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 №4 |
| «Разработка формы логина» |
| 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) Добавьте в приложение поддержку одной из моделей безопасности средствами springsecurity

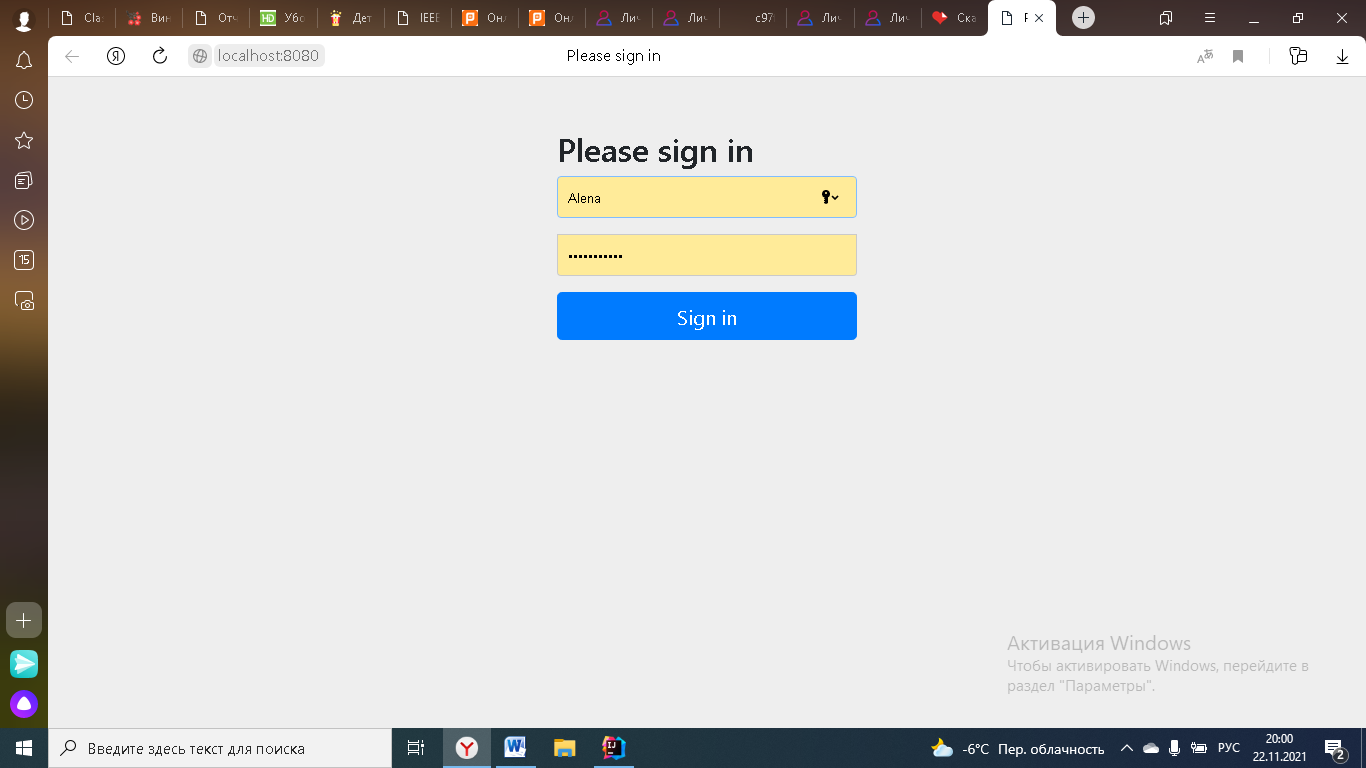
2) Защитите ресурсы, требующие запись и введите аудит.

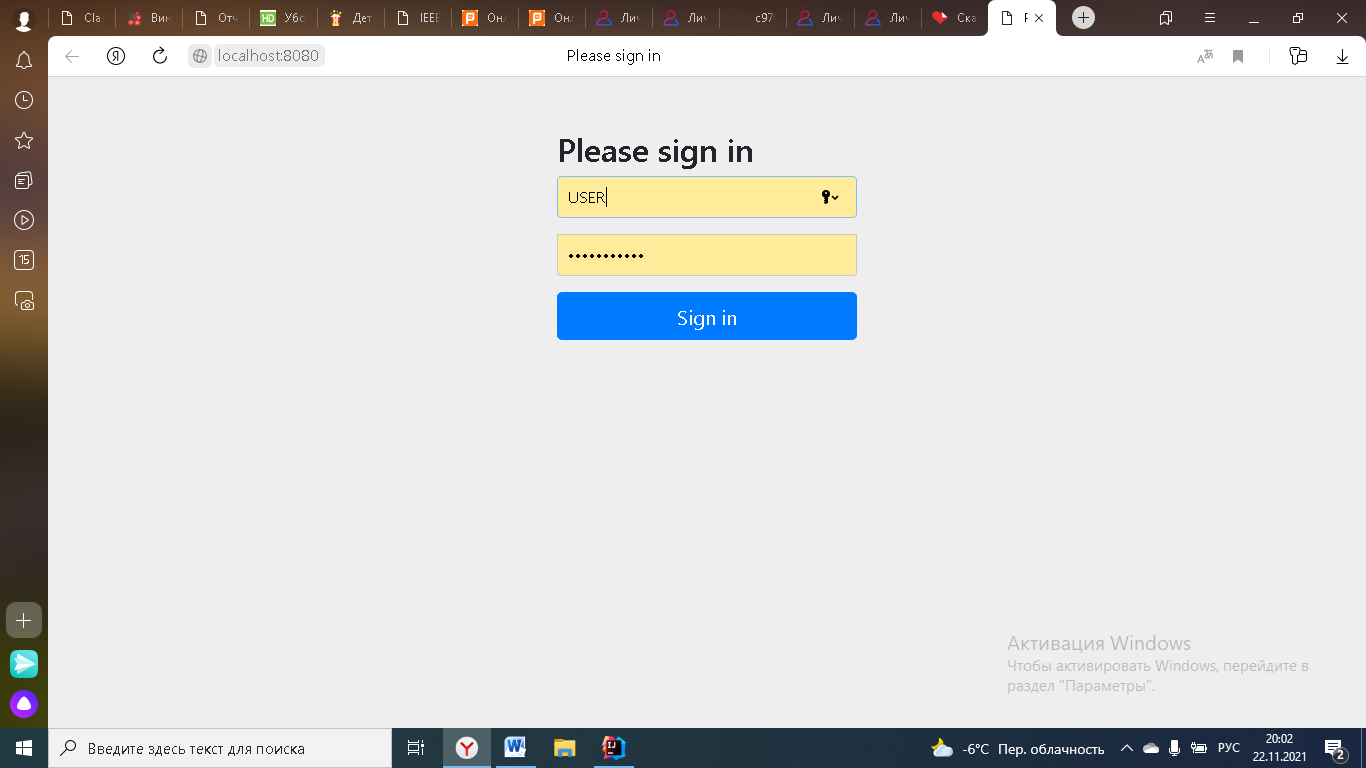
**Вариант 18:**

Учет показателей потребления электроэнергии и водоснабжения

**Ход Работы:**

1. Разработанная форма входа:

