

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

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

Курс «Разработка интернет приложений»

Отчет по лабораторной работе №5
«Работа с СУБД. Обработка данных с использованием Django ORM.»

Выполнил:
студент группы ИУ5-55Б
Родионов Д.А.

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.

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 двух таблиц базы данных:

```
USE Buildings;
```

```
CREATE TABLE dbo.City
```

```
(
    ID_city INT IDENTITY(1, 1) CONSTRAINT pk_city PRIMARY KEY,
    City_name NVARCHAR(100),
    Descript NVARCHAR(100),
    ID_building INT
);
```

```
CREATE TABLE dbo.Building
```

```
(
    ID_building INT IDENTITY(1, 1) CONSTRAINT pk_building PRIMARY KEY,
    Building_name NVARCHAR(100),
    Descript NVARCHAR(100),
    Creation_year INT
);
```

```
ALTER TABLE dbo.City
```

```
ADD CONSTRAINT fk_city_build
```

```
FOREIGN KEY (ID_building) REFERENCES City(ID_building)
```

```
INSERT INTO dbo.City SELECT 'London', 'London is the capital of GB!', 1 UNION ALL SELECT 'NYC', 'NYC is the most popular american city', 2
```

```
INSERT INTO dbo.Building SELECT 'Big Ben', 'Neo-Gothic design', 1859 UNION ALL SELECT 'Empire State Building', 'Art Deco style', 1931
```

Скрипт подключения к базе данных, добавления записей в таблицу и выборки данных:

```
import pyodbc

connectionString = ("Driver={SQL Server Native Client 11.0};"
                    "Server=LAPTOP-EUS1P6PQ\\SQLEXPRESS;"
                    "Database=Buildings;"
                    "Trusted_Connection=yes;")

request1 = "SELECT * FROM dbo.City"
request2 = "SELECT * FROM dbo.Building"

connection = pyodbc.connect(connectionString, autocommit=True)
dbCursor = connection.cursor()
dbCursor.execute(request1)
dbCursor.execute(request2)
for row in dbCursor:
    print(f"{row.ID_building} {row.Building_name} {row.Descript}"
          {row.Creation_year}")
connection.commit()
dbCursor.close()
connection.close()
```

Результат:

```
C:\Users\Dmitry\AppData\Local\Programs\Python\Python39\python.exe C:/Users/Dmitry/PycharmProjects/Lab5/connection_script.py
1 Big Ben Neo-Gothic design 1859
2 Empire State Building Art Deco style 1931

Process finished with exit code 0
```

Файл main_app/models.py с созданием классов моделей таблиц:

```
from django.db import models

class Building(models.Model):
    id_building = models.AutoField(db_column='ID_building', primary_key=True) #
    Field name made lowercase.
    building_name = models.CharField(db_column='Building_name', max_length=100,
    blank=True, null=False) # Field name made lowercase.
    descript = models.CharField(db_column='Descript', max_length=1000,
    blank=True, null=True) # Field name made lowercase.
    creation_year = models.IntegerField(db_column='Creation_year', blank=True,
    null=True) # Field name made lowercase.

    class Meta:
        db_table = 'Building'
```

```

class City(models.Model):
    id_city = models.AutoField(db_column='ID_city', primary_key=True) # Field
name made lowercase.
    city_name = models.CharField(db_column='City_name', max_length=100,
blank=True, null=False) # Field name made lowercase.
    descript = models.CharField(db_column='Descript', max_length=100,
blank=True, null=True) # Field name made lowercase.
    id_building = models.ForeignKey('Building', models.DO_NOTHING,
db_column='ID_building', blank=True, null=True) # Field name made lowercase.

    class Meta:
        db_table = 'City'

```

Файл main_app/urls.py:

```

from django.urls import path
from . import views

urlpatterns = [
    path('', views.index),
    path('<str:model_name>/', views.list, name='list')
]

