Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Московский государственный технический университет имени Н.Э. Баумана»

(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления (ИУ5)

О т ч е т

по домашнему заданию

Создание прототипа веб-приложения на основе базы данных с использованием фреймворка Django

Дисциплина: Разработка Интернет-Приложений

Студент гр. ИУ5-53Б _____ Терентьев В.О.
(Подпись, дата) (Фамилия И.О.)

Преподаватель _____ Гапанюк Ю.Е.
(Подпись, дата) (Фамилия И.О.)

Москва, 2020

1. Описание задания

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.

Расширенные задания:

1. Реализация домашнего задания с использованием фреймворка для разработки SPA-приложений (React, Angualar, ...).
2. Реализация связи много-ко-многим (с возможностью редактирования данных в пользовательском интерфейсе) и развертывание приложения на облачном сервисе (например, heroku).
3. Реализация в отчете (пункт 3 стандартного задания) графика на основе данных отчета с использованием библиотек JavaScript (например, https://c3js.org/) и развертывание приложения на облачном сервисе (например, heroku).
4. Подготовка черновика статьи по тематике курса (тематика статьи согласовывается с преподавателем).

5. Текст программы

5.1. mysite\models.py

```
from django.db import models

# Create your models here.


class Course(models.Model):
    name = models.CharField('Название', max_length=100)

    class Meta:
        verbose_name = 'Курс'
        verbose_name_plural = 'Курсы'

    def __str__(self):
        return self.name


class Course_Lab(models.Model):
    course = models.ForeignKey(Course, on_delete=models.CASCADE)
    name = models.CharField('Название', max_length=100)
    task = models.TextField('Задание')

    class Meta:
```

```
        verbose_name = 'Лабораторная работа курса'
        verbose_name_plural = 'Лабораторные работы курса'

    def __str__(self):
        return self.course.name + " " + self.name


class Student(models.Model):
    surname = models.CharField('Фамилия', max_length=25)
    name = models.CharField('Имя', max_length=25)
    patronymic = models.CharField(
        'Отчество', max_length=25, blank=True, null=True)
    course = models.PositiveSmallIntegerField('Курс')
    group = models.CharField('Группа', max_length=10)
    labs = models.ManyToManyField(Course_Lab, through='Student_Lab_Course')

    class Meta:
        verbose_name = 'Студент'
        verbose_name_plural = 'Студенты'

    def __str__(self):
        return self.group + " " + self.surname + " " + self.name + " " + self.pat
ronymic


class Student_Lab_Course(models.Model):
    student = models.ForeignKey(Student, on_delete=models.CASCADE)
    course_lab = models.ForeignKey(Course_Lab, on_delete=models.CASCADE)
    report = models.FileField('Отчет', blank=True, null=True)
    issued = models.DateTimeField('Выдана', auto_now_add=True)
    completed = models.DateTimeField('Выполнена', blank=True, null=True)
    changed = models.DateTimeField('Изменена', auto_now=True, null=True)

    class Meta:
        verbose_name = 'Лабораторная работа студента'
        verbose_name_plural = 'Лабораторные работы студентов'

    def __str__(self):
        return self.course_lab.name + " " + self.student.surname + " " + self.stu
dent.name + " " + self.student.patronymic
```

5.2. mysite\api\serializers.py

```
from rest_framework import serializers

from ..models import Student, Student_Lab_Course, Course_Lab, Course


class StudentSerializer(serializers.ModelSerializer):

    class Meta:
```

```
        model = Student
        fields = '__all__'


class StudentDetailSerializer(serializers.ModelSerializer):

    labs = serializers.SerializerMethodField()

    class Meta:
        model = Student
        fields = '__all__'

    @staticmethod
    def get_labs(obj):
        return Student_Lab_CourseSerializer(Student_Lab_Course.objects.filter(stu
dent=obj), many=True).data


class TempCourseSerializer(serializers.ModelSerializer):

    class Meta:
        model = Course
        fields = '__all__'


class TempStudent_Lab_CourseSerializer(serializers.ModelSerializer):

    student = StudentSerializer()

    class Meta:
        model = Student_Lab_Course
        fields = '__all__'


class Course_LabSerializer(serializers.ModelSerializer):

    course = TempCourseSerializer()
    labs = serializers.SerializerMethodField()

    class Meta:
        model = Course_Lab
        fields = '__all__'

    @staticmethod
    def get_labs(obj):
        return TempStudent_Lab_CourseSerializer(Student_Lab_Course.objects.filter
(course_lab=obj), many=True).data


class CourseSerializer(serializers.ModelSerializer):
```

```
    course_labs = serializers.SerializerMethodField()

    class Meta:
        model = Course
        fields = '__all__'

    @staticmethod
    def get_course_labs(obj):
        return Course_LabSerializer(Course_Lab.objects.filter(course=obj), many=T
rue).data


class Student_Lab_CourseSerializer(serializers.ModelSerializer):

    course_lab = Course_LabSerializer()

    class Meta:
        model = Student_Lab_Course
        fields = '__all__'
```

