SAE 2.03

BRISSE Chloé

MIRA Imène

SCHIMPF Luca

19/06/2025

—

Mettre en place une solution informatique pour l'entreprise

—

ALBERT Arnaud

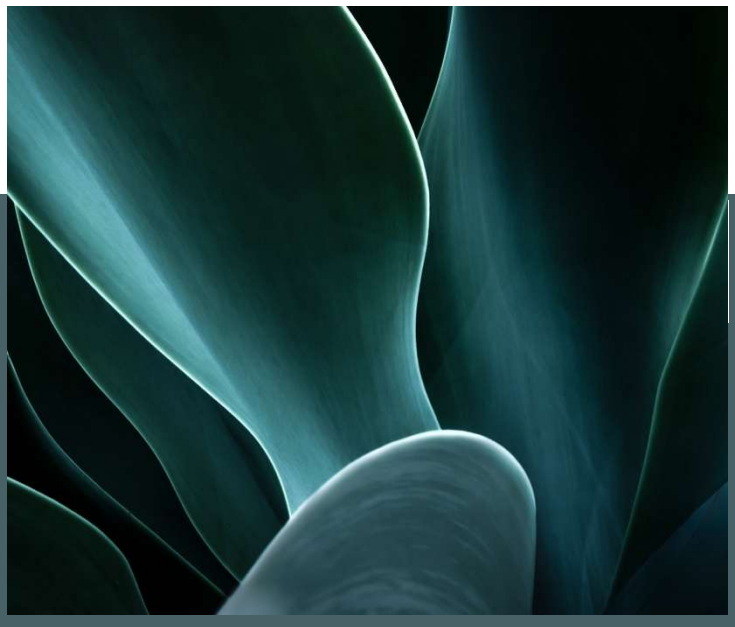
INSTRUCTIONS

La SAE 2.03 est un projet en groupe de 3 personnes portant sur la gestion d'une banque de films.

Il est demandé pour réaliser un site web avec une base de données le tout s'appuyant sur un ensemble de machine physique ou virtuelle.

Pour cela il faut créer 6 CRUD qui seront par la suite implémenté sur deux machines virtuelles. Une machine serveur base de données et une autres serveur web.

Les CRUD sont les suivant : Catégories de film, Films, Acteurs, Lien Acteurs/Films, Personnes(internaute) et Commentaire sur les films.



SOMMAIRE

DE CREATION DES CRUDs

Consignes

Ce sujet va vous permettre de fournir une interface de gestion d'une banque de films. Des usagers pourront commenter les films. Le schéma de données est le suivant

- Des catégories de film (id, nom, descriptif)
- Des films (id, titre, année de sortie, affiche, réalisateur, catégorie)
- Des acteurs (id, nom, prénom, âge, photos)
- Un lien entre les acteurs et les films
- Des personnes (id, pseudo, nom prénom, mail, mot de passe, type => professionnel ou amateur)
- Des commentaires sur les films (film, personnes, note, commentaire, date)

Vous devez implémenter un CRUD pour chacun de ces types de données. Vous préparerez la base en avance et la remplirez avec des catégories, des films et des acteurs, et des liens entre films et acteurs.

Votre site web devra permettre la saisie de nouveaux usagers et des commentaires qu'ils font sur le film. Vous devrez aussi pouvoir insérer de nouveau film avec les acteurs qui jouent dans le film au travers d'un fichier. La structure du fichier attendu devra bien sûr être décrite soit dans une aide, soit en préambule de la page de chargement. Vous calculerez pour chaque film la moyenne des notes pour chaque type de personnes commentant. Vous devrez aussi mettre en avant le commentaire avec le plus haute note et celui avec la plus basse note.

Vous devrez être à même de pouvoir générer une fiche de film avec le casting et l'ensemble des commentaires publiés.

Outils utilisés

PyCharm est un environnement de développement intégré utilisé pour programmer en Python.

Il permet l'analyse de code et contient un débogueur graphique. Il permet également la gestion des tests unitaires, l'intégration de logiciel de gestion de versions, et supporte le développement web avec Django.