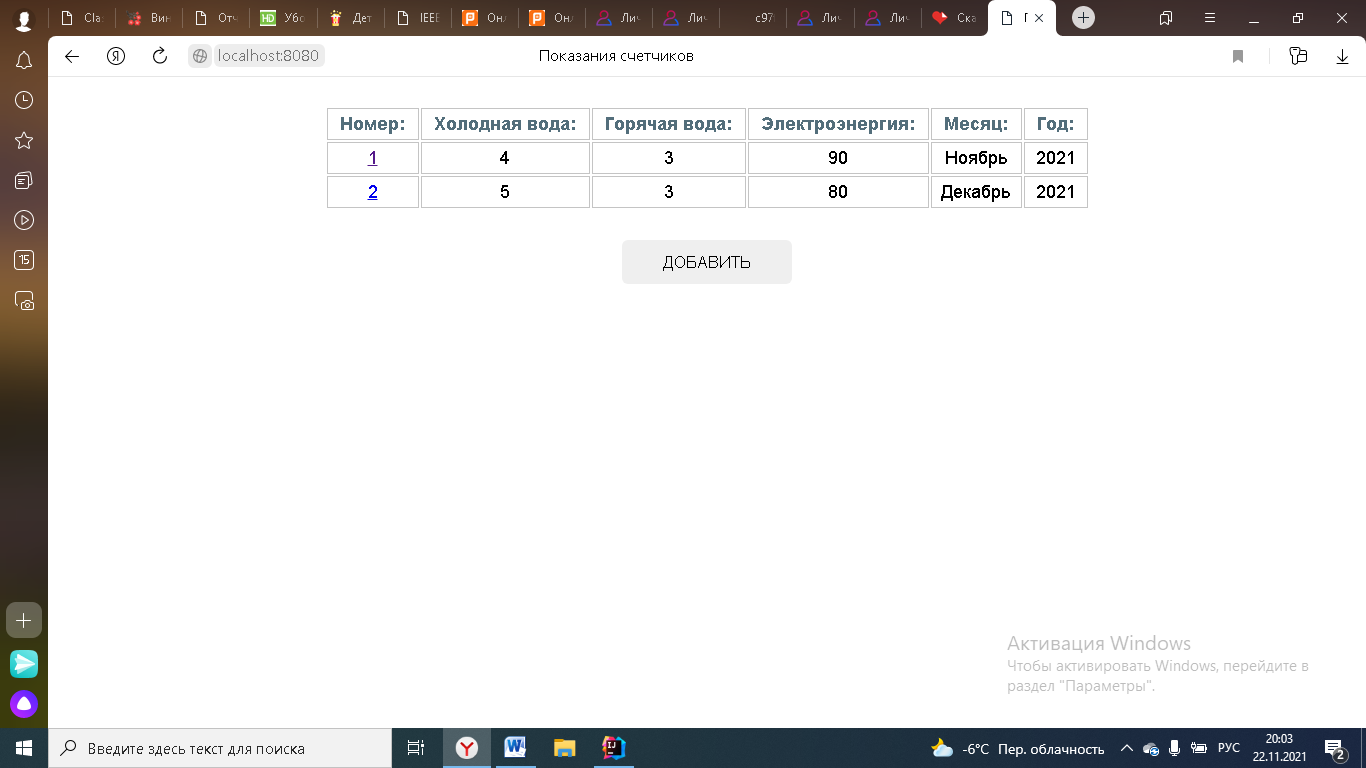




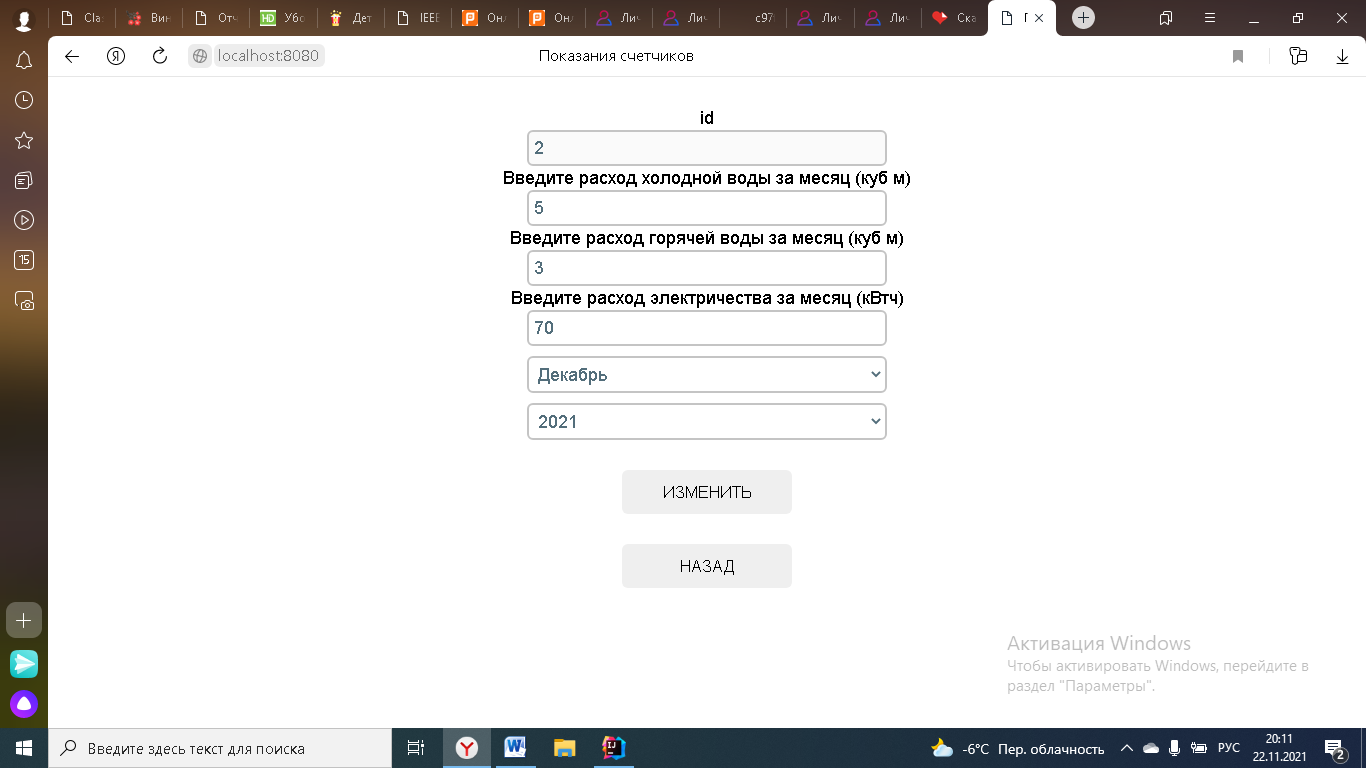
Созданы два аккаунта: Alena – с правами администратора и User, с ограниченными правами.

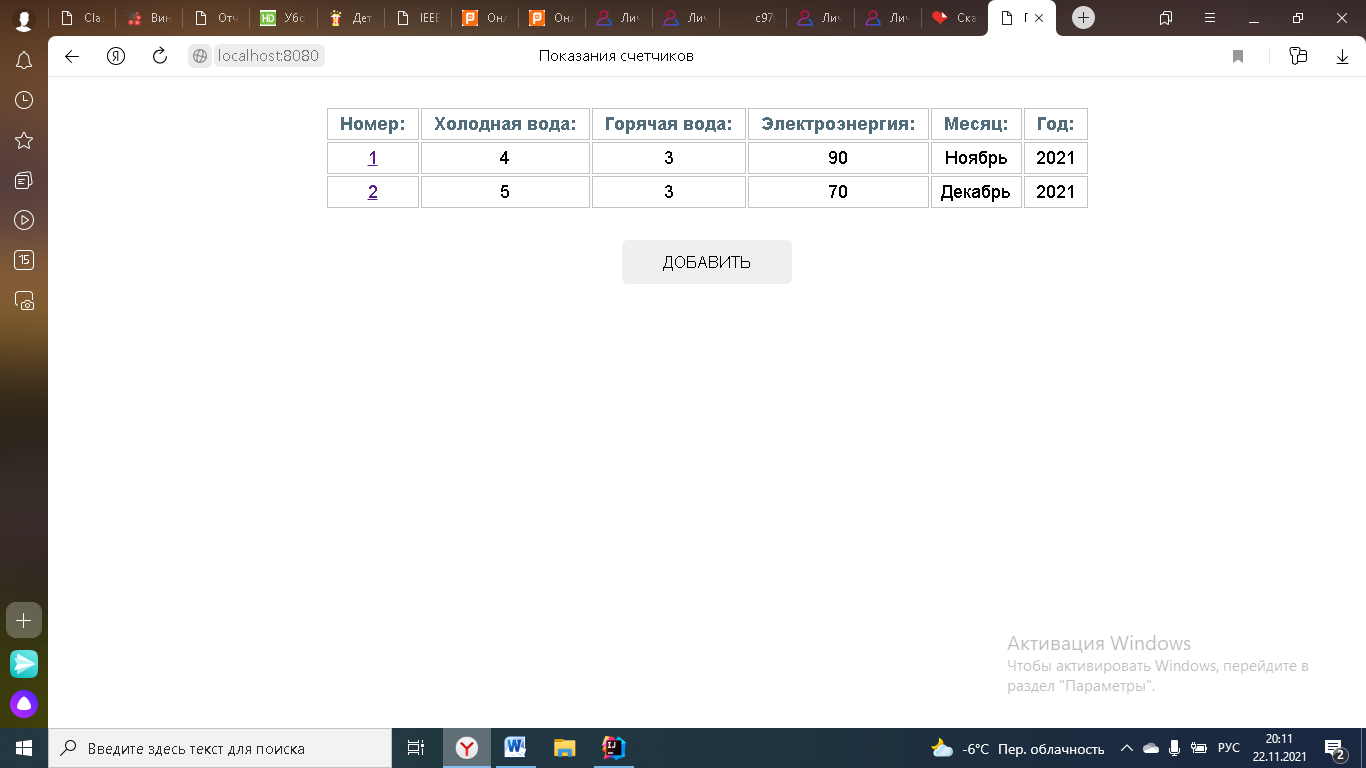
Администратор может добавлять, изменять и удалять данные, пользователь может работать только с существующими записями и не может добавлять новые.