5.3. mysite\api\views.py

```
from rest_framework import viewsets

from .serializers import StudentSerializer, Student_Lab_CourseSerializer, Student
DetailSerializer, CourseSerializer, Course_LabSerializer
from ..models import Student, Student_Lab_Course, Course, Course_Lab


class StudentsViewSet(viewsets.ModelViewSet):
    queryset = Student.objects.all()
    serializer_class = StudentSerializer

    action_to_serializer = {
        "retrieve": StudentDetailSerializer
    }

    def get_serializer_class(self):
        return self.action_to_serializer.get(
            self.action,
            self.serializer_class
        )


class Student_Lab_CourseViewSet(viewsets.ModelViewSet):
    queryset = Student_Lab_Course.objects.all()
    serializer_class = Student_Lab_CourseSerializer


class CoursesViewSet(viewsets.ModelViewSet):
    queryset = Course.objects.all()
```

```
    serializer_class = CourseSerializer


class CoursesLabsViewSet(viewsets.ModelViewSet):
    queryset = Course_Lab.objects.all()
    serializer_class = Course_LabSerializer
```

5.4. mysite-ui\src\App.js

```
import React from 'react';
import './App.css';
import 'bootstrap/dist/css/bootstrap.min.css';
import Navbar from "./components/Navigation/Navbar";
import Students from "./components/Student/Students";
import Courses from "./components/Courses/Courses";
import StudentDetail from "./components/Student/StudentDetail";
import CourseLabs from "./components/Courses/CourseLabs";
import LabDetail from "./components/Labs/LabDetail";
import { BrowserRouter as Router, Switch, Route } from 'react-router-dom';
import CourseLabDetail from './components/Courses/CourseLabDetail';
import StudentsEdit from "./components/Student/StudentsEdit";
import Report from './components/Report/Report';

function App() {

  return (
    <div className="App">
      <Router>
        <Navbar />
        <Switch>
          <Route path="/students/" exact component={Students} />
          <Route path="/courses/" exact component={Courses} />
          <Route path="/students-edit/" exact component={StudentsEdit} />
          <Route path="/report/" exact component={Report} />
          <Switch>
            <Route path="/labs/:id" exact component={LabDetail} />
            <Route path="/students/:id" exact component={StudentDetail} />
            <Route path="/courses/:id" exact component={CourseLabs} />
            <Route path="/courses-labs/:id" exact component={CourseLabDetail} />
          </Switch>
        </Switch>

      </Router>
    </div>
  );
}

export default App;
```

5.5. mysite-ui\src\components\Navigation\Navbar.js

```
import React from 'react';
import 'bootstrap/dist/css/bootstrap.min.css';
import { Link } from 'react-router-dom';

function Navbar() {

    return (
        <div className="App">
            <nav class="navbar navbar-expand-lg navbar-light bg-light">
                <div class="container-fluid">
                    <a class="navbar-brand" href="/">Мой сайт</a>
                    <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-
expanded="false" aria-label="Toggle navigation">
                        <span class="navbar-toggler-icon"></span>
                    </button>
                    <div class="collapse navbar-collapse" id="navbarNav">
                        <ul class="navbar-nav">
                            <li class="nav-item">
                                <a class="nav-link active" aria-
current="page" href="/">Главная</a>
                            </li>
                            <li class="nav-item">
                                <Link class="nav-
link" to={{ pathname: `/students/`, fromDashboard: false }}>Студенты</Link>
                            </li>
                            <li class="nav-item">
                                <Link class="nav-
link" to={{ pathname: `/courses/`, fromDashboard: false }}>Курсы</Link>
                            </li>
                            <li class="nav-item">
                                <Link class="nav-
link" to={{ pathname: `/report/`, fromDashboard: false }}>Отчет</Link>
                            </li>
                        </ul>
                    </div>
                </div>
            </nav>
        </div>
    );
}

export default Navbar;
```

5.6. mysite-ui\src\components\Student\StudentDetail.js

```
import React, { useEffect, useState } from 'react';
import axios from 'axios';
import { Link } from 'react-router-dom';
```

```
function StudentDetail({ match }) {

    const [student, setStudent] = useState({})
    const [labs, setLabs] = useState([])
    const id = match.params.id

    useEffect(() => {
        axios({
            method: "GET",
            url: `https://h0mew0rk.herokuapp.com/api/students/${id}/`
        }).then(response => {
            setStudent(response.data)
            setLabs(response.data.labs)
        })
    }, [id])

    return (
        <div>
            <hr />
            <p>Студент: <strong>{student.surname} {student.name} {student.patrony
mic} </strong>{student.group}</p>
            <hr />
            {labs.map(l => (
                <div className="card" key={l.id}>
                    <div class="card-body">
                        <h5 class="card-title">{l.course_lab.name}</h5>
                        <h6 class="card-subtitle mb-2 text-
muted">Курс: {l.course_lab.course.name}</h6>
                        <p class="card-text"><small class="text-
muted">Выполнена: {l.completed}</small></p>
                        <Link class="btn btn-
primary" to={{ pathname: `/labs/${l.id}`, fromDashboard: false }}>Детали</Link>
                    </div>
                </div>
            ))}
        </div>
    );
}

export default StudentDetail;
```

5.7. mysite-ui\src\components\Student\StudentsEdit.js

```
import React from 'react';
import 'bootstrap/dist/css/bootstrap.min.css';
import { Formik, Form, Field } from 'formik';
import axios from 'axios';
import getCookie from 'js-cookie'

axios.defaults.xsrfCookieName = 'csrftoken'
axios.defaults.xsrfHeaderName = 'X-CSRFToken'
```

```
const csrftoken = getCookie('csrftoken');

function Students() {
    return (
        <div>
            <h4>Добавление студента</h4>
            <Formik
                initialValues={{ surname: '', name: '', patronymic: '', course: 1
, group: '' }}
                validate={() => { }}
                onSubmit={values => {
                    axios({
                        method: 'post',
                        url: 'https://h0mew0rk.herokuapp.com/api/students/',
                        data: values
                    })
                }}
            >
                <Form>
                    <div class="mb-3">
                        <label class="form-label" for="surname">Фамилия:</label>
                        <br />
                        <Field type="text" name="surname" />
                    </div>
                    <div class="mb-3">
                        <label class="form-label">Имя</label>
                        <br />
                        <Field type="text" name="name" />
                    </div>
                    <div class="mb-3">
                        <label class="form-
label" for="patronymic">Отчество:</label>
                        <br />
                        <Field type="text" name="patronymic" />
                    </div>
                    <div class="mb-3">
                        <label class="form-label" for="course">Курс:</label>
                        <br />
                        <Field type="number" name="course" />
                    </div>
                    <div class="mb-3">
                        <label class="form-label" for="group">Группа:</label>
                        <br />
                        <Field type="text" name="group" />
                    </div>
                    <button class="btn btn-
primary" type="submit" >Добавить</button>
                </Form>
            </Formik>
        </div>
```

```
    );
}

export default Students;
```

5.8. mysite-ui\src\components\Labs\LabDetail.js

```
import React, { useEffect, useState } from 'react';
import axios from 'axios';
import getCookie from 'js-cookie'

axios.defaults.xsrfCookieName = 'csrftoken'
axios.defaults.xsrfHeaderName = 'X-CSRFToken'

const csrftoken = getCookie('csrftoken');

function LabDetail({ match }) {

    const [lab, setLab] = useState({})
    const [course_lab, setCourse_lab] = useState([])
    const [course, setCourse] = useState([])
    const id = match.params.id

    useEffect(() => {
        axios({
            method: "GET",
            url: `https://h0mew0rk.herokuapp.com/api/labs/${id}/`
        }).then(response => {
            setLab(response.data)
            setCourse_lab(response.data.course_lab)
            setCourse(response.data.course_lab.course)
        })
    }, [id])

    return (
        <div>
            <h4><strong>{course_lab.name}</strong></h4>
            <h6><strong>Курс:</strong> {course.name}</h6>
            <p><strong>Задание:</strong> {course_lab.task}</p>
            <p><strong>Выдана:</strong> {lab.issued}</p>
            <p><strong>Выполнена:</strong> {lab.completed}</p>
        </div>
    );
}

export default LabDetail;
```

