**THE STATE UNIVERSITY OF ZANZIBAR(SUZA)**

****

**SCHOOL OF BUSINESS (SoB)**

**DEPERTMENT OF COMPUTER SCIENCE AND INFORMATION**

**TECHNPLOGY**

**BARCHELOR DEGREE INFORMATION TECHNOLOGY WITH**

**ACCOUNTING**

**STUDENT NAME : RAMADHAN ALI KHAMIS**

**REGISTRATION : BITA/5/21/032/TZ**

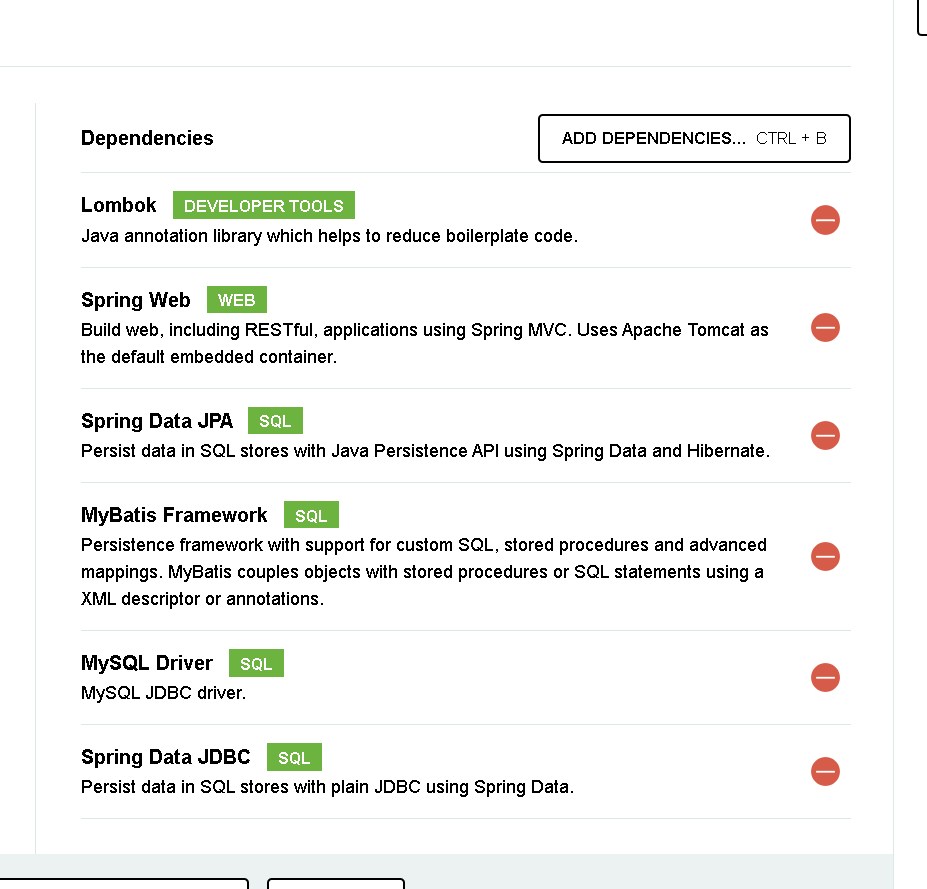
**TASK: MