



University of Windsor

COMP 8347

Internet Applications and Distributed Systems

Lab 5

Submitted to:

Dr. Adel Abusitta

June 12, 2023

Submitted By:

Hamza Baig (110089314)

Harsimran Singh (110090200)

Vivek Verma (110094354)

Talha Muhammad Shamoon Chaudhry (110087321)

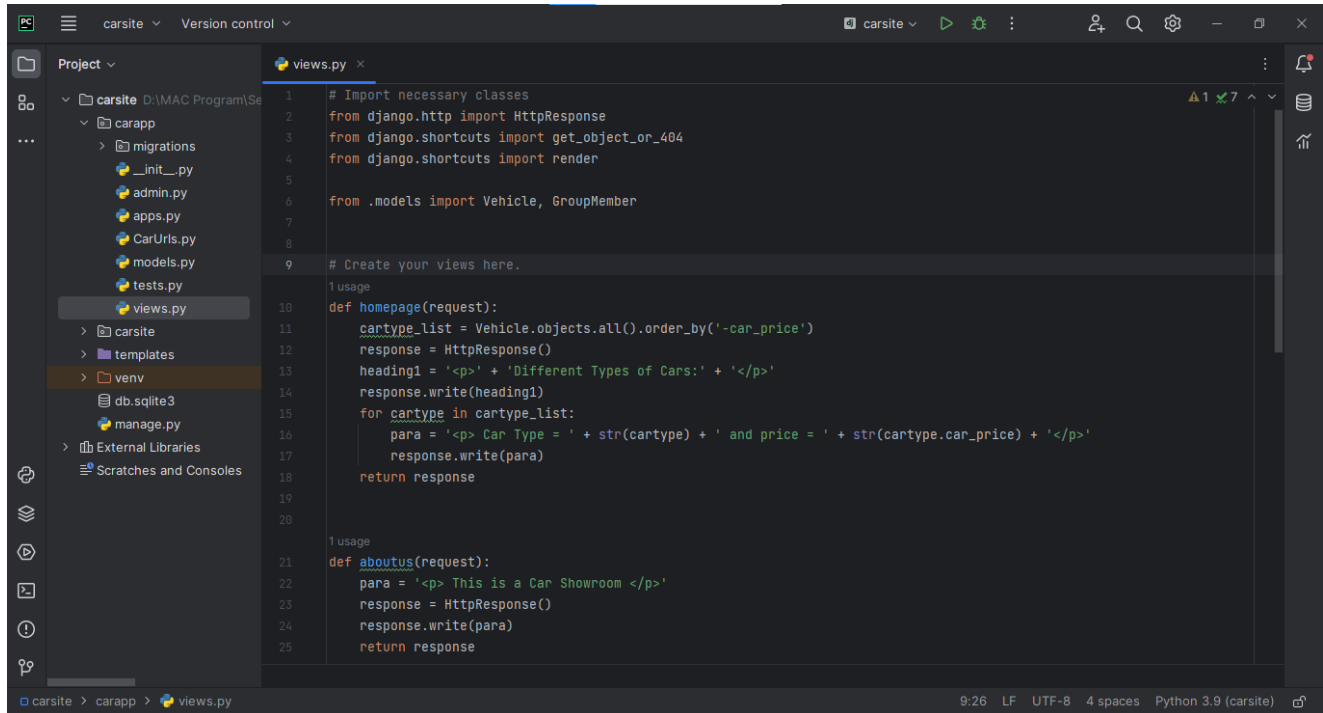
Vidya Sehgal (110090510)

Komalben Kishorbhai Patel (110091595)

Abhishek Singh (110089745)

Part 1

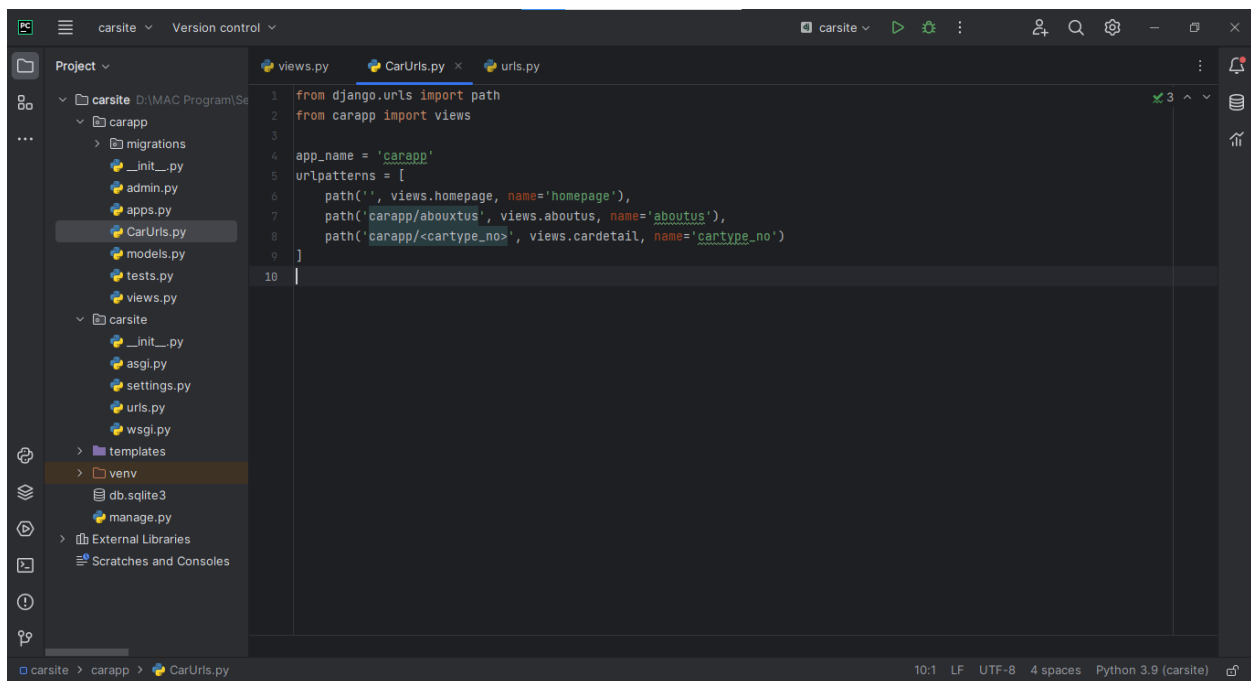
a.) Below screenshot displays changed views.py file.



```

1 # Import necessary classes
2 from django.http import HttpResponse
3 from django.shortcuts import get_object_or_404
4 from django.shortcuts import render
5
6 from .models import Vehicle, GroupMember
7
8
9 # Create your views here.
10
11 usage
12 def homepage(request):
13     cartype_list = Vehicle.objects.all().order_by('-car_price')
14     response = HttpResponse()
15     heading1 = '<p>' + 'Different Types of Cars:' + '</p>'
16     response.write(heading1)
17     for cartype in cartype_list:
18         para = '<p> Car Type = ' + str(cartype) + ' and price = ' + str(cartype.car_price) + '</p>'
19         response.write(para)
20     return response
21
22 usage
23 def aboutus(request):
24     para = '<p> This is a Car Showroom </p>'
25     response = HttpResponse()
26     response.write(para)
27     return response
  
