



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 ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

Отчет по лабораторной работе №5

Студент Смыслов Максим Алексеевич
фамилия, имя, отчество

Группа ИУ5-55Б

Студент 18.12.2021 Смыслов М.А.
подпись, дата *фамилия, и.о.*

Преподаватель 18.12.2021 Гапанюк Ю.Е.
подпись, дата *фамилия, и.о.*

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.

Текст программы:

“insert_db.py”:

```
import psycopg2

db = psycopg2.connect(
    host="localhost",
    user="arsenvardumyan",
    dbname="lab5_db"
)

c = db.cursor()
c.execute('INSERT INTO "Lab4_person" (name, experience, technology_id) VALUES (%s, %s, %s);',
        ('Natali Harris', 4, 'py'))
db.commit()
c.close()
db.close()
```

“models.py”:

```
from django.db import models

# Create your models here.
class Technology(models.Model):
    id = models.CharField(max_length=5, primary_key=True)
    name = models.CharField(max_length=15)
    description = models.TextField()

class Person(models.Model):
    name = models.CharField(max_length=30)
    experience = models.IntegerField()
    technology = models.ForeignKey(Technology, on_delete=models.CASCADE)
```

“views.py”:

```
from django.shortcuts import render
from Lab4.models import Technology
from Lab4.models import Person

def get_langs(request):
    return render(request, 'langs.html', {
        'langs': Technology.objects.all(),
    })

def get_lang(request, lang_id):
    return render(request, 'lang.html', {
        'lang': Technology.objects.filter(id=lang_id)[0],
        'persons': Person.objects.all().filter(technology_id=lang_id)
    })
```

“lang.html”:

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

{% block title %}{{ lang.name }}{% endblock %}

{% block content %}
    {% load static %}
    <div class="container">
        <div class="card">
            <div class="card__header" style="text-align: center">
                {% if lang.id == "py" %}
                    
                {% else %}
                    
                {% endif %}
            </div>
            <div class="card__body">
                <h4>{{ lang.name }}</h4>
                <p>{{ lang.description }}</p>
            </div>
            <div class="card__footer">
                <div class="user">
                    <div class="user__info">
                        <a href="{% url 'master_url' %}" class="link">
                            <span class="mask">
                                <div class="link-container">
                                    <span class="link-title1 title">Back</span>
                                    <span class="link-title2 title">Back</span>
                                </div>
                            </span>
                        </a>
                    </div>
                </div>
            </div>
        </div>
    </div>

    <div class="main-menu">
```

```

    <p3>People worked with this technology: </p3><br><br>
    <ul>
        {% for person in persons %}
            <li>{{ person.name }} Experience: {{ person.experience }}</li><br>
        {% empty %}
            <p4>Список пуст</p4>
        {% endfor %}
    </ul>
</div>
</div>
{% endblock %}

```

“settings.py”:

```

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'lab5_db',
        'USER': 'postgres',
        'PASSWORD': '',
        'HOST': 'localhost',
        'PORT': 5432,
    }
}

```

“migrations/0001_initial.py”:

```

# Generated by Django 3.2.9 on 2021-12-18 23:55

from django.db import migrations, models
import django.db.models.deletion

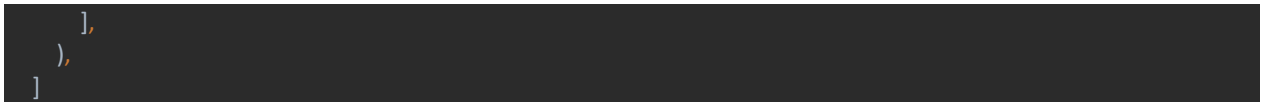
class Migration(migrations.Migration):

    initial = True

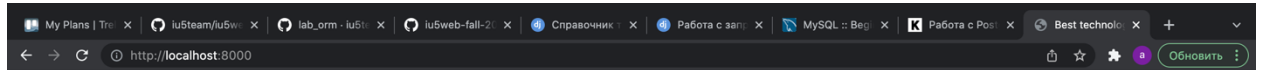
    dependencies = [
    ]

    operations = [
        migrations.CreateModel(
            name='Technology',
            fields=[
                ('id', models.CharField(max_length=5, primary_key=True, serialize=False)),
                ('name', models.CharField(max_length=15)),
                ('description', models.TextField()),
            ],
        ),
        migrations.CreateModel(
            name='Person',
            fields=[
                ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
                ('name', models.CharField(max_length=30)),
                ('experience', models.IntegerField()),
                ('technology', models.ForeignKey(on_delete=django.db.models.deletion.CASCADE, to='Lab4.technology')),
            ],
        ),
    ]

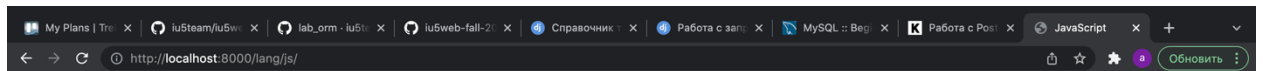
```



Экранные формы с примерами выполнения программы:



JavaScript
Golang
Python



JavaScript

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

[Back](#)

People worked with this technology:

Victor Yakinov
Experience: 5

Rose Black
Experience: 2



Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

[Back](#)

People worked with this technology:

Robe Banks
Experience: 6

Natali Harris
Experience: 4