1. Войдя в аккаунт админа создадим пару показателей

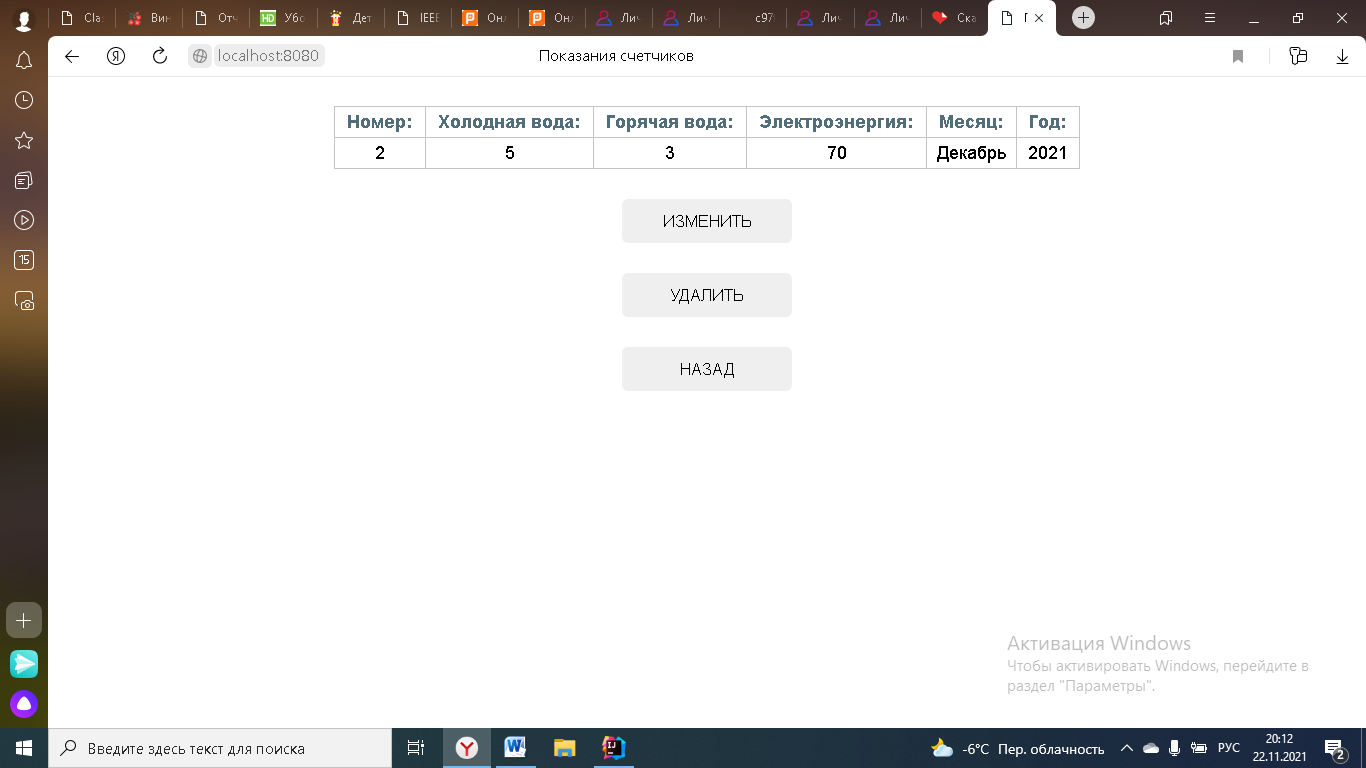


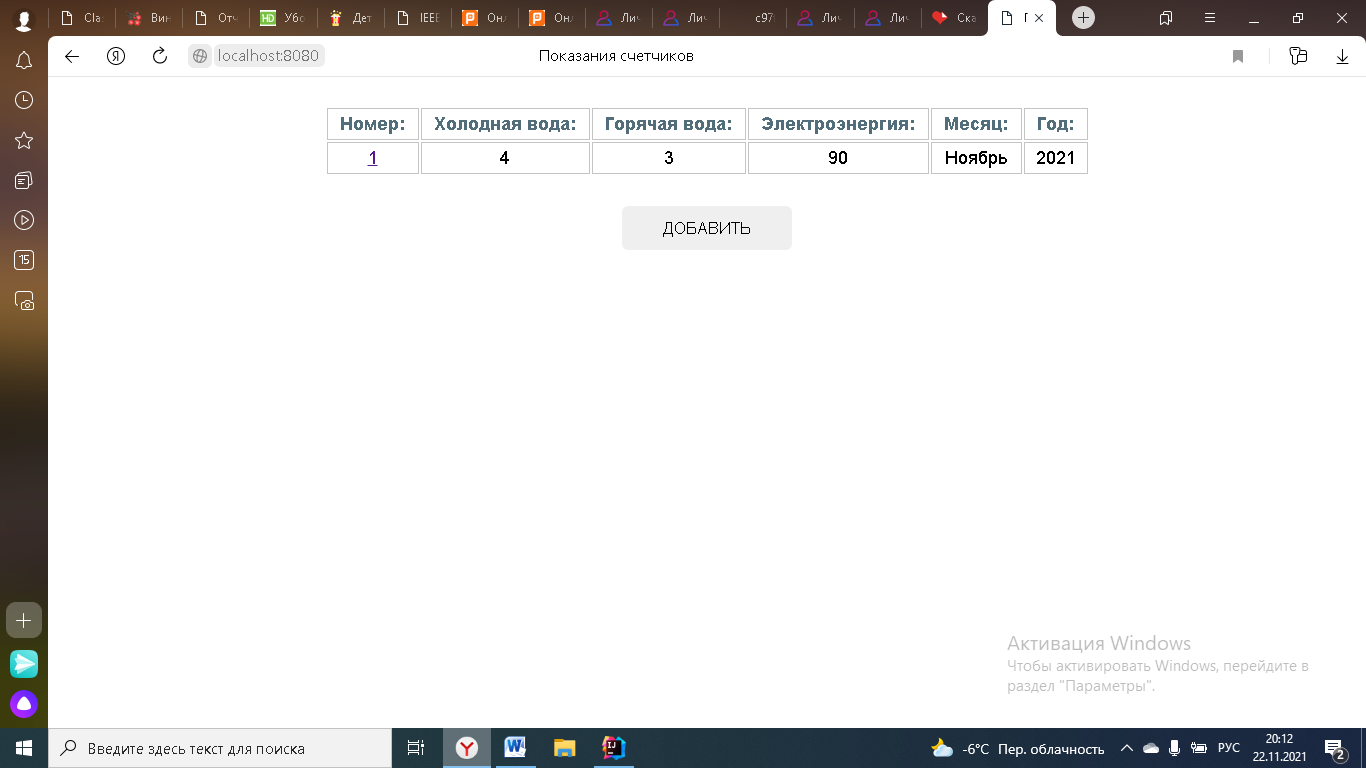
Изменим показатели за декабрь:





