МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

И ПРОГРАМНОЙ ИНЖИНЕРИИ

№ 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| канд. тех.наук |  |  |  | Степанов П.А. |
| должность, звание |  | подпись, дата |  | фамилия, инициалы |

|  |
| --- |
| ОТЧЕТ К ЛАБОРАТОРНОЙ РАБОТЕ №5 |
| «Разработка приложения c использованием thymeleaf» |
| по дисциплине: Технология разработки серверных информационных систем |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТГР. № | 4936 |  |  |  | Судак.А.А. |
|  |  |  | подпись, дата |  | фамилия, инициалы |

**Санкт-Петербург 2021**

**Задание:**

Выполните следующие задачи:

1) Реализуйте задачи из лабораторной работы 1 средствами Thymeleaf

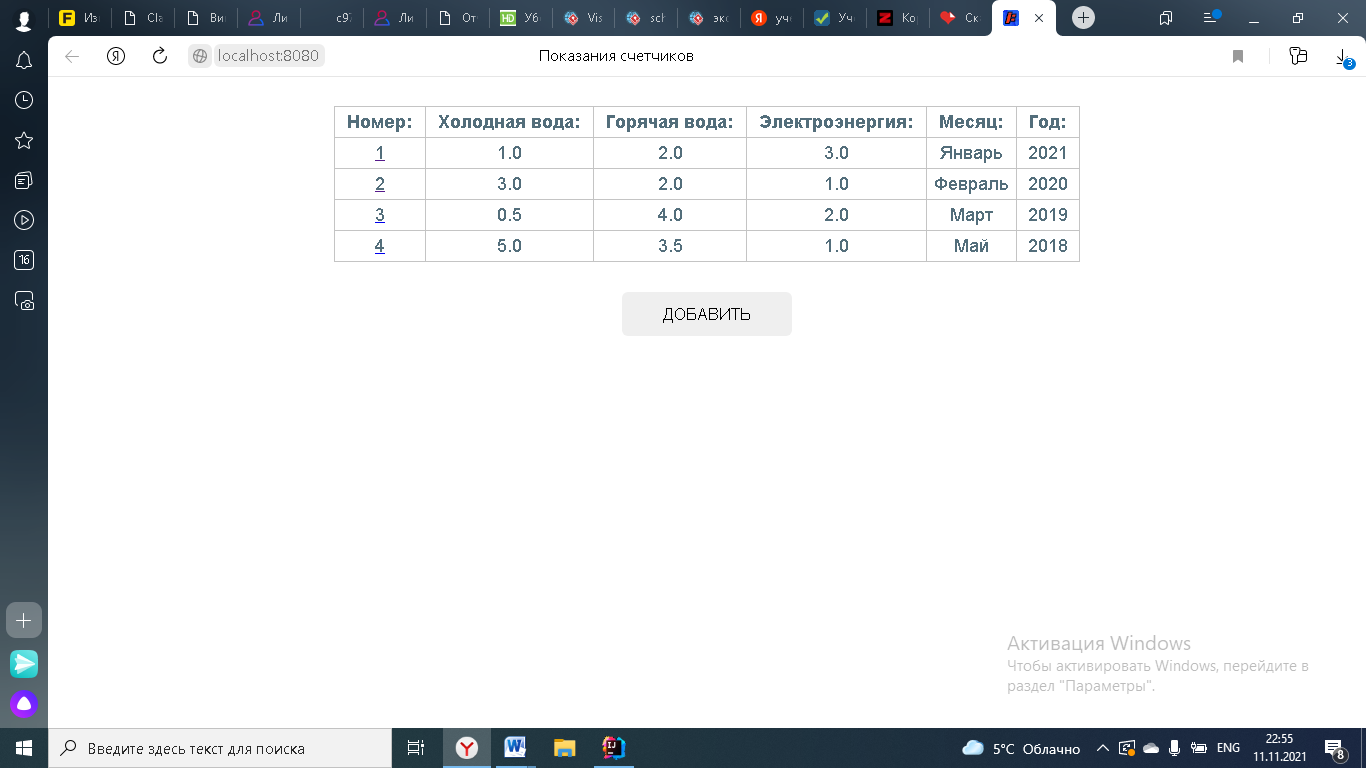
2) Обратите внимание на локализацию приложения. Русский язык должен поддерживаться.

**Вариант 18:**

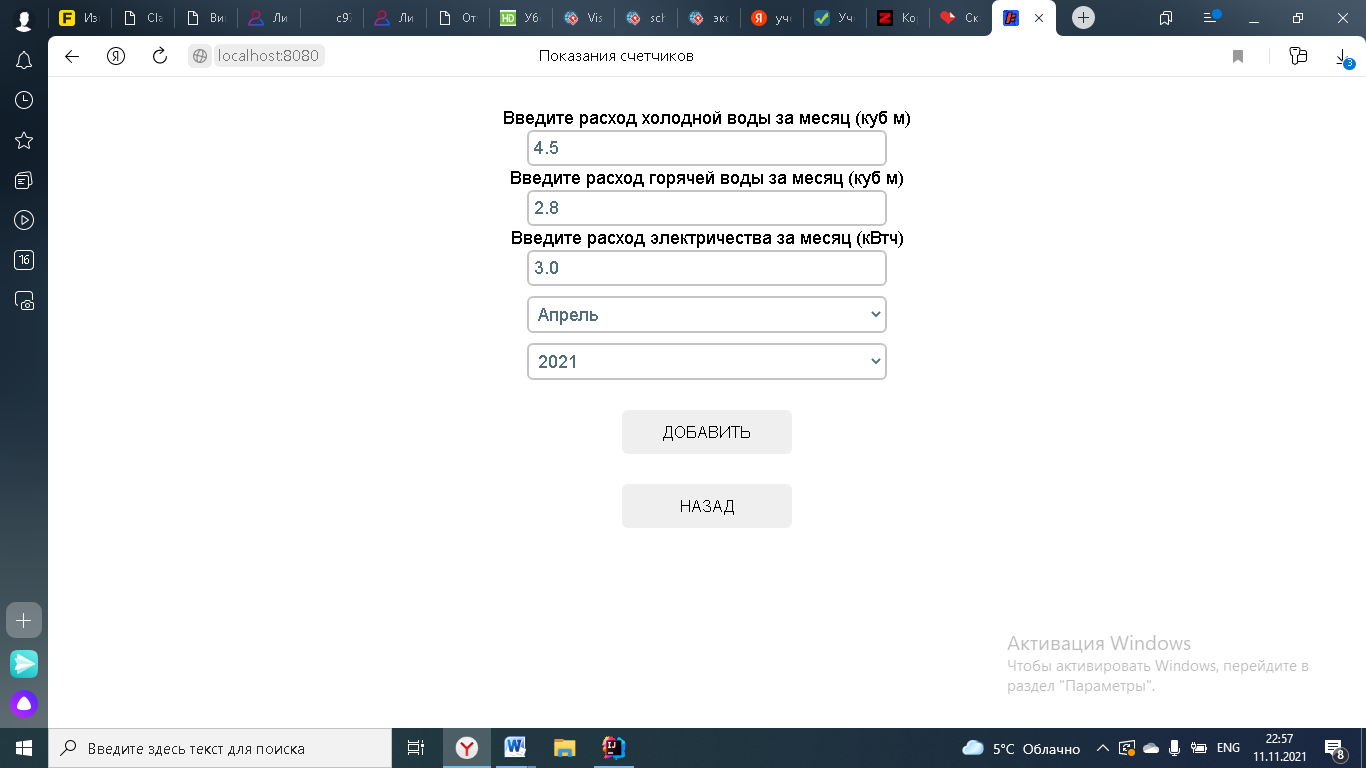
Учет показателей потребления электроэнергии и водоснабжения

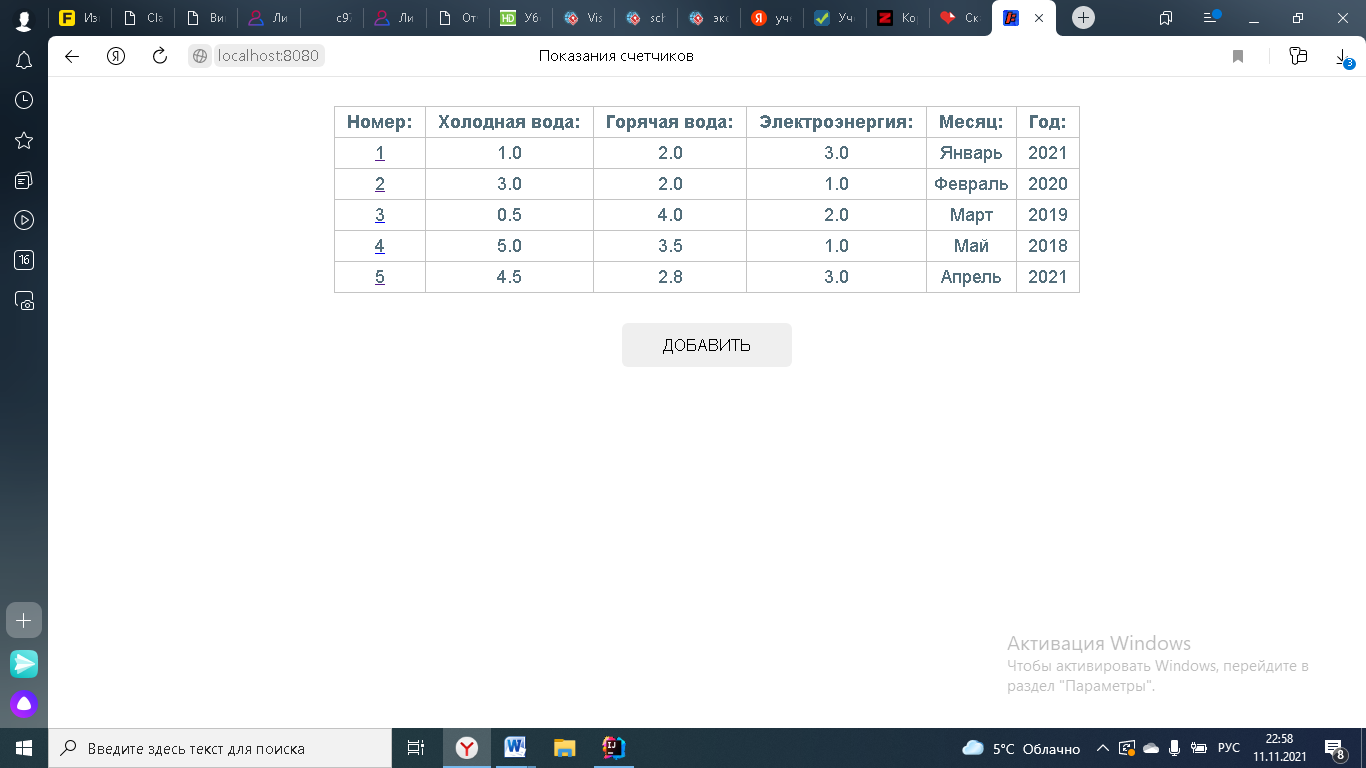
**Ход Работы:**

1. В ходе работы была разработана форма для добавления, удаления и изменения записей потребления электроэнергии и водоснабжения.

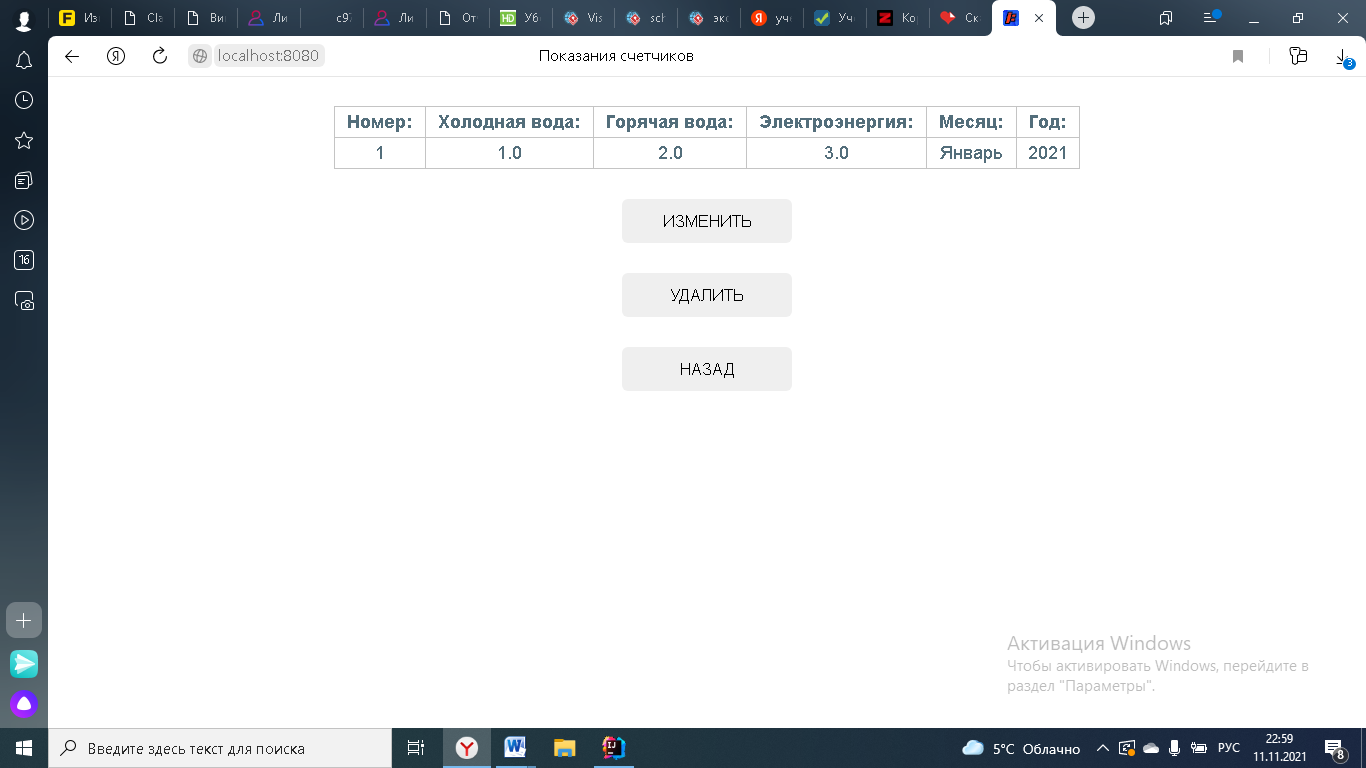


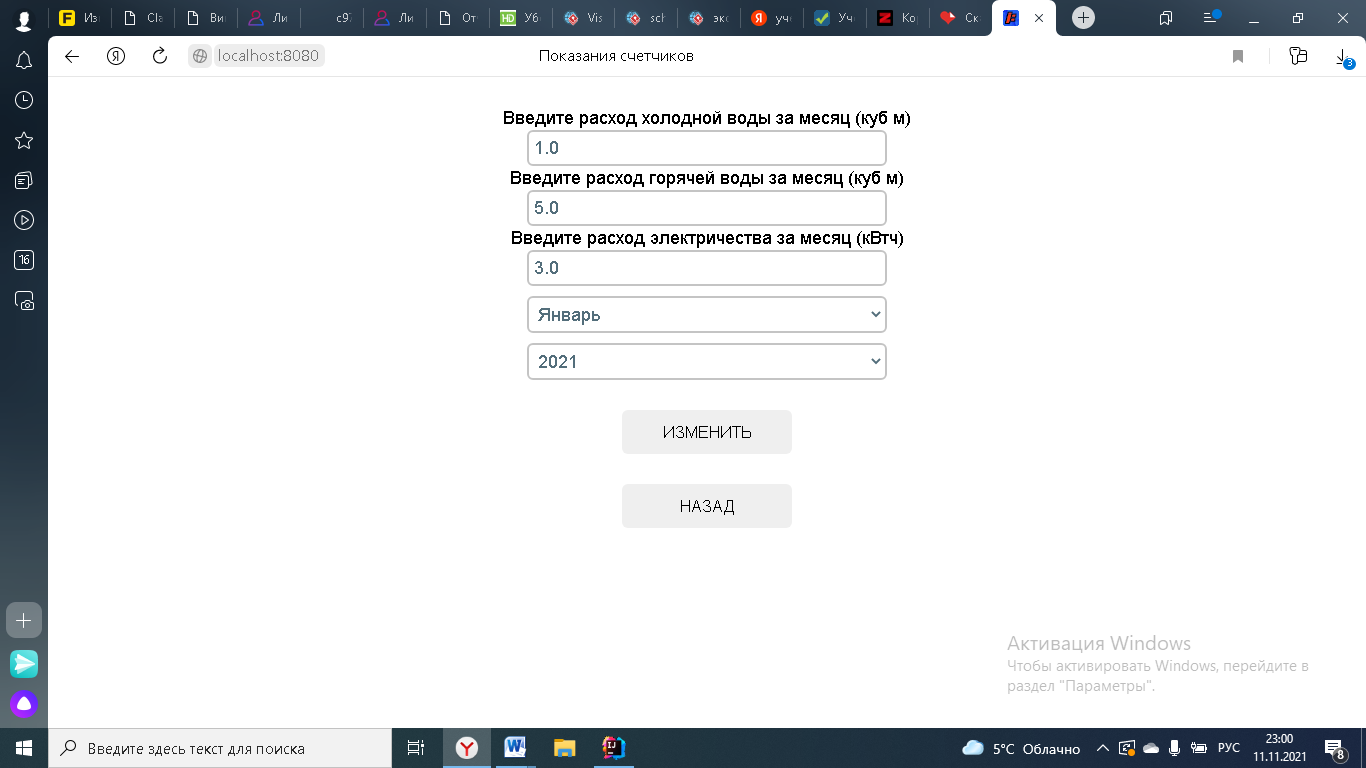
Добавим данные, нажав на кнопку добавить:

