

# #1 Installations

>>make sure you are in virtual environment

```
pip install dj-database-url
```

```
pip install gunicorn
```

```
pip install psycopg2
```

```
pip install psycopg2-binary
```

```
pip install whitenoise
```

# Procfile

>> Create Procfile without any extension

>> At Project Level directory

>> Add below code in proc file

>> Save (CTRL+S)

```
web: gunicorn project_name.wsgi --log-file -
```

>> If your project name is myshop

>> Then use myshop.wsgi

>> Instead of project\_name.wsgi

# runtime.txt

- >> Create runtime.txt
- >> At Project Level directory
- >> Write python version in it
- >> SAVE

>> Example:-

python-3.9.2

# requirements.txt

- >> Create requirements.txt
- >> At Project Level directory
- >> Start cmd at Project Level directory
- >> Activate Virtual Environment
- >> Write pip freeze > requirements.txt
- >> To create requirements.txt file

>> File look like below:-

```
asgiref==3.4.1  
dj-database-url==0.5.0  
Django==3.2.8  
gunicorn==20.1.0  
psycopg2==2.9.1  
pytz==2021.3  
sqlparse==0.4.2  
whitenoise==5.3.0
```

## static (folder)

- >> Create static (folder)
- >> At Project Level directory
- >> In static create style.css file
- >> SAVE

# settings.py (project folder)

>> Edit settings.py

>> In Project directory

>> SAVE

>> Changes are below:-

>> CHANGE #1

```
import dj_database_url
db_from_env=dj_database_url.config(conn_max_age=500)
DATABASES['default'].update(db_from_env)
```

>> CHANGE #2

```
Import os
```

```
STATIC_URL = '/static/'
STATIC_ROOT = os.path.join(BASE_DIR, "staticfiles")
```

```
STATICFILES_DIRS = [
    os.path.join(BASE_DIR, "static"),
]
```

```
STATICFILES_STORAGE =
    'whitenoise.storage.
      CompressedManifestStaticFilesStorage'
```

## settings.py (project folder)

### >> CHANGE #3

```
MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',  
    'whitenoise.middleware.WhiteNoiseMiddleware',  
]
```

### >> CHANGE #4

```
DEBUG = False
```

```
ALLOWED_HOSTS = [ '127.0.0.1' , 'other all domains' ]
```

# Heroku Git CLI

## >> Heroku Login Using CLI

>> heroku login

## >> Create Heroku App

>> heroku create heroku\_appname

## >> Reset Heroku App Repo

>> heroku repo:reset - -app heroku\_appname

## >> Add Heroku Remote

>> heroku git:remote - a heroku\_appname

## >> Deploy On Heroku

>> git push heroku master

## >> Heroku Help

>> heroku help

## >> Ensure Atleast one instance is running

>> heroku ps:scale web=1

## >> Collect Static Files Locally

>> python manage.py collectstatic

## >> Run Locally in Windows

>> heroku local web -f Procfile.windows

## >> Run Locally using Gunicorn

>> heroku local web



# Heroku Git CLI

## >> Managing Addons

>> heroku addons

## >> Adding Addons

>> heroku addons:open heroku-postgresql

## >> Managing Configurations

>> heroku config

## >> Config SECRET\_KEY

>> heroku config:set DJANGO\_SECRET\_KEY = 'key code'

## >> Config DEBUG

>> heroku config:set DJANGO\_DEBUG = False

## >> Heroku Logs

>> heroku logs

## >> Heroku Show Current Logs

>> heroku logs --tail

## >> Additional logging for CollectStatic

>> heroku config:set DEBUG\_COLLECTSTATIC=1

## >> Heroku Display Dyno Status

>> heroku ps

# Heroku Git CLI Push Local Changes

>> install requests locally

>> pip install requests

>> Then Add it to requirements.txt

>> Import requests in views.py

>> Example:-

```
import requests
```

```
def home_page(request):
```

```
    r=requests.get('http://httpbin.org/status/418')
```

```
    return HttpResponse('<pre>' + r.txt + ' </pre>')
```

>> Create Addons

>> heroku addons:create papertrail

>> Heroku Open Addons

>> heroku addons:open papertail

>> Heroku Show Addons

>> heroku addons

>> Start Python shell

>> heroku run python manage.py shell

Don't forget to type exit to exit shell and terminate dyno

>> Database\_URL

>> heroku pg

# Heroku Git CLI Python

>> Heroku Migrate

>> heroku run python manage.py migrate

>> Connect Models to Postgresql

>> heroku pg:psql

Make sure postgresql installed locally

>> Models

```
def db(request):  
    greeting = Greeting() greeting.save()  
    greetings = Greeting.objects.all()  
    return render(request, 'db.html', {'greetings':greetings})
```