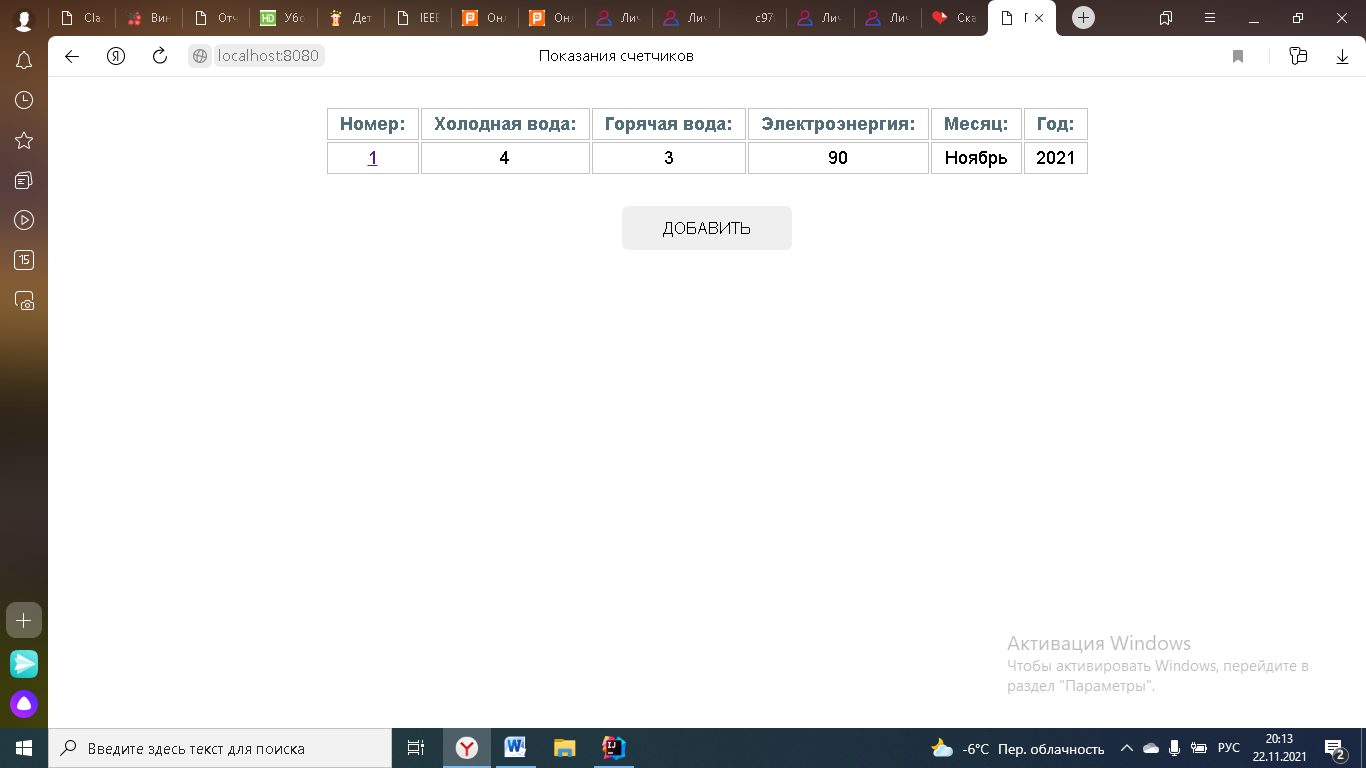
Удалим декабрьские показатели:



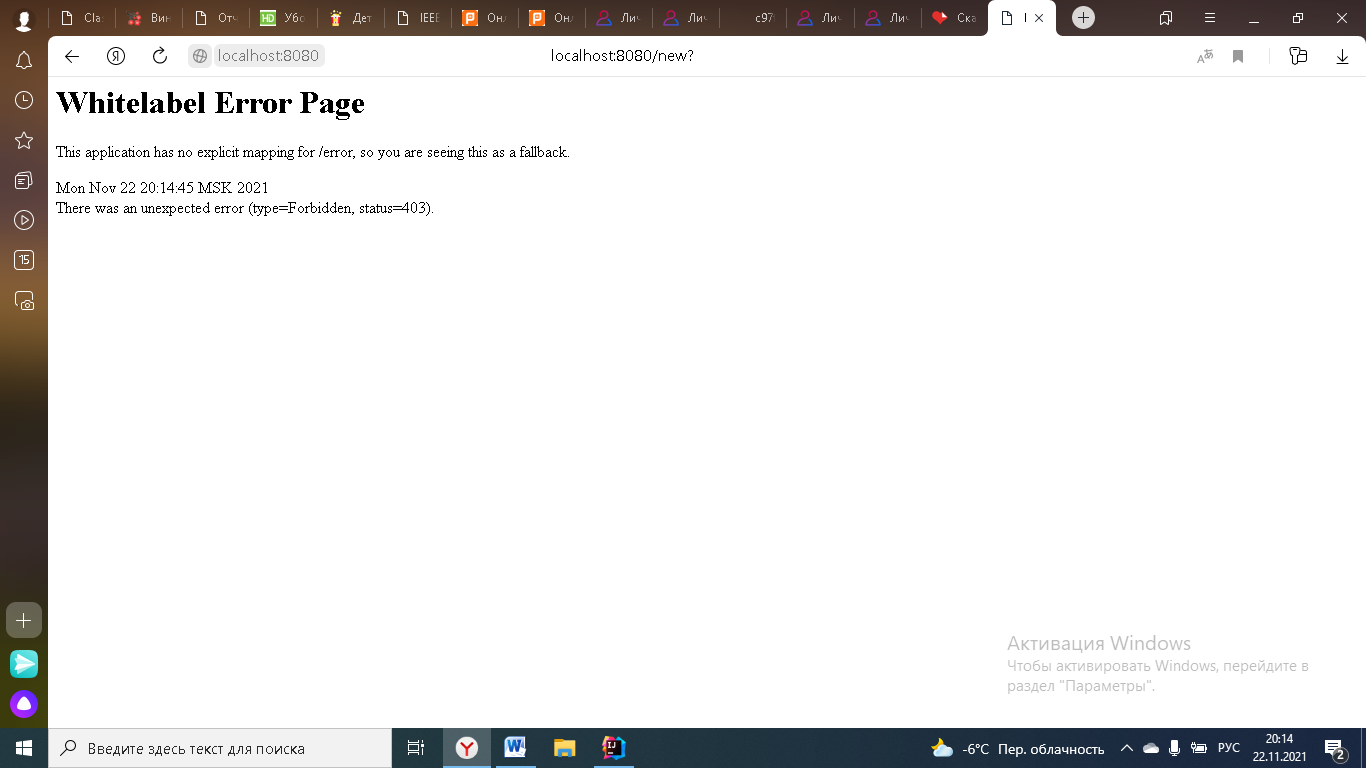


Теперь зайдем в аккаунт пользователя:

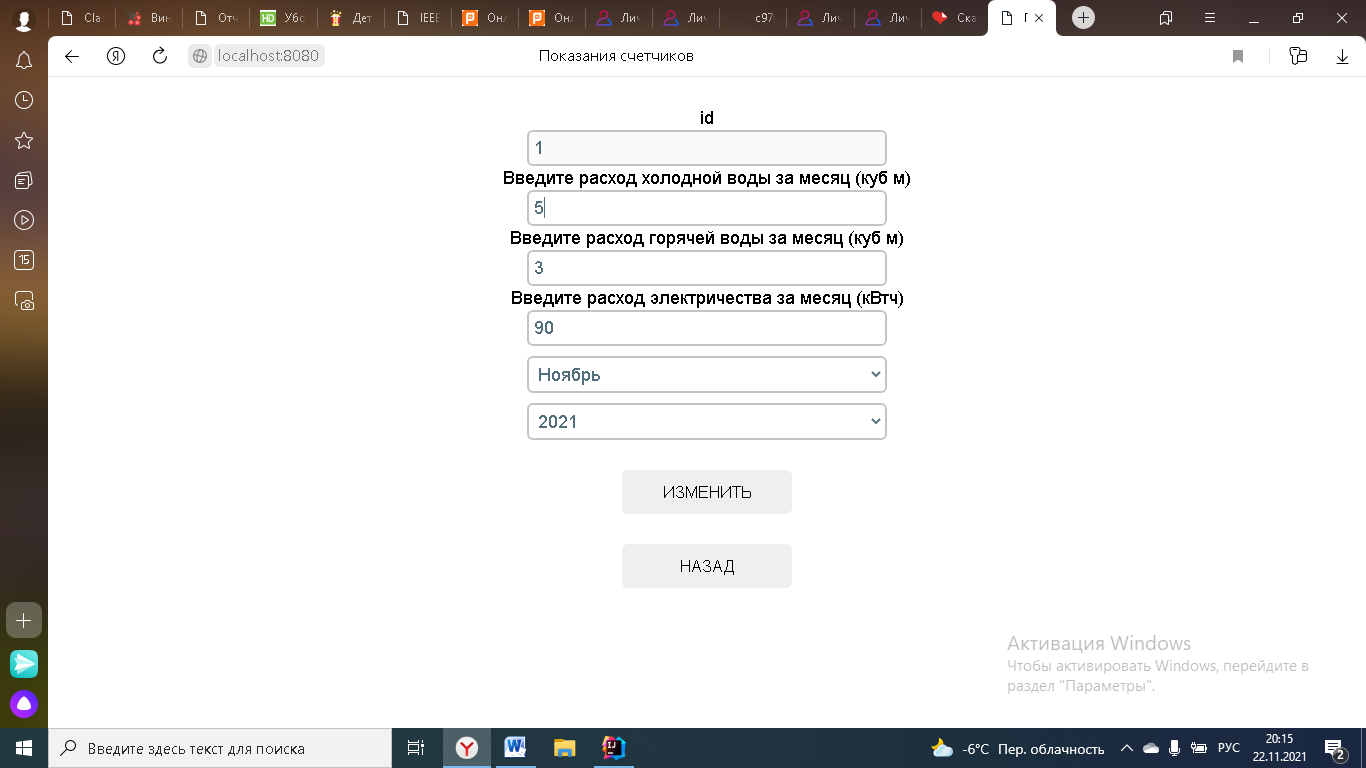
Видим существующие данные

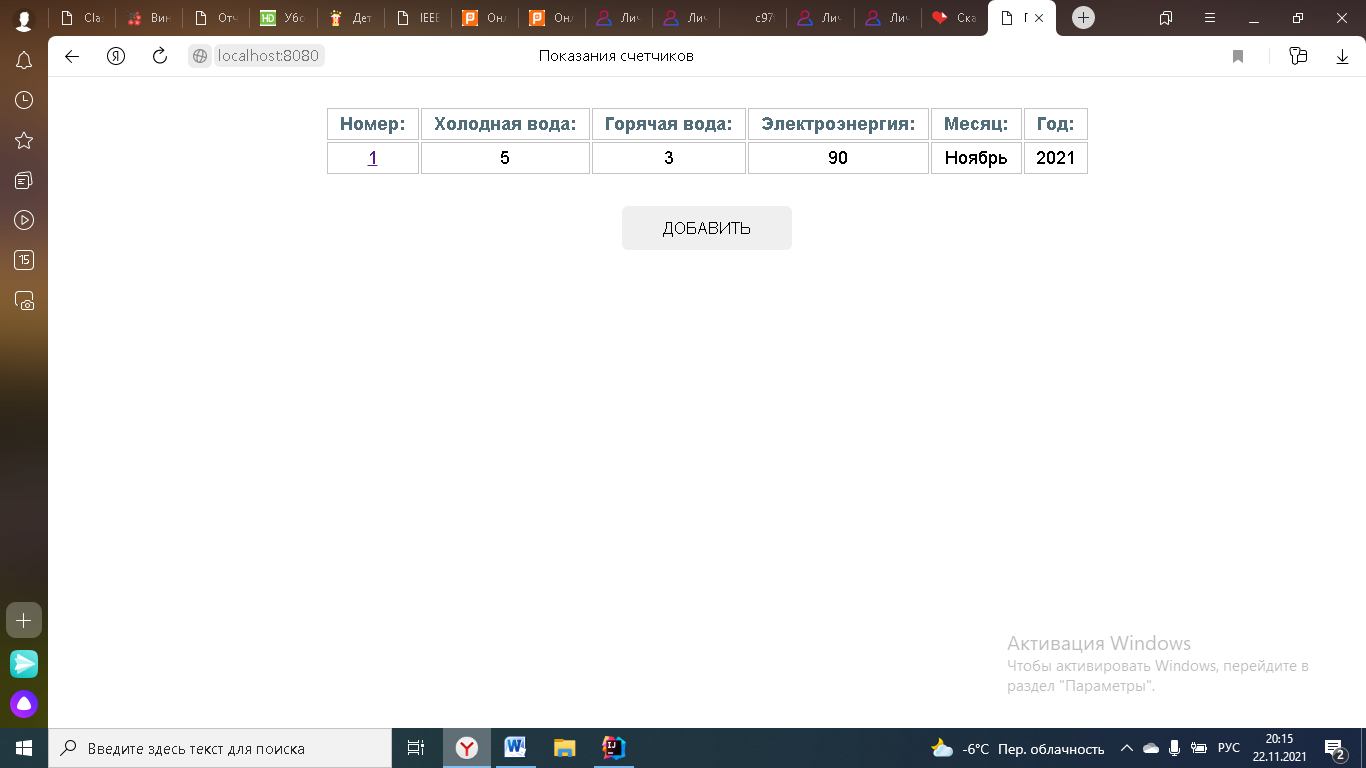


Нажав на кнопку добавить получим ошибку:

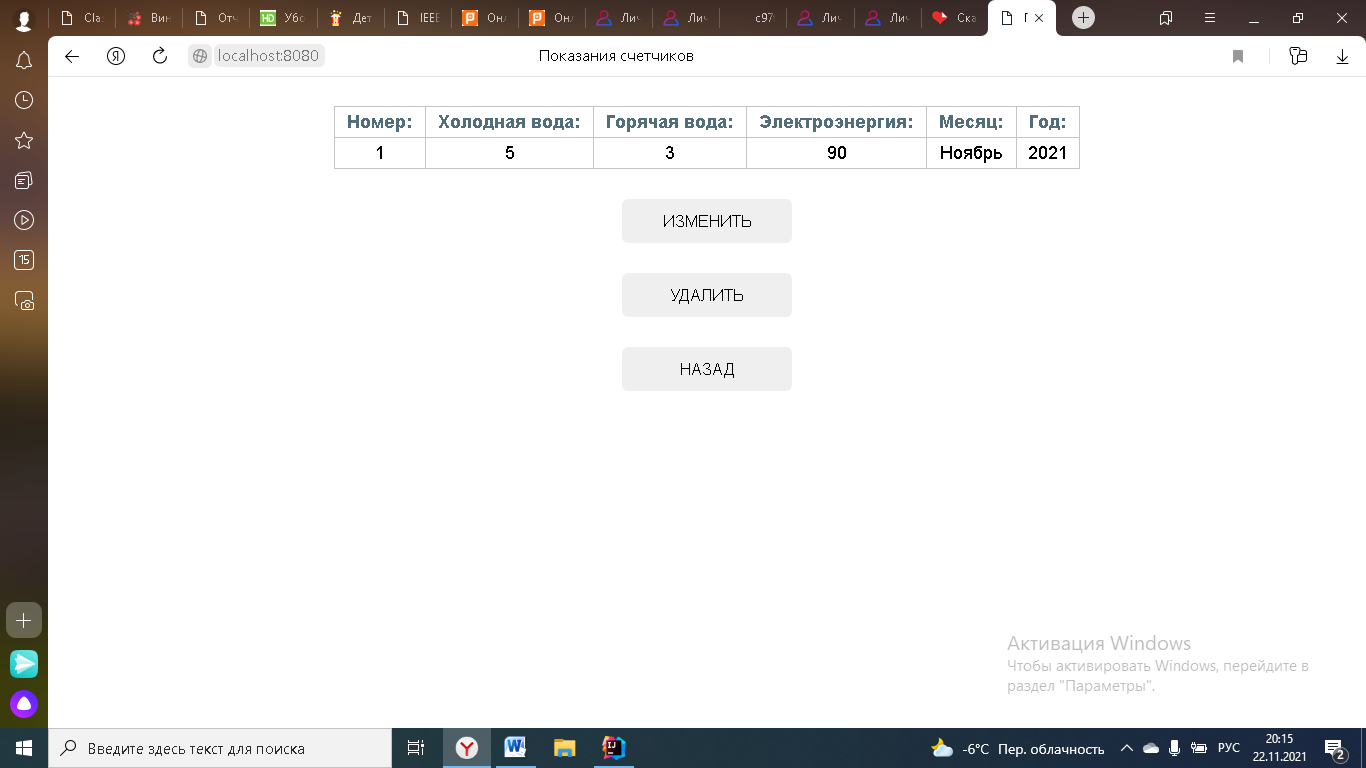


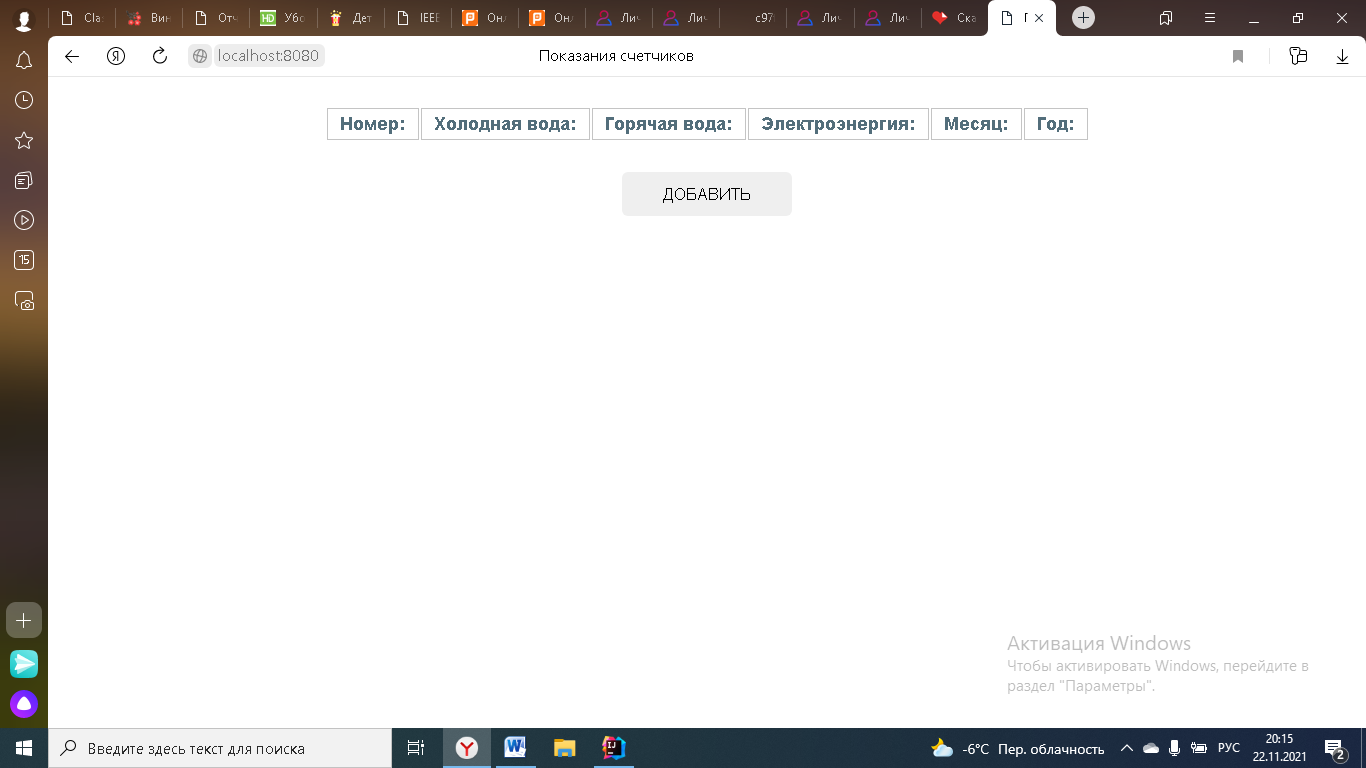
Изменим текущие данные:





Удалим показатели:





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>LAB3\_4</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>LAB3\_4</name>  
 <description>LAB3\_4</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>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-security</artifactId>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

WebSecurityConfig.java

package com.mobest1an.labs.LAB3\_4.config;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;  
import org.springframework.security.config.annotation.web.builders.HttpSecurity;  
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;  
import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;  
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;  
  
@Configuration  
@EnableWebSecurity  
public class WebSecurityConfig {  
 @Configuration  
 public class WebSecurityAdapter extends WebSecurityConfigurerAdapter {  
  
 @Autowired  
 private BCryptPasswordEncoder bCryptPasswordEncoder;  
  
 @Bean  
 public BCryptPasswordEncoder bCryptPasswordEncoder() {  
 return new BCryptPasswordEncoder();  
 }  
  
 @Autowired  
 public void configureGlobal(AuthenticationManagerBuilder auth) throws Exception {  
 auth  
 .inMemoryAuthentication()  
 .withUser("Alena").password(bCryptPasswordEncoder.encode("password123")).roles("ADMIN");  
 auth  
 .inMemoryAuthentication()  
 .withUser("User").password(bCryptPasswordEncoder.encode("password321")).roles("USER");  
 }  
  
 @Override  
 protected void configure(HttpSecurity http) throws Exception {  
 http  
 .httpBasic().disable()  
 .csrf().disable()  
 .authorizeRequests()  
 .antMatchers("/new","/edit","/hcs/delete").hasRole("ADMIN")  
 .anyRequest().authenticated()  
 .and()  
 .formLogin()  
 .permitAll()  
 .defaultSuccessUrl("/")  
 .permitAll()  
 .and()  
 .logout()  
 .invalidateHttpSession(true)  
 .clearAuthentication(true)  
 .logoutSuccessUrl("/login?logout")  
 .permitAll();  
 }  
 }  
  
}

HCSController.java

package com.mobest1an.labs.LAB3\_4.controllers;  
  
import com.mobest1an.labs.LAB3\_4.models.ServiceResponse;  
import com.mobest1an.labs.LAB3\_4.services.MetersDataService;  
import com.mobest1an.labs.LAB3\_4.models.MetersData;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.HttpStatus;  
import org.springframework.http.MediaType;  
import org.springframework.http.ResponseEntity;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.Model;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.List;  
  
@RestController  
@RequestMapping("/hcs")  
@CrossOrigin("/localhost:8080")  
public class HCSController {  
  
 @Autowired  
 private MetersDataService metersDataService;  
  
 @GetMapping  
 public List<MetersData> getAll(){  
 return metersDataService.findAll();  
 }  
  
 @GetMapping("{id}")  
 public MetersData show(@PathVariable("id") int id) {  
 return metersDataService.show(id);  
 }  
  
 @PostMapping("/add")  
 public ResponseEntity<Object> create(@RequestBody MetersData metersData) {  
 metersDataService.save(metersData);  
 ServiceResponse<MetersData> response = new ServiceResponse<MetersData>("success", metersData);  
 return new ResponseEntity<>(response,HttpStatus.*OK*);  
 }  
  
 @PutMapping("/update")  
 public ResponseEntity<Object> update(@RequestBody MetersData metersData) {  
 metersDataService.update(metersData);  
 ServiceResponse<MetersData> response = new ServiceResponse<MetersData>("success", metersData);  
 return new ResponseEntity<>(response,HttpStatus.*OK*);  
 }  
  
}

MainController.java

package com.mobest1an.labs.LAB3\_4.controllers;  
  
import com.mobest1an.labs.LAB3\_4.services.MetersDataService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.\*;  
  
@Controller  
@RequestMapping  
@CrossOrigin("/localhost:8080")  
public class MainController {  
  
  
 @Autowired  
 private MetersDataService metersDataService;  
  
 @GetMapping("/index")  
 public String index() {  
 return "index.html";  
 }  
  
  
 @GetMapping("/new")  
 public String create() {  
 return "new.html";  
 }  
  
 @GetMapping("/show/{id}")  
 public String show(@PathVariable int id) {  
 return "show.html";  
 }  
  
 @GetMapping("/edit/{id}")  
 public String edit(@PathVariable int id) {  
 return "edit.html";  
 }  
  
 @GetMapping("/delete/{id}")  
 public String delete(@PathVariable("id") int id) {  
 metersDataService.delete(id);  
 return "redirect:/";  
 }  
  
