**Web Technology—Lab Assignment 5**

**Title:** - : **Design and develop a responsive website to prepare one semester result of VIT students using REACT, Springboot and MySQL. Take any four subjects with MSE Marks (30%) ESE Marks (70%).**

**Code: -**

**Client-side code:**

**Dashboard:**

import React from 'react';

import { Link,useNavigate } from 'react-router-dom';

function FrontPage(){

    return (

        <div className="front-page">

            <h1>Welcome To VIT</h1>

            <div>

                <img

                    src={require("../profile.png")}

                    className='profile'

                />

            </div>

            <div className='main'>

                <button class="button-10" role="button"><Link to="/AddStudents">Registered Student &rarr;</Link></button>

                <button class="button-10" role="button"><Link to="/showCertificate">Show certificate &rarr;</Link></button>

            </div>

        </div>

    );

}

export default FrontPage;

AddStudent Page:

import React from 'react';

import { useState } from 'react';

import { Link,useNavigate } from 'react-router-dom';

function AddStudents(){

    const [name,setName] = useState("");

    const [grno,setgr] = useState("");

    const [cdm,setcdm] = useState("");

    const [cde,setcde] = useState("");

    const [ccm,setccm] = useState("");

    const [cce,setcce] = useState("");

    const [wtm,setwtm] = useState("");

    const [wte,setwte] = useState("");

    const [daam,setdaam] = useState("");

    const [daae,setdaae] = useState("");

    const navigate = useNavigate();

    async function submit(){

        let ar = {name,grno,cdm,cde,ccm,cce,wtm,wte,daam,daae};

        // alert(ar);

        if(!name || !grno || !cdm || !cde || !ccm || !wtm || !wte || !cce || !daam || !daae){

            alert("No field should be empty");

            return false;

        }

            // e.preventDefault()

            // const student={name,address}

            // console.log(student)

            fetch("http://localhost:8080/student/add",{

            method:"POST",

            headers:{"Content-Type":"application/json"},

            body:JSON.stringify(ar)

        }).then(()=>{

            alert("New Student added");

            // window.location.reload();

            navigate("/");

        })

    }

    return (

        <div className="Student">

        <h1>Student Registration</h1>

            <section className="grid">

                <div className='flexSignup'>

                    <label style={{marginright:"10px"}}>Name: </label>

                    <input className="inputtext" placeholder='Your Name' value={name} onChange={e=>setName(e.target.value)}/>

                </div>

                <div className='flexSignup'>

                    <label>GRNO: </label>

                    <input className="inputtext" placeholder='Your GRNO' value={grno}

                        onChange={e=>setgr(e.target.value)}

                    />

                </div>

                <div className='flexSignup'>

                    <label>Compiler Design: </label>

                    <div>

                        <input placeholder='MSE' value={cdm} onChange={e=>setcdm(e.target.value)} />

                        <input placeholder='ESE' value={cde} onChange={e=>setcde(e.target.value)} />

                    </div>

                </div>

                <div className='flexSignup'>

                    <label>Cloud Computing: </label>

                    <div>

                    <input placeholder='MSE' value={ccm} onChange={e=>setccm(e.target.value)} />

                        <input placeholder='ESE' value={cce} onChange={e=>setcce(e.target.value)} />

                    </div>

                </div>

                <div className='flexSignup'>

                    <label>Web Technology: </label>

                    <div>

                    <input placeholder='MSE' value={wtm} onChange={e=>setwtm(e.target.value)} />

                        <input placeholder='ESE' value={wte} onChange={e=>setwte(e.target.value)} />

                    </div>

                </div>

                <div className='flexSignup'>

                    <label>DAA: </label>

                    <div>

                        <input placeholder='MSE' value={daam} onChange={e=>setdaam(e.target.value)} />

                        <input placeholder='ESE' value={daae} onChange={e=>setdaae(e.target.value)} />

                    </div>

                </div>

            </section>

                <div>

                    <button  className='button-10 btn' onClick={submit}>Submit</button>

                </div>

        </div>

    )

}

export default AddStudents;

