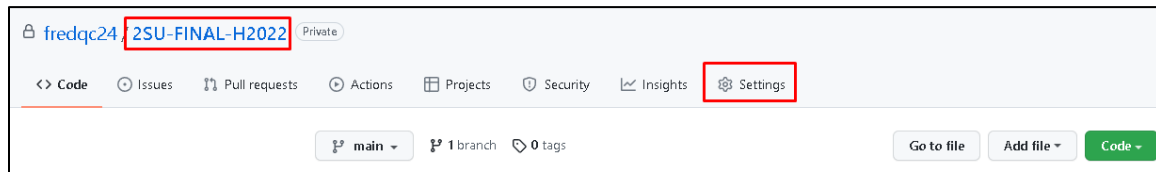
**Description (optional)**

☐ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☒ **Private**  
You choose who can see and commit to this repository.

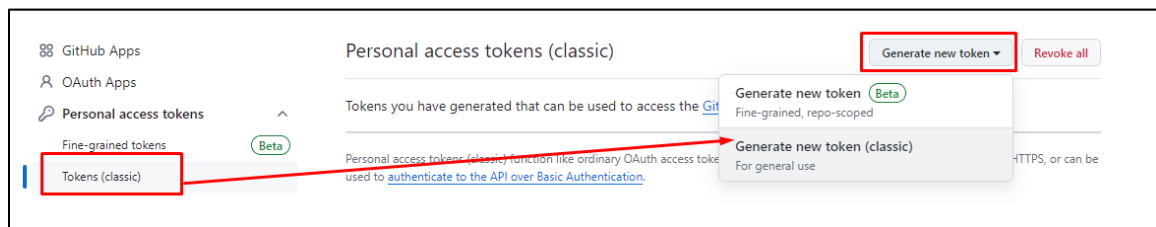
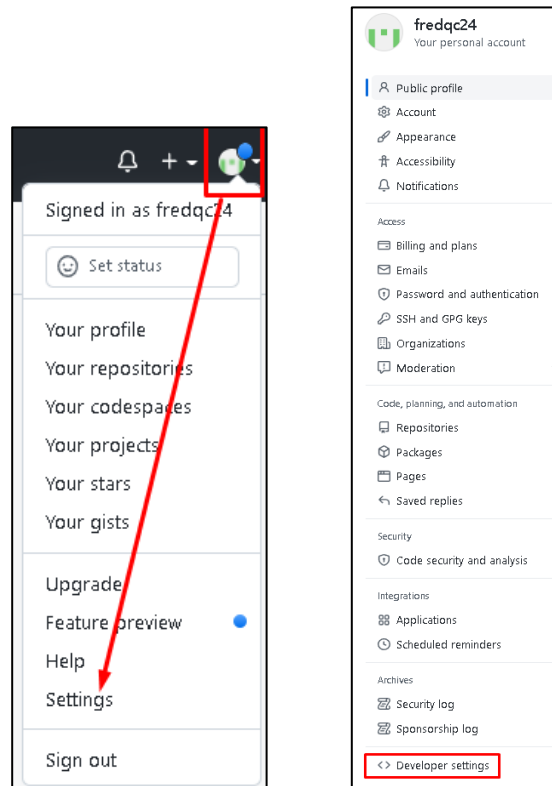
### Étape #3

Partagez-moi votre dépôt. Mon nom d'utilisateur est : **fredqc24**



#### Étape #4

Sur Github, créez-vous un token de sécurité. Dès qu'il est généré, **il faut IMPÉRATIVEMENT que vous le sauvez quelque part** (longue chaîne de caractères). Vous n'y aurez plus accès après coup. :



**New personal access token (classic)**

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

**Note**

Pour développement projet final 25U

What's this token for?

**Expiration \***

90 days The token will expire on Sun, Jul 7 2024

**Select scopes**

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo:deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events



### Étape #5

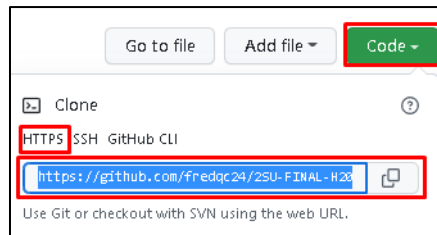
Allez dans votre environnement de développement (VM) et installez le client *git*. Il faut être connecté en tant que *root*.

```
apt-get install git
```

### Étape #6

Se déconnecter de *root*. Clonez le dépôt.

- L'URL de votre dépôt :



- Notez-le « **point** » à la fin de la commande « clone ». Il est important.
- Également, il vous sera demandé votre nom d'utilisateur GitHub ainsi que votre mot de passe à l'exécution de la commande. Donnez votre nom d'utilisateur GitHub. Par contre, au mot de passe, donnez le TOKEN généré précédemment.

```
cd /var/www/dossier_serveur_virtuel
```

```
git clone https://urldevotred depot .
```

**\*\*\*Note : Ce n'est pas la meilleure des pratiques de cloner un dépôt directement dans le répertoire d'un serveur Web, mais dans le cadre de notre développement, c'est le plus simple. De plus, notre serveur n'est pas exposé sur internet.\*\*\***

### Étape #7

Copiez le contenu du projet de départ dans /var/www/dossier\_serveur\_virtuel

## Étape #8

Ajout du code sur le dépôt distant.

```
cd /var/www/dossier_serveur_virtuel
git config --global --add safe.directory /var/www/dossier_serveur_virtuel
git add .
git config --global user.email "votrecourriel@exemple.com"
git config --global user.name "Votre Nom"
git commit -m "Premier commit de mon projet final."
git config --global credential.helper store
git push origin main
```

## Dernière étape (à répéter régulièrement)

```
git add .
git commit -m "Un message significatif"
git push
```

## Grille de correction

La grille de correction est disponible sous format Excel avec l'énoncé.

## Intelligence artificielle

L'utilisation de l'intelligence artificielle n'est pas interdite.

Toutefois, lors de votre correction, vous devez être en mesure de m'expliquer **TOUTES LIGNES** de code que je vous demanderai de m'expliquer. Si vous n'êtes pas en mesure de m'expliquer le code de manière satisfaisante, vous aurez 0 pour la ou les fonctionnalité(s) en question.

## Remise

La correction se fait « live » avec moi en temps réel.

**Si vous n'êtes pas présent, pas de correction, donc 0.**

Pour savoir le moment de votre correction, voir sur Moodle.