

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение высшего образования «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина» (УрФУ)

Институт фундаментального образования

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к Лабораторной работе №6

по теме: \_\_Разработка приложения с двумя Web формами и БД\_

по дисциплине: \_\_\_\_\_Технологии программирования\_\_\_\_\_

Студент: \_\_\_Пряминин Т. В.\_\_\_

(ФИО)

Группа:\_\_РИЗ-120938у

Преподаватель: \_\_\_\_Архипов Н. А.\_\_\_\_\_\_\_

(ФИО)

Екатеринбург

2023

**ПОСТАНОВКА ЗАДАЧИ**

Разработать веб-приложения с двумя веб-формами и базой данных, с использованием Spring Boot 2, Java 8+, Postman, H2 Database.

**РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ**

Результат работы в Web браузере

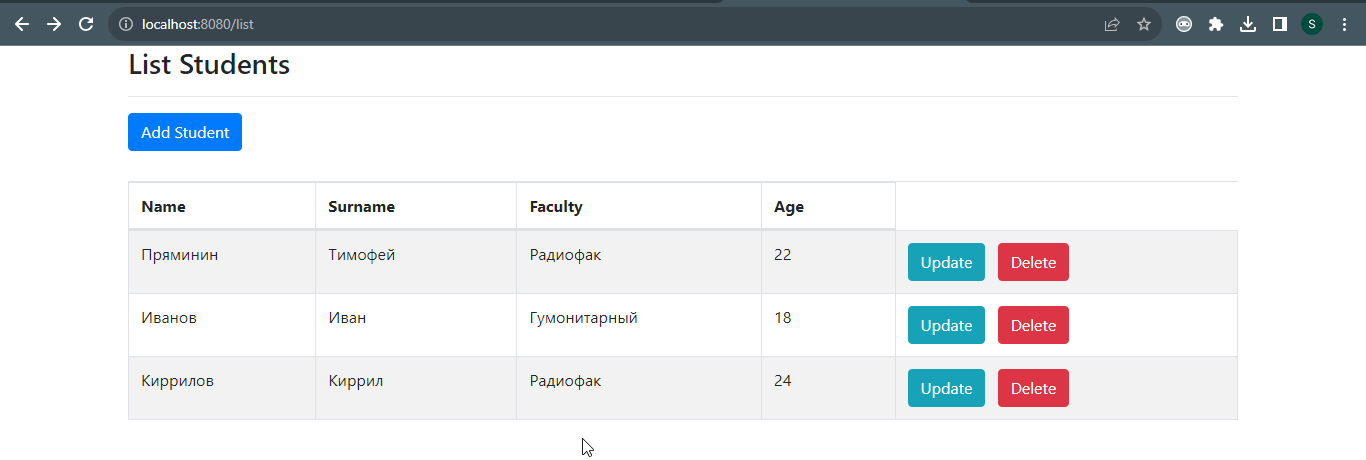


Рисунок 1 – Отображение 3 студентов

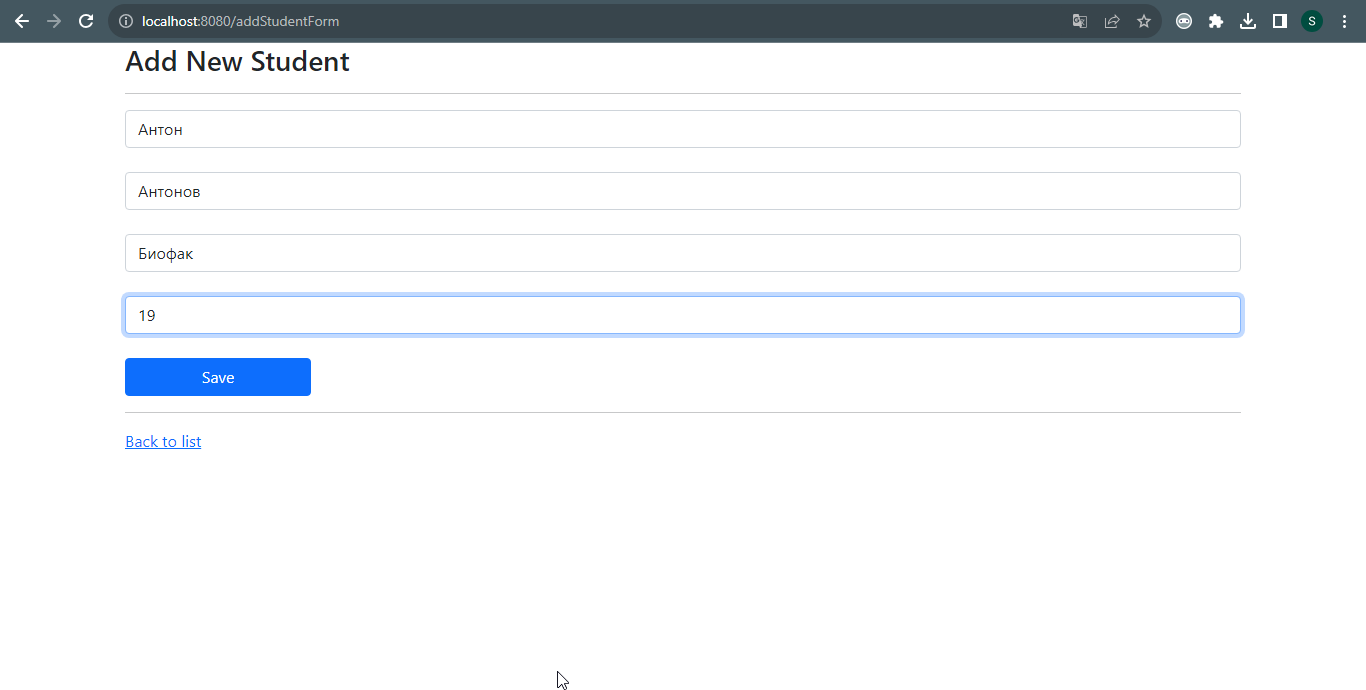


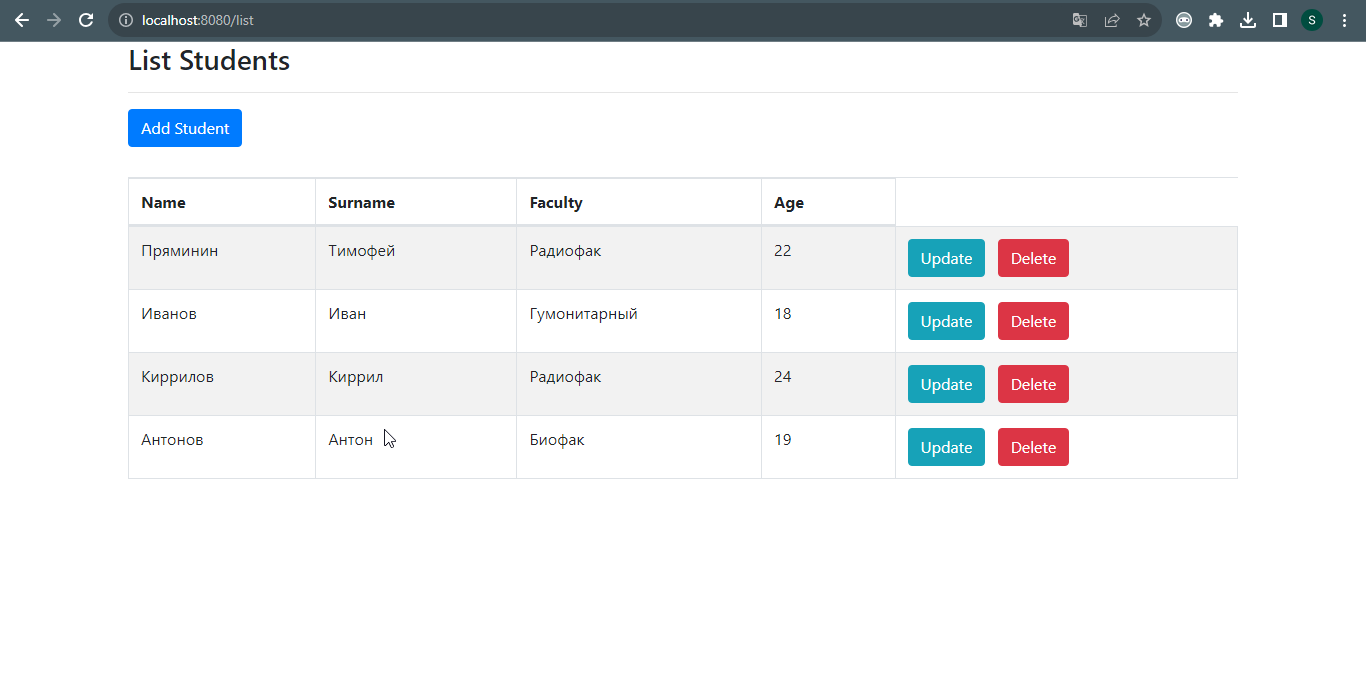
Рисунок 2 – Добавление студента

Рисунок 3 – Отображение 4 студентов

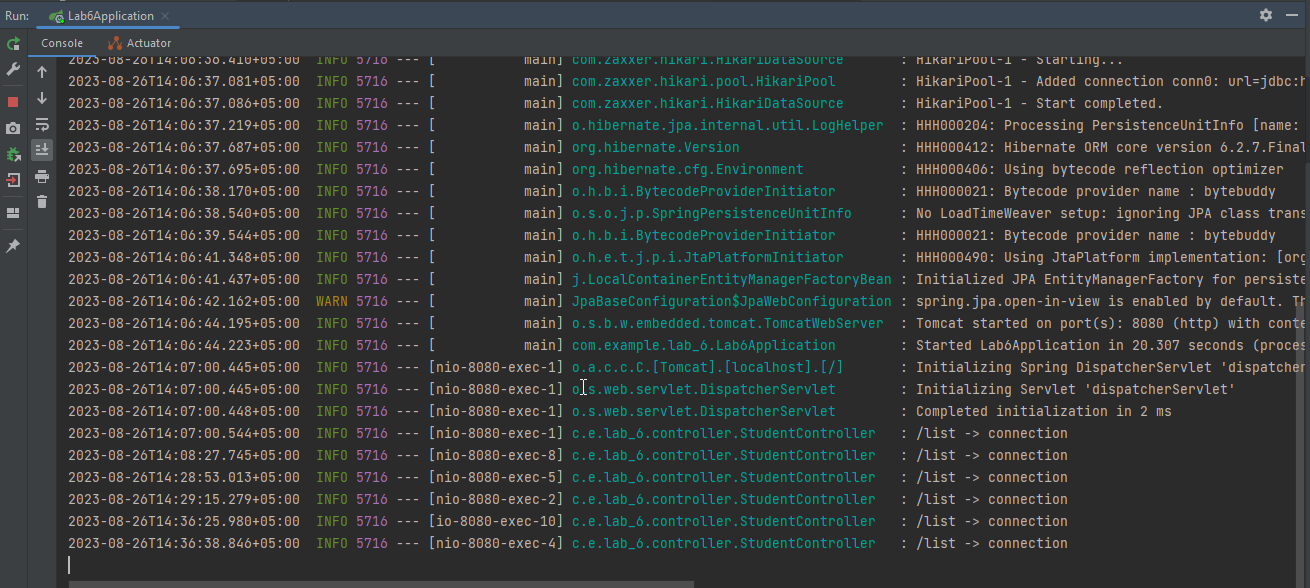


Рисунок 4 – Логи сервера

**ОПИСАНИЕ ПРОГРАММЫ**

**Класс StudentController**

@Slf4j  
@Controller  
public class StudentController {  
 @Autowired  
 private StudentRepository studentRepository;  
  
 @GetMapping({"/list", "/"})  
 public ModelAndView getAllStudents() {  
 *log*.info("/list -> connection");  
 ModelAndView mav = new ModelAndView("list-students");  
 mav.addObject("students", studentRepository.findAll());  
 return mav;  
 }  
  
 @GetMapping("/addStudentForm")  
 public ModelAndView addStudentForm() {  
 ModelAndView mav = new ModelAndView("add-student-form");  
 Student student = new Student();  
 mav.addObject("student", student);  
 return mav;  
 }  
  
 @PostMapping("/saveStudent")  
 public String saveStudent(@ModelAttribute Student student) {  
 studentRepository.save(student);  
 return "redirect:/list";  
 }  
  
 @GetMapping("/showUpdateForm")  
 public ModelAndView showUpdateForm(@RequestParam Long studentId) {  
 ModelAndView mav = new ModelAndView("add-student-form");  
 Optional<Student> optionalStudent = studentRepository.findById(studentId);  
 Student student = new Student();  
 if(optionalStudent.isPresent()) {  
 student = optionalStudent.get();  
 }  
 mav.addObject("student", student);  
 return mav;  
 }  
  
