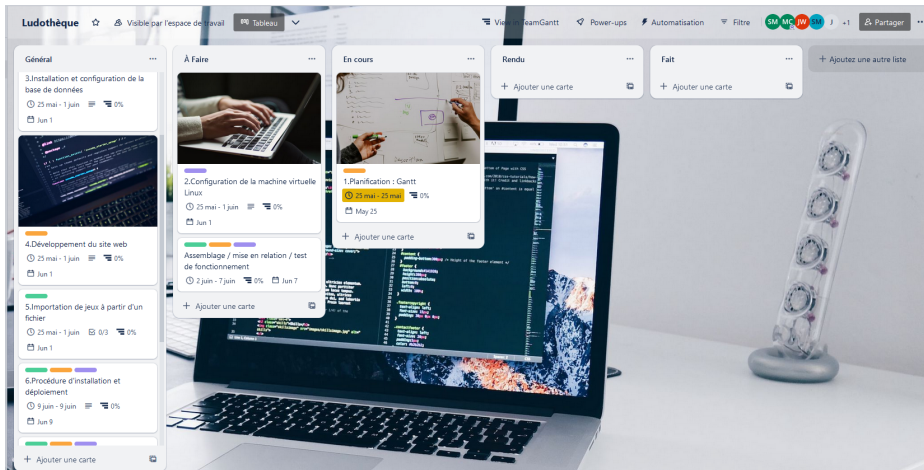


Notre travail:

Un planning, un diagramme de Gantt de votre projet (affectation des tâches, et temps) (par groupe);

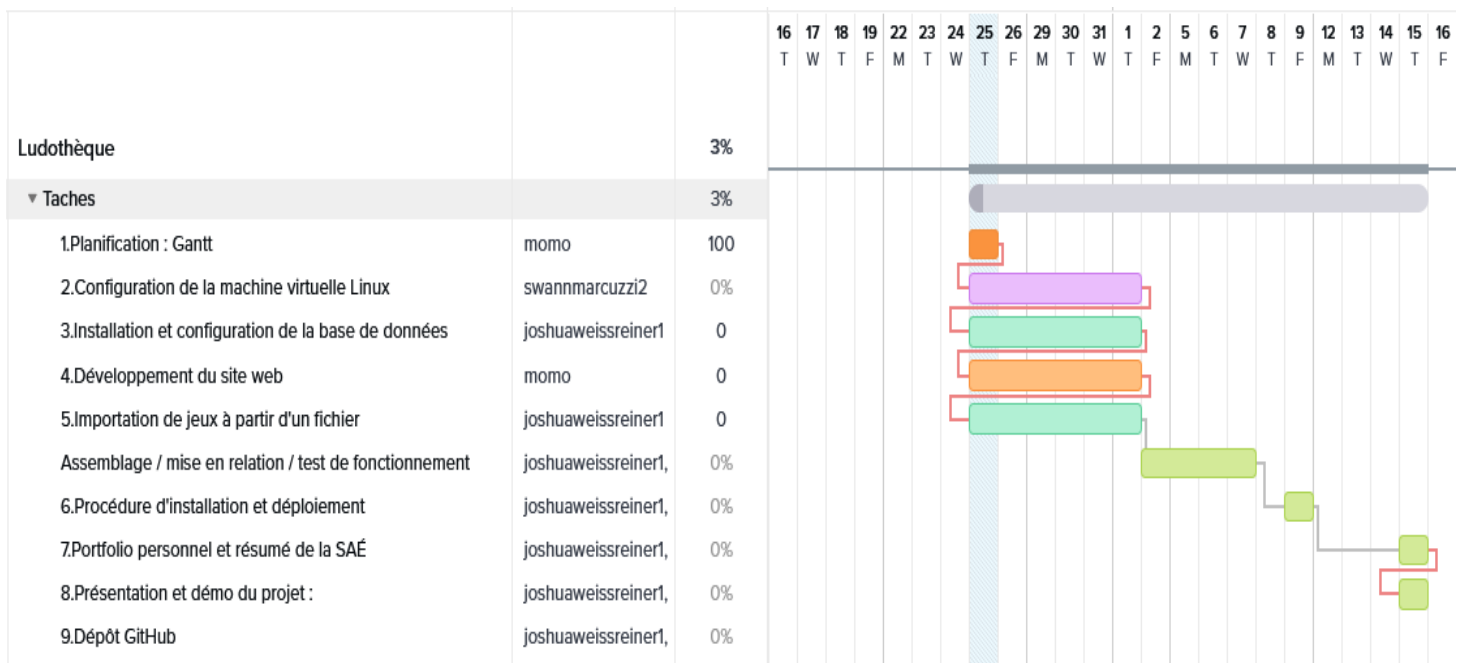
Pour cela nous utilisons le logiciel gratuit et en ligne “Trello”.<https://trello.com>