```

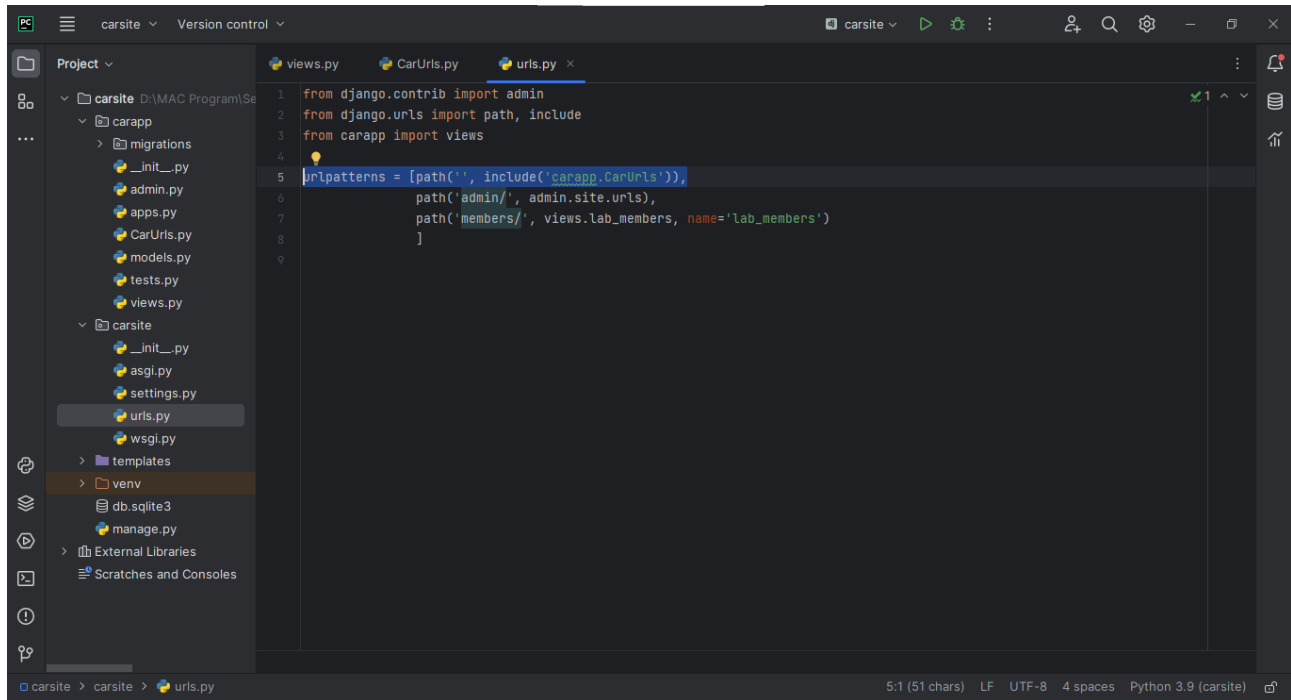
b.) Below screenshot shows changed urls.py file



```

1 from django.urls import path
2 from carapp import views
3
4 app_name = 'carapp'
5 urlpatterns = [
6     path('', views.homepage, name='homepage'),
7     path('carapp/aboutus', views.aboutus, name='aboutus'),
8     path('carapp/<cartype_no>', views.cardetail, name='cartype_no')
9 ]
10
  
```

c.)

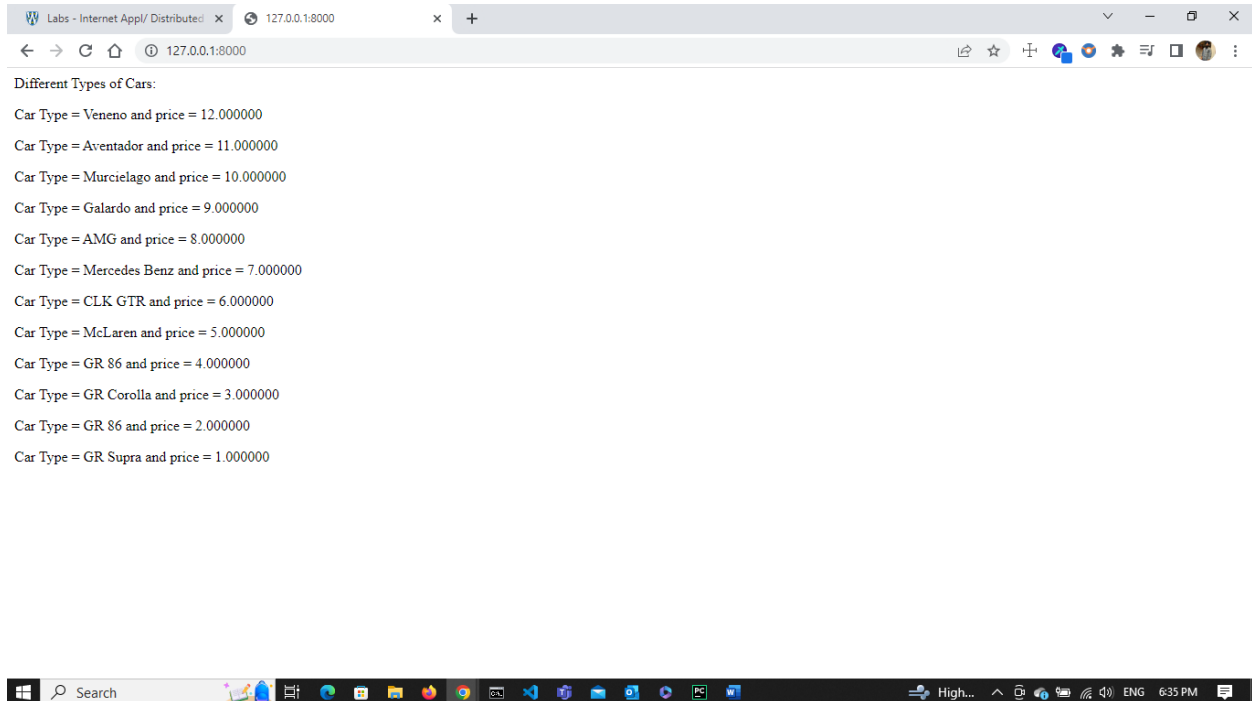


```

1 from django.contrib import admin
2 from django.urls import path, include
3 from carapp import views
4
5 urlpatterns = [path('', include('carapp.CarUrls')),
6                 path('admin/', admin.site.urls),
7                 path('members/', views.lab_members, name='lab_members')]
8
9

```

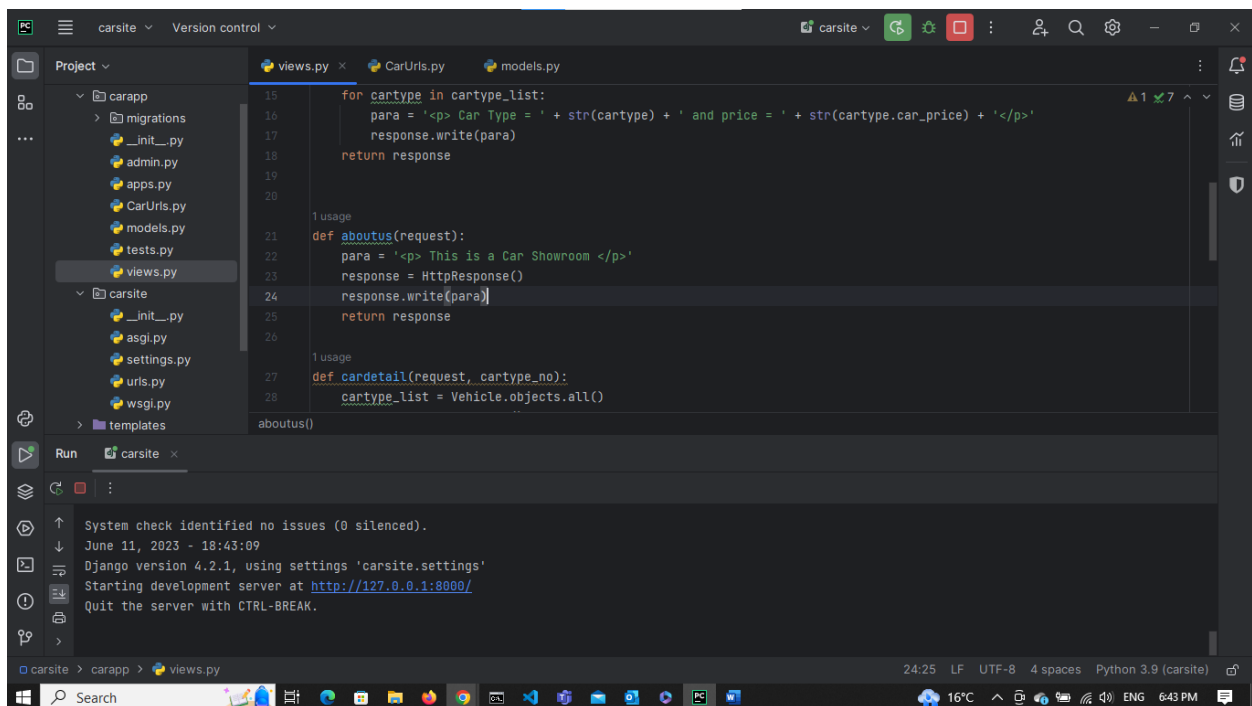
d.) Started server and 127.0.0.1:8000 show the all car details.



e.) We can use `[:10]` to get only 10 listings and order by function to sort the listings.

```
def homepage(request):
    cartype_list = Vehicle.objects.order_by('-car_price')[:10]
    response = HttpResponse()
    heading1 = '<p>' + 'Different Types of Cars:' + '</p>'
    response.write(heading1)
    for cartype in cartype_list:
        para = '<p> Car Type = ' + str(cartype) + ' and price = ' + str(cartype.car_price) + '</p>'
        response.write(para)
    return response
```

