Steps for deployment

**Step1 :**  
>sudo apt update && sudo apt upgrade -y  
**step2 :** Install Required Packages

-Python 3 & pip, venv, MySQL Server, Nginx, curl for testing

> apt install python3-pip python3-venv nginx mysql-server curl git -y

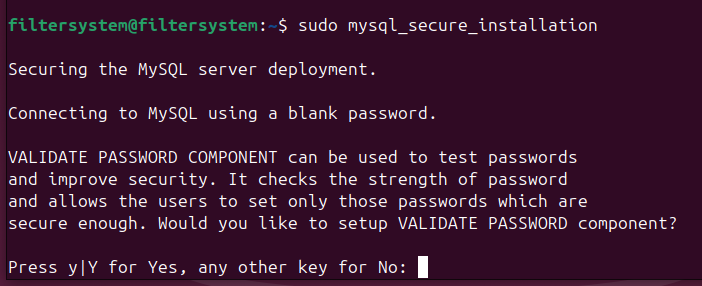
-Node.js + npm for frontend

>curl -fsSL https://deb.nodesource.com/setup\_18.x | sudo -E bash -

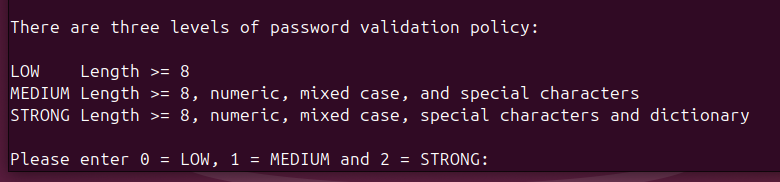
>sudo apt install -y nodejs

**Step 3:** Secure MySQL & Create Database

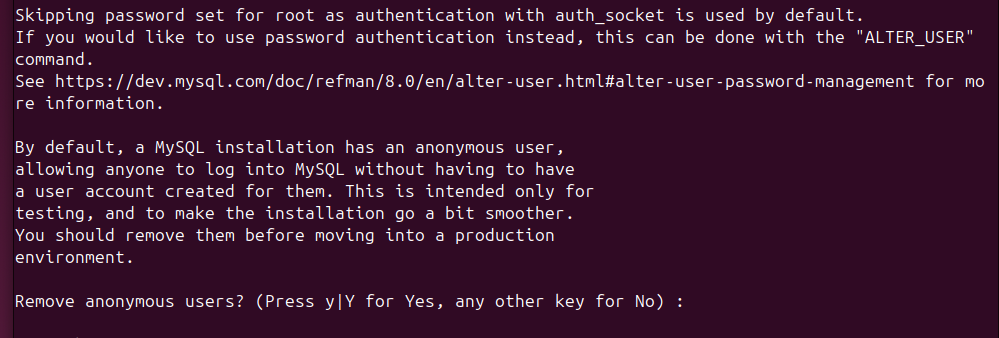
Set password for root (pwd : admin): sudo mysql\_secure\_installation



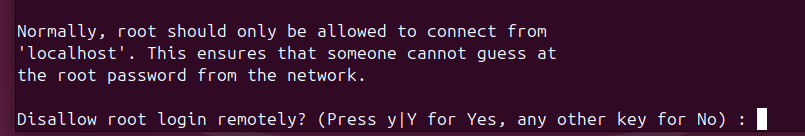
Enter y



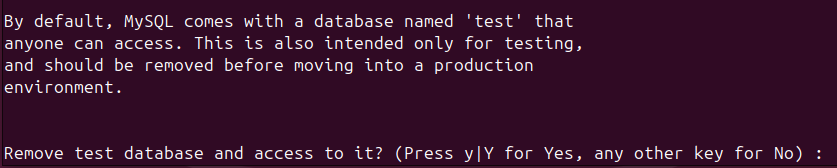
Enter 0



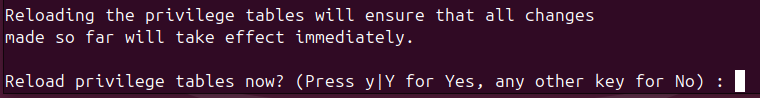
Enter y



Enter y



Enter y



Enter y

Build database:

>sudo mysql -u root -p

mysql>CREATE DATABASE filter\_db CHARACTER SET UTF8MB4 COLLATE utf8mb4\_general\_ci;

mysql > ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'Admin123!';

mysql >GRANT ALL PRIVILEGES ON filter\_db.\* TO 'root'@'localhost';

mysql >FLUSH PRIVILEGES;

mysql> EXIT;

To check if Mysql server is running :

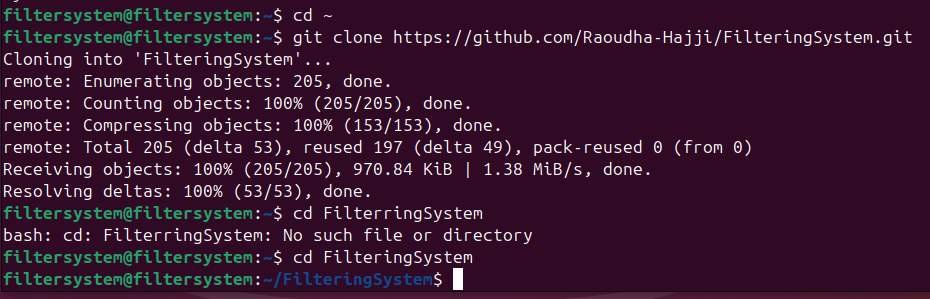
>sudo systemctl status mysql

**Step 4:** Clone Project from git to home directory:

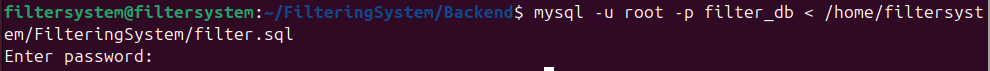
cd ~

>git clone <https://github.com/Raoudha-Hajji/FilteringSystem.git>

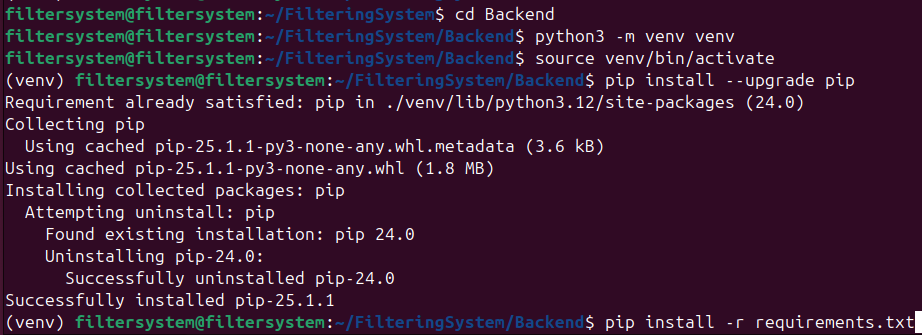
>cd FilteringSystem



Add the data to the database by importing:

> sudo mysql -u root -p filter\_db < /home/filtersystem/FilteringSystem/filter.sql  


**Step 5:** Set Up Virtual Environment (inside Backend folder)



Note: if you get an error run :  
>sudo apt install -y pkg-config libmysqlclient-dev build-essential default-libmysqlclient-dev  
>source venv/bin/activate

