

## Topic: Car Listing: Deployment using Heroku Part 19A

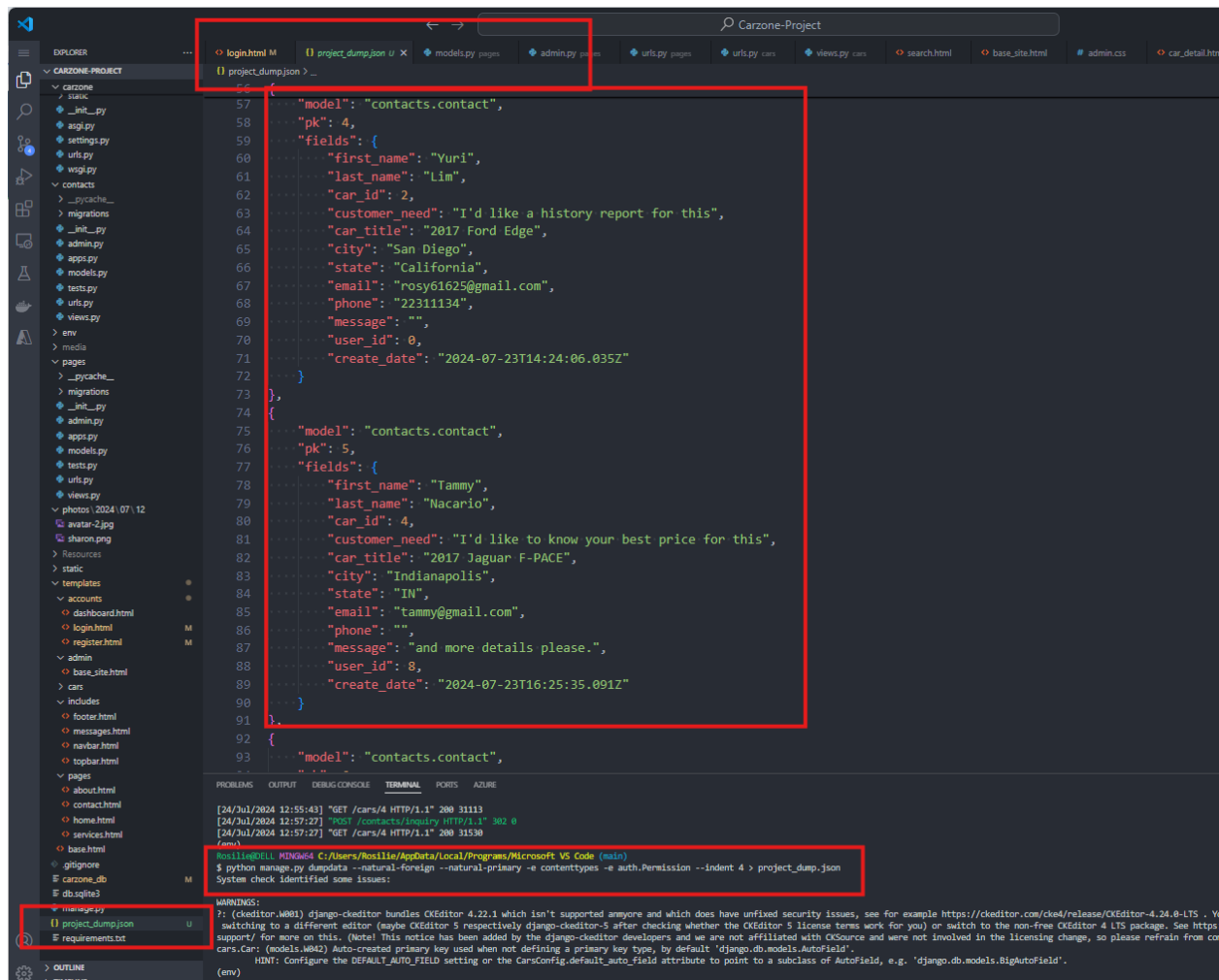
Speaker: / Notebook: Django Project: Car Listing



1. Dump or create a copy of your PostgreSQL database and its records into a JSON file. We issue the command below in our GitBash Terminal

```
$ python manage.py dumpdata --natural-foreign --natural-primary -e contenttypes -e auth.Permission --indent 4 > project_dump.json
```

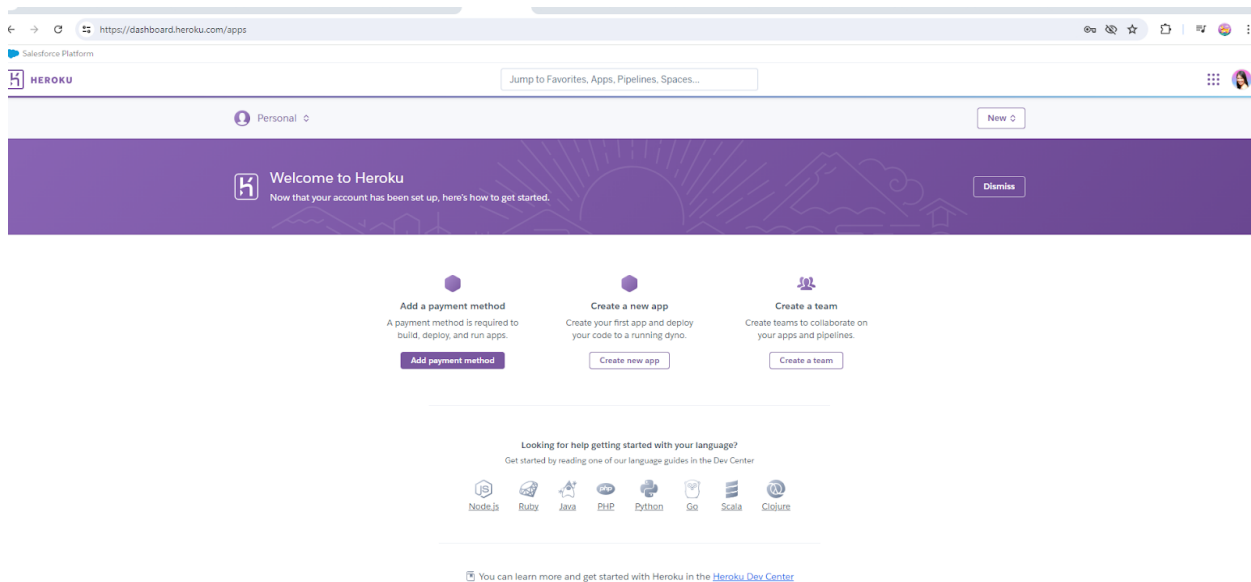
This will produce a new file called PROJECT\_DUMP.JSON that contains all the models and the records we have made.



```
{
  "model": "contacts.contact",
  "pk": 4,
  "fields": {
    "first_name": "Yuri",
    "last_name": "Lim",
    "car_id": 2,
    "customer_need": "I'd like a history report for this",
    "car_title": "2017 Ford Edge",
    "city": "San Diego",
    "state": "California",
    "email": "rosy61625@gmail.com",
    "phone": "22311134",
    "message": "",
    "user_id": 0,
    "create_date": "2024-07-23T14:24:06.035Z"
  }
},
{
  "model": "contacts.contact",
  "pk": 5,
  "fields": {
    "first_name": "Tammy",
    "last_name": "Nacarrio",
    "car_id": 4,
    "customer_need": "I'd like to know your best price for this",
    "car_title": "2017 Jaguar F-PACE",
    "city": "Indianapolis",
    "state": "IN",
    "email": "tammy@gmail.com",
    "phone": "",
    "message": "and more details please.",
    "user_id": 8,
    "create_date": "2024-07-23T16:25:35.091Z"
  }
},
{
  "model": "contacts.contact",
  "pk": 6,
  "fields": {
    "first_name": "John",
    "last_name": "Doe",
    "car_id": 1,
    "customer_need": "I want to know more about this car",
    "car_title": "2018 Toyota Camry",
    "city": "New York",
    "state": "NY",
    "email": "john.doe@example.com",
    "phone": "555-123-4567",
    "message": "I am interested in this car",
    "user_id": 1,
    "create_date": "2024-07-23T10:00:00.000Z"
  }
}
```

```
python manage.py dumpdata --natural-foreign --natural-primary -e contenttypes -e auth.Permission --indent 4 > project_dump.json
```

