Московский государственный технический университет им. Н. Э. Баумана

Факультет «Информатика и системы управления»

| Кафедра ИУ5. Курс «Разработка интернет-приложени | 1Й» |
|--|-----|
| Отчет по домашнему заданию | |
| Вариант 3. | |

| Выполнил: | Ірове | рил: |
|-----------|-------|------|
|-----------|-------|------|

Студент группы ИУ5-51Б Гапанюк Ю. Е.

Ноздрова Валентина

Дата: 27.12.2021 Дата:

Подпись:

Цель:

Изучение возможностей создания прототипа веб-приложения на основе базы данных с использованием фреймворка Django.

Задание:

На основе результатов лабораторных работ 4 и 5, создайте прототип веб-приложения с использованием фреймворка Django на основе базы данных, реализующий концепцию master/detail. Прототип должен содержать:

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

Листинг программы:

```
-----models.py------
from django.db import models
# Create your models here.
class Service(models.Model):
   class Meta:
       verbose name = ' услугу'
       verbose name plural = 'Услуги'
   service name = models.CharField(max length=100, verbose name='Наименование услуги')
   price = models.DecimalField(max_digits=8, decimal_places=2,
verbose name='Стоимость')
   image = models.TextField(verbose name='Изображение', null=True)
   description = models.TextField(verbose name='Описание', null=True)
   last update = models.DateTimeField(auto now=True, verbose name='Дата последнего
обновления ')
class Appointment(models.Model):
   phone = models.CharField(max length=12, verbose name='Телефон')
class Doctor(models.Model):
   class Meta:
       verbose name = ' врача'
       verbose_name_plural = 'Врачи'
   doc_name = models.CharField(max_length=60, verbose name='ΦΜΟ')
   doc spec = models.CharField(max length=60, verbose name='Специальность',
default=None)
   def __str__(self):
       return self.doc name
days = ((1,'\Pi H'), (2,'BT'), (3,'CP'), (4,'\Psi T'), (5,'\Pi T'), (6,'CE'))
class Schedule(models.Model):
   class Meta:
       verbose name = 'расписание'
```

```
unique together = ('doc', 'day')
   doc = models.ForeignKey(Doctor, on delete=models.CASCADE, verbose name='Bpau')
   day = models.PositiveSmallIntegerField(verbose name='День недели', choices=days)
   sch = models.TextField(verbose name='Расписание', null=True)
   def display doc name(self):
       return self.doc.doc name
   display_doc_name.short_description = 'ΦИО'
   def display doc spec(self):
       return self.doc.doc spec
   display_doc_spec.short_description = 'Специальность'
from Vet.models import Service
from Vet.models import Appointment
from Vet.models import Doctor
from Vet.models import Schedule
from rest framework import serializers
class ServiceSerializer(serializers.ModelSerializer):
   class Meta:
       # Модель, которую мы сериализуем
       model = Service
       # Поля, которые мы сериализуем
       fields = ["id", "service name", "price", "last update", "image", "description"]
class AppointmentSerializer(serializers.ModelSerializer):
   class Meta:
       # Модель, которую мы сериализуем
       model = Appointment
       # Поля, которые мы сериализуем
       fields = ["id", "phone"]
class DoctorSerializer(serializers.ModelSerializer):
   class Meta:
       model = Doctor
       fields = ['id', 'doc name', 'doc spec']
class ScheduleSerializer(serializers.ModelSerializer):
   class Meta:
       model = Schedule
       fields = ['id', 'doc id', 'day', 'sch']
-----views.py------
from django.shortcuts import render
# Create your views here.
from rest framework import viewsets
from Vet.serializers import ServiceSerializer
from Vet.models import Service
from Vet.serializers import AppointmentSerializer
from Vet.models import Appointment
from Vet.serializers import DoctorSerializer
from Vet.models import Doctor
from Vet.serializers import ScheduleSerializer
```