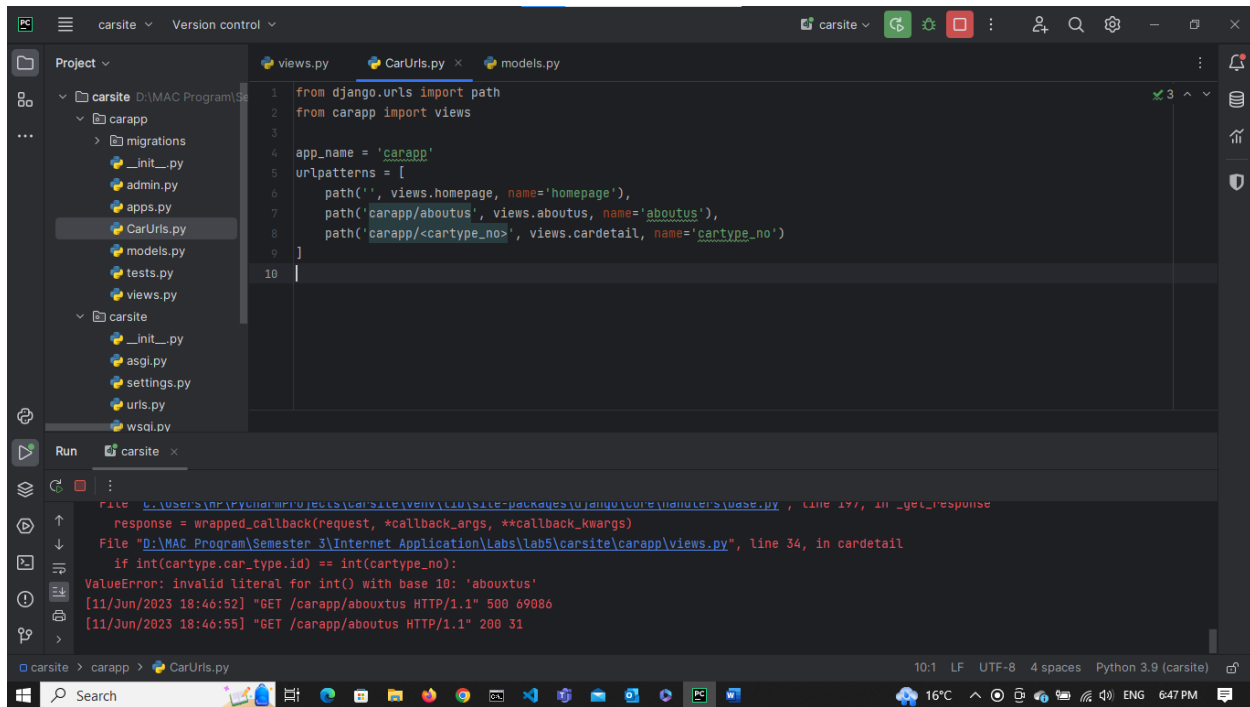
f.)



```
views.py
15 for cartype in cartype_list:
16     para = '<p> Car Type = ' + str(cartype) + ' and price = ' + str(cartype.car_price) + '</p>'
17     response.write(para)
18     return response
19
20 1 usage
21 def aboutus(request):
22     para = '<p> This is a Car Showroom </p>'
23     response = HttpResponse()
24     response.write(para)
25     return response
26
27 1 usage
28 def cardetail(request, cartype_no):
29     cartype_list = Vehicle.objects.all()
30     aboutus()
```

Run console output:

```
System check identified no issues (0 silenced).
June 11, 2023 - 18:43:09
Django version 4.2.1, using settings 'carsite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

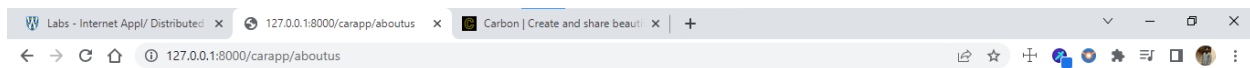


The screenshot shows a code editor with the following code in `views.py`:

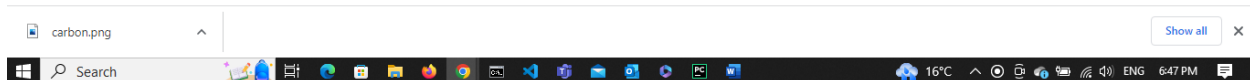
```
1 from django.urls import path
2 from carapp import views
3
4 app_name = 'carapp'
5 urlpatterns = [
6     path('', views.homepage, name='homepage'),
7     path('carapp/aboutus', views.aboutus, name='aboutus'),
8     path('carapp/<cartype_no>', views.ccardetail, name='cartype_no')
9 ]
10
```

The terminal window shows the following error messages:

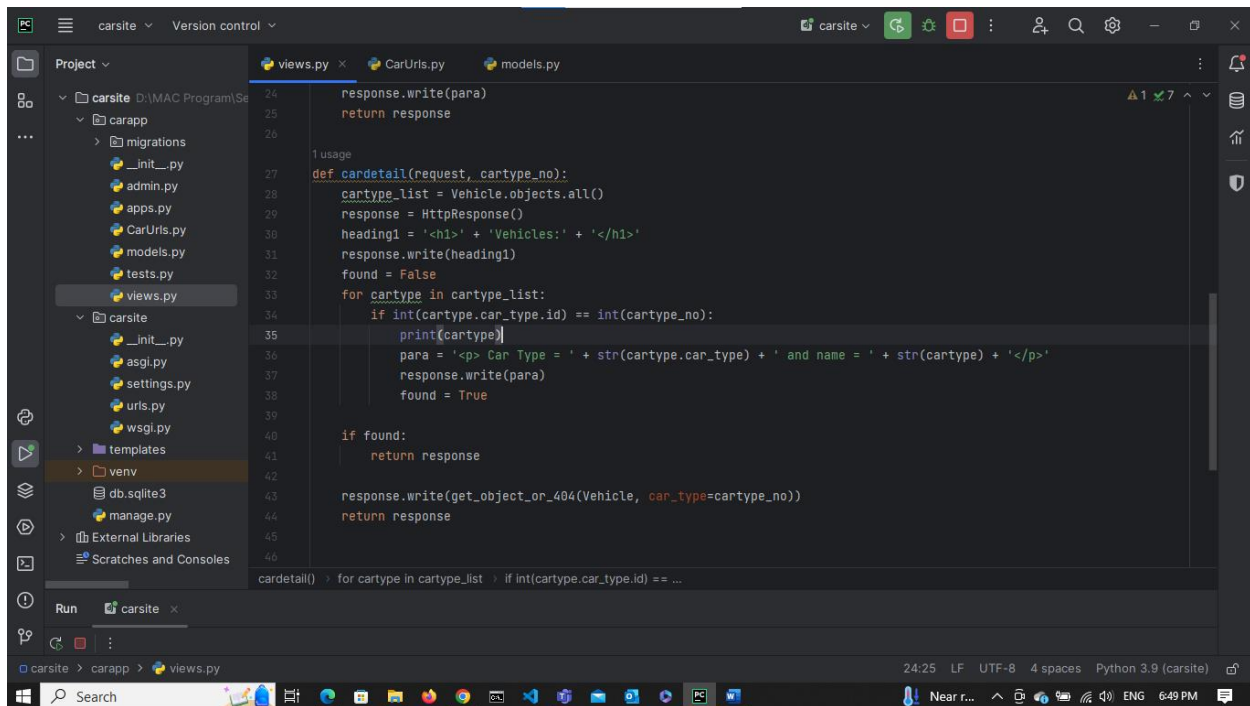
```
File "C:\Users\m\Python\Projects\carsite\venv\lib\site-packages\django\core\handlers\base.py", line 177, in _get_response
    response = wrapped_callback(request, *callback_args, **callback_kwargs)
File "D:\MAC Program\Semester 3\Internet Application\Labs\lab5\carsite\carapp\views.py", line 34, in ccardetail
    if int(cartype.car_type.id) == int(cartype_no):
ValueError: invalid literal for int() with base 10: 'aboutus'
[11/Jun/2023 18:46:52] "GET /carapp/aboutus HTTP/1.1" 500 69086
[11/Jun/2023 18:46:55] "GET /carapp/aboutus HTTP/1.1" 200 31
```



This is a Car Showroom



g.) Below screenshots shows the details of getting car details.



