DEPLOYMENT DOCUMENTATION

Download & install winscp, putty

Deployment details

There is a WinSCP column that contains the username, IP address, and password. Use these credentials to log in to WinSCP.

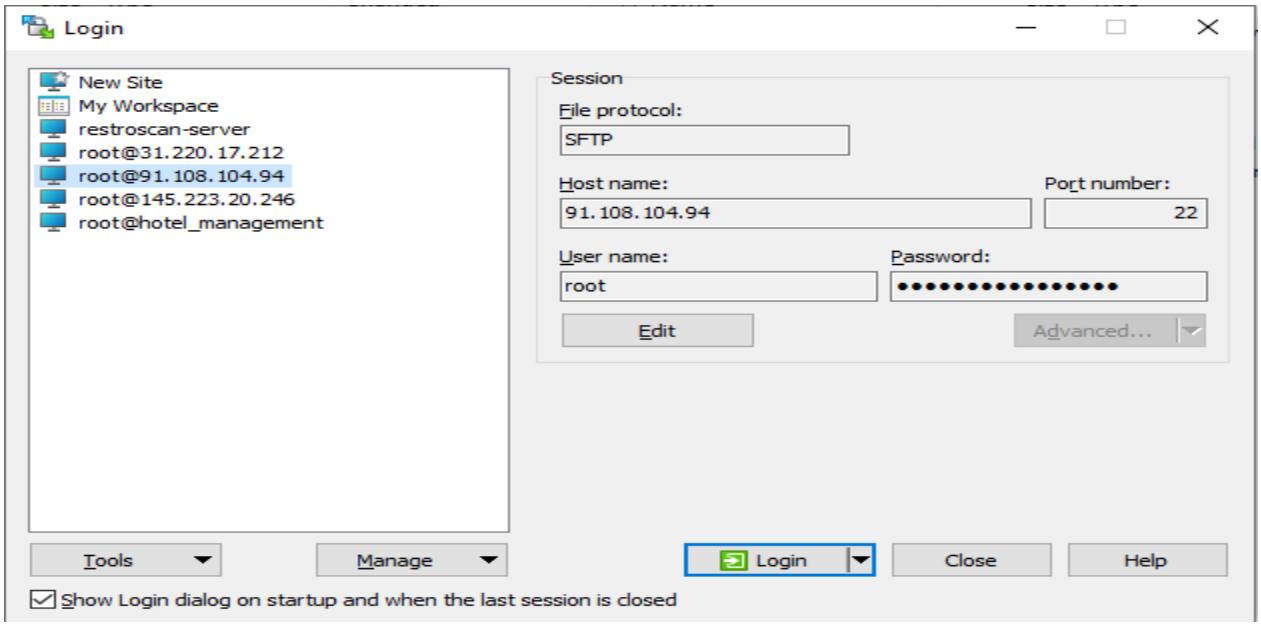
I have kept local code in the LHS (left-hand side) and server code in the RHS (right-hand side), so we will use this setup throughout the entire document.

- Most of the projects are hosted on the 91.108.104.94 server. (Adyant server)
- Xenia Hotel is hosted on 145.223.20.247.
- GPLLC is hosted on 31.220.17.212.
- 9M is hosted on 145.223.20.246.

For frontend

- First, we pull the latest code from Git.
- Then, we copy all the folders and paste them into our local.
- After that, we install the Node modules for the React application.
- Then, we add the domain name (for example, https://adyant.co.in) in the React application's configuration file (for example, config.jsx).
- After that, we run the command npm run build.
- Finally, we upload all the contents of the build folder to the project's directory on the server, located at /usr/share/nginx/html/project_directory

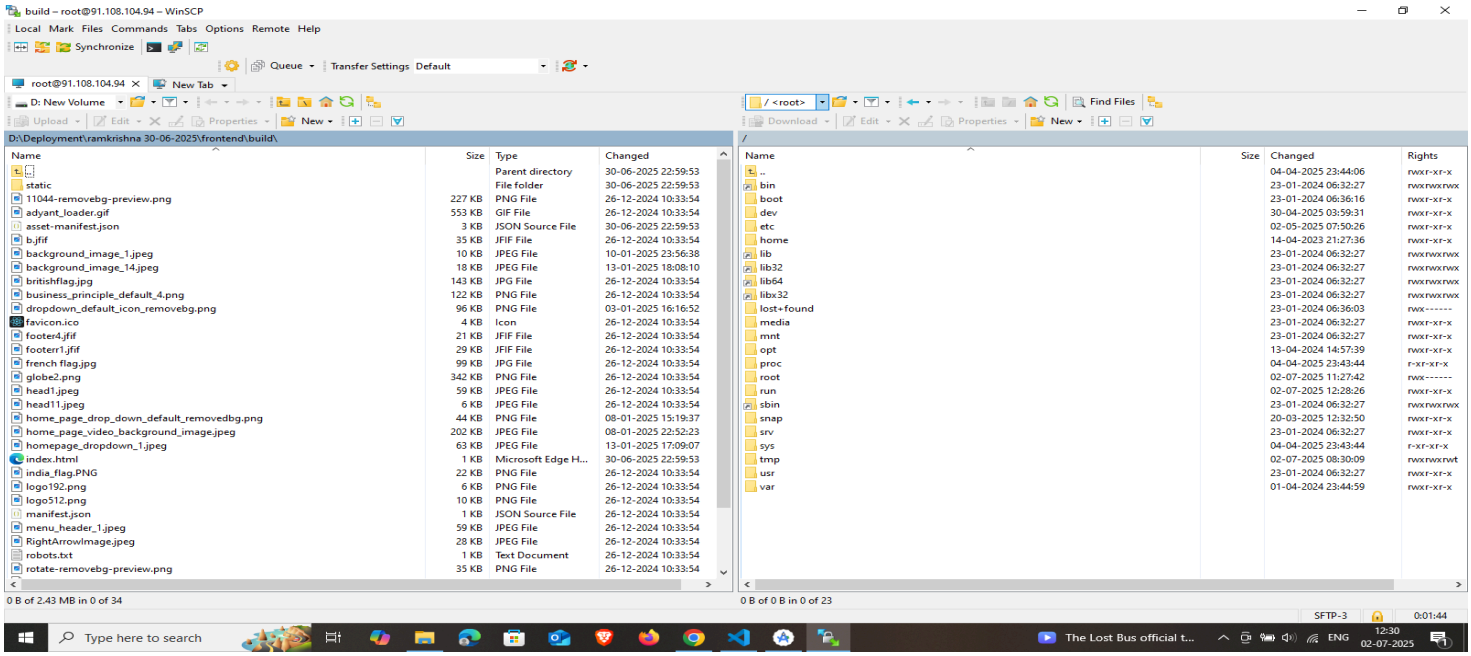
After a successful login in winscp



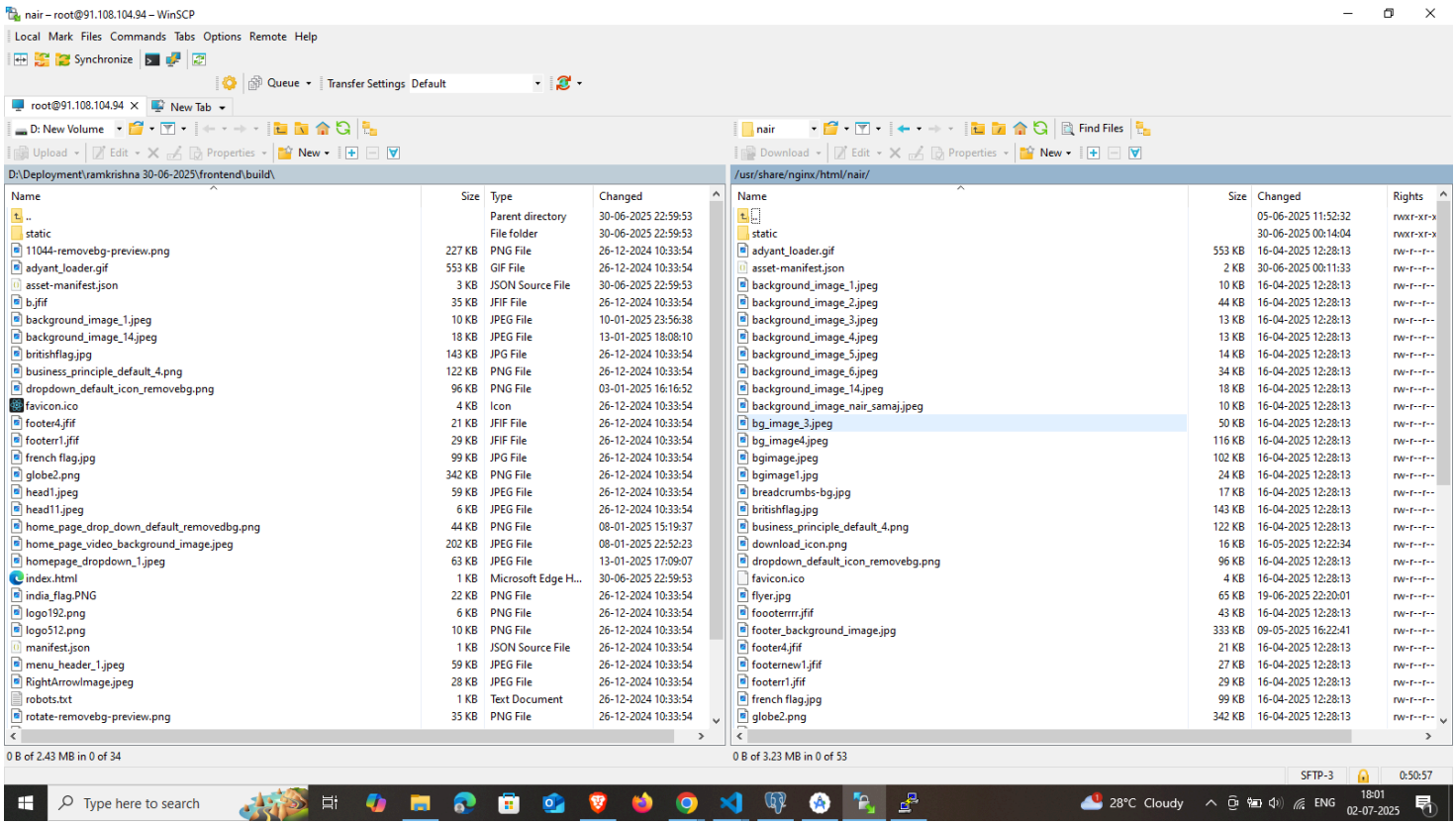
We navigate to the directory

<root> usr/share/nginx/html

Inside this directory, we will find folders for the respective projects.
(Check the below SS and see RHS section)



To deploy a specific project, we need to merge the build folder of our React app into the corresponding project folder. (Check below SS)



For backend

In settings.py, find the middleware section.

After 'django.middleware.common.CommonMiddleware', add 'whitenoise.middleware.WhiteNoiseMiddleware'.

Change the database name and other details as needed.

Add CSRF_COOKIE_SECURE = True
Add CSRF_TRUSTED_ORIGINS = [].

For example, set CSRF_TRUSTED_ORIGINS=['https://raipurnairsamajam.com', 'http://raipurnairsamajam.com'].

Example MIDDLEWARE list:

```
MIDDLEWARE = [
    "corsheaders.middleware.CorsMiddleware",
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'whitenoise.middleware.WhiteNoiseMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

Example DATABASES settings:

```
DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.postgresql",
        "NAME": "nair",
        "USER": "admin",
        "PASSWORD": "Adyant@0122",
        "HOST": "localhost",
        "PORT": "5432",
    }
```

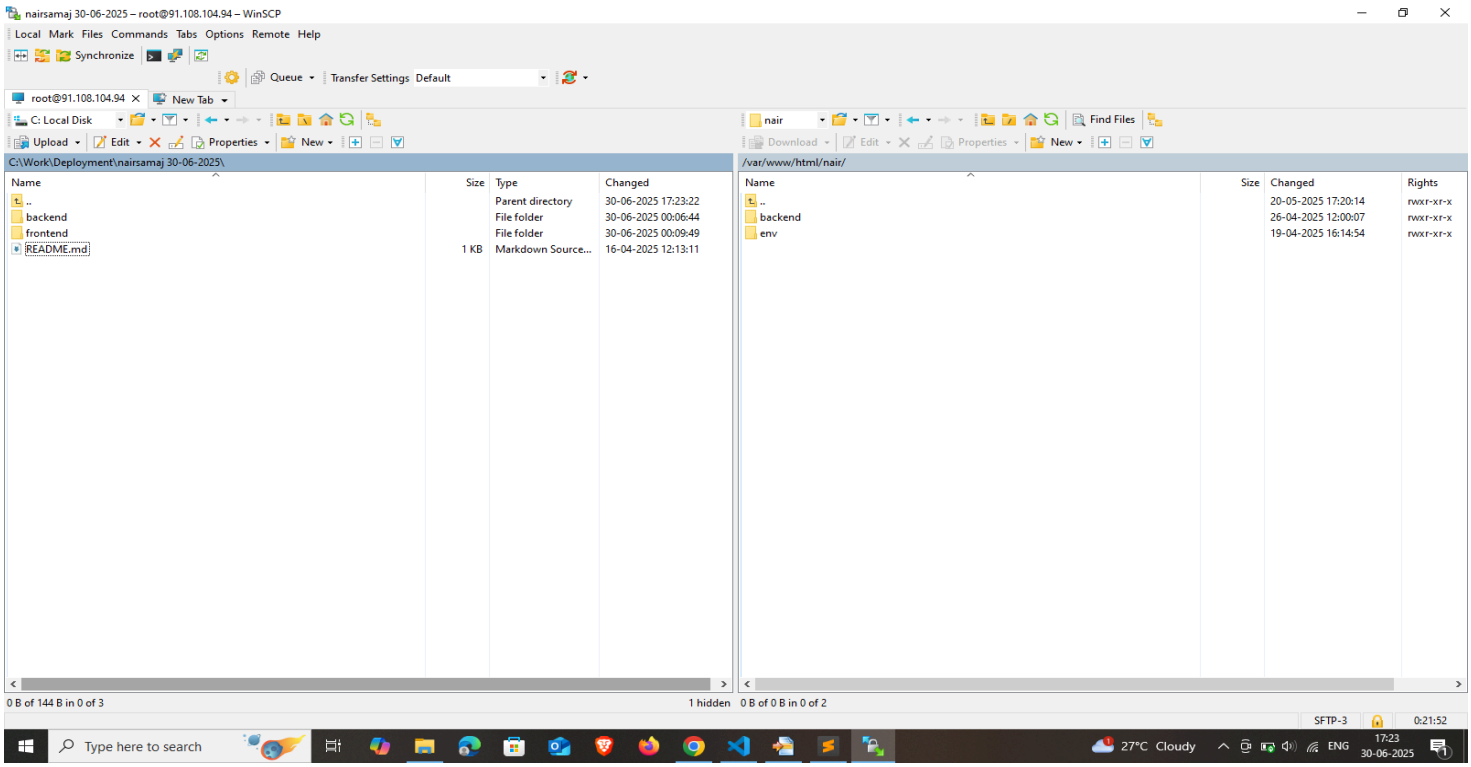
}
Change SITE_URL to your domain, for example: SITE_URL = "https://raipurnairsamajam.com/".
Change static_url = ‘static/’ to static_url = ‘staticfiles/’
And add
STATICFILES_DIRS = [
    BASE_DIR / "static",
]

In requirements.txt, add:

- psycopg2-binary==2.9.5
- whitenoise==6.0.0
- gunicorn==20.1.0

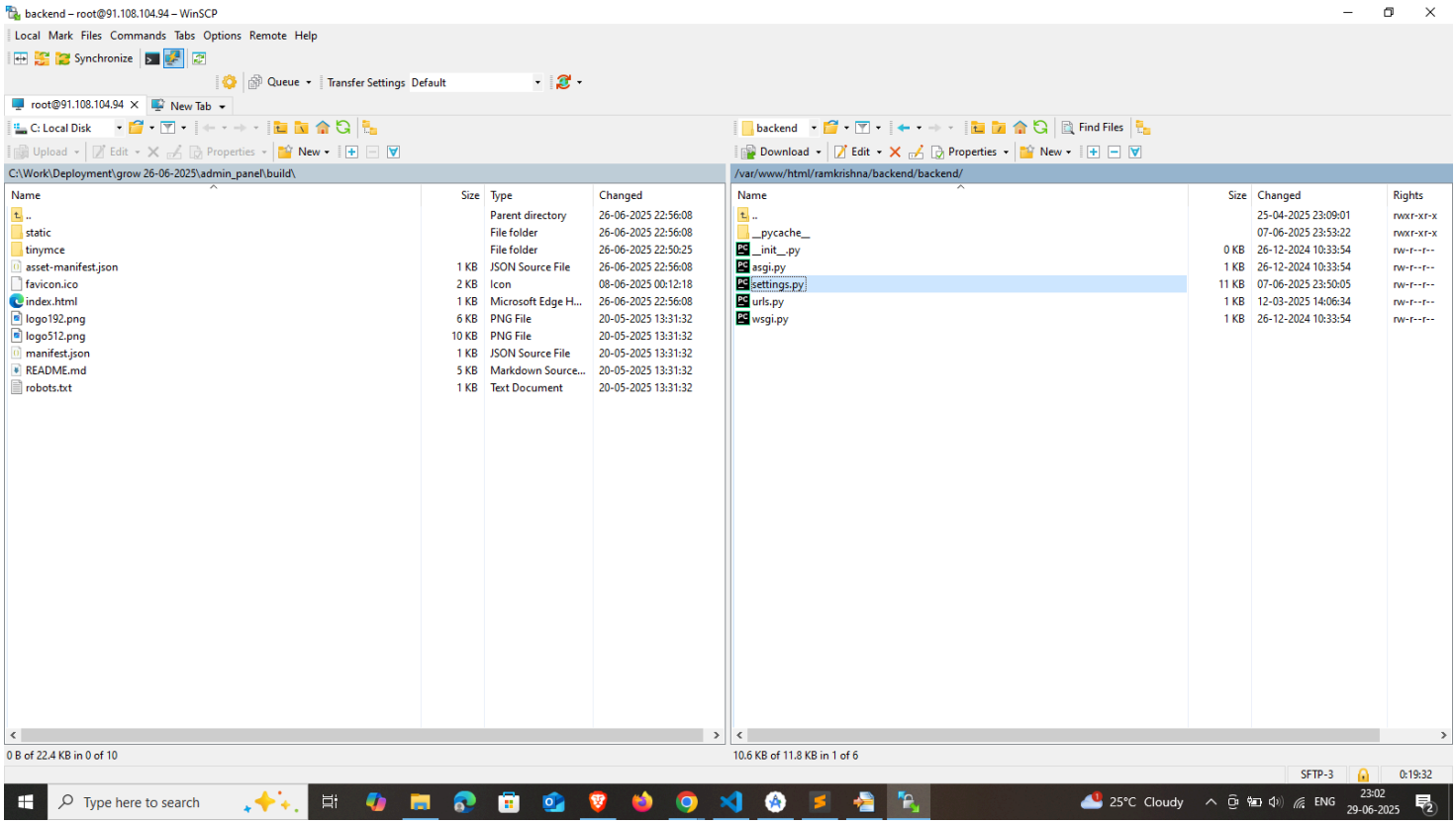
Keep the migrations in sequence.

Upload the entire backend folder to /var/www/html/project_directory (Take reference from below SS)



Now to open putty

Please login to winscp using ip and password and open putty (on top left side, below Tabs)



Give password Adyant@0122 or Adyant@012024 (Please check the deployment sheet also as mentioned in starting of the document)

For backend

Navigate using below commands

- ➔ Navigate to root directory and web directory:
  - ◆ cd /root/
  - ◆ cd /var/www/html/

- ➔ create folder for project-directory
  - ◆ mkdir project_directory_name (eg mkdir nair)

- ➔ navigate to directory
  - ◆ cd project_directory_name/ (eg cd nair/)

- ➔ create virtual environment
  - ◆ python -m venv env or python3 -m venv env

- ➔ In the same directory activate virtual environment

- source env/scripts/activate/

➔ Then will navigate to
- cd backend/
- Install the libraries pip install -r requirements.txt
  (we can check the installed libraries using pip freeze)

Create new database (Check the below SS)