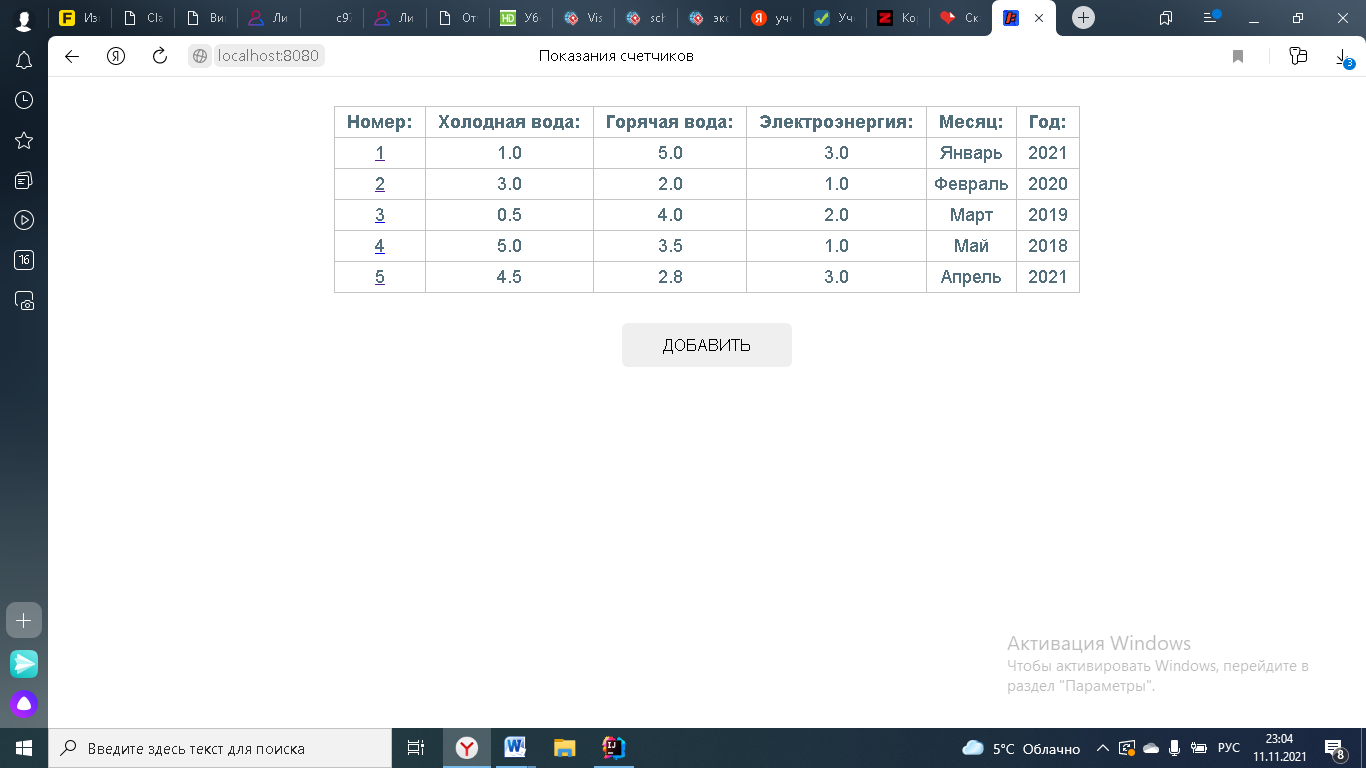




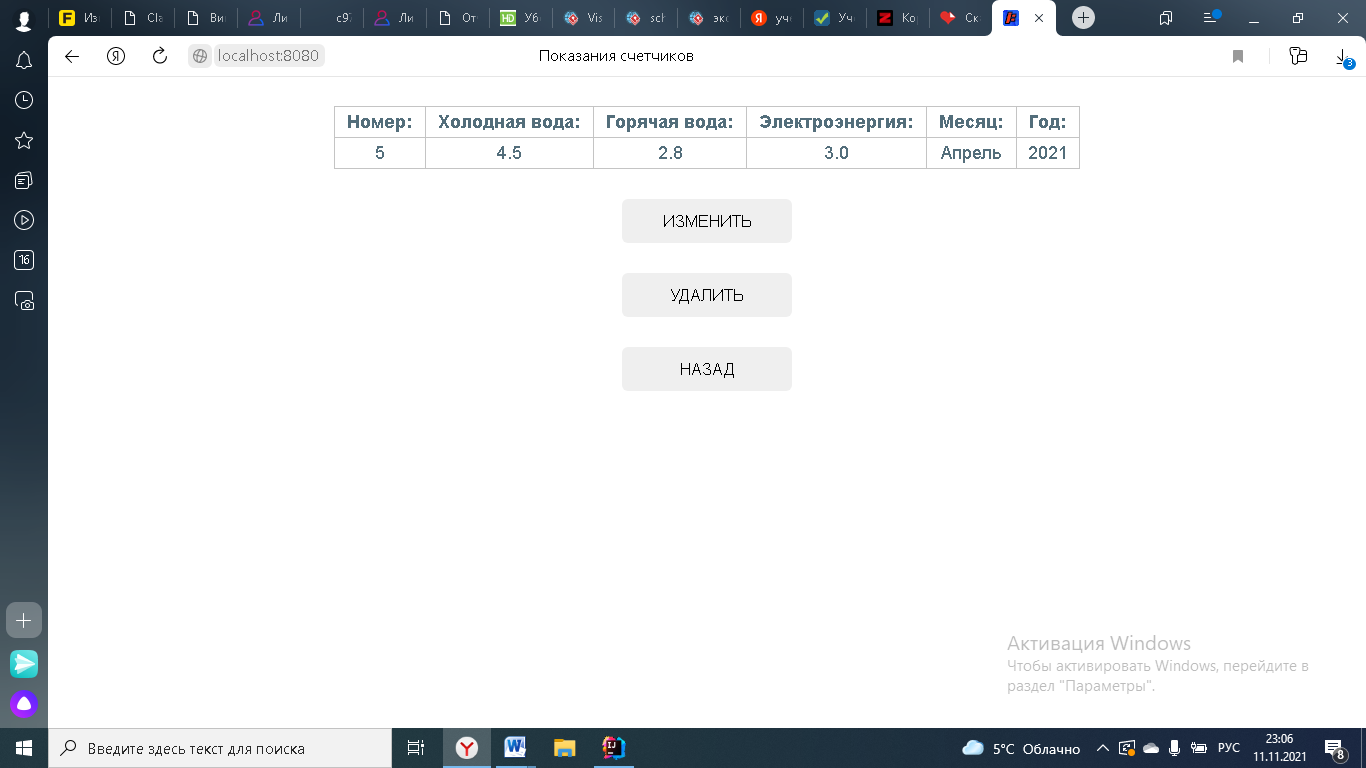
Изменим показатели за январь, нажав на нужный номер, а затем на кнопку изменить:

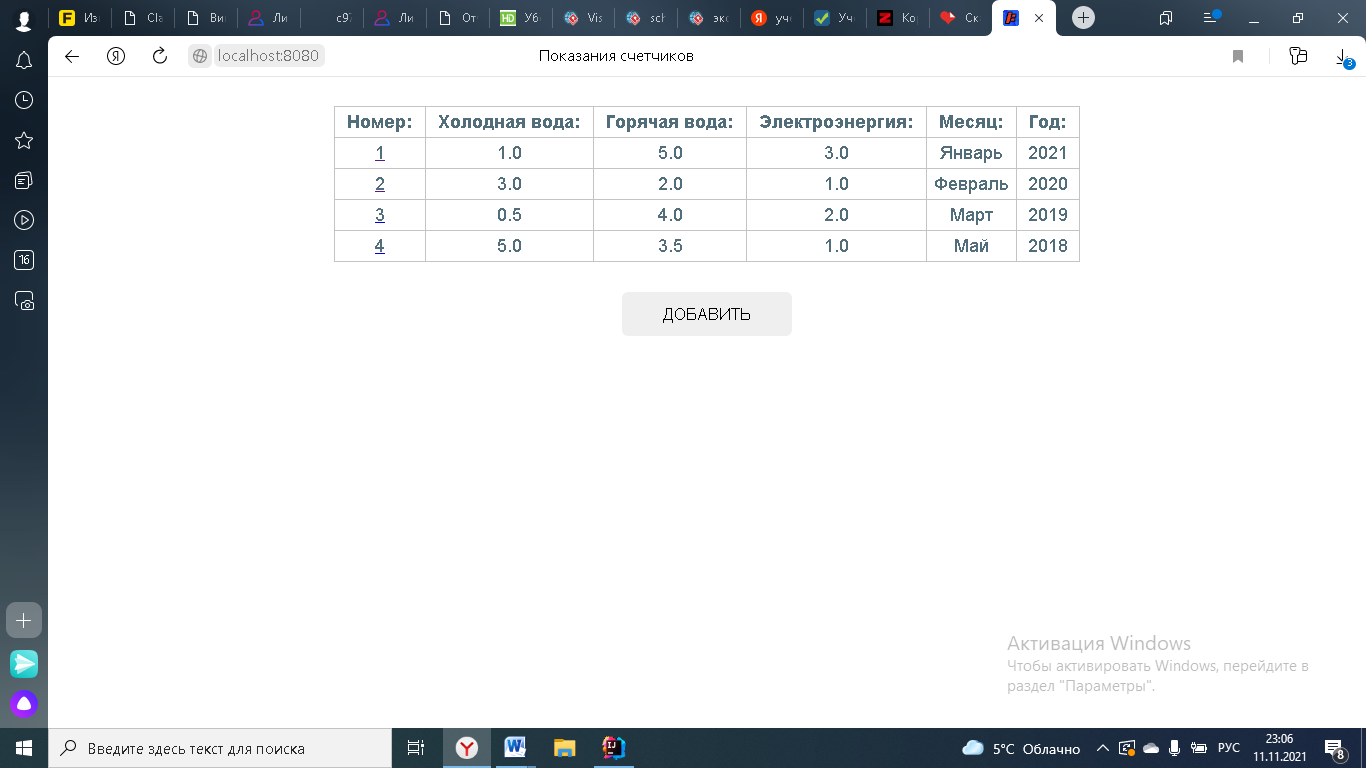






Удалим показатели за апрель, нажав на нужный номер, а затем на удалить:





1. Листинг основных файлов:

Pom.xml:

<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>2.5.6</version>  
 <relativePath/> <!-- lookup parent from repository -->  
 </parent>  
 <groupId>com.mobest1an.labs</groupId>  
 <artifactId>LAB2</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>LAB2</name>  
 <description>LAB2</description>  
 <properties>  
 <java.version>11</java.version>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-thymeleaf</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 <exclusions>  
 <exclusion>  
 <groupId>org.junit.vintage</groupId>  
 <artifactId>junit-vintage-engine</artifactId>  
 </exclusion>  
 </exclusions>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

HCSController.java

package com.mobest1an.labs.LAB2.controllers;  
  
import com.mobest1an.labs.LAB2.dao.MetersDataDAO;  
import com.mobest1an.labs.LAB2.models.MetersData;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.Model;  
import org.springframework.web.bind.annotation.\*;  
  
@Controller  
@RequestMapping("/hcs")  
public class HCSController {  
  
 private final MetersDataDAO metersDataDAO;  
  
 @Autowired  
 public HCSController(MetersDataDAO metersDataDAO) {  
 this.metersDataDAO = metersDataDAO;  
 }  
  
 @GetMapping()  
 public String index(Model model) {  
 model.addAttribute("metersDataList", metersDataDAO.index());  
 return "hcs/index";  
 }  
  
 @GetMapping("/{id}")  
 public String show(@PathVariable("id") int id, Model model) {  
 model.addAttribute("metersData", metersDataDAO.show(id));  
 return "hcs/show";  
 }  
  
 @GetMapping("/new")  
 public String newMetersData(@ModelAttribute("metersData") MetersData metersData) {  
 return "hcs/new";  
 }  
  
 @PostMapping()  
 public String create(@ModelAttribute("metersData") MetersData metersData) {  
 metersDataDAO.save(metersData);  
 return "redirect:/hcs";  
 }  
  
 @GetMapping("/{id}/edit")  
 public String edit(Model model, @PathVariable("id") int id) {  