5.9. mysite-ui\src\components\Report\Report.js

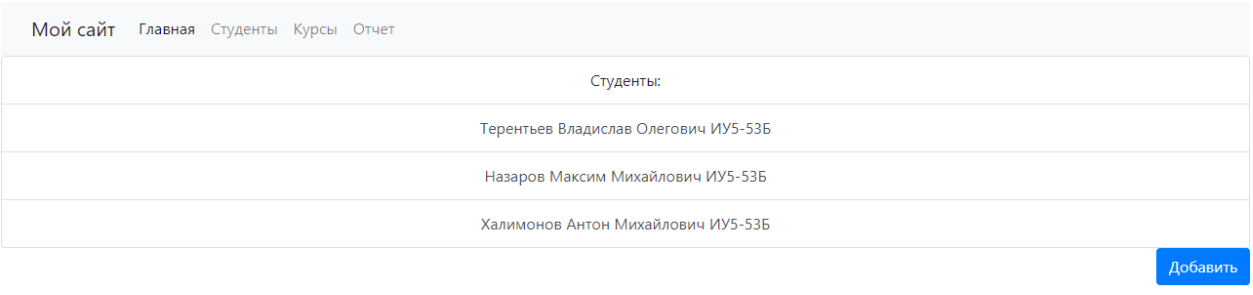
```
import React, { useState, useEffect } from 'react';
```

```
import 'bootstrap/dist/css/bootstrap.min.css';
import axios from 'axios';
import 'c3/c3.css';
import c3 from 'c3/c3.min.js';

function Report() {

    const [courseLabs, setSCourseLabs] = useState([])

    useEffect(() => {
        axios({
            method: "GET",
            url: 'https://h0mew0rk.herokuapp.com/api/courses-labs/'
        }).then(response => {
            setSCourseLabs(response.data)
        })
    }, [])

    var courses = new Set()
    var tmp = new Map()
    courseLabs.map(cl => {
        courses.add(cl.course.id)
        tmp.set(cl.course.id, cl.course.name)
    })

    var xd = []
    var data = ['Количество лабораторных работ']
    Array.from(courses).map(cs => {
        xd.push(tmp.get(cs))
        data.push(0)
        courseLabs.filter(cl => cl.course.id == cs).map(c => data[cs]++)
    })

    var chart = c3.generate({
        data: {
            columns: [
                data
            ]
        },
        axis: {
            x: {
                type: 'category',
                categories: xd
            }
        }
    });


    return (
        <div>
            <ul class="list-group">
```

