

Topic: Car Listing: Deployment using Heroku Part 19C

Speaker: Udemy Instructor Rathan Kumar | Notebook: Django Project: Car Listing



1. After the corrections made in my SETTINGS.PY

```
2
3 from django.contrib.messages import constants as messages
4
5 import os
6 # we add this block for HEROKU deployment
7 import django_heroku
8 import dj_database_url
```

The ALLOWED_HOSTS is HEROKU'S provided domain name:

```
# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = False

# Customize this to add for your Heroku's default domain, and your purchased domain name
ALLOWED_HOSTS = ['lit-cliffs-98316-70180276948a.herokuapp.com', ]
```

Make sure that WHITENOISE MIDDLEWARE is above other middleware because 3rd party applications may require this. See the [documentation](#) here.

```
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    # for HEROKU deployment
    'whitenoise.middleware.WhiteNoiseMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
    # allauth account middleware for social login
    'allauth.account.middleware.AccountMiddleware',
]
```

This is the DATABASE_URL after you made the second app and deleted the first one.

```
# HEROKU CREDENTIALS AND USE OF ENVIRONMENT VARIABLES PROVIDED
DATABASES = {'default': dj_database_url.config(
    default='postgres://uacp2joia9miv5:pfdb946d08413449c74196a586d2757b3a0366f1cd746406af51e9f57efd69ada@c81j070d5ubs83.cluster-czrs8kj4isg7.us-east-1.rds.amazonaws.com:5432/d44bqh6timinvv')}
```

This is the STATIC FILE CONFIGURATIONS where STATIC_ROOT uses STATICFILES folder instead of the STATIC value that Rathans uses. This is what the HEROKU documentation requires.

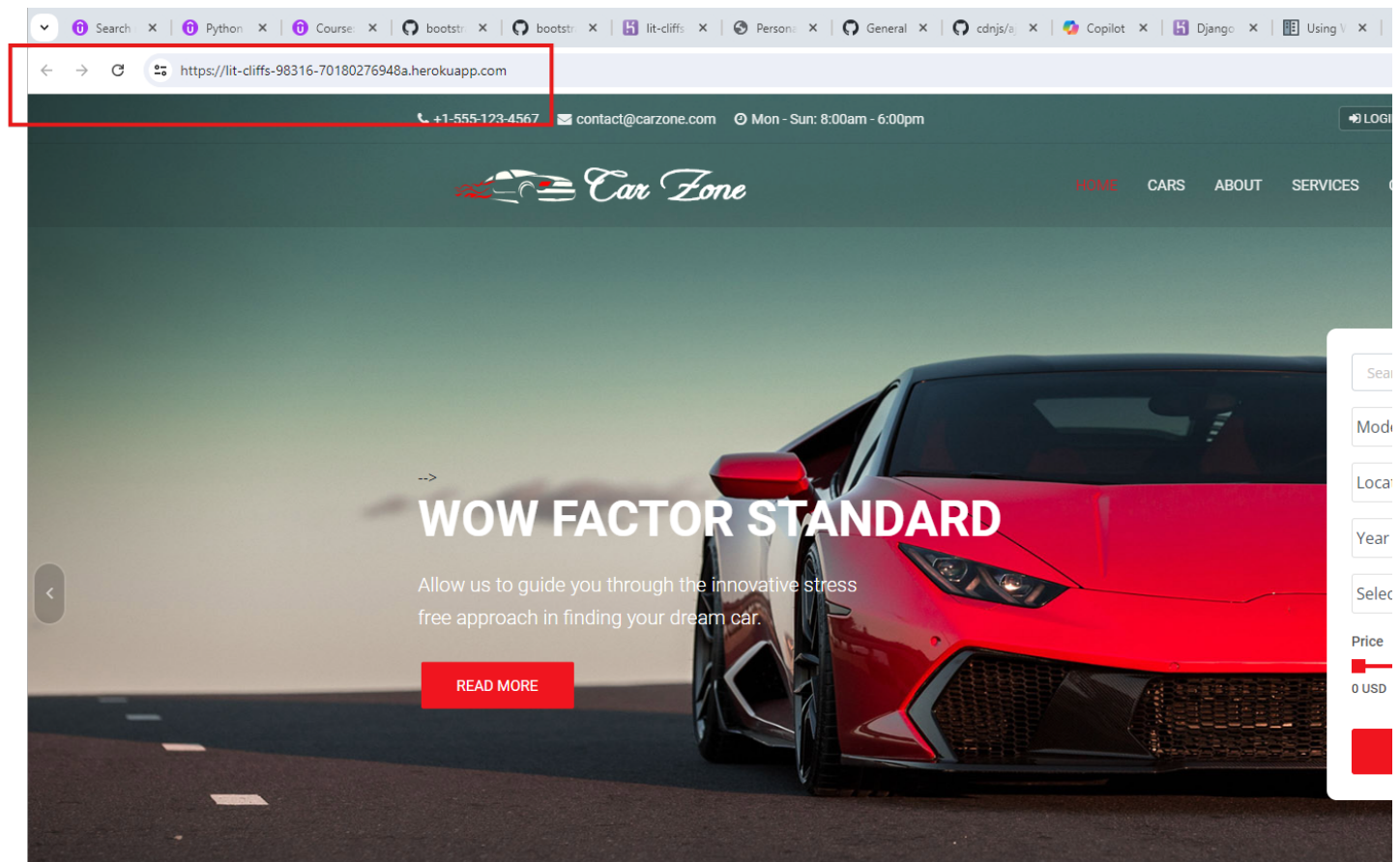
```
# STATIC FILE CONFIGURATON
STATIC_URL = '/static/'
# creates this new folder and copies all the static files TO this new folder after COLLECTSTATIC is issued
STATIC_ROOT = os.path.join(BASE_DIR, 'staticfiles')
# this is where our static files are copied FROM
STATICFILES_DIRS = [
    os.path.join(BASE_DIR, 'carzone/static'),
]
```

This is our STATIC STORAGE:

```
# Whitenoise Configuration for HEROKU deployment
STATICFILES_STORAGE = 'whitenoise.storage.CompressedManifestStaticFilesStorage'
WHITENOISE_SKIP_MISSING_FILES = True
WHITENOISE_MANIFEST_STRICT = False
```

2. Run `HEROKU OPEN` and the app should open. The address is the random name that Heroku gave our app. You can replace this if you purchase a DOMAIN like from NAMECHEAP or GoDADDY!

```
prune
(env)
Rosilie@DELL MINGW64 ~/OneDrive/Desktop/LEARNING DJANGO PROJECTS/Carzone-Project (main)
$ heroku open
(env)
Rosilie@DELL MINGW64 ~/OneDrive/Desktop/LEARNING DJANGO PROJECTS/Carzone-Project (main)
```



3. When you open the app, the images for the TEAM, FEATURED CARS wont show up. So you now need to load the JSON file to upload all our sample records that we did locally through our computer/laptop.

Featured Cars

Lorem ipsum dolor sit amet, consectetur adipiscing elit.



Latest Cars

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Executive Team

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod

4. We remove the /MEDIA file from .GITIGNORE.

5. We load our JSON file into HEROKU SERVER

```
python manage.py loaddata project_dump.json
```

6. We changed our DEBUG to TRUE

7. We pushed our changes to Heroku Server and we now see our media files.

8. But we encounter errors when we login. We can fix this via the ADMIN PANEL if you can access it otherwise, we use the Python shell to activate our admin site.

