

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

Факультет «Информатика и системы управления»

**Отчёт по лабораторной работе №5
по курсу «Разработка интернет-приложений»**

Работа с СУБД. Обработка данных с использованием Django ORM.

Выполнил:

**студент группы ИУ5-51Б
Бондаренко И. Г.**

Проверил:

**преподаватель каф. ИУ5
Гапанюк Ю. Е.**

Подпись и дата: 24.12.2021

Подпись и дата:

Москва, 2021 г.

Описание задания.

В этой лабораторной работе Вы познакомитесь с популярной СУБД MySQL, создадите свою базу данных. Также Вам нужно будет дополнить свои классы предметной области, связав их с созданной БД. После этого Вы создадите свои модели с помощью Django ORM, отобразите объекты из БД с помощью этих моделей.

1. Создайте сценарий с подключением к БД и несколькими запросами, примеры рассмотрены в методических указаниях.
2. Реализуйте модели Вашей предметной области из предыдущей ЛР (минимум две модели, т.е. две таблицы).
3. Создайте представления и шаблоны Django для отображения списка данных по каждой из сущностей.

Создание веб-приложения

urls.py:

```
from django.contrib import admin
from django.urls import path
from hotel import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.getHotels),
    path('hotel/<int:id>', views.getHotelById, name='hotel_url'),
]
```

views.py:

```
from django.shortcuts import render
from datetime import date
from hotel.models import Hotel

def getHotels(request):
    return render(request, 'hotels/hotels.html', {'data' : {
        'current_date': date.today(),
        'hotels': Hotel.objects.all()
    }})

def getHotelById(request, id):
    return render(request, 'hotels/hotel.html', {'data' : {
        'current_date': date.today(),
        'hotel': Hotel.objects.filter(id=id)[0]
    }})
```

Создание двух моделей (Страны и города): models.py:

```
from django.db import models

class Hotel(models.Model):
    name = models.CharField(max_length=255, blank=True, null=True)
```

```

url = models.CharField(max_length=255, blank=True, null=True)
stars = models.IntegerField(blank=True, null=True)
country_code = models.IntegerField(blank=True, null=True)

class Meta:
    managed = False
    db_table = 'Hotel'

```

base.html:

```

<!doctype html>
<html lang="en" class="h-100">
<head>
    <meta charset="utf-8">
    <title>{% block title %}{% endblock %}</title>
</head>
<body>
    Hello BMSTU students!
    {% block content %}{% endblock %}
</body>
</html>

```

index.html:

```

{% load static %}
<!doctype html>
<html lang="en" class="h-100">
<head>
    <meta charset="utf-8">
    <title>BMSTU</title>
</head>
<body>
    Hello BMSTU students!
    Today is {{ data.current_date }}
<ul>
    {% for var in data.list %}
        <li>We like {{ var }}</li>
    {% endfor %}
</ul>
</body>
</html>

```

hotels.html:

```

{% extends 'base.html' %}
{% load static %}

{% block title %}Список отелей{% endblock %}

{% block content %}
<ul>
    {% for hotel in data.hotels %}
        <li><a href="{% url 'hotel_url' hotel.id %}">{{hotel.name}}</a></li>
    {% empty %}
        <li>Список пуст</li>
    {% endfor %}

```

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

hotel.html:

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

{% block title %}Отель №{{ data.hotel.id }}{% endblock %}

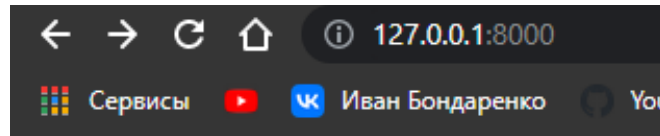
{% block content %}
    <div>Название: {{ data.hotel.name }}</div>
    <div>Ссылка: <a href="{{data.hotel.url}}">{{data.hotel.url}}</a></div>
    <div>Количество звезд: {{ data.hotel.stars }}</div>
    <div>Код страны: {{ data.hotel.country_code }}</div>
{% endblock %}
```

Подключение к БД:

settings.py:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'hotels',
        'USER': 'root',
        'PASSWORD': 'root',
        'HOST': 'localhost',
        'PORT': 3306, # Стандартный порт MySQL
        'OPTIONS': {'charset': 'utf8'},
        'TEST_CHARSET': 'utf8',
    }
}
```

Запуск сервера:



Hello BMSTU students!

- [Космос](#)
- [Хилтон Москва Ленинградская](#)
- [Парк Инн от Рэдиссон Сад](#)
- [Votsalakia Hotel & Apartments](#)
- [Petritis Guesthouse](#)

