Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Автоматизированные системы обработки информации и управления»



**Отчет**

**Домашнее задание**

**По курсу «Разработка интернет приложений»**

**ИСПОЛНИТЕЛЬ:**

Группа ИУ5-55Б

Васильев С.Д.

**ПРЕПОДАВАТЕЛЬ:**

Гапанюк Ю.Е.

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

"\_\_"\_\_\_\_\_\_\_\_\_\_\_2021 г.

Москва 2021

1. **Задание**

Создайте прототип веб-приложения с использованием фреймворка Django на основе базы данных, реализующий концепцию master/detail. Прототип должен содержать:

* Две модели, связанные отношением один-ко-многим
* Стандартное средство администрирования Django позволяет редактировать данные моделей. Желательно настроить русификацию ввода и редактирования данных.
* Веб0приложение формирует отчет в виде отдельного view/template, отчет выводит HTML-страницу, содержащую связанные данные из двух моделей.
* Для верстки шаблонов используется фреймворк Bootstrap, или аналогичный фреймворк по желанию студента

1. **Текст программы**

**DZ\settings.py**

*"""  
Django settings for DZ1 project.  
  
Generated by 'django-admin startproject' using Django 3.1.5.  
  
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  
  
# 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/  
  
# SECURITY WARNING: keep the secret key used in production secret!  
SECRET\_KEY = '=v\*ltfw9lirf=+1s^4\_r=i\_e\_(@9lk7rd#4s&$\*#sq!2fp)f6z'  
  
# 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',  
 'sorev'  
]  
  
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 = 'DZ1.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 = 'DZ1.wsgi.application'  
  
  
# Database  
# https://docs.djangoproject.com/en/3.1/ref/settings/#databases  
  
DATABASES = {  
 'default': {  
 'ENGINE': 'django.db.backends.sqlite3',  
 'NAME': BASE\_DIR / 'db.sqlite3',  
 }  
}  
  
  
# 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 = 'ru'  
  
TIME\_ZONE = 'UTC'  
  
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/'

**DZ\urls.py**

from django.contrib import admin  
from django.urls import path, include  
  
urlpatterns = [  
 path('admin/', admin.site.urls),  
 path("", include("sorev.urls"))  
]

**sorev\models.py**

from django.db import models  
from datetime import date  
  
from django.urls import reverse  
  
  
class Student(models.Model):  
  
 name = models.CharField("Имя", max\_length=100)  
 age = models.PositiveSmallIntegerField("Возраст", default=0)  
 description = models.TextField("Описание")  
  
 def \_\_str\_\_(self):  
 return self.name  
  
 class Meta:  
 verbose\_name = "Ученики"  
 verbose\_name\_plural = "Ученики"  
  
  
class Teacher(models.Model):  
 title = models.CharField("Название", max\_length=100)  
 tagline = models.CharField("Слоган", max\_length=100, default='')  
 description = models.TextField("Описание")  
 name = models.ManyToManyField(Student, verbose\_name="Ученики", related\_name="prepod\_student")  
  
  
 def \_\_str\_\_(self):  
 return self.title  
  
 def get\_absolute\_url(self):  
 return reverse("teacher\_detail", kwargs={"slug": self.url})  
  
 class Meta:  
 verbose\_name = "Преподователь"  
 verbose\_name\_plural = "Преподователи"

**sorev\admin.py**

from django.contrib import admin  
from .models import Teacher, Student  
  
  
admin.site.register(Teacher)  
  
admin.site.register(Student)

**sorev \ursl.py**

from django.urls import path  
  
from . import views  
  
  
urlpatterns = [  
 path("", views.TeacherView.as\_view()),  
 path("<int:pk>/", views.TeacherDetailView.as\_view(), name="teacher"),  
  
]

**sorev \views.py**

from django.shortcuts import render  
  
# Create your views here.  
  
from django.views.generic.base import View  
  
from .models import Teacher  
  
  
class TeacherView(View):  
 def get(self, request):  
 teachers = Teacher.objects.all()  
 return render(request, "DZ/teacher\_list.html", {"teacher\_list": teachers})  
  
class TeacherDetailView(View):  
 def get(self, request, pk):  
 teacher = Teacher.objects.get(id=pk)  
 return render(request, "DZ/1.html", {"teacher": teacher})

**templates\ DZ \teacher\_list.html**

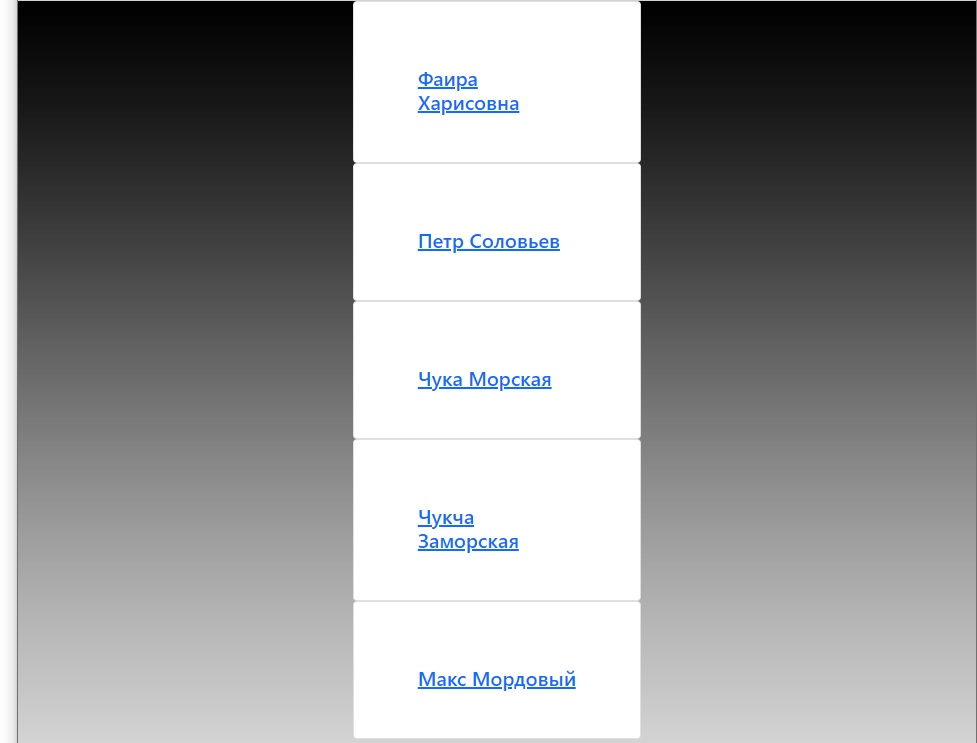
<!DOCTYPE html>  
<html lang="en">  
<head>  
  
 <meta charset="UTF-8">  
 <title>Васильев Семён ИУ5-52Б</title>  
  
 </style>  
  
 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.3/umd/popper.min.js"></script>  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1" crossorigin="anonymous">  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-ygbV9kiqUc6oa4msXn9868pTtWMgiQaeYH7/t7LECLbyPA2x65Kgf80OJFdroafW" crossorigin="anonymous"></script>  
</head>  
<body>  
 <style type="text/css">  
 BODY {  
 background: url(https://www.s-d-a.pl/wp-content/uploads/2017/09/gradient.png) repeat-x; /\* Параметры фона \*/  
 text-align: center; /\* Выравнивание блока по центру \*/  
 }  
 DIV {  
 margin: auto; /\* Отступы вокруг блока \*/  
 width: 75%; /\* Ширина блока \*/  
 height: 90%; /\* Высота блока \*/  
 text-align: left; /\* Выравнивание текста по левому краю \*/  
 padding: 10px; /\* Поля вокруг текста \*/  
 background: white; /\* Цвет фона \*/  
 }  
 </style>  
{% for teacher in teacher\_list %}  
 <div class="card px-3 py-3 pt-md-5 pb-md-4 mx-auto text-center" style="width: 18rem;">  
 <div class="card-body">  
 <h5 class="card-title">  
 <a href="{% url "teacher" pk=teacher.pk %}" class="editContent">  
 {{ teacher.title }}  
 </a>  
 </h5>  
  