```

Файл main_app/views.py:

```

from django.shortcuts import render
from .models import *
from django.apps import apps

def index(request):
    models = apps.get_app_config('main_app').get_models()
    models_names = [model._meta.db_table for model in models]
    models_dict = {model_id: models_name for model_id, models_name in
zip(range(len(models_names)), models_names)}
    params = {'models_dict': models_dict}
    print(params)
    return render(request, 'index.html', params)

def list(request, model_name):
    models = apps.get_app_config('main_app').get_models()
    model = ''
    for elem in models:
        if elem._meta.db_table == model_name:
            model = elem
    params = { 'model_name': model._meta.db_table,
               'objects': model.objects.values() }
    return render(request, 'list.html', params)

```

Файл base.html:

```

<!doctype html>
<html>
<head>

```

```

    <meta charset="utf-8">
    <title>{% block title %}{% endblock %}</title>
</head>
<body>
    <a href="/">Главная</a>
    {% block content %}{% endblock %}
</body>
</html>

```

Файл index.py:

```

{% extends 'base.html' %}

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

{% block content %}
<h2>Подключенная база данных содержит следующие сущности:</h2>
<ol>
    {% for model_id, model_name in models_dict.items %}
    <li><a href="{% url 'list' model_name %}">{{ model_name }}</a></li>
    <br>
    {% endfor %}
</ol>
{% endblock %}

```

Файл list.py:

```

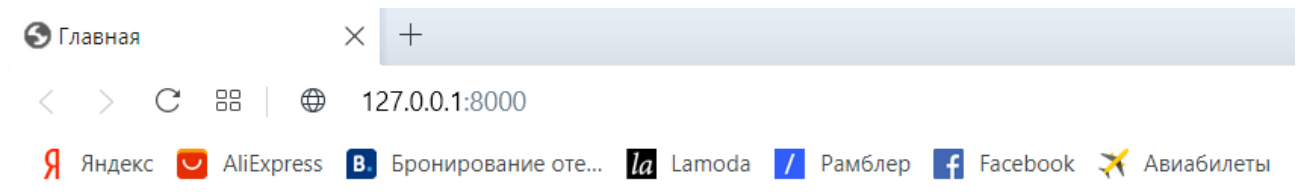
{% extends 'base.html' %}

{% block title %}
{{ model_name }}
{% endblock %}

{% block content %}
<h2>Сущность <i>{{ model_name }}</i></h2>
<ul>
    {% for object in objects %}
    <li>
        {% for key, value in object.items %}
        <i>{{key}}</i>: {{value}}
        <br>
        {% endfor %}
    </li>
    <br>
    {% endfor %}
</ul>
{% endblock %}

```

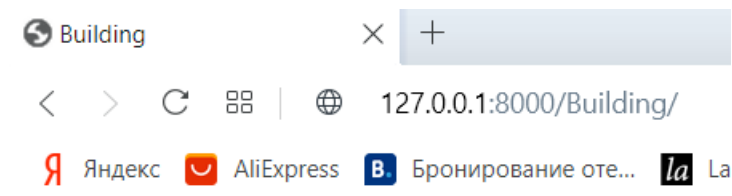
Прототип веб-приложения:



[Главная](#)

Подключенная база данных содержит следующие сущности:

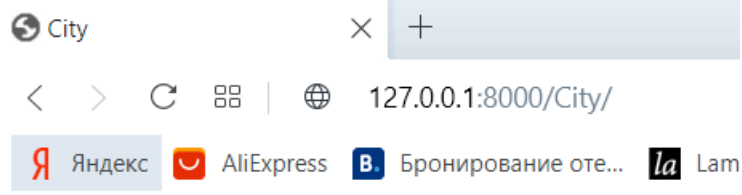
1. [Building](#)
2. [City](#)



[Главная](#)

Сущность *Building*

- *id_building*: 1
building_name: Big Ben
descript: Neo-Gothic design
creation_year: 1859
- *id_building*: 2
building_name: Empire State Building
descript: Art Deco style
creation_year: 1931



[Главная](#)

Сущность *City*

- *id_city*: 1
city_name: London
descript: London is the capital of GB!
id_building_id: 1
- *id_city*: 2
city_name: NYC
descript: NYC is the most popular american city
id_building_id: 2