  
}

MetersData.java

package com.mobest1an.labs.LAB3\_4.models;  
  
import org.springframework.web.util.pattern.PathPattern;  
  
import java.awt.\*;  
  
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 MetersData(double coldWater, double hotWater, double electricity, String month, String year) {  
  
 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;  
 }  
}

ServiceResponse.java

package com.mobest1an.labs.LAB3\_4.models;  
  
  
  
public class ServiceResponse<T> {  
  
 private String status;  
 private T data;  
  
 public ServiceResponse() {  
 }  
  
 public ServiceResponse(String status, T data) {  
 this.status = status;  
 this.data = data;  
 }  
  
 public String getStatus() {  
 return status;  
 }  
  
 public void setStatus(String status) {  
 this.status = status;  
 }  
  
 public T getData() {  
 return data;  
 }  
  
 public void setData(T data) {  
 this.data = data;  
 }  
  
 @Override  
 public String toString() {  
 return "ServiceResponse{" +  
 "status='" + status + '\'' +  
 ", data=" + data +  
 '}';  
 }  
}

DataServices.java

package com.mobest1an.labs.LAB3\_4.services;  
  
import com.mobest1an.labs.LAB3\_4.models.MetersData;  
  
import java.util.List;  
  
public interface DataServices {  
  
 List<MetersData> findAll();  
  
 MetersData show(int id);  
  
 MetersData save(MetersData metersData);  
  
 MetersData update(MetersData updatedMetersData);  
  
 int delete(int id);  
}

MetersDataService.java

package com.mobest1an.labs.LAB3\_4.services;  
  
import com.mobest1an.labs.LAB3\_4.models.MetersData;  
import org.springframework.stereotype.Service;  
  
import java.util.ArrayList;  
import java.util.List;  
  
@Service  
public class MetersDataService implements DataServices {  
  
 private List<MetersData> metersDataList = new ArrayList<>();  
 private static int *RECORD\_COUNT*=0;  
  
 @Override  
 public List<MetersData> findAll() {  
 return metersDataList;  
 }  
  
 @Override  
 public MetersData show(int id) {  
 return metersDataList.stream().filter(metersData -> metersData.getId() == id).findAny().orElse(null);  
 }  
  
 @Override  
 public MetersData save(MetersData metersData) {  
 metersData.setId(++*RECORD\_COUNT*);  
 metersDataList.add(metersData);  
 return metersData;  
 }  
  
 @Override  
 public MetersData update(MetersData updatedMetersData) {  
 MetersData metersDataToBeUpdated = show(updatedMetersData.getId());  
 metersDataToBeUpdated.setColdWater(updatedMetersData.getColdWater());  
 metersDataToBeUpdated.setHotWater(updatedMetersData.getHotWater());  
 metersDataToBeUpdated.setElectricity(updatedMetersData.getElectricity());  
 metersDataToBeUpdated.setMonth(updatedMetersData.getMonth());  
 metersDataToBeUpdated.setYear(updatedMetersData.getYear());  
 return updatedMetersData;  
 }  
  
 @Override  
 public int delete(int id) {  
 metersDataList.removeIf(metersData -> metersData.getId() == id);  
 return id;  
 }  
}

delete.js

$(***document***).ready(  
 async function () {  
 let value = ***location***.pathname.split('/')[2];  
 let del = ***document***.getElementById('del');  
 del.action = '/delete/' + value;  
 })

Edit.js

$(***document***).ready(  
 async function () {  
 let value = ***location***.pathname.split('/')[2];  
 let id = ***document***.getElementById('id');  
 let coldWater = ***document***.getElementById('coldWater');  
 let hotWater = ***document***.getElementById('hotWater');  
 let electricity = ***document***.getElementById('electricity');  
 let month = ***document***.getElementById('month');  
 let year = ***document***.getElementById('year');  
  
 let route = '/hcs/' + value;  
 let data = await fetch(route)  
 .then((response) => {  
 return response.json();  
 })  
 .then((data) => {  
 ***console***.log(data);  
 return data;  
 });  
  
 id.value = data.id;  
 coldWater.value = data.coldWater;  
 hotWater.value = data.hotWater;  
 electricity.value = data.electricity;  
 year.value = data.year;  
 month.value = data.month;  
 // SUBMIT FORM  
 $("#dataForm").submit(function (event) {  
 // Prevent the form from submitting via the browser.  
 event.preventDefault();  
 ajaxPost();  
 });  
  
 function ajaxPost() {  
  
 // PREPARE FORM DATA  
 formData = {  
 id: $("#id").val(),  
 coldWater: $("#coldWater").val(),  
 hotWater: $("#hotWater").val(),  
 electricity: $("#electricity").val(),  
 month: $("#month").val(),  
 year: $("#year").val()  
 }  
  
 // DO POST  
 $.ajax({  
 type: "PUT",  
 contentType: "application/json",  
 url: "/hcs/update",  
 data: ***JSON***.stringify(formData),  
 dataType: 'json',  
 success: function (result) {  
 if (result.status == "success") {  
 $("#postResultDiv").html(  
 "Запись успешно изменена! <br>");  
 } else {  
 $("#postResultDiv").html("Запись не изменена!");  
 }  
 console.log(result);  
 },  
 error: function (e) {  
 alert("Error!")  
 console.log("ERROR: ", e);  
 }  
 });  
  
 }  
  
 })

getOne.js

$(***document***).ready(async function () {  
 let value = ***location***.pathname.split('/')[2] ;  
 let edit = ***document***.getElementById('edit');  
 edit.action = '/edit/' + value;  
  
  
 let tbodyOne = ***document***.getElementById('tbodyOne');  
  
 ***console***.log("ready!");  
  
 let route = '/hcs/' + value;  
 let data = await fetch(route)  
 .then((response) => {  
 return response.json();  
 })  
 .then((data) => {  
 ***console***.log(data);  
 return data;  
 });  
  
  
 tbodyOne.innerHTML += `<tr><td>${data.id}</td><td>${data.coldWater}</td><td>${data.hotWater}</td><td>${data.electricity}</td><td>${data.month}</td><td>${data.year}</td></tr>`  
  
});

Getrequest.js

$( ***document*** ).ready(async function () {  
 let id = ***document***.getElementById('id');  
 let coldWater = ***document***.getElementById('coldWater');  
 let hotWater = ***document***.getElementById('hotWater');  
 let electricity = ***document***.getElementById('electricity');  
 let month = ***document***.getElementById('month');  
 let year = ***document***.getElementById('year');  
 let tbody = ***document***.getElementById('tbody');  
  
  
 ***console***.log("ready!");  
 let data = await fetch('/hcs/')  
 .then((response) => {  
 return response.json();  
 })  
 .then((data) => {  
 return data;  
 });  
  
  
  
 for (let i=0 ; i < data.length ; i++){  
 tbody.innerHTML += `<tr><td><a href="show/` + data[i].id + `">${data[i].id}</a></td><td>${data[i].coldWater}</td><td>${data[i].hotWater}</td><td>${data[i].electricity}</td><td>${data[i].month}</td><td>${data[i].year}</td></tr>`  
 }  
});

Postrequest.js

$(***document***).ready(  
 function() {  
 // SUBMIT FORM  
 $("#dataForm").submit(function(event) {  
 // Prevent the form from submitting via the browser.  
 event.preventDefault();  
 ajaxPost();  
 });  
  
 function ajaxPost() {  
  
 // PREPARE FORM DATA  
 formData = {  
 coldWater : $("#coldWater").val(),  
 hotWater : $("#hotWater").val(),  
 electricity : $("#electricity").val(),  
 month : $("#month").val(),  
 year : $("#year").val()  
 }  
  
 // DO POST  
 $.ajax({  
 type : "POST",  
 contentType : "application/json",  
 url : "hcs/add",  
 data : ***JSON***.stringify(formData),  
 dataType : 'json',  
 success : function(result) {  
 if (result.status == "success") {  
 $("#postResultDiv").html(  
 "Запись успешно добавлена! <br>");  
 } else {  
 $("#postResultDiv").html("Запись не добавлена!");  
 }  
 ***console***.log(result);  
 },  
 error : function(e) {  
 alert("Error!")  
 ***console***.log("ERROR: ", e);  
 }  
 });  
  