```

24 response.write(para)
25 return response
26
27 1 usage
28 def cardetail(request, cartype_no):
29     cartype_list = Vehicle.objects.all()
30     response = HttpResponse()
31     heading1 = '<h1>' + 'Vehicles:' + '</h1>'
32     response.write(heading1)
33     found = False
34     for cartype in cartype_list:
35         if int(cartype.car_type.id) == int(cartype_no):
36             print(cartype)
37             para = '<p> Car Type = ' + str(cartype.car_type) + ' and name = ' + str(cartype) + '</p>'
38             response.write(para)
39             found = True
40
41     if found:
42         return response
43
44     response.write(get_object_or_404(Vehicle, car_type=cartype_no))
45     return response
46
47 cardetail() -> for cartype in cartype_list -> if int(cartype.car_type.id) == ...

```



```

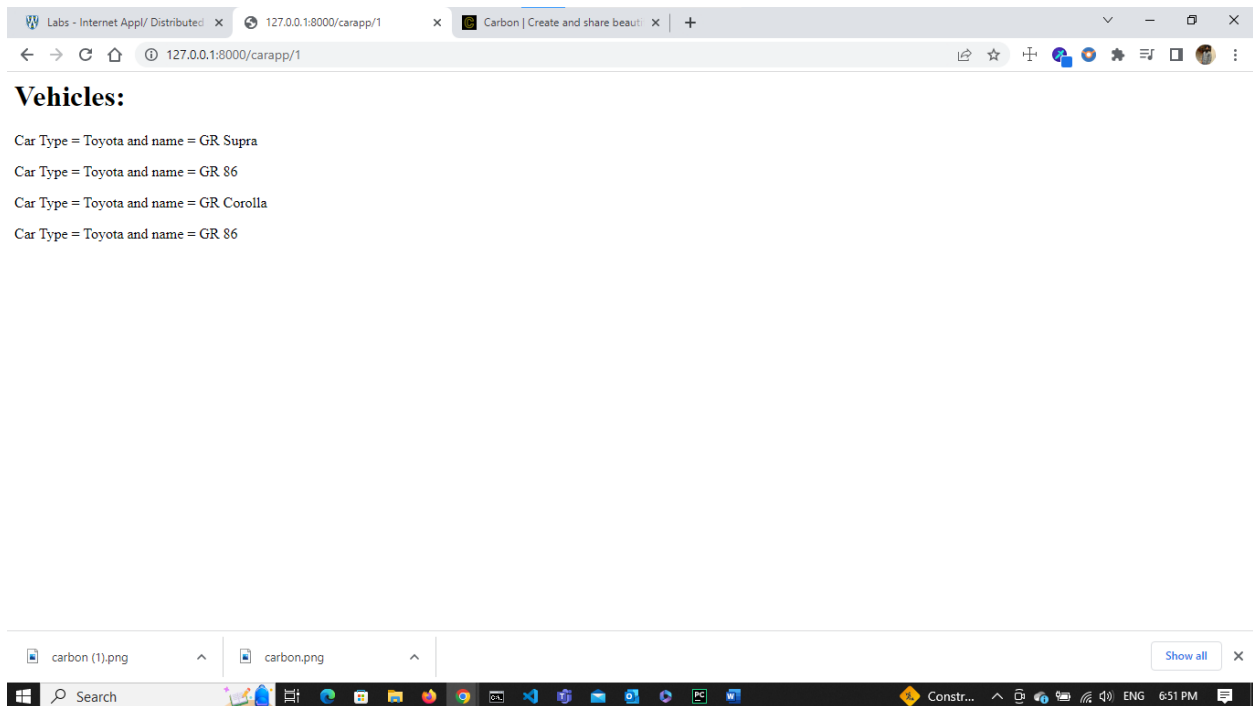
def cardetail(request, cartype_no):
    cartype_list = Vehicle.objects.all()
    response = HttpResponse()
    heading1 = '<h1>' + 'Vehicles:' + '</h1>'
    response.write(heading1)
    found = False
    for cartype in cartype_list:
        if int(cartype.car_type.id) == int(cartype_no):
            print(cartype)
            para = '<p> Car Type = ' + str(cartype.car_type) + ' and name = ' + str(cartype) + '</p>'
            response.write(para)
            found = True