 model.addAttribute("metersData", metersDataDAO.show(id));  
 return "hcs/edit";  
 }  
  
 @PatchMapping("/{id}")  
 public String update(@ModelAttribute("metersData") MetersData metersData, @PathVariable("id") int id) {  
 metersDataDAO.update(id, metersData);  
 return "redirect:/hcs";  
 }  
  
 @DeleteMapping("/{id}")  
 public String delete(@PathVariable("id") int id) {  
 metersDataDAO.delete(id);  
 return "redirect:/hcs";  
 }  
}

MetersDataDAO.java

package com.mobest1an.labs.LAB2.dao;  
  
import com.mobest1an.labs.LAB2.models.MetersData;  
import org.springframework.stereotype.Component;  
  
import java.util.ArrayList;  
import java.util.List;  
  
@Component  
public class MetersDataDAO {  
  
 private List<MetersData> metersDataList;  
 private static int *RECORD\_COUNT*;  
  
 {  
 metersDataList = new ArrayList<>();  
  
 metersDataList.add(new MetersData(++*RECORD\_COUNT*, 1, 2, 3, "Январь", "2021"));  
 metersDataList.add(new MetersData(++*RECORD\_COUNT*, 3, 2, 1, "Февраль", "2020"));  
 metersDataList.add(new MetersData(++*RECORD\_COUNT*, 0.5, 4, 2, "Март", "2019"));  
 metersDataList.add(new MetersData(++*RECORD\_COUNT*, 5, 3.5, 1, "Май", "2018"));  
 }  
  
 public List<MetersData> index() {  
 return metersDataList;  
 }  
  
 public MetersData show(int id) {  
 return metersDataList.stream().filter(metersData -> metersData.getId() == id).findAny().orElse(null);  
 }  
  
 public void save(MetersData metersData) {  
 metersData.setId(++*RECORD\_COUNT*);  
 metersDataList.add(metersData);  
 }  
  
 public void update(int id, MetersData updatedMetersData) {  
 MetersData metersDataToBeUpdated = show(id);  
 metersDataToBeUpdated.setColdWater(updatedMetersData.getColdWater());  
 metersDataToBeUpdated.setHotWater(updatedMetersData.getHotWater());  
 metersDataToBeUpdated.setElectricity(updatedMetersData.getElectricity());  
 metersDataToBeUpdated.setMonth(updatedMetersData.getMonth());  
 metersDataToBeUpdated.setYear(updatedMetersData.getYear());  
 }  
  
 public void delete(int id) {  
 metersDataList.removeIf(metersData -> metersData.getId() == id);  
 }  
}