```
Last login: Sun Jun 29 18:41:24 2025 from 152.59.24.31
root@srv502469:~# cd /root/
root@srv502469:~# cd /var/www/html/nair/
root@srv502469:/var/www/html/nair# source env/bin/activate
(env) root@srv502469:/var/www/html/nair# cd backend/
(env) root@srv502469:/var/www/html/nair/backend# su - postgres
postgres@srv502469:~$ psql
psql (15.5 (Ubuntu 15.5-0ubuntu0.23.04.1))
Type "help" for help.

postgres=# create database database_name;
```

1. Switch to the postgres user:
   su - postgres

2. Access the PostgreSQL prompt:
   psql

3. Create a new database:
   create database database_name; (e.g. create database nair;)

4. Connect to the newly created database:
   \c database_name (e.g. \c nair)

5. Grant privileges to the admin user:
   GRANT CREATE ON SCHEMA public TO admin;

6. Exit the PostgreSQL prompt:
   Press Ctrl + D

7. Exit from the postgres user session:
   Press Ctrl + D again

```
Last login: Wed Jul  2 05:56:15 2025 from 152.59.26.37
root@srv502469:~# cd /root/
root@srv502469:~# cd /var/www/html/nair/
root@srv502469:/var/www/html/nair# source env/bin/activate
(env) root@srv502469:/var/www/html/nair# cd backend/
(env) root@srv502469:/var/www/html/nair/backend#
```

1. Ensure you are inside the backend directory:
   cd backend/

2.Run database migrations:
   python manage.py migrate

3. Create a Django superuser:

python manage.py createsuperuser
Follow the prompts to enter username, email, and password.

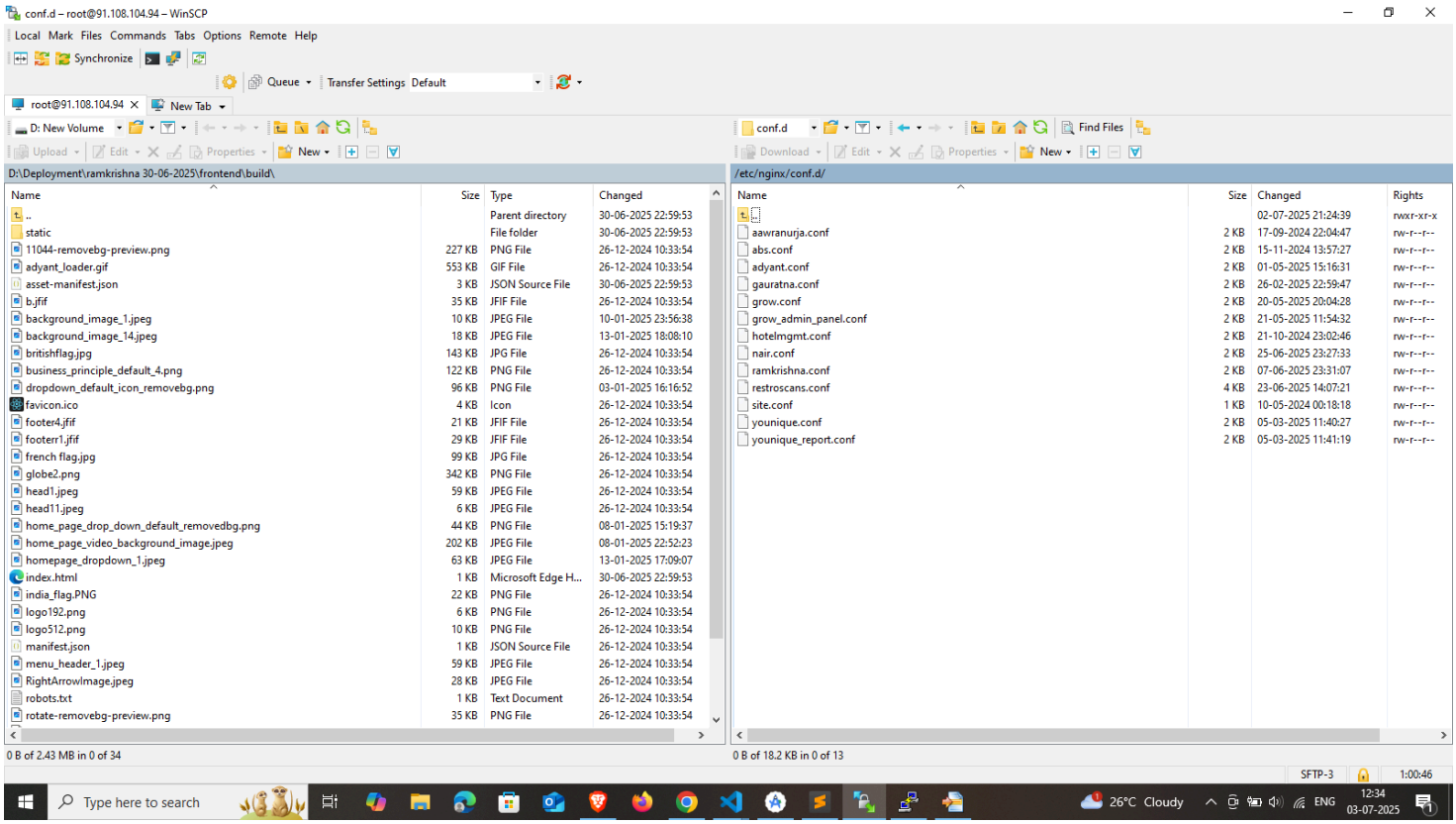
4. Start the server using Gunicorn (in the background):
gunicorn --workers 3 --bind 0.0.0.0:port_number backend.wsgi --daemon
(eg gunicorn --workers 3 --bind 0.0.0.0:8013 backend.wsgi --daemon)
Replace port_number with your desired port (e.g. 8013).

5. Restart the Nginx service to apply changes:
service nginx restart

Note: Make sure to update the port number if required.

For SSL certificate generation

Open winscp (then navigate to nginx configuration folder)
