

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

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет приложений»

Домашнее задание (вариант 3).
Разработка прототипа веб-приложения с использованием фреймворка
Django

Выполнил:
студент группы ИУ5-55Б
Родионов Дмитрий

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.

Москва, 2021 г.

Описание задания

На основе результатов лабораторных работ 4 и 5, создайте прототип веб-приложения с использованием фреймворка Django на основе базы данных, реализующий концепцию master/detail. Прототип должен содержать:

1. Две модели, связанные отношением один-ко-многим.
2. Стандартное средство администрирования Django позволяет редактировать данные моделей. Желательно настроить русификацию ввода и редактирования данных.
3. Веб-приложение формирует отчет в виде отдельного view/template, отчет выводит HTML-страницу, содержащую связанные данные из двух моделей.
4. Для верстки шаблонов используется фреймворк Bootstrap, или аналогичный фреймворк по желанию студента.

Текст программы

Settings.py

```
from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/4.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-u*)1wt9g-shw_w!h+_t#lucbsbq#ydgY(fei5nv1o7bx+gs5&i'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'dzapp.apps.DzappConfig'
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
```

```

'django.contrib.sessions.middleware.SessionMiddleware',
'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'DZ.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR / 'templates'],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    ],
]

WSGI_APPLICATION = 'DZ.wsgi.application'

# Database
# https://docs.djangoproject.com/en/4.0/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}

# Password validation
# https://docs.djangoproject.com/en/4.0/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

```

```

# Internationalization
# https://docs.djangoproject.com/en/4.0/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.0/howto/static-files/

STATIC_URL = 'static/'


# Default primary key field type
# https://docs.djangoproject.com/en/4.0/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

```

Urls.py

```

from django.contrib import admin
from django.urls import path
from dzapp import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.index),
    path('otchet/', views.otchet),
    path('create_build/', views.create_build),
    path('create_style/', views.create_style),
    path('otchet/edit_build/<int:id>', views.edit_build),
    path('otchet/edit_style/<int:id>', views.edit_style),
    path('otchet/delete_build/<int:id>', views.delete_build),
    path('otchet/delete_style/<int:id>', views.delete_style),
]

```

Admin.py

```

from django.contrib import admin

from .models import Build, Style
admin.site.register(Build)
admin.site.register(Style)

```

Models.py

```

from django.db import models

class Style(models.Model):
    style_name = models.CharField(max_length=200)
    size = models.CharField(max_length=200)
    def str (self):

```

```

        return self.style_name

class Build(models.Model):
    name = models.CharField(max_length=200)
    date = models.CharField(max_length=200)
    city = models.CharField(max_length=200)
    idstyle = models.ForeignKey(Style, on_delete=models.CASCADE)

```

Views.py

```

from django.shortcuts import render
from django.shortcuts import render
from django.http import HttpResponseRedirect
from django.http import HttpResponseRedirect
from .models import Build, Style

def index(request):
    styles = Style.objects.all()
    builds = Build.objects.all()
    return render(request, "index.html", {"styles": styles, "builds": builds})

def otchet(request):
    styles = Style.objects.all()
    builds = Build.objects.all()
    return render(request, "otchet.html", {"styles": styles, "builds": builds})

def create_build(request):
    if request.method == "POST":
        d = Build()
        d.name = request.POST.get("name")
        d.date = request.POST.get("date")
        d.city = request.POST.get("city")
        d.idstyle = Style.objects.get(name=request.POST.get("idstyle"))
        d.save()
    return HttpResponseRedirect("/otchet/")

def create_style(request):
    if request.method == "POST":
        dl = Style()
        dl.style_name = request.POST.get("style_name")
        dl.size = request.POST.get("size")
        dl.save()
    return HttpResponseRedirect("/otchet/")

def edit_build(request, id):
    try:
        build = Build.objects.get(id=id)
        if request.method == "POST":
            build.name = request.POST.get("name")
            build.date = request.POST.get("date")
            build.city = request.POST.get("city")
            build.idstyle = Style.objects.get(name=request.POST.get("idstyle"))
            build.save()
            return HttpResponseRedirect("/")
        else:
            return render(request, "edit_build.html", {"build": build})

```

```

except Build.DoesNotExist:
    return HttpResponseRedirect("<h2>Build not found</h2>")

def edit_style(request, id):
    try:
        cur_type = Style.objects.get(id=id)

        if request.method == "POST":
            cur_type.style_name = request.POST.get("style_name")
            cur_type.size = request.POST.get("size")
            cur_type.save()
            return HttpResponseRedirect("/")
        else:
            return render(request, "edit_style.html", {"style": cur_type})
    except Style.DoesNotExist:
        return HttpResponseRedirect("<h2>Style not found</h2>")

def delete_build(request, id):
    try:
        build = Build.objects.get(id=id)
        build.delete()
        return HttpResponseRedirect("/otchet/")
    except build.DoesNotExist:
        return HttpResponseRedirect("<h2>Build not found</h2>")

def delete_style(request, id):
    try:
        cur_type = Style.objects.get(id=id)
        cur_type.delete()
        return HttpResponseRedirect("/otchet/")
    except cur_type.DoesNotExist:
        return HttpResponseRedirect("<h2>Style not found</h2>")