 @GetMapping("/deleteStudent")  
 public String deleteStudent(@RequestParam Long studentId) {  
 studentRepository.deleteById(studentId);  
 return "redirect:/list";  
 }  
  
}

**Класс Student**

@Getter  
@Setter  
@Entity  
@ToString  
@AllArgsConstructor  
@Table(name="STUDENTS")  
public class Student {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 @Column(name="id")  
 private long id;  
  
 @Column(name = "name")  
 private String name;  
  
 @Column(name = "surname")  
 private String surname;  
  
 @Column(name = "faculty")  
 private String faculty;  
  
 @Column(name = "age")  
 private int age;  
  
 public Student() {  
  
 }  
  
 public Student(String name, String surname, String faculty, int age) {  
 this.name = name;  
 this.surname = surname;  
 this.faculty = faculty;  
 this.age = age;  
 }  
}

**HTML страница добавления студента**

<html lang="en" xmlns:th="http://www.thymeleaf.org">  
  
<head>  
 <title>Add Student</title>  
  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"  
 rel="stylesheet"  
 integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"  
 crossorigin="anonymous">  
</head>  
<body>  
  
<div class="container">  
  
 <h3>Add New Student</h3>  
 <hr/>  
  
 <form th:action="@{/saveStudent}" th:object="${student}" method="POST">  
  
 <input type="text" th:field="\*{name}" class="form-control col-4 mb-4" placeholder="Enter Name"/>  
  
 <input type="text" th:field="\*{surname}" class="form-control col-4 mb-4" placeholder="Enter Surname"/>  
  
 <input type="text" th:field="\*{faculty}" class="form-control col-4 mb-4" placeholder="Enter Faculty"/>  
  
 <input type="text" th:field="\*{age}" class="form-control col-4 mb-4" placeholder="Enter Age"/>  
  
 <button type="submit" class="btn btn-primary col-2">Save</button>  
  
 <input type="hidden" th:field="\*{id}"/>  
  
 </form>  
 <hr/>  
 <a th:href="@{/list}">Back to list</a>  
</div>  
  
</body>  
</html>

**HTML страница отображения студентов**

<html lang="en" xmlns:th="http://www.thymeleaf.org">  
  
<head>  
 <title>List Students</title>  
  
 <link rel="stylesheet"  
 type="text/css"  
 href="https://cdn.datatables.net/v/bs4/dt-1.10.25/datatables.min.css" />  
 <link rel="stylesheet"  
 href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css"  
 integrity="sha384-B0vP5xmATw1+K9KRQjQERJvTumQW0nPEzvF6L/Z6nronJ3oUOFUFpCjEUQouq2+l"  
 crossorigin="anonymous" />  
</head>  
  
<body>  
  
<div class="container">  
  
 <h3>List Students</h3>  
  
 <hr/>  
 <a th:href="@{/addStudentForm}" class="btn btn-primary">Add Student</a>  
 <br/><br/>  
 <table class="table table-bordered table-striped" id="studentTable">  
  
 <thead>  
 <tr>  
 <th>Name</th>  
 <th>Surname</th>  
 <th>Faculty</th>  
 <th>Age</th>  
 </tr>  
 </thead>  
 <tbody>  
 <tr th:each="student: ${students}">  
 <td th:text="${student.surname}" />  
 <td th:text="${student.name}" />  
 <td th:text="${student.faculty}" />  
 <td th:text="${student.age}" />  
 <td>  
 <a th:href="@{/showUpdateForm(studentId=${student.id})}" class="btn btn-info">Update</a>  
  
 <a th:href="@{/deleteStudent(studentId=${student.id})}" class="btn btn-danger ml-2">Delete</a>  
 </td>  
 </tr>  
 </tbody>  
  
 </table>  
</div>  
  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>  
<script type="text/javascript" src="https://cdn.datatables.net/v/bs4/dt-1.10.25/datatables.min.js"></script>  
<script>  
 $(***document***).ready(function() {  
 $("#studentTable").DataTable({  
 'aoColumnDefs': [{  
 'bSortable': false,  
 'aTargets': [-1] /\* 1st one, start by the right \*/  
 }]  
 });  
 })  
</script>  
</body>  
  
</html>

**ВЫВОД**

В ходе выполнения данной лабораторной работы, были получены навыки работы с Web формами при помощи Spring Boot.

Ссылка на github с выполненными заданиями: *https://github.com/Prodiks/spring\_labs*