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

Попов М.А.

"13" января 2020 г.

**ПРЕПОДАВАТЕЛЬ:**

Гапанюк Ю.Е.

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

"\_\_"\_\_\_\_\_\_\_\_\_\_\_2020 г.

Москва 2020

1. **Задание**

Создайте прототип веб-приложения с использованием фреймоврка 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

Расширенные задания:

* Реализация связи много-ко-многим (с возможностью редактирования данных в пользовательском интерфейсе) и развертывание приложения на облачном сервисе (например, heroku)
* Реализация в отчете (пункт 3 стандартного задания) графика на основе данных отчета с использованием библиотек JavaScript (например, c3js) и развертывание приложения на облачном сервисе (например, heroku)

1. **Текст программы**

HW\settings.py

*"""  
Django settings for HW 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  
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/3.1/howto/deployment/checklist/  
  
# SECURITY WARNING: keep the secret key used in production secret!  
SECRET\_KEY = 'y\*!-975h7$d\*$-&r5mxzdt+(t5bg#7knza&^26hpbm5@7shc2r'  
  
# SECURITY WARNING: don't run with debug turned on in production!  
DEBUG = False  
  
ALLOWED\_HOSTS = ['schedulate.herokuapp.com', '127.0.0.1']  
  
  
# Application definition  
  
INSTALLED\_APPS = [  
 'schedule.apps.ScheduleConfig',  
 'django.contrib.admin',  
 'django.contrib.auth',  
 'django.contrib.contenttypes',  
 'django.contrib.sessions',  
 'django.contrib.messages',  
 'django.contrib.staticfiles',  
]  
  
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',  
 'whitenoise.middleware.WhiteNoiseMiddleware',  
]  
  
ROOT\_URLCONF = 'HW.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 = 'HW.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/'  
STATIC\_ROOT = os.path.join(BASE\_DIR, 'staticfiles')

HW\urls.py

*"""HW URL Configuration  
  
The `urlpatterns` list routes URLs to views. For more information please see:  
 https://docs.djangoproject.com/en/3.1/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 include, path  
  
urlpatterns = [  
 path('schedule/', include('schedule.urls')),  
 path('admin/', admin.site.urls),  
]

Procfile

web: gunicorn HW.wsgi

schedule\models.py

from django.db import models  
  
  
class Exam(models.Model):  
 name = models.CharField(max\_length=50, verbose\_name='Название')  
 lecturer = models.CharField(max\_length=50, verbose\_name='Преподаватель')  
  
 def \_\_str\_\_(self):  
 return self.name  
  
 class Meta:  
 verbose\_name = 'Экзамен'  
 verbose\_name\_plural = 'Экзамены'  
 ordering = ['name']  
  
  
class Group(models.Model):  
 name = models.CharField(max\_length=50, verbose\_name='Номер')  
 lessons = models.ManyToManyField(Exam, through='Schedule')  
  
 def \_\_str\_\_(self):  
 return self.name  
  
 class Meta:  
 verbose\_name = 'Группа'  
 verbose\_name\_plural = 'Группы'  
 ordering = ['name']  
  
  
class Schedule(models.Model):  
 group = models.ForeignKey(Group, on\_delete=models.CASCADE, verbose\_name='Группа')  
 lesson = models.ForeignKey(Exam, on\_delete=models.CASCADE, verbose\_name='Экзамен')  
 lesson\_time = models.DateTimeField(default='2021-01-10', verbose\_name='Время экзамена')  
 consultation\_time = models.DateTimeField(default='2021-01-10', verbose\_name='Время консультации')  
  
 class Meta:  
 verbose\_name = 'Расписание'  
 verbose\_name\_plural = 'Расписание'  
 ordering = ['group']

schedule\admin.py

from django.contrib import admin  
from .models import Exam, Group, Schedule  
  
  
class ExamAdmin(admin.ModelAdmin):  
 list\_display = ('name', 'lecturer')  
 search\_fields = ('name', 'lecturer')  
  
  
class ScheduleAdmin(admin.ModelAdmin):  
 list\_display = ('group', 'lesson', 'lesson\_time', 'consultation\_time')  
  
  
admin.site.register(Exam, ExamAdmin)  
admin.site.register(Group)  
admin.site.register(Schedule, ScheduleAdmin)

schedule\ursl.py

from django.urls import path  
from . import views  
  
app\_name = 'schedule'  
urlpatterns = [  
 path('', views.index, name='index'),  
 path('<int:group\_id>/', views.detail\_student, name='detail\_student'),  
 path('<str:name>/', views.detail\_lecturer, name='detail\_lecturer')  
]

schedule\views.py

from django.shortcuts import get\_list\_or\_404, render  
from django.http import HttpResponseRedirect  
from django.template import loader  
from .models import Exam, Group, Schedule  
  
  
def index(request):  
 group\_list = Group.objects.all()  
 exam\_list = Exam.objects.all()  
 context = {'group\_list': group\_list, 'exam\_list': exam\_list}  
 return render(request, 'schedule/index.html', context)  
  
  
def detail\_student(request, group\_id):  
 group\_list = Group.objects.all()  
 for i in group\_list:  
 if i.id == group\_id:  
 group = i  
 schedule\_list = get\_list\_or\_404(Schedule, group=group\_id)  
 for i in range(len(schedule\_list) - 1):  
 for j in range(len(schedule\_list) - i - 1):  
 if schedule\_list[j].lesson\_time > schedule\_list[j + 1].lesson\_time:  
 schedule\_list[j], schedule\_list[j + 1] = schedule\_list[j + 1], schedule\_list[j]  
 exam\_list = []  
 for i in schedule\_list:  
 exam\_list.append(i.lesson)  
 # for exam in exam\_list:  
 return render(request, 'schedule/detail\_student.html', {'group': group,  
 'schedule\_list': schedule\_list,  
 'exam\_list': exam\_list})  
  
  
def detail\_lecturer(request, name):  
 exam\_list = Exam.objects.all()  
 for i in exam\_list:  
 if i.lecturer == name:  
 exam = i  
 schedule\_list = get\_list\_or\_404(Schedule, lesson=exam)  
 for i in range(len(schedule\_list) - 1):  
 for j in range(len(schedule\_list) - i - 1):  
 if schedule\_list[j].lesson\_time > schedule\_list[j + 1].lesson\_time:  
 schedule\_list[j], schedule\_list[j + 1] = schedule\_list[j + 1], schedule\_list[j]  
 group\_list = []  
 for i in schedule\_list:  
 group\_list.append(i.group)  
  
 # формируем листы дат консультаций и экзаменов  
 # а также формируем листы количества консультаций и экзаменов в датологическом порядке  
 schedule\_exam\_data\_list = []  
 schedule\_cons\_data\_list = []  
 count\_ex = 1  
 count\_cons = 1  
 cons\_count = [1]  
 exam\_count = [1]  
 for schedule in schedule\_list:  
 current\_exam\_date\_str = str(schedule.lesson\_time)  
 current\_cons\_date\_str = str(schedule.consultation\_time)  
 current\_exam\_date = current\_exam\_date\_str.split(' ')  
 current\_cons\_date = current\_cons\_date\_str.split(' ')  
 if len(schedule\_exam\_data\_list) != 0:  
 if current\_exam\_date[0] == schedule\_exam\_data\_list[-1]:  
 exam\_count[-1] += 1  
 else:  
 exam\_count.append(count\_ex)  
 count\_ex = 1  
 schedule\_exam\_data\_list.append(current\_exam\_date[0])  
 else:  
 schedule\_exam\_data\_list.append(current\_exam\_date[0])  
 if len(schedule\_cons\_data\_list) != 0:  
 if current\_cons\_date[0] == schedule\_cons\_data\_list[-1]:  
 cons\_count[-1] += 1  
 else:  
 cons\_count.append(count\_cons)  
 count\_cons = 1  
 schedule\_cons\_data\_list.append(current\_cons\_date[0])  
 else:  
 schedule\_cons\_data\_list.append(current\_cons\_date[0])  
  
 # создаем полный лист дат для графика  
 schedule\_data\_list = []  
 i = j = 0  
 while (i < len(schedule\_exam\_data\_list)) and (j < len(schedule\_cons\_data\_list)):  
 if schedule\_exam\_data\_list[i] < schedule\_cons\_data\_list[j]:  
 schedule\_data\_list.append(schedule\_exam\_data\_list[i])  
 i += 1  
 else:  
 schedule\_data\_list.append(schedule\_cons\_data\_list[j])  
 j += 1  
 while i < len(schedule\_exam\_data\_list):  
 schedule\_data\_list.append(schedule\_exam\_data\_list[i])  
 i += 1  
 while j < len(schedule\_cons\_data\_list):  
 schedule\_data\_list.append(schedule\_cons\_data\_list[j])  
 j += 1  
  