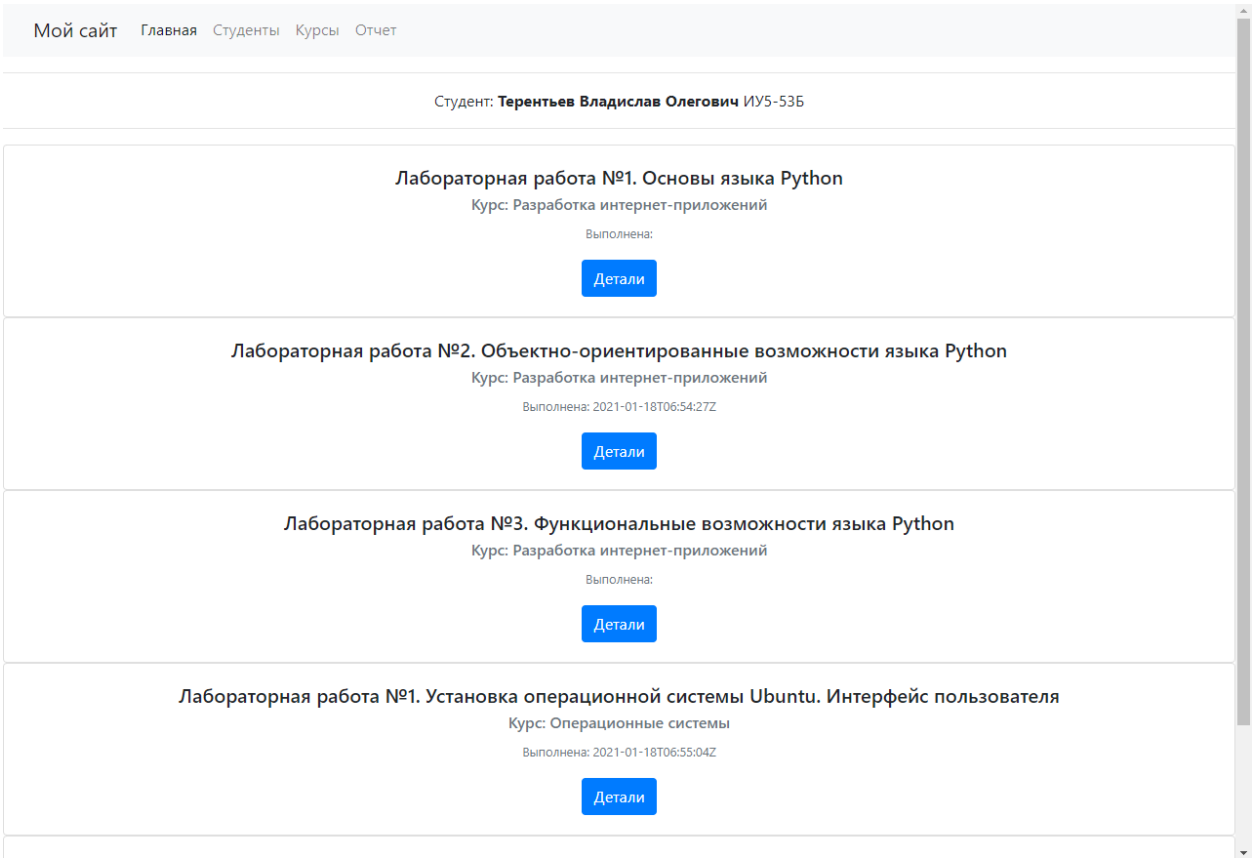
```
                {Array.from(courses).map(cs => (
                    <li class="list-group-item"><h4>{tmp.get(cs)}</h4>
                        <ul class="list-group">
                            {courseLabs.filter(cl => cl.course.id == cs).map(c =>
 (
                                <li class="list-group-item" class="text-
left"><h5>{c.name}</h5>
                                    <ul class="list-group">
                                        {c.labs.map(l => (
                                            <li class="list-group-item">
                                                <p><strong>Студент: </strong>{l.s
tudent.surname} {l.student.name} {l.student.patronymic} {l.student.group}</p>
                                                <p><strong>Выполнена: </strong>{l
.completed}</p>
                                            </li>
                                        ))}
                                    </ul>
                                </li>
                            ))}
                        </ul>
                    </li>
                ))}
            </ul>
            <div id="chart"></div>
        </div>
    );
}

export default Report;
```