>pip install -r requirements.txt

**Step 6:** INSTALL OLLAMA for the llm

>curl -fsSL https://ollama.com/install.sh | sh

Enable Ollama systemd service

>sudo systemctl enable ollama

>sudo systemctl start ollama

Download the model

> ollama pull mistral

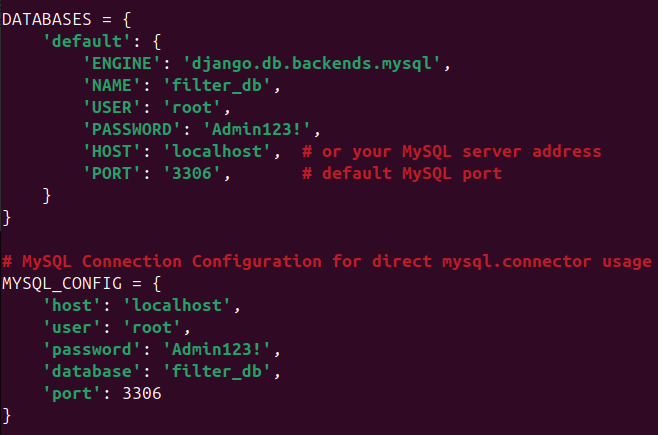
Test it

> ollama run mistral

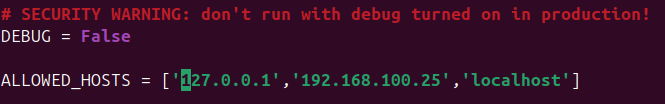
**Step 7:** Django settings

Edit settings.py from Filteringsystem/Backend/filterproject

> nano /home/filtersystem/FilteringSystem/Backend/filterproject/settings.py



-put your Mysql user and password and database name.



-Add server-IP and localhost. Set debug to false



-Configure Static files

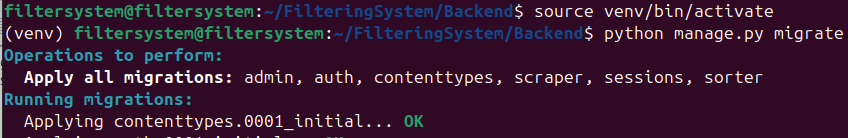
STATIC\_URL = ‘/backend-static/’

STATIC\_ROOT = BASE\_DIR / 'staticfiles'

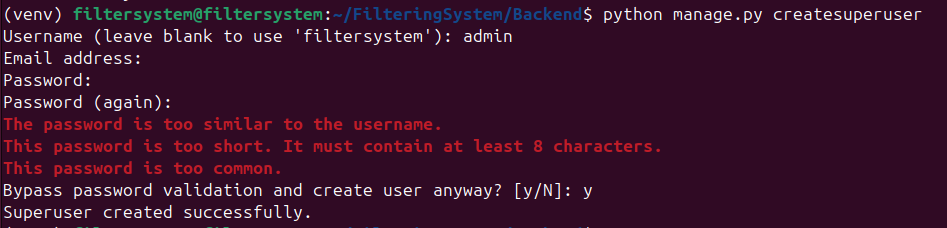
-To create superuser (the admin of the website)

> source venv/bin/activate

> python manage.py migrate



> python manage.py createsuperuser



Note: for me user is admin, pwd is admin , no email

> python manage.py collectstatic

**Step 8:** Set up system for run\_jobs.py (scheduled jobs)

>sudo nano /etc/systemd/system/filter-jobs.service  
copy paste this:

[Unit]

Description=Background job scheduler for FilterProject

After=network.target mysql.service

[Service]

User=filtersystem

Group=filtersystem

WorkingDirectory=/home/filtersystem/FilteringSystem/Backend

ExecStart=/home/filtersystem/FilteringSystem/Backend/venv/bin/python run\_jobs.py

Restart=always

[Install]

