МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ИНСТИТУТ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И ПРОГРАММИРОВАНИЯ

КАФЕДРА компьютерных технологий и программной инженерии

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | Кочин Д.А. |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2 |
| «Разработка ресурса REST/JSON сервиса» |
| по курсу: Технологии разработки серверных информационных систем |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4932 |  | 15.09.21 |  | А.И.Белов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

1. **Задание:**

Подключите к проекту Maven фреймворк Spring (spring-boot), определите перечень Rest-сервисов, выполняющих те же действия, что и в лабораторной работе 1. Внимательно отнеситесь к вопросу какой HTTP метод использует тот или иной сервис и какие коды HTTP он может возвращать. Реализуйте эти сервисы

Вариант: 2. Поликлиника (запись на прием к врачу)

1. **Описание разрабатываемого продукта:**

Веб-приложение для организации управления поликлиникой. Функционал: добавление, удаление, изменение информации о врачах и пациентах, организация записей.

Формы:

Главное окно: Изображение выглядит как текст

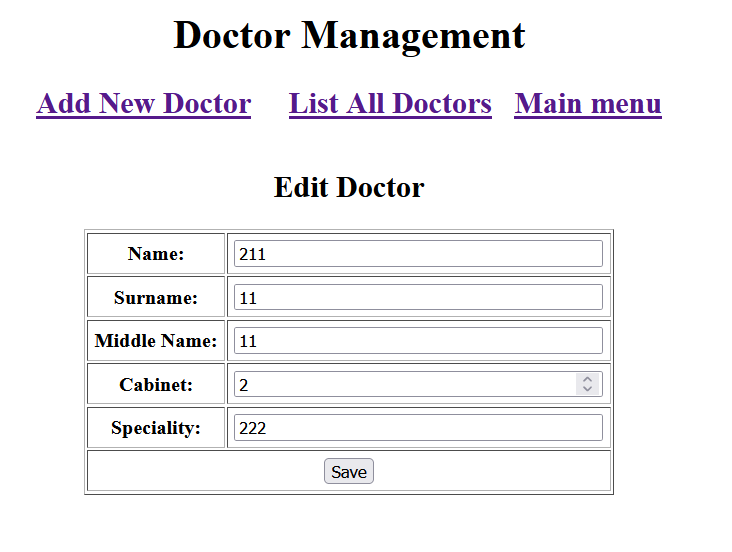
Автоматически созданное описание

Список врачей:

Изображение выглядит как стол

Автоматически созданное описание

Форма добавления/редактирования врача:

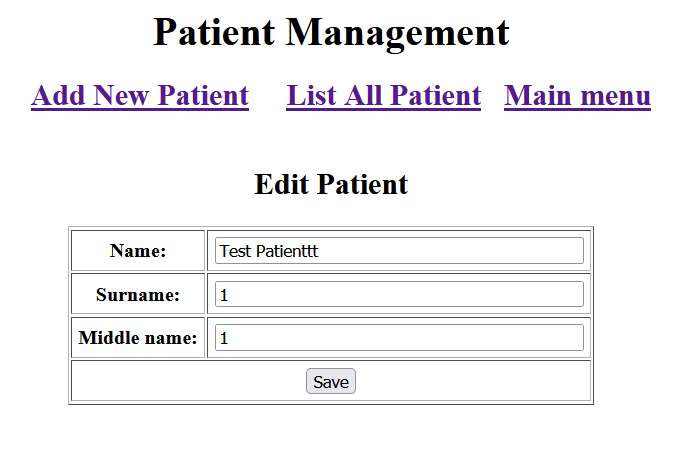


Список пациентов:

Изображение выглядит как стол

Автоматически созданное описание

SФорма добавления/редактирования пациента:

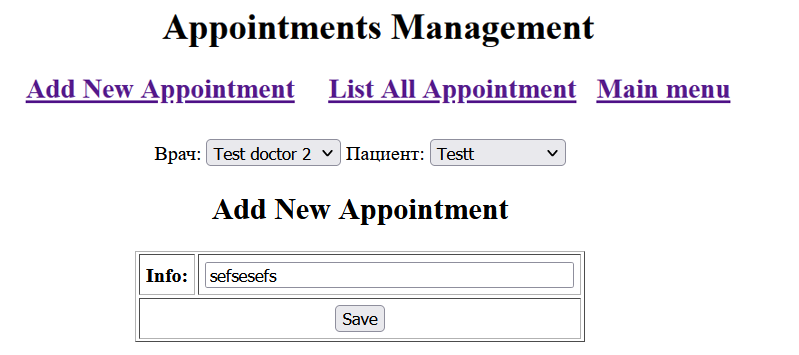


Список записей:

