



**Министерство науки и высшего образования Российской
Федерации**
Федеральное государственное бюджетное образовательное учреждение высшего образования
**«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

**Отчет по домашнему заданию
Курса «Разработка интернет-приложений»**

Студент: Яровенко М.В.

Группа: ИУ5Ц-72Б

Преподаватель: Гапанюк Ю.Е.

2021 г.

Оглавление

Задание	3
Текст программы	3
Результат выполнения программы	17

Задание

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:

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

Текст программы

Файл models.py

```
from django.db import models
```

```
class Staff(models.Model):
```

```
    id = models.AutoField('id', primary_key=True)
```

```
    name = models.CharField(max_length=30)
```

```
    family = models.CharField(max_length=30)
```

```
    status = models.CharField(max_length=30)
```

```
    def __str__(self):
```

```
        return f'{self.name}'
```

```
class Meta:
```

```
    managed = False
```

```
    db_table = 'staff'
```

```
class Restaurants(models.Model):
    id = models.AutoField('id', primary_key=True)
    name = models.CharField(max_length=30)
```

```
class Meta:
    managed = False
    db_table = 'restaurants'
```

Файл views.py

```
from django.shortcuts import render
from django.views.generic.edit import CreateView, UpdateView,
DeleteView
from django import forms
from .models import *
from datetime import datetime
```

```
def index(request):
    return render(request, 'index.html')
```

```
def report(request):
    restaurants = Restaurants.objects.all()
    staff = Staff.objects.all()
    params = {'restaurants': restaurants, 'staff': staff, 'date':
datetime.now().strftime("%Y-%m-%d %H:%M:%S")}
    return render(request, 'report.html', params)
```

```

def staff_list(request):
    staff = Staff.objects.all().values()
    params = {'entity': 'Staff', 'objects': staff}
    return render(request, 'list.html', params)


def restaurants_list(request):
    restaurants = Restaurants.objects.all().values()
    params = {'entity': 'Restaurants', 'objects': restaurants}
    return render(request, 'list.html', params)


class StaffCreate(CreateView):
    model = Staff
    fields = ['name', 'family', 'status']
    success_url = '/staff'
    template_name = 'staff_form.html'


class StaffUpdate(UpdateView):
    model = Staff
    fields = ['name', 'family', 'status']
    pk_url_kwarg = 'id'
    success_url = '/staff'
    template_name = 'staff_form.html'


class StaffDelete>DeleteView):
    model = Staff

```

```
pk_url_kwarg = 'id'
success_url = '/staff'
template_name = 'staff_delete_form.html'
```

```
class RestaurantsCreate(CreateView):
    model = Restaurants
    fields = ['name']
    success_url = '/restaurants'
    template_name = 'restaurants_form.html'
```

```
class RestaurantsUpdate(UpdateView):
    model = Restaurants
    fields = ['name']
    pk_url_kwarg = 'id'
    success_url = '/restaurants'
    template_name = 'restaurants_form.html'
```

```
class RestaurantsDelete>DeleteView):
    model = Restaurants
    pk_url_kwarg = 'id'
    success_url = '/restaurants'
    template_name = 'restaurants_delete_form.html'
```

Файл urls.py

```
from django.urls import path
from . import views
```

```
urlpatterns = [
```

```

    path("", views.index),
    path('staff', views.staff_list),
    path('staff/create', views.StaffCreate.as_view()),
    path('staff/<int:id>/update',          views.StaffUpdate.as_view(),
name='staff_update'),
    path('staff/<int:id>/delete',          views.StaffDelete.as_view(),
name='staff_delete'),
    path('restaurants', views.restaurants_list),
    path('restaurants/create', views.RestaurantsCreate.as_view()),
    path('restaurants/<int:id>/update',    views.RestaurantsUpdate.as_view(),
name='restaurants_update'),
    path('restaurants/<int:id>/delete',    views.RestaurantsDelete.as_view(),
name='restaurants_delete'),
    path('report', views.report)
]

```

Φαῖλ stings.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-2psfze#(y@m3t2ka$s6chet3_p39ad*31f&-n8y1w@1)%2ymkp'

```

```

# 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',  
    'rk2_app.apps.Rk2AppConfig',  
]
```

```
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 = 'rk2.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 = 'rk2.wsgi.application'
```

```
# Database
```

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

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'rk2',
        'USER': 'dbuser',
        'PASSWORD': 'admin',
        'HOST': 'localhost',
        'PORT': 3306,
```

```

        'OPTIONS': {'charset': 'utf8'},
        'TEST_CHARSET': 'utf8',
    }
}

# 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'
```

```
Файл base.html
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <meta charset='utf-8'>
```

```
    <title>
```

```
        {% block title %} {% endblock %}
```

```
    </title>
```

```
    <style>
```

```

a, a:visited{
    color: black
}

p {
    font-size: 1.2rem;
    font-family: sans-serif;
}

</style>
</head>
<body>
<div>
    <a style='margin: 5px;' href='/'>Главная</a>
    <a style='margin: 5px;' href='/restaurants'>Рестораны Джанго</a>
    <a style='margin: 5px;' href='/staff'>Сотрудники</a>
    <a style='margin: 5px;' href='/report'>Отчёт</a>
</div>
{% block content %} {% endblock %}
</body>
</html>

```

Файл index.html

```

{% extends 'base.html' %}

{% block title %}
Главная страница
{% endblock %}

{% block content %}
<h2>Домашнее задание по РИП. Вариант 3</h2>

```

Студент Яровенко Максим, группа ИУ5Ц-72Б

```
{% endblock % }
```

Файл restaurants_delete_from.html

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

```
{% block content % }
```

<h2>Удаление записи</h2>

<p>Подтвердите удаление ресторана <i>{{ object.name }}</i></p>

<form action="" method="POST">

```
{% csrf_token % }
```

<input type="submit" value="Удалить" />

</form>

```
{% endblock % }
```

Файл restaurants_from.html

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

```
{% block title % }
```

Изменение

```
{% endblock % }
```

```
{% block content % }
```

<h2>Параметры записи</h2>

<form action="" method="post">

```
{% csrf_token % }
```

```

<table>
  {{ form.as_table }}
</table>

<br>

<input type="submit" value="Сохранить" />
</form>

```

```
{% endblock %}
```

Файл list.html

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

```
{% block title %}
```

```
{{ entity }}
```

```
{% endblock %}
```

```
{% block content %}
```

```
<h2>Сущность {{ entity }}</h2>
```

```
<p>Список:</p>
```

```
<ul>
```

```
  {% for elem in objects %}
```

```
    <hr>
```

```
    <li>
```

```
      {% if entity == 'Staff' %}
```

```
      <p>{{ elem.name }} {{ elem.family }}, {{ elem.status }}</p>
```

```
      <a href='{% url "staff_update" elem.id %}'>Изменить</a> <br>
```

```
      <a href='{% url "staff_delete" elem.id %}'>Удалить</a>
```

```
      {% endif %}
```

```

    { % if entity == 'Restaurants' % }
    <p> { { elem.name } } </p><br>
    <a href='{ % url "restaurants_update" elem.id % }'>Изменить</a>
<br>
    <a href='{ % url "restaurants_delete" elem.id % }'>Удалить</a>
    { % endif % }
</li>
{ % endfor % }
<hr>
</ul>

```

```

{ % if entity == 'Staff' % }
<a href='/staff/create'>Добавить запись</a>
{ % endif % }

```

```

{ % if entity == 'Restaurants' % }
<a href='/restaurants/create'>Добавить запись</a>
{ % endif % }
{ % endblock % }