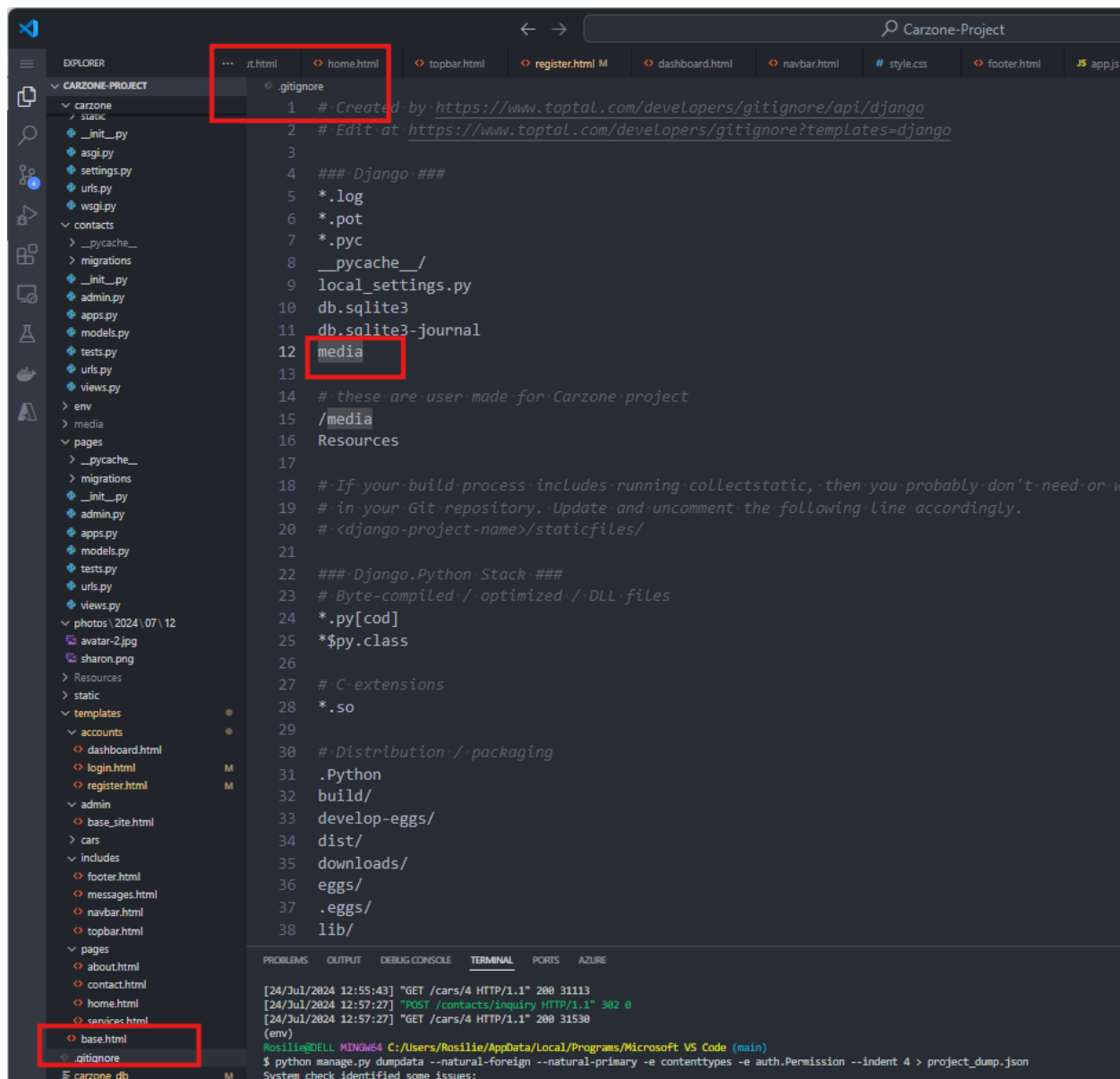
2. Now, to deploy the project in HEROKU, create a new account in HEROKU.



