**🚀 Hosting Your Django Application on a Spare Laptop Without VS Code**

To run your Django app as a **background service** (without keeping a console open), follow these steps.

**🛠 Step 1: Set Up Django on Your Spare Laptop**

1. **Ensure Python is Installed**
   * Check Python version:

sh

CopyEdit

python --version

* + If not installed, download from [python.org](https://www.python.org/downloads/) and install.

1. **Set Up a Virtual Environment**  
   Navigate to your project directory and create a virtual environment:

sh

CopyEdit

python -m venv env

Activate it:

* + **Windows (Command Prompt or PowerShell)**

sh

CopyEdit

env\Scripts\activate

* + **Linux/macOS**

sh

CopyEdit

source env/bin/activate

1. **Install Dependencies**  
   Ensure all Django dependencies are installed:

sh

CopyEdit

pip install -r requirements.txt

**🛠 Step 2: Run Django as a Background Service**

**🔹 Option 1: Use screen (For Linux)**

If you're using **Linux**, you can run Django in the background using screen:

sh

CopyEdit

screen -S django-server

python manage.py runserver 0.0.0.0:8000

Press Ctrl + A, then D to detach from the screen session and keep the server running.

To reattach later:

sh

CopyEdit

screen -r django-server

**🔹 Option 2: Use Windows Task Scheduler**

1. **Open Task Scheduler** (taskschd.msc from Run or Start menu).
2. **Create a new task**:
   * Go to **Action > Create Basic Task**.
   * Name it **Django Server**.
   * Under **Triggers**, select **At Startup**.
   * Under **Actions**, choose **Start a Program**.
   * **Browse** to python.exe (inside your virtual environment).
   * In the **Arguments** field, add:

CopyEdit

manage.py runserver 0.0.0.0:8000

* + In **Start In**, set your Django project directory.
  + Click **Finish** and enable it.

**🔹 Option 3: Run Django as a Windows Service**

You can use **NSSM (Non-Sucking Service Manager)** to create a Windows service:

1. **Download NSSM** from nssm.cc.
2. **Install Django as a Service**:

sh

CopyEdit

nssm install DjangoServer

* + Set Application to your **Python executable** (inside your virtual env).
  + Set Arguments to:

CopyEdit

manage.py runserver 0.0.0.0:8000

* + Set Start Directory to your Django project folder.
  + Click **Install Service**.

Start the service with:

sh

CopyEdit

nssm start DjangoServer

To stop it:

sh

CopyEdit

nssm stop DjangoServer

**🛠 Step 3: Configure NGROK for External Access**

Since **ngrok is already installed**, create an **ngrok tunnel** to expose your Django app publicly.

Run:

sh

CopyEdit

ngrok http 8000

This will generate a **public URL** that you can use to access your Django app from any device.

**🔹 Make ngrok Run at Startup (Windows)**

1. Open **Task Scheduler** (taskschd.msc).
2. **Create a New Task**:
   * Under **Triggers**, select **At Startup**.
   * Under **Actions**, choose **Start a Program**.
   * Set **Program** to:

vbnet

CopyEdit

C:\path\to\ngrok.exe

* + In **Arguments**, enter:

yaml

CopyEdit

http 8000

* + Click **OK**.

Now **ngrok will automatically start on boot**, exposing your Django app to the web.

**🎯 Summary of Setup**

1. **Run Django as a Background Service**
   * Use **screen** (Linux) or **Task Scheduler/NSSM** (Windows).
2. **Expose Your Server with ngrok**
   * Use ngrok http 8000 to generate a public link.
3. **Ensure the Laptop Runs at Startup**
   * Use **Task Scheduler** to auto-start Django & ngrok.

🚀 **Now your Django app will run without needing VS Code or a terminal open!** Let me know if you need further refinements. 😊🔥