**SKG\_LAB\_EMPLOYEE\_NOTLIKE**

**Controller:**package com.examly.springapp.controller;

import java.util.List;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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

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

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

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

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

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

import com.examly.springapp.model.Employee;

import com.examly.springapp.service.EmployeeService;

@RestController

@RequestMapping("/employees")

public class EmployeeController {

    private final EmployeeService employeeService;

    public EmployeeController(EmployeeService employeeService) { this.employeeService = employeeService;

    }

    @PostMapping

    public ResponseEntity<Employee> createEmployee(@RequestBody Employee employee)

    {

        Employee e=employeeService.saveEmployee(employee);

        if(e!=null)

        {

            return new ResponseEntity<>(e,HttpStatus.CREATED);

        }

        else{

            return new ResponseEntity<>(e,HttpStatus.INTERNAL\_SERVER\_ERROR);

        }

    }

    @GetMapping

    public List<Employee> getAllEmployees()

    {

        return employeeService.getAllEmployees();

    }

    @GetMapping("/notContaining/{name}")

    public List<Employee> getEmployeesNotContainingName(@PathVariable String name)

    {

         return employeeService.getEmployeesNotContainingName(name);

    }

    @GetMapping("/notContains/{designation}")

    public List<Employee> getEmployeesNotContainingDesignation(@PathVariable String designation)

    {

        return employeeService.getEmployeesNotContainingDesignation(designation);

    }

    @GetMapping("/notLike/{searchTerm}")

    public List<Employee> getEmployeesNotLike(@PathVariable String searchTerm)

    {

        return employeeService.getEmployeesNotLike(searchTerm);

    }

}

**Model:**package com.examly.springapp.model;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class Employee

{

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private int id;

    private String name;

    private String designation;

    public Employee() {}

    public Employee(String name, String designation) { this.name = name;

    this.designation = designation;

    }

    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 getDesignation()

    {

        return designation;

    }

    public void setDesignation(String designation)

    {

        this.designation = designation;

    }

}

**Repository:**

package com.examly.springapp.repository;

import java.util.List;

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

import org.springframework.stereotype.Repository;

import com.examly.springapp.model.Employee;

@Repository

public interface EmployeeRepo extends JpaRepository<Employee, Integer> {

    List<Employee> findByNameNotContaining(String name);

    List<Employee> findByDesignationNotContains(String designation);

}

**Service:**

package com.examly.springapp.service;

import java.util.List;

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

import org.springframework.stereotype.Service;

import com.examly.springapp.model.Employee;

import com.examly.springapp.repository.EmployeeRepo;

@Service

public class EmployeeService

{

    @Autowired

    EmployeeRepo employeeRepo;

    public EmployeeService(EmployeeRepo employeeRepo) { this.employeeRepo = employeeRepo;

    }

    public Employee saveEmployee(Employee employee)

    {

        return employeeRepo.save(employee);

    }

    public List<Employee> getAllEmployees()

    {

        return employeeRepo.findAll();

    }

    public List<Employee> getEmployeesNotContainingName(String name)

    {

        return employeeRepo.findByNameNotContaining(name);

    }

    public List<Employee> getEmployeesNotContainingDesignation(String designation)

    {

        return employeeRepo.findByDesignationNotContains(designation);

    }

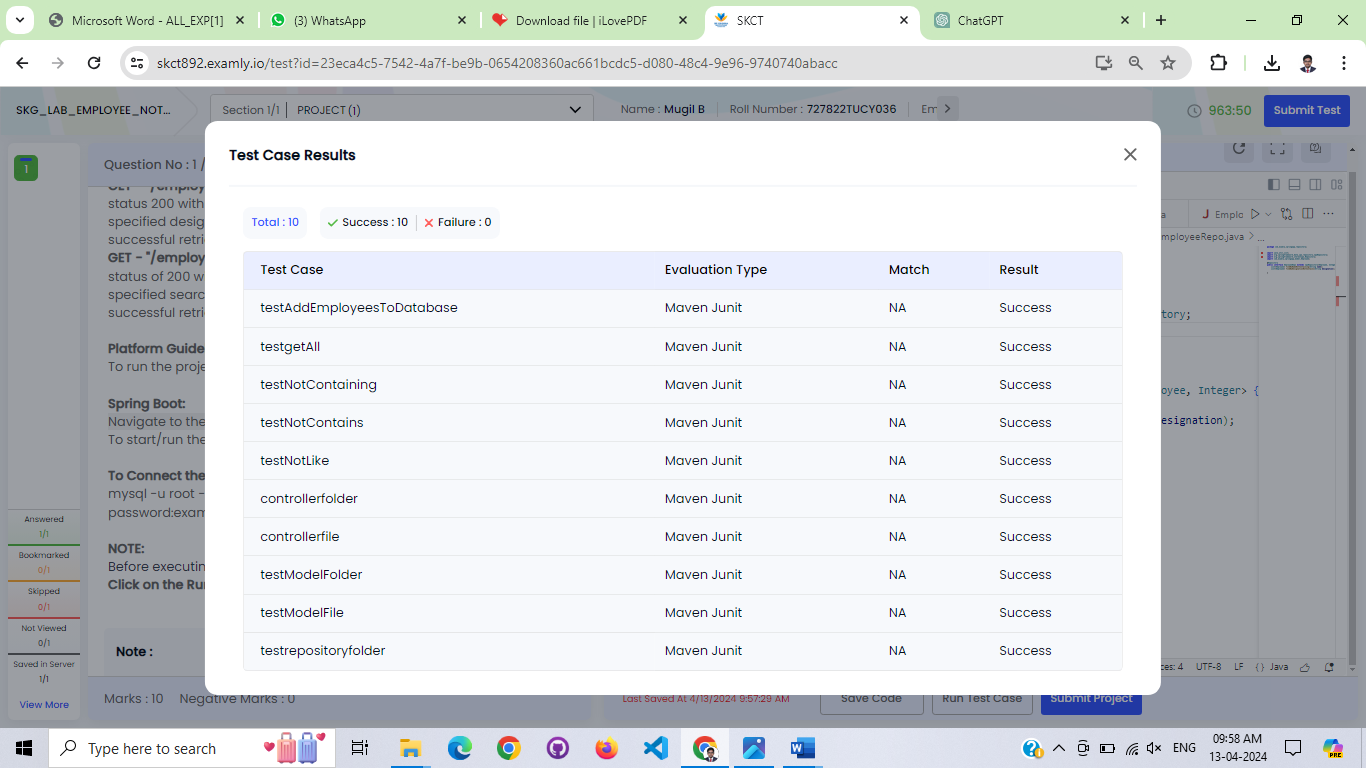
    public List<Employee> getEmployeesNotLike(String searchTerm)

    {

        return employeeRepo.findByNameNotContaining(searchTerm);

    }

}

**Test Case:**