Outil simple d'utilisation qui nous permet de nous organiser.

Simple, flexibles et puissants. Les tableaux, les listes et les cartes suffisent pour obtenir une vue claire de qui fait quoi et des tâches qu'il vous reste à faire.

Voici notre diagramme de



gantt

Nous avons donc répartie les tâches comme cela:

momo : Développement du site web

- Commencez par concevoir l'architecture de votre application web en utilisant un framework ou une bibliothèque adaptée à votre langage de programmation (par exemple, Django pour Python, Laravel pour PHP).
- Implémentez les fonctionnalités CRUD pour chaque type de données (catégories, jeux, auteurs, joueurs, commentaires, liste de jeux) en utilisant les bonnes pratiques de développement.
- Créez les pages nécessaires pour l'inscription des nouveaux utilisateurs, la gestion des commentaires et la visualisation des jeux, des auteurs, des joueurs, etc.
- Calculez la moyenne des notes pour chaque jeu en fonction du type de personne (professionnel ou particulier) et affichez-la sur les pages correspondantes.
- Mettez en évidence les commentaires avec la note la plus élevée et la note la plus basse sur la page de chaque jeu.

swann : Configuration de la machine virtuelle Linux

- Configurez une machine virtuelle Linux sur la plateforme de votre choix (par exemple, Vmware).
- Installez les composants nécessaires pour votre développement, tels que le serveur web (Apache), le serveur de base de données (MySQL) et les outils de développement (PHP, Python, etc.).

joshua : Installation et configuration de la base de données

- Installez et configurez le serveur de gestion de base de données sur votre machine virtuelle (par exemple, MySQL).
- Créez une base de données pour votre projet et configurez les tables correspondantes au schéma de données fourni (catégories, jeux, auteurs, joueurs, commentaires, liste de jeux).

Machine virtuelle

nom de la machine: **debian.SAE**

mot de passe superutilisateur : **toto**

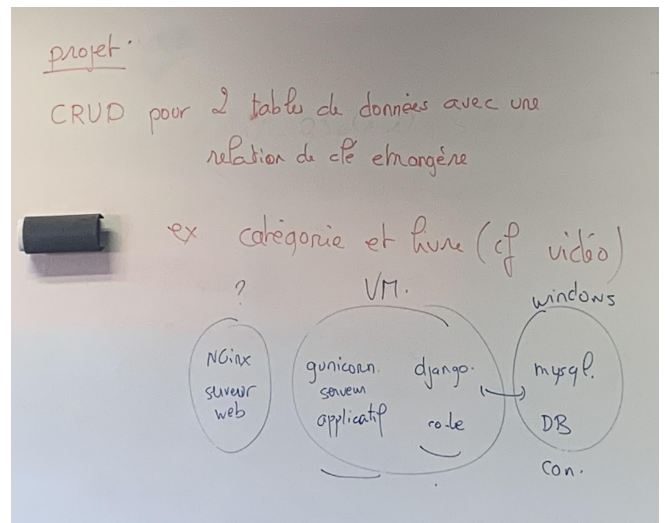
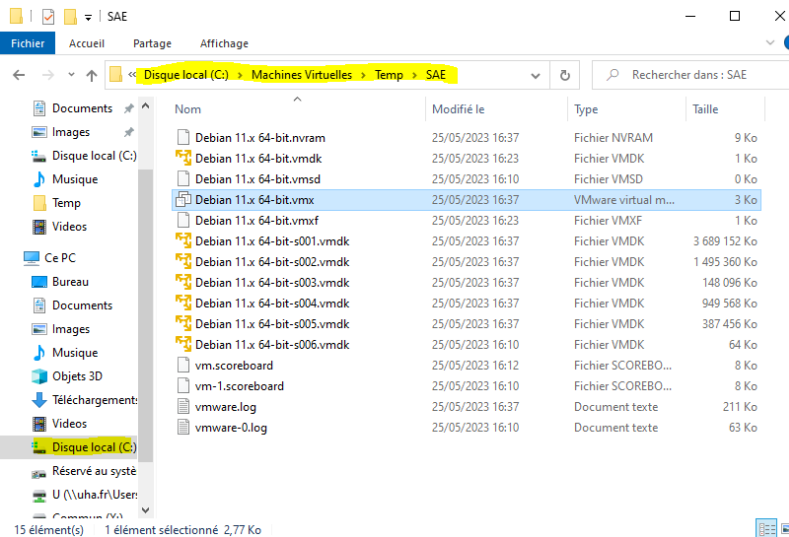
mot de passe utilisateur: **toto**