Développé par l'entreprise tchèque JetBrains, c'est un logiciel multi-plateforme qui fonctionne sous Windows, macOS et Linux. Il est décliné

Lancement de l'environnement virtuelle

Dans le prompt :

```
> cd.\<nom_du_dossier_de_travail>\
> python -m venv venv
> .\venv\Scripts\activate
> Pip install django
> cd.\monprojet\
```

en édition professionnelle, diffusé sous licence propriétaire, et en édition communautaire diffusé sous licence Apache. (fr.wikipedia.org)

Django est un framework web open source en Python. Il a pour but de rendre le développement d'applications web simple et basé sur la réutilisation de code. Développé en 2003 pour le journal local de Lawrence (État du Kansas, aux États-Unis), Django a été publié sous licence BSD à partir de juillet 2005. (fr.wikipedia.org)

> python -m django startprojet monprojet

> cd.\monprojet\

> python manage.py startapp monapp

> python manage.py makemigration

> python manage.py migrate

> python manage.py runserver

Création d'un CRUD

Pour cela, il faut créer une base dans le dossier models.py, la rappeler dans le wiews.py et interagir dessus avec des fonctions dans forms.py. Le tout est connecté par le fichier urls.py qui renseigne tous les chemins, même ceux vers les fichiers HTML contenus dans le dossier templates de monapp.

Dans mon projet, il faut penser à renseigner le chemin vers mon app dans le fichier urls.py. Ainsi que dans le settings.py.

Création d'autre CRUD

Pour la création d'autres CRUD, il faut simplement respecter les étapes de « Création d'un CRUD ». Pas besoin de créer un nouveau monapp, il suffit de rajouter les éléments dans le fichier models.py, wiews.py, forms.py et urls.py et éventuellement de créer de nouvelles pages HTML. Ne pas oublier le « python manage.py makemigration » et « python manage.py migrate » avant de relancer le serveur pour mettre à jour la table et prendre en compte les modifications.

PROCESSUS DE CREATION DES VMs

Consignes

Ce sujet va vous permettre de fournir une interface de gestion d'une banque de films. Des usagers pourront commenter les films. Le schéma de données est le suivant

- Des catégories de film (id, nom, descriptif)
- Des films (id, titre, année de sortie, affiche, réalisateur, catégorie)
- Des acteurs (id, nom, prénom, âge, photos)
- Un lien entre les acteurs et les films
- Des personnes (id, pseudo, nom prénom, mail, mot de passe, type => professionnel ou amateur)
- Des commentaires sur les films (film, personnes, note, commentaire, date)

Vous devez implémenter un CRUD pour chacun de ces types de données. Vous préparerez la base en avance et la remplirez avec des catégories, des films et des acteurs, et des liens entre films et acteurs. /

Votre site web devra permettre la saisie de nouveaux usagers et des commentaires qu'ils font sur le film. Vous devrez aussi pouvoir insérer de nouveau film avec les acteurs qui jouent dans le film au travers d'un fichier. La structure du fichier attendu devra bien sûr être décrite soit dans une aide, soit en préambule de la page de chargement. Vous calculerez pour chaque film la moyenne des notes pour chaque type de personnes commentant. Vous devrez aussi mettre en avant le commentaire avec le plus haute note et celui avec la plus basse note.

Vous devrez être à même de pouvoir générer une fiche de film avec le casting et l'ensemble des commentaires publiés.

Outils utilisés

NGINX Open Source ou **NGINX** est un logiciel libre de serveur Web (ou HTTP) ainsi qu'un proxy inverse écrit par Igor Sysoev, dont le développement a débuté en 2002 pour les besoins d'un site russe à très fort trafic (Rambler). La documentation est disponible dans plusieurs langues. C'est depuis 2020, le serveur web le plus utilisé au monde devant Apache. (fr.wikipedia.org)

MariaDB est un système de gestion de base de données édité sous licence GPL. Il s'agit d'un embranchement communautaire de MySQL : la gouvernance du projet est assurée par la

Création du Serveur Base De Données

La création du serveur commence avec le lancement d'une VM, ici Debian 12.

