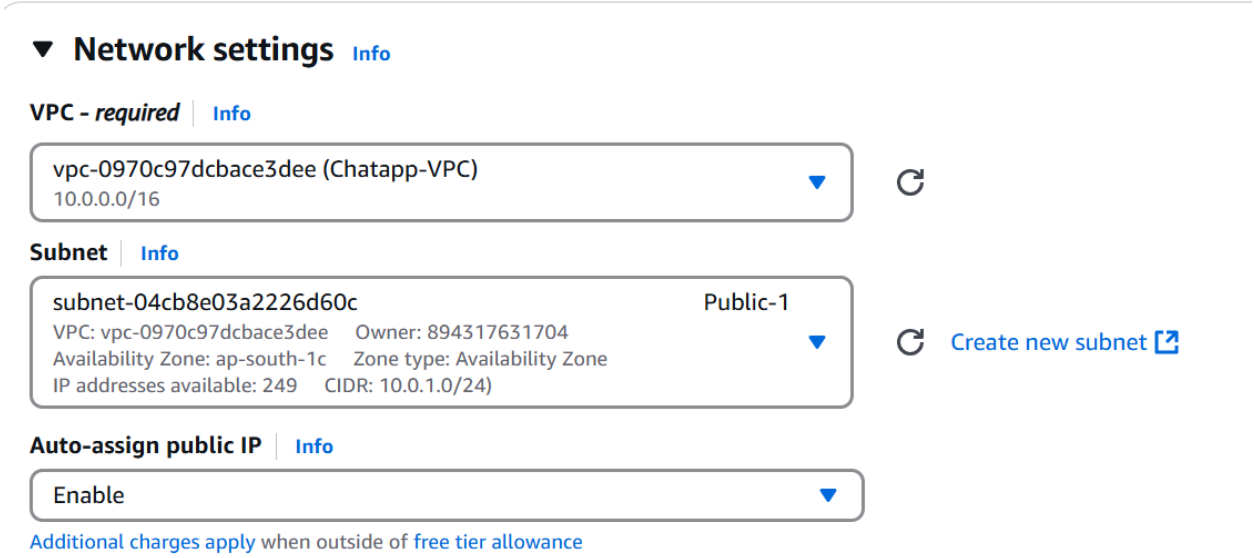
Activity-2

Deploy a 3-tier Python-Django based Chat Application.
Tier 1 - Web Server (Nginx)
Tier 2 - Application Server/Backend (Python Django)
Tier 3 - Database (MySQL)A few pointers:
1. The application needs a database (MySQL)
2. Application is written in the Python Django framework. You need to look into how to set up the Python Django Environment to run/deploy these applications.
3. For a web server, you need to set up a Reverse Proxy Web Server - Nginx.
One more important point:
1. For the backend, make sure you are using Ubuntu 22.04 version of AMI.
2. The Application supports python 3.8 version. Make sure you install the correct one.

First we will create a VPC (named ChatApp-VPC) with the CIDR range 10.0.0.0/16
Then we will create 4 subnets(2 public and 2 private). Eg. 10.0.1.0/24, 10.0.2.0/24, 10.0.3.0/24, 10.0.4.0/24
We will create an Internet Gateway and attach it to our VPC
We will create 2 route tables(Public-RT & Private-RT) and add routes to them
For Public-RT, Add route for 0.0.0.0/0 and give the source as internet gateway and select the new internet gateway id. Add the subnets to this route table by clicking on edit subnet associations
Now we will create a NAT Gateway for which we need to allocate an Elastic IP then we will go on the NAT Gateway tab and create a NAT Gateway and associate it to the EIP
Now we will go to the Private Route Table and add the route of 0.0.0.0/0 and give the source as NAT Gateway and associate the 2 private subnet to it

With this our VPC is Created, Now we will launch 3 instances each for each layer
We will select a common key-pair for all of the instances
We will edit the network setting so that we can change the VPC to ChatApp-VPC , set the subnet to public subnet -1 & enable out auto assign IP



For the rest of the 2 instances we will give the same key-pair and the VPC but the subnet will be the private ones and disable the auto-assign public IP

In Nginx Instance:

Now we will upload our key-pair to the nginx server
Now we will ssh into the public instance using Git Bash