/etc/nginx/conf.d/



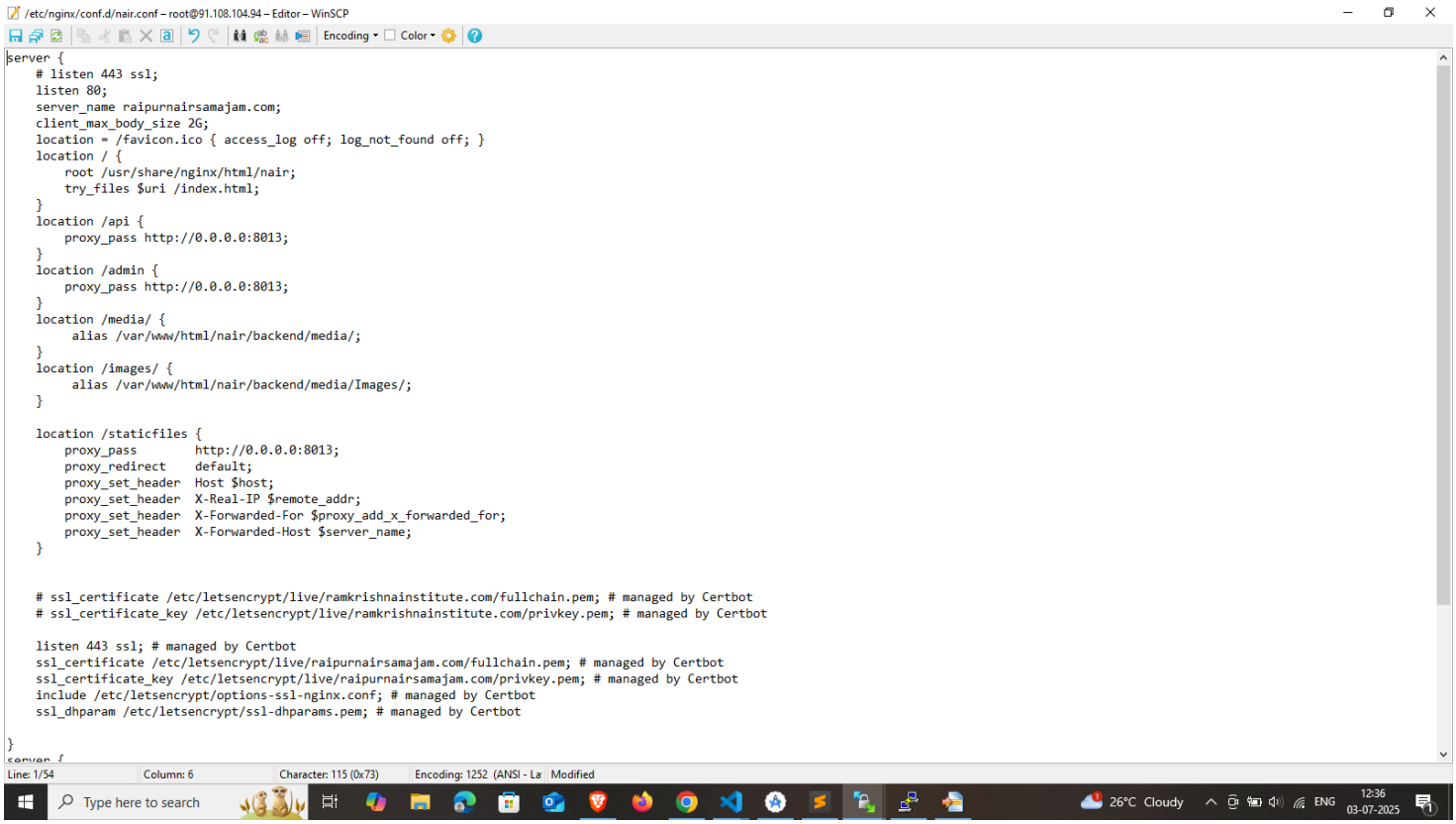
Here create .conf file. Take reference from any previous file (say nair.conf)

- ➔ Change the server_name
  - ◆ Update the domain name

- ➔ Path of backend & frontend directory
  - ◆ location /
  - ◆ location /media/
  - ◆ location /images/

- ➔ Update the port number
  - ◆ location /api
  - ◆ location /admin
  - ◆ location /staticfiles (in proxy_pass)

Remove the ssl_certificate /etc/letsencrypt/live/raipurnairsamajam.com/fullchain.pem; and ssl_certificate_key /etc/letsencrypt/live/raipurnairsamajam.com/privkey.pem;



Open putty (give password)

- ➔ Navigate to the specific directory
  - cd /root/
  - cd /var/www/html/
  - Then navigate to specific directory (eg cd nair/)
  - Activate environment (source env/bin/activate)
  - Then run command history|grep cerbot (you will see list of commands)

Below is the command used to generate SSL certificate
certbot --nginx -d raipurnairsamajam.com -d www.raipurnairsamajam.com
(Change the domain name)
This command will create ssl_certificate & ssl_certificate_key in .conf file (nginx configuration file)

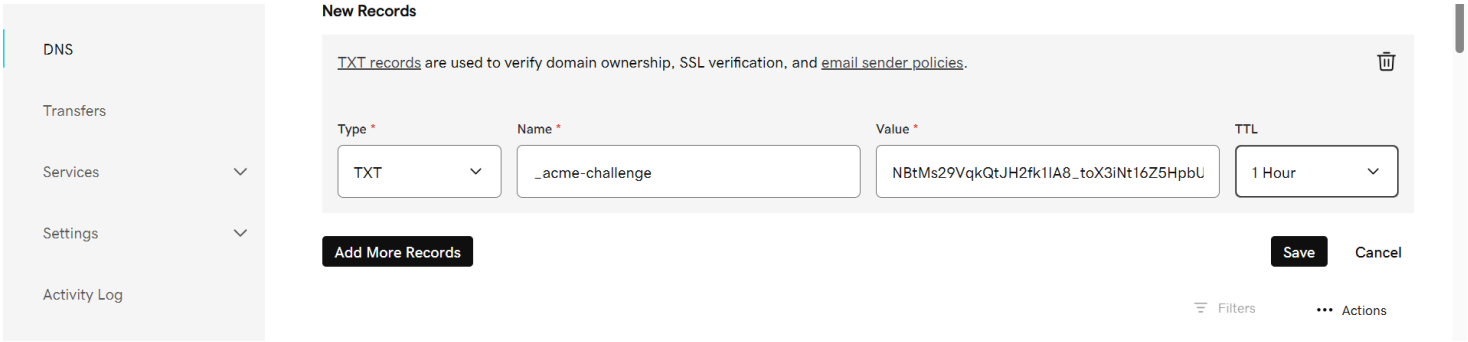
```
(env) root@srv502469:/var/www/html/nair/backend# history|grep certbot
 1067  history|grep certbot
 1088  history|grep certbot
 1151  history|grep certbot
 1152  certbot certonly  --agree-tos --email info@adyant.co.in --manual --preferred-challenges=dns -d *.restroscans.com --server https://acme-v02.api.letsencrypt.org/directory -v
 1198  ps -ef|grep certbot
 1199  history -ef|grep certbot
 1200  history | grep certbot
 1203  history|grep certbot
 1899  history|grep certbot
 1900  certbot --nginx -d raipurnairsamajam.com -d www.raipurnairsamajam.com
 1914  certbot --nginx -d raipurnairsamajam.com -d www.raipurnairsamajam.com
 2003  history|grep certbot
 2007  history|grep certbot
(env) root@srv502469:/var/www/html/nair/backend#
```