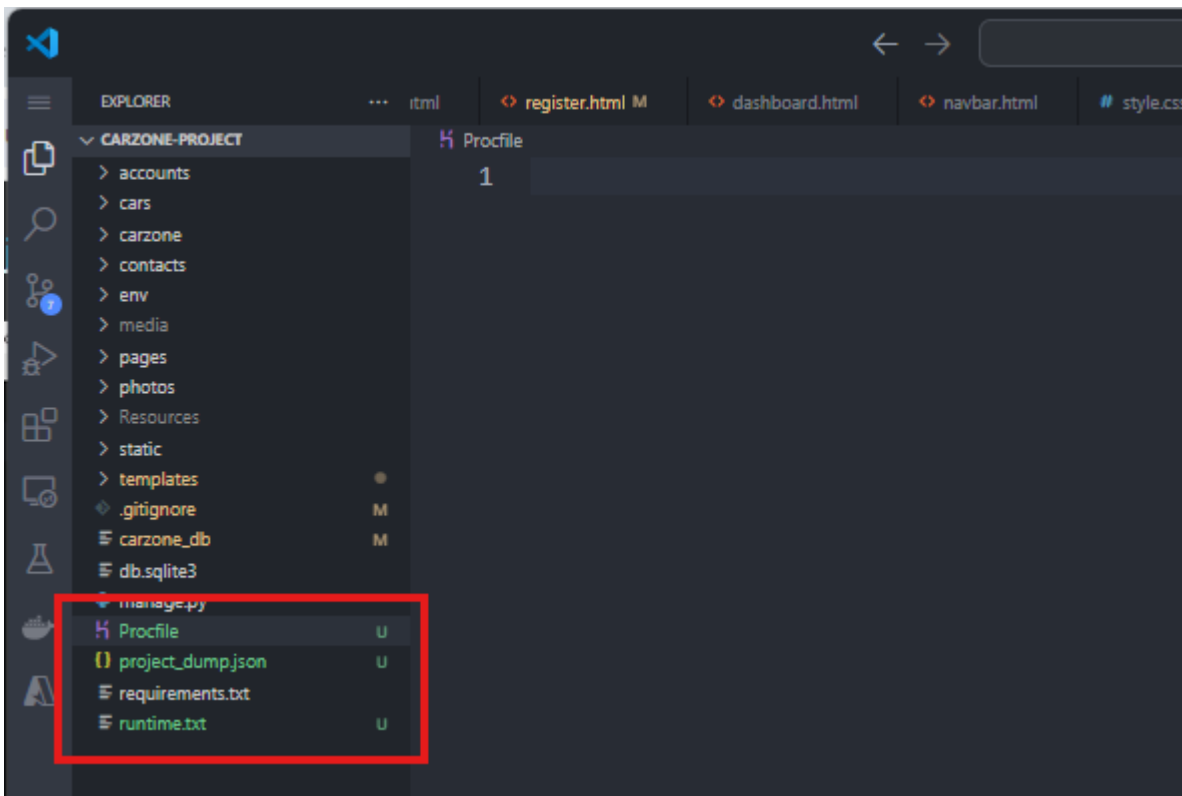
3. Update your .GITIGNORE FILE, remove the following from the .GITIGNORE file:

MEDIA

STATIC (ONLY IF YOU FOUND IT. IN OUR CASE, THERE IS NONE)



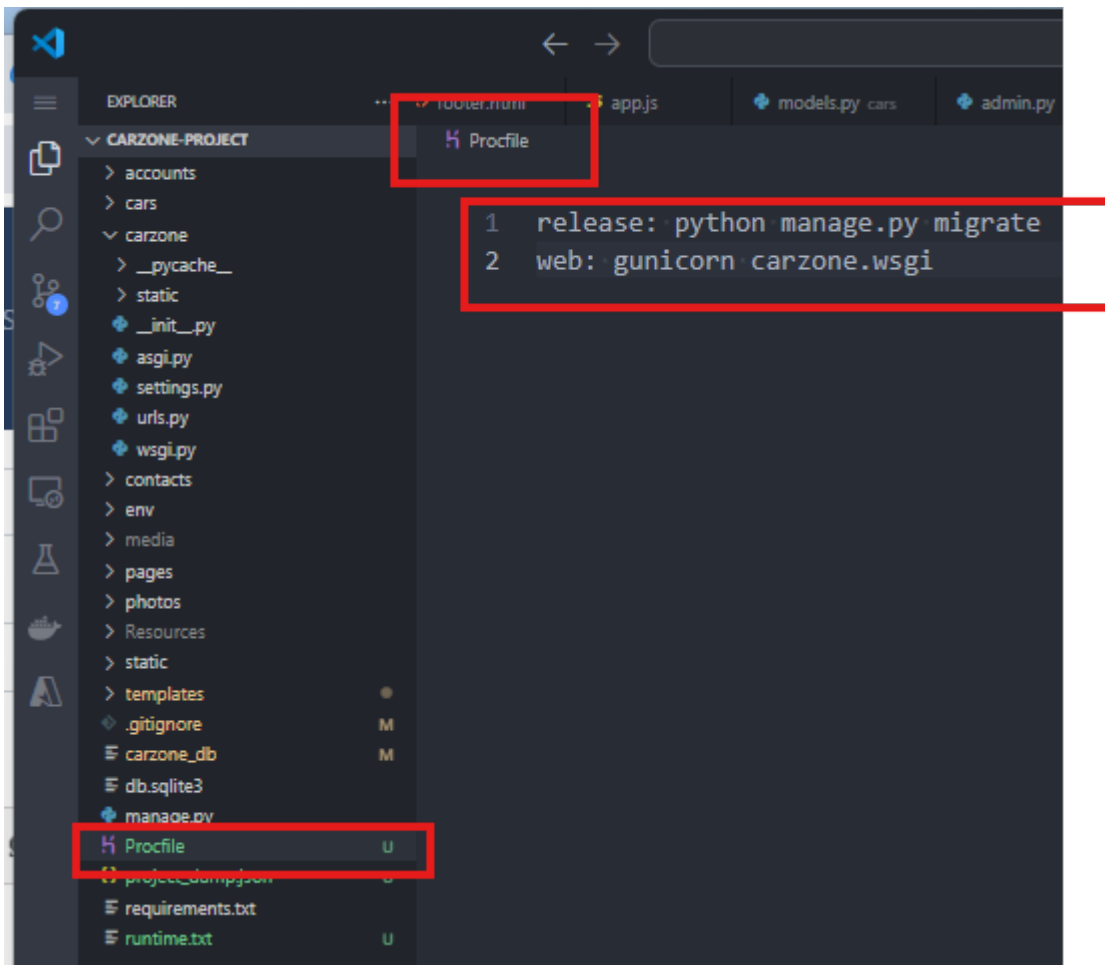
4. Create new files called runtime.txt and Profic in the root directory.