 }  
  
 })

Edit.html

<!DOCTYPE html>  
<html lang="en" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>Показания счетчиков</title>  
 <link href="/style.css" rel="stylesheet" />  
 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>  
 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.0/umd/popper.min.js"></script>  
 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/js/bootstrap.min.js"></script>  
</head>  
<body>  
<div id="postResultDiv" align="center"></div>  
<div class="container-main">  
 <form id="dataForm">  
 <div class="cold-water">  
 <label for="id">id</label>  
 <br>  
 <input type="text" id="id" placeholder="идентификтор" name="id" disabled>  
 </div>  
 <div class="cold-water">  
 <label for="coldWater">Введите расход холодной воды за месяц (куб м)</label>  
 <br>  
 <input type="text" id="coldWater" placeholder="Холодная вода" name="coldWater">  
 </div>  
 <div class="hot-water">  
 <label for="hotWater">Введите расход горячей воды за месяц (куб м)</label>  
 <br>  
 <input type="text" id="hotWater" placeholder="Горячая вода" name="hotWater">  
 </div>  
 <div class="electricity">  
 <label for="electricity">Введите расход электричества за месяц (кВтч)</label>  
 <br>  
 <input type="text" id="electricity" placeholder="Электроэнергия" name="electricity">  
 </div>  
 <div class="month">  
 <select id="month" name="month">  
 <option value="Январь" text="Январь">Январь</option>  
 <option value="Февраль" text="Февраль">Февраль</option>  
 <option value="Март" text="Март">Март</option>  
 <option value="Апрель" text="Апрель">Апрель</option>  
 <option value="Май" text="Май">Май</option>  
 <option value="Июнь" text="Июнь">Июнь</option>  
 <option value="Июль" text="Июль">Июль</option>  
 <option value="Август" text="Август">Август</option>  
 <option value="Сентябрь" text="Сентябрь">Сентябрь</option>  
 <option value="Октябрь" text="Октябрь">Октябрь</option>  
 <option value="Ноябрь" text="Ноябрь">Ноябрь</option>  
 <option value="Декабрь" text="Декабрь">Декабрь</option>  
 </select>  
 </div>  
 <div class="year">  
 <select id="year" name="year">  
 <option value="2021" text="2021">2021</option>  
 <option value="2020" text="2020">2020</option>  
 <option value="2019" text="2019">2019</option>  
 <option value="2018" text="2018">2018</option>  
 <option value="2017" text="2017">2017</option>  
 <option value="2016" text="2016">2016</option>  
 <option value="2015" text="2015">2015</option>  
 <option value="2014" text="2014">2014</option>  
 <option value="2013" text="2013">2013</option>  
 <option value="2012" text="2012">2012</option>  
 <option value="2011" text="2011">2011</option>  
 <option value="2010" text="2010">2010</option>  
 </select>  
 </div>  
 <div class="buttons">  
 <button type="submit" class="btn btn-default">Изменить</button>  
 </div>  
 </form>  
  
 <form th:method="GET" th:action="@{/index}">  
 <input class="btn btn-submit" type="submit" value="Назад">  
 </form>  
  
</div>  
<script src="/edit.js"></script>  
</body>  
</html>

Login.html

<!DOCTYPE html>  
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="https://www.thymeleaf.org"  
 xmlns:sec="https://www.thymeleaf.org/thymeleaf-extras-springsecurity3">  
<head>  
 <title>Spring Security Example </title>  
</head>  
<body>  
<div th:if="${param.error}">  
 Invalid username and password.  
</div>  
<div th:if="${param.logout}">  
 You have been logged out.  
</div>  
<form th:action="@{/login}" method="post">  
 <div><label> User Name : <input type="text" name="username"/> </label></div>  
 <div><label> Password: <input type="password" name="password"/> </label></div>  
 <div><input type="submit" value="Sign In"/></div>  
</form>  
</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 href="style.css" rel="stylesheet" />  
 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>  
 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.0/umd/popper.min.js"></script>  
 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/js/bootstrap.min.js"></script>  
</head>  
<body >  
<div class="container-main d-flex justify-content-center" >  
  
 <table id="getResultDiv" style="margin: 0 auto">  
  
 <tr id="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>  
 <tbody id="tbody">  
 </tbody>  
 </table>  
  
 <form th:method="GET" th:action="@{/new}">  
 <input class="btn btn-submit" type="submit" value="Добавить">  
 </form>  
</div>  
  
<script src="getrequest.js"></script>  
</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 href="style.css" rel="stylesheet"/>  
 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>  
 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.0/umd/popper.min.js"></script>  
 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/js/bootstrap.min.js"></script>  
 <script src="postrequest.js"></script>  
</head>  
<body>  
<div id="postResultDiv" align="center"></div>  
<div class="container-main">  
 <form id="dataForm">  
 <div class="cold-water">  
 <label for="coldWater">Введите расход холодной воды за месяц (куб м)</label>  
 <br>  
 <input type="text" id="coldWater" placeholder="Холодная вода" name="coldWater">  
 </div>  
 <div class="hot-water">  
 <label for="hotWater">Введите расход горячей воды за месяц (куб м)</label>  
 <br>  
 <input type="text" id="hotWater" placeholder="Горячая вода" name="hotWater">  
 </div>  
 <div class="electricity">  
 <label for="electricity">Введите расход электричества за месяц (кВтч)</label>  
 <br>  
 <input type="text" id="electricity" placeholder="Электроэнергия" name="electricity">  
 </div>  
 <div class="month">  
 <select id="month" name="month">  
 <option value="Январь" text="Январь">Январь</option>  
 <option value="Февраль" text="Февраль">Февраль</option>  
 <option value="Март" text="Март">Март</option>  
 <option value="Апрель" text="Апрель">Апрель</option>  
 <option value="Май" text="Май">Май</option>  
 <option value="Июнь" text="Июнь">Июнь</option>  
 <option value="Июль" text="Июль">Июль</option>  
 <option value="Август" text="Август">Август</option>  
 <option value="Сентябрь" text="Сентябрь">Сентябрь</option>  
 <option value="Октябрь" text="Октябрь">Октябрь</option>  
 <option value="Ноябрь" text="Ноябрь">Ноябрь</option>  
 <option value="Декабрь" text="Декабрь">Декабрь</option>  
 </select>  
 </div>  
 <div class="year">  
 <select id="year" name="year">  
 <option value="2021" text="2021">2021</option>  
 <option value="2020" text="2020">2020</option>  
 <option value="2019" text="2019">2019</option>  
 <option value="2018" text="2018">2018</option>  
 <option value="2017" text="2017">2017</option>  
 <option value="2016" text="2016">2016</option>  
 <option value="2015" text="2015">2015</option>  
 <option value="2014" text="2014">2014</option>  
 <option value="2013" text="2013">2013</option>  
 <option value="2012" text="2012">2012</option>  
 <option value="2011" text="2011">2011</option>  
 <option value="2010" text="2010">2010</option>  
 </select>  
 </div>  
 <div class="buttons">  
 <button type="submit" class="btn btn-default">Добавить</button>  
 </div>  
 </form>  
  
 <form th:method="GET" th:action="@{/index}">  
 <input class="btn btn-submit" type="submit" value="Назад">  
 </form>  
</div>  
  
</body>  
</html>

Show.html

<!DOCTYPE html>  
<html lang="en" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>Показания счетчиков</title>  
 <link href="/style.css" rel="stylesheet"/>  
 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>  
 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.0/umd/popper.min.js"></script>  
 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/js/bootstrap.min.js"></script>  
</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>  
 <tbody id="tbodyOne">  
 </tbody>  
 </table>  
  
 <div class="buttons">  
 <form th:method="GET" id="edit">  
 <input class="btn btn-submit" type="submit" value="Изменить">  
 </form>  
  
 <form th:method="GET" id="del" >  
 <input class="btn btn-submit" type="submit" value="Удалить">  
 </form>  
  
 <form th:method="GET" th:action="@{/index}">  
 <input class="btn btn-submit" type="submit" value="Назад">  
 </form>  
 </div>  
  
</div>  
<script src="/getOne.js"></script>  
<script src="/delete.js"></script>  
</body>  
</html>

Вывод: В ходе работы были получены и применены на практике основные навыки защиты информации средствами SpringSecurity.