    if found:
        return response

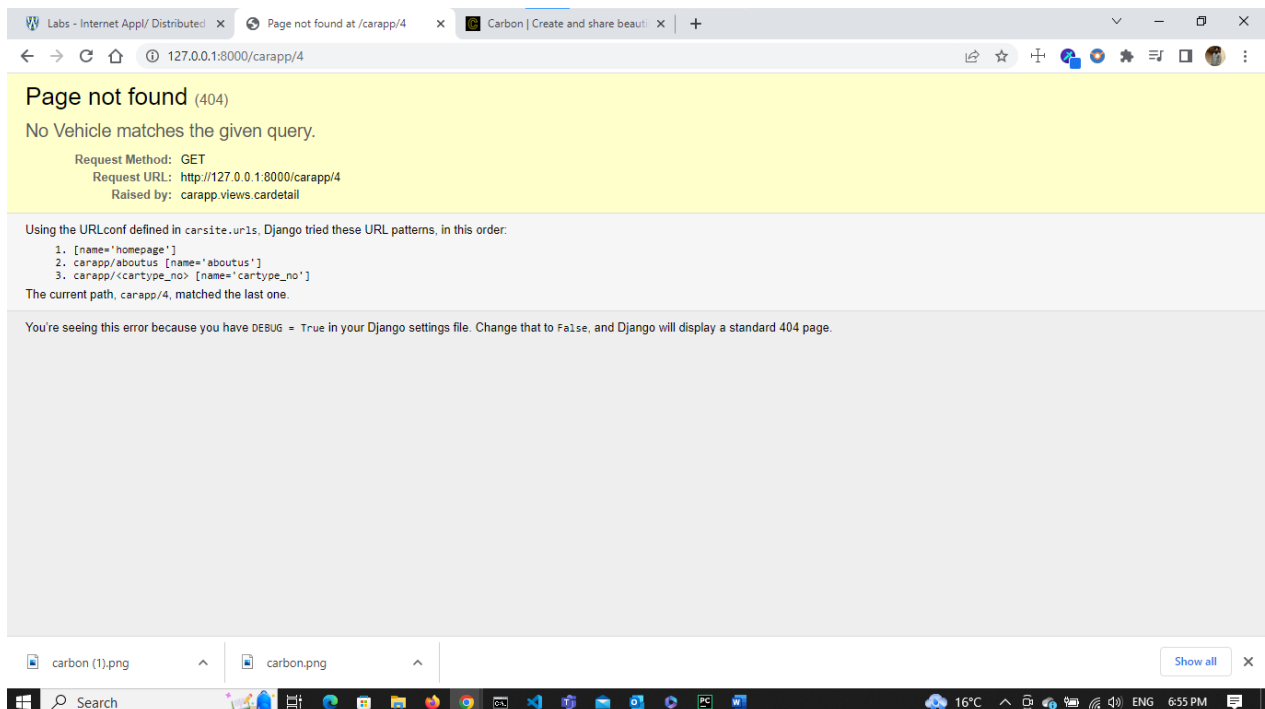
    response.write(get_object_or_404(Vehicle, car_type=cartype_no))
    return response

```

Below screenshots shows the details of all the cars with type 1.

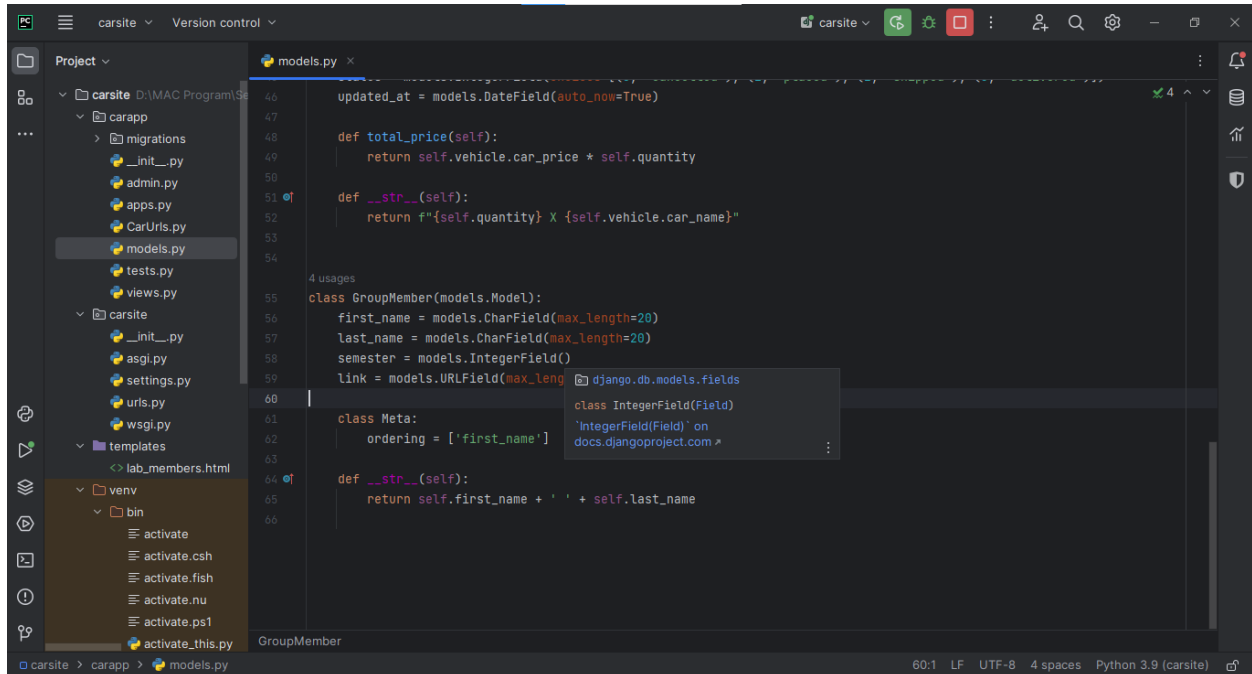


h.) If no type is present user will see the message that No Vehicle matches display 404 error page.



Part 2

a.) Model.py showing Group member fields.

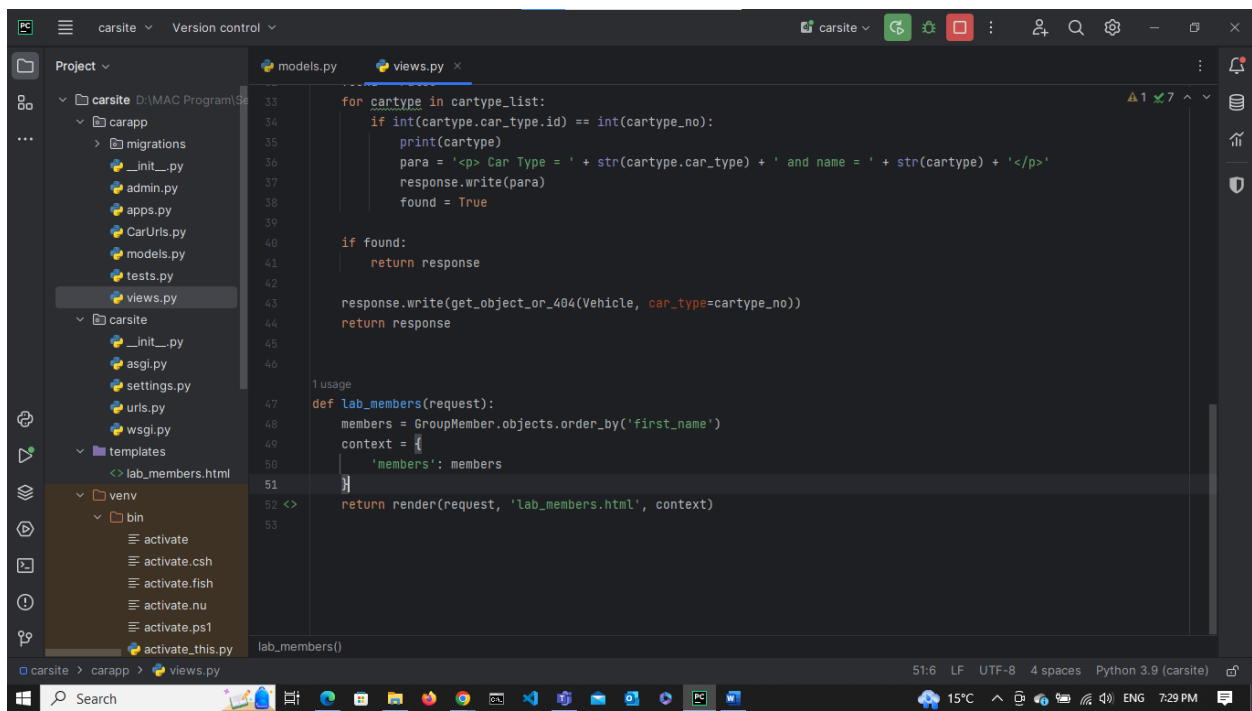