Изображение выглядит как стол

Автоматически созданное описание

Форма добавления записи:



1. **Листинг:**

// @filename \src\main\java\api\lab\Models\Doctor.java

package api.lab.Models;

public class Doctor {

public Integer id;

public String name;

public String surname;

public String midname;

public String speciality;

public Integer cabinet;

public Integer getId(){

return id;

}

public String getName(){

return name;

}

public String getSpeciality() {

return speciality;

}

public Integer getCabinet() {

return cabinet;

}

public void setId(Integer id) {

this.id = id;

}

public void setName(String name) {

this.name = name;

}

public String getSurname() {

return surname;

}

public void setSurname(String surname) {

this.surname = surname;

}

public String getMidname() {

return midname;

}

public void setMidname(String midname) {

this.midname = midname;

}

public void setSpeciality(String speciality) {

this.speciality = speciality;

}

public void setCabinet(Integer cabinet) {

this.cabinet = cabinet;

}

}

// @filename \src\main\java\api\lab\Models\Doctor\_Patient.java

package api.lab.Models;

import java.util.Date;

public class Doctor\_Patient {

public Integer id;

public Doctor doc;

public Patient pat;

public String info;

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public Doctor getDoc() {

return doc;

}

public Patient getPat() {

return pat;

}

public String getInfo() {

return info;

}

public void setDoc(Doctor doc) {

this.doc = doc;

}

public void setPat(Patient pat) {

this.pat = pat;

}

public void setInfo(String info) {

this.info = info;

}

}

// @filename \src\main\java\api\lab\Models\Patient.java

package api.lab.Models;

import java.util.Date;

public class Patient {

public int id;

public String name;

public String surname;

public String midname;

public Integer getId(){

return id;

}

public String getName(){

return name;

}

public void setId(int id) {

this.id = id;

}

public void setName(String name) {

this.name = name;

}

public String getSurname() {

return surname;

}

public void setSurname(String surname) {

this.surname = surname;

}

public String getMidname() {

return midname;

}

public void setMidname(String midname) {

this.midname = midname;

}

}

// @filename \src\main\java\api\lab1\AppointmentDAO.java

package api.lab1;

import api.lab.Models.Doctor;

import api.lab.Models.Doctor\_Patient;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

public class AppointmentDAO {

public List<Doctor\_Patient> All() throws SQLException {

List<Doctor\_Patient> result = new ArrayList<>();

String command = "SELECT \* FROM [Doctor\_Patient]";

Statement statement = SQLCon.getSqlConnection().createStatement();

ResultSet resultSet = statement.executeQuery(command);

while(resultSet.next()){

Doctor\_Patient newAppointment = new Doctor\_Patient();

newAppointment.id= resultSet.getInt("Id");

newAppointment.pat= new PatientDAO().Get(resultSet.getInt("Id\_Patient"));

newAppointment.doc= new DoctorDAO().Get(resultSet.getInt("Id\_Doctor"));

newAppointment.info = resultSet.getString("Info");

result.add(newAppointment);

}

return result;

};

public boolean Add(Doctor\_Patient doctor\_patient) throws SQLException {

String command = "INSERT INTO Doctor\_Patient (Id\_Doctor, Id\_Patient, Info) VALUES (?,?,?)";

PreparedStatement statement = SQLCon.getSqlConnection().prepareStatement(command);

statement.setInt(1, doctor\_patient.doc.id);

statement.setInt(2, doctor\_patient.pat.id);

statement.setString(3, doctor\_patient.info);

boolean result = statement.executeUpdate()>0;

return result;

};

public boolean Delete(Doctor\_Patient doctor\_patient) throws SQLException {

String command = "DELETE FROM Doctor\_Patient WHERE Id=?";

PreparedStatement statement = SQLCon.getSqlConnection().prepareStatement(command);

statement.setInt(1,doctor\_patient.id);

boolean result = statement.executeUpdate()>0;

return result;