 # определяем кол-во консультаций и экзаменов в каждую из дат  
 i = 0  
 for i in range(len(schedule\_data\_list)):  
 if schedule\_data\_list[i] not in schedule\_exam\_data\_list:  
 j = 0  
 for date in schedule\_exam\_data\_list:  
 if schedule\_data\_list[i] < date:  
 schedule\_exam\_data\_list.append('0')  
 k = len(schedule\_exam\_data\_list) - 2  
 while k >= j:  
 schedule\_exam\_data\_list[k + 1] = schedule\_exam\_data\_list[k]  
 k -= 1  
 schedule\_exam\_data\_list[j] = schedule\_data\_list[i]  
 break  
 else:  
 j += 1  
 exam\_count.append(0)  
 k = len(exam\_count) - 2  
 while k >= j:  
 exam\_count[k + 1] = exam\_count[k]  
 k -= 1  
 exam\_count[j] = 0  
 i += 2  
 i = 1  
 for i in range(len(schedule\_data\_list)):  
 if schedule\_data\_list[i] not in schedule\_cons\_data\_list:  
 j = 0  
 for date in schedule\_cons\_data\_list:  
 if schedule\_data\_list[i] < date:  
 schedule\_cons\_data\_list.append('0')  
 k = len(schedule\_cons\_data\_list) - 2  
 while k >= j:  
 schedule\_cons\_data\_list[k + 1] = schedule\_cons\_data\_list[k]  
 k -= 1  
 schedule\_cons\_data\_list[j] = schedule\_data\_list[i]  
 break  
 else:  
 j += 1  
 cons\_count.append(0)  
 k = len(cons\_count) - 2  
 while k >= j:  
 cons\_count[k + 1] = cons\_count[k]  
 k -= 1  
 cons\_count[j] = 0  
 i += 2  
  
 # убираем повторные даты  
 i = 0  
 schedule\_data\_list\_no\_repetitions = []  
 for i in range(len(schedule\_data\_list) - 1):  
 if schedule\_data\_list[i] != schedule\_data\_list[i + 1]:  
 schedule\_data\_list\_no\_repetitions.append(schedule\_data\_list[i])  
 if i == len(schedule\_data\_list) - 2:  
 schedule\_data\_list\_no\_repetitions.append(schedule\_data\_list[i + 1])  
 i += 1  
 continue  
  
 if request.method == "POST":  
 exam.name = request.POST.get("exam\_name")  
 exam.save()  
 name\_group\_change = request.POST.get("group\_name")  
 for group in group\_list:  
 if group.name == name\_group\_change:  
 group\_change = group  
 break  
 for schedule in schedule\_list:  
 if schedule.group == group\_change:  
 schedule.lesson\_time = request.POST.get("lesson\_time")  
 schedule.consultation\_time = request.POST.get("consultation\_time")  
 schedule.save()  
 return HttpResponseRedirect(f"/schedule/{name}")  
 else:  
 return render(request, 'schedule/detail\_lecturer.html', {'group\_list': group\_list,  
 'schedule\_list': schedule\_list,  
 'exam': exam,  
 'schedule\_date\_list': schedule\_data\_list\_no\_repetitions,  
 'exam\_count': exam\_count,  
 'cons\_count': cons\_count})

templates\schedule\index.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Schedule</title>  
  
 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/5.0.0-alpha1/css/bootstrap.min.css" integrity="sha384-r4NyP46KrjDleawBgD5tp8Y7UzmLA05oM1iAEQ17CSuDqnUK2+k9luXQOfXJCJ4I" crossorigin="anonymous">  
 {% load static %}  
 <link rel="stylesheet" type="text/css" href="{% static 'schedule/css/style.css' %}">  
</head>  
  
<body>  
 <nav class="navbar navbar-expand-lg navbar-dark bg-dark">  
 <div class="container-fluid modal-header">  
 <a class="navbar-brand" href="/schedule/">  
<!-- <img src="{% static "schedule/images/BMSTU\_logo.png" %}" alt="BMSTU logo" width="70" height="70" class="img-fluid">-->  
 <h3>Расписание экзаменов</h3>  
 </a>  
 <div class="float-end">  
 <div class="justify-content-center">  
 <table>  
 <tr>  
 <td class="text-center">  
 <img class="img-fluid" src="{% static "schedule/images/BMSTU\_logo.png" %}" alt="BMSTU logo" width="50" height="50">  
 </td>  
 </tr>  
 <tr>  
 <td class="text-center">  
 <p class="text-light">МГТУ им. Н.Э. Баумана</p>  
 </td>  
  
