Installation Guide Using Script

1. Clone the “deployment “ branch from the GitHub repo:

   git clone -b deployment https://github.com/Aishwary13/smp-portal.git

```
iiitd@smpvm:~/smp$ git clone -b deployment https://github.com/Aishwary13/smp-portal.git
Cloning into 'smp-portal'...
remote: Enumerating objects: 12944, done.
remote: Counting objects: 100% (2651/2651), done.
remote: Compressing objects: 100% (1162/1162), done.
remote: Total 12944 (delta 1485), reused 2170 (delta 1437), pack-reused 10293
Receiving objects: 100% (12944/12944), 19.62 MiB | 9.90 MiB/s, done.
Resolving deltas: 100% (4491/4491), done.
iiitd@smpvm:~/smp$
```

2. Extract the files and run the following commands:
   a. cd smp-portal
   b. chmod +x setup.sh
   c. ./setup.sh
3. Here's an overview of the steps included in the setup script:
   1. Install Python and pip: This step updates the package list and installs Python 3.11.5 and pip.
   2. Install Nginx: Nginx is installed to serve as the web server.
   3. Install PostgreSQL: PostgreSQL is installed as the database management system.
      a. Create PostgreSQL database and user: This involves accessing the PostgreSQL console and executing commands to create a database and user.
   4. Allow port 8000 through firewall: This step opens port 8000 to allow incoming connections.
   5. BackEnd:
      a. Install virtualenv: Virtualenv is installed to create isolated Python environments.
      b. Create and activate virtual environment: A virtual environment named "sdos" is created and activated.
      c. Install Python dependencies: Required Python packages are installed using pip.
      d. Move to the backend directory and start Django models: The script moves to the backend directory, where Django models are managed.
      e. Migration commands are executed to apply database changes.

6. FrontEnd:
    a. Install Node.js: Node.js and npm are installed for frontend development.

Manual Installation Guide

Please follow each of the following steps exactly as mentioned.

1. Clone the “deployment “ branch from the GitHub repo:

    git clone -b deployment https://github.com/Aishwary13/smp-portal.git

BackEnd:

1. Install Python and pip:
    - Update the package list: `sudo apt-get update`
    - Install Python 3.11.5 and pip: `sudo apt-get install -y python3=3.11.5 python3-pip`

2. Install Nginx:
    - Install Nginx: `sudo apt install python3-pip python3-dev nginx`

3. Install PostgreSQL:
    - Install PostgreSQL: `sudo apt-get install -y postgresql`

4. Create PostgreSQL database and user:
    - Access PostgreSQL console: `sudo -u postgres psql`
    - Execute the following commands in the PostgreSQL console:
        ```
        CREATE DATABASE smp;
        CREATE USER admin WITH PASSWORD '12345';
        ALTER ROLE admin SET client_encoding TO 'utf8';
        ALTER ROLE admin SET timezone TO 'UTC';
        GRANT ALL PRIVILEGES ON DATABASE smp TO admin;
        ```

5. Allow port 8000 through firewall:
    - Allow port 8000: `sudo ufw allow 8000`

6. Install virtualenv:
    - Install virtualenv: `pip3 install virtualenv`

7. Create and activate virtual environment:
    - Create virtual environment: `python3 -m venv sdos`
    - Activate virtual environment: `source sdos/bin/activate`

8. Install Python dependencies:
    - Install required Python packages:

pip3 install pytz typing-extensions sqlparse psycopg2 psutil asgiref Django==4.2.5 djangorestframework==3.14.0 django-cors-headers==4.3.0

9. Move to the backend directory and start Django models:
    - Move to backend directory: `cd backend`
    - Run migrations:
        ```
        python3 manage.py makemigrations
        python3 manage.py migrate
        ```

FrontEnd:

1. Install Node.js:
    - Install Node.js and npm:
        sudo apt install nodejs npm
        npm install

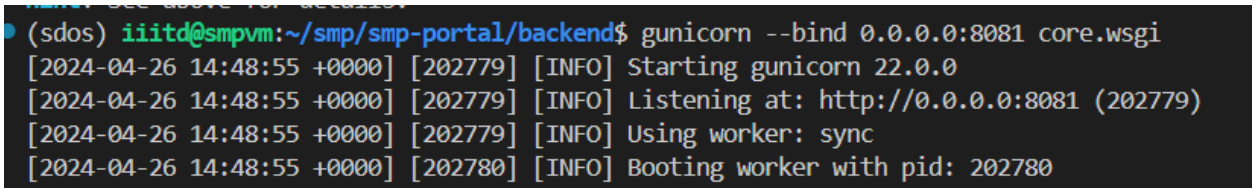
2. move to the directory ./smp and run the command “npm start”