 </div>  
 </div>  
{% endfor %}  
</body>  
</html>

**templates\ sorev \1.** <!DOCTYPE html>  
<html lang="en">  
<head>  
  
 </style>  
  
 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.3/umd/popper.min.js"></script>  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1" crossorigin="anonymous">  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-ygbV9kiqUc6oa4msXn9868pTtWMgiQaeYH7/t7LECLbyPA2x65Kgf80OJFdroafW" crossorigin="anonymous"></script>  
  
  
</head>  
<body>  
  
<style type="text/css">  
 BODY {  
 background: url(https://www.s-d-a.pl/wp-content/uploads/2017/09/gradient.png) repeat-x; /\* Параметры фона \*/  
 text-align: center; /\* Выравнивание блока по центру \*/  
 }  
 DIV {  
 margin: auto; /\* Отступы вокруг блока \*/  
 width: 75%; /\* Ширина блока \*/  
 height: 90%; /\* Высота блока \*/  
 text-align: left; /\* Выравнивание текста по левому краю \*/  
 padding: 10px; /\* Поля вокруг текста \*/  
 background: white; /\* Цвет фона \*/  
 }  
 </style>  
<li><a href="/">Назад</a></li>  
 </ul>  
  
<div class="left-ads-display col-lg-8">  
 <div class="row">  
  
 <div class="desc1-right col-md-6 pl-lg-4">  
 <h3 class="editContent" style="outline: none; cursor: inherit;">  
 {{ teacher.title }}  
 </h3>  
 <h5 class="editContent"></h5>  
 <ul>  
 </li>  
 <li style="list-style: none">  
 <span><b>Описание:</b>  
 {{ teacher.description }}  
 </span>  
 </li>  
 <li style="list-style: none">  
 <span><b>Ученик:</b>  
 {% for name in teacher.name.all %}  
 {{ name.name }}  
 {% endfor %}  
 </span>  
 </li>  
  
  
 </ul>  
 </div>  
 </div>  
</div>  
</body>  
</html>

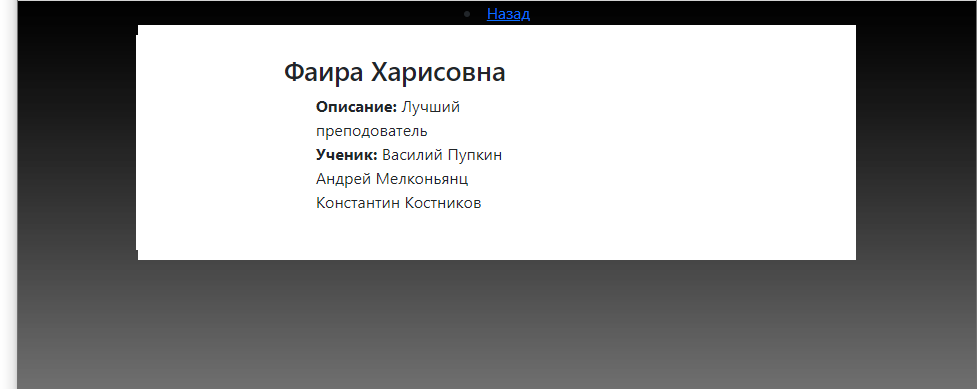
1. **Экранные формы с примерами выполнения программы**

