

**Московский государственный технический университет
им. Н. Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5. Курс «Разработка интернет-приложений»

Отчет по домашнему заданию

Вариант 3.

Выполнил:

Студент группы ИУ5-51Б

Ноздрова Валентина

Проверил:

Гапанюк Ю. Е.

Дата: 27.12.2021

Дата:

Подпись:

Подпись:

Москва, 2021 г.

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 фреймворк по желанию студента.

Листинг программы:

```
-----models.py-----

from django.db import models

# Create your models here.
class Service(models.Model):
    class Meta:
        verbose_name = 'услуги'
        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='ФИО')
    doc_spec = models.CharField(max_length=60, verbose_name='Специальность',
    default=None)

    def __str__(self):
        return self.doc_name

days = ((1, 'ПН'), (2, 'ВТ'), (3, 'СР'), (4, 'ЧТ'), (5, 'ПТ'), (6, 'СБ'))

class Schedule(models.Model):
    class Meta:
        verbose_name = 'расписание'
```

```
verbose_name_plural = 'Расписание'
unique_together = ('doc', 'day')
doc = models.ForeignKey(Doctor, on_delete=models.CASCADE, verbose_name='Врач')
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 = 'ФИО'

def display_doc_spec(self):
    return self.doc.doc_spec
display_doc_spec.short_description = 'Специальность'
```

-----serializers.py-----

```
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
```

```
from Vet.models import Schedule
```

```
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),  
]
```

-----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>Расписание</BreadcrumbItem>
            </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 [isLoading, 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) => (sch[id-1].find(i => i.day===day))

  if (error) {
    return <div>Ошибка: {error.message}</div>;
  } else if (!isLoading) {
    return <div className="loadingBg" align='center'><Spinner
animation="border"/></div>;
  } else {
    return (
      <Table hover>
        <thead>
          <tr>
            <th>Врач</th>
            <th>Специальность</th>
            <th>ПН</th>
            <th>ВТ</th>
            <th>СР</th>
            <th>ЧТ</th>
            <th>ПТ</th>
            <th>СБ</th>
          </tr>

```

```

    </thead>
    <tbody>
    {docs.map(item => (
      <tr key={item.id}>
        <td>{item.doc_name}</td>
        <td>{item.doc_spec}</td>
        {Array.from(Array(6).keys()).map(i =>
          (<td>{x(item.id, i+1) && <p>{x(item.id, i+1).sch}</p>}
            {!x(item.id, i+1) && <p>-</p></td>)))}
        </tr>
      ) ) }
    </tbody>
  </Table>
);
}
//
export default SchTable;
```

Результаты выполнения:

Django-admin

Администрирование Vetclinic

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

Администрирование сайта

VET

Врачи

+ Добавить

✎ Изменить

Расписание

+ Добавить

✎ Изменить

Услуги

+ Добавить

✎ Изменить

ПОЛЬЗОВАТЕЛИ И ГРУППЫ

Группы

+ Добавить

✎ Изменить

Пользователи

+ Добавить

✎ Изменить

Последние действия

Мои действия

Недоступно

Выберите врача для изменения

ДОБАВИТЬ ВРАЧА +

Q

Найти

Действие:

Выполнить

Выбрано 0 объектов из 5

☐

ФИО

СПЕЦИАЛЬНОСТЬ

☐

Соколова Екатерина Владиславовна

Специалист по экзотическим животным

☐

Митрофанов Артем Ярославович

Орнитолог

☐

Глухов Александр Викторович

Офтальмолог

☐

Иванова Ксения Максимовна

Терапевт

☐

Егорова Елизавета Тимофеевна

Терапевт

5 Врачи

ФИЛЬТР

Специальность

Все

Орнитолог

Офтальмолог

Специалист по экзотическим животным

Терапевт

Соколова Екатерина Владиславовна

ИСТОРИЯ

ФИО:

Соколова Екатерина Владиславовна

Специальность:

Специалист по экзотическим животным

РАСПИСАНИЕ

ДЕНЬ НЕДЕЛИ	РАСПИСАНИЕ	УДАЛИТЬ?
Schedule object (15)		
ВТ	12:00-15:00	<input type="checkbox"/>
Schedule object (16)		
ПТ	10:00-13:00	<input type="checkbox"/>

Выберите расписание для изменения

ДОБАВИТЬ РАСПИСАНИЕ +

🔍

Найти

Действие:

Выполнить

Выбрано 0 объектов из 16

<input type="checkbox"/>	ФИО	СПЕЦИАЛЬНОСТЬ	ДЕНЬ НЕДЕЛИ	РАСПИСАНИЕ
<input type="checkbox"/>	Соколова Екатерина Владиславовна	Специалист по экзотическим животным	ПТ	10:00-13:00
<input type="checkbox"/>	Соколова Екатерина Владиславовна	Специалист по экзотическим животным	ВТ	12:00-15:00
<input type="checkbox"/>	Митрофанов Артем Ярославович	Орнитолог	ПТ	10:00-15:00
<input type="checkbox"/>	Митрофанов Артем Ярославович	Орнитолог	СР	10:00-15:00
<input type="checkbox"/>	Митрофанов Артем Ярославович	Орнитолог	ПН	10:00-15:00
<input type="checkbox"/>	Глухов Александр Викторович	Офтальмолог	ПТ	12:00-16:00
<input type="checkbox"/>	Глухов Александр Викторович	Офтальмолог	ЧТ	12:00-16:00
<input type="checkbox"/>	Глухов Александр Викторович	Офтальмолог	СР	12:00-16:00
<input type="checkbox"/>	Глухов Александр Викторович	Офтальмолог	ВТ	12:00-16:00
<input type="checkbox"/>	Глухов Александр Викторович	Офтальмолог	ПН	12:00-16:00

ФИЛЬТР

День недели

Все

ПН

ВТ



СР


ЧТ

ПТ

СБ

Schedule object (16)

Врач: Соколова Екатерина Владиславовна  

День недели: ПТ 

Расписание:

10:00-13:00


Удалить

Сохранить и добавить другой объект


Сохранить и продолжить редактирование



СОХРАНИТЬ

Выберите услугу для изменения

ДОБАВИТЬ УСЛУГУ 

Найти

Действие: -----  Выполнить Выбрано 0 объектов из 5

<input type="checkbox"/>	НАИМЕНОВАНИЕ УСЛУГИ	СТОИМОСТЬ	ОПИСАНИЕ	ИЗОБРАЖЕНИЕ	ПРЕВЬЮ
<input type="checkbox"/>	Прием офтальмолога	<div>1500,00</div>	Ветеринар-офтальмолог окажет квалифицированную помощь вашему питомцу. Он проведет консультацию или обследование, поставит диагноз, назначит лечение и проведет необходимые процедуры.	<div>https://poli-vet.ru/wp-content/uploads/2020/08/word-image-29-e1596902787808.png</div>	
<input type="checkbox"/>	Прием специалиста по экзотическим животным	<div>2300,00</div>	Специалист по экзотическим животным окажет квалифицированную помощь экзотическим питомцам. Он проведет	<div>https://vetson.ru/upload/medialibrary/c2d/c2df6a0676dcce8f9e4f511150c7babc.jpg</div>	

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

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 Максимовна	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