



**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)**

**Факультет «Радиотехнический»
Кафедра ИУ5 «Системы обработки информации и управления»**

**Лабораторная работа №5
по дисциплине «Разработка интернет-приложений»**

**Выполнил:
студент группы РТ5-51Б
Плешаков В. И.**

**Проверил:
преподаватель
Балашов А. М.**

2021 г.

Цель лабораторной работы: изучение возможностей СУБД MySQL для разработки веб-приложений. Изучение возможностей обработки данных с использованием Django ORM.

Задание:

1. Создайте сценарий с подключением к БД и несколькими запросами
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.

Скрипт для подключения к БД, добавления одной записи, вывода всех записей и удаления всех записей:

```
import MySQLdb

db = MySQLdb.connect(
    host='localhost',
    user='dbuser',
    passwd='123',
    db='first_db'
)

c = db.cursor()
c.execute("INSERT INTO books (name, dicription) VALUES (%s, %s);", ('Книга', 'Описание'))
db.commit()
c.execute("SELECT * FROM books")
records = c.fetchall()
for row in records:
    print(row)
c.execute("DELETE FROM BOOKS")
db.commit()
c.close()
db.close()
```

Результат выполнения:

```
(1, 'Мастер и Маргарита', 'Крутая книга')
(2, 'Book', 'Description')
(4, 'Книга', 'Описание')

Process finished with exit code 0
```

Подключение БД к проекту:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'phones',
        'USER': 'dbuser',
        'PASSWORD': '123',
        'HOST': 'localhost',
        'PORT': '3306',
        'OPTIONS': {'charset': 'utf8'},
        'TEST_CHARSET': 'utf8',
    }
}
```

models.py:

```
from django.db import models

class Manufacturer(models.Model):
    idmanufacturer = models.AutoField(db_column='idManufacturer', primary_key=True) # Field name made lowercase.
    name = models.CharField(max_length=45)
    country = models.CharField(max_length=45)

    def __str__(self):
        return self.name

    class Meta:
        managed = False
        db_table = 'manufacturer'

class Phone(models.Model):
    idphone = models.AutoField(db_column='idPhone', primary_key=True) # Field name made lowercase.
    model = models.CharField(max_length=45)
    manufactid = models.ForeignKey(Manufacturer, models.DO_NOTHING, db_column='manufactId') # Field name made lowercase
    diag = models.IntegerField()
    capacity = models.IntegerField()

    def __str__(self):
        return self.model

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

views.py:

```
from django.shortcuts import render
from django.views import generic
from phones.models import Phone, Manufacturer

def mainView(request):
    return render(request, template_name='main.html')

class PhonesView(generic.ListView):
    context_object_name = 'phones'
    template_name = 'index.html'

    def get_queryset(self):
        return Phone.objects.all()

class ManufacView(generic.ListView):
    context_object_name = 'manufacs'
    template_name = 'index.html'

    def get_queryset(self):
        return Manufacturer.objects.all()

class PhoneDetail(generic.DetailView):
    template_name = 'detail.html'
    model = Phone

class ManufacDetail(generic.DetailView):
    template_name = 'detail.html'
    model = Manufacturer
```

phones\urls.py:

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

app_name = 'phones'
urlpatterns = [
    path('', views.mainView, name='main'),
    path('phones/', views.PhonesView.as_view(), name='phonesList'),
    path('manufact/', views.ManufacView.as_view(), name='manufacList'),
    path('phones/<int:pk>/', views.PhoneDetail.as_view(), name='phonesDetail'),
    path('manufact/<int:pk>/', views.ManufacDetail.as_view(), name='manufacDetail'),
]
```

main.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
    <a href="{% url 'phones:phonesList' %}"> Перейти к списку телефонов </a><br>
    <a href="{% url 'phones:manufacList' %}"> Перейти к списку производителей </a>
</body>
</html>
```

index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
    {% if phones %}
        {% for phone in phones %}
            <li> <a href="{% url 'phones:phonesDetail' phone.idphone %}"> {{ phone.model }} </a> </li>
        {% endfor %}
    {% else %}
        {% for manufac in manufacs %}
            <li> <a href="{% url 'phones:manufacDetail' manufac.idmanufacturer %}"> {{ manufac.name }} </a></li>
        {% endfor %}
    {% endif %}
    <a href="{% url 'phones:main' %}">На главную</a>
</body>
</html>
```

detail.html:

```
{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
{% if phone %}
    Модель: {{ phone.model }} <br>
    Производитель: {{ phone.manufactid }} <br>
    Диагональ экрана: {{ phone.diag }} <br>
    Емкость аккумулятора: {{ phone.capacity }}<br>
    <a href="{% url 'phones:phonesList' %}">Назад</a>
{% else %}
    Компания: {{ manufacturer.name }} <br>
    Страна: {{ manufacturer.country }} <br>
    <a href="{% url 'phones:manufacList' %}">Назад</a>
{% endif %}
</body>
</html>
```

Пример выполнения программы:



Список телефонов:



Подробная информация об одном из телефонов:



Список производителей:



Подробная информация об одном из производителей:

