Dynamic Navbar – Cars App and Model

Section 5

1. First make the copyride year dynamic

{% now ‘Y’ %}

<p class="copy">© {% now 'Y' %} Carzone Corp. All Rights Reserved.</p>

Capital Y will give you 2020

Small y will give you only 20.

1. Footer links and
2. banner read more button redirect to services page
3. get in touch button redirect to contact page

**Dynamic navbar**

When I go to any page and stay so that page link actively highlight jab tak me switch na karu.

Navbar.html

For the home link we use this {% if '/' == request.path %} we can only use == for home.

<li class="nav-item">

<a

{% if '/' == request.path %}

class="nav-link active"

{% else %}

class="nav-link"

{% endif %}

href="{% url 'home' %}" >

Home

</a>

</li>

For other links we use the **in** operator

We can not use for car now

About

li class="nav-item">

<a

{% if 'about' in request.path %}

class="nav-link active"

{% else %}

class="nav-link"

{% endif %}

href="{% url 'about' %}">

About

</a>

</li>

Do same for services and contact

Cars app

1. **create the app**

Here you are the virtual environment so you can simply use this command for creating the app.

Python manage.py startapp cars.

1. Make the urls.py class
2. Register the cars url in main urls.py

path('cars/',include('cars.urls')),

1. Create the home page for car app

path('',views.cars, name='cars'),

1. Create the views for the cars

def cars(request):

return render(request,'cars/cars.html')

1. Create the html file for cars
   1. Templates
      1. Cars
         1. Cars.html
2. Now open the carhouse file and open cars.html
3. Extends the base class in cars.html
4. Copy from sub banner start and feature cars start to end of featured cars end and paste it between block content and endblock.
5. If logo is not working than go to the carzone/static and use ../ so it will work clear the cache.

../static/img/logos/black-logo.png'

**Cars Model.**

Create the all fields. And copy the choice\_list car model from the recourse section of this lecture.

This 4 are the dropdowns

1. state\_choice = (……)
2. year\_choice = []

for i in range(2000,(datetime.now().year+1)):

year\_choice.append((i,i))

1. features\_choices = (……)
2. door\_choices = (……)

class Car(models.Model):

car\_title = models.CharField(max\_length=255)

state = models.CharField(choices=state\_choice, max\_length=100)

city = models.CharField(max\_length=100)

color = models.CharField(max\_length=100)

model = models.CharField(max\_length=100)

year = models.IntegerField(('year'), choices = year\_choice)

condition = models.CharField(max\_length=100)

price = models.IntegerField()

description = models.TextField(max\_length=500)

car\_photo = models.ImageField(upload\_to = 'photos/%y/%m/%d/')

car\_photo\_1 = models.ImageField(upload\_to = 'photos/%y/%m/%d/', blank=True)

car\_photo\_2 = models.ImageField(upload\_to = 'photos/%y/%m/%d/', blank=True)

car\_photo\_3 = models.ImageField(upload\_to = 'photos/%y/%m/%d/', blank=True)

car\_photo\_4 = models.ImageField(upload\_to = 'photos/%y/%m/%d/', blank=True)

feature = models.CharField(choices = features\_choices,max\_length=200)

body\_style = models.CharField(max\_length=100)

engine = models.CharField(max\_length=100)

transmission = models.CharField(max\_length=100)

interior = models.CharField(max\_length=100)

miles = models.IntegerField()

doors = models.CharField(choices=door\_choices, max\_length=10)

passengers = models.IntegerField()

vin\_no = models.CharField(max\_length=100)

milage = models.IntegerField()

fuel\_type = models.CharField(max\_length=50)

no\_of\_owners = models.CharField(max\_length=100)

is\_featured = models.BooleanField(default=False)

created\_date = models.DateTimeField(default=datetime.now, blank=True)

**Problems.**

1. I want to use editor in description where I can change the font and use other designing purpose field

So we will use the CKEditor

1. use the multi cursor in featured section

use the Multi Cursor Field

**CKEDITOR**

1. pip install Django-ckeditor
2. resister in installed\_apps at the last

'ckeditor',

1. Go to the models.py / description field  
   description = RichTextField()
2. Import the RichTextField
3. from ckeditor.fields import RichTextField

than check at admin site

**MultiSelectField**

1. Pip install Django-multiselectfield
2. from multiselectfield import MultiSelectField
3. feature = MultiSelectField(choices = features\_choices)

**Car Admin Customization**

First add the photos from the admin side.

Do the same thing which you did in pages/admin.py

And also use the is\_featured editable from outside.

list\_editable = ('is\_featured',)

Don’t forgot to register the CarAdmin in admin.site.register(Car,CarAdmin).

from django.contrib import admin

from .models import Car

from django.utils.html import format\_html

class CarAdmin(admin.ModelAdmin):

def thumbnail(self,object):

return format\_html('<img src="{}" width="40" style="border-radius:50%"/>'.format(object.car\_photo.url))

thumbnail.short\_description = 'photo'

list\_display = ('id','thumbnail','car\_title','city','price','color','model','condition','year','is\_featured')

list\_display\_links = ('id','thumbnail','car\_title')

list\_editable = ('is\_featured',)

search\_fields = ('id','car\_title','color')

list\_filter = ('city','model','body\_style','fuel\_type')

admin.site.register(Car,CarAdmin)

**Featured car latest car and image gallery**

1. I want to show only featured car on home page.

Pages/view.py

is\_featured = Car.objects.order\_by('-created\_date').filter(is\_featured=True)

use this is\_featured in context

def home(request):

teams = Team.objects.all()

is\_featured = Car.objects.order\_by('-created\_date').filter(is\_featured=True)

data = {

'teams': teams,

'is\_featured': is\_featured

}

return render(request,'pages/home.html',data)

1. Home.html

Featured car section

Remove all the static featured car div except one.

