Министерство науки и высшего образования Российской Федерации

Муромский институт (филиал)

Федерального государственного бюджетного образовательного учреждения высшего образования

«Владимирский государственный университет   
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет ИТР

Кафедра ПИн

ЛАБОРАТОРНАЯ

РАБОТА №2

По Разработка корпоративных приложений

Тема Создание сервисного слоя в приложении Spring

Руководитель

Кульков Я.Ю.

(фамилия, инициалы)

(подпись) (дата)

Студент ПИН - 121

(группа)

Ермилов М.В.

(фамилия, инициалы)

(подпись) (дата)

Муром 2024

Лабораторная работа №2

Тема: создание сервисного слоя в приложении Spring.

Цели и задачи: изучить структуру многоуровневого Spring-приложения.

Ход работы: задание: реализовать Spring-приложение, обрабатывающее список студентов. Каждый студент должен иметь поля: имя, фамилия, отчество, город, пол, возраст, факультет, группа;

- реализовать добавление нового студента, отображение списка студентов, детальную информацию, а также удаление выбранного;

- при отображении страниц использовать стили оформления.

Листинг кода 1 – MainController.java

package ru.kafpin.firsttt;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.Model;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.ModelAttribute;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.PostMapping;  
import ru.kafpin.firsttt.service.StudentService;  
  
import java.util.List;  
  
@Controller  
public class MainController {  
 private StudentService studentService;  
  
 @Autowired  
 public void setStudentService(StudentService studentService){  
 this.studentService = studentService;  
 }  
  
 @GetMapping("/")  
 public String initPage(Model model){  
 return "redirect:/institut";  
 }  
  
 @GetMapping("/institut")  
 public String mainPage(Model model) {  
 List<Student> allStudents = studentService.getAllStudents();  
 model.addAttribute("students", allStudents);  
 return "institut";  
 }  
 @GetMapping("/details/{id}")  
 public String detailsPage(Model model, @PathVariable("id") Long id) {  
 Student selectedStudent = studentService.getStudentById(id);  
 model.addAttribute("selectedStudent", selectedStudent);  
 return "details";  
 }  
 @GetMapping("/students/delete/{id}")  
 public String deleteStudentById(@PathVariable("id") Long id){  
 studentService.deleteStudentById(id);  
 return "redirect:/institut";  
 }

@GetMapping("/students/add")  
 public String addStudent(Model model){  
 model.addAttribute("student", new Student());  
 return "addstudent";  
 }  
  
 @PostMapping("/students/add")  
 public String mainForm(@ModelAttribute Student student, Model  
 model){  
 model.addAttribute("student", student);  
 studentService.addStudent(student);  
 return "redirect:/institut";  
 }  
}

Листинг кода 2 – Student.java

package ru.kafpin.firsttt;  
  
import lombok.Data;  
import lombok.NoArgsConstructor;  
  
@Data  
@NoArgsConstructor  
public class Student {  
 private Long id;  
 private String surname;  
 private String name;  
 private String patronym;  
 private String city;  
 private String gender;  
 private int age;  
 private String faculty;  
 private String group;  
  
 public Student(Long id, String surname, String name, String patronym, String city, String gender, int age, String faculty, String group) {  
 this.id = id;  
 this.surname = surname;  
 this.name = name;  
 this.patronym = patronym;  
 this.city = city;  
 this.gender = gender;  
 this.age = age;  
 this.faculty = faculty;  
 this.group = group;  
 }  
}

Листинг кода 3 – StudentRepository.java

package ru.kafpin.firsttt;  
  
import jakarta.annotation.PostConstruct;  
import org.springframework.stereotype.Component;  
  
import java.util.ArrayList;  
import java.util.List;  
  
@Component  
public class StudentRepository {  
 private List<Student> students;  
 public List<Student> getStudents() {  
 return students;  
 }  
 public void deleteById(Long id) {  
 for (int i = 0; i < students.size(); i++) {  
 if (students.get(i).getId().equals(id)) {  
 students.remove(i);  
 return;  
 }  
 }  
 }  
 public void addStudent(Student student){  
 students.add(student);  
 }  
 @PostConstruct  
 public void init() {  
 students = new ArrayList<>();  
 students.add(new Student(1L, "Ермилов","Михаил","Анатольевич","Муром","муж.",22,"ИТР","ПИН"));  
 students.add(new Student(2L, "Сафронов","Владислав","Анатольевич","Муром","муж.",22,"ИТР","ПИН"));  
 students.add(new Student(3L, "Карасёв","Эдуард","Анатольевич","Муром","муж.",22,"ИТР","ПИН"));  
  