```

46     updated_at = models.DateField(auto_now=True)
47
48     def total_price(self):
49         return self.vehicle.car_price * self.quantity
50
51     def __str__(self):
52         return f"{self.quantity} X {self.vehicle.car_name}"
53
54
55     4 usages
56     class GroupMember(models.Model):
57         first_name = models.CharField(max_length=20)
58         last_name = models.CharField(max_length=20)
59         semester = models.IntegerField()
60         link = models.URLField(max_length=200)
61
62     class Meta:
63         ordering = ['first_name']
64
65     def __str__(self):
66         return self.first_name + ' ' + self.last_name

```

b.) View.py

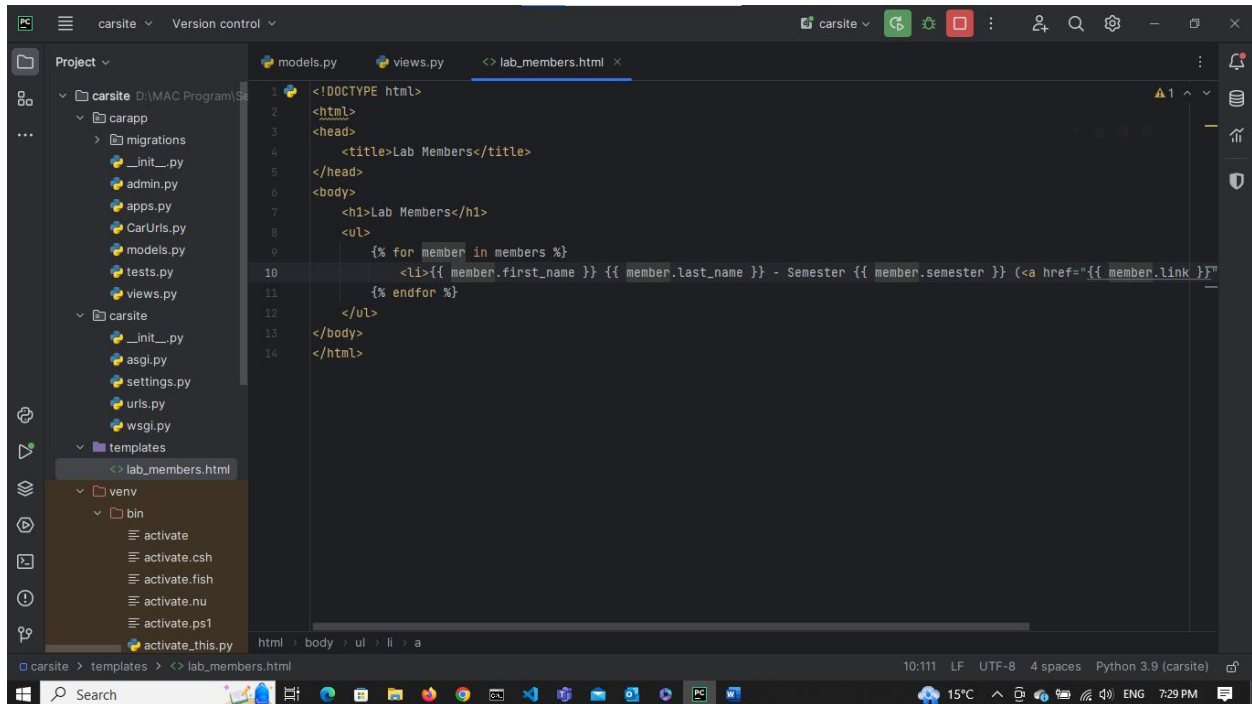