 </tr>  
 </table>  
 </div>  
 </div>  
 </div>  
 </nav>  
  
 <main class="container-fluid bg-dark">  
 <div class="row">  
 <div class="col-2">  
 <div class="card text-white bg-dark">  
 <div class="card-header">  
 <h4>Группы</h4>  
 </div>  
 {% if group\_list %}  
 <ul class="list-group list-group-flush">  
 {% for group in group\_list %}  
 <li class="list-group-item bg-dark">  
 <a class="list" href="{% url 'schedule:detail\_student' group.id %}"><h5>{{ group.name }}</h5></a>  
 </li>  
 {% endfor %}  
 </ul>  
 {% else %}  
 <p>No group are available.</p>  
 {% endif %}  
 </div>  
 </div>  
 <div class="col-3">  
 <div class="card text-white bg-dark">  
 <div class="card-header">  
 <h4>Преподаватели</h4>  
 </div>  
 {% if exam\_list %}  
 <ul class="list-group list-group-flush">  
 {% for exam in exam\_list %}  
 <li class="list-group-item bg-dark">  
 <a class="list" href="{% url 'schedule:detail\_lecturer' exam.lecturer %}"><h5>{{ exam.lecturer }}</h5></a>  
 </li>  
 {% endfor %}  
 </ul>  
 {% else %}  
 <p>No lecturer are available.</p>  
 {% endif %}  
 </div>  
 </div>  
 </div>  
 <div class="container-fluid high">  
  
 </div>  
 </main>  
  
 <footer class="modal-footer bg-dark">  
 <address class="container-fluid">  
 <p style="color: #d3d3d4">Контакты: +7-(922)-724-47-88</p>  
 </address>  
 <p class="container-fluid text-center" style="color: #d3d3d4">2020 BMSTU</p>  
 </footer>  
</body>  
</html>

templates\schedule\detail\_student.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Detail for student</title>  
  
 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/5.0.0-alpha1/css/bootstrap.min.css" integrity="sha384-r4NyP46KrjDleawBgD5tp8Y7UzmLA05oM1iAEQ17CSuDqnUK2+k9luXQOfXJCJ4I" crossorigin="anonymous">  
 <script src="https://kit.fontawesome.com/941547f2e9.js" crossorigin="anonymous"></script>  
 {% load static %}  
 <link rel="stylesheet" type="text/css" href="{% static 'schedule/css/style.css' %}">  
</head>  
<body>  
 <nav class="navbar navbar-expand-lg navbar-dark bg-dark">  
 <div class="container-fluid modal-header">  
 <div>  
 <a class="navbar-brand" href="/schedule/"><i class="fas fa-reply text-light navbar-brand"></i>Назад</a>  
 </div>  
 </div>  
 </nav>  
 <main class="container-fluid bg-dark">  
 <div class="row justify-content-around">  
 <div class="col-6">  
 <h1 class="text-center text-light">{{ group.name }}</h1>  
 <table class="table table-dark table-borderless table-striped">  
 <tr>  
 <th class="text-center">Экзамен</th>  
 <th class="text-center">Консультация</th>  
 <th class="text-center">Экзамен</th>  
 <th class="text-center">Преподаватель</th>  
 </tr>  
 {% for exam in exam\_list %}  
 <tr>  
 <td class="text-center">{{ exam.name }}</td>  
 {% for schedule in schedule\_list %}  
 {% if schedule.lesson == exam %}  
 <td class="text-center"><time datetime="">{{ schedule.consultation\_time|date:'d-m-Y H:i' }}</time></td>  
 <td class="text-center">{{ schedule.lesson\_time|date:'d-m-Y H:i' }}</td>  
 {% endif %}  
 {% endfor %}  
 <td class="text-center">{{ exam.lecturer }}</td>  
 </tr>  
 {% endfor %}  
 </table>  
 </div>  
 </div>  
 </main>  
 <footer class="high bg-dark">  
 </footer>  
</body>  
</html>

templates\schedule\detail\_lecturer.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Detail for lecturer</title>  
  
 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/5.0.0-alpha1/css/bootstrap.min.css" integrity="sha384-r4NyP46KrjDleawBgD5tp8Y7UzmLA05oM1iAEQ17CSuDqnUK2+k9luXQOfXJCJ4I" crossorigin="anonymous">  
 <script src="https://kit.fontawesome.com/941547f2e9.js" crossorigin="anonymous"></script>  
  