Suivre les commandes suivantes :

```
# su – Passer en super user pour plus de simplicités
```

(Les installations)

```
# apt-get update
# apt install mariadb-server
# systemctl enable mariadb
```

fondation MariaDB, et sa maintenance par la société Monty Program AB, créateur du projet. Cette gouvernance confère au logiciel l'assurance de rester libre. (fr.wikipedia.org)

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au mond, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, PostgreSQL et Microsoft SQL Server. (fr.wikipedia.org)

Gunicorn, pour « Green Unicorn » (Licorne Verte), est un serveur web HTTP WSGI écrit en Python et disponible pour Unix. Son modèle d'exécution est basé sur des sous-processus créés à l'avance, adapté du projet Ruby Unicorn. Le serveur Gunicorn est compatible avec un grand nombre de frameworks web, repose sur une implémentation simple, légère en ressources et relativement rapide.(fr.wikipedia.org)

systemctl start mariadb

(La sécurité)

mysql_secure_installation

(toto ; y ; n ; n ; n ; n)

(Passer en SQL)

mariadb

(Création de la db)

> CREATE DATABASE SAE_203 CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;

(Création d'un utilisateur)

> CREATE USER 'toto'@'%' IDENTIFIED BY 'toto';

(Initialisation des privilèges utilisateur)

> GRANT ALL PRIVILEGES ON SAE_203.* TO 'toto'@'%';

> FLUSH PRIVILEGES;

> EXIT;

nano /etc/mysql/mariadb.conf.d/50-server.cnf

(Bind-address = 0.0.0.0)

systemctl restart mariadb

Création du Serveur Web

La création du serveur commence avec le lancement d'une VM, ici Debian 12.

Suivre les commandes suivantes :

(Installations depuis la racine)

apt update

install python3-pip python3-dev libmariadb-dev

(Installation de l'environnement virtuel depuis le dossier /var/www)

Installation Service Web

L'installation du service web se fait ici par le clonage du git sur lequel le projet est enregistré.

Suivre les commandes suivantes :

cd /var/www/

git clone https://github.com/utilisateur/le_repositorie.git nom_de_la_db

cd nom_de_la_db

apt install python3-venv

python3 -m venv venv

source venv/bin/activate

(Création d'un projet Django intitulé monprojet)

django-admin startproject monprojet

cd monprojet

(Création d'une application Django intitulé monapp)

python manage.py startapp monapp

(Modification du fichier settings.py)

nano settings.py

(*import os*

…

ALLOWED_HOSTS=['']*

….

DATABASES = = {

'default': {

'ENGINE': 'django.db.backends.mysql',

'NAME': 'nom_de_la_bd',

'USER': 'toto',

'PASSWORD': 'toto',

'HOST": '@ip_du_Serveur_bases_de_données',

'PORT": '3306',

}

…

SATIC_URL = '/static/'

STATIC_ROOT = os.path.join(BASE_DIR,'staticfiles'))

(Depuis la racine)

nano /etc/nginx/site_available/nom_de_la_db

(*server{*

(Si l'environnement virtuel n'est plus actif le réactiver)

source venv/bin/activate

(Installer les dernières tables et dépendances du code ex :pillow si vous avez des images)

pip install django

pip install mysqlclient

pip install pillow

(Faire les dernières migrations)

python manage.py makemigrations

python manage.py migrate

(Lancer le service web)

gunicorn –bind 127.0.0.1 :8000 monprojet.wsgi :application

Maintenant il ne vous reste plus qu'à tester le service à partir de votre navigateur web. Saisissez l'adresse IP du service web Django (127.0.0.1 : 80000). Si vous avez bien configuré les chemins dans les fichiers urls.py (de respectivement monapp et monprojet), vous devriez tout de suite tomber sur l'interface de votre site. Sinon Django affichera automatiquement (sauf si vous avez supprimé l'option) la source des erreurs et le type. Le débogage est ainsi plus simple.