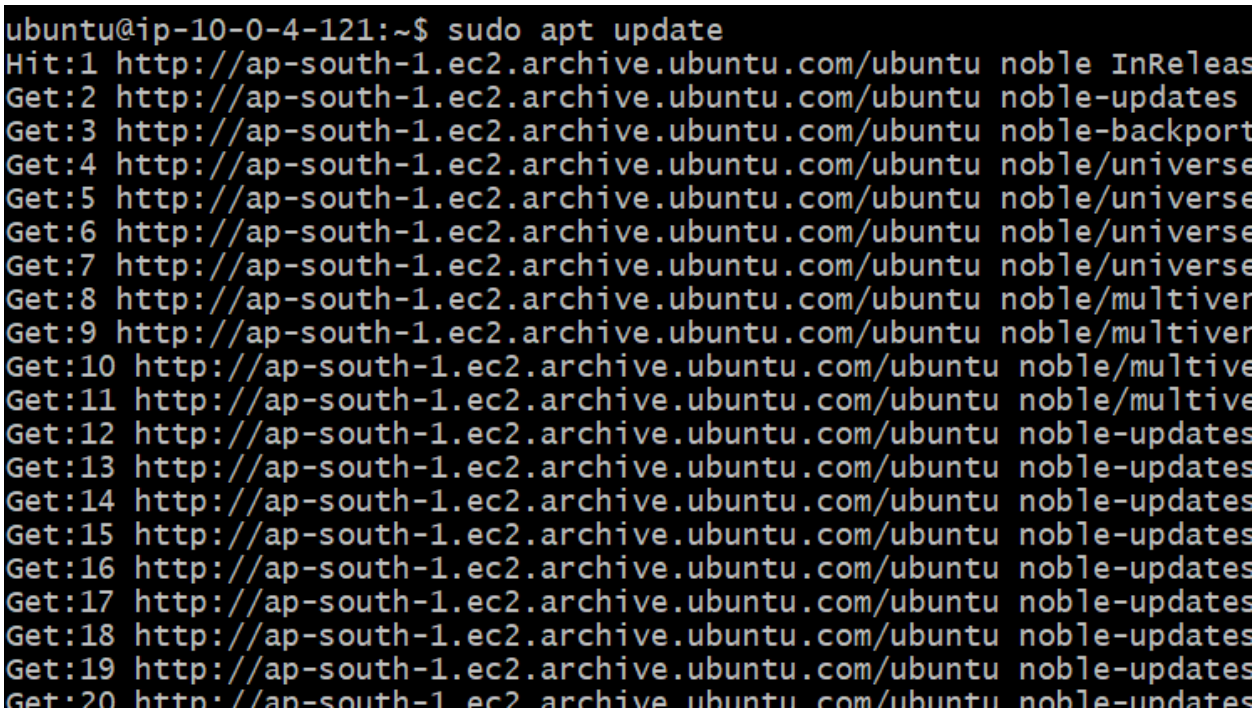
```
pgalh@Parth MINGW64 ~/Downloads
$ scp -i ChatApp.pem ChatApp.pem ubuntu@13.200.207.167:~/
ChatApp.pem

pgalh@Parth MINGW64 ~/Downloads
$ ssh -i ChatApp.pem ubuntu@13.200.207.167
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1024-aws x86_64)
```

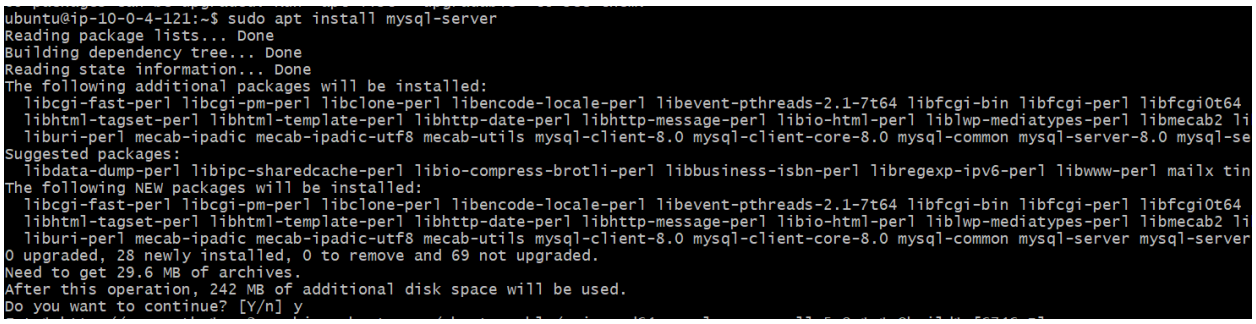
This is done so that we can ssh into the mysql instance by using the ssh command
ssh -i your-pem-key.pem ubuntu@<Your private IP>

In MySQL Instance(By SSH):

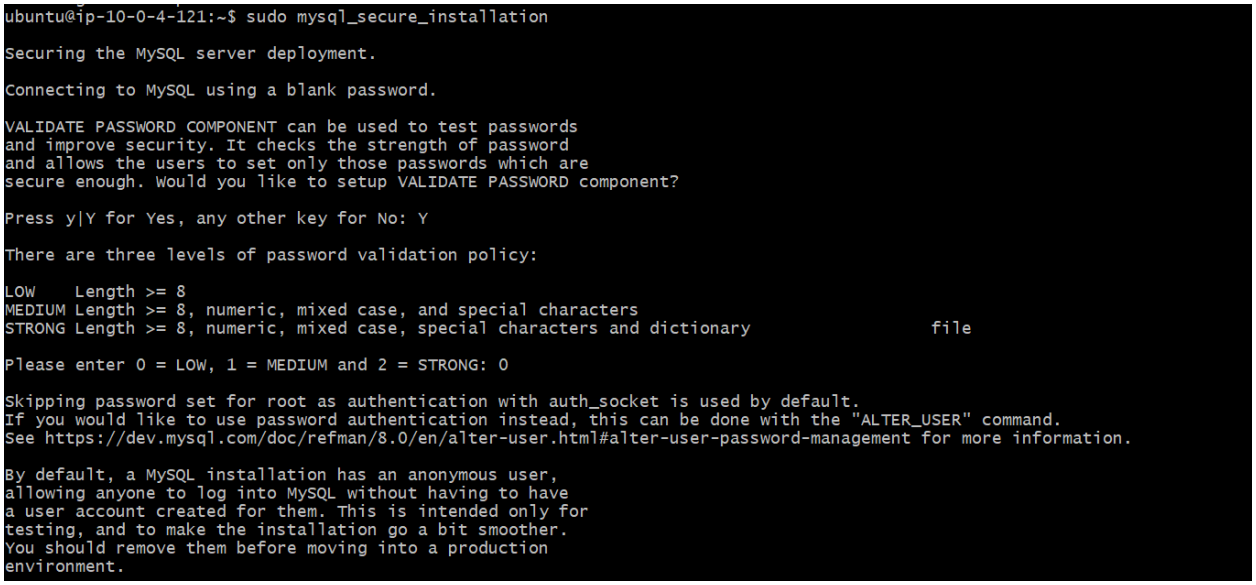
Now we will update all the packages in the database layer



Then we will install the mysql server package



As we are using a database in a secure mode will install a mysql secure installation package which will require authentication to use mysql



You'll be prompted for several steps:

VALIDATE PASSWORD PLUGIN: Choose whether to enforce strong passwords (optional).