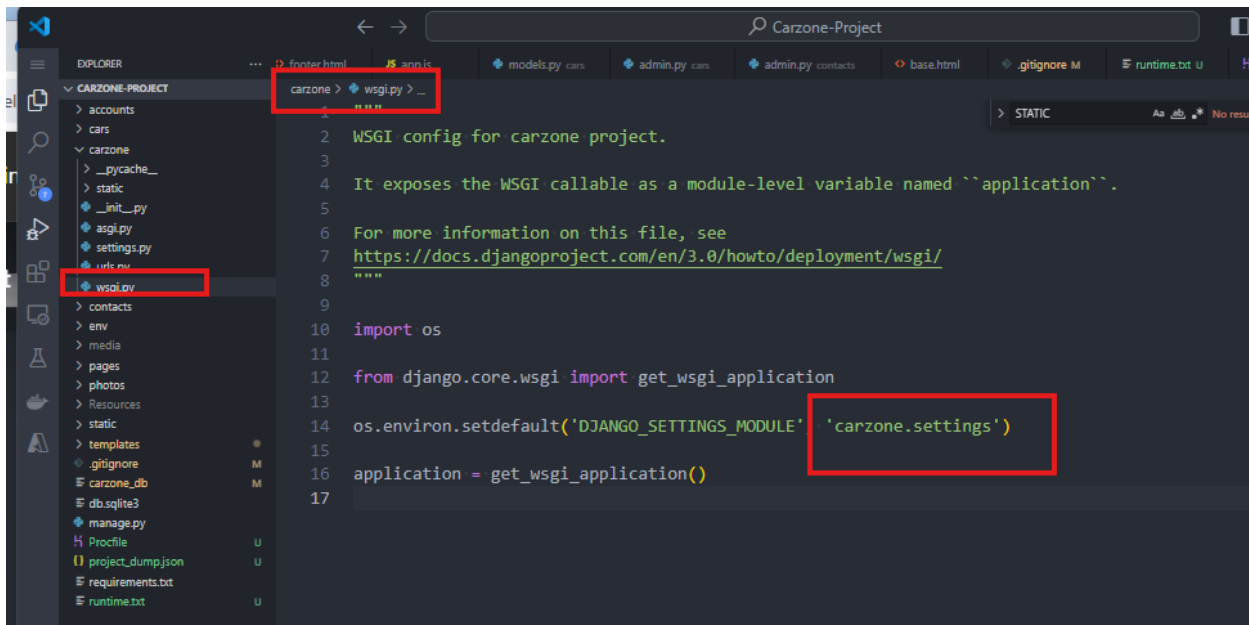


5. In the Procfile file, type the following. Carzone here is the name that we see in CARZONE\WSGI.PY file. So, if your project name is different, then update accordingly.

