Daha önceki videolarda kullanıcı yönetimi ile işlemler yapmış tema dosyalarını oluşturmuştuk. Django’nın kullanıcı yönetimi tema dosyaları templates/registration/ klasöründe bulunduğunu söylemiştik. Bu temaları daha önceden yaptığımız için tekrardan yapmamıza gerek yoktur.

Settings.py de login login redirect url değiştirmemiz ve temalardaki extend edilen dosyayı değiştirmemiz yeterli olacaktır.Kullanıcı giriş yaptığında tekrardan login sayfasına yönlensin. Role sayfasını oluşturduğumuzda oraya yönlendireceğiz.

{% extends 'project/layout/base.html' %}

LOGIN\_REDIRECT\_URL = '/accounts/login/'

Bu sayede kullanıcı giriş ve çıkışı, şifre değiştirme, şifre sıfırlama şu anda hazır ve çalışmaktadır.

Daha önce learning kısmında kullanmış olduğumuz tema sisteminin aynısını burada da kullanalım. Hatırlarsanız layout klasöründe base, header ve footer html kullanmıştık. Content bloğunda da ilgili view içeriğini getirmiştik. Bu sefer header ve footer bilgilerini de base içerisinde kullanalım.

base.html hazırlamış olduk. base içerisinde 3 adet bloğumuz bununmaktadır.

{% block title %}

Bu blok içerisinde yer alan bilgiler sayfanın title bilgisini oluşturmaktadır

{% endblock %}

{% block top %}

Bu blok içerisinde yer alan bilgiler sayfanın üst bölümünde yer alan kısmı oluşturmaktadır.

{% endblock %}

{% block content %}

Bu blok arasında yer alan bilgiler sayfanın content kısmını oluşturmaktadır. Her view tarafından oluşturulan çıktılar bu bölümde görüntülenecektir.

{% endblock %}

Navbarımız halen learning için oluşturulan navbarı içermektedir. Bu navbarı mevcut sistemimiz için güncellememiz gerekiyor. Giriş yapılan kullanıcı tipine göre de farklı alanlar içermeli. Ziyaretçiler, bayiler ve distribitörlere farklı navbar gözükmesi gerekmektedir. İlk etapta her kullanıcı yönetimi için kullanacağımız menüleri ekleyelim. Daha sonra viewleri oluşturdukça farklılaştırırız.

{% load static %}  
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>  
 {% block title %}  
 {% endblock %}  
 </title>  
 <link href="{% static 'css/bootstrap.min.css' %}" rel="stylesheet">  
 <link href="{% static 'css/jumbotron.css' %}" rel="stylesheet">  
</head>  
<body>  
  
  
<nav class="navbar navbar-expand-md navbar-dark fixed-top bg-dark">  
 <a class="navbar-brand" href="#">STOCKS PROJECT</a>  
  
 <div class="collapse navbar-collapse" id="navbarsExampleDefault">  
 <ul class="navbar-nav mr-auto">

{% if not user.is\_authenticated %}  
  
 <li class="nav-item {% if request.resolver\_match.url\_name == 'login' %}active{% endif %}">  
 <a class="nav-link" href="{% url 'login' %}">Üye Girişi</a>  
 </li>  
 {% else %}

<li class="nav-item dropdown">  
 <a class="nav-link dropdown-toggle" href="#" id="dropdown01" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">Üye İşlemleri</a>  
 <div class="dropdown-menu" aria-labelledby="dropdown01">  
 <a class="dropdown-item" href="{% url 'password\_change' %}">Şifre Değiştir</a>  
 <a class="dropdown-item" href="{% url 'logout' %}">Çıkış</a>  
 </div>  
 </li>  
 {% endif %}  
 </ul>  
 <form class="form-inline my-2 my-lg-0">  
 <input class="form-control mr-sm-2" type="text" placeholder="arama kelimesi..." aria-label="Search">  
 <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Ara</button>  
 </form>  
 </div>  
</nav>  
  
<main role="main">  
 {% block top %}  
 {% endblock %}  
  
 <div class="container">  
 {% block content %}  
 {% endblock %}  
 <hr>  
  
 </div>   
</main>  
  
<footer class="container">  
 <p>**&copy;** Created for Django Course 2019</p>  
</footer>  
  
<script src="{% static 'js/jquery-slim.min.js' %}"></script>  
<script src="{% static 'js/bootstrap.min.js' %}"></script>  
</body>  
</html>

Üye girişi yapıldığında role sayfamıza yönlendireceğimizi söylemiştik. Şimdi role sayfamızı hazırlamaya başlayalım. Projeyi oluşturmaya başladığımızda distribütör ve bayilerin yetkililerinin olduğunu ve bayi ve distribütör için personel eklenebileneceğini söylemiştik.

Bayi modelini hatırlarsak eğer kullanıcı author alanında ise Bayinin yöneticisi konumunda olacaktır. Bayiler ile Kullanıcı modeli arasında da ilişki kurmuştuk eğer ilişki tablosunda varsa bayi personeli olduğunu anlarız. Distribütörler içinde aynı durum geçerlidir.

Yani kullanıcı giriş yaptığında hangi bayi ve distribütörlerde yetkisinin olduğunu ve yetki seviyesini tespit etmemiz gerekli.

* Distribütör yöneticisi olabilir
* Bayi yöneticisi olabilir
* Distribütör personeli olabilir
* Bayi personeli olabilir
* Birden fazla bayi veya distribütörde rolü olabilir

project/views/ klasörüne role.py oluşturalım ve yetkileri tespit edecek kodumuzu yazalım.

from django.views import View  
from django.shortcuts import render  
from project.models import Dealer**,** Distributor  
from django.contrib.auth.mixins import LoginRequiredMixin  
from django.db.models import Q  
  
  
class RoleView(LoginRequiredMixin**,** View):  
  
 def get(self**,** request):  
 dealers = Dealer.objects.filter(Q(author=request.user) | Q(staff=request.user))  
 distributors = Distributor.objects.filter(Q(author=request.user) | Q(staff=request.user))  
  
 context = {  
 'dealers': dealers**,** 'distributors': distributors**,** }  
  
 return render(request=request**,** template\_name='project/accounts/role.html'**,** context=context)

\_\_init\_\_.py de tanıtalım.

from .role import RoleView

template dosyamızda yetkileri listeleyelim.