ShowStudents Page:

import React from "react";

import { useState } from 'react';

import { Link,useNavigate } from 'react-router-dom';

function ShowCertificate(){

    const [flag,setFlag] = useState(false);

    const [grno,setGrno] = useState(0);

    const [data,setData] = useState([]);

    const [mse,setMse] = useState(0);

    const [ese,setEse] = useState(0);

    const navigate = useNavigate();

    // let id = "12011004";

    function back() {

        navigate("/");

    }

    function print(){

        //Set the print button visibility to 'hidden'

        var printButton = document.getElementById("printpagebutton");

        var backBtn = document.getElementById("back")

        printButton.style.visibility = 'hidden';

        backBtn.style.visibility = 'hidden';

        window.print();

        printButton.style.visibility = 'visible';

        backBtn.style.visibility = 'visible';

    }

    async function showres(){

        fetch(`http://localhost:8080/student/getStudent?name=${grno}`)

        .then(res=>res.json())

            .then((result)=>{

                console.log(result);

                // return ;

                if(result.grno===grno){

                    setFlag(true);

                    setData(result);

                    setMse(parseInt(result.cdm)+parseInt(result.wtm)+parseInt(result.ccm)+parseInt(result.daam));

                    setEse(parseInt(result.cde)+parseInt(result.wte)+parseInt(result.cce)+parseInt(result.daae))

                    console.log("result",parseInt(result.cdm));

                }else{

                    alert("enter valid no");

                }

            }

            )

    }

    return (

        <div>

        {!flag?

            <section className="resultPage">

                    <label>Enter GRNo. </label>

                    <input placeholder="Enter GRNo" value={grno} onChange={e=>setGrno(e.target.value)}/>

                    <button className="button-10 btn" onClick={showres}>Submit</button>

            </section>

        :

        <div>

        <h1 className="heading">Vit Result</h1>

        <section className="result">

                <div className="name">

                    <label>Name: </label>

                    <label>{data.name}</label>

                </div>

                <div className="grno">

                    <label>GRNO: </label>

                    <label>{data.grno}</label>

                </div>

                <table>

                    <tr>

                        <th>Subject</th>

                        <th>MSE</th>

                        <th>ESE</th>

                    </tr>

                    <tr>

                        <td>Compiler Design</td>

                        <td>{data.cdm}</td>

                        <td>{data.cde}</td>

                    </tr>

                    <tr>

                        <td>Cloud Computing</td>

                        <td>{data.ccm}</td>

                        <td>{data.ccm}</td>

                    </tr>

                    <tr>

                        <td>Web Technology</td>

                        <td>{data.wtm}</td>

                        <td>{data.wte}</td>

                    </tr>

                    <tr>

                        <td>DAA</td>

                        <td>{data.daam}</td>

                        <td>{data.daae}</td>

                    </tr>

                    <tr>

                        <td>Total</td>

                        <td>{mse}</td>

                        <td>{ese}</td>

                    </tr>

                </table>

                {console.log(mse)}

                <div>

                    <h1>Total Percentage={mse\*0.3+ese\*0.7}</h1>

                </div>

                {/\* <button onClick={showres}>Submit</button> \*/}

                <div>

                    <button id="printpagebutton" className="button-9 " onClick={print}>Print</button>

                    <button id="back" className="button-9 " onClick={back}>Back</button>

                </div>

            </section>

            </div>

            }

        </div>

    )

}

export default ShowCertificate;

App.js

import logo from './logo.svg';

import './App.css';

import FrontPage from './componenets/FrontPage';

import AddStudents from './componenets/AddStudents';

import ShowCertificate from './componenets/ShowStudent';

import { BrowserRouter,Routes,Route } from 'react-router-dom';

function App() {

  return (

    <div className="App">

      <BrowserRouter>

      <Routes>

      <Route path="/" element={<FrontPage/>}/>

      <Route path="/AddStudents" element={<AddStudents/>}/>

      <Route path="/showCertificate" element={<ShowCertificate/>}/>

      </Routes>

      </BrowserRouter>

    </div>

  );

}

export default App;

**Server Side code:**

**Main File:**

package com.arjuncodes.studentsystem;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class StudentsystemApplication {

    public static void main(String[] args) {

        SpringApplication.run(StudentsystemApplication.class, args);