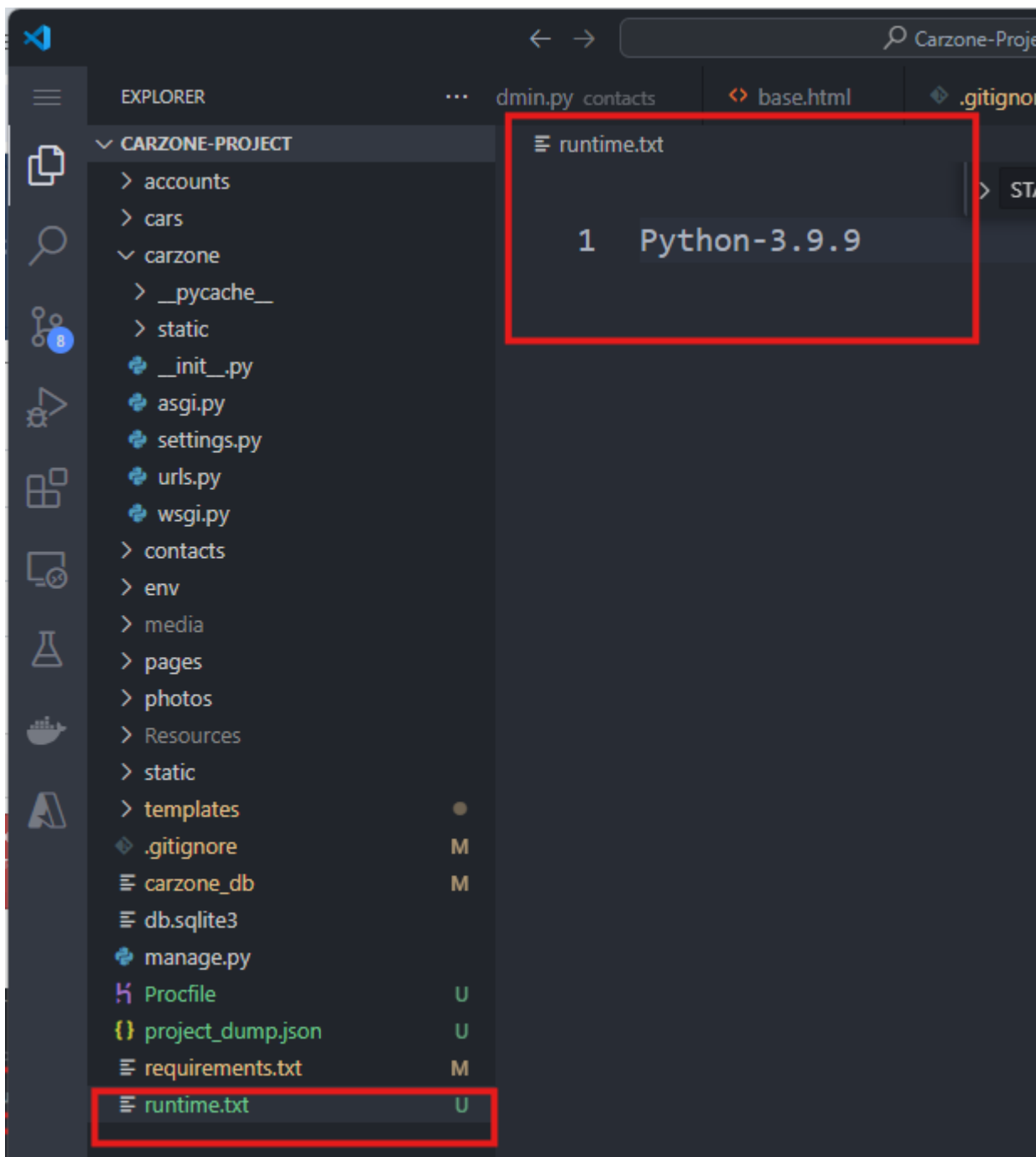
release: python manage.py migrate

web: gunicorn carzone.wsgi





6. In the RUNTIME.TXT, we type the version of Python that we used.

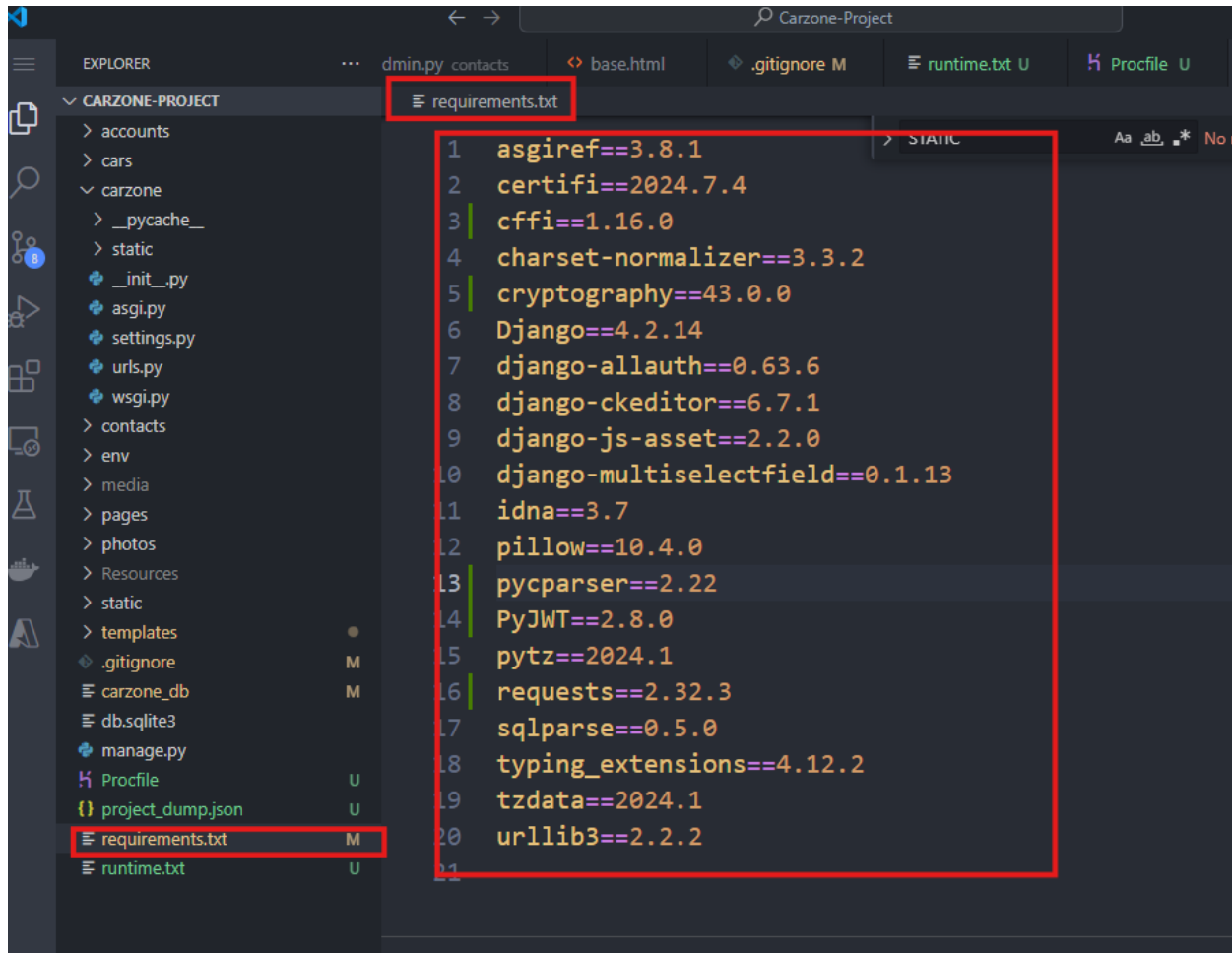


To find out your Python version, type

```
(env)
Rosilie@DELL MINGW64 C:/Users/Rosilie/OneDrive/Desktop/Carzone-Project
$ python --version
Python 3.9.9
(env)
```