mot de passe superviseur : **toto**

nom complet du nv utilisateur : **totosae**

identifiant: **totosae**

```
1 .encoding = "windows-1252"
2 config.version = "8"
3 virtualHW.version = "19"
4 pciBridge0.present = "TRUE"
5 pciBridge4.present = "TRUE"
6 pciBridge4.virtualDev = "pcieRootPort"
7 pciBridge4.functions = "8"
8 pciBridge5.present = "TRUE"
9 pciBridge5.virtualDev = "pcieRootPort"
10 pciBridge5.functions = "8"
11 pciBridge6.present = "TRUE"
12 pciBridge6.virtualDev = "pcieRootPort"
13 pciBridge6.functions = "8"
14 pciBridge7.present = "TRUE"
15 pciBridge7.virtualDev = "pcieRootPort"
16 pciBridge7.functions = "8"
17 vmci0.present = "FALSE"
18 hpet0.present = "TRUE"
19 nvram = "Debian 11.x 64-bit.nvram"
20 virtualHW.productCompatibility = "hosted"
21 gui.exitOnCLIHLT = "FALSE"
22 powerType.powerOff = "soft"
23 powerType.powerOn = "soft"
24 powerType.suspend = "soft"
25 powerType.reset = "soft"
26 displayName = "Debian 11.x 64-bit"
27 usb.vbluetooth.startConnected = "TRUE"
28 guestOS = "debian11-64"
29 tools.syncTime = "FALSE"
30 sound.autoDetect = "TRUE"
31 sound.fileName = "-1"
32 sound.present = "TRUE"
33 vcpu.hotadd = "TRUE"
34 memsize = "2048"
35 scsi0.virtualDev = "lsilogic"
```



salle 201

1) Pour l'installation de MySQL-server

documentation: [Installer MySQL sur Debian 11 en 3 étapes \(tutobox.fr\)](https://tutobox.fr/installer-mysql-sur-debian-11-en-3-etapes)

```
sudo apt update
```

```
sudo apt upgrade
```

```
sudo apt install wget
```

```
wget https://dev.mysql.com/get/mysql-apt-config_0.8.22-1_all.deb
```

```
sudo apt-get install ./mysql-apt-config_0.8.22-1_all.deb
```

```
sudo apt-get update
```