6. Экранные формы веб-приложения

6.1. students

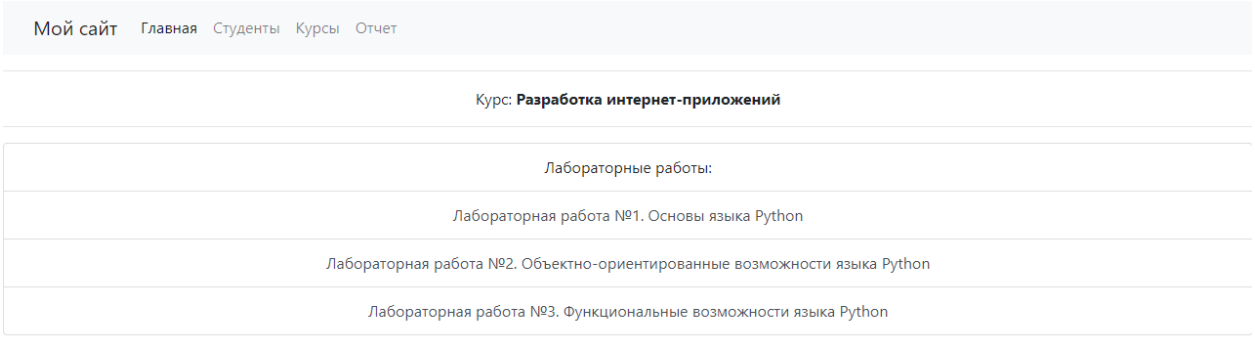


6.2. students/1

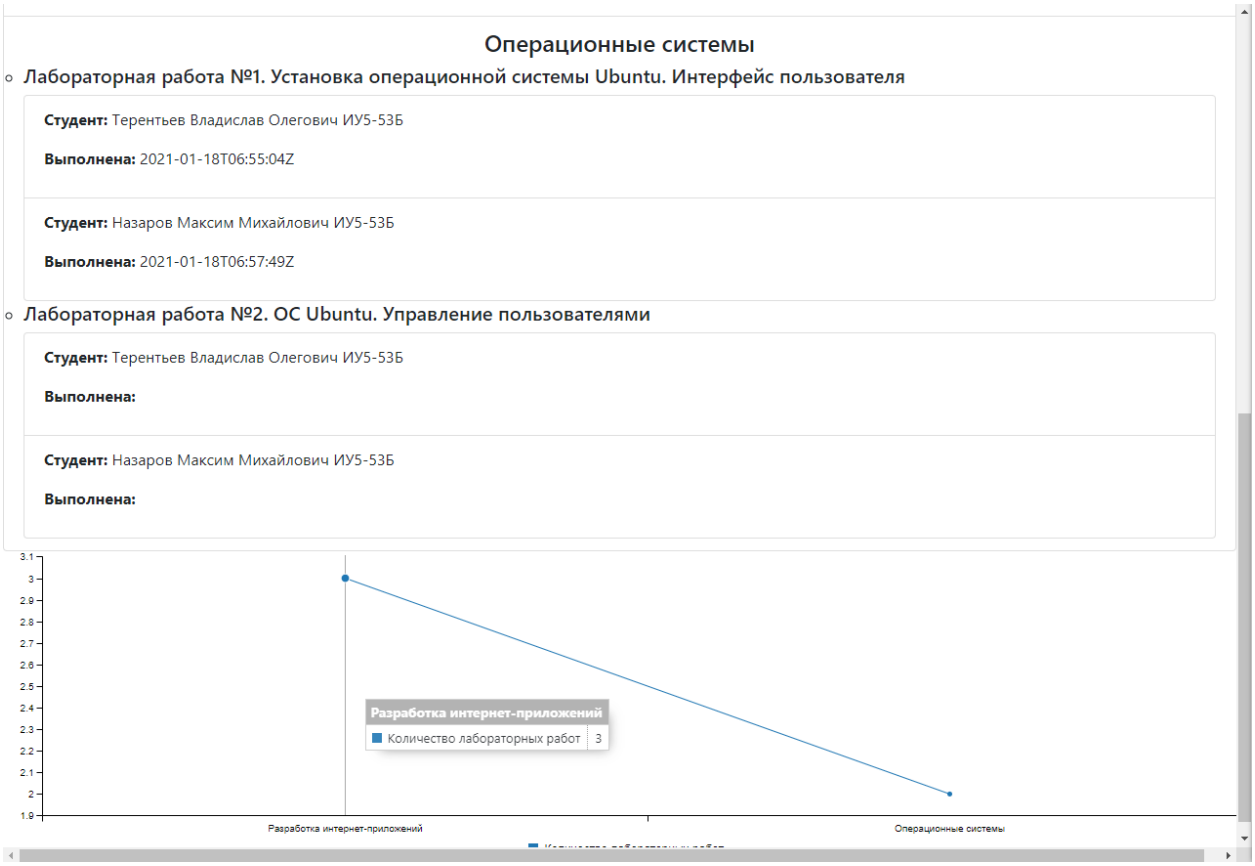


6.3. labs/1

6.4. courses/1



6.5. report



7. Ссылка на приложение в облачном сервисе

https://h0mew0rk.herokuapp.com/