URLS

from django.urls import path  
from . import views  
  
urlpatterns = [  
  
 path('dealer/create/'**,** views.DealerCreateView.as\_view()**,** name='create-dealer')**,** path('dealer/edit/<int:pk>/'**,** views.DealerUpdateView.as\_view()**,** name='edit-dealer')**,** path('dealer/delete/<int:pk>/'**,** views.DealerDeleteView.as\_view()**,** name='delete-dealer')**,** path('dealer/list/'**,** views.DealerListView.as\_view()**,** name='list-dealer')**,** path('product/create/'**,** views.ProductCreateView.as\_view()**,** name='create-product')**,** path('product/edit/<int:pk>/'**,** views.ProductUpdateView.as\_view()**,** name='edit-product')**,** path('product/delete/<int:pk>/'**,** views.ProductDeleteView.as\_view()**,** name='delete-product')**,** path('product/list/'**,** views.ProductListView.as\_view()**,** name='list-product')**,**]

MIXINS

class PermissionDeniedRedirectMixin:  
 def handle\_no\_permission(self):  
 if self.raise\_exception:  
 raise PermissionDenied(self.get\_permission\_denied\_message())  
 return redirect('role-list')

class CheckProductDistributorMixin(PermissionDeniedRedirectMixin**,** UserPassesTestMixin):  
  
 def test\_func(self):  
 role\_id = self.request.session.get('role\_id'**,** False)  
 role\_page = self.request.session.get('role\_page'**,** False)  
  
 if not role\_id or role\_page != 'distributor':  
 return False  
  
 instance = self.get\_object()  
  
 if not instance.distributor\_id != role\_id:  
 return False  
  
 return True

INIT

from .create import ProductCreateView  
from .edit import ProductUpdateView  
from .delete import ProductDeleteView  
from .list import ProductListView

ÜRÜN LİSTELE

from django.views.generic import ListView  
from django.contrib.auth.mixins import LoginRequiredMixin**,** PermissionRequiredMixin  
from project.models import Product  
from stocks.mixins import CheckDistributorMixin  
  
  
class ProductListView(CheckDistributorMixin**,** PermissionRequiredMixin**,** LoginRequiredMixin**,** ListView):  
 model = Product  
 template\_name = "project/product/list.html"  
 context\_object\_name = 'product\_list'  
 ordering = ['-created']  
 permission\_required = ('project.manage\_product'**,**)  
  
 def get\_queryset(self):  
 distributor\_id = self.request.session.get('role\_id')  
 return self.model.objects.filter(distributor\_\_id=distributor\_id)

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

{% block top %}  
 <div class="jumbotron"><h4>Ürün Listele</h4></div>  
 {% if messages %}  
 {% for message in messages %}  
 <div class="alert alert-danger" role="alert">  
 {{ message }}  
 </div>  
 {% endfor %}  
 {% endif %}  
{% endblock %}  
  
{% block content %}  
 <div style="margin-bottom: **15**px**;**text-align: right"><a href="{% url 'create-product' %}" class="btn btn-primary">Ürün Ekle</a></div>  
  
 <table class="table table-striped">  
 <thead>  
 <tr>  
 <th scope="col">#</th>  
 <th scope="col">Ürün Adı</th>  
 <th scope="col">Marka</th>  
 <th scope="col">Fiyat</th>  
 <th scope="col">Stok Adedi</th>  
 <th scope="col"></th>  
 </tr>  
 </thead>  
 <tbody>  
 {% for product in product\_list %}  
 <tr>  
 <th scope="row">{{ forloop.counter }}</th>  
 <td>{{ product.name }}</td>  
 <td>{{ product.brand }}</td>  
 <td>{{ product.price }}</td>  
 <td>{{ product.stock\_count }}</td>  
 <td style="text-align: right">  
 <a href="{% url 'delete-product' product.pk %}" class="btn btn-danger btn-sm">Sil</a>  
 <a href="{% url 'edit-product' product.pk %}" class="btn btn-warning btn-sm">Düzenle</a>  
 </td>  
 </tr>  
 {% endfor %}  
 </tbody>  
 </table>  
{% endblock %}

ÜRÜN EKLE

from django.views.generic import CreateView  
from django.contrib.auth.mixins import LoginRequiredMixin**,** PermissionRequiredMixin**,** UserPassesTestMixin  
from project.models import Product  
from django.urls import reverse\_lazy  
from django.contrib.messages.views import SuccessMessageMixin  
from stocks.mixins import CheckDistributorMixin  
  
  
class ProductCreateView(CheckDistributorMixin**,** PermissionRequiredMixin**,** SuccessMessageMixin**,** LoginRequiredMixin**,** CreateView):  
 model = Product  
 fields = ('name'**,** 'content'**,** 'brand'**,** 'price'**,** 'stock\_count'**,** 'active')  
 template\_name = "project/product/create.html"  
 success\_url = reverse\_lazy('list-product')  
 success\_message = "%(name)s başarıyla oluşturuldu."  
 permission\_required = ('project.manage\_product'**,**)  
  
 def form\_valid(self**,** form):  
 form.instance.distributor\_id = self.request.session.get('role\_id')  
 return super(ProductCreateView**,** self).form\_valid(form)

{% extends 'project/layout/base.html' %}  
  
{% block top %}  
 <div class="jumbotron"><h4>Ürün Ekle</h4></div>  
{% endblock %}  
  
