

Отчет по лабораторной работе № 4 по курсу
"Разработка Интернет-Приложений"

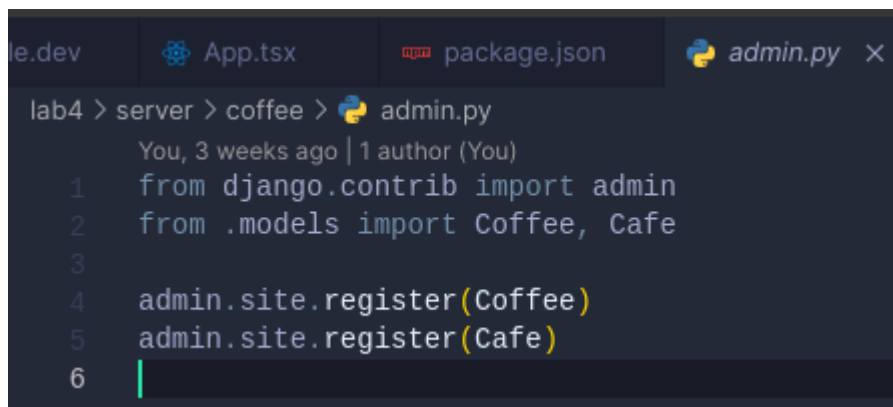
Выполнил:
Студент группы
ИУ5-55 Б
Зубарева Антонина Михайловна

Москва, МГТУ – 2021

Задание:

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).

Текст программы:



The screenshot shows a code editor with a dark theme. At the top, there are tabs for 'le.dev', 'App.tsx', 'package.json', and 'admin.py'. The 'admin.py' tab is active. Below the tabs, the file path 'lab4 > server > coffee > admin.py' is displayed. The code content is as follows:

```
1 from django.contrib import admin
2 from .models import Coffee, Cafe
3
4 admin.site.register(Coffee)
5 admin.site.register(Cafe)
6
```

settings.py

```
"""
Django settings for lab4 project.

Generated by 'django-admin startproject' using Django 3.1.4.

For more information on this file, see
https://docs.djangoproject.com/en/3.1/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/3.1/ref/settings/
"""

from pathlib import Path

import environ

import os

env = environ.Env(DEBUG=(bool, True))

# 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/3.1/howto/deployment/checklist/

# Take environment variables from .env file

environ.Env.read_env(os.path.join(BASE_DIR, '.env'))

SECRET_KEY = env('SECRET_KEY')


DEBUG = env('DEBUG')


ALLOWED_HOSTS = ['localhost', '127.0.0.1', '0.0.0.0']


# Application definition

INSTALLED_APPS = [

    'coffee.apps.CoffeeConfig',

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',
```

```
'corsheaders',

'rest_framework',

]

MIDDLEWARE = [

    'django.middleware.security.SecurityMiddleware',

    'django.contrib.sessions.middleware.SessionMiddleware',

    'corsheaders.middleware.CorsMiddleware',

    'django.middleware.common.CommonMiddleware',

    'django.middleware.csrf.CsrfViewMiddleware',

    'django.contrib.auth.middleware.AuthenticationMiddleware',

    'django.contrib.messages.middleware.MessageMiddleware',

    'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

# If this is used then `CORS_ALLOWED_ORIGINS` will not have any effect

CORS_ALLOW_ALL_ORIGINS = True

CORS_ALLOW_CREDENTIALS = True

CORS_ALLOWED_ORIGINS = [

    'http://localhost:3000',

] # If this is used, then not need to use `CORS_ALLOW_ALL_ORIGINS = True`

CORS_ALLOWED_ORIGIN_REGEXES = [
```

```
    'http://localhost:3000',  
]  
  
ROOT_URLCONF = 'lab4.urls'  
  
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': [],  
        '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 = 'lab4.wsgi.application'
```

```
# Database

# https://docs.djangoproject.com/en/3.1/ref/settings/#databases

DATABASES = {

    'default': {

        'ENGINE': env('POSTGRES_ENGINE',
default='django.db.backends.sqlite3'),

        'NAME': env('POSTGRES_NAME', default=(os.path.join(BASE_DIR,
'db.sqlite3'))),

        'USER': env('POSTGRES_USER'),

        'PASSWORD': env('POSTGRES_PASSWORD'),

        'HOST': env('POSTGRES_HOST', default='localhost'),

        'PORT': env('POSTGRES_PORT', default='5432'),

    }

}

# Password validation

#
https://docs.djangoproject.com/en/3.1/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/3.1/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'Europe/Moscow'

