Sportzy

This project is purely built on django with help of nodejs and using PostgreSQL as the database using redis for caching. Plus it uses razorpay as its payment gateway.

Tools required for this project

- Python 3.10.9
- Redis
- Nodejs 19.3.0 with npm
- PostgreSQL 15

Steps to start this project

First extract the project to the location where you want to keep it, then at that location open the terminal window in that directory and run the following commands.

```
pip install -r requirements.txt
```

- After that setup your Postgres Database
- Rename the `sample.conf` file to to `.conf`
- Set the `REDIS_URL` if you have set the redis to be run on custom server.
- Set the `DATABASE_URL` parameter according to the following syntax `postgres://USER:PASSWORD@HOST:PORT/DATABASE_NAME`
- Go to Razorpay create your account there and see here how to create your api keys click here `Note create the API KEYS in **Live Mode** only.`
- Now set the the `RAZOR_KEY_ID` and `RAZOR_KEY_SECRET` accordingly in `.conf` file.
- Now login to your Google Business Account and generate an custom app password. Click here to see how app passwords are generated `Note: While`

`selecting the app, select the **CUSTOM** option and device should be **WEBSITE**.`

- In the `.conf` file set the `EMAIL_HOST_USER` as the email address of your Google Business Account and set the `EMAIL_HOST_PASSWORD` to the newly generated app password from google.
- When hosting in the production server, also set the following things in `.conf`

```
[Security]
PRODUCTION_SERVER = 1
DEBUG = 0
SECRET_KEY = any_random_key

[Logging]
LOGGING = 1

[DATABASE]
REDIS_URL=redis://HOST:PORT
DATABASE_URL = postgres://USER:PASSWORD@HOST:PORT/DATABASE_NAME
POSTGRES = 1
SQLITE = 0

[STATICFILES]
WHITENOISE=1
COMPRESS_ENABLED=1
COMPRESS_OFFLINE=1
```

- Now run the following commands in the same terminal which you have opened it earlier.

```
python manage.py migrate
python manage.py css_init
python manage.py collectcompress
python manage.py tailwind build
```

- To create a superuser type the following command in the terminal.

```
python manage.py createsuperuser
```

- Also set a cron-job to run the following command in the project directory.

```
python manage.py clearsessions
```

- If on windows machine to run the project, type the following commands in seperate terminals for each command in the same directory.

```
uvicorn ecommerce.asgi:application
```

For more advance usage on Uvicorn visit its docs

```
celery -A ecommerce worker  -Q celery -l info -E --without-gossip --without-mingle --without-heartbeat
```

If a linux or unix type machine, then type the following commands in seperate terminals for each command in the same directory.

```
gunicorn ecommerce.asgi:application -k ecommerce.workers.DynamicUvicornWorker --timeout 500
```

For more advance usage on gunicorn visit its docs

```
celery -A ecommerce worker -B -Q celery -l info -E --without-gossip --without-mingle --without-heartbeat
```