

Защищено:
Гапанюк Ю.Е.

Демонстрация:
Гапанюк Ю.Е.

"__" _____ 2021 г.

"__" _____ 2021 г.

**Отчет по лабораторной работе № 4 по курсу
Разработка интернет-приложений
ГУИМЦ**

**Тема работы: " Создание прототипа веб-приложения с
использованием фреймворка Django "**

14

(количество листов)

Вариант № 4

ИСПОЛНИТЕЛЬ:

студент группы ИУ5Ц-72Б

Чиварзин А.Е.

(подпись)

"__" _____ 2021 г.

СОДЕРЖАНИЕ

1. Описание задания	3
2. Текст программы	4
3. Результаты выполнения программы	12

1. Описание задания

1. Создайте прототип веб-приложения с использованием фреймворка Django:
 - Создайте виртуальное окружение.
 - Установите в него Django.
 - Создайте проект и приложение Django.
2. Создайте представления и шаблоны (по желанию можно использовать модели), реализующие концепцию master/detail со следующей функциональностью:
 - На странице master в виде списка HTML выводится информация о трех объектах (например, о трех сортах мороженого). Каждая строка списка представляет собой гиперссылку, при нажатии на которую происходит переход к странице detail.
 - Страница detail содержит детальное описание объекта (сорта мороженого), фотографию, гиперссылку на master-страницу.
 - Фотография относится к статическому содержимому сайта.
 - Страница detail должна выводить данные с использованием таблицы HTML.
 - Шаблон страницы detail получает от представления данные о детальном объекте с использованием контекста.
 - **НЕОБЯЗАТЕЛЬНЫЙ ПУНКТ.** По желанию можно использовать верстку с применением Bootstrap (или аналогичного фреймворка), а также представления на основе классов (class-based views).

2. Текст программы

LAB_4/asgi.py

```
"""
ASGI config for LAB_4 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_4.settings')

application = get_asgi_application()
```

LAB_4/settings.py

```
"""
Django settings for LAB_4 project.

Generated by 'django-admin startproject' using Django 3.2.8.

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-
jf4pbm$v0bz&_!un9j4a3zvzgplu%6845txtdvhtq9^av4y6e5'

# 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',
'master.apps.LabAppConfig',
]

```

```

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_4.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_4.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 = '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 = [
    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_4/urls.py

```

"""LAB_4 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('master/', include('master.urls')),
    path('admin/', admin.site.urls)
]

```

LAB_4/wsgi.py

```

"""
WSGI config for LAB_4 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_4.settings')

application = get_wsgi_application()

```

master/admin.py

```

from django.contrib import admin
from .models import Animal

# Register your models here.
admin.site.register(Animal)

```

master/apps.py

```

from django.apps import AppConfig

class LabAppConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'master'
    verbose_name = 'JIP-4'

```

master/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 животного', 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'

```

master/tests.py

```

from django.test import TestCase

# Create your tests here.

```

master/urls.py

```

from django.urls import path
from . import views

app_name = 'master'
urlpatterns = [
    path('', views.index, name='index'),
    path('<int:animal_id>', views.detail, name='detail')
]

```

master/views.py

```

from django.http import Http404
from django.shortcuts import render
from .models import Animal

def index(request):
    a_list = Animal.objects.all()
    return render(request, 'master/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!')
    return render(request, 'master/detail.html', dict([('animal', a)])

```


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/master/list.html

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

{% block title %}ЛП №4{% endblock %}

{% block content %}

    <center><h1>СПИСОК ЖИВОТНЫХ</h1></center><hr><br>

    {% if a_list %}
        {% for a in a_list %}
            <a href="{% url 'master:detail' a.id %}">{{ a.animal_name
}}</a><br>
        {% endfor %}
    {% else %}
        Список пуст
    {% endif %}

{% endblock %}
```

templates/master/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>#}
    {# #}
    <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></th>
    </tr>
    </table>
    <br><hr><br>
    <a href="/master">Назад</a>

{% endblock %}
```