};

}

// @filename \src\main\java\api\lab1\DoctorDAO.java

package api.lab1;

import api.lab.Models.Doctor;

import api.lab.Models.Patient;

import javax.print.Doc;

import java.sql.\*;

import java.util.ArrayList;

import java.util.List;

public class DoctorDAO {

public Doctor Get(int id) throws SQLException {

Doctor result = new Doctor();

String command = "SELECT \* FROM Doctor WHERE Id="+id;

Statement statement = SQLCon.getSqlConnection().createStatement();

ResultSet resultSet = statement.executeQuery(command);

if(resultSet.next()){

Doctor newDoctor = getDoctorFromResultSet(resultSet);

result = newDoctor;

}

return result;

};

public List<Doctor> All() throws SQLException {

List<Doctor> result = new ArrayList<>();

String command = "SELECT \* FROM [Doctor]";

Statement statement = SQLCon.getSqlConnection().createStatement();

ResultSet resultSet = statement.executeQuery(command);

while(resultSet.next()){

Doctor newDoctor = getDoctorFromResultSet(resultSet);

result.add(newDoctor);

}

return result;

};

public boolean Add(Doctor doctor) throws SQLException {

String command = "INSERT INTO Doctor (Name, Cabinet, Speciality, Surname, Midname) VALUES (?,?,?,?,?)";

PreparedStatement statement = getStatWithDoctor(doctor,command);

boolean result = statement.executeUpdate()>0;

return result;

};

public boolean Edit(Doctor doctor) throws SQLException {

String command = "UPDATE Doctor SET Name=?, Cabinet=?, Speciality=?, Surname=?, Midname=? WHERE Id=?";

PreparedStatement statement = getStatWithDoctor(doctor, command);

statement.setInt(6, doctor.id);

boolean result = statement.executeUpdate()>0;

return result;

};

public boolean Delete(Doctor doctor) throws SQLException {

String command = "DELETE FROM Doctor WHERE Id=?";

PreparedStatement statement = SQLCon.getSqlConnection().prepareStatement(command);

statement.setInt(1, doctor.id);

boolean result = statement.executeUpdate()>0;

return result;

};

public PreparedStatement getStatWithDoctor(Doctor doctor, String command) throws SQLException {

PreparedStatement statement = SQLCon.getSqlConnection().prepareStatement(command);

statement.setString(1, doctor.name);

statement.setInt(2, doctor.cabinet);

statement.setString(3,doctor.speciality);

statement.setString(4, doctor.surname);

statement.setString(5, doctor.midname);

return statement;

};

public Doctor getDoctorFromResultSet(ResultSet resultSet) throws SQLException {

Doctor doctor = new Doctor();

doctor.id=resultSet.getInt("Id");

doctor.name=resultSet.getString("Name");

doctor.cabinet=resultSet.getInt("Cabinet");

doctor.speciality=resultSet.getString("Speciality");

doctor.surname=resultSet.getString("Surname");

doctor.midname=resultSet.getString("Midname");

return doctor;

};

}

// @filename \src\main\java\api\lab1\PatientDAO.java

package api.lab1;

import api.lab.Models.Patient;

import java.sql.\*;

import java.util.ArrayList;

import java.util.List;

public class PatientDAO {

public Patient Get(int id) throws SQLException {

Patient result = new Patient();

String command = "SELECT \* FROM Patient WHERE Id="+id;

Statement statement = SQLCon.getSqlConnection().createStatement();

ResultSet resultSet = statement.executeQuery(command);

if(resultSet.next()){

Patient newPatient = getPatientFromResultSet(resultSet);

result = newPatient;

}

return result;

};

public List<Patient> All() throws SQLException {

List<Patient> result = new ArrayList<>();

String command = "SELECT \* FROM [Patient]";

Statement statement = SQLCon.getSqlConnection().createStatement();

ResultSet resultSet = statement.executeQuery(command);

while(resultSet.next()){

Patient newPatient = getPatientFromResultSet(resultSet);

result.add(newPatient);

}

return result;

};

public boolean Add(Patient patient) throws SQLException {

String command = "INSERT INTO Patient (Name, Surname, Midname) VALUES (?,?,?)";

PreparedStatement statement = getStatFromPatient(patient, command);

boolean result = statement.executeUpdate()>0;

return result;

};

public boolean Edit(Patient patient) throws SQLException {

String command = "UPDATE Patient SET Name=?, Surname=?, Midname=? WHERE Id=?";

PreparedStatement statement = getStatFromPatient(patient, command);

statement.setInt(4, patient.id);

boolean result = statement.executeUpdate()>0;

return result;

};

public boolean Delete(Patient patient) throws SQLException {

String command = "DELETE FROM Patient WHERE Id=?";

PreparedStatement statement = SQLCon.getSqlConnection().prepareStatement(command);

statement.setInt(1,patient.id);

boolean result = statement.executeUpdate()>0;

return result;

};

public Patient getPatientFromResultSet(ResultSet resultSet) throws SQLException {

Patient patient = new Patient();

patient.id=resultSet.getInt("Id");

patient.name=resultSet.getString("Name");

patient.midname=resultSet.getString("Midname");

patient.surname=resultSet.getString("Surname");

return patient;

}

public PreparedStatement getStatFromPatient(Patient patient, String command) throws SQLException {

PreparedStatement statement = SQLCon.getSqlConnection().prepareStatement(command);

statement.setString(1, patient.name);

statement.setString(2, patient.surname);

statement.setString(3, patient.midname);

return statement;

}

}

// @filename \src\main\java\api\lab1\SQLCon.java

package api.lab1;

import java.sql.\*;

public class SQLCon {

static String connectionString = "jdbc:sqlserver://127.0.0.1\\SQLEXPRESS;databaseName=PolyclinicDB;username=sa;password=test";

public SQLCon() {

String connectionUrl = "jdbc:sqlserver://127.0.0.1\\SQLEXPRESS;databaseName=PolyclinicDB;username=sa;password=test";

this.connectionString=connectionUrl;

try {

Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

}

public static Connection getSqlConnection() throws SQLException {

String connectionString = "jdbc:sqlserver://127.0.0.1\\SQLEXPRESS;databaseName=PolyclinicDB;username=sa;password=test";

try {

Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

Connection connection = DriverManager.getConnection(connectionString);

return connection;

}

}

// @filename \src\main\java\api\lab1\AppointmentController.java

package com.example.lab2\_3;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.\*;

import javax.print.Doc;

import java.sql.SQLException;

import java.util.List;

@RestController

public class AppointmentController {

AppointmentDAO appointmentDAO = new AppointmentDAO();

@GetMapping("appointments/all")

public List<Doctor\_Patient> all() throws SQLException{

List<Doctor\_Patient> list = appointmentDAO.All();

return list;

}

@PostMapping(value = "appointments/post")

public ResponseEntity post(@RequestBody Doctor\_Patient doctor\_patient) throws SQLException {

doctor\_patient.doc=new DoctorDAO().Get(doctor\_patient.iddoc);

doctor\_patient.pat=new PatientDAO().Get(doctor\_patient.idpat);

appointmentDAO.Add(doctor\_patient);

return ResponseEntity.status(HttpStatus.CREATED).body(null);

}

@DeleteMapping(value="appointments/delete{id}")

public ResponseEntity delete(@PathVariable Integer id) throws SQLException {

Doctor\_Patient doctor\_patient = new Doctor\_Patient();

doctor\_patient.id=id;

appointmentDAO.Delete(doctor\_patient);

return ResponseEntity.status(HttpStatus.ACCEPTED).body(null);

}

}

// @filename \src\main\java\api\lab1\DoctorController.java

package com.example.lab2\_3;

import org.springframework.http.HttpStatus;

import org.springframework.http.MediaType;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.\*;

import java.sql.SQLException;

import java.util.List;

@RestController

public class DoctorController {

DoctorDAO doctorDAO = new DoctorDAO();

@RequestMapping(value = "doctors/all", method = RequestMethod.GET)

public List<Doctor> all() throws SQLException{

List<Doctor> list = doctorDAO.All();

return list;

}

@GetMapping(value = "doctors/get{id}")

public Doctor get(@PathVariable int id) throws SQLException {

Doctor doc = doctorDAO.Get(id);

return doc;

}

@PostMapping(value = "doctors/post")

public ResponseEntity post(@RequestBody Doctor doctor) throws SQLException {

doctorDAO.Add(doctor);

