Deploy Django to Render

Below are the steps to deploy your Django project to Render, including creating the necessary .gitignore and bash script.

Update settings.py
ALLOWED_HOSTS = ["*"]

Install gunicorn and create requirements.txt - in terminal run:
>> pip install gunicorn
>> pip freeze > requirements.txt

create .gitignore file (remember period at beginning of file name)
==== .gitignore file =====
Django
*.log
*.pot
*.pyc
__pycache__
db.sqlite3
media

Backup files
*.bak

If you are using PyCharm
User-specific stuff
.idea/**/workspace.xml
.idea/**/tasks.xml
.idea/**/usage.statistics.xml
.idea/**/dictionaries
.idea/**/shelf

AWS User-specific
.idea/**/aws.xml

Generated files
.idea/**/contentModel.xml

Sensitive or high-churn files
.idea/**/dataSources/
.idea/**/dataSources.ids
.idea/**/dataSources.local.xml
.idea/**/sqlDataSources.xml

```
.idea/**/dynamic.xml
.idea/**/uiDesigner.xml
.idea/**/dbnavigator.xml

# Gradle
.idea/**/gradle.xml
.idea/**/libraries

# File-based project format
*.iws

# IntelliJ
out/

# JIRA plugin
atlassian-ide-plugin.xml

# Python #
*.py[cod]
*$py.class

# Distribution / packaging
.Python build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
*.whl
*.egg-info/
.installed.cfg
*.egg
*.manifest
*.spec

# Installer logs
pip-log.txt
pip-delete-this-directory.txt
```

```
# Unit test / coverage reports
htmlcov/
.tox/
.coverage
.coverage.*
.cache
.pytest_cache/
nosetests.xml
coverage.xml
*.cover
.hypothesis/

# Jupyter Notebook
.ipynb_checkpoints

# pyenv
.python-version

# celery
celerybeat-schedule.*

# SageMath parsed files
*.sage.py

# Environments
.env
.venv
env/
venv/
ENV/
env.bak/
venv.bak/

# mkdocs documentation
/site

# mypy
.mypy_cache/

# Sublime Text #
*.tmlanguage.cache
*.tmPreferences.cache
*.stTheme.cache
```

*.sublime-workspace
*.sublime-project

sftp configuration file
sftp-config.json

Package control specific files Package
Control.last-run
Control.ca-list
Control.ca-bundle
Control.system-ca-bundle
GitHub.sublime-settings

Visual Studio Code
.vscode/*
!.vscode/settings.json
!.vscode/tasks.json
!.vscode/launch.json
!.vscode/extensions.json
.history

======= END FILLE =====

Initialize Git and make our first commit (in terminal run)
>> git init
>> git add .
>> git commit -m “init commit”

Create an account and a new repo on GitHub (https://github.com/)

Push your local repo to github (follow instructions on github)

Create an account on Render (https://render.com/)

Create a new ‘Web Service’

Keep default settings accept:
Start Command set to “gunicorn config.wsgi”

Update for static files

Install gunicorn & update requirements.txt
>> pip install whitenoise
>> pip freeze > requirements.txt

Update settings.py file

Update middleware

```
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    "whitenoise.middleware.WhiteNoiseMiddleware", ← add this line here
…
]
```

Update static settings

```
STATIC_URL = 'static/'
STATIC_ROOT = 'staticfiles'
STATICFILES_STORAGE = "whitenoise.storage.CompressedManifestStaticFilesStorage"
```

Create build.sh file

```
==== FILE START =====
#!/usr/bin/env bash
# exit on error
set -o errexit

pip install -r requirements.txt
python manage.py collectstatic --no-input
python manage.py migrate

==== FILE END ====
```

Commit and push changes to github (in terminal run)

```
>> git add .
>> git commit -m “updated static settings”
>> git push origin master
```

Update build command in Render settings

Start Command set to “./build.sh”