```

Файл *staff_delete_from.html*

```

{ % extends "base.html" % }

```

```

{ % block content % }

```

```

<h2>Удаление записи</h2>

```

```

<p>Подтвердите удаления сотрудника { { object.name } } </p>

```

```

<form action="" method="POST">

```

```
{% csrf_token %}
<input type="submit" value="Удалить" />
</form>
{% endblock %}
```

Файл staff_form.html

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

```
{% block title %}
```

Изменение

```
{% endblock %}
```

```
{% block content %}
```

```
<h2>Параметры записи</h2>
```

```
<form action="" method="post">
```

```
    {% csrf_token %}
```

```
    <table>
```

```
        {{ form.as_table }}
```

```
    </table>
```

```
    <br>
```

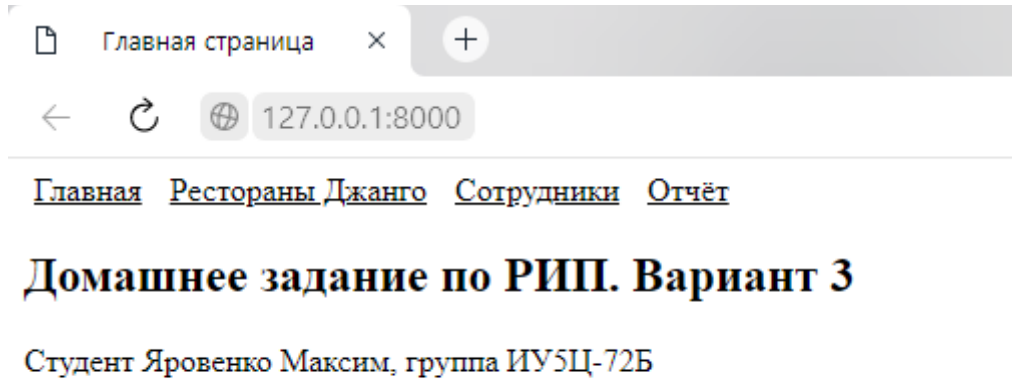
```
    <input type="submit" value="Сохранить" />
```

```
</form>
```

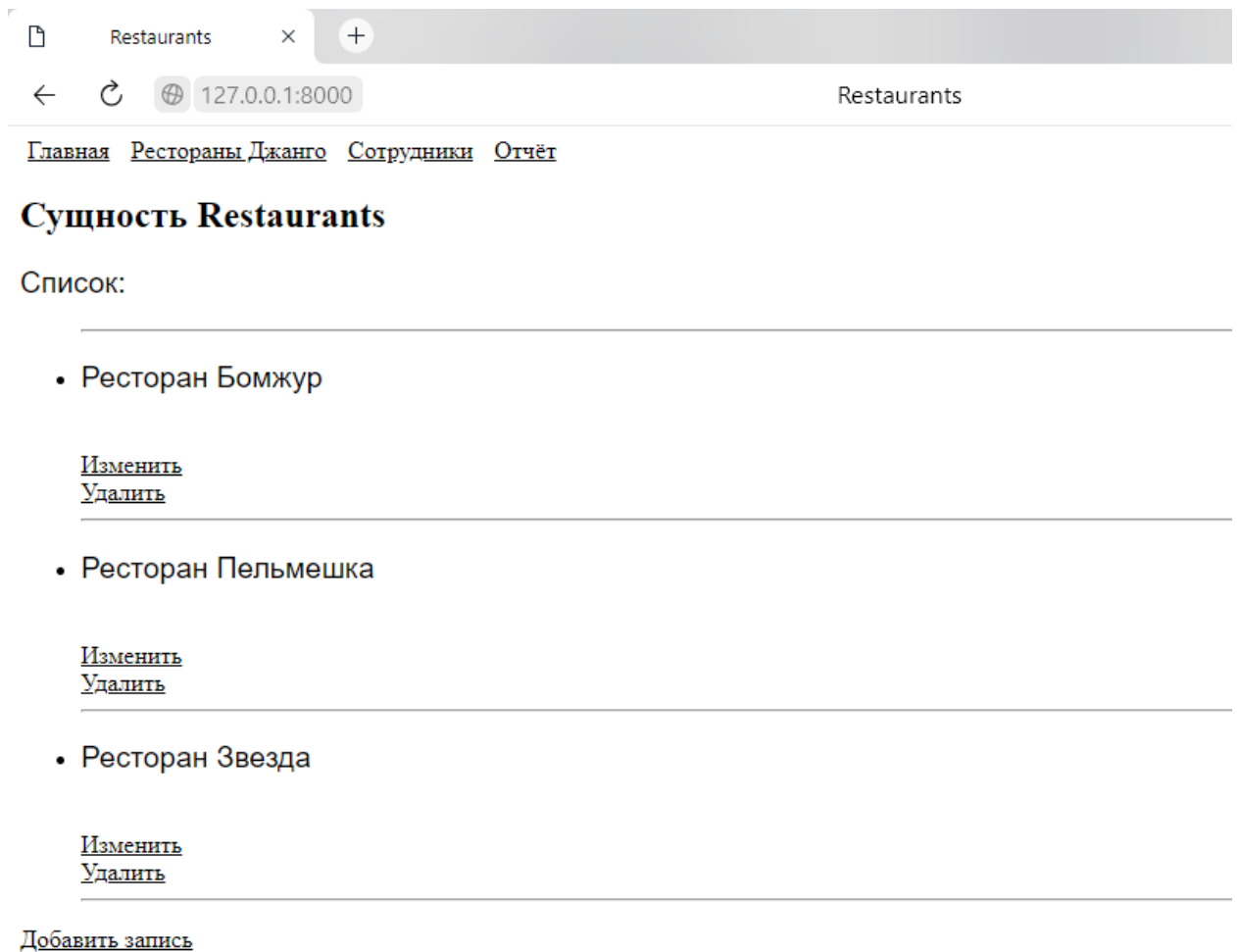
```
{% endblock %}
```


Результат выполнения программы

Главная страница веб-сайта:



Список ресторанов:



Список сотрудников:

[Главная](#) [Рестораны](#) [Джанго](#) [Сотрудники](#) [Отчёт](#)

Сущность Staff

Список:

-
- Илья Ильяновск, Официант

[Изменить](#)

[Удалить](#)

- Андрей Дягтерев, Повар

[Изменить](#)

[Удалить](#)

- Игорь Аморфов, Уборщик

[Изменить](#)

[Удалить](#)

- Виктор Баринов, Повар

[Изменить](#)

[Удалить](#)

Добавление/изменение записи:

Изменение

×

+

←

↺

127.0.0.1:8000

[Главная](#) [Рестораны Django](#) [Сотрудники](#) [Отчёт](#)

Параметры записи

Name:

Family:

Status:

Сохранить

Изменение

×

+

←

↺

127.0.0.1:8000

Изменение

[Главная](#) [Рестораны Django](#) [Сотрудники](#) [Отчёт](#)

Параметры записи

Name:

Сохранить

Удаление записи:

127.0.0.1:8000/restaurant

×

+

←

↺

127.0.0.1:8000

127.0.0.1:8000

[Главная](#) [Рестораны Django](#) [Сотрудники](#) [Отчёт](#)

Удаление записи

Подтвердите удаление ресторана *Ресторан Бомжур*

Удалить

Отчет:



[Главная](#) [Рестораны Django](#) [Сотрудники](#) [Отчёт](#)

Список всех ресторанов и сотрудников:

- Ресторан Бомжур
 - Илья Ильяновск, Официант
 - Андрей Дяттерев, Повар
 - Виктор Баринов, Повар
 - Игорь Аморфов, Уборщик
- Ресторан Пельмешка
 - Александр Невский, Повар
 - Антон Дмитриев, Уборщик
- Ресторан Звезда
 - Влад Тонкий, Официант
 - Игорь Севский, Повар