Now dynamically fetch the data.

Here for the images I used {{ car.car\_photo.url }} -> here car\_photo is a folder name.

On the home page if the images are not in proper box than use the style tag

style="min-height:262px; max\_height:262px">

{% for car in is\_featured %}

<div class="slick-slide-item">

<div class="car-box-3">

<div class="car-thumbnail">

<a href="car-details.html" class="car-img">

<div class="tag-2">Featured</div>

<div class="price-box">

<span>${{ car.price }}</span>

</div>

<img class="d-block w-100" src="{{ car.car\_photo.url }}" alt="car" style="min-height:262px; max\_height:262px">

</a>

<div class="carbox-overlap-wrapper">

<div class="overlap-box">

<div class="overlap-btns-area">

<div class="car-magnify-gallery">

<a href="{{ car.car\_photo\_1 }}" class="overlap-btn">

<i class="fa fa-expand"></i>

<img class="hidden" src="img/car/car-1.jpg">

</a>

<a href="{{ car.car\_photo\_2 }}" class="hidden">

<img class="hidden" src="img/car/car-2.jpg">

</a>

<a href="{{ car.car\_photo\_3 }}" class="hidden">

<img class="hidden" src="img/car/car-3.jpg">

</a>

<a href="{{ car.car\_photo\_4 }}" class="hidden">

<img class="hidden" src="img/car/car-4.jpg">

</a>

</div>

</div>

</div>

</div>

</div>

<div class="detail">

<h1 class="title">

<a href="car-details.html">{{ car.car\_title }}</a>

</h1>

<div class="location">

<a href="car-details.html">

<i class="flaticon-pin"></i>{{ car.state }} {{ car.city }}

</a>

</div>

<ul class="facilities-list clearfix">

<li>{{ car.fuel\_type }}</li>

<li>{{ car.miles }}</li>

<li>{{ car.transmission }}</li>

<li>{{ car.body\_style }}</li>

<li>{{ car.color }}</li>

<li>{{ car.year }}</li>

</ul>

</div>

</div>

</div>

{% endfor %}

**Latest car section**

**Do the same as you do above**

View.py

All\_cars (same as is\_featured except filter)

Pass in context

Home.html

Car rent start

Cut from the <div class="col-lg-4 col-md-6">

Do same work as above

**Fix the image Gallery**

In the latest car and featured car if we click any of the car than it will now show the car photos.

And also the main car photo is mandatory but other car photo is option. So for that we have to check that that picture is available or not.

You can see

Entry 5 or [2018 Toyota Corolla](http://127.0.0.1:8000/admin/cars/car/5/change/) in this car 5th image is not there. So it will give the error (valuError)

That is why we are using the if condition.

Car on rent (latest car) section

In if condition we can not use **url**

<!-- first pic is mandatory so we don't need if condition here -->

<a href="{{ car.car\_photo.url }}" class="overlap-btn">

<i class="fa fa-expand"></i>

<img class="hidden" src="{{ car.car\_photo.url }}">

</a>

{% if car.car\_photo\_1 %}

<a href="{{ car.car\_photo\_1.url }}" class="hidden">

<img class="hidden" src="{{ car.car\_photo\_1.url }}">

</a>

{% endif %}

{% if car.car\_photo\_2 %}

<a href="{{ car.car\_photo\_2.url }}" class="hidden">

<img class="hidden" src="{{ car.car\_photo\_2.url }}">

</a>

{% endif %}

{% if car.car\_photo\_3 %}

<a href="{{ car.car\_photo\_3.url }}" class="hidden">

<img class="hidden" src="{{ car.car\_photo\_3.url }}">

</a>

{% endif %}

{% if car.car\_photo\_4 %}

<a href="{{ car.car\_photo\_4.url }}" class="hidden">

<img class="hidden" src="{{ car.car\_photo\_4.url }}">

</a>

{% endif %}

Do the same for featured car.

**Car humanize**

This is use for comma in between digits

* First add the humanize in installed\_apps

'django.contrib.humanize',

* Than change in the price and miles section

<p class="price">${{ car.price | intcomma}}</p>

* Than load the humanize

{% load humanize %}

**Git push**

**Come outside the env.**

If you get this kind of error

fatal: Unable to create 'C:/Users/Vishal/Desktop/carZone/.git/index.lock': File exists. Another git process seems to be running in this repository, e.g. an editor opened by 'git commit'. Please make sure all processes are terminated then try again. If it still fails, a git process may have crashed in this repository earlier: remove the file manually to continue.

Than first run this command

rm -f ./.git/index.lock

than run

git add -A

git commit -m “Team and car model”

**if changes not reflected on github than run this command**

git push origin main