Hosting the Application

BackEnd:

1. After running the setup script first run the following command to start the venv
    source sdos/bin/activate

2. After that check that gunicorn is successfully installed:
    gunicorn --bind 0.0.0.0:8000 core.wsgi



3. Deactivate the virtual environment with :

   deactivate

```
(sdos) iiitd@smpvm:~/smp/smp-portal/backend$ deactivate
```

4. Next create a system socket for gunicorn run:

   sudo vim /etc/systemd/system/gunicorn.socket

5. And then paste the following content into the file:

```
[Unit]
Description=gunicorn socket

[Socket]
ListenStream=/run/gunicorn.sock

[Install]
WantedBy=sockets.target
```

   press escape and type :wq to save the file

6. Next, we will create a service file for gunicorn:

   sudo vim /etc/systemd/system/gunicorn.service

7. Paste the contents below inside this file:

```
[Unit]
Description=Gunicorn instance to serve core
Requires=gunicorn.socket
After=network.target

[Service]
User=iiitd
Group=iiitd
WorkingDirectory=/home/iiitd/smp-portal/backend
ExecStart=/home/iiitd/smp-portal/sdos/bin/gunicorn  \
      --access-logfile - \
      --workers 3 \
      --bind unix:/run/gunicorn.sock \
      core.wsgi:application
Restart=always
RestartSec=3
[Install]
WantedBy=multi-user.target
```

   press escape and type :wq to save the file

8. Lets now start and enable the gunicorn socket

sudo systemctl start gunicorn.socket
sudo systemctl enable gunicorn.socket

9. Check that the gunicorn is running:

   sudo systemctl status gunicorn

Nginx:

10. Configure Nginx: Create a new configuration file for the React app in Nginx's sites-available directory:

    sudo nano /etc/nginx/sites-available/smpportal.iiitd.edu.in

11. Inside this file, configure Nginx to serve the React app:

    Create a configuration file for Nginx using the following command:
    sudo vim /etc/nginx/sites-available/smpportal.iiitd.edu.in

12. Paste the below contents inside the file created

```
server {
  listen 80;
  server_name smpportal.iiitd.edu.in;

  location /api/ {
    include proxy_params;
    proxy_pass http://unix:/run/gunicorn.sock;
  }
  location / {
    root /home/iiitd/smp-portal/build;
    index index.html;
    try_files $uri $uri/ /index.html;
  }
}


server {
    listen 443 ssl;
  listen [::]:443 ssl;
```

```
    server_name smpportal.iiitd.edu.in;

    ssl_certificate /etc/letsencrypt/live/smpportal.iiitd.edu.in/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/smpportal.iiitd.edu.in/privkey.pem;

    location /api/ {
        include proxy_params;
        proxy_pass http://unix:/run/gunicorn.sock;
    }
    location / {
        root /home/iiitd/smp-portal/build;
        index index.html;
        try_files $uri $uri/ /index.html;
    }
}
```

13. Enable Nginx Configuration: Create a symbolic link to enable Nginx configuration file in the sites-enabled directory:

    sudo ln -s /etc/nginx/sites-available/smpportal.iiitd.edu.in /etc/nginx/sites-enabled/

14. Restart Nginx: restart Nginx to apply the changes:

    sudo systemctl daemon-reload

    sudo service gunicorn restart

    sudo service nginx restart

    sudo systemctl restart nginx

To check logs in gunicorn :

journalctl -u gunicorn

To check logs in nginx:

```
sudo tail -f /var/log/nginx/access.log
```

```
sudo tail -f /var/log/nginx/error.log
```

Check Gunicorn Service Status:

sudo systemctl status gunicorn

Check Gunicorn is accepting requests:

curl --unix-socket /run/gunicorn.sock localhost

Check Nginx Service Status:

sudo systemctl status nginx

Changes for the VM/Deployment

1. Build the react application using: `npm run build`
2. and Merge the changes in the branch named “deployment”, {there will be some conflicts as running for testing requires the endpoints to be local host where as for deployment purposes we require the remote IP: change them to respective urls}.

Go to the VM machine and pull all the changes and run the following commands:

FrontEnd:

If there are any changes in the frontend, Run the following commands:

1) npm run build

3) sudo systemctl restart nginx

BackEnd:

If there are any changes in the backend, Run the following commands:

1) sudo service gunicorn restart

2) sudo service nginx restart