```

33     for cartype in cartype_list:
34         if int(cartype.car_type.id) == int(cartype_no):
35             print(cartype)
36             para = '<p> Car Type = ' + str(cartype.car_type) + ' and name = ' + str(cartype) + '</p>'
37             response.write(para)
38             found = True
39
40     if found:
41         return response
42
43     response.write(get_object_or_404(Vehicle, car_type=cartype_no))
44     return response
45
46
47     1 usage
48     def lab_members(request):
49         members = GroupMember.objects.order_by('first_name')
50         context = {}
51         context['members'] = members
52
53     return render(request, 'lab_members.html', context)

```


c.) Template for displaying lab members.



```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Lab Members</title>
5 </head>
6 <body>
7 <h1>Lab Members</h1>
8 <ul>
9     {% for member in members %}
10     <li>{{ member.first_name }} {{ member.last_name }} - Semester {{ member.semester }} (<a href="{{ member.link }}">Personal Page</a>)
11     {% endfor %}
12 </ul>
13 </body>
14 </html>

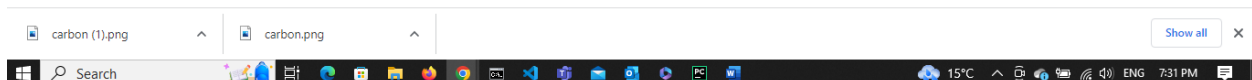
```

d.) Screenshot below displays when running <http://127.0.0.1:8000/members/> displays all the lab members details sorted according first name.



Lab Members

- Abhishek Singh - Semester 3 ([Personal Page](#))
- Hamza Baig - Semester 3 ([Personal Page](#))
- Harsimran Singh - Semester 3 ([Personal Page](#))
- Komal Patel - Semester 3 ([Personal Page](#))
- Talha Choudhary - Semester 3 ([Personal Page](#))
- Vidya Sehgal - Semester 3 ([Personal Page](#))
- Vivek Verma - Semester 3 ([Personal Page](#))



Part 3

FBV:

```
def cardetail(request, cartype_no):
    cartype_list = Vehicle.objects.all()
    response = HttpResponse()
    heading1 = '<h1>' + 'Vehicles:' + '</h1>'
    response.write(heading1)
    found = False
    for cartype in cartype_list:
        if int(cartype.car_type.id) == int(cartype_no):
            print(cartype)
            para = '<p> Car Type = ' + str(cartype.car_type) + ' and name = ' + str(cartype) + '</p>'
            response.write(para)
            found = True

    if found:
        return response

    response.write(get_object_or_404(Vehicle, car_type=cartype_no))
    return response
```

CBV:

```
class CarDetailView(View):
    def get(self, request, cartype_no):
        cartype_list = Vehicle.objects.all()
        heading1 = '<h1>' + 'Vehicles:' + '</h1>'
        found = False
        cartype_html = []
        for cartype in cartype_list:
            if int(cartype.car_type.id) == int(cartype_no):
                print(cartype)
                para = f'<p> Car Type = {cartype.car_type} and name = {cartype}</p>'
                cartype_html.append(para)
                found = True

        if found:
            response_html = ''.join(cartype_html)
            return HttpResponse(heading1 + response_html)

        vehicle = get_object_or_404(Vehicle, car_type=cartype_no)
        para = f'<p> Car Type = {vehicle.car_type} and name = {vehicle}</p>'
        return HttpResponse(heading1 + para)
```

Changes to URLs:

```
from django.urls import path
from .views import HomeableView, AboutUsView, CarDetailView, LabMembersView

app_name = 'carapp'

urlpatterns = [
    path('', HomeableView.as_view(), name='homepage'),
    path('aboutus/', AboutUsView.as_view(), name='aboutus'),
    path('car/<int:cartype_no>/', CarDetailView.as_view(), name='cardetail'),
    path('labmembers/', LabMembersView.as_view(), name='labmembers'),
]
```

```
from django.urls import path, include
from django.contrib import admin
from carapp.views import HomeableView, AboutUsView, CarDetailView, LabMembersView

urlpatterns = [
    path('', HomeableView.as_view(), name='homepage'),
    path('aboutus/', AboutUsView.as_view(), name='aboutus'),
    path('car/<int:cartype_no>/', CarDetailView.as_view(), name='cardetail'),
    path('members/', LabMembersView.as_view(), name='lab_members'),
    path('admin/', admin.site.urls),
]
```

Differences Noticed:

```
# Differences between FBV and CBV
# 1. The syntax for both views is different, as one is function based and the other uses class definitions.
# 2. CBVs inherit from the View class and the FBVs don't have a base class
# 3. CBVs have methods such as get() and post(), we have only used get() in this lab.
# 4. Code organization is much better in classes, so CBVs have a much organized code, which doesn't seem like a big
# problem for this lab as the code isn't that long but as the code base increases in size, CBVs will prove to be easily
# readable.
```

Differences between FBV and CBV

- # 1. The syntax for both views is different, as one is function based and the other uses class definitions.
- # 2. CBVs inherit from the View class and the FBVs don't have a base class
- # 3. CBVs have methods such as get() and post(), we have only used get() in this lab.
- # 4. Code organization is much better in classes, so CBVs have a much organized code, which doesn't seem like a big problem for this lab as the code isn't that long but as the code base increases in size, CBVs will prove to be easily readable.