List<Doctor> items = doctorDAO.All();

return ResponseEntity.status(HttpStatus.ACCEPTED).body(items.get(items.size() - 1).id);

}

@PutMapping(value="doctors/edit", consumes = MediaType.APPLICATION\_JSON\_VALUE,

produces = MediaType.APPLICATION\_JSON\_VALUE)

public ResponseEntity edit(@RequestBody Doctor doctor) throws SQLException {

doctorDAO.Edit(doctor);

return ResponseEntity.status(HttpStatus.ACCEPTED).body(null);

}

@DeleteMapping(value="doctors/delete{id}")

public ResponseEntity delete(@PathVariable Integer id) throws SQLException {

doctorDAO.Delete(doctorDAO.Get(id));

return ResponseEntity.status(HttpStatus.ACCEPTED).body(null);

}

}

// @filename \src\main\java\api\lab1\PatientController.java

package com.example.lab2\_3;

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.sql.SQLException;

import java.util.List;

@RestController

public class PatientController {

PatientDAO patientDAO = new PatientDAO();

@RequestMapping(value = "patients/all", method = RequestMethod.GET)

public List<Patient> all() throws SQLException{

List<Patient> list = patientDAO.All();

return list;

}

@GetMapping(value = "patients/get{id}")

public Patient get(@PathVariable Integer id) throws SQLException {

Patient patient = patientDAO.Get(id);

return patient;

}

@PostMapping(value = "patients/post")

public ResponseEntity post(@RequestBody Patient patient) throws SQLException {

patientDAO.Add(patient);

List<Patient> items = patientDAO.All();

return ResponseEntity.status(HttpStatus.CREATED).body(items.get(items.size() - 1).id);

}

@PutMapping(value="patients/edit", consumes = MediaType.APPLICATION\_JSON\_VALUE,

produces = MediaType.APPLICATION\_JSON\_VALUE)

public ResponseEntity edit(@RequestBody Patient patient) throws SQLException {

patientDAO.Edit(patient);

return ResponseEntity.status(HttpStatus.ACCEPTED).body(null);

}

@DeleteMapping(value="patients/delete{id}")

public ResponseEntity delete(@PathVariable Integer id) throws SQLException {

patientDAO.Delete(patientDAO.Get(id));

return ResponseEntity.status(HttpStatus.ACCEPTED).body(null);

}

}}

// @filename \src\main\java\api\lab1\WebController.java

package com.example.lab2\_3;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

@Controller

public class WebController {

@GetMapping("/")

public String MainPage() {

return "MainMenu";

}

@GetMapping("doctors")

public String DocIndex() {

return "DoctorIndex";

}

@GetMapping("doctors/view{id}")

public String DocGet(@PathVariable int id, Model model){

model.addAttribute("docId", id);

return "DoctorForm";

}

@GetMapping("patients")

public String PatIndex() {

return "PatientIndex";

}

@GetMapping("patients/view{id}")

public String PatGet(@PathVariable int id, Model model){

model.addAttribute("patId", id);

return "PatientForm";

}

@GetMapping("appointments")

public String AppIndex() {

return "AppointmentsIndex";

}

@GetMapping("appointments/view")

public String AppGet(){

return "AppointmentForm";

}

}

// @filename \src\resources\templates\MainMenu.html

<!DOCTYPE HTML>

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Getting Started: Serving Web Content</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

</head>

<body>

<center>

<h1>Поликлиника имени плохого кода"</h1>

<h2>

</h2>

<table border="1" cellpadding="5">

<caption>

<h2>

Main Menu

</h2>

</caption>

<tr>

<td>

<a href="/appointments">Записи к врачу</a>

</td>

<td>

<a href="/doctors">Врачи</a>

</td>

<td>

<a href="/patients">Пациенты</a>

</td>

</tr>

</table>

</center>

</body>

</html>

// @filename \src\resources\templates\AppointmentIndex.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<script src="JQuery.js"></script>

</head>

<script type="text/javascript">

$(document).ready(function (){

refresh();