    }

}

**Controller :**

package com.arjuncodes.studentsystem.controller;

import com.arjuncodes.studentsystem.model.Student;

import com.arjuncodes.studentsystem.service.StudentService;

import com.arjuncodes.studentsystem.service.StudentServiceImpl;

import org.springframework.beans.factory.annotation.Autowired;

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

import java.util.List;

@RestController

@RequestMapping("/student")

@CrossOrigin

public class StudentController {

    @Autowired

    private StudentService studentService;

    private StudentServiceImpl service ;

    @PostMapping("/add")

    public String add(@RequestBody Student student){

        studentService.saveStudent(student);

        return "New student is added";

    }

    @GetMapping("/getAll")

    public List<Student> list(){

        return studentService.getAllStudents();

    }

    @GetMapping("/getStudent")

    public Student search(@RequestParam(value="name", defaultValue="") String name){

        System.out.println("idh feoicw eocwoihfknckshdi"+name);

        // return null;

        return studentService.getStudent(name);

    }

}

**Model:**

package com.arjuncodes.studentsystem.model;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class Student {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private int id;

    private String name;

    private String grno;

    private String cdm;

    private String cde;

    private String ccm;

    private String cce;

    private String wtm;

    private String wte;

    private String daam;

    private String daae;

    public Student() {

    }

    public int getId() {

        return id;

    }

    public void setId(int id) {

        this.id = id;

    }

    public String getName() {

        return name;

    }

    public void setName(String name) {

        this.name = name;

    }

    public String getGrno() {

        return grno;

    }

    public void setGrno(String grno) {

        this.grno = grno;

    }

    // Getter methods

    public String getCdm() {

        return cdm;

    }

    public String getCde() {

        return cde;

    }

    public String getCcm() {

        return ccm;

    }

    public String getCce() {

        return cce;

    }

    public String getWtm() {

        return wtm;

    }

    public String getWte() {

        return wte;

    }

    public String getDaam() {

        return daam;

    }

    public String getDaae() {

        return daae;

    }

    // Setter methods

    public void setCdm(String cdm) {

        this.cdm = cdm;

    }

    public void setCde(String cde) {

        this.cde = cde;

    }

    public void setCcm(String ccm) {

        this.ccm = ccm;

    }

    public void setCce(String cce) {

        this.cce = cce;

    }

    public void setWtm(String wtm) {

        this.wtm = wtm;

    }

    public void setWte(String wte) {

        this.wte = wte;

    }

    public void setDaam(String daam) {

        this.daam = daam;

    }

    public void setDaae(String daae) {

        this.daae = daae;

    }

}

**Repository:**

package com.arjuncodes.studentsystem.repository;

import com.arjuncodes.studentsystem.model.Student;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

@Repository

public interface StudentRepository extends JpaRepository<Student,Integer> {

}

**Service:**

**File1:**

package com.arjuncodes.studentsystem.service;

import com.arjuncodes.studentsystem.model.Student;

import java.util.List;

public interface StudentService {

    public Student saveStudent(Student student);

    public List<Student> getAllStudents();

    public Student getStudent(String grno);

}

**File2:**

package com.arjuncodes.studentsystem.service;

import com.arjuncodes.studentsystem.model.Student;

import com.arjuncodes.studentsystem.repository.StudentRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class StudentServiceImpl implements StudentService {

    @Autowired

    private StudentRepository studentRepository;

    @Override

    public Student saveStudent(Student student) {

        return studentRepository.save(student);

    }

    @Override

    public List<Student> getAllStudents() {

        return studentRepository.findAll();

    }

    @Override

    public Student getStudent(String grno) {

        System.out.print("Riteshlndlcwncdsc ldsjoewejlcmksncksn"+grno);

        Iterable<Student> list = studentRepository.findAll(); //

        for(Student student : list){

            System.out.println(student.getGrno());

            if(student.getGrno().equals(grno))

            {

                return student;

            }

        }

        Student s = new Student();

        s.setGrno(null);

        return s ;

    }

}

**Config File:**

#configuration

spring.jpa.hibernate.ddl-auto=update

spring.datasource.url=jdbc:mysql://localhost:3307/vitresult

spring.datasource.username=root

spring.datasource.password=

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

**Output: -**





