* Unable to login into Django admin ->

That’s because no super user was created in the data base. Using shell prompt on the project on the server super user was created using following commands.

MADHUR@MADHUR-PC MINGW64 ~/desktop/Cashback\_Uploaded\_22may (master)

$ heroku run python manage.py shell

Running python manage.py shell on stark-reef-61252... starting, run.2546 (Free)

Running python manage.py shell on stark-reef-61252... connecting, run.2546 (Free )

Running python manage.py shell on stark-reef-61252... up, run.2546 (Free)

Python 3.6.0 (default, Dec 24 2016, 04:15:54)

[GCC 5.4.0 20160609] on linux

Type "help", "copyright", "credits" or "license" for more information.

(InteractiveConsole)

>>> from django.contrib.auth.models import User

from django.contrib.auth.models import User

>>> user = User.objects.create\_user('madhur',password='madhur123')

user = User.objects.create\_user('madhur,password='madhur123')'')

File "<console>", line 1

user = User.objects.create\_user('madhur,password='madhur123')'')

^

SyntaxError: invalid syntax

>>> user = User.objects.create\_user('madhur',password='madhur123')

user = User.objects.create\_user('madhur',password='madhur123')'

File "<console>", line 1

user = User.objects.create\_user('madhur',password='madhur123')'

^

SyntaxError: EOL while scanning string literal

>>> user = User.objects.all()

user = User.objects.all()

>>> user

user

<QuerySet [<User: abc>]>

>>> user = User.objects.create\_user('madhur',password='madhur123')

user = User.objects.create\_user('madhur',password='madhur123')

>>> user.is\_superuser=True

user.is\_superuser=True

>>> user.is\_staff=True

user.is\_staff=True

>>> user.save()

user.save()

>>> exit()

exit()

MADHUR@MADHUR-PC MINGW64 ~/desktop/Cashback\_Uploaded\_22may (master)

$ heroku open

* Any changes to database done by Django admin were temporary -> The database ‘sqlite3’ was not an add-on. Since SQLite **does not run as a service**, each dyno would run a separate running copy. Each of these copies need their own disk backed store. This would mean that each dyno powering your app would have a different set of data since the disks are not synchronized. In short, data was not stored permanently. Solution was to have a permanent database i.e. postgresql which is an add-on provided by Heroku. For that settings.py underwent few changes in DATABASE settings.

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.postgresql\_psycopg2',

'NAME': 'd4m9hrlcarare3', //name of database

'USER': 'jkjpdiniusjbnt',

'PASSWORD': '7d63a42050623ce0e1a19c93d0eb3f34106cb0785e982f9c1ea762d160be4e52',

'HOST': 'ec2-23-21-224-199.compute-1.amazonaws.com',

'PORT': '5432',

}

}

This info about database was obtained on Heruko/PersonalApps/ Settings/ Database\_url/ (actual url)

Syntax of actual url was -> scheme://username:password@host:port/database

MADHUR@MADHUR-PC MINGW64 ~/desktop

$ cd Cashback\_Uploaded\_22may

MADHUR@MADHUR-PC MINGW64 ~/desktop/Cashback\_Uploaded\_22may (master)

$ heroku config

=== stark-reef-61252 Config Vars

DATABASE\_URL: postgres://jkjpdiniusjbnt:7d63a42050623ce0e1a19c93d0eb3f34106cb0785e982f9c1ea762d160be4e52@ec2-23-21-224-199.compute-1.amazonaws.com:5432/d4m9hrlcarare3

MADHUR@MADHUR-PC MINGW64 ~/desktop/Cashback\_Uploaded\_22may (master)

$ git add .

MADHUR@MADHUR-PC MINGW64 ~/desktop/Cashback\_Uploaded\_22may (master)

$ git commit -m "Change db to postgres"

[master 5953ce0] Change db to postgres

3 files changed, 22 insertions(+), 6 deletions(-)

MADHUR@MADHUR-PC MINGW64 ~/desktop/Cashback\_Uploaded\_22may (master)

$ git push heroku master

Counting objects: 7, done.

Delta compression using up to 8 threads.

Compressing objects: 100% (7/7), done.

Writing objects: 100% (7/7), 1.70 KiB | 0 bytes/s, done.

Total 7 (delta 5), reused 0 (delta 0)

remote: Compressing source files... done.

remote: Building source:

remote:

remote: -----> Python app detected

remote: -----> Installing requirements with pip

remote:

remote: -----> $ python manage.py collectstatic --noinput

remote: 66 static files copied to '/tmp/build\_b414ae9f169a7e124733c73fd8db9e2f/static'.

remote:

remote: -----> Discovering process types

remote: Procfile declares types -> web

remote:

remote: -----> Compressing...

remote: Done: 74.7M

remote: -----> Launching...

remote: Released v9

remote: https://stark-reef-61252.herokuapp.com/ deployed to Heroku

remote:

remote: Verifying deploy... done.

To https://git.heroku.com/stark-reef-61252.git

1bf796c..5953ce0 master -> master

MADHUR@MADHUR-PC MINGW64 ~/desktop/Cashback\_Uploaded\_22may (master)

$ heroku open

Then the project was added to git, committed and pushed on heroku.

This will not work locally because the postgresql database conf would be different on local machine. It is better to use sqlite3 locally and revert the settings while pushing to production.

* Host not allowed error -> change allowed hosts in settings.py to \* (allow all hosts)
* STATIC\_ROOT was not defined in settings.py