 }  
}

Листинг кода 4 – StudentService.java

package ru.kafpin.firsttt.service;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import ru.kafpin.firsttt.Student;  
import ru.kafpin.firsttt.StudentRepository;  
  
import java.util.List;  
  
@Service  
public class StudentService {  
 private StudentRepository studentRepository;  
 @Autowired  
 public void setStudentRepository(StudentRepository studentRepository) {  
 this.studentRepository = studentRepository;  
 }  
 public List<Student> getAllStudents() {  
 return studentRepository.getStudents();  
 }  
 public Student getStudentById(Long id) {  
 return studentRepository.getStudents().get(id.intValue() - 1);  
 }  
 public void deleteStudentById(Long id) {  
 studentRepository.deleteById(id);  
 }  
 public void addStudent(Student student) { studentRepository.addStudent(student); }  
}

Листинг кода 5 – addstudent.html

<!DOCTYPE HTML>  
<html xmlns:th="https://www.thymeleaf.org">  
<head>  
 <title>Добавление студента</title>  
 <link rel = "stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">  
 <script src = "https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>  
 <script src = "https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script>  
</head>  
<body>  
<div class="container">  
 <h1>Информация о студенте</h1>  
 <a class="btn" href="/institut">К списку студентов</a>  
 <form class="customfrm" action="#" th:action="@{/students/add}" th:object="${student}" method="post">  
 <div class="form-group">  
 <p>Id: <input type="number" class="form-control" th:field="\*{id}" /></p>  
 <p>Фамилия: <input type="text" class="form-control" placeholder="Введите фамилию" th:field="\*{surname}" /></p>  
 <p>Имя: <input type="text" class="form-control" placeholder="Введите имя" th:field="\*{name}" /></p>  
 <p>Отчество: <input type="text" class="form-control" placeholder="Введите отчество" th:field="\*{patronym}" /></p>  
 <p>Город: <input type="text" class="form-control" placeholder="Введите город" th:field="\*{city}" /></p>  
 <p>Пол: <select name="gender" class="form-control" th:field="\*{gender}">  
 <option value="male">муж.</option>  
 <option value="female">жен.</option>  
 </select></p>  
 <p>Возраст: <input type="number" class="form-control" th:field="\*{age}" /></p>  
 <p>Факультет: <input type="text" class="form-control" placeholder="Введите факультет" th:field="\*{faculty}" /></p>  
 <p>Группа: <input type="text" class="form-control" placeholder="Введите группу" th:field="\*{group}" /></p>  
 </div>  
 <p class="singleRow">  
 <input type="submit" class="btn btn-success" value="Добавить" />  
 <input type="reset" class="btn btn-secondary" value="Очистить поля ввода" />  
 </p>  
 </form>  
</div>  
</body>  
</html>

Листинг кода 6 – details.html

<!DOCTYPE html>  
<html lang="en" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>Информация о студенте</title>  
  
 <link rel = "stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">  
 <script src = "https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>  
 <script src = "https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script>  
</head>  
<body>  
<div class="container">  
 <h1>Студент</h1>  
 <ul class="list-group">  
 <li class="list-group-item">ID: <span th:text="${selectedStudent.id}"></span></li>  
 <li class="list-group-item">Фамилия: <span th:text="${selectedStudent.surname}"></span></li>  
 <li class="list-group-item">Имя: <span th:text="${selectedStudent.name}"></span></li>  
 <li class="list-group-item">Отчество: <span th:text="${selectedStudent.patronym}"></span></li>  
 <li class="list-group-item">Город: <span th:text="${selectedStudent.city}"></span></li>  
 <li class="list-group-item">Пол: <span th:text="${selectedStudent.gender}"></span></li>  
 <li class="list-group-item">Возраст: <span th:text="${selectedStudent.age}"></span></li>  
 <li class="list-group-item">Факультет: <span th:text="${selectedStudent.faculty}"></span></li>  
 <li class="list-group-item">Группа: <span th:text="${selectedStudent.group}"></span></li>  
 </ul>  
  
 <a th:href="@{'/institut'}" class="btn btn-primary mt-3">Назад к списку</a>  
</div>  
</body>  
</html>