```
(env) root@srv631326:/var/www/html/aams/backend# certbot certonly  --agree-tos --email info@adyant.co.in --manual --preferred-challenges=dns -d *.moreshop.co.in --server https://acme-v02.api.letsencrypt.org/directory -v
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator manual, Installer None
Requesting a certificate for *.moreshop.co.in
Performing the following challenges:
dns-01 challenge for moreshop.co.in

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Please deploy a DNS TXT record under the name:

_acme-challenge.moreshop.co.in.

with the following value:

Ysu3CsKziozyn6yrRSUz8bK3GKX_M_Df8v4UI8M2YgA

Before continuing, verify the TXT record has been deployed. Depending on the DNS
provider, this may take some time, from a few seconds to multiple minutes. You can
check if it has finished deploying with aid of online tools, such as the Google
Admin Toolbox: https://toolbox.googleapps.com/apps/dig/#TXT/_acme-challenge.moreshop.co.in.
Look for one or more bolded line(s) below the line ';ANSWER'. It should show the
value(s) you've just added.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Press Enter to Continue
Waiting for verification...
Cleaning up challenges

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/moreshop.co.in/fullchain.pem
Key is saved at:         /etc/letsencrypt/live/moreshop.co.in/privkey.pem
This certificate expires on 2025-10-23.
These files will be updated when the certificate renews.

NEXT STEPS:
- This certificate will not be renewed automatically. Autorenewal of --manual certificates requires the use of an authentication hook script (--manual-auth-hook) but one was not provided. To renew this certificate, repeat this same certbot command before the certificate's expiry date.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