verbose name plural = 'Расписание'

```
class ServiceViewSet(viewsets.ModelViewSet):
   API endpoint, который позволяет просматривать и редактировать услуги клиники
    # queryset всех пользователей для фильтрации по дате последнего изменения
   queryset = Service.objects.all().order by('id')
   serializer class = ServiceSerializer # Сериализатор для модели
class AppointmentViewSet(viewsets.ModelViewSet):
   API endpoint, который позволяет просматривать телефонные номера клиентов
    # queryset всех пользователей для фильтрации по дате последнего изменения
   queryset = Appointment.objects.all().order by('id')
   serializer class = AppointmentSerializer # Сериализатор для модели
class DoctorViewSet(viewsets.ModelViewSet):
   API endpoint, который позволяет просматривать информацию о врачах
    # queryset всех пользователей для фильтрации по дате последнего изменения
   queryset = Doctor.objects.all().order by('id')
   serializer class = DoctorSerializer # Сериализатор для модели
class ScheduleViewSet(viewsets.ModelViewSet):
   API endpoint, который позволяет просматривать расписание врачей
   # queryset всех пользователей для фильтрации по дате последнего изменения
   queryset = Schedule.objects.all().order by('id')
   serializer_class = ScheduleSerializer # Сериализатор для модели
-----urls.py-----
from django.contrib import admin
from django.urls import include, path
from Vet import views as vetviews
from rest framework import routers
from admin tools import urls
router = routers.DefaultRouter()
router.register(r'services', vetviews.ServiceViewSet)
router.register(r'appointments', vetviews.AppointmentViewSet)
router.register(r'doctors', vetviews.DoctorViewSet)
router.register(r'schedule', vetviews.ScheduleViewSet)
urlpatterns = [
   path('', include(router.urls)),
   path('api-auth/', include('rest framework.urls', namespace='rest framework')),
   path('admin/', admin.site.urls),
1
-----admin.py------
from django.contrib import admin
from Vet.models import Service, Doctor, Schedule
from django.utils.html import format html
from django.forms import TextInput, Textarea
from django.db import models
```

```
from django import forms
# Register your models here.
admin.AdminSite.site header = 'Администрирование Vetclinic'
class ServiceForm(forms.ModelForm):
   class Meta:
       widgets = {
           'image': TextInput(attrs={'size': '20'}),
class ServiceAdmin(admin.ModelAdmin):
   list display = ['service name', 'price', 'description', 'image', 'show image']
   list display links = ['service name']
   list editable = ['price', 'image']
   search_fields = ['service']
   form = ServiceForm
   def show image(self, obj):
       return format html('<image src="{}" style="height:200px" />', obj.image)
   show image.short description = 'Превью'
admin.site.register(Service, ServiceAdmin)
class ScheduleInline(admin.TabularInline):
   model = Schedule
   extra = 0
class DoctorAdmin(admin.ModelAdmin):
    list display = ['doc name', 'doc spec']
   list filter=['doc spec']
   search fields = ['doc name', 'doc spec']
    inlines = [ScheduleInline]
admin.site.register(Doctor, DoctorAdmin)
class ScheduleAdmin(admin.ModelAdmin):
   list_display = ('display_doc_name', 'display_doc_spec', 'day', 'sch')
   list filter = ['day']
   search_fields = ['doc__doc_name', 'doc__doc_spec']
admin.site.register(Schedule, ScheduleAdmin)
-----Schedule.js-----
import {Breadcrumb, BreadcrumbItem} from "react-bootstrap";
import React from "react";
import SchTable from "../components/ScheduleComponent";
import BackButton from "../components/Button";
function Schedule() {
   return (
       <div className='page'>
           <Breadcrumb className='bdcrumb'>
               <BreadcrumbItem href='/'>Главная</BreadcrumbItem>
               <BreadcrumbItem active>Pacписание
           </Breadcrumb>
           <SchTable/>
           <BackButton backUrl='http://localhost:3000/' buttonTitle='Вернуться'
active/>
       </div>
   )
}
```