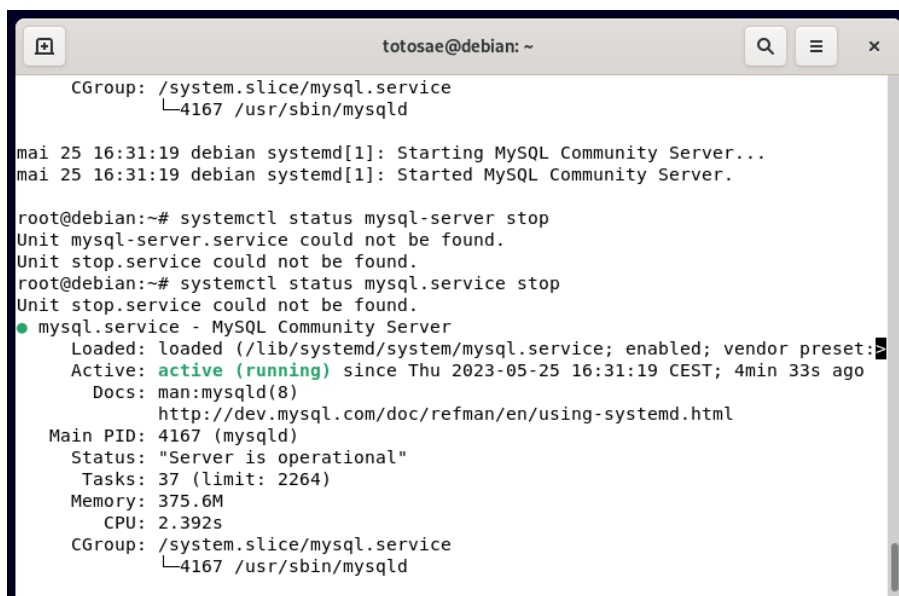
```
sudo apt-get install mysql-server
```

```
sudo systemctl mysql status
```

```
systemctl status mysql.service
```

pour modifier le statut

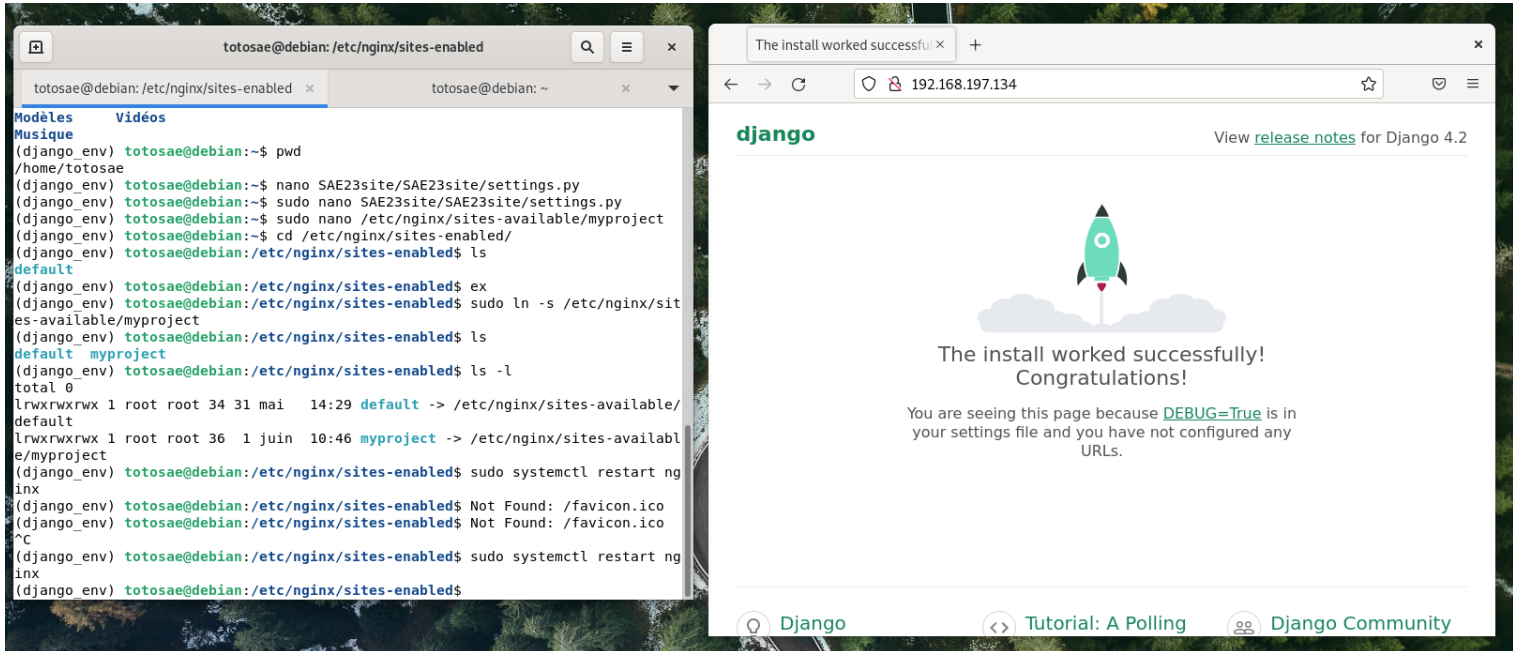
sois start ou stop.



```
totosae@debian: ~  
CGroup: /system.slice/mysql.service  
└─4167 /usr/sbin/mysqld  
  
mai 25 16:31:19 debian systemd[1]: Starting MySQL Community Server...  
mai 25 16:31:19 debian systemd[1]: Started MySQL Community Server.  
  
root@debian:~# systemctl status mysql-server stop  
Unit mysql-server.service could not be found.  
Unit stop.service could not be found.  
root@debian:~# systemctl status mysql.service stop  
Unit stop.service could not be found.  
● mysql.service - MySQL Community Server  
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset:   
   Active: active (running) since Thu 2023-05-25 16:31:19 CEST; 4min 33s ago  
     Docs: man:mysqld(8)  
           http://dev.mysql.com/doc/refman/en/using-systemd.html  
   Main PID: 4167 (mysqld)  
    Status: "Server is operational"  
     Tasks: 37 (limit: 2264)  
  Memory: 375.6M  
       CPU: 2.392s  
   CGroup: /system.slice/mysql.service  
           └─4167 /usr/sbin/mysqld
```