Листинг кода 7 – institut.html

<!DOCTYPE html>  
<html lang="en" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>Институт</title>  
 <link rel="stylesheet" th:href="@{bootstrap.min.css}">  
 <script th:src = "@{jquery.min.js}"></script>  
 <script th:src = "@{bootstrap.min.js}"></script>  
</head>  
<body>  
<div class="container">  
 <h1>Институт</h1>  
 <span th:if="${students.size() == 0}">Список студентов пустой</span>  
 <table th:if="${students.size() > 0}" class="table table-hover">  
 <thead class="thead-dark">  
 <tr>  
 <td>ID</td>  
 <td>Фамилия</td>  
 <td>Имя</td>  
 <td>Отчество</td>  
 <td>Город</td>  
 <td>Пол</td>  
 <td>Возраст</td>  
 <td>Факультет</td>  
 <td>Группа</td>  
 <td></td>  
 <td></td>  
 </tr>  
 </thead>  
 <tbody>  
 <tr th:each="student : ${students}">  
 <td th:text="${student.id}"/>  
 <td th:text="${student.surname}"/>  
 <td th:text="${student.name}"/>  
 <td th:text="${student.patronym}"/>  
 <td th:text="${student.city}"/>  
 <td th:text="${student.gender}"/>  
 <td th:text="${student.age}"/>  
 <td th:text="${student.faculty}"/>  
 <td th:text="${student.group}"/>  
 <td><a class="btn btn-secondary" th:href="@{'/details/' + ${student.id}}">Подробная информация</a></td>  
 <td><a class="btn btn-danger" th:href="@{'/students/delete/' + ${student.id}}">Удалить</a></td>  
 </tr>  
 </tbody>  
 </table>  
 <a class="btn" th:href="@{'/students/add'}">Добавить студента</a>  
</div>  
</body>  
</html>

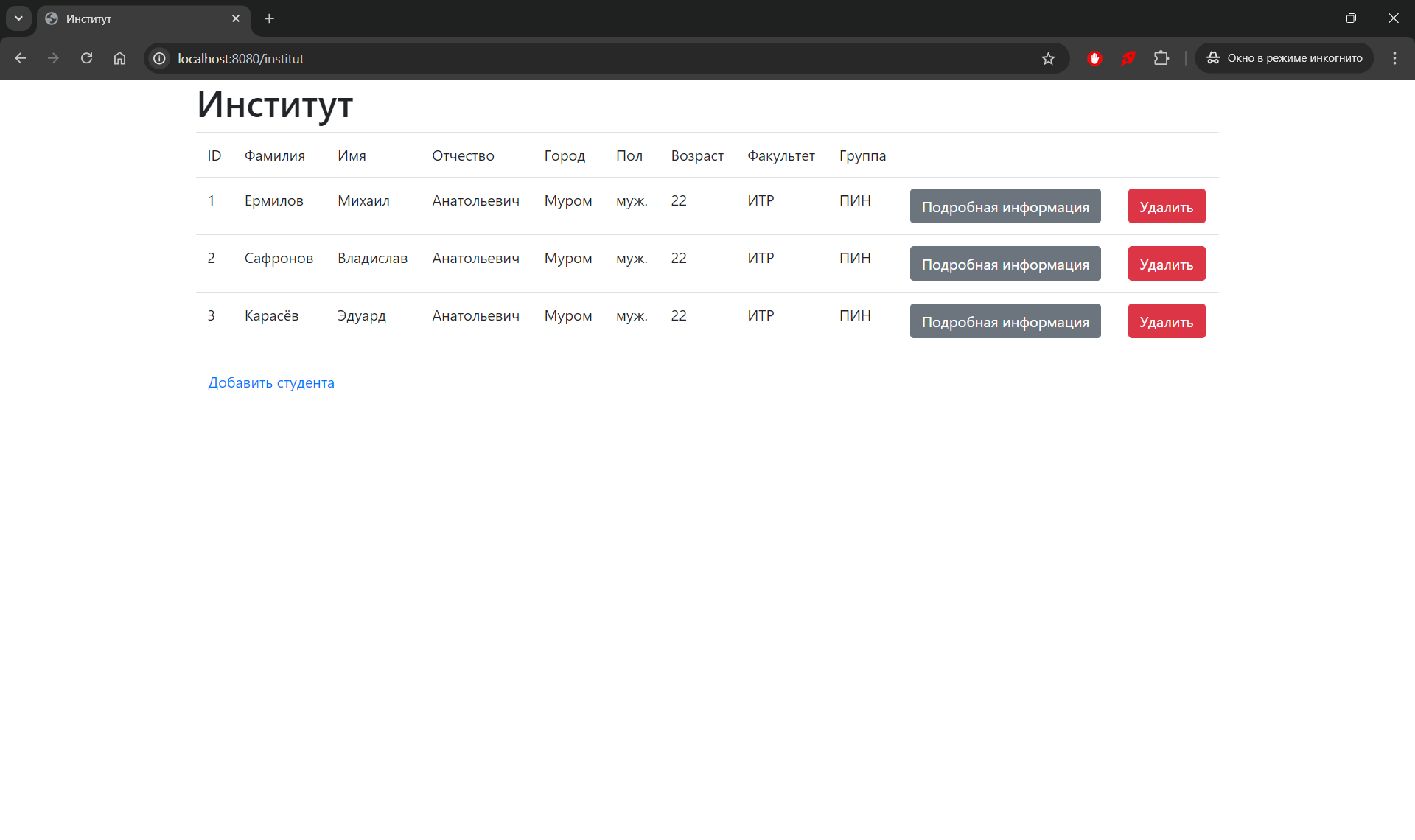


Рисунок 1 – Главная страница

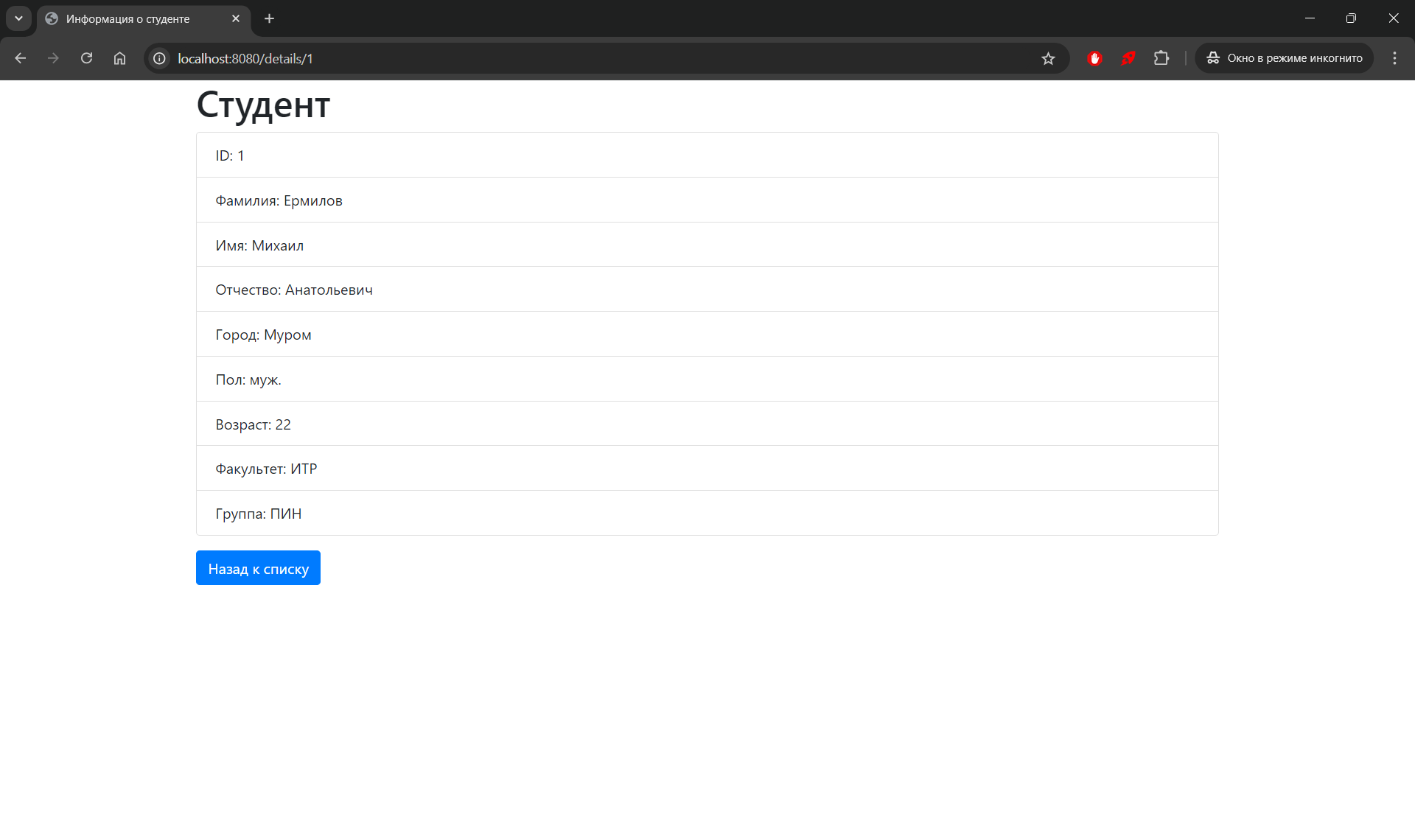


Рисунок 2 – Страница подробной информации