})

function refresh(){

$.get('appointments/all',function (data){

console.log(data);

let table = "<table border='1'>";

table = table+"<caption><h2>List of Appointment</h2></caption>\n" +

" <tr>\n" +

" <th>Name Doctor</th>\n" +

" <th>Name Patient</th>\n" +

" <th>Info</th>\n" +

" </tr>";

for(i=0; i<data.length; i++){

//table=table+"<tr><td>"+data[i].name+"</td><td>"+data[i].surname+"</td><td><button onclick='deleteDoc("+data[i].id+")'>DELETE</button></td></tr>";

table = table +"<tr>\n" +

" <td>"+data[i].doc.name+"</td>\n" +

" <td>"+data[i].pat.name+"</td>\n" +

" <td>"+data[i].info+"</td>\n" +

" <td>\n" +

" <button onclick='deleteApp("+data[i].id+")'>Delete</button>\n" +

" </td>\n" +

" </tr>";

}

table = table + "</table>";

$("#test\_app").html(table);

})

}

function deleteApp(idDel){

$.ajax({

url:'appointments/delete'+idDel,

type: "DELETE",

dataType: "json",

contentType: "application/json; charset=utf-8",

success: function (){

refresh();

},

error: function (){

console.log("fuck");

}

})

}

</script>

<body>

<center>

<h1>App Management</h1>

<h2>

<a th:href="@{/appointments/view-1}">Add New Doctor</a>

&nbsp;&nbsp;&nbsp;

<a th:href="@{/doctors}"></a>

&nbsp;

<a th:href="@{/}">Main menu</a>

</h2>

</center>

<div align="center" id="test\_app">

<form>

<input type="">

</form>

</div>

</body>

</html>

// @filename \src\resources\templates\AppointmentForm.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<script src="JQuery.js"></script>

</head>

<script type="text/javascript">

$(document).ready(function (){

getDocs();

getPats();

})

function getDocs(){

$.get('/doctors/all', function (data){

var select = document.getElementById('iddoc');

$(select).html('');

for(i=0; i<data.length; i++){

$(select).append('<option value=' + data[i].id + '>' + data[i].name + '</option>');

}

})

}

function getPats(){

$.get('/patients/all', function (data){

var select = document.getElementById('idpat');

$(select).html('');

for(i=0; i<data.length; i++){

$(select).append('<option value=' + data[i].id + '>' + data[i].name + '</option>');

}

})

}

function saveData() {

console.log("aa");

$.ajax({

url: "/appointments/post/",

dataType: "json",

contentType: "application/json",

type: "POST",

cache: false,

data: JSON.stringify({

idpat: $("#idpat").val(),

iddoc: $("#iddoc").val(),

info: $("#txtInfo").val()

}),

success: function () {

console.log("bbb");

},

error: function (error) {

console.log(error);

}

})

}

</script>

<body>

<div align="center" id ="formdoc"></div>

<form>

<table border="1" cellpadding="5">

<caption>

<h2>Appointment</h2>

</caption>

<tr>

<th>Doctor: </th>

<td>

<select size=1 name="iddoc" id="iddoc" required="required"></select>

</td>

</tr>

<tr>

<th>Patient: </th>

<td>

<select size=1 name="idpat" id="idpat" required="required"></select>

</td>

</tr>

<tr>

<th>Middle Name: </th>

<td>

<input type="text" name="info" size="45" id="txtInfo" />

</td>

</tr>

<tr>

<td colspan="2" align="center"><button type="submit" value="save" onclick='saveData()'>Save</button>

</table>

</form>

</body>

</html>

// @filename \src\resources\templates\DoctorIndex.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<script src="JQuery.js"></script>

</head>

<script type="text/javascript">

$(document).ready(function (){

refresh();

})

function getDoc(idGet){

window.location.href='/doctors/view'+idGet;

}

function refresh(){

$.get('doctors/all',function (data){

console.log(data);

let table = "<table border='1'>";

table = table+"<caption><h2>List of Doctors</h2></caption>\n" +

" <tr>\n" +

" <th>Name</th>\n" +

" <th>Surname</th>\n" +

" <th>Middle Name</th>\n" +

" <th>Cabinet</th>\n" +

" <th>Speciality</th>\n" +

" </tr>";

for(i=0; i<data.length; i++){

//table=table+"<tr><td>"+data[i].name+"</td><td>"+data[i].surname+"</td><td><button onclick='deleteDoc("+data[i].id+")'>DELETE</button></td></tr>";

table = table +"<tr>\n" +

" <td>"+data[i].name+"</td>\n" +

" <td>"+data[i].surname+"</td>\n" +

" <td>"+data[i].midname+"</td>\n" +

" <td>"+data[i].cabinet+"</td>\n" +

" <td>"+data[i].speciality+"</td>\n" +

" <td>\n" +

" <button onclick='getDoc("+data[i].id+")'>Edit</button>\n"+

// " <button th:formaction=\"@{/doctors/get{id}(id=${doctor.id})}\">Edit</button>\n" +