2) Pour l'installation le serveur websu -

<https://www.youtube.com/watch?v=YnrgBelRtvo>



le server web nginx est bien fonctionnelle

pour lancer le serveur web:

source django_env/bin/activate

gunicorn -c conf/gunicorn_config.py SAE23site.wsgi

Site WEB

Pour utiliser Bootstrap avec Django, vous pouvez suivre ces étapes :

Assurez-vous que vous avez Bootstrap et jQuery inclus dans votre projet Django. Vous pouvez le faire en téléchargeant les fichiers Bootstrap et jQuery et en les ajoutant manuellement à votre projet, ou en utilisant des méthodes telles que l'installation via npm ou l'utilisation de CDN (Content Delivery Network). Par exemple, vous pouvez ajouter les liens CDN dans la section <head> de votre fichier HTML :

html

Copy code

```
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css">

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js"
></script>
```

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
```

Configurez vos fichiers statiques dans Django. Dans votre fichier settings.py, assurez-vous que vous avez défini le chemin approprié pour vos fichiers statiques. Par exemple :

python

Copy code

```
STATIC_URL = '/static/'

STATICFILES_DIRS = [

    os.path.join(BASE_DIR, 'static'),

]
```

Créez un dossier static dans le répertoire de votre application Django, s'il n'existe pas déjà. À l'intérieur de ce dossier, créez un dossier css et un dossier js. Par exemple :

arduino

Copy code

```
your_project/

  your_app/

    ...

  static/

    css/

    js/
```

Ajoutez les fichiers Bootstrap et jQuery à vos dossiers css et js respectivement. Vous pouvez les télécharger depuis le site Bootstrap ou utiliser les fichiers que vous avez précédemment ajoutés à votre projet.

Dans votre fichier HTML, incluez les fichiers Bootstrap et jQuery en utilisant la balise `{% load static %}` pour charger les fichiers statiques, puis utilisez la balise `<link>` pour inclure le fichier CSS et la balise `<script>` pour inclure le fichier JavaScript. Par exemple :

html

Copy code

```
{% load static %}
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <link rel="stylesheet" type="text/css" href="{% static 'css/bootstrap.min.css' %}">
```

```
</head>
```

```
<body>
```

```
    <!-- Contenu de votre page -->
```

```
    <script src="{% static 'js/jquery.min.js' %}"></script>
```

```
    <script src="{% static 'js/bootstrap.bundle.min.js' %}"></script>
```

```
</body>
```

```
</html>
```

Maintenant, vous pouvez utiliser les classes et composants de Bootstrap dans votre code HTML pour styliser votre application Django. Référez-vous à la documentation Bootstrap pour en savoir plus sur les différentes classes et composants disponibles.

Assurez-vous de redémarrer votre serveur Django pour que les modifications prennent effet. Votre application Django devrait maintenant utiliser Bootstrap pour le stylage des éléments HTML.