FICHE DE PROCÉDURE D'INSTALLATION

# I. Présentation

Ce guide détaillé vous assistera dans l'installation et le déploiement de votre serveur web et de votre application web, en fournissant des instructions étape par étape pour garantir une mise en place réussie.

# II. Installation

## I.I Installation des applications nécessaires :

Pour séparer le service web et la base de données, vous devez utiliser deux machines virtuelles :

●Windows 10 pour la base de données : Configurez une VM Windows 10 pour héberger MySQL.

●Linux Debian 12 pour le serveur web : Configurez une VM Debian 12 pour héberger votre serveur web Apache et l'application Django.

Guides d'installation :

►Guide d’installation Debian 12 : https://linuxgenie.net/how-to-download-and-install-debian-12on-vmware-workstation/

►Guide d'installation Windows 10 : https://www.malekal.com/comment-creer-une-vm-vmwareworkstation-player-et-installer-windows-10/

Installation de MySQL :

>Téléchargez et installez la dernière version stable de MySQL.

>Configurez un utilisateur et un mot de passe lors de l'installation.

>Lien vers l’installation de MySQL : https://dev.mysql.com/downloads/installer/

Projet Django :

>Clonez le projet depuis GitHub. Ce dossier contient le site web dynamique nécessaire pour la production.

## I.II Installation des paquets :

Assurez-vous que toutes les fonctionnalités nécessaires sont installées. Voici les commandes à exécuter sur Linux dans cet ordre :

-`apt install python3` - Installe Python, requis pour faire fonctionner Django.

-`apt install python3-django` - Installe Django.

-`apt install python3-pip3` - Installe pip, le gestionnaire de packages Python.

-`apt install python3-virtualenv` - Installe l'environnement virtuel pour Django.

-`apt install apache2` - Installe Apache2 pour héberger le serveur web.

-`apt install apache2 libapache2-mod-wsgi-py3` - Installe des modules supplémentaires pour Apache2.

-`apt install python3-dev` - Nécessaire pour l'installation de certains packages.

-`apt install pkg-config libmariadb-dev-compat libmariadb-dev` - Nécessaire pour utiliser MySQL avec Django.

-\*\*Activer l'environnement virtuel avant d'exécuter la commande suivante\*\* : `pip3 install mysqlclient`

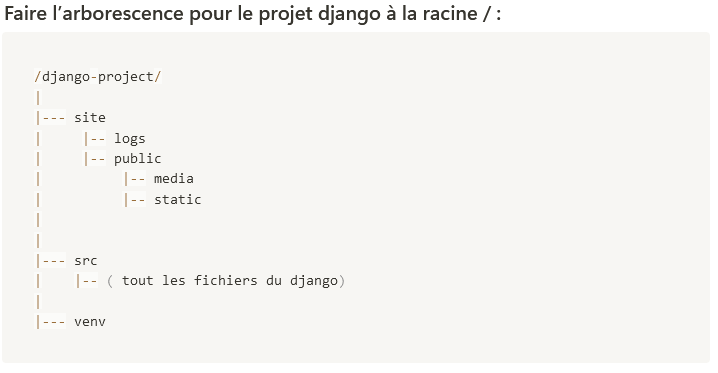
Cette section assure l'installation correcte de tous les outils et dépendances nécessaires pour faire fonctionner votre application web.

# III. Virtualisation

## III.I Virtualisation côté Linux

Pour configurer le serveur web sous Linux :

1. Accédez à la racine de la machine Linux : `cd /`
2. Créez le dossier pour le projet Django : `mkdir django-project`
3. Recréez l'arborescence suivante :



Commandes pour créer des dossiers :

* `mkdir` - Crée des dossiers.
* `cd` - Change de dossier (utiliser `cd ..` pour revenir en arrière). - `virtualenv venv -p python3` - Crée l'environnement virtuel.

Transférer les fichiers du projet Django :

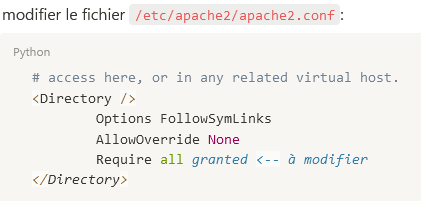
* Utilisez FileZilla pour transférer les fichiers du GitHub vers la VM Linux. - Accordez les droits aux fichiers : `chmod 777 /django-project/`.

Activer l'environnement virtuel :

* `source venv/bin/activate`

Configuration d'Apache :

* Remplacez les fichiers de configuration `apache2.conf` et `000-default.conf` avec ceux du GitHub :
* `apache2.conf` se trouve dans `/etc/apache2`.



* `000-default.conf` se trouve dans `/etc/apache2/sites-available`.



Cette section guide la configuration de l'environnement de développement sous Linux, en veillant à ce que toutes les dépendances soient correctement installées et configurées.

## III.II Virtualisation côté MySQL

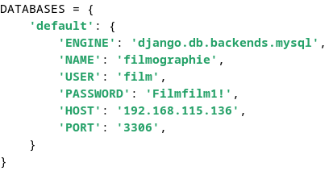
Pour configurer la base de données sous MySQL :

1. Ouvrez la ligne de commande MySQL sur la VM Windows 10.
2. Exécutez les commandes suivantes :

* `CREATE USER nom\_user;` - Crée un nouvel utilisateur.
* `CREATE DATABASE filmographie;` - Crée une nouvelle base de données nommée `filmographie`.
* `GRANT ALL PRIVILEGES ON filmographie.\* TO 'nom\_user@'%';` - Accorde tous les privilèges à l'utilisateur sur la base de données.
* `FLUSH PRIVILEGES;` - Applique les privilèges.

Vérification du fichier settings.py :

* Vérifiez que le fichier `settings.py` correspond à la base de données créée : `nano /djangoproject/src/Gestion\_film/settings.py`



-Puis dans le dossier static il faut ajouter et modifier les URLs et transférer le section static :



Cette section explique comment configurer et sécuriser la base de données MySQL, et comment vérifier la configuration de votre application Django pour qu'elle se connecte correctement à la base de données.

# IV. Finalisation

Pour finaliser l'installation, exécutez les commandes suivantes dans le répertoire `/djangoproject/src/myproject/` :

-`python3 manage.py makemigration` - Prépare les migrations pour votre base de données.

-`python3 manage.py migrate` - Applique les migrations à la base de données.

-`python3 manage.py runserver` - Lance le serveur web.

Vérifier les modifications dans la base de données :

* `USE filmographie;`
* `SHOW TABLES;`

Accédez enfin à votre site web via l'adresse locale `127.0.0.1:8000`.

Cette section vous guide à travers les étapes finales pour mettre en place votre base de données et lancer votre serveur web, vous permettant d'accéder à votre application web localement pour vérification et test.