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 ИУ5-52Б
Васильченко Дарья

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, подключение к базе данных, добавление записей в таблицу и выборка данных:

```
import sqlite3

try:
    sqlite_connection = sqlite3.connect('db.sqlite3')
    cursor = sqlite_connection.cursor()
    print("Подключен к SQLite")

    cursor.execute("""CREATE TABLE IF NOT EXISTS IT_Company(
       ID_company INT PRIMARY KEY,
       Name_company TEXT,
       Foundation_year INT,
       Specialization TEXT);
    """)
    sqlite_connection.commit()

    cursor.execute("""INSERT INTO IT_Company(ID_company, Name_company,
Foundation_year, Specialization)
       VALUES(1, 'Dell Technologies', 1984, 'Production of computers');""")
    sqlite_connection.commit()

    cursor.execute("""INSERT INTO IT_Company(ID_company, Name_company,
Foundation_year, Specialization)
               VALUES(2, 'IBM', 1911, 'Software');""")
    sqlite_connection.commit()

    cursor.execute("""INSERT INTO IT_Company(ID_company, Name_company,
Foundation_year, Specialization)
               VALUES(3, 'Cisco Systems', 1984, 'Network equipment');""")
    sqlite_connection.commit()

    cursor.execute("""CREATE TABLE IF NOT EXISTS Founder(
```

```
            ID_founder INT PRIMARY KEY,
            Founder_name TEXT,
            Career INT,
            ID_company INT);
        """)
    sqlite_connection.commit()

    cursor.execute("""INSERT INTO Founder(ID_founder, Founder_name, Career,
ID_company)
       VALUES(1, 'Michael Dell', 'Businessman', 1);""")
    sqlite_connection.commit()

    cursor.execute("""INSERT INTO Founder(ID_founder, Founder_name, Career,
ID_company)
        VALUES(2, 'Charles Flint', 'Businessman', 2);""")
    sqlite_connection.commit()

    cursor.execute("""INSERT INTO Founder(ID_founder, Founder_name, Career,
ID_company)
        VALUES(3, 'Sandra Lerner', 'Businesswoman', 3);""")
    sqlite_connection.commit()

    cursor.execute("""INSERT INTO Founder(ID_founder, Founder_name, Career,
ID_company)
        VALUES(4, 'Leonard Bosak', 'Businessman', 3);""")
    sqlite_connection.commit()

    cursor.execute("SELECT * FROM Founder WHERE Founder.id_company = 1;")
    one_result = cursor.fetchone()
    print(one_result)

    cursor.close()

except sqlite3.Error as error:
    print("Ошибка при работе с SQLite", error)
finally:
    if sqlite_connection:
        sqlite_connection.close()
        print("Соединение с SQLite закрыто")
```

Файл main_app/models.py:

```
from django.db import models


class ITCompany(models.Model):
    id_company = models.AutoField(db_column='ID_company', primary_key=True)
    name_company = models.CharField(db_column='Name_company', max_length=100,
blank=True, null=False)
    foundation_year = models.IntegerField(db_column='Foundation_year',
blank=True, null=True)
    specialization = models.CharField(db_column='Specialization',
max_length=1000, blank=True, null=True)

    class Meta:
        db_table = 'IT_Company'


class Founder(models.Model):
```

```
    id_founder = models.AutoField(db_column='ID_founder', primary_key=True)
    founder_name = models.CharField(db_column='Founder_name', max_length=100,
blank=True, null=False)
    career = models.CharField(db_column='Career', max_length=1000, blank=True,
null=True)
    id_company = models.ForeignKey('ITCompany', models.DO_NOTHING,
db_column='ID_company', blank=True, null=True)

    class Meta:
        db_table = 'Founder'
```

Файл Lab5/urls.py:

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('main_app.urls'))
]
```

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 main_app/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 main_app/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 main_app/list.py:

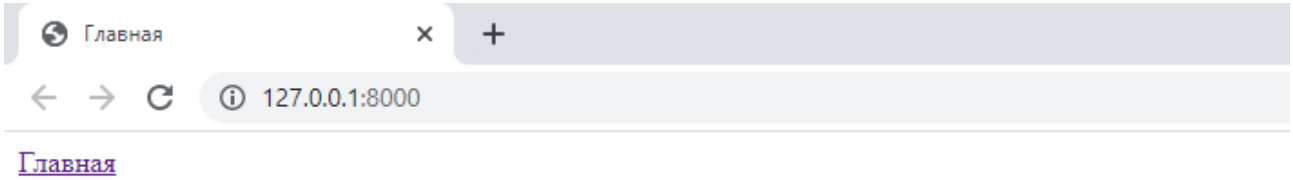
```
{% extends 'base.html' %}


{% 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 %}
```

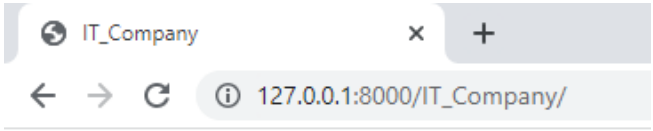
Прототип веб-приложения:



Главная

Подключенная база данных содержит следующие сущности:

1. IT_Company
2. Founder



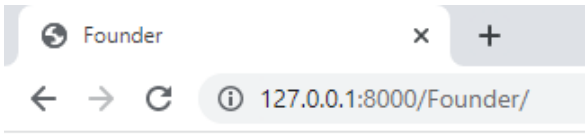
Главная

Сущность *IT_Company*

- *id_company*: 1
  name_company: Dell Technologies
  foundation_year: 1984
  specialization: Production of computers

- *id_company*: 2
  name_company: IBM
  foundation_year: 1911
  specialization: Software

- *id_company*: 3
  name_company: Cisco Systems
  foundation_year: 1984
  specialization: Network equipment



Главная

Сущность *Founder*

- *id_founder*: 1
  founder_name: Michael Dell
  career: Businessman
  id_company_id: 1

- *id_founder*: 2
  founder_name: Charles Flint
  career: Businessman
  id_company_id: 2

- *id_founder*: 3
  founder_name: Sandra Lerner
  career: Businesswoman
  id_company_id: 3

- *id_founder*: 4
  founder_name: Leonard Bosak
  career: Businessman
  id_company_id: 3