Московский государственный технический университет им. Н.Э. Баумана.

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»

Отчет по рубежному контролю № 2

Вариант 6

Выполнила:
студентка группы ИУ5-54

Зонова А.В.

Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.

Подпись и дата:

Описание задания:

1. Создайте проект Python Django с использованием стандартных средств Django.
2. Создайте модель Django ORM, содержащую две сущности, связанные отношением один-ко-многим в соответствии с Вашим вариантом из условий рубежного контроля №1.
3. С использованием стандартного механизма Django сгенерируйте по модели макет веб-приложения, позволяющий добавлять, редактировать и удалять данные.
4. Создайте представление и шаблон, формирующий отчет, который содержит соединение данных из двух таблиц.

№ варианта	Класс 1	Класс 2
6	Здание	Улица

Ход выполнения работы:

Создание моделей

models.py:

```
from django.db import models
 class
Building(models.Model):
    building_name    =    models.CharField(max_length=50,    verbose_name="ФИО
владельца")
    floors = models.DecimalField(max_digits=8, decimal_places=0,
verbose_name="Кол-во этажей")

Street(models.Model):
        street_name  =  models.CharField(max_length=50,  verbose_name="Название
улицы")
        street_id = models.DecimalField(max_digits=10,decimal_places=0,
verbose_name="ID улицы")
```

Создание сериализаторов

serializers.py:

```
from Buildings.models import Building
from Streets.models import Street
from rest_framework import serializers   class
StreetSerializer(serializers.modelSerializer):
class Meta:
        # Модель, которую мы сериализуем
model = Street
        # Поля, которые мы сериализуем
```

```
        fields = ["street_name", "street_id"]
 class
BuildingSerializer(serializers.ModelSerializer):
class Meta:
        # Модель, которую мы сериализуем
model = Building
        # Поля, которые мы сериализуем
        fields = ["building_name", "floors"]
```

View

view.py:

```
from rest_framework import viewsets
from buildings.serializers import
BuildingSerializer from buildings.serializers
import StreetSerializer from Buildings.models
import Building from Buildings.models import
Street from django.shortcuts import render
 class StreetViewSet(viewsets.ModelViewSet):
    queryset = Street.objects.all().order_by('street_name')
serializer_class = StreetSerializer  # Сериализатор для модели class
BuildingViewSet(viewsets.ModelViewSet):
    queryset = Building.objects.all().order_by('street_name')
serializer_class = BuildingSerializer  # Сериализатор для модели def
StreetList(request):
    return render(request, 'streets.html', {'data': {
        'streets': Street.objects.all(),
    }}) def
BuildingList(request):
    return render(request, 'buildings.html', {'data': {
        'buildings': Building.objects.all(),
    }})  def
GetStreet(request, id):
    return render(request, 'street.html', {'data': {
'street': Street.objects.filter(id=id)[0]
    }}) def
GetBuilding(request, id):
    return render(request, 'building.html', {'data': {
'building': Building.objects.filter(id=id)[0],
        'streets': Street.objects.all(),
}})
```

URL

urls.py

```
from django.contrib import admin
from buildings import views
from django.urls import include, path
from rest_framework import routers

router = routers.DefaultRouter()
router.register(r'streets', views.StreetViewSet)
router.register(r'buildings', views.BuildingViewSet)

urlpatterns = [
                    path('', include(router.urls)),
path('',  api-      include('rest_framework.urls',
          auth/     namespace='rest_framework')),
                                        ', admin.site.urls),
 path('admin/
 path('rk/strt                          ', views.StreetList),
                                        ', views.BuildingList),

 path('rk/bld
 path('                                              ', views.GetStreet,
 path('                                             name='street_url'),
      ]rk/strt/<int:id>/                           ', views.GetBuilding,
                                                    name='building_url')
       rk/bld/<int:id>/
```

proc.html

```
<!doctype html>
<html lang="en" class="h-100">
<head>
  <meta charset="utf-8">
    <title>{% block title %}{% endblock %}</title>
</head>
<body style="background-image:
url(https://phonoteka.org/uploads/posts/2021-
03/1616520072_8-p-foni-dlya-prezentatsii-krasivie-
strogie-8.jpg);  background-repeat: no-repeat">
    {% block content %}{% endblock %}
</body>
</html>
```

streets.html

```
{% extends 'base.html' %}

{% block title %}Рубежный контроль{% endblock %}

{% block content %}
    <h2>Список улиц:</h2>
    <h3>
        <ul>
    {% for street in data.streets %}
        <li><a href="{% url 'street_url' street.id
%}">{{street.street_name}}</a></li>
    {% empty %}
        <li>Список пуст</li>
    {% endfor %}
```

```
        </ul>
    </h3>
{% endblock %}
```

street.html

```
{% extends 'base.html' %}

{% block title %}{{ data.street.street_name }}{% endblock %}

{% block content %}
    <div>Название: {{ data.street.street_name }}</div>
        <div>ID: {{ data.street.id }}</div>
{% endblock %}
```

building.html

```
{% extends 'base.html' %}

{% block title %}Рубежный контроль{% endblock %}

{% block content %}
     <h2>Список домов:</h2>
<h3><ul>
    {% for building in data.buildings %}
        <li><a href="{% url 'building_url' building.id
%}">{{building.building_name}}</a></li>
    {% empty %}
        <li>Список пуст</li>
    {% endfor %}
</ul>
</h3> {% endblock %}
```

main.html

```
{% extends 'base.html' %}


{% block title %}{{ data.building.building_name }}{% endblock
%}
{% block content %}
    <div>Название: {{ data.building.building_name }}</div>
     {% for street in data.streets %}
         {% if  data.building.street_id == street.id %}
        <div>Улица: {{ street.street_name }}</div>
         {% endif %}
    {% endfor %}
        <div>Этажи: {{ data.building.floors }}</div> {% endblock %}
```