 {% load static %}  
 <link rel="stylesheet" type="text/css" href="{% static 'schedule/css/style.css' %}">  
 <link href="{% static 'schedule/css/c3.min.css' %}" rel="stylesheet">  
</head>  
<body>  
 <nav class="navbar navbar-expand-lg navbar-dark bg-dark">  
 <div class="container-fluid modal-header">  
 <div>  
 <a class="navbar-brand" href="/schedule/"><i class="fas fa-reply text-light navbar-brand"></i>Назад</a>  
 </div>  
 <div class="float-end">  
 <!-- Button trigger modal -->  
 <button type="button" class="btn btn-info" data-toggle="modal" data-target="#signUpModal">  
 Внести изменения  
 </button>  
  
 <!-- Modal -->  
 <div class="modal fade" id="signUpModal" tabindex="-1" aria-labelledby="signUpModalLabel" aria-hidden="true">  
 <div class="modal-dialog">  
 <div class="modal-content">  
 <div class="modal-header">  
 <h5 class="modal-title" id="signUpModalLabel">Внести изменения</h5>  
 <button type="button" class="close" data-dismiss="modal" aria-label="Close">  
 <span aria-hidden="true">&times;</span>  
 </button>  
 </div>  
 <div class="modal-body">  
 <form class="g-3 needs-validation signup" method="post" novalidate>  
 {% csrf\_token %}  
 <label for="validationCustom01" class="form-label">Название предмета</label>  
 <input type="text" class="form-control" id="validationCustom01" value="{{ exam.name }}" name="exam\_name" required>  
 <div class="invalid-feedback">  
 Пожалуйста введите название предмета  
 </div>  
 <p></p>  
 <label for="validationCustom02" class="form-label">Группа</label>  
 <select class="form-select" id="validationCustom02" value="{{ group.name }}" name="group\_name" required>  
 <option selected disabled value="">Выберите...</option>  
 {% for i in group\_list %}  
 <option>{{ i }}</option>  
 {% endfor %}  
 </select>  
 <div class="invalid-feedback">  
 Пожалуйста выберите группу  
 </div>  
 <p></p>  
 <label for="validationCustom03" class="form-label">Время консультации</label>  
 <input type="datetime-local" class="form-control" id="validationCustom03" value="{{ schedule.consultation\_time }}" name="consultation\_time" required>  
 <div class="invalid-feedback">  
 Пожалуйста выберите время консультации  
 </div>  
 <p></p>  
 <label for="validationCustom04" class="form-label">Время экзамена</label>  
 <input type="datetime-local" class="form-control" id="validationCustom04" value="{{ schedule.lesson\_time }}" name="lesson\_time" required>  
 <div class="invalid-feedback">  
 Пожалуйста выберите время экзамена  
 </div>  
 <p></p>  
 <div class="modal-footer">  
 <input class = "btn btn-success" type="submit" value="Save" name="\_save">  
 <button class="btn btn-danger" type="button" data-dismiss="modal">Close</button>  
 </div>  
 <script>  
 // JavaScript for disabling form submissions if there are invalid fields  
 (function () {  
 'use strict'  
  
 // Fetch all the forms we want to apply custom Bootstrap validation styles to  
 var forms = ***document***.querySelectorAll('.needs-validation')  
  
 // Loop over them and prevent submission  
 ***Array***.prototype.slice.call(forms)  
 .forEach(function (form) {  
 form.addEventListener('submit', function (event) {  
 if (!form.checkValidity()) {  
 event.preventDefault()  
 event.stopPropagation()  
 }  
  
 form.classList.add('was-validated')  
 }, false)  
 })  
 })()  
 </script>  
 </form>  
 </div>  
 </div>  
 </div>  
 </div>  
 </div>  
 </div>  
 </nav>  
 <main class="container-fluid bg-dark">  
 <div class="row">  
 <div class="col-6">  
 <h1 class="text-light">{{ exam.name }}</h1>  
 <h2 class="text-light">{{ exam.lecturer }}</h2>  
 <table class="table table-dark table-borderless table-striped">  
 <tr>  
 <th class="text-center">Группа</th>  
 <th class="text-center">Консультация</th>  
 <th class="text-center">Экзамен</th>  
 </tr>  
 {% for group in group\_list %}  
 <tr>  
 <td class="text-center">{{ group.name }}</td>  
 {% for schedule in schedule\_list %}  
 {% if schedule.group == group %}  
 <td class="text-center">{{ schedule.consultation\_time|date:'d-m-Y H:i' }}</td>  
 <td class="text-center">{{ schedule.lesson\_time|date:'d-m-Y H:i' }}</td>  
 {% endif %}  
 {% endfor %}  
 </tr>  
 {% endfor %}  
 </table>  
 </div>  
 <div class="col-5" id="chart"></div>  
 </div>  
 </div>  
 </main>  
 <footer class="high bg-dark">  
 </footer>  
  