Set root password: Yes → set a secure root password.

Remove anonymous users: Yes.

Disallow root remote login: Yes.

Remove test database: Yes.
Reload privilege tables: Yes.
Now we will change the data directory of mysql from /var/lib/mysql to /data/mysql
To do this, first we have to stop the mysql service then create a folder /data/mysql and make its owner as mysql.

Then we have to copy existing data to a new location using rsync command

```
ubuntu@ip-10-0-4-121:~$ sudo systemctl stop mysql
ubuntu@ip-10-0-4-121:~$ sudo mkdir -p /data/mysql
ubuntu@ip-10-0-4-121:~$ sudo chown -R mysql:mysql /data/mysql
ubuntu@ip-10-0-4-121:~$ sudo rsync -av /var/lib/mysql/ /data/mysql
sending incremental file list
./
#ib_16384_0.dblwr
#ib_16384_1.dblwr
auto.cnf
binlog.000001
binlog.000002
binlog.000003
binlog.index
ca-key.pem
ca.pem
client-cert.pem
client-key.pem
debian-5.7.flag
ib_buffer_pool
ibdata1
mysql.ibd
private_key.pem
public_key.pem
server-cert.pem
server-key.pem
undo_001
undo_002
#innodb_redo/
```

Now we have to edit the mysql config file to change the data directory to /data/mysql

```
ubuntu@ip-10-0-4-121:~$ sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

```
user            = mysql
# pid-file      = /var/run/mysqld/mysqld.pid
# socket        = /var/run/mysqld/mysqld.sock
# port          = 3306
datadir = /data/mysql
```

Now we can start the mysql service but there will be an error as the apparmor which restricts the mysql's access is not edited

```
ubuntu@ip-10-0-4-121:~$ sudo systemctl start mysql
Job for mysql.service failed because the control process exited with error code.
See "systemctl status mysql.service" and "journalctl -xeu mysql.service" for details.
```

We will change wherever /var/lib/mysql is written to /data/mysql

```
ubuntu@ip-10-0-4-121:~$ sudo nano /etc/apparmor.d/usr.sbin.mysqld
```

```
# Allow data dir access
  /data/mysql/ r,
  /data/mysql/** rwk,
```

Then we will restart the app armor by giving this command and verify the changes

```
ubuntu@ip-10-0-4-121:~$ sudo apparmor_parser -r /etc/apparmor.d/usr.sbin.mysqld
ubuntu@ip-10-0-4-121:~$ sudo systemctl start mysql
ubuntu@ip-10-0-4-121:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
     Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
     Active: active (running) since Tue 2025-04-29 08:59:53 UTC; 28s ago
    Process: 2924 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
   Main PID: 2933 (mysqld)
     Status: "Server is operational"
      Tasks: 38 (limit: 1077)
     Memory: 373.2M (peak: 386.9M)
        CPU: 1.090s
     CGroup: /system.slice/mysql.service
             └─2933 /usr/sbin/mysqld

Apr 29 08:59:51 ip-10-0-4-121 systemd[1]: Starting mysql.service - MySQL Community Server...
Apr 29 08:59:53 ip-10-0-4-121 systemd[1]: Started mysql.service - MySQL Community Server.
```

Now we can create the database and grant it all the privileges

```
mysql> create database chatapp_db;
Query OK, 1 row affected (0.01 sec)

mysql> create user 'chatuser'@'%' identified by 'strongpassword123';
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
mysql> show variables like 'validate_password%';
+--------------------------------------------------+--------+
| Variable_name                                    | Value  |
+--------------------------------------------------+--------+
| validate_password.changed_characters_percentage  | 0      |
| validate_password.check_user_name                | ON     |
| validate_password.dictionary_file                |        |
| validate_password.length                         | 8      |
| validate_password.mixed_case_count               | 1      |
| validate_password.number_count                   | 1      |
| validate_password.policy                         | MEDIUM |
| validate_password.special_char_count             | 1      |
+--------------------------------------------------+--------+
8 rows in set (0.01 sec)
```

```
mysql> set global validate_password.policy = LOW;
Query OK, 0 rows affected (0.00 sec)

mysql> set global validate_password.length = 6;
Query OK, 0 rows affected (0.00 sec)

mysql> create user 'chatuser'@'%' identified by 'strongpassword123';
Query OK, 0 rows affected (0.02 sec)

mysql> grant all privileges on chatapp_db.* to 'chatuser'@'%';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> exit;
Bye
```

Here we got an error as the validate policy was set as medium and we have now set it as low

In Django Instance(By SSH):

Now we have to clone the repo from git to django so we have to ssh into django from nginx and update all the packages in it first

```
ubuntu@ip-10-0-1-11:~$ ssh -i ChatApp.pem ubuntu@10.0.3.151
The authenticity of host '10.0.3.151 (10.0.3.151)' can't be established.
ED25519 key fingerprint is SHA256:M2DmTYoVirCMagNxmvwQN6O3XYk/e43ZrA88iMuUGI4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.3.151' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1024-aws x86_64)

