

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

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

«Работа с СУБД. Обработка данных с использованием Django ORM».

Выполнил:
Анцифров Н. С.
студент группы ИУ5-51Б

Проверил:
Гапанюк Ю. Е.

Подпись и дата:

Подпись и дата:

Москва, 2021 г.

Содержание

1. Задание лабораторной работы.	2
2. Текст программы.	3
3. Выполнение программы.	8

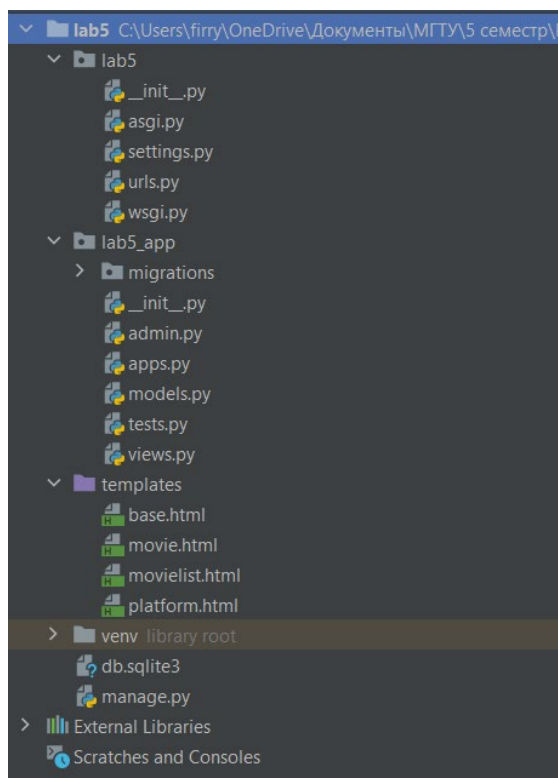
1. Задание лабораторной работы.

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.

2. Текст программы.

Структура проекта:



__init__.py

```
import pymysql

pymysql.install_as_MySQLdb()
```

settings.py

```
from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-
c)tpz*(rrvwa^_6qielwypt3mw3dli%03^&#(ibs!_!u*8%0lt'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
```

```

    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'lab5_app.apps.Lab5AppConfig',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'lab5.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR / 'templates']
        ,
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    ],
]

WSGI_APPLICATION = 'lab5.wsgi.application'

# Database
# https://docs.djangoproject.com/en/3.2/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}

# Password validation
# https://docs.djangoproject.com/en/3.2/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator',
    },

```

```

    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/3.2/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.2/howto/static-files/

STATIC_URL = '/static/'

# Default primary key field type
# https://docs.djangoproject.com/en/3.2/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

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

```

urls.py

```

from django.contrib import admin
from django.urls import path

from lab5_app import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.MovieList),
    path('movie/<int:id>', views.GetMovie, name='movie_url'),
    path('platform/<int:id>', views.GetPlatform, name='platform_url')
]

```

models.py

```
from django.db import models

# Create your models here.

class Platform(models.Model):
    name = models.CharField(max_length=30)
    site = models.CharField(max_length=30)

    class Meta:
        managed = False
        db_table = 'platforms'

class Movie(models.Model):
    name = models.CharField(max_length=30)
    description = models.CharField(max_length=255)
    platform = models.CharField(max_length=30)

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

views.py

```
from django.shortcuts import render

# Create your views here.

from lab5_app.models import Movie
from lab5_app.models import Platform

def MovieList(request):
    return render(request, 'movielist.html', {'data': {
        'platforms': Platform.objects.all(),
        'movies': Movie.objects.all(),
    }})

def GetPlatform(request, id):
    return render(request, 'platform.html', {'data': {
        'platform': Platform.objects.filter(id=id)[0]
    }})

def GetMovie(request, id):
    return render(request, 'movie.html', {'data': {
        'movie': Movie.objects.filter(id=id)[0],
        'platforms': Platform.objects.all(),
    }})
```

base.html

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

movie.html

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

{% block title %}{ { data.movie.name } }{% endblock %}

{% block content %}
  <div>Название: { { data.movie.name } }</div>
  {% for platform in data.platforms %}
    {% if data.movie.platform == platform.id %}
      <div>Платформа: { { platform.name } }</div>
    {% endif %}
  {% endfor %}
  <div>Описание: { { data.movie.description } }</div>
{% endblock %}
```

movielist.html

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

{% block title %}Лабораторная работа №5{% endblock %}

{% block content %}
  <h2>Фильмы:</h2>
<h3><ul>
  {% for movie in data.movies %}
    <li><a href="{% url 'movie_url' movie.id %}">{{movie.name}}</a></li>
  {% empty %}
    <li>Список пуст</li>
  {% endfor %}
</ul>
</h3>
  <h2>Платформы:</h2>
  <h3>
    <ul>
      {% for platform in data.platforms %}
        <li><a href="{% url 'platform_url' platform.id %}">{{platform.name}}</a></li>
      {% empty %}
        <li>Список пуст</li>
      {% endfor %}
    </ul>
  </h3>
{% endblock %}
```

platform.html

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

{% block title %}{{ data.platform.name }}{% endblock %}

{% block content %}
    <div>Название: {{ data.platform.name }}</div>
    <div>Сайт: {{ data.platform.site }}</div>
{% endblock %}
```

3. Выполнение программы.

База данных:

```
mysql> show databases;
+-----+
| Database |
+-----+
| first_db |
| information_schema |
| moviechoice |
| mysql |
| performance_schema |
| sakila |
| sys |
| world |
+-----+
8 rows in set (0.00 sec)

mysql> use moviechoice;
Database changed
```

```
mysql> select * from movies;
+-----+-----+-----+-----+
| id | name | description | platform |
+-----+-----+-----+-----+
| 1 | Назад в будущее (1985) | Подросток Марти Макфлай с помощью машины времени, сооружённой его другом-профессором доком Брауном, попадает из 80-х в далёкие 50-е. Там он встречается со своими будущими родителями, ещё подростками, и другом-профессором, совсем молодым. | 1 |
| 2 | Назад в будущее 2 (1989) | Продолжение фантастической истории о приключениях американского подростка во времени. На этот раз с помощью модернизированной Доком машины времени Марти из 80-х попадает в будущее. Дети Марти в беде, и их надо выручать. Приходится повозиться со злодеем... | 2 |
| 3 | Назад в будущее 3 (1990) | Несмотря на просьбы Дока не навещать его в XIX веке, Марти вновь хочет вмешаться в ход событий и на машине времени уносится в 1885 год, навстречу Дикому Западу. | 1 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from platforms;
+-----+-----+-----+
| id | name | site |
+-----+-----+-----+
| 1 | Кинопоиск | kinopoisk.ru |
| 2 | ivi.ru | ivi.ru |
+-----+-----+-----+
2 rows in set (0.00 sec)
```


Работа программы:

← → ↻ ⓘ 127.0.0.1:8000

Фильмы:

- [Назад в будущее \(1985\)](#)
- [Назад в будущее 2 \(1989\)](#)
- [Назад в будущее 3 \(1990\)](#)

Платформы:

- [Кинопоиск](#)
- [ivi.ru](#)

← → ↻ ⓘ 127.0.0.1:8000/movie/1/

Название: Назад в будущее (1985)

Платформа: Кинопоиск

Описание: Подросток Марти Макфлай с помощью машины времени, сооружённой его другом-профессором доком Брауном, попадает из 80-х в далекие 50-е. Там он встречается со своими будущими родителями, ещё подростками, и другом-профессором, совсем молодым.

← → ↻ ⓘ 127.0.0.1:8000/movie/2/

Название: Назад в будущее 2 (1989)

Платформа: ivi.ru

Описание: Продолжение фантастической истории о приключениях американского подростка во времени. На этот раз с помощью модернизированной Доком машины времени Марти из 80-х попадает в будущее. Дети Марти в беде, и их надо выручать. Приходится повозиться со злодеем...

← → ↻ ⓘ 127.0.0.1:8000/movie/3/

Название: Назад в будущее 3 (1990)

Платформа: Кинопоиск

Описание: Несмотря на просьбы Дока не навещать его в XIX веке, Марти вновь хочет вмешаться в ход событий и на машине времени уносится в 1885 год, навстречу Дикому Западу.

← → ↻ ⓘ 127.0.0.1:8000/platform/1/

Название: Кинопоиск

Сайт: kinopoisk.ru

← → ↻ ⓘ 127.0.0.1:8000/platform/2/

Название: ivi.ru

Сайт: ivi.ru