 <script src="https://d3js.org/d3.v5.min.js"></script>  
 <script src="{% static 'schedule/js/c3.min.js' %}"></script>  
  
 <script>  
 var ***chart*** = c3.generate({  
 data: {  
 size: {  
 width: 400  
 },  
 x: 'x',  
 columns: [  
 ['x', "{{ schedule\_date\_list.0 }}", "{{ schedule\_date\_list.1 }}", "{{ schedule\_date\_list.2 }}", "{{ schedule\_date\_list.3 }}", "{{ schedule\_date\_list.4 }}", "{{ schedule\_date\_list.5 }}", "{{ schedule\_date\_list.6 }}", "{{ schedule\_date\_list.7 }}", "{{ schedule\_date\_list.8 }}", "{{ schedule\_date\_list.9 }}"],  
 ['Количество экзаменов', "{{ exam\_count.0 }}", "{{ exam\_count.1 }}", "{{ exam\_count.2 }}", "{{ exam\_count.3 }}", "{{ exam\_count.4 }}", "{{ exam\_count.5 }}", "{{ exam\_count.6 }}", "{{ exam\_count.7 }}", "{{ exam\_count.8 }}", "{{ exam\_count.9 }}"],  
 ['Количество консультаций', "{{ cons\_count.0 }}", "{{ cons\_count.1 }}", "{{ cons\_count.2 }}", "{{ cons\_count.3 }}", "{{ cons\_count.4 }}", "{{ cons\_count.5 }}", "{{ cons\_count.6 }}", "{{ cons\_count.7 }}", "{{ cons\_count.8 }}", "{{ cons\_count.9 }}"]  
 ],  
 type: 'bar',  
 groups: [  
 ['Количество экзаменов', 'Количество консультаций']  
 ]  
 },  
 bar: {  
 width: {  
 ratio: 0.5  
 }  
 },  
 axis: {  
 x: {  
 type: 'timeseries',  
 tick: {  
 format: '%d-%m'  
 }  
 },  
 y: {  
 max: 3,  
 min: 0,  
 }  
 },  
 color: {  
 pattern: ['#ca4141', '#40bcbc']  
 }  
 });  
 </script>  
  
 <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>  
 <script src="https://stackpath.bootstrapcdn.com/bootstrap/5.0.0-alpha1/js/bootstrap.min.js" integrity="sha384-oesi62hOLfzrys4LxRF63OJCXdXDipiYWBnvTl9Y9/TRlw5xlKIEHpNyvvDShgf/" crossorigin="anonymous"></script>  
</body>  
</html>

static\schedule\style.css

1. .img-fluid {  
    max-width: 100%;  
    height: auto;  
   }  
   .text-center {  
    text-align: center;  
   }  
   .text-light {  
    color: aliceblue;  
   }  
   .td-text-short {  
    width: 50%;  
   }  
   .high {  
    height: 700px;  
   }  
   .list {  
    color: aliceblue; /\* Цвет обычной ссылки \*/  
    text-decoration: none; /\* Убираем подчеркивание у ссылок \*/  
    }  
   a:hover {  
    color: aqua; /\* Цвет ссылки при наведении на нее курсора мыши \*/  
    text-decoration: none; /\* Добавляем подчеркивание \*/  
   }  
   li {  
    list-style-type: none;  
    }  
   ul {  
    margin-left: 0;  
    padding-left: 0;  
   }  
   .logo {  
    width: 30px;  
    height: 30px;  
   }  
   .navbar-logo {  
    font-size: 2em;  
    color: aliceblue; /\* Цвет обычной ссылки \*/  
    text-decoration: none; /\* Убираем подчеркивание у ссылок \*/  
   }  
   .navbar-logo:hover {  
    color: aliceblue; /\* Цвет ссылки при наведении на нее курсора мыши \*/  
    text-decoration: none; /\* Добавляем подчеркивание \*/  
   }  
   img {  
    align-content: center;  
   }
2. **Экранные формы с примерами выполнения программы**



