 * Documentation:  https://help.ubuntu.com
```

Now we install git in the django instance and clone the repo

```
ubuntu@ip-10-0-3-151:~$ sudo apt install git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.2).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 69 not upgraded.
ubuntu@ip-10-0-3-151:~$ git clone https://github.com/ARPIT226/chat_app.git
Cloning into 'chat_app'...
remote: Enumerating objects: 242, done.
remote: Counting objects: 100% (242/242), done.
remote: Compressing objects: 100% (185/185), done.
remote: Total 242 (delta 47), reused 242 (delta 47), pack-reused 0 (from 0)
Receiving objects: 100% (242/242), 1.49 MiB | 16.75 MiB/s, done.
Resolving deltas: 100% (47/47), done.
ubuntu@ip-10-0-3-151:~$
```

Now cd into the repo

```
ubuntu@ip-10-0-3-151:~$ cd chat_app
ubuntu@ip-10-0-3-151:~/chat_app$
```

Now to install django we have to first install python

To install Python we need to create a virtual environment

```
ubuntu@ip-10-0-3-151:~/chat_app$ sudo apt install software-properties-common -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-software-properties
The following packages will be upgraded:
  python3-software-properties software-properties-common
2 upgraded, 0 newly installed, 0 to remove and 67 not upgraded.
Need to get 44.3 kB of archives.
After this operation, 1024 B of additional disk space will be used.
```

Here we have a ppa deadsnakes which help in installing the latest version of python in our system

```
ubuntu@ip-10-0-3-151:~/chat_app$ sudo add-apt-repository ppa:deadsnakes/ppa -y
Repository: 'Types: deb
URIs: https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu/
Suites: noble
Components: main
'
Description:
This PPA contains more recent Python versions packaged for Ubuntu.

Disclaimer: there's no guarantee of timely updates in case of security problems or other issues. If you want t
y, on a production server), you do so at your own risk.

Update Note
===========
Please use this repository instead of ppa:fkrull/deadsnakes.

Reporting Issues
```

```
ubuntu@ip-10-0-3-151:~/chat_app$ sudo apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu noble InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
67 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-10-0-3-151:~/chat_app$ sudo apt install python3.8 python3.8-venv python3.8-dev -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

```
ubuntu@ip-10-0-3-151:~/chat_app$ sudo apt install python3.8 python3.8-venv python3.8-dev -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core fonts-dejavu-mono libaom3 libc-dev-bin libc-devtools libc6-dev libcrypt-dev libde
  libfontconfig1 libgd3 libheif-plugin-aomdec libheif-plugin-aomenc libheif-plugin-libde265 libheif1 libjbig0 libjpeg-t
  libpython3.8-dev libpython3.8-minimal libpython3.8-stdlib libsharpyuv0 libtiff6 libwebp7 libxpm4 linux-libc-dev linux
  python3.8-lib2to3 python3.8-minimal rpcsvc-proto
Suggested packages:
  glibc-doc libgd-tools libheif-plugin-x265 libheif-plugin-ffmpegdec libheif-plugin-jpegdec libheif-plugin-jpegenc libh
  libheif-plugin-rav1e libheif-plugin-svtenc binutils binfmt-support
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core fonts-dejavu-mono libaom3 libc-dev-bin libc-devtools libc6-dev libcrypt-dev libde
  libgd3 libheif-plugin-aomdec libheif-plugin-aomenc libheif-plugin-libde265 libheif1 libjbig0 libjpeg-turbo8 libjpeg8
  libpython3.8-minimal libpython3.8-stdlib libsharpyuv0 libtiff6 libwebp7 libxpm4 linux-libc-dev manpages-dev python3.8
  python3.8-lib2to3 python3.8-minimal python3.8-venv rpcsvc-proto
The following packages will be upgraded:
  libexpat1 linux-tools-common
2 upgraded, 39 newly installed, 0 to remove and 65 not upgraded.
Need to get 26.2 MB of archives.
After this operation, 88.3 MB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libexpat1 amd64 2.6.1-2ubuntu0.3 [88.0 k
```

Now we have installed python in our system, We will create a virtual environment

```
ubuntu@ip-10-0-3-151:~/chat_app$ python3.8 -m venv ven
ubuntu@ip-10-0-3-151:~/chat_app$ source ven/bin/activate
```

Now we will install all our requirements written in the requirement.txt file

```
(ven) ubuntu@ip-10-0-3-151:~/chat_app$ pip install -r requirements.txt
```