MetersData.java

package com.mobest1an.labs.LAB2.models;  
  
public class MetersData {  
  
 private int id;  
 private double coldWater;  
 private double hotWater;  
 private double electricity;  
 private String month;  
 private String year;  
  
 public MetersData() {}  
  
 public MetersData(int id, double coldWater, double hotWater, double electricity, String month, String year) {  
 this.id = id;  
 this.coldWater = coldWater;  
 this.hotWater = hotWater;  
 this.electricity = electricity;  
 this.month = month;  
 this.year = year;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public double getColdWater() {  
 return coldWater;  
 }  
  
 public void setColdWater(double coldWater) {  
 this.coldWater = coldWater;  
 }  
  
 public double getHotWater() {  
 return hotWater;  
 }  
  
 public void setHotWater(double hotWater) {  
 this.hotWater = hotWater;  
 }  
  
 public double getElectricity() {  
 return electricity;  
 }  
  
 public void setElectricity(double electricity) {  
 this.electricity = electricity;  
 }  
  
 public String getMonth() {  
 return month;  
 }  
  
 public void setMonth(String month) {  
 this.month = month;  
 }  
  
 public String getYear() {  
 return year;  
 }  
  
 public void setYear(String year) {  
 this.year = year;  
 }  
}

edit.html

<!DOCTYPE html>  
<html lang="en" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>Показания счетчиков</title>  
 <link th:href="@{/css/style.css}" rel="stylesheet" />  
</head>  
<body>  
<div class="container-main">  
 <form th:method="PATCH" th:action="@{/hcs/{id}(id=${metersData.getId()})}" th:object="${metersData}">  
 <div class="cold-water">  
 <label for="coldWater">Введите расход холодной воды за месяц (куб м)</label>  
 <br>  
 <input type="text" th:field="\*{coldWater}" id="coldWater">  
 </div>  
 <div class="hot-water">  
 <label for="hotWater">Введите расход горячей воды за месяц (куб м)</label>  
 <br>  
 <input type="text" th:field="\*{hotWater}" id="hotWater">  
 </div>  
 <div class="electricity">  
 <label for="electricity">Введите расход электричества за месяц (кВтч)</label>  
 <br>  
 <input type="text" th:field="\*{electricity}" id="electricity">  
 </div>  
 <div class="month">  
 <select th:field="\*{month}">  
 <option th:value="'Январь'" th:text="Январь"></option>  
 <option th:value="'Февраль'" th:text="Февраль"></option>  
 <option th:value="'Март'" th:text="Март"></option>  
 <option th:value="'Апрель'" th:text="Апрель"></option>  
 <option th:value="'Май'" th:text="Май"></option>  
 <option th:value="'Июнь'" th:text="Июнь"></option>  
 <option th:value="'Июль'" th:text="Июль"></option>  
 <option th:value="'Август'" th:text="Август"></option>  
 <option th:value="'Сентябрь'" th:text="Сентябрь"></option>  
 <option th:value="'Октябрь'" th:text="Октябрь"></option>  
 <option th:value="'Ноябрь'" th:text="Ноябрь"></option>  
 <option th:value="'Декабрь'" th:text="Декабрь"></option>  
 </select>  
 </div>  
 <div class="year">  
 <select th:field="\*{year}">  
 <option th:value="'2021'" th:text="2021"></option>  
 <option th:value="'2020'" th:text="2020"></option>  
 <option th:value="'2019'" th:text="2019"></option>  
 <option th:value="'2018'" th:text="2018"></option>  
 <option th:value="'2017'" th:text="2017"></option>  
 <option th:value="'2016'" th:text="2016"></option>  
 <option th:value="'2015'" th:text="2015"></option>  
 <option th:value="'2014'" th:text="2014"></option>  
 <option th:value="'2013'" th:text="2013"></option>  
 <option th:value="'2012'" th:text="2012"></option>  
 <option th:value="'2011'" th:text="2011"></option>  
 <option th:value="'2010'" th:text="2010"></option>  
 </select>  
 </div>  
 <div class="buttons">  
 <input class="btn btn-submit" type="submit" value="Изменить">  
 </div>  
 </form>  
  
 <form th:method="GET" th:action="@{/hcs}">  
 <input class="btn btn-submit" type="submit" value="Назад">  
 </form>  
</div>  
</body>  
</html>

Index.html

<!DOCTYPE html>  
<html lang="en" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>Показания счетчиков</title>  
 <link th:href="@{/css/style.css}" rel="stylesheet" />  
</head>  
<body>  
<div class="container-main">  
 <table id="last-results-table">  
 <tr>  
 <th><div class="last-results-key">Номер:</div></th>  
 <th><div class="last-results-key">Холодная вода:</div></th>  
 <th><div class="last-results-key">Горячая вода:</div></th>  
 <th><div class="last-results-key">Электроэнергия:</div></th>  
 <th><div class="last-results-key">Месяц:</div></th>  
 <th><div class="last-results-key">Год:</div></th>  
 </tr>  
 <tr th:each="metersData : ${metersDataList}">  
 <td><a th:href="@{hcs/{id}(id = ${metersData.getId()})}" th:utext="'<div>' + ${metersData.getId()} + '</div>'"></a></td>  
 <td th:utext="'<div>' + ${metersData.getColdWater()} + '</div>'"></td>  
 <td th:utext="'<div>' + ${metersData.getHotWater()} + '</div>'"></td>  
 <td th:utext="'<div>' + ${metersData.getElectricity()} + '</div>'"></td>  
 <td th:utext="'<div>' + ${metersData.getMonth()} + '</div>'"></td>  
 <td th:utext="'<div>' + ${metersData.getYear()} + '</div>'"></td>  
 </tr>  
 </table>  
  