7. Update our REQUIREMENTS.TXT by using the code below:

```
$ pip freeze > requirements.txt
```

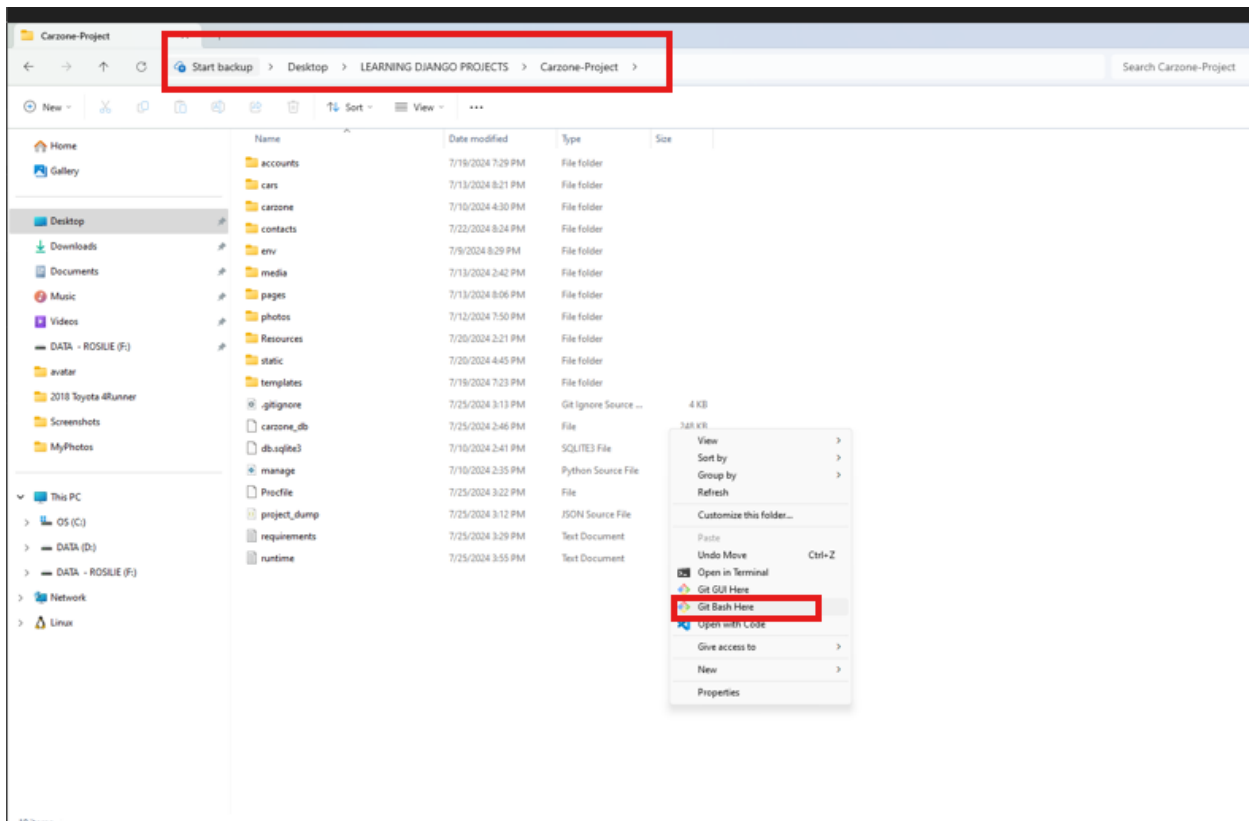


The screenshot shows the Visual Studio Code interface for a project named 'Carzone-Project'. In the Explorer view on the left, the file 'requirements.txt' is highlighted with a red box. In the Editor view on the right, the contents of 'requirements.txt' are displayed, also enclosed in a red box. The file contains a list of Python dependencies with their versions, each preceded by a line number from 1 to 20.

```
1 asgiref==3.8.1
2 certifi==2024.7.4
3 cffi==1.16.0
4 charset-normalizer==3.3.2
5 cryptography==43.0.0
6 Django==4.2.14
7 django-allauth==0.63.6
8 django-ckeditor==6.7.1
9 django-js-asset==2.2.0
10 django-multiselectfield==0.1.13
11 idna==3.7
12 pillow==10.4.0
13 pycparser==2.22
14 PyJWT==2.8.0
15 pytz==2024.1
16 requests==2.32.3
17 sqlparse==0.5.0
18 typing_extensions==4.12.2
19 tzdata==2024.1
20 urllib3==2.2.2
```

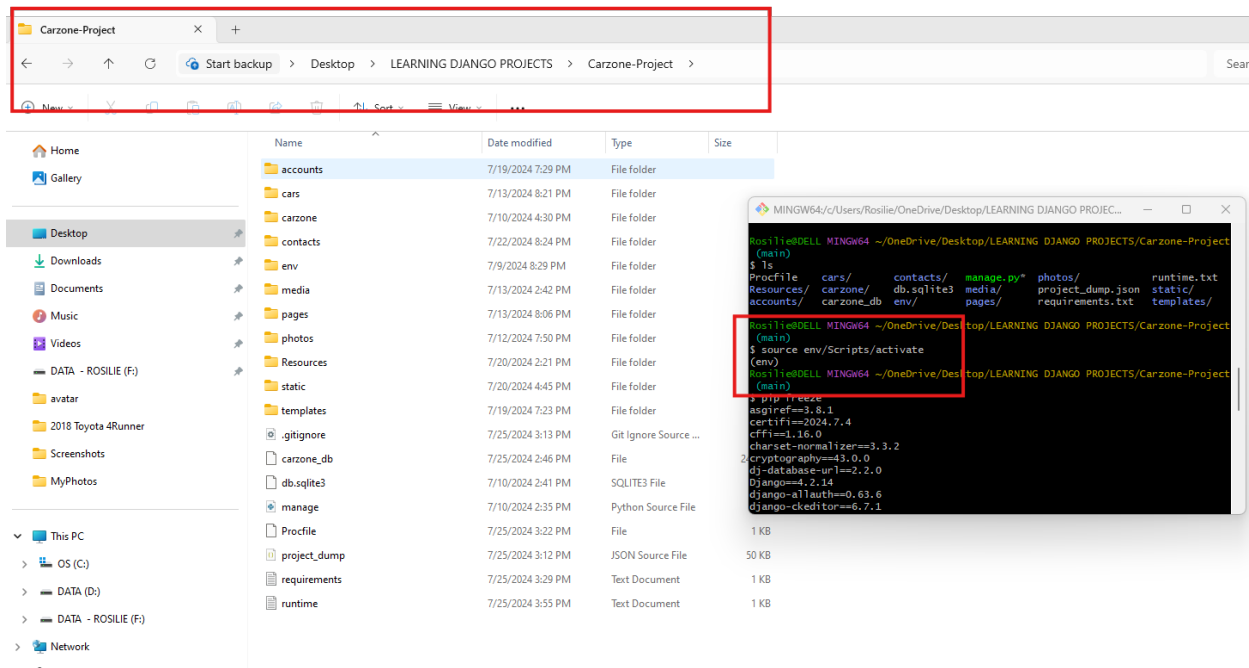
8. Use this [HEROKU DOCUMENTATION](#) as a guide. We need to [INSTALL HEROKU CLI](#).

We get inside our project folder and open GitBash here. PREVIOUSLY, WHAT YOU HAVE BEEN USING WAS THE GITBASH TERMINAL IN VS CODE.