```

edit_build.html

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>edit_build </title>
</head>
<body class="container">
    <form method="POST">
        {% csrf_token %}
        <p>
            <label>Введите название здания</label><br>
            <input type="text" name="name" value="{{build.name}}" />
        </p>
        <p>
            <label>Введите дату постройки</label><br>
            <input type="text" name="date" value="{{build.date}}" />
        </p>
        <p>
            <label>Введите город</label><br>
            <input type="text" name="city" value="{{build.city}}" />
        </p>
        <p>
            <label>Введите стиль</label><br>

```

```

        <input type="text" name="idstyle"
value="{{build.idstyle.style_name}}" />
    </p>
    <input type="submit" value="Сохранить" >
</form>
</body>
</html>

```

edit_style.html

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>edit type</title>
</head>
<body class="container">
    <form method="POST">
        {% csrf_token %}
        <p>
            <label>Введите название стиля</label><br>
            <input type="text" name="style_name" value="{{style.style_name}}" />
        </p>
        <p>
            <label>Введите описание стиля</label><br>
            <input type="text" name="size" value="{{style.size}}" />
        </p>
        <input type="submit" value="Сохранить" >
    </form>
</body>
</html>

```

index. html

```

    <div class="col-md-6">
        <div class="h-100 p-5 bg-dark text-white rounded-3">
            <form method="POST" action="create_style/">
                {% csrf_token %}
                <h2>Добавить стиль</h2>
                <p>
                    <label>Название </label><br>
                    <input type="text" name="style_name" />
                </p>
                <p>
                    <label>Описание </label><br>
                    <input type="text" name="size" />
                </p>
                <input type="submit" value="Сохранить" >
            </form>
        </div>
    </div>
    <div class="col-md-6">
        <div class="h-100 p-5 bg-light text-dark rounded-3">
            <form method="POST" action="create_build/">
                {% csrf_token %}
                <h2>Добавить здание</h2>
                <p>
                    <label>Название </label><br>
                    <input type="text" name="name" />

```

```

        </p>
        <p>
            <label>Дата постройки </label><br>
            <input type="text" name="date" />
        </p>
        <p>
            <label>Город </label><br>
            <input type="text" name="city"/>
        </p>
        <p>
            <label>Стиль </label><br>
            <input type="text" name="idstyle"/>
        </p>
        <input type="submit" value="Сохранить" >
    </form>
</div>
</div>
</div>
<div>
    <p>

    </p>
</div>
</body>
</html>

```

otchet. html

```

<html>
    <head>
        <div class="container">
            <header class="d-flex justify-content-center py-3">
                <ul class="nav nav-pills">
                    <li class="nav-item"><a href="http://127.0.0.1:8000/"
class="nav-link " aria-current="page">Главная</a></li>
                    <li class="nav-item"><a href="http://127.0.0.1:8000/otchet/"
class="nav-link active">Отчёт</a></li>
                </ul>
            </header>
        </div>
        <meta charset="utf-8" />
        <title>Дз РИП</title>
        <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
crossorigin="anonymous">
        <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
integrity="sha384-
ChfqquxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
crossorigin="anonymous"></script>
    </head>
    <body>
        <div class="row align-items-md-stretch justify-content-center">
            <div class="col-md-6">
                <div class="h-100 p-5 bg-light text-dark rounded-3">
                    {% for style in styles %}
                    <h2>Стиль</h2>
                    <ul>

```



```

        <li>Название: {{ style.style_name }}</li>
        <li>Описание: {{ style.size }}</li>
        <a href="edit_style/{{style.id}}"> Изменить</a> | <a
href="delete_style/{{style.id}}">Удалить</a>
        {% if builds.count > 0 %}
        <h2>Здания </h2>
        {% for build in builds %}
        {% if build.idstyle == style%}
        <ul>
            <li>Название: {{ build.name }}</li>
            <li>Дата постройки: {{ build.date }}</li>
            <li>Город: {{ build.city }}</li>
            <li>Стиль: {{ build.idstyle.style_name }}</li>
            <a href="edit_build/{{build.id}}"> Изменить</a>
| <a href="delete_build/{{build.id}}">Удалить</a>
        </ul>
        {%endif%}
        {% endfor %}
        {% endif %}
        <br>
    </ul>
    {% endfor %}
</div>
</div>
</div>
</body>
</html>

```

Экранные формы с примерами выполнения программы

127.0.0.1:8000

[AliExpress](#)
[Бронирование оте...](#)
[Lamoda](#)
[Рамблер](#)
[Facebook](#)
[Авиабилеты](#)
[Яндекс](#)

Главная

Отчёт

Добавить стиль

Название

Неоготика

Описание

популярный стиль 19 ве

Сохранить

Добавить здание

Название

Биг Бен

Дата постройки

1883

Город

Лондон

Стиль

Эклектика

Сохранить

Стиль

- Название: Неоготика
- Описание: популярный стиль 19 века

[Изменить](#) | [Удалить](#)

Стиль

- Название: Готика
- Описание: появился по Франции в 13 веке

[Изменить](#) | [Удалить](#)