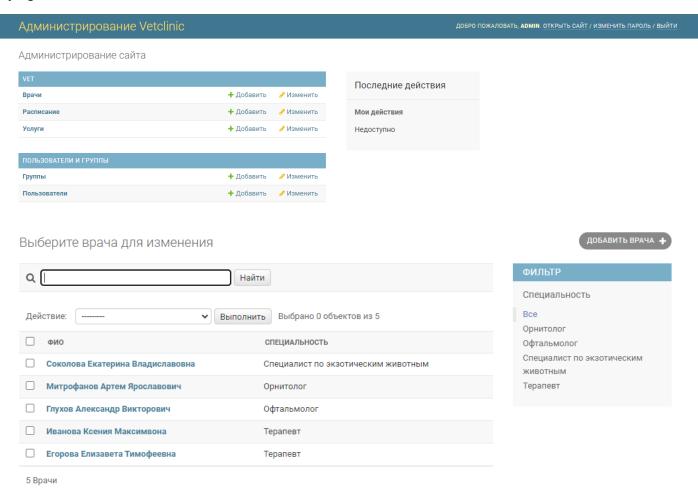
```
export default Schedule;
```

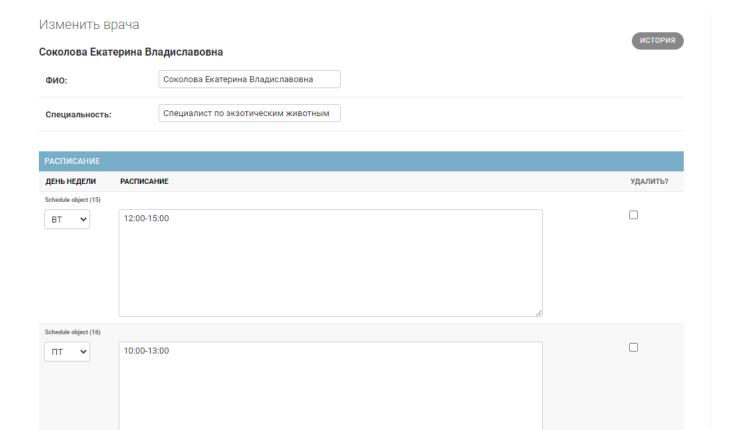
```
-----ScheduleComponent.js-----
import React, {useEffect, useState} from "react";
import {Spinner, Table} from "react-bootstrap";
function SchTable() {
   const [error, setError] = useState(null);
   const [isLoaded, setIsLoaded] = useState(false);
   const [data, setData] = useState([]);
   const [docs, setDocs] = useState([]);
   useEffect(() => {
       fetch('http://localhost:8000/doctors/')
           .then(res => res.json())
           .then(
               (result) => {
                  setIsLoaded(true);
                  setDocs(result);
               },
               (error) => {
                  setIsLoaded(true);
                  setError(error);
               }
           )
   }, [])
   useEffect(() => {
       fetch('http://localhost:8000/schedule/')
           .then(res => res.json())
           .then(
               (result) => {
                  setIsLoaded(true);
                  setData(result);
               (error) => {
                  setIsLoaded(true);
                  setError(error);
               }
           )
   }, [])
   const sch = docs.map(item => (data.filter(i => (i.doc id === item.id))))
   const x = (id, day) \Rightarrow (sch[id-1].find(i \Rightarrow i.day===day))
   if (error) {
       return <div>Ошибка: {error.message}</div>;
    } else if (!isLoaded) {
       return <div className="loadingBg" align='center'><Spinner</pre>
animation="border"/></div>;
   } else {
       return (
           <Table hover>
               <thead>
               >
                   Bpau
                  Cпециальность
                  <th>\PiH
                  BT
                  CP
                  YT
                   <th>\PiT</th>
                   CE
```

```
</thead>
            {docs.map(item => (
               {item.doc name}
                  {item.doc spec}
                  {Array.from(Array(6).keys()).map(i =>
                      ({x(item.id, i+1) && <p}{x(item.id, i+1).sch}</p>}
                   \{!x(item.id, i+1) && -\}))}
               ))}
            </Table>
      );
   }
}
export default SchTable;
```

Результаты выполнения:

Django-admin





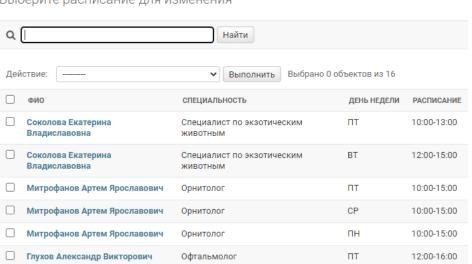
Выберите расписание для изменения

Глухов Александр Викторович

Глухов Александр Викторович

Плухов Александр Викторович

Плухов Александр Викторович



Офтальмолог

Офтальмолог

Офтальмолог

Офтальмолог



12:00-16:00

12:00-16:00

12:00-16:00

12:00-16:00

ЧΤ

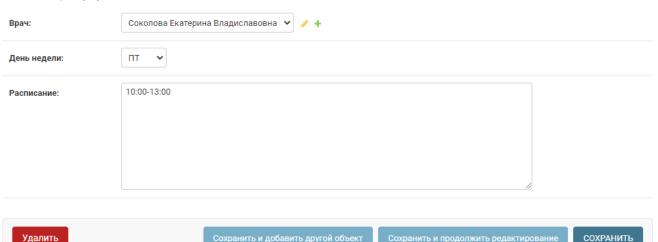
CP

BT

ΠН

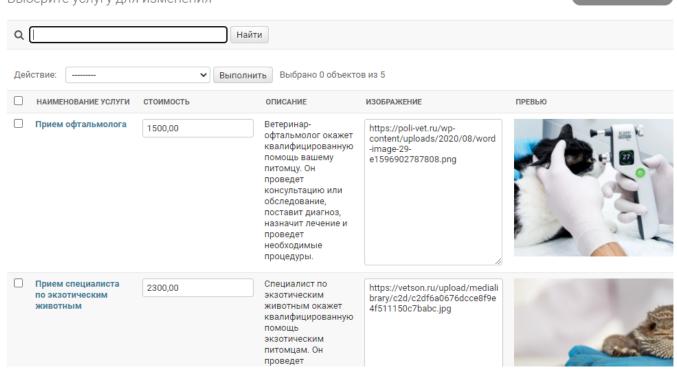
Изменить расписание

Schedule object (16)



Выберите услугу для изменения





Изменить услугу

Service object (8)

| Наименование услуги: | Прием офтальмолога |
|----------------------|---|
| Стоимость: | 1500,00 |
| Изображение: | https://poli-vet.ru/wp-con |
| Описание: | Ветеринар-офтальмолог окажет квалифицированную помощь вашему питомцу. Он проведет консультацию или обследование, поставит диагноз, назначит лечение и проведет необходимые процедуры. |
| | |
| Удалить | Сохранить и добавить другой объект Сохранить и продолжить редактирование СОХРАНИТЬ |

история

Отчет по двум таблицам



Vetclinic

Главная / Расписание

| Врач | Специальность | пн | ВТ | СР | ЧТ | ПТ | СБ |
|----------------------------------|-------------------------------------|-------------|-------------|-------------|-------------|-------------|------------|
| Егорова Елизавета Тимофеевна | Терапевт | 9:00-17:00 | - | - | 9:00-17:00 | 9:00-17:00 | - |
| Иванова Ксения Максимвона | Терапевт | - | 9:00-17:00 | 9:00-17:00 | - | - | 9:00-17:00 |
| Глухов Александр Викторович | Офтальмолог | 12:00-16:00 | 12:00-16:00 | 12:00-16:00 | 12:00-16:00 | 12:00-16:00 | - |
| Митрофанов Артем Ярославович | Орнитолог | 10:00-15:00 | - | 10:00-15:00 | - | 10:00-15:00 | - |
| Соколова Екатерина Владиславовна | Специалист по экзотическим животным | - | 12:00-15:00 | - | - | 10:00-13:00 | - |

Вернуться