Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Автоматизированные системы обработки информации и управления»



**Отчет**

**Лабораторная работа № 5**

**По курсу «Разработка интернет приложений»**

**Тема работы: “**Работа с СУБД. Обработка данных с использованием Django ORM.”

**ИСПОЛНИТЕЛЬ:**

Группа ИУ5Ц-74Б

Семенова А.А.

**ПРЕПОДАВАТЕЛЬ:**

Гапанюк Ю.Е.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

"\_\_"\_\_\_\_\_\_\_\_\_\_\_2021 г.

Москва 2021

**СОДЕРЖАНИЕ**

[1. Описание задания 3](#_Toc88129114)

[2. Текст программы 4](#_Toc88129115)

[3. Результаты выполнения программы 14](#_Toc88129116)

# 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.](https://github.com/iu5team/iu5web-fall-2021/blob/main/tutorials/lab5/lab5_tutorial.md)
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

**LAB\_5/asgi.py**

*"""  
ASGI config for LAB\_5 project.  
  
It exposes the ASGI callable as a module-level variable named ``application``.  
  
For more information on this file, see  
https://docs.djangoproject.com/en/3.2/howto/deployment/asgi/  
"""*import os  
  
from django.core.asgi import get\_asgi\_application  
  
os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'LAB\_5.settings')  
  
application = get\_asgi\_application()

**LAB\_5/settings.py**

*"""  
Django settings for LAB\_5 project.  
  
Generated by 'django-admin startproject' using Django 3.2.9.  
  
For more information on this file, see  
https://docs.djangoproject.com/en/3.2/topics/settings/  
  
For the full list of settings and their values, see  
https://docs.djangoproject.com/en/3.2/ref/settings/  
"""*from pathlib import Path  
import os  
  
*# 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-u3dv^@aurnum@$-i43e$5ui$4dlmy9v^ff)+-z+y&l!l9(=%x4'  
  
*# 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',  
 'App.apps.AppConfig',  
]  
  
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 = 'LAB\_5.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 = 'LAB\_5.wsgi.application'  
  
  
*# Database  
# https://docs.djangoproject.com/en/3.2/ref/settings/#databases*DATABASES = {  
 'default': {  
 'ENGINE': 'django.db.backends.mysql',  
 'NAME': 'rip\_lab\_4',  
 'USER': 'admin',  
 'PASSWORD': '1234567890',  
 'HOST': 'SRV2022',  
 'PORT': 3306, *# Стандартный порт MySQL* 'OPTIONS': {'charset': 'utf8'},  
 'TEST\_CHARSET': 'utf8',  
 }  
}  
  
  
*# 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 = 'ru-ru'  
  
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/'  
  
STATICFILES\_DIRS = [  
 BASE\_DIR / "static",  
 os.path.join(BASE\_DIR, "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'

**LAB\_5/urls.py**

*"""LAB\_5 URL Configuration  
  
The `urlpatterns` list routes URLs to views. For more information please see:  
 https://docs.djangoproject.com/en/3.2/topics/http/urls/  
Examples:  
Function views  
 1. Add an import: from my\_app import views  
 2. Add a URL to urlpatterns: path('', views.home, name='home')  
Class-based views  
 1. Add an import: from other\_app.views import Home  
 2. Add a URL to urlpatterns: path('', Home.as\_view(), name='home')  
Including another URLconf  
 1. Import the include() function: from django.urls import include, path  
 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))  
"""*from django.contrib import admin  
from django.urls import path, include  
  
urlpatterns = [  
 path('', include('App.urls')),  
 path('admin/', admin.site.urls)  
]

**LAB\_5/wsgi.py**

*"""  
WSGI config for LAB\_5 project.  
  
It exposes the WSGI callable as a module-level variable named ``application``.  
  
For more information on this file, see  
https://docs.djangoproject.com/en/3.2/howto/deployment/wsgi/  
"""*import os  
  
from django.core.wsgi import get\_wsgi\_application  
  
os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'LAB\_5.settings')  
  
application = get\_wsgi\_application()

**App/admin.py**

from django.contrib import admin  
from .models import Animal  
from .models import Animal\_review  
  
*# Register your models here.*admin.site.register(Animal)  
admin.site.register(Animal\_review)

**App/apps.py**

from django.apps import AppConfig  
  
  
class AppConfig(AppConfig):  
 default\_auto\_field = 'django.db.models.BigAutoField'  
 name = 'App'

**App/models.py**

from django.db import models  
  
  
*# Create your models here.*class Animal(models.Model):  
 animal\_type = models.CharField('Вид животного', max\_length=50)  
 animal\_name = models.CharField('Кличка животного', max\_length=50)  
 animal\_photo = models.URLField('Фото жифотного', max\_length=200)  
  
 def \_\_str\_\_(self):  
 return f'Тип: {self.animal\_type}; Имя: {self.animal\_name}; URL фото: {self.animal\_photo}'  
  
 class Meta:  
 verbose\_name = 'Животное'  
 verbose\_name\_plural = 'Животные'  
  
  
class Animal\_review(models.Model):  
 animal = models.ForeignKey(Animal, on\_delete=models.CASCADE)  
 author = models.CharField('Имя автора', max\_length=50)  
 review\_text = models.CharField('Текст отзыва', max\_length=200)  
  
 def \_\_str\_\_(self):  
 return f'Автор: {self.author}, Отзыв: {self.review\_text}'  
  
 class Meta:  
 verbose\_name = 'Отзыв'  
 verbose\_name\_plural = 'Отзывы'

**App/tests.py**

from django.test import TestCase  
  
*# Create your tests here.*

**App/urls.py**

from django.urls import path  
from . import views  
  
  
app\_name = 'App'  
urlpatterns = [  
 path('', views.index, name='index'),  
 path('<int:animal\_id>', views.detail, name='detail'),  
 path('<int:animal\_id>/leave\_review', views.leave\_review, name='leave\_review')  
]

**App/views.py**

from django.http import Http404, HttpResponseRedirect  
from django.shortcuts import render  
from django.urls import reverse  
from .models import Animal  
  
  
def index(request):  
 a\_list = Animal.objects.all()  
 return render(request, 'App/list.html', dict([('a\_list', a\_list)]))  
  
  
def detail(request, animal\_id):  
 try:  
 a = Animal.objects.get(id=animal\_id)  
 except:  
 raise Http404('Животное не найдено!')  
 reviews = a.animal\_review\_set.order\_by('-id')  
 return render(request, 'App/detail.html', dict([('animal', a), ('review\_list', reviews)]))  
  
  
  
def leave\_review(request, animal\_id):  
 try:  
 a = Animal.objects.get(id=animal\_id)  
 except:  
 raise Http404('Животное не найдено!')  
 a.animal\_review\_set.create(author=request.POST['name'], review\_text=request.POST['text'])  
 return HttpResponseRedirect(reverse('App:detail', args=(a.id,)))

**static/albus.jpg**



**static/barsa.jpg**



**static/stesha.jpg**



**templates/base.html**

<!DOCTYPE html>  
<html lang="ru">  
<head>  
 <meta charset="UTF-8">  
 <title>{% block title %} Лабораторная работа № 4{% endblock %}</title>  
</head>  
<body>  
  
{% block content %}{% endblock %}  
  
<hr>  
<center>Чиварзин А. Е. ИУ5Ц-72Б</center>  
  
</body>  
</html>

**templates/App/list.html**

{% extends 'base.html' %}  
  
{% block title %}ЛР №5{% endblock %}  
  
{% block content %}  
  
 <center><h1>Список животных</h1></center><hr><br>  
  
 {% if a\_list %}  
 {% for a in a\_list %}  
 <a href="{% url 'App:detail' a.id %}">{{ a.animal\_name }}</a><br>  
 {% endfor %}  
 {% else %}  
 Список пуст  
 {% endif %}  
  
{% endblock %}

**templates/App/detail.html**

{% extends 'base.html' %}  
  
{% block title %}{{ animal.animal\_name}}{% endblock %}  
  
{% block content %}  
  
 <center><h1>{{ animal.animal\_name }}</h1></center><hr><br>  
*{# Вид животного: {{ animal.animal\_type }}<br>#}  
{# Имя животного: {{ animal.animal\_name }}<br>#}  
{# <br>#}  
{# Фото животного<br>#}  
{# <img src="{{ animal.animal\_photo }}" alt="Фото животного не найдено">#}* <table border="1" cellpadding="5">  
 <tr>  
 <th>Вид животного</th>  
 <th>{{ animal.animal\_type }}</th>  
 </tr>  
 <tr>  
 <th>Имя животного</th>  
 <th>{{ animal.animal\_name }}</th>  
 </tr>  
 <tr>  
 <th>Фото животного</th>  
 <th><img src="{{ animal.animal\_photo }}" alt="Фото животного не найдено"></th>  
 </tr>  
 </table>  
 <br>  
 <form action="{% url 'App:leave\_review' animal.id %}" method="post">  
 {% csrf\_token %}  
 <input type="text" required placeholder="Имя" name="name"><br>  
 <textarea name="text" required="" placeholder="Текст отзыва" cols="30" rows="10"></textarea>  
 <button type="submit">Отправить отзыв</button>  
  
 </form><hr>  
  
 {% if review\_list %}  
 <table border="1" cellpadding="5">  
 <tr>  
 <th>Автор</th>  
 <th>Отзыв</th>  
 </tr>  
 {% for r in review\_list %}  
 <tr>  
 <th>{{ r.author }}</th>  
 <th>{{ r.review\_text }}</th>  
 </tr>  
 {% endfor %}  
 </table>  
 {% else %}  
 Отзывы не найдены! Оставьте первый!  
 {% endif %}  
  
 <hr><br>  
 <a href="/">Назад</a>  
  
{% endblock %}

# Результаты выполнения программы

**Лог**

"C:\Program Files\JetBrains\PyCharm 2021.2\bin\runnerw.exe" "A:\anzh\YandexDisk\МГТУ\7-й семестр\РИП\ЛР\Py\LAB\_5\Scripts\python.exe" "A:/anzh/YandexDisk/МГТУ/7-й семестр/РИП/ЛР/Py/LAB\_5/manage.py" runserver 8000

Performing system checks...

Watching for file changes with StatReloader

System check identified no issues (0 silenced).

November 18, 2021 - 11:47:43

Django version 3.2.9, using settings 'LAB\_5.settings'

Starting development server at http://127.0.0.1:8000/

Quit the server with CTRL-BREAK.

[18/Nov/2021 11:47:47] "GET / HTTP/1.1" 200 485

Not Found: /favicon.ico

[18/Nov/2021 11:47:47] "GET /favicon.ico HTTP/1.1" 404 2655

[18/Nov/2021 11:49:30] "GET /2 HTTP/1.1" 200 2167

[18/Nov/2021 11:49:30,347] - Broken pipe from ('127.0.0.1', 56007)

[18/Nov/2021 11:51:47] "GET /admin/ HTTP/1.1" 302 0

[18/Nov/2021 11:51:47] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 2300

[18/Nov/2021 11:51:47] "GET /static/admin/css/login.css HTTP/1.1" 200 939

[18/Nov/2021 11:51:47] "GET /static/admin/css/nav\_sidebar.css HTTP/1.1" 200 2271

[18/Nov/2021 11:51:47] "GET /static/admin/css/base.css HTTP/1.1" 200 19513

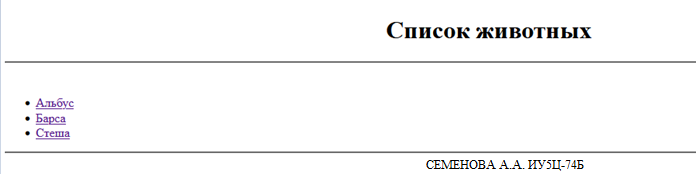
[18/Nov/2021 11:51:47] "GET /static/admin/css/responsive.css HTTP/1.1" 200 18545

[18/Nov/2021 11:51:47] "GET /static/admin/js/nav\_sidebar.js HTTP/1.1" 200 1360

[18/Nov/2021 11:51:47] "GET /static/admin/css/fonts.css HTTP/1.1" 200 423

………………………………………………………………………………………………

**Главная страница (Список животных)**



**Информация о животном (/<int>)**



**Администрирование (/admin)**