" &nbsp;&nbsp;&nbsp;&nbsp;\n" +

" <button onclick='deleteDoc("+data[i].id+")'>Delete</button>\n" +

" </td>\n" +

" </tr>";

}

table = table + "</table>";

$("#test\_doctor").html(table);

})

}

function deleteDoc(idDel){

$.ajax({

url:'doctors/delete'+idDel,

type: "DELETE",

dataType: "json",

contentType: "application/json; charset=utf-8",

success: function (){

refresh();

},

error: function (){

console.log("fuck");

}

})

}

</script>

<body>

<center>

<h1>Doctors Management</h1>

<h2>

<a th:href="@{/doctors/view-1}">Add New Doctor</a>

&nbsp;&nbsp;&nbsp;

<a th:href="@{/doctors}"></a>

&nbsp;

<a th:href="@{/}">Main menu</a>

</h2>

</center>

<div align="center" id="test\_doctor">

<form>

<input type="">

</form>

</div>

</body>

</html>

// @filename \src\resources\templates\DoctorForm.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<script src="JQuery.js"></script>

</head>

<script type="text/javascript">

var docId = [[${docId}]];

console.log(docId);

$(document).ready(function (){

if(docId!=-1){

getData()

};

})

function getData(){

$.ajax({

url:'/doctors/get'+docId,

type: "GET",

dataType: "json",

success: function (result){

$("#txtId").val(docId);

$("#txtName").val(result.name);

$("#txtSurname").val(result.surname);

$("#txtMidname").val(result.midname);

$("#txtCabinet").val(0);

$("#txtSpeciality").val(result.speciality);

},

error: function (){

}

})

}

function saveData() {

if (docId !== -1) {

$.ajax({

url: "/doctors/edit/",

dataType: "json",

contentType: "application/json",

type: "PUT",

cache: false,

data: JSON.stringify({

id: $("#txtId").val(),

name: $("#txtName").val(),

surname: $("#txtSurname").val(),

midname: $("#txtMidname").val(),

//cabinet:$("#txtCabinet").val(),

cabinet: 0,

speciality: $("#txtSpeciality").val()

}),

success: function () {

getData();

},

error: function () {

}

});

} else {

$.ajax({

url: "/doctors/post/",

dataType: "json",

contentType: "application/json",

type: "POST",

cache: false,

data: JSON.stringify({

id: $("#txtId").val(),

name: $("#txtName").val(),

surname: $("#txtSurname").val(),

midname: $("#txtMidname").val(),

//cabinet:$("#txtCabinet").val(),

cabinet: 0,

speciality: $("#txtSpeciality").val()

}),

success: function (data) {

docId=data;

getData();

},

error: function () {

}

})

}

}

</script>

<body>

<div align="center" id ="formdoc"></div>

<form>

<table border="1" cellpadding="5">

<caption>

<h2>Doctor</h2>

</caption>

<input type="hidden" name="id" id="txtId"/>

<tr>

<th>Name: </th>

<td>

<input type="text" name="name" size="45" id="txtName" />

</td>

</tr>

<tr>

<th>Surname: </th>

<td>

<input type="text" name="surname" size="45" id="txtSurname" />

</td>

</tr>

<tr>

<th>Middle Name: </th>

<td>

<input type="text" name="midname" size="45" id="txtMidname" />

</td>

</tr>

<tr>

<th>Cabinet: </th>

<td>

<input type="number" name="cabinet" size="45" id="txtCabinet" />

</td>

</tr>

<tr>

<th>Speciality: </th>

<td>

<input type="text" name="speciality" size="45" id="txtSpeciality"/>

</td>

</tr>

<tr>

<td colspan="2" align="center"><button type="submit" value="save" onclick='saveData()'>Save</button>

</table>

</form>

</body>

</html>

// @filename \src\resources\templates\PatientIndex.htm

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<script src="JQuery.js"></script>