 * Donating to EFF:                    https://eff.org/donate-le
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(env) root@srv631326:/var/www/html/aams/backend# service nginx restart
(env) root@srv631326:/var/www/html/aams/backend# service nginx restart
(env) root@srv631326:/var/www/html/aams/backend#
```

For multiple domain names (tenant project)

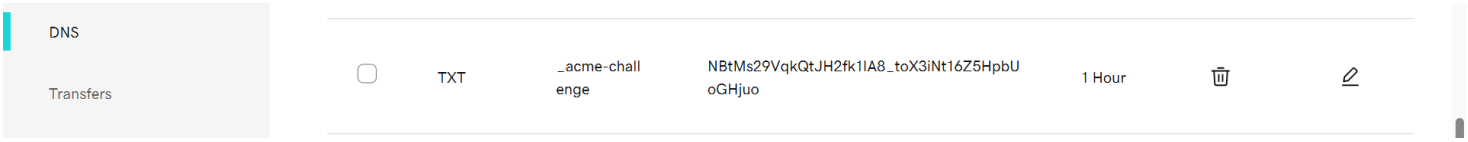
certbot certonly --agree-tos --email info@adyant.co.in --manual --preferred-challenges=dns -d *.restroscans.com --server https://acme-v02.api.letsencrypt.org/directory -v
Change the domain name restroscans.com
(For reference check above SS)

To add a new DNS record in GoDaddy, go to the domain 'retroscans.com', click on 'DNS', and then select 'Add New Record'.

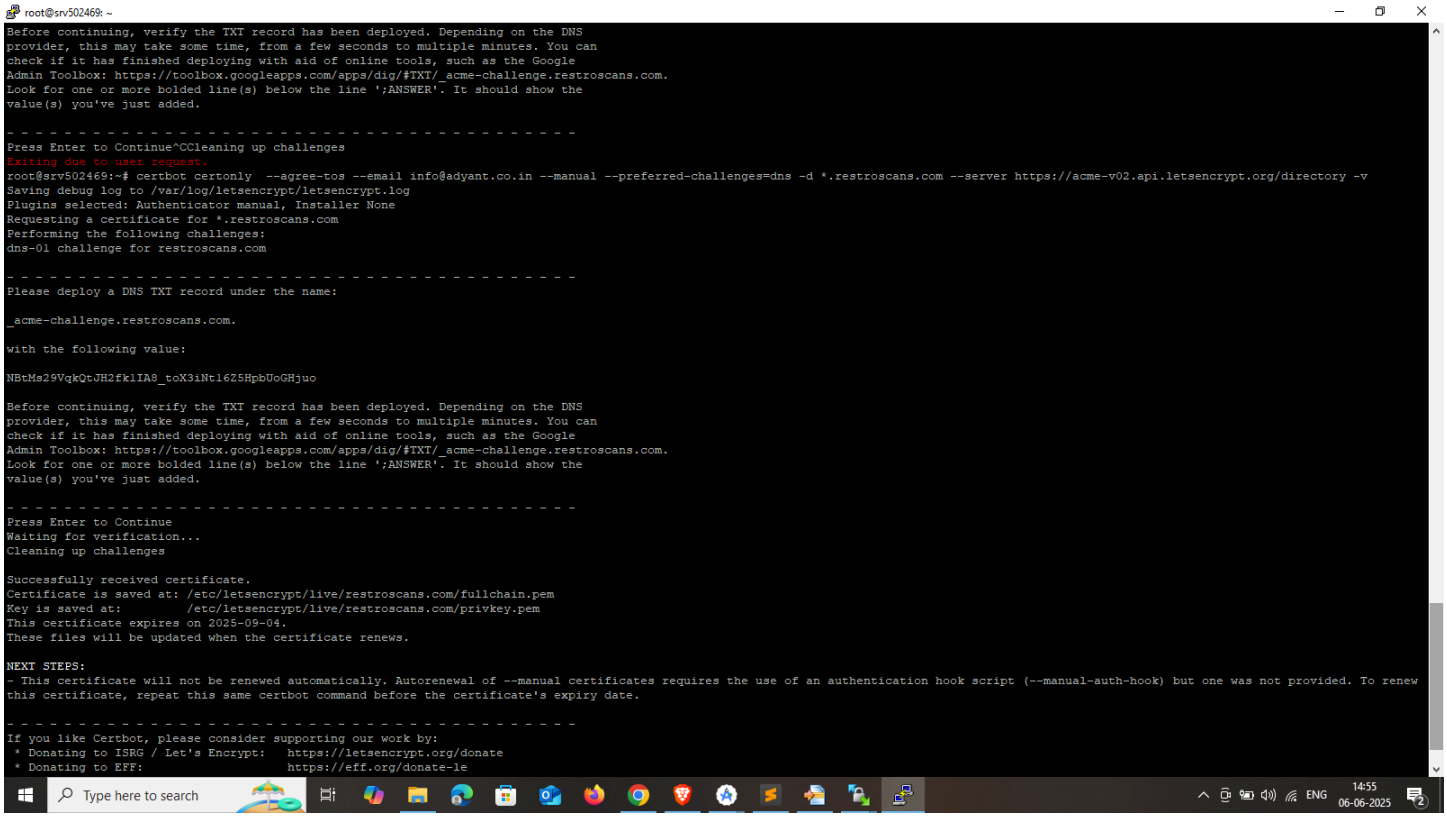


From the Type dropdown, select TXT. In the Name field, enter _acme-challenge, and in the Value field, paste the following value from the PuTTY terminal:
NBtMs29VqkQtJH2fk1IA8_toX3iNt16Z5HpbUoGHjuo
Finally, set the TTL to 1 Hour.
(Please check above SS)

Then save the details



Then in putty click on Enter to continue



Please note down the date when certificate will expire

If a project is already deployed and is being deployed again, then

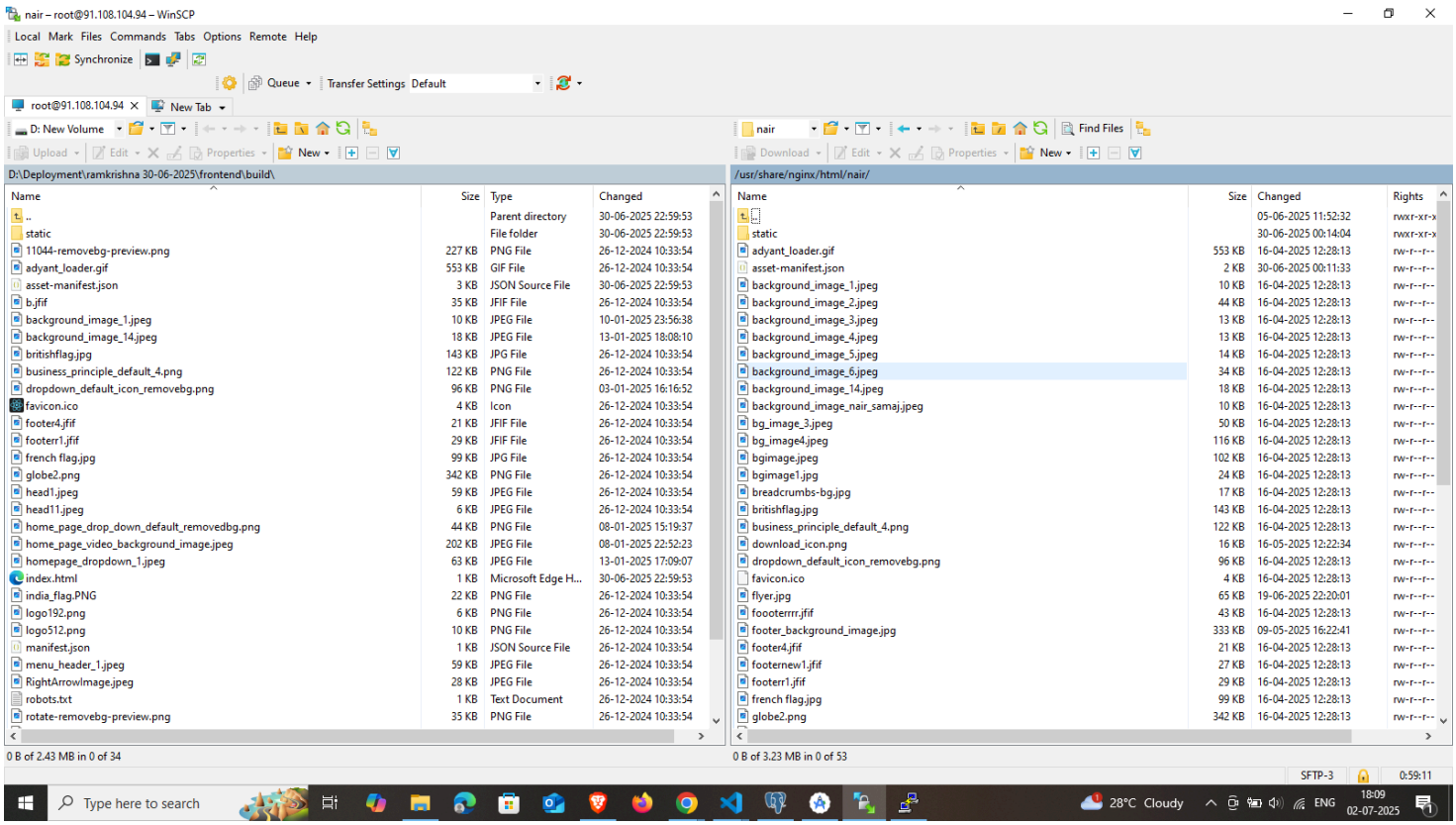
First open winscp and then putty
Give password then run commands
cd /root/
cd /var/www/html/project_directory_name/ (eg cd /var/www/html/nair)
Then grab the gunicorn process using ps -ef|grep guni

```
root     1350477       1  0 Jun29 ?        00:00:36 /var/www/html/nair/env/bin/python3 /var/www/html/nair/env/bin/gunicorn --workers 3 --bind 0.0.0.0:8013 backend.wsgi --daemon
root     1350478 1350477  0 Jun29 ?        00:00:28 /var/www/html/nair/env/bin/python3 /var/www/html/nair/env/bin/gunicorn --workers 3 --bind 0.0.0.0:8013 backend.wsgi --daemon
root     1350479 1350477  0 Jun29 ?        00:00:27 /var/www/html/nair/env/bin/python3 /var/www/html/nair/env/bin/gunicorn --workers 3 --bind 0.0.0.0:8013 backend.wsgi --daemon
root     1350480 1350477  0 Jun29 ?        00:00:27 /var/www/html/nair/env/bin/python3 /var/www/html/nair/env/bin/gunicorn --workers 3 --bind 0.0.0.0:8013 backend.wsgi --daemon
```

Kill the process by using command kill -9 process_id for the specific port number (eg Check above SS & give command kill -9 1350477)
After one minute, check again if any more process is running or not by using ps -ef|grep guni if any more process are running then kill the process using command kill -9 process_id

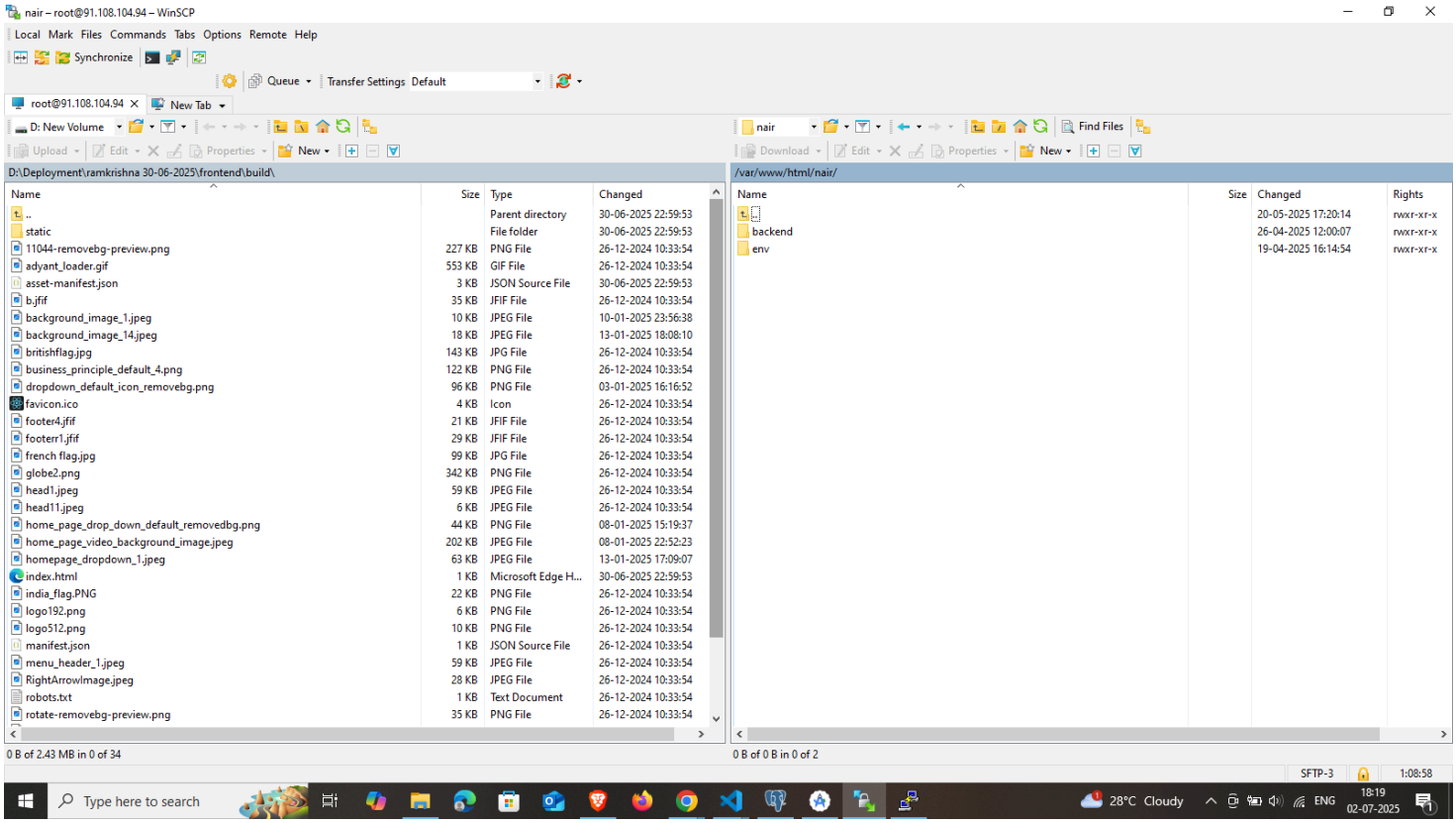
When no process are running for that particular port number then merge the backend and frontend code

For frontend



- Please check the above SS in Right side column (/usr/share/nginx/html/nair/)
- Delete all the build from RHS and upload the latest build from LHS

For backend



Check RHS and navigate to backend directory

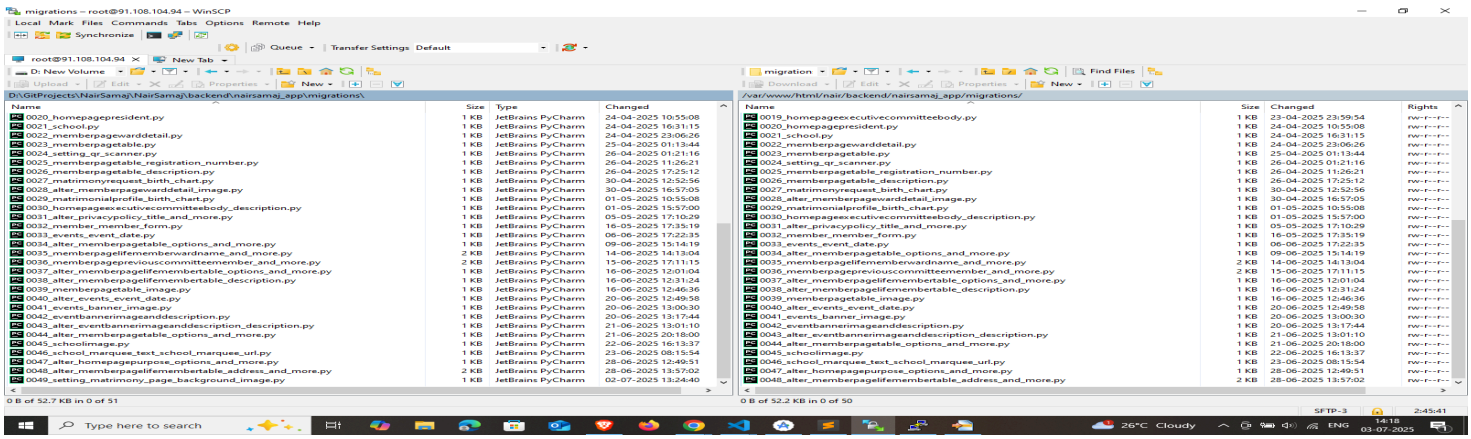
Delete only the following specific files in backend

- models.py
- serializers.py
- urls.py
- views.py
- admin.py

Do NOT delete the following

- requirements.txt
- settings.py
- migration files

And merge the new migration files (if new migration files are present)



➔ Open putty and give password

- cd /root/ (navigate to root directory)
- cd /var/www/html/ (navigate to backend folder)
- cd project_directory/ (navigate to specific project eg cd nair/)
- source env/bin/activate (activate virtual environment)
- cd backend/ (navigate to backend folder)
- python manage.py migrate (if migration files are present)
- gunicorn --workers 3 --bind 0.0.0.0:port_number backend.wsgi --daemon (eg gunicorn --workers 3 --bind 0.0.0.0:8013 backend.wsgi --daemon To start server in backend)
- We can runserver using python manage.py runserver 0.0.0.0:port_number (eg python manage.py runserver 0.0.0.0:8013 This command is used to start the Django development server, typically for debugging or to verify that the backend is functioning correctly, Press Ctrl + C to stop the server)
- service nginx restart (To start nginx server)
- Now check both frontend & backend in browser