We will install the required build dependencies

```
(ven) ubuntu@ip-10-0-3-151:~/chat_app$ sudo apt install default-libmysqlclient-dev build-essential -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu bzip2 cpp cpp-13 cpp-13-x86-64-linux-gnu cpp-x86-64-linux-gnu dpkg-dev fakeroot g+
  g++-x86-64-linux-gnu gcc gcc-13 gcc-13-base gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu libalgorithm-diff-perl libalgorithm-diff-xs-
  libasan8 libatomic1 libbinutils libcc1-0 libctf-nobfd0 libctf0 libdpkg-perl libfakeroot libfile-fcntllock-perl libgcc-13-dev libgompl
  libitm1 liblsan0 libmpc3 libmysqlclient-dev libmysqlclient21 libquadmath0 libsframe1 libssl-dev libstdc++-13-dev libtsan2 libubsan1 l
  mysql-common zlib1g-dev
Suggested packages:
  binutils-doc gprofng-gui bzip2-doc cpp-doc gcc-13-locales cpp-13-doc debian-keyring g++-multilib g++-13-multilib gcc-13-doc gcc-multi
  bison gdb gcc-doc gcc-13-multilib gdb-x86-64-linux-gnu bzr libssl-doc libstdc++-13-doc make-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential bzip2 cpp cpp-13 cpp-13-x86-64-linux-gnu cpp-x86-64-linux-gnu defa
  fakeroot g++ g++-13 g++-13-x86-64-linux-gnu g++-x86-64-linux-gnu gcc gcc-13 gcc-13-base gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan8 libatomic1 libbinutils libcc1-0 libctf-nobfd0 libctf0 libdpkg-perl libfake
  libgcc-13-dev libgomp1 libgprofng0 libhwasan0 libisl23 libitm1 liblsan0 libmpc3 libmysqlclient-dev libmysqlclient21 libquadmath0 libs
  libtsan2 libubsan1 libzstd-dev lto-disabled-list make mysql-common zlib1g-dev
0 upgraded, 54 newly installed, 0 to remove and 65 not upgraded.
Need to get 73.4 MB of archives.
After this operation, 260 MB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 binutils-common amd64 2.42-4ubuntu2.5 [240 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libsframe1 amd64 2.42-4ubuntu2.5 [15.5 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libbinutils amd64 2.42-4ubuntu2.5 [577 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libctf-nobfd0 amd64 2.42-4ubuntu2.5 [97.7 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libctf0 amd64 2.42-4ubuntu2.5 [94.5 kB]
```

We will now update the setting.py to connect to the database and give it the private ip of the database layer instance

```
(ven) ubuntu@ip-10-0-3-151:~/chat_app$ cd fundoo/fundoo
(ven) ubuntu@ip-10-0-3-151:~/chat_app/fundoo/fundoo$ nano settings.py
```

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': os.getenv('DB_NAME', 'chatapp_db'),
        'USER': os.getenv('DB_USER', 'chatuser'),
        'PASSWORD': os.getenv('DB_PASSWORD', 'strongpassword123'),
        'HOST': os.getenv('DB_HOST', '10.0.4.121'),
        'PORT': os.getenv('DB_PORT', '3306'),
    }
}
```

Now we migrate the database and collect static files

To migrate the database we need to change the security group of our database instance so that it can listen to the IP of the django instance

Then we have to install mysqlclient and check if it is active

```
Last login: Tue Apr 29 10:23:54 2025 from 10.0.1.11
ubuntu@ip-10-0-4-121:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
     Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
     Active: active (running) since Tue 2025-04-29 08:59:53 UTC; 1h 29min ago
    Process: 2924 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS
   Main PID: 2933 (mysqld)
     Status: "Server is operational"
      Tasks: 38 (limit: 1077)
     Memory: 370.5M (peak: 386.9M)
        CPU: 35.271s
     CGroup: /system.slice/mysql.service
             └─2933 /usr/sbin/mysqld
```

We can see if mysql is listening to localhost

```
^[[Aubuntu@ip-10-0-4-121:~$ sudo netstat -tulnp | grep 3306
tcp        0      0 127.0.0.1:33060         0.0.0.0:*               LISTEN      2933/mysqld
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      2933/mysqld
ubuntu@ip-10-0-4-121:~$ sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
ubuntu@ip-10-0-4-121:~$ sudo systemctl restart mysql
ubuntu@ip-10-0-4-121:~$ sudo netstat -tulnp | grep 3306
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN      3542/mysqld
tcp        0      0 127.0.0.1:33060         0.0.0.0:*               LISTEN      3542/mysqld
ubuntu@ip-10-0-4-121:~$ nc -zv 10.0.4.121 3306
```

Here we have edited the mysql config file to listen to host

We have also used nc command to ensure that mysql is listening to its IP

Now we have activated the virtual environment and migrated the manage.py

file

```
(ven) ubuntu@ip-10-0-3-151:~/chat_app/fundoo$ python manage.py makemigrations
/home/ubuntu/chat_app/ven/lib/python3.8/site-packages/django/core/management/commands/makemigrations.py:160: RuntimeWarning: Got an error ch
story performed for database connection 'default': (2003, "Can't connect to MySQL server on '10.0.4.121:3306' (110)")
  warnings.warn(
No changes detected
(ven) ubuntu@ip-10-0-3-151:~/chat_app/fundoo$ python manage.py makemigrations
No changes detected
(ven) ubuntu@ip-10-0-3-151:~/chat_app/fundoo$
(ven) ubuntu@ip-10-0-3-151:~/chat_app/fundoo$ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, fundooapp, sessions, sites
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
```

And we have used python manage.py collectstatic command to overwrite the static file

Now we run the server on port 8000 on local host

```
(ven) ubuntu@ip-10-0-3-151:~/chat_app/fundoo$ python manage.py runserver 0.0.0.0:8000
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
April 29, 2025 - 10:45:14
Django version 4.2.8, using settings 'fundoo.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.

^C(ven) ubuntu@ip-10-0-3-151:~/chat_app/fundoo$ deactivate
```

If it works then we have to install Gunicorn in the venv

```
(ven) ubuntu@ip-10-0-3-151:~/chat_app$ pip install gunicorn
Requirement already satisfied: gunicorn in ./ven/lib/python3.8/site-packages (20.1.0)
Requirement already satisfied: setuptools>=3.0 in ./ven/lib/python3.8/site-packages (from gunicorn) (75.3.2)

[notice] A new release of pip is available: 23.0.1 -> 25.0.1
[notice] To update, run: pip install --upgrade pip
```

Now we bind the gunicorn service to run the application for us.