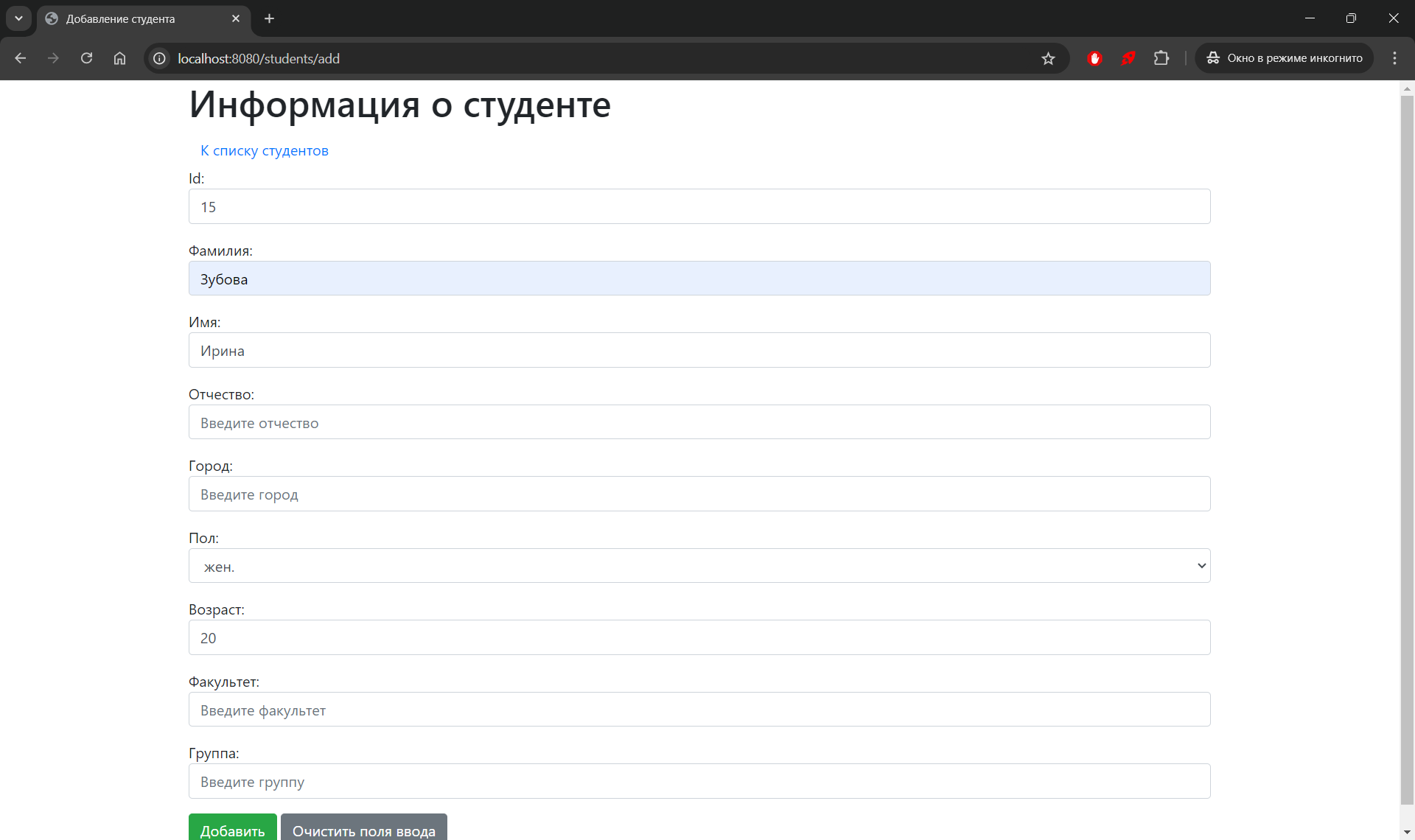


Рисунок 3 – Страница добавления

Вывод: в ходе работы изучили структуру многоуровневого Spring-приложения.