Asynchronous Web Chat Project

Guidelines for development server

September 30, 2022

1 Development environment

This section details how to setup up the VM for the development environment. The same step will have to be followed for staging, testing and production environment.

1. Using VirtualBox, install a VM for the development server with the following requirements :
   - 4GB of RAM
   - 20GB of storage
   - a recent version of Ubunti (20.04 or 22.04)
2. Type in the terminal :
   ```
   $ : sudo apt update
   $ : sudo apt upgrade
   ```
3. **If not already installed,** install python 3.8 :
   ```
   $ : sudo apt install software-properties-common
   $ : sudo add-apt-repository ppa:deadsnakes/ppa
   $ : sudo apt install python3.8 python3.8-dev python3-pip python3.8-venv
   ```
4. Install dependencies :
   ```
   $ : sudo apt install libpq-dev postgresql postgresql-contrib nginx curl unzip libgl1
   ```
5. Get the GitHub repo :
   ```
   $ : git clone https://github.com/ayman-makki/WebChatApp.git
   ```

6. Set up a virtual environment with Python 3.8 in the same folder and install requirements for the project :

```
$ : cd WebChatApp
$ : python3.8 - venv env
$ : source env/bin/activate
$ (env) : pip install -r requirements.txt
```

2 Development server

1. **Set up the database :**

```
$ (env) : sudo -u postgres psql
    CREATE DATABASE djangodb
    CREATE USER django WITH PASSWORD ’password’;
    ALTER ROLE django SET client_encoding TO ’utf8’;
    ALTER ROLE django SET default_transaction_isolation TO ’read committed’;
    ALTER ROLE django SET timezone TO ’UTC’;
    GRANT ALL PRIVILEGES ON DATABASE djangodb TO django;
    \q
```

2. Change database settings in ”WebChatApp/ChatServerPlayground/settings.py” with the settings you just set in the previous instruction.

3. Update database :

```
$ (env) : python manage.py makemigrations
$ (env) : python manage.py migrate
```

4. Install redis :

```
$ : sudo apt install redis-server
```

5. Add ”*” in ALLOWED_HOSTS dans ”WebChatApp/ChatServerPlayground/settings.py”.

6. Create a super user in the django application :

```
$ (env) : python manage.py createsuperuser
```

7. Launch the WSGI server by using the following command :

   ```
   $ (env) : python manage.py runserver
   ```

8. Create a public chat room in the database by connecting to 127.0.0.1:8000/admin and using the super user account you just created.

9. Try to log in into the website, create multiple accounts and send messages between users.