We issue the codes using Python shell:

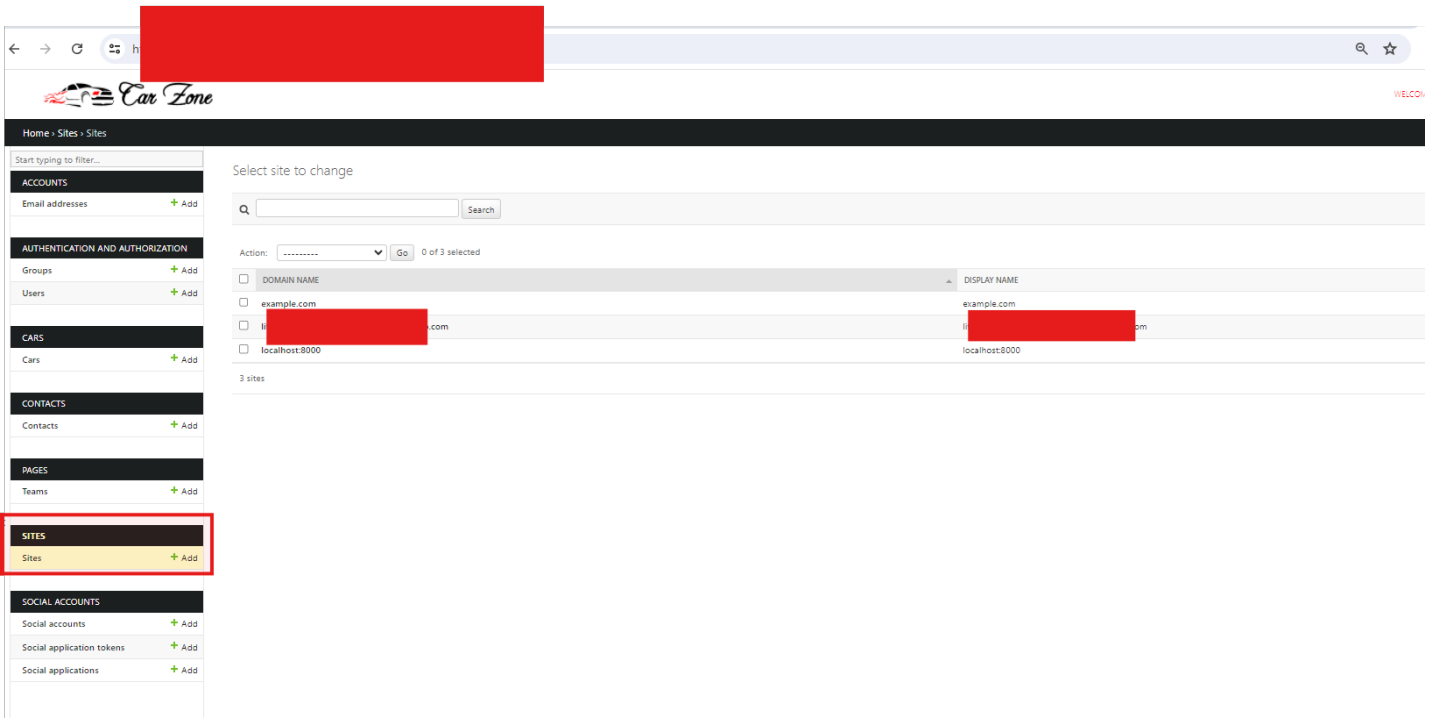
9. To get the SITE_ID of our heroku app:

```
>>> print(Site.objects.get(name='lit-cliffs-98316-70180276948a.herokuapp.com').id)
print(Site.objects.get(name='lit-cliffs-98316-70180276948a.herokuapp.com').id)
3
>>> Terminate batch job (Y/N)? Y
Y
```