9. In the GitBash terminal, make sure you activate your virtual environment.

`$ source env/Scripts/activate`



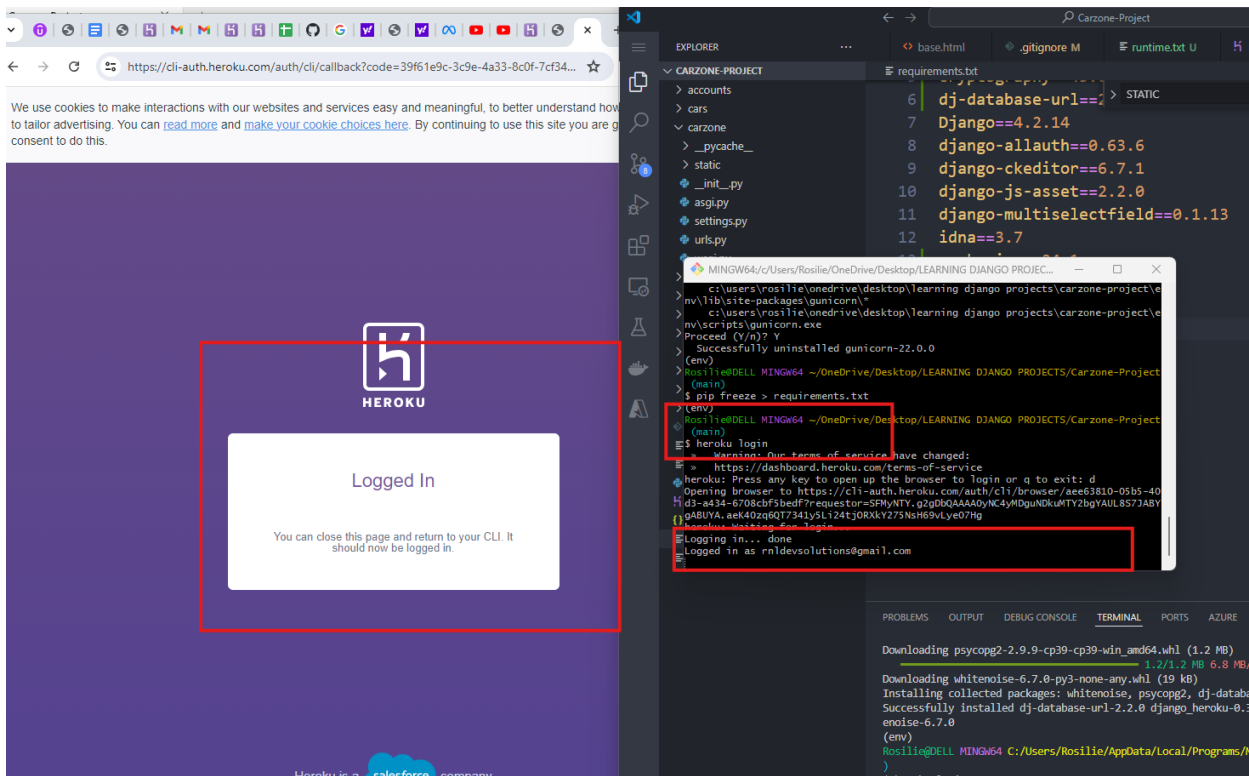
10. Now, when you type HEROKU LOGIN in the terminal, you should be able to use this:

```
MINGW64:/c:/Users/Rosilie/OneDrive/Desktop/LEARNING DJANGO PROJEC...
(main)
$ pip uninstall gunicorn
Found existing installation: gunicorn 22.0.0
Uninstalling gunicorn-22.0.0:
  Would remove:
    c:\users\rosilie\onedrive\desktop\learning django projects\carzone-project\env\lib\site-packages\gunicorn-22.0.0.dist-info\*
    c:\users\rosilie\onedrive\desktop\learning django projects\carzone-project\env\lib\site-packages\gunicorn\*
    c:\users\rosilie\onedrive\desktop\learning django projects\carzone-project\env\scripts\gunicorn.exe
Proceed (Y/n)? Y
Successfully uninstalled gunicorn-22.0.0
(env)
Rosilie@DELL MINGW64 ~/OneDrive/Desktop/LEARNING DJANGO PROJECTS/Carzone-Project
(main)
$ pip freeze > requirements.txt
(env)
Rosilie@DELL MINGW64 ~/OneDrive/Desktop/LEARNING DJANGO PROJECTS/Carzone-Project
(main)
$ heroku login
» Warning: Our terms of service have changed:
» https://dashboard.heroku.com/terms-of-service
heroku: Press any key to open up the browser to login or q to exit: .....
```

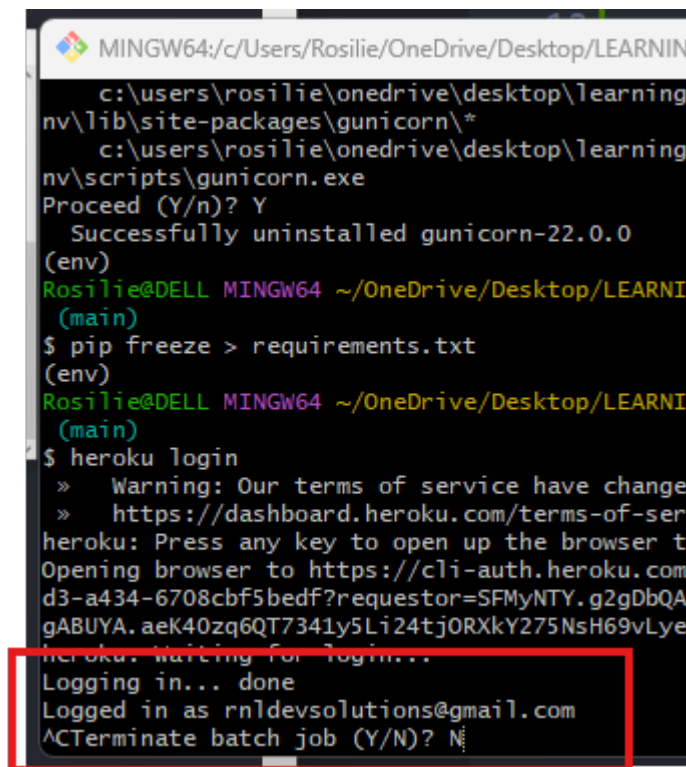
11. This should open the HEROKU CLI PAGE:

The screenshot displays the Heroku CLI login process. On the left, a web browser window shows the Heroku CLI login page at <https://cli-auth.heroku.com/auth/cli/browser/ae63810-05b5-40d3-a434-6708cbf5b...>. The page features the Heroku logo and a "Log in to the Heroku CLI" button. On the right, a terminal window shows the output of the `heroku login` command. The terminal output indicates that the user is being redirected to the login page and that the browser is opening. The terminal also shows the output of the `pip freeze > requirements.txt` command, which lists the installed packages: `django==4.2.14`, `django-allauth==0.63.6`, `django-ckeditor==6.7.1`, and `whitenoise==6.7.0`.

Click LOGIN. If you were not currently logged in Heroku, you will need to login using your email address and password.



12. Press CTRL+C to terminate from the HEROKU INTERFACE to install GUNICORN. Type N'



13. To know who is the logged in user or if you have logged out. Type

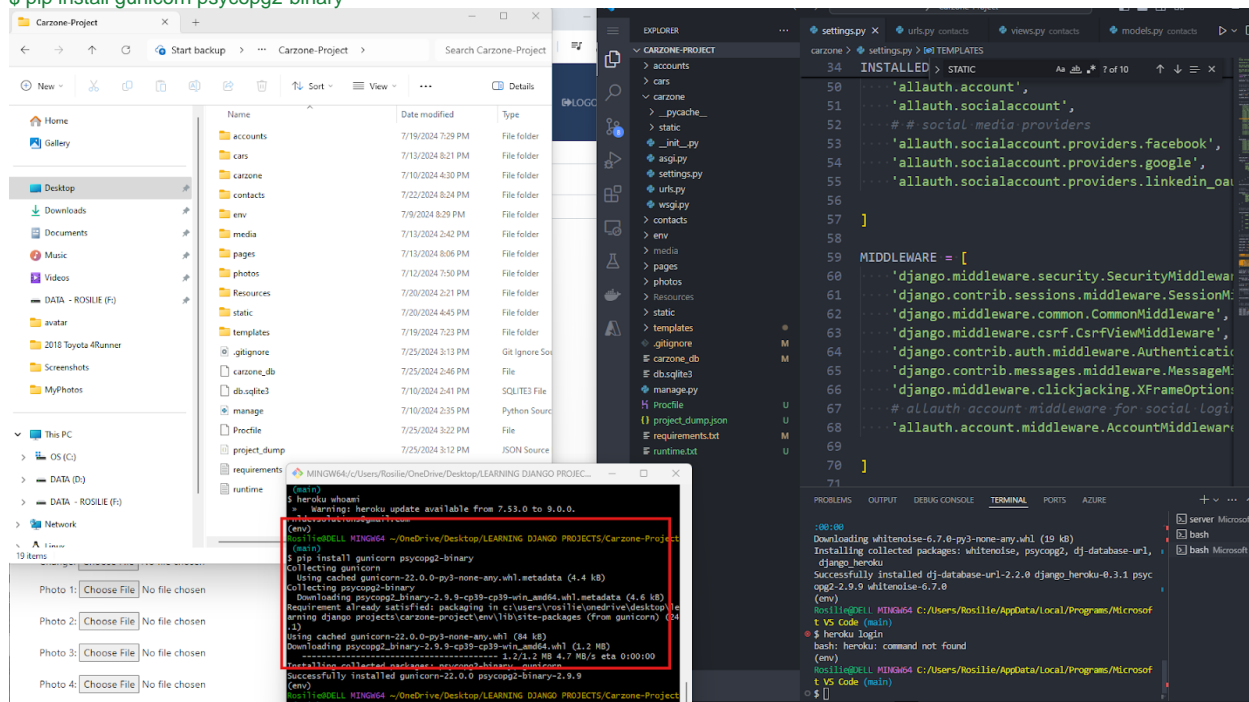


```
MINGW64/c/Users/Rosilie/OneDrive/Desktop/LEARNING DJANGO PROJEC...
(main)
$ heroku login
  » Warning: Our terms of service have changed:
  » https://dashboard.heroku.com/terms-of-service
heroku: Press any key to open up the browser to login or q to exit: d
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/ae63810-05b5-40d3-a434-6708cbf5bedf?requestor=SFMyNTY.g2gDbQAAAA0yNC4yMDguNDkuMTY2bG9AYUL8S7JABYgABUYA.aeK40zq6QT7341y5Li24tjORXkY275NsH69vLye07Hg
heroku: Waiting for login...
Logging in... done
Logged in as rnldevsolutions@gmail.com
^CTerminate batch job (Y/N)? N
N

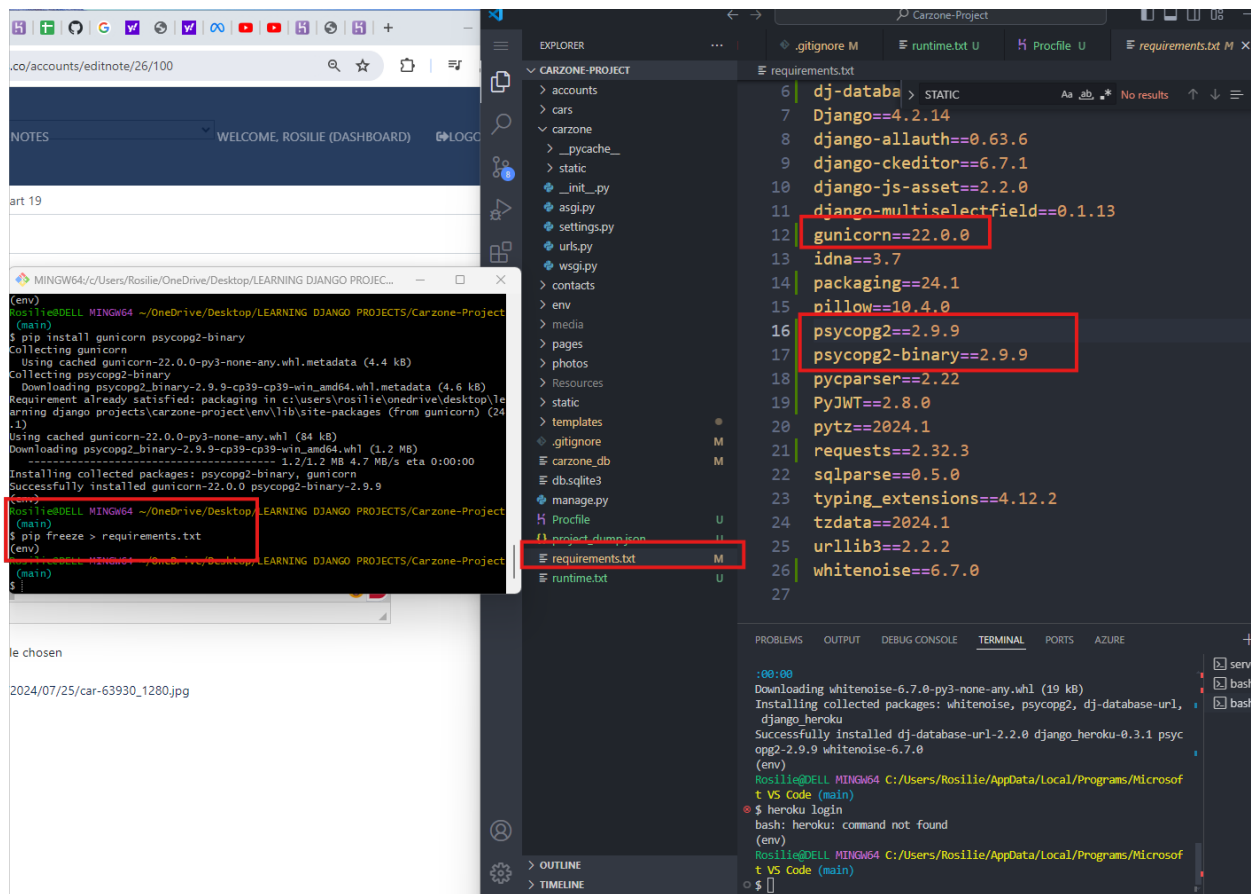
(env)
Rosilie@DELL MINGW64 ~/OneDrive/Desktop/LEARNING DJANGO PROJECTS/Carzone-Project
(main)
$ heroku whoami
  » Warning: heroku update available from 7.53.0 to 9.0.0.
rnldevsolutions@gmail.com
(env)
Rosilie@DELL MINGW64 ~/OneDrive/Desktop/LEARNING DJANGO PROJECTS/Carzone-Project
(main)
$
```

14. Install the 2 packages, GUNICORN and PSYCOPG2-BINARY. If you open your DIGITAL NOTEBOOK PROJECTS, these are the packages that were used too since both projects use PostgreSQL.

\$ pip install gunicorn psycopg2-binary



15. With the newly installed packages, update your REQUIREMENTS.TXT using PIP FREEZE.



16. See the next part in Heroku deployment.