Dynamic Law Application Installation Guide

## [SECTION 1] Connect to the Server

ssh vendor@103.135.74.146 -p 6622

📝 Note: Replace with your actual IP and credentials if needed.

## [SECTION 2] System Preparation

sudo apt update  
sudo apt install wget ca-certificates -y

📌 Installs basic utilities for secure downloads.

## [SECTION 3] PostgreSQL Installation

sudo apt install postgresql-17 -y

Start PostgreSQL shell as the postgres user:  
sudo -u postgres psql

Inside psql, run the following SQL:

CREATE USER jodeqa WITH PASSWORD '@2103191108030702ff#DD';  
CREATE DATABASE dynamic\_law OWNER jodeqa;  
GRANT ALL PRIVILEGES ON DATABASE dynamic\_law TO jodeqa;  
\q

🛡️ Important: Replace plaintext password with a .env file or secret manager.

## [SECTION 4] Configure PostgreSQL for Remote Access

Edit postgresql.conf:  
sudo nano /etc/postgresql/17/main/postgresql.conf

Add or modify:  
listen\_addresses = '\*'

Edit pg\_hba.conf:  
sudo nano /etc/postgresql/17/main/pg\_hba.conf

Append this line:  
host all all 0.0.0.0/0 md5

Restart PostgreSQL:  
sudo systemctl restart postgresql

## [SECTION 5] Firewall Configuration

Check current firewall status:  
sudo ufw status

Delete unused rule (optional):  
sudo ufw delete allow 5432/tcp

Allow necessary ports:  
sudo ufw allow 6622/tcp # SSH  
sudo ufw allow 8000/tcp # Web App  
sudo ufw enable

⚠️ You may see Python warnings; these are safe to ignore.

## [SECTION 6] Python 3.12 Installation

sudo add-apt-repository ppa:deadsnakes/ppa  
sudo apt update  
sudo apt install python3.12 python3.12-venv -y

🧠 Python 3.12 is needed for compatibility with your app.

## [SECTION 7] Project Setup

mkdir -p dynamic\_law/{\_doc,routes,services,templates,static,uploaded\_files}  
cd ~/dynamic\_law

Create virtual environment:  
python3.12 -m venv venv  
source venv/bin/activate

Install build essentials and DB libraries:  
sudo apt install build-essential libpq-dev -y

Install Python dependencies:  
pip install -r requirements.txt  
pip install gunicorn

## [SECTION 8] Run the App with Gunicorn

gunicorn -w 4 -b 0.0.0.0:8000 run:app

✅ Replace run:app if your app entry point is different (e.g., main:app)

## Final Notes

- Consider using systemd to manage Gunicorn as a service  
- Use .env files to manage secrets  
- Restrict PostgreSQL access to trusted IP ranges only

## [SECTION 9] NGINX Configuration (Reverse Proxy)

Install NGINX:  
sudo apt install nginx -y

Create a configuration file for the app:  
sudo nano /etc/nginx/sites-available/dynamic\_law

Paste the following content:

server {  
 listen 80;  
 server\_name your\_domain\_or\_ip;

location / {  
 proxy\_pass http://127.0.0.1:8000;  
 proxy\_set\_header Host $host;  
 proxy\_set\_header X-Real-IP $remote\_addr;  
 proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  
 proxy\_set\_header X-Forwarded-Proto $scheme;  
 }  
}

Enable the site and restart NGINX:  
sudo ln -s /etc/nginx/sites-available/dynamic\_law /etc/nginx/sites-enabled/  
sudo nginx -t  
sudo systemctl restart nginx

## [SECTION 10] Running Gunicorn with systemd

Create a Gunicorn systemd service file:  
sudo nano /etc/systemd/system/dynamic\_law.service

Paste the following content:

[Unit]  
Description=Gunicorn instance to serve dynamic\_law  
After=network.target

[Service]  
User=vendor  
Group=www-data  
WorkingDirectory=/home/vendor/dynamic\_law  
Environment="PATH=/home/vendor/dynamic\_law/venv/bin"  
ExecStart=/home/vendor/dynamic\_law/venv/bin/gunicorn -w 4 -b 127.0.0.1:8000 run:app

[Install]  
WantedBy=multi-user.target

Enable and start the service:  
sudo systemctl daemon-reexec  
sudo systemctl daemon-reload  
sudo systemctl enable dynamic\_law  
sudo systemctl start dynamic\_law  
sudo systemctl status dynamic\_law