</head>

<script type="text/javascript">

$(document).ready(function (){

refresh();

})

function getPat(idGet){

window.location.href='/patients/view'+idGet;

}

function refresh(){

$.get('patients/all',function (data){

console.log(data);

let table = "<table border='1'>";

table = table+"<caption><h2>List of Patients</h2></caption>\n" +

" <tr>\n" +

" <th>Name</th>\n" +

" <th>Surname</th>\n" +

" <th>Middle Name</th>\n" +

" </tr>";

for(i=0; i<data.length; i++){

//table=table+"<tr><td>"+data[i].name+"</td><td>"+data[i].surname+"</td><td><button onclick='deleteDoc("+data[i].id+")'>DELETE</button></td></tr>";

table = table +"<tr>\n" +

" <td>"+data[i].name+"</td>\n" +

" <td>"+data[i].surname+"</td>\n" +

" <td>"+data[i].midname+"</td>\n" +

" <td>\n" +

" <button onclick='getPat("+data[i].id+")'>Edit</button>\n"+

// " <button th:formaction=\"@{/doctors/get{id}(id=${doctor.id})}\">Edit</button>\n" +

" &nbsp;&nbsp;&nbsp;&nbsp;\n" +

" <button onclick='deletePat("+data[i].id+")'>Delete</button>\n" +

" </td>\n" +

" </tr>";

}

table = table + "</table>";

$("#test\_patient").html(table);

})

}

function deletePat(idDel){

$.ajax({

url:'patients/delete'+idDel,

type: "DELETE",

dataType: "json",

contentType: "application/json; charset=utf-8",

success: function (){

refresh();

},

error: function (){

console.log("fuck");

}

})

}

</script>

<body>

<center>

<h1>Patient Management</h1>

<h2>

<a th:href="@{/patients/view-1}">Add New Patient</a>

&nbsp;&nbsp;&nbsp;

<a th:href="@{/patients}"></a>

&nbsp;

<a th:href="@{/}">Main menu</a>

</h2>

</center>

<div align="center" id="test\_patient">

<form>

<input type="">

</form>

</div>

</body>

</html>

// @filename \src\resources\templates\PatientForm.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<script src="JQuery.js"></script>

</head>

<script type="text/javascript">

var patId = [[${patId}]];

$(document).ready(function (){

if(patId>0){

getData(patId)

};

})

function getData(idget){

$.ajax({

url:'/patients/get'+idget,

type: "GET",

dataType: "json",

success: function (result){

$("#txtId").val(patId);

$("#txtName").val(result.name);

$("#txtSurname").val(result.surname);

$("#txtMidname").val(result.midname);

},

error: function (){

}

})

}

function saveData() {

if (patId > 0) {

$.ajax({

url: "/patients/edit/",

dataType: "json",

contentType: "application/json",

type: "PUT",

cache: false,

data: JSON.stringify({

id: $("#txtId").val(),

name: $("#txtName").val(),

surname: $("#txtSurname").val(),

midname: $("#txtMidname").val(),

}),

success: function () {

getData();

},

error: function () {

}

});

} else {

$.ajax({

url: "/patients/post/",

dataType: "json",

contentType: "application/json",

type: "POST",

cache: false,

data: JSON.stringify({

id: $("#txtId").val(),

name: $("#txtName").val(),

surname: $("#txtSurname").val(),

midname: $("#txtMidname").val(),

}),

success: function (data) {

console.log(data);

window.location.href='/doctors/view'+data;

},

error: function (error) {

console.log(error);

}

})

}

}

</script>

<body>

<div align="center" id ="formdoc"></div>

<form>

<table border="1" cellpadding="5">

<caption>

<h2>Patient</h2>

</caption>

<input type="hidden" name="id" id="txtId"/>

<tr>

<th>Name: </th>

<td>

<input type="text" name="name" size="45" id="txtName" />

</td>

</tr>

<tr>

<th>Surname: </th>

<td>

<input type="text" name="surname" size="45" id="txtSurname" />

</td>

</tr>

<tr>

<th>Middle Name: </th>

<td>

<input type="text" name="midname" size="45" id="txtMidname" />

</td>

</tr>

<tr>

<td colspan="2" align="center"><button type="submit" value="save" onclick='saveData()'>Save</button>

</table>

</form>

</body>

</html>