```



```
USE_I18N = True
```

```
USE_L10N = True
```

```
USE_TZ = True
```

```
# Static files (CSS, JavaScript, Images)
```

```
# https://docs.djangoproject.com/en/3.1/howto/static-files/
```

```
STATIC_URL = '/static/'
```

```
STATICFILES_DIRS = [
```

```
    BASE_DIR / "static",
```

```
]
```

```
# Media path
```

```
MEDIA_URL = '/media/'
```

```
MEDIA_ROOT = BASE_DIR / 'media'
```

models.py

```
models.py x
lab4 > server > coffee > models.py > Coffee
You, 3 weeks ago | 1 author (You)
1 from django.db import models
2
3
4 You, 3 weeks ago | 1 author (You)
5 class Cafe(models.Model):
6     name = models.CharField(max_length=200)
7     address = models.CharField(max_length=300)
8
9     def __str__(self):
10         return self.name
11
12 You, 3 weeks ago | 1 author (You)
13 class Coffee(models.Model):
14     name = models.CharField(max_length=200, verbose_name="Name")
15     price = models.PositiveIntegerField(verbose_name="Price")
16     description = models.TextField(verbose_name="Description")
17     score = models.PositiveSmallIntegerField(verbose_name="Score")
18     country = models.CharField(max_length=200, verbose_name="Country")
19     main_image = models.ImageField(
20         upload_to='coffee_assets/',
21         blank=True
22     )
23     cafe = models.ForeignKey(
24         'Cafe',
25         on_delete=models.CASCADE,
26     )
27
28     def __str__(self):
29         return self.name
```

views.py

```
views.py 1 x
lab4 > server > coffee > views.py > ...
You, 3 weeks ago | 1 author (You)
1 from rest_framework.viewsets import ModelViewSet
2 from .serializers import CoffeeSerializer, CafeSerializer
3 from .models import Coffee, Cafe
4
5
You, 3 weeks ago | 1 author (You)
6 class CoffeeViewSet(ModelViewSet):
7     queryset = Coffee.objects.all()
8     serializer_class = CoffeeSerializer
9
10
You, 3 weeks ago | 1 author (You)
11 class CafeViewSet(ModelViewSet):
12     queryset = Cafe.objects.all()
13     serializer_class = CafeSerializer
14
```

urls.py

```
urls.py 1 x
lab4 > server > lab4 > urls.py > ...
You, 3 weeks ago | 1 author (You)
1 from django.contrib import admin
2 from django.urls import path, include
3 from coffee import views as coffee_views
4 from rest_framework import routers
5
6 from django.conf import settings
7 from django.conf.urls.static import static
8
9 router = routers.DefaultRouter()
10 router.register(r'coffee', coffee_views.CoffeeViewSet)
11 router.register(r'cafe', coffee_views.CafeViewSet)
12
13 urlpatterns = [
14     path('admin/', admin.site.urls),
15     path('', include(router.urls)),
16     path('api-auth/', include('rest_framework.urls', namespace='rest_framework')),
17 ]
18
19 urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
20
```

Результат работы программы:



INDONESIA FRINSA MANIS

Brrrew!

Indonesia

Лот Маниса - по вкусу напоминает черри колу, красное яблоко,



COLOMBIA EL PARAISO Y_Y

Brrrew!

Colombia

Сочный фруктовый микс из личи, абрикосового джема, персиков и вишни, молочного улуна, сбалансированная кислотность, сладкое,



INDONESIA FRINSA MANIS

Brrrew!

Ethiopia

Лот Маниса - по вкусу напоминает черри колу, красное яблоко,

bloom-n-brew

Москва, ул. Красносельская

Delete

Edit

Brrrew!

Бауманская

Delete

Edit