WantedBy=multi-user.target

>sudo chown filtersystem:filtersystem Backend/run\_jobs.py

>sudo chmod 775 Backend/run\_jobs.py

>sudo systemctl daemon-reload

>sudo systemctl enable filter-jobs

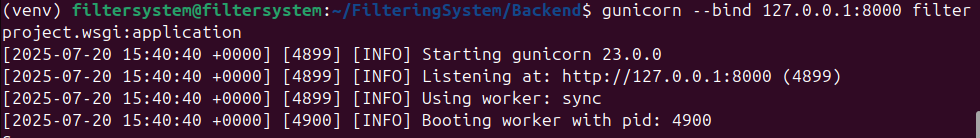
>sudo systemctl start filter-jobs

>sudo systemctl status filter-jobs

**Step 9:** install and test Gunicorn (make sure to be inside the virtual environment)

> pip install gunicorn

> gunicorn --bind 127.0.0.1:8000 filterproject.wsgi:application



**Step 10 :** Set Up Systemd Service for Gunicorn, to start it at boot

> sudo nano /etc/systemd/system/gunicorn.service

Copy paste this, make sure to use your username and folder path, and your Msql credentials:

[Unit]

Description=gunicorn daemon

After=network.target mysql.service

[Service]

User=filtersystem

Group=filtersystem

WorkingDirectory=/home/filtersystem/FilteringSystem/Backend

ExecStart=/home/filtersystem/FilteringSystem/Backend/venv/bin/gunicorn filterproject.wsgi:application --bind 0.0.0.0:8000 --workers 1 --access-logfile /home/filtersystem/FilteringSystem/Backend/logs/access.log --error-logfile /home/filtersystem/FilteringSystem/Backend/logs/error.log --log-level info

Environment="DB\_NAME=filter\_db"

Environment="DB\_USER=root"

Environment="DB\_PASSWORD=Admin123!"

Environment="DB\_HOST=localhost"

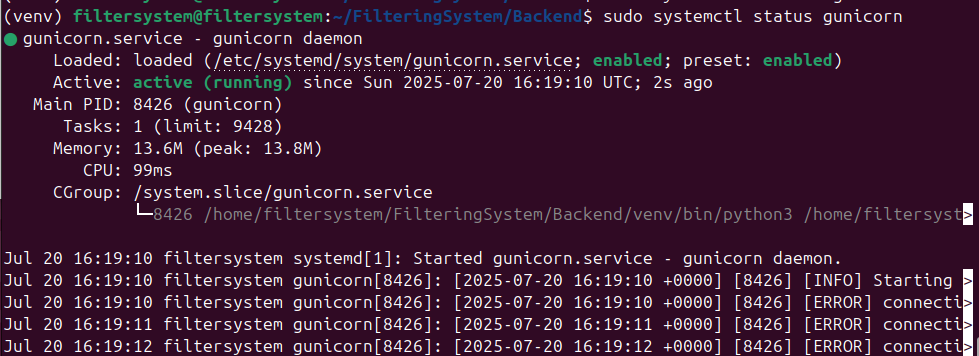
[Install]