```
(ven) ubuntu@ip-10-0-3-151:~/chat_app$ gunicorn --bind 0.0.0.0:8000 chat_app.wsgi
[2025-04-29 10:48:13 +0000] [5327] [INFO] Starting gunicorn 20.1.0
[2025-04-29 10:48:13 +0000] [5327] [INFO] Listening at: http://0.0.0.0:8000 (5327)
[2025-04-29 10:48:13 +0000] [5327] [INFO] Using worker: sync
[2025-04-29 10:48:14 +0000] [5329] [INFO] Booting worker with pid: 5329
[2025-04-29 10:48:14 +0000] [5329] [ERROR] Exception in worker process
```

But this is done manually so we need to automate it.

This is done by editing the systemd file to create a daemon process which will automatically run our application through gunicorn

```
ubuntu@ip-10-0-3-151:~$ sudo nano /etc/systemd/system/gunicorn.service
```

```
ubuntu@ip-10-0-3-151: ~
  GNU nano 7.2                                                    /etc/systemd/system/gunicorn.servic
[Unit]
Description=Gunicorn service for Django ChatApp
After=network.target

[Service]
User=ubuntu
Group=www-data
WorkingDirectory=/home/ubuntu/chat_app/fundoo
Environment="PATH=/home/ubuntu/chat_app/ven/bin"
ExecStart=/home/ubuntu/chat_app/ven/bin/gunicorn --workers 3 --bind 0.0.0.0:8000 fundoo.wsgi:application

[Install]
WantedBy=multi-user.target
```

This will autostart our application on boot

We will have to reload, enable and start gunicorn to save the changes

```
(ven) ubuntu@ip-10-0-3-151:~/chat_app/fundoo$ sudo systemctl daemon-reload
(ven) ubuntu@ip-10-0-3-151:~/chat_app/fundoo$ sudo systemctl restart gunicorn
(ven) ubuntu@ip-10-0-3-151:~/chat_app/fundoo$ sudo systemctl status gunicorn
● gunicorn.service - Gunicorn service for Django ChatApp
     Loaded: loaded (/etc/systemd/system/gunicorn.service; enabled; preset: enabled)
     Active: active (running) since Tue 2025-04-29 11:05:53 UTC; 12s ago
   Main PID: 5967 (gunicorn)
      Tasks: 4 (limit: 1077)
     Memory: 93.7M (peak: 93.9M)
        CPU: 975ms
     CGroup: /system.slice/gunicorn.service
             ├─5967 /home/ubuntu/chat_app/ven/bin/python3.8 /home/ubuntu/chat_app/ven/bin/gunicorn --workers 3 --bind 0.0.0.0:8000 fundoo.wsgi:a
             ├─5969 /home/ubuntu/chat_app/ven/bin/python3.8 /home/ubuntu/chat_app/ven/bin/gunicorn --workers 3 --bind 0.0.0.0:8000 fundoo.wsgi:a
             ├─5970 /home/ubuntu/chat_app/ven/bin/python3.8 /home/ubuntu/chat_app/ven/bin/gunicorn --workers 3 --bind 0.0.0.0:8000 fundoo.wsgi:a
             └─5971 /home/ubuntu/chat_app/ven/bin/python3.8 /home/ubuntu/chat_app/ven/bin/gunicorn --workers 3 --bind 0.0.0.0:8000 fundoo.wsgi:a
```

We can check if the application is running by using curl

```
(ven) ubuntu@ip-10-0-3-151:~/chat_app/fundoo$ curl http://localhost:8000

<link rel="stylesheet" type="text/css" href="/static/css/style.css">
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Raleway">
<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Index</title>


<center>
    <h1> welcome to Chat application Bridge For Sonata users 4 </h1><br><br>
    <button class="button" style="vertical-align:middle"><span><a href="/sign_in/">Login</a></span></button><br>
    <h3><a href="/signup/">Click Here to register!</a></h3><br><br>
    </center>
</center>>
```

Now we need to include the frontend to our application.

In Nginx Instance(By SSH):

This is done using Nginx

Now on the Nginx Instance, we need to first update all the packages

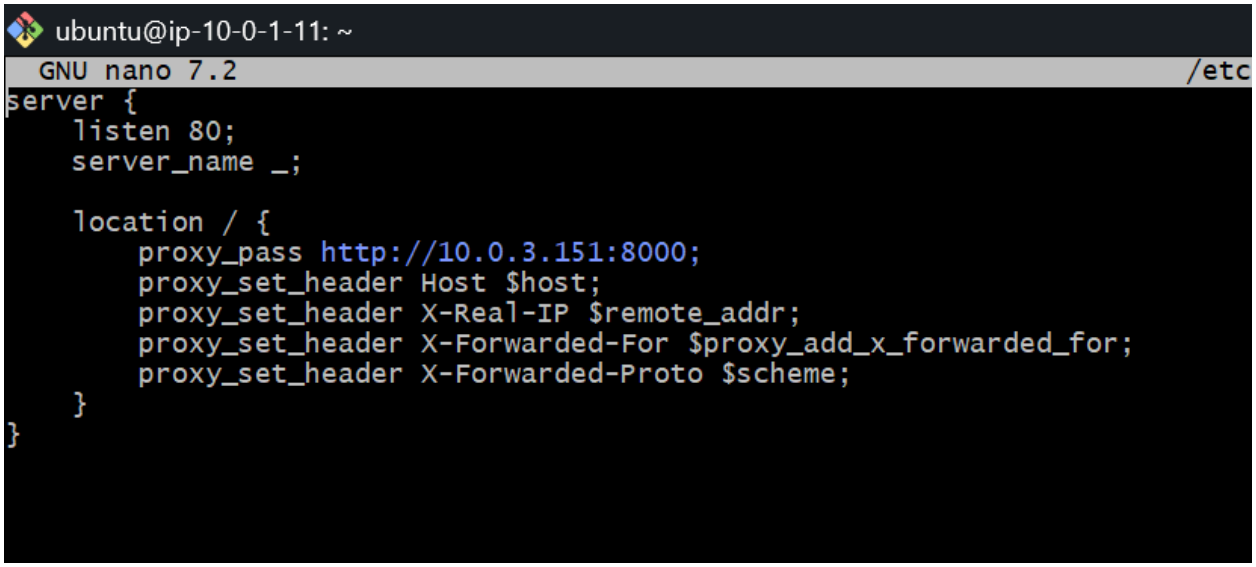
```
ubuntu@ip-10-0-1-11:~$ sudo apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 k
B]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126
 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages
[15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en
[5982 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Component
s [3871 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Met
adata [301 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Package
s [269 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-
en [118 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Compon
ents [35.0 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f
Metadata [8328 B]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Pack
ages [1028 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translatio
n-en [224 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Comp
```

Then we need to install nginx in the instance