Pour interagir avec les bases de données et vérifier les erreurs :

(Affichage général des db disponibles)

> SHOW DATABASES ;

(Sélection d'une table)

> USE nom_de_la_db ;

(Choix d'une tables et affichage de celle-ci)

> SHOW TABLES ;

(Description de la table)

> DESCRIBE nom_de_la_db ;

>

```
Lsiten80 ;

Server_name @ip_Server_Web

location=/favicon.ico { access_log off; log_not_found off; }

location /static/ {

root/var/www/SAE_203/chemin_vers_le_fichier_static;

}

location /media/ {

root/var/www/SAE_203/chemain_vers_le_fichier_media;


}

location / {ip a

proxy_pass http://127.0.0.1:8000;

proxy_set_header Host $host;

proxy_set_header X-Real-IP $remote_addr;

}

})
```

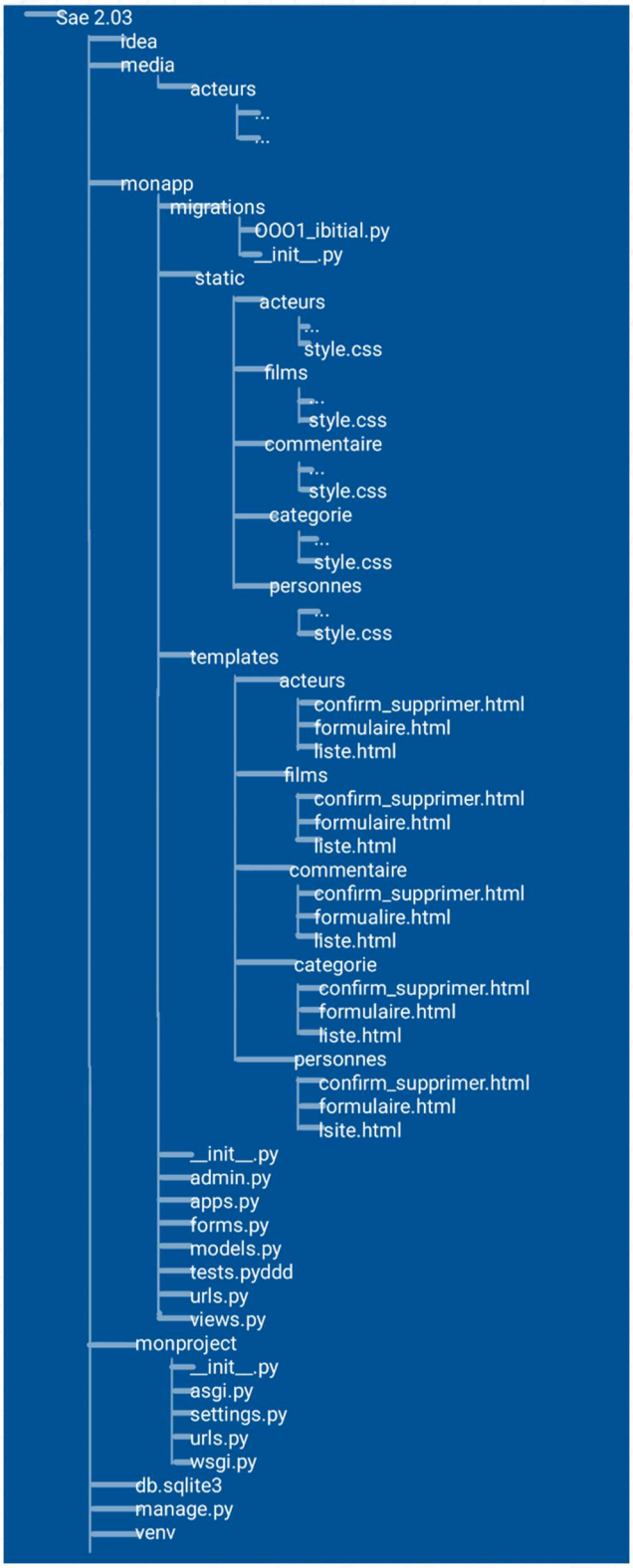
(Sélection des 10 premières lignes de la table)

> SELECT * FROM nom_de_la_table LIMIT 10 ;

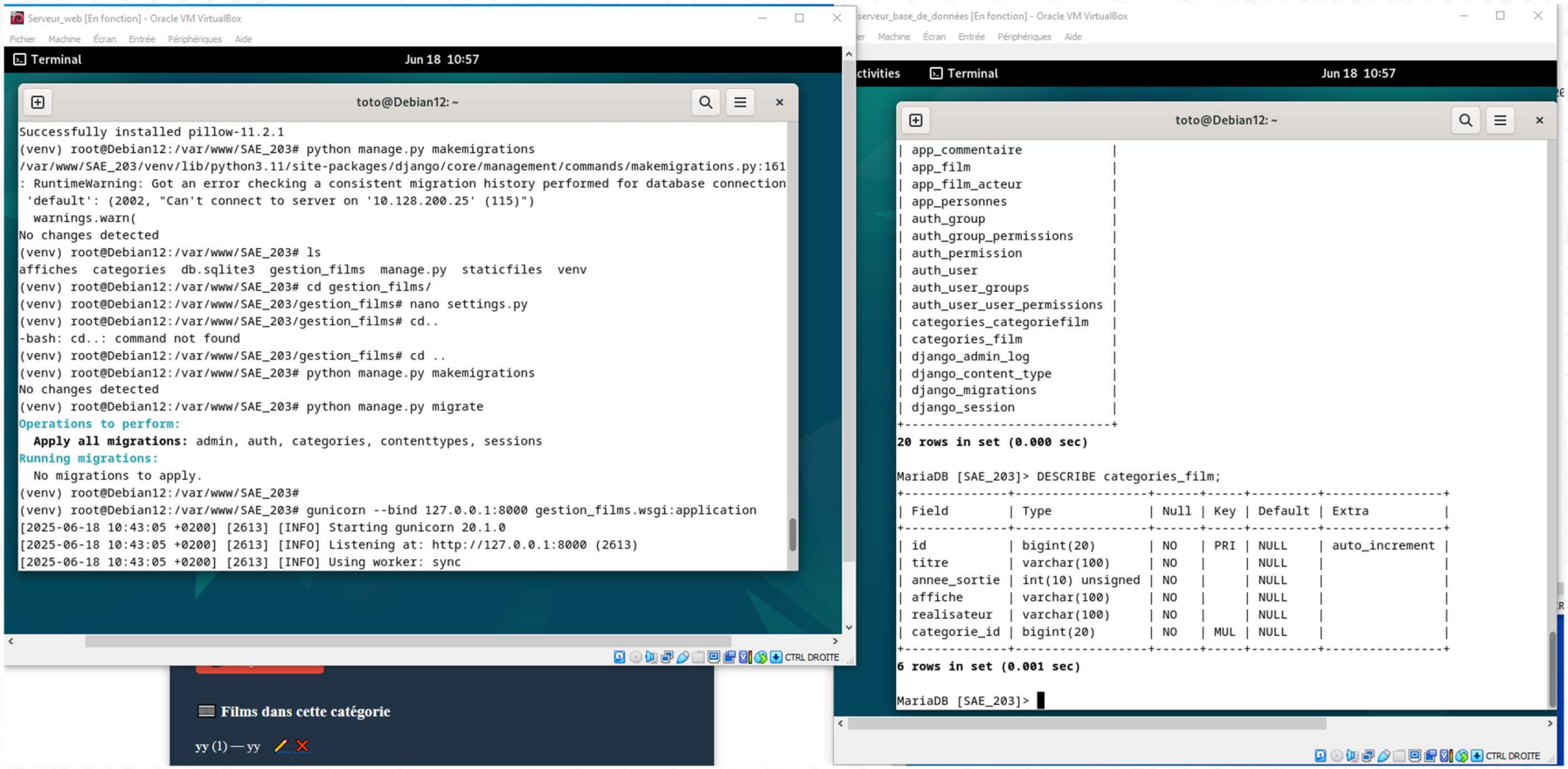
> EXIT ;

Attention les « ; » sont essentielle à la fin des commandes pour leur prises en coompte.

Aperçu structure projet git



Aperçu final possible



Sources et remercîments

Sources

Wikipedia.org

djangoproject.com

digitalocean.com

mariadb.org

nginx.org

Forums de discussions

IA en dernière nécessité

Remercîments

Professeurs

Camarades de 131 et RT1

Maurer Loic en 2ème année DevCloud.