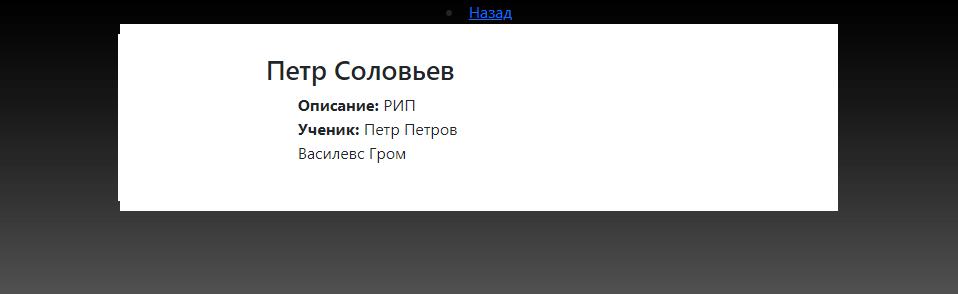
Стартовая страничка.

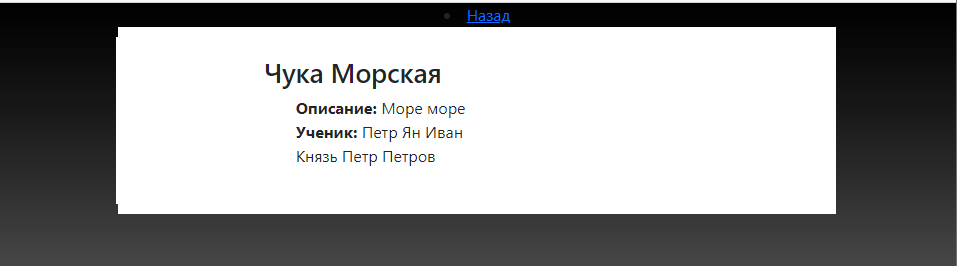
Предметная область: Сеть магазинов «Учебный состав»

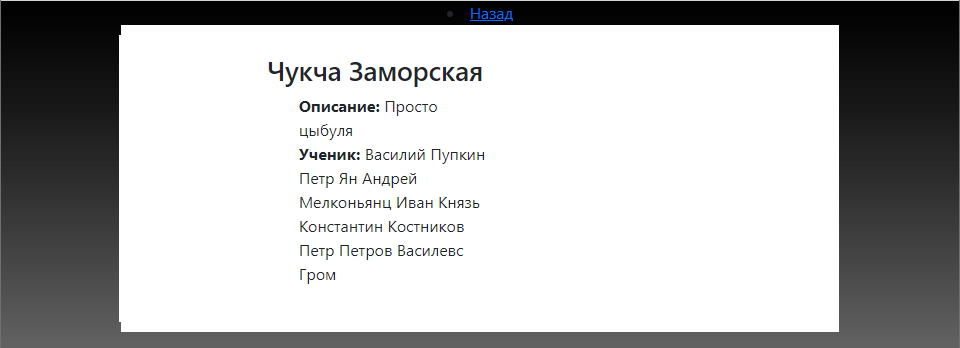


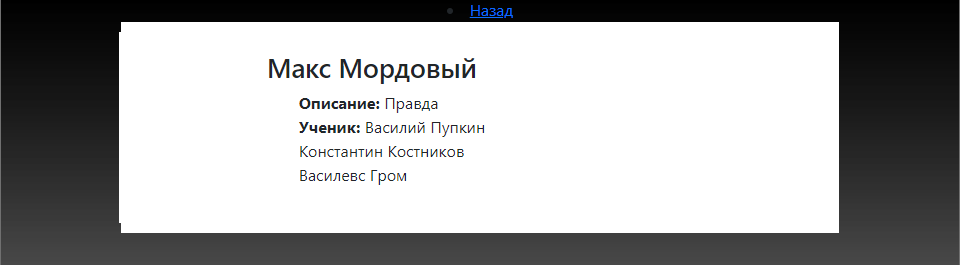
При переходе по ссылке для каждого отдельного преподователя











Русифицированная административная панель