```
ubuntu@ip-10-0-1-11:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  nginx-common
Suggested packages:
  fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  nginx nginx-common
0 upgraded, 2 newly installed, 0 to remove and 80 not upgraded.
Need to get 551 kB of archives.
After this operation, 1596 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx
-common all 1.24.0-2ubuntu7.3 [31.2 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx
 amd64 1.24.0-2ubuntu7.3 [520 kB]
Fetched 551 kB in 0s (22.7 MB/s)
```

To Set the reverse proxy on nginx we need to edit the /etc/nginx/sites-available file and set the configurations of the application

```
ubuntu@ip-10-0-1-11:~$ sudo nano /etc/nginx/sites-available/chatapp
```



We also need to keep in mind that there is a default site enabled which we need to disable

Now we need to enable the sites configured

```
ubuntu@ip-10-0-1-11:~$ sudo ln -s /etc/nginx/sites-available/chatapp /etc/nginx/site
s-enabled/
ubuntu@ip-10-0-1-11:~$ sudo nginx -t
```

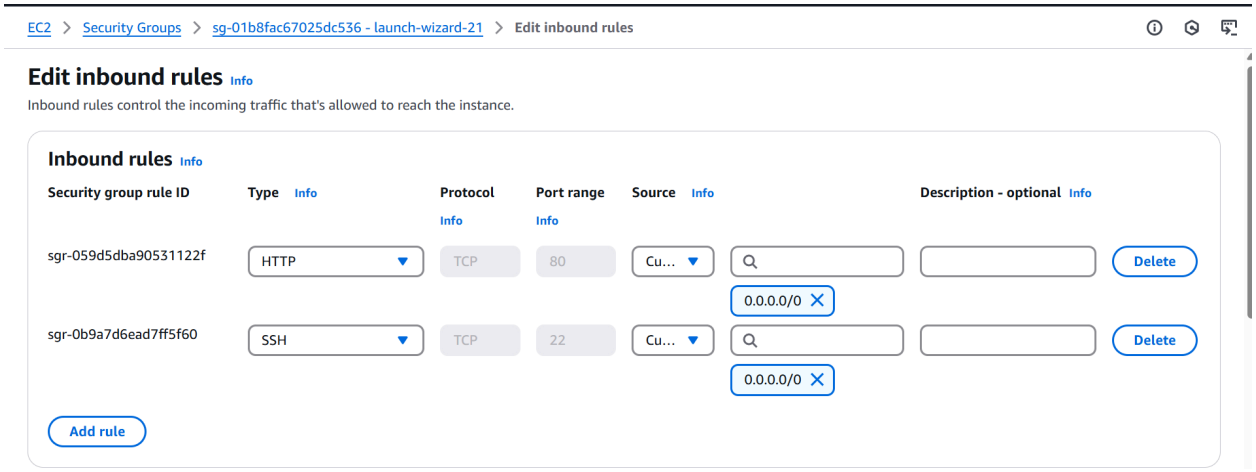
We need to check if the syntax and configurations in the nginx server is ok or not

After this we need to reload or restart the server

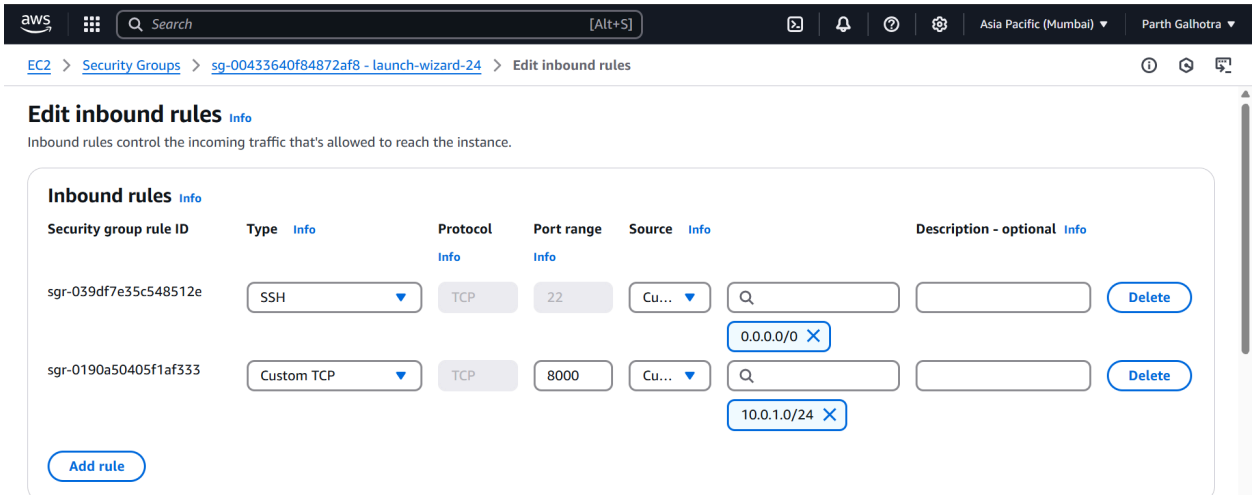
```
ubuntu@ip-10-0-1-11:~$ sudo nginx -t
2025/04/30 05:12:41 [warn] 1981#1981: conflicting server name "_" on 0.0.0.0:80, ign
ored
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
ubuntu@ip-10-0-1-11:~$ sudo systemctl restart nginx
```

We also have to set the security group of the nginx instance and the django instance

For nginx



For Django



Now we can verify the app is working or not by using curl

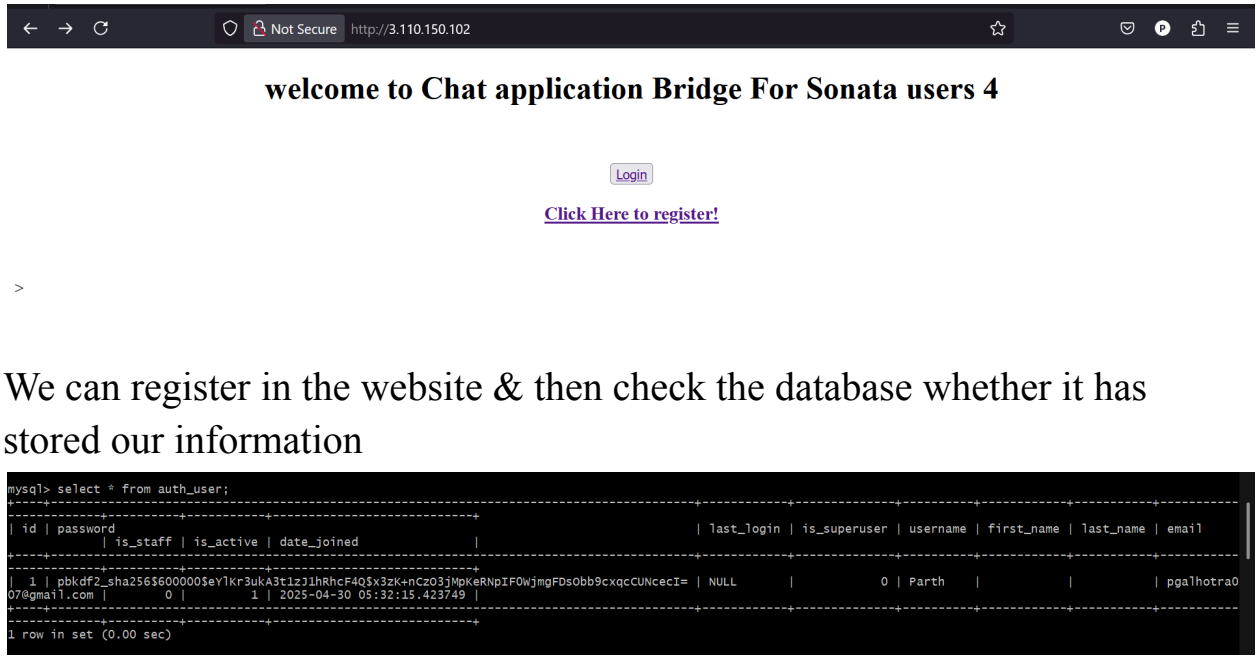
```
ubuntu@ip-10-0-3-151:~$ curl http://localhost:8000

<link rel="stylesheet" type="text/css" href="/static/css/style.css">
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Raleway">
<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Index</title>


<center>
    <h1> welcome to Chat application Bridge For Sonata users 4 </h1><br><br>
    <button class="button" style="vertical-align:middle"><span><a href="/sign_in/">Login</a></span></button><br>
    <h3><a href="/signup/">Click Here to register!</a></h3><br><br>
    </center>
</center>>
```

We can also verify it by going to the browser and typing the nginx's public IP



We can register in the website & then check the database whether it has stored our information

```
mysql> select * from auth_user;
+----+------------------------------------------------------------------------------------------------+------------+--------------+----------+------------+-----------+-----------
------------+----------+----------+----------------------------+
| id | password                                                                                       | last_login | is_superuser | username | first_name | last_name | email
           | is_staff | is_active | date_joined                |
+----+------------------------------------------------------------------------------------------------+------------+--------------+----------+------------+-----------+-----------
------------+----------+----------+----------------------------+
|  1 | pbkdf2_sha256$600000$eYlKr3ukA3tIzJlhRhcF4Q$x3zK+nCzO3jMpKeRNpIFOWjmgFDsObb9cxqcCUNcecI= | NULL       |            0 | Parth    |            |           | pgalhotra0
07@gmail.com |        0 |         1 | 2025-04-30 05:32:15.423749 |
+----+------------------------------------------------------------------------------------------------+------------+--------------+----------+------------+-----------+-----------
------------+----------+----------+----------------------------+
1 row in set (0.00 sec)
```