{% block content %}  
 <form method="post">  
 {% csrf\_token %}  
 {% for field in form %}  
 <p>  
 {{ field.label\_tag }}<br>  
 {{ field }}  
  
 {% for error in field.errors %}  
 <p style="color: red">{{ error }}</p>  
 {% endfor %}  
 </p>  
 {% endfor %}  
 <button type="submit">Ekle</button>  
 </form>  
{% endblock %}

ÜRÜN DÜZENLE

from django.views.generic import UpdateView  
from django.contrib.auth.mixins import LoginRequiredMixin**,** PermissionRequiredMixin**,** UserPassesTestMixin  
from project.models import Product  
from django.urls import reverse\_lazy  
from django.contrib.messages.views import SuccessMessageMixin  
from stocks.mixins import CheckProductDistributorMixin  
  
  
class ProductUpdateView(CheckProductDistributorMixin**,** SuccessMessageMixin**,** PermissionRequiredMixin**,** LoginRequiredMixin**,** UpdateView):  
 model = Product  
 fields = ('name'**,** 'content'**,** 'brand'**,** 'price'**,** 'stock\_count'**,** 'active')  
 template\_name = "project/product/edit.html"  
 success\_url = reverse\_lazy('list-product')  
 success\_message = "%(name)s başarıyla düzenlendi."  
 permission\_required = ('project.manage\_product'**,**)

{% extends 'project/layout/base.html' %}  
  
{% block top %}  
 <div class="jumbotron"><h4>Ürün Düzenle</h4></div>  
{% endblock %}  
  
{% block content %}  
 <form method="post">  
 {% csrf\_token %}  
 {% for field in form %}  
 <p>  
 {{ field.label\_tag }}<br>  
 {{ field }}  
  
 {% for error in field.errors %}  
 <p style="color: red">{{ error }}</p>  
 {% endfor %}  
 </p>  
 {% endfor %}  
 <button type="submit">Düzenle</button>  
 </form>  
{% endblock %}

ÜRÜN SİL

from django.views.generic import DeleteView  
from django.contrib.auth.mixins import LoginRequiredMixin**,** PermissionRequiredMixin  
from project.models import Product  
from django.urls import reverse\_lazy  
from django.contrib.messages.views import SuccessMessageMixin  
from stocks.mixins import CheckProductDistributorMixin  
  
  
class ProductDeleteView(CheckProductDistributorMixin**,** SuccessMessageMixin**,** PermissionRequiredMixin**,** LoginRequiredMixin**,** DeleteView):  
 model = Product  
 template\_name = "project/product/delete.html"  
 success\_url = reverse\_lazy('list-product')  
 success\_message = "%(name)s başarıyla silindi."  
 permission\_required = ('project.manage\_product'**,**)

{% extends 'project/layout/base.html' %}  
  
{% block top %}  
 <div class="jumbotron"><h4>Ürün Sil</h4></div>  
{% endblock %}  
  
{% block content %}  
<form method="post">  
 {% csrf\_token %}  
 Silmek istediğinize emin misiniz "{{ object }}" ?  
 <input type="submit" value="Sil" />  
</form>  
{% endblock %}

PAGINATION

from django.views.generic import ListView  
from django.contrib.auth.mixins import LoginRequiredMixin**,** PermissionRequiredMixin  
from project.models import Product  
from stocks.mixins import CheckDistributorMixin  
  
  
class ProductListView(CheckDistributorMixin**,** PermissionRequiredMixin**,** LoginRequiredMixin**,** ListView):  
 . . .  
 paginate\_by = **1** . . .

{% block content %}  
 <div style="margin-bottom: **15**px**;**text-align: right"><a href="{% url 'create-product' %}" class="btn btn-primary">Ürün Ekle</a></div>  
  
 <table class="table table-striped">  
 <thead>  
 <tr>  
 <th scope="col">#</th>  
 <th scope="col">Ürün Adı</th>  
 <th scope="col">Marka</th>  
 <th scope="col">Fiyat</th>  
 <th scope="col">Stok Adedi</th>  
 <th scope="col"></th>  
 </tr>  
 </thead>  
 <tbody>  
 {% for product in product\_list %}  
 <tr>  
 <th scope="row">{{ page\_obj.start\_index|add:forloop.counter0 }}</th>  
 <td>{{ product.name }}</td>  
 <td>{{ product.brand }}</td>  
 <td>{{ product.price }}</td>  
 <td>{{ product.stock\_count }}</td>  
 <td style="text-align: right">  
 <a href="{% url 'delete-product' product.pk %}" class="btn btn-danger btn-sm">Sil</a>  
 <a href="{% url 'edit-product' product.pk %}" class="btn btn-warning btn-sm">Düzenle</a>  
 </td>  
 </tr>  
 {% endfor %}  
 </tbody>  
 </table>  
 {% if is\_paginated %}  
 <div class="pagination">  
 <span class="page-links">  
 {% if page\_obj.has\_previous %}  
 <a href="?page={{ page\_obj.previous\_page\_number }}">Önceki</a>  
 {% endif %}  
 <span class="page-current">  
 Toplam {{ page\_obj.paginator.num\_pages }} sayfadan {{ page\_obj.number }} sayfa.  
 </span>  
 {% if page\_obj.has\_next %}  
 <a href="?page={{ page\_obj.next\_page\_number }}">Sonraki</a>  
 {% endif %}  
 </span>  
 </div>  
 {% endif %}  
{% endblock %}

class Product(SlugMixin):  
 . . .  
 class Meta:  
 . . .

ordering = ['-created']  
 . . .