 <form th:method="GET" th:action="@{/hcs/new}">  
 <input class="btn btn-submit" type="submit" value="Добавить">  
 </form>  
</div>  
</body>  
</html>

new.html

<!DOCTYPE html>  
<html lang="en" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>Показания счетчиков</title>  
 <link th:href="@{/css/style.css}" rel="stylesheet" />  
</head>  
<body>  
<div class="container-main">  
 <form th:method="POST" th:action="@{/hcs}" th:object="${metersData}">  
 <div class="cold-water">  
 <label for="coldWater">Введите расход холодной воды за месяц (куб м)</label>  
 <br>  
 <input type="text" th:field="\*{coldWater}" id="coldWater">  
 </div>  
 <div class="hot-water">  
 <label for="hotWater">Введите расход горячей воды за месяц (куб м)</label>  
 <br>  
 <input type="text" th:field="\*{hotWater}" id="hotWater">  
 </div>  
 <div class="electricity">  
 <label for="electricity">Введите расход электричества за месяц (кВтч)</label>  
 <br>  
 <input type="text" th:field="\*{electricity}" id="electricity">  
 </div>  
 <div class="month">  
 <select th:field="\*{month}">  
 <option th:value="'Январь'" th:text="Январь"></option>  
 <option th:value="'Февраль'" th:text="Февраль"></option>  
 <option th:value="'Март'" th:text="Март"></option>  
 <option th:value="'Апрель'" th:text="Апрель"></option>  
 <option th:value="'Май'" th:text="Май"></option>  
 <option th:value="'Июнь'" th:text="Июнь"></option>  
 <option th:value="'Июль'" th:text="Июль"></option>  
 <option th:value="'Август'" th:text="Август"></option>  
 <option th:value="'Сентябрь'" th:text="Сентябрь"></option>  
 <option th:value="'Октябрь'" th:text="Октябрь"></option>  
 <option th:value="'Ноябрь'" th:text="Ноябрь"></option>  
 <option th:value="'Декабрь'" th:text="Декабрь"></option>  
 </select>  
 </div>  
 <div class="year">  
 <select th:field="\*{year}">  
 <option th:value="'2021'" th:text="2021"></option>  
 <option th:value="'2020'" th:text="2020"></option>  
 <option th:value="'2019'" th:text="2019"></option>  
 <option th:value="'2018'" th:text="2018"></option>  
 <option th:value="'2017'" th:text="2017"></option>  
 <option th:value="'2016'" th:text="2016"></option>  
 <option th:value="'2015'" th:text="2015"></option>  
 <option th:value="'2014'" th:text="2014"></option>  
 <option th:value="'2013'" th:text="2013"></option>  
 <option th:value="'2012'" th:text="2012"></option>  
 <option th:value="'2011'" th:text="2011"></option>  
 <option th:value="'2010'" th:text="2010"></option>  
 </select>  
 </div>  
 <div class="buttons">  
 <input class="btn btn-submit" type="submit" value="Добавить">  
 </div>  
 </form>  
  
 <form th:method="GET" th:action="@{/hcs}">  
 <input class="btn btn-submit" type="submit" value="Назад">  
 </form>  
</div>  
</body>  
</html>

Show.html

<div class="container-main">  
 <table id="last-results-table">  
 <tr>  
 <th><div class="last-results-key">Номер:</div></th>  
 <th><div class="last-results-key">Холодная вода:</div></th>  
 <th><div class="last-results-key">Горячая вода:</div></th>  
 <th><div class="last-results-key">Электроэнергия:</div></th>  
 <th><div class="last-results-key">Месяц:</div></th>  
 <th><div class="last-results-key">Год:</div></th>  
 </tr>  
 <tr>  
 <td th:utext="'<div>' + ${metersData.getId()} + '</div>'"></td>  
 <td th:utext="'<div>' + ${metersData.getColdWater()} + '</div>'"></td>  
 <td th:utext="'<div>' + ${metersData.getHotWater()} + '</div>'"></td>  
 <td th:utext="'<div>' + ${metersData.getElectricity()} + '</div>'"></td>  
 <td th:utext="'<div>' + ${metersData.getMonth()} + '</div>'"></td>  
 <td th:utext="'<div>' + ${metersData.getYear()} + '</div>'"></td>  
 </tr>  
 </table>  
  
 <div class="buttons">  
 <form th:method="GET" th:action="@{/hcs/{id}/edit(id=${metersData.getId()})}">  
 <input class="btn btn-submit" type="submit" value="Изменить">  
 </form>  
  
 <form th:method="DELETE" th:action="@{/hcs/{id}(id=${metersData.getId()})}">  
 <input class="btn btn-submit" type="submit" value="Удалить">  
 </form>  
  
 <form th:method="GET" th:action="@{/hcs}">  
 <input class="btn btn-submit" type="submit" value="Назад">  
 </form>  
 </div>  
</div>  
</body>  
</html>

Вывод: В ходе работы были получены и применены на практике основные навыки разработки приложения с помощью Thymleaf.