WantedBy=multi-user.target

>sudo systemctl start gunicorn

>sudo systemctl enable gunicorn

>sudo systemctl status gunicorn



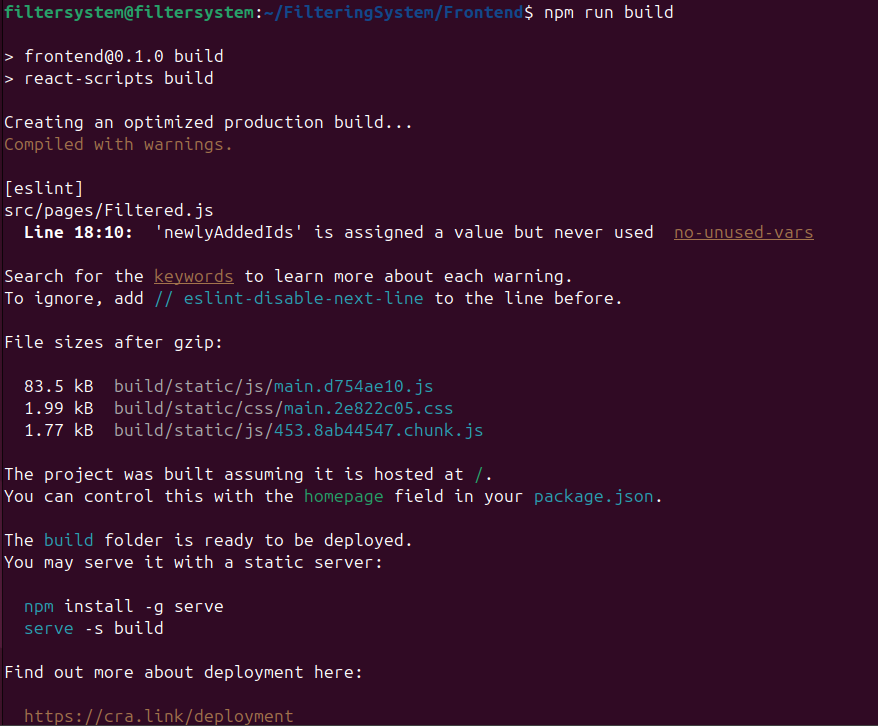
**Step 10:** Build Frontend

>cd Frontend

>npm install

> npm install material-react-table @mui/material @emotion/react @emotion/styled

>npm run build



> sudo cp -r build/\* /var/www/frontend

**Step 11:** Configure Nginx

Create a new config file:

>sudo nano /etc/nginx/sites-available/filterproject

Copy paste this :  
server {

listen 80;

server\_name 192.168.63.32;

location = /favicon.ico { access\_log off; log\_not\_found off; }

# Django static files

location /backend-static/ {

alias /home/filtersystem/FilteringSystem/Backend/staticfiles/;

}

# Django media files

location /media/ {

alias /home/filtersystem/FilteringSystem/Backend/media/;

}

# Backend API (Django via Gunicorn)

location /api/ {

include proxy\_params;

proxy\_pass <http://127.0.0.1:8000>;

}

# Sorter API endpoints (Django via Gunicorn)

location /sorter/ {

include proxy\_params;

proxy\_pass http://127.0.0.1:8000;

}

# Frontend React static files

location /static/ {

root /var/www/frontend;

}

# React frontend (serves index.html for all unmatched routes)

location / {

root /var/www/frontend;

index index.html index.htm;

try\_files $uri /index.html;

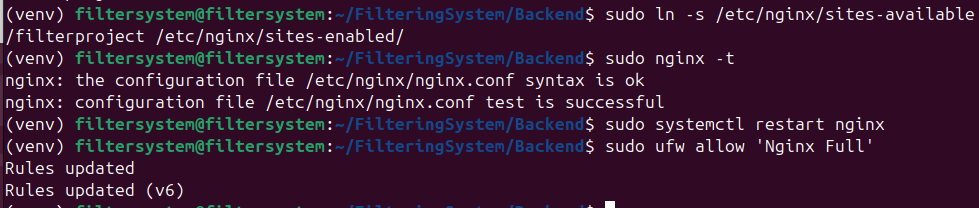
}

}

> sudo ln -s /etc/nginx/sites-available/filterproject /etc/nginx/sites-enabled/

> sudo nginx -t

> sudo systemctl restart nginx

  
  
Note, in case you make changes:

If you make changes to frontend

>cd Frontend

>npm run build

>sudo rm -rf /var/www/frontend/\*

>sudo cp -r build/\* /var/www/frontend/

>sudo systemctl restart nginx

If you make changes to backend

>cd Backend

>source venv/bin/activate

# If you changed models:

>python manage.py makemigrations

>python manage.py migrate

# If you changed views, models, or logic:

>sudo systemctl restart gunicorn

# If you changed Nginx config:

>sudo systemctl reload nginx

#

sudo systemctl daemon-reexec

sudo systemctl daemon-reload

sudo systemctl restart gunicorn

sudo systemctl status gunicorn