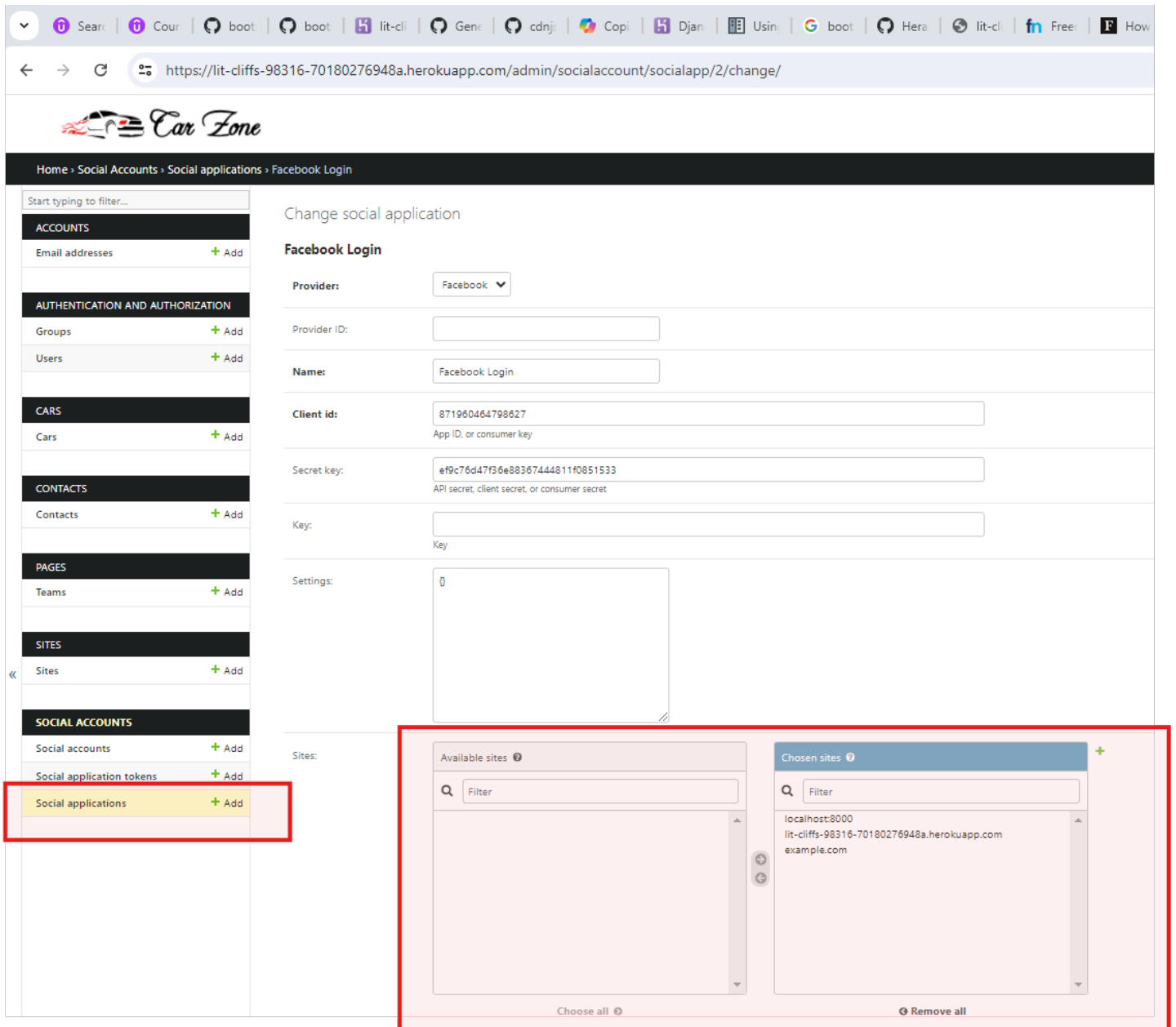
So we changed SITE_ID in SETTINGS.PY:

```
87
88 # SOCIAL MEDIA CONFIGURATION
89 # SITE_ID = 1 # localhost ID
90 SITE_ID = 3 # heroku's current app URL
91
```

10. Run your admin panel and you should be able to see your registered app domain name.



11. We get the error because of Facebook and Google configurations. So go back to the admin panel and make sure we add our newly registered domain.



DoesNotExist at /accounts/login

No exception message supplied

```
Request Method: GET
Request URL: https://lit-cliffs-98316-70180276948a.herokuapp.com/accounts/login
Django Version: 4.2.14
Exception Type: DoesNotExist
Exception Location: /app/.heroku/python/lib/python3.9/site-packages/allauth/socialaccount/adapter.py, line 304, in get_app
Raised during: accounts.views.login
Python Executable: /app/.heroku/python/bin/python
Python Version: 3.9.0
Python Path: ['/app/.heroku/python/bin',
              '/app',
              '/app/.heroku/python/lib/python3.9.zip',
              '/app/.heroku/python/lib/python3.9',
              '/app/.heroku/python/lib/python3.9/lib-dynload',
              '/app/.heroku/python/lib/python3.9/site-packages']
Server time: Fri, 02 Aug 2024 20:47:42 +0000
```

Error during template rendering

1 template /app/templates/accounts/login.html, error at line 64

Could not get exception message)

```
54         <i class="flaticon-lock"></i>
55     </div>
56     <div class="form-group mb-0 clearfix">
57         <button type="submit" class="btn-md btn-theme float-left">Login</button>
58     </div>
59     <div class="extra-login clearfix">
60         <span>Or Login With</span>
61     </div>
62     <div class="clearfix"></div>
63     <ul class="social-list">
64         <li><a href="{% provider_login_url 'facebook' method='oauth2' %}" class="facebook-bg":
65             class="fa fa-facebook facebook-i"></i><span>Facebook</span></a></li>
66         <li><a href="{% provider_login_url 'google' %}" class="google-bg"><i
67             class="fa fa-google google-i"></i><span>Google</span></a></li>
68     </ul>
69
70     </div>
71     <p>Don't have an account? <a href="{% url 'register' %}" class="thembo"> Register here</a></p>
72 </div>
73 </div>
74 </div>
```

Traceback [Switch to copy-and-paste view](#)

app/.heroku/python/lib/python3.9/site-packages/django/core/handlers/exception.py, line 55, in inner

12. You can now log in and add like messages. The admin can view the details of this messages via the admin dashboard.

13. Next we need to fix the Social Media Login since we have a new domain name BUT YOU NEED YOUR SSL so get this from heroku dashboard but this requires configuration. For practice, just copy the website's complete address that has SSL

Go to DEVELOPERS.FACEBOOK.COM and update your new domain name.

FROM:

TO:

14. We add the **Valid OAuth Redirect URIs** t

too, <https://lit-cliffs-98316-70180276948a.herokuapp.com/socialaccounts/facebook/login/callback/> Take note of the URL you have when you login. This is the URL you set in your URLS.PY. So on FACEBOOK DEVELOPERS dashboard, we add Click on

Use cases > Customize

Customize
Customize use case settings and permissions to make your app work the way you want it to.

Added to app

Facebook Login

- Permissions
- Settings**
- Quickstart

Manage use cases

Redirect URI Validator

Redirect URI to Check

Client OAuth settings

Client OAuth login
Enables the standard OAuth client token flow. Secure your application and prevent abuse by locking down which token redirect URIs are allowed with the options below. Disable globally if not used. [?]

Web OAuth login
Enables web-based Client OAuth Login. [?]

Force Web OAuth reauthentication
When on, prompts people to enter their Facebook password in order to log in on the web. [?]

Use Strict Mode for redirect URIs
Only allow redirects that exactly match the Valid OAuth Redirect URIs. Strongly recommended. [?]

Valid OAuth Redirect URIs
A manually specified redirect_uri used with Login on the web must exactly match one of the URIs listed here. This list is also used by the JavaScript SDK for in-app browsers that suppress popups. [?]

Login from Devices
Enables the OAuth client login flow for devices like a smart TV [?]

Allowed Domains for the JavaScript SDK
Login and signed-in functionality of the JavaScript SDK will only be available on these domains. [?]

Allowed Domains for the JavaScript SDK

You should be able to proceed now with your Facebook login:

You previously logged in to Django Carzone Listing with Facebook.
Would you like to continue?

By continuing, Django Carzone Listing will receive ongoing access to the information you share and Facebook will record when Django Carzone Listing accesses it. [Learn more](#) about this sharing and the settings you have.
[Django Carzone Listing's Privacy Policy](#)

15. We do the same for GOOGLE LOGIN. Go to [Google Developers' Console](#) and update the URLs.

16. Now load and test your page.