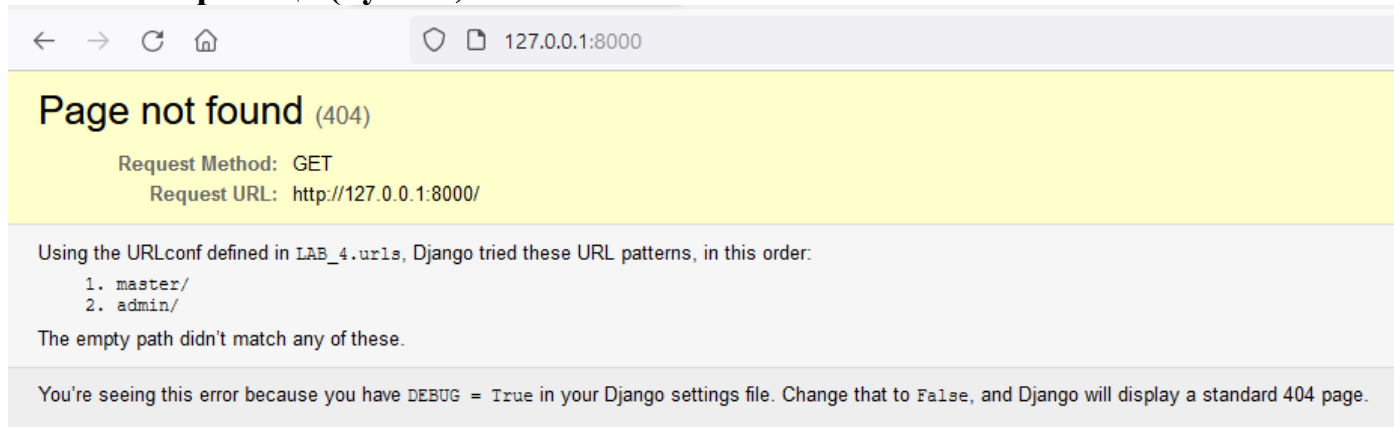
3. Результаты выполнения программы

Лог

```
"C:\Program Files\JetBrains\PyCharm 2021.2\bin\runnerw.exe"  
"A:\sasha\YandexDisk\МГТУ\7-й  
семестр\РИП\ЛР\Py\LAB_4\venv\Scripts\python.exe"  
"A:/sasha/YandexDisk/МГТУ/7-й семестр/РИП/ЛР/Py/LAB_4/manage.py"  
runserver 8000  
Performing system checks...
```

```
Watching for file changes with StatReloader  
System check identified no issues (0 silenced).  
October 20, 2021 - 13:52:08  
Django version 3.2.8, using settings 'LAB_4.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CTRL-BREAK.  
Not Found: /  
[20/Oct/2021 13:52:12] "GET / HTTP/1.1" 404 2165  
[20/Oct/2021 13:52:20] "GET /master/ HTTP/1.1" 200 463  
Not Found: /master/111111  
[20/Oct/2021 13:52:25] "GET /master/111111 HTTP/1.1" 404 2527  
[20/Oct/2021 14:21:53] "GET /master/1 HTTP/1.1" 200 721  
[20/Oct/2021 14:21:53] "GET /static/stesha.jpg HTTP/1.1" 304 0  
[20/Oct/2021 14:22:00] "GET /master/ HTTP/1.1" 200 463  
.....
```

Главная страница (пустая)



← → ↻ 🏠 127.0.0.1:8000

Page not found (404)

Request Method: GET
Request URL: http://127.0.0.1:8000/

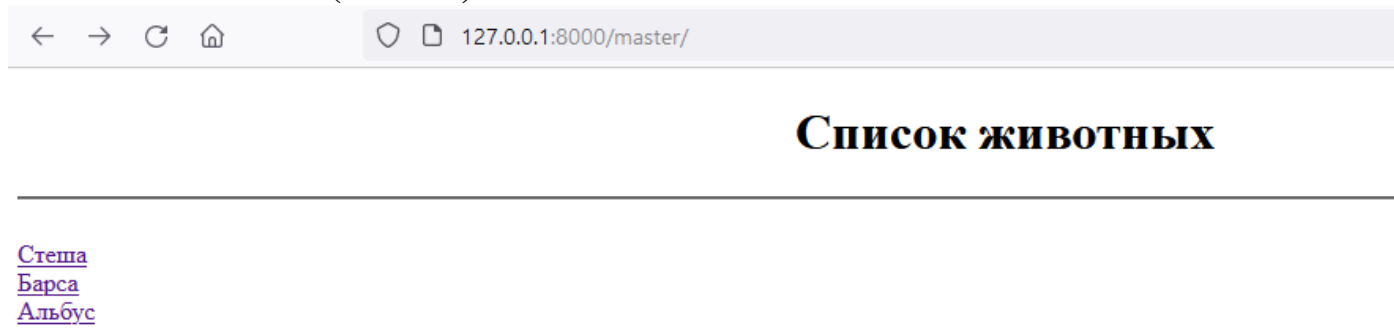
Using the URLconf defined in LAB_4.urls, Django tried these URL patterns, in this order:

1. master/
2. admin/

The empty path didn't match any of these.

You're seeing this error because you have `DEBUG = True` in your Django settings file. Change that to `False`, and Django will display a standard 404 page.

Список животных (/master)



← → ↻ 🏠 127.0.0.1:8000/master/


Список животных

[Стеша](#)
[Барса](#)
[Альбус](#)

Чиварзин А. Е. ИУ5Ц-72Б

Информация о животном (/master/<int>)

Барса

Вид животного	Кошка
Имя животного	Барса
Фото животного	

[Назад](#)

Чиварин А. Е. ИУ5Ц-72Б

Администрирование (/admin)

[←](#) [→](#) [↺](#) [🏠](#) [🔒](#) [📄](#) 127.0.0.1:8000/admin/login/?next=/admin/

Администрирование Django

Имя пользователя:

Пароль:

Войти

← → ↺ 🏠

🔒 📄 🔑 127.0.0.1:8000/admin/

Администрирование Django

Администрирование сайта

ЛР-4

Животные

+ Добавить

✎ Изменить

ПОЛЬЗОВАТЕЛИ И ГРУППЫ

Группы

+ Добавить

✎ Изменить

Пользователи

+ Добавить

✎ Изменить

Последние действия

Мои действия

✎ Тип: Кошка; Имя: Стеша; URL фото: http://127.0.0.1:8000/static/stesha.jpg

Животное

✎ Тип: Собака; Имя: Альбус; URL фото: http://127.0.0.1:8000/static/albus.jpg

Животное

✎ Тип: Кошка; Имя: Барса; URL фото: http://127.0.0.1:8000/static/barsa.jpg

Животное

✖ administrator

Пользователь

← → ↺ 🏠

🔒 📄 🔑 127.0.0.1:8000/admin/master/animal/

📄 ☆ 📄 📄 📄 📄 >> ☰

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, ADMIN. ОТКРЫТЬ САЙТ / ИЗМЕНИТЬ ПАРОЛЬ / ВЫЙТИ

Начало > ЛР-4 > Животные

ЛР-4

Животные

+ Добавить

ПОЛЬЗОВАТЕЛИ И ГРУППЫ

Группы

+ Добавить

Пользователи

+ Добавить

Выберите Животное для изменения

ДОБАВИТЬ ЖИВОТНОЕ +

Действие:

Выполнить

 Выбрано 0 объектов из 3

☐ ЖИВОТНОЕ

☐ Тип: Собака; Имя: Альбус; URL фото: http://127.0.0.1:8000/static/albus.jpg

☐ Тип: Кошка; Имя: Барса; URL фото: http://127.0.0.1:8000/static/barsa.jpg

☐ Тип: Кошка; Имя: Стеша; URL фото: http://127.0.0.1:8000/static/stesha.jpg

3 Животные

← → ↺ 🏠

🔒 📄 🔑 127.0.0.1:8000/admin/master/animal/2/change/

📄 ☆ 📄 📄 📄 📄 >> ☰

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, ADMIN. ОТКРЫТЬ САЙТ / ИЗМЕНИТЬ ПАРОЛЬ / ВЫЙТИ

Начало > ЛР-4 > Животные > Тип: Кошка; Имя: Барса; URL фото: http://127.0.0.1:8000/static/barsa.jpg

ЛР-4

Животные

+ Добавить

ПОЛЬЗОВАТЕЛИ И ГРУППЫ

Группы

+ Добавить

Пользователи

+ Добавить

Изменить Животное

ИСТОРИЯ

Тип: Кошка; Имя: Барса; URL фото: http://127.0.0.1:8000/static/barsa.jpg

Вид животного:

Кошка

Кличка животного:

Барса

Фото животного:

Сейчас: http://127.0.0.1:8000/static/barsa.jpg

Изменить:

http://127.0.0.1:8000/static/barsa.jpg

Удалить

Сохранить и добавить другой объект

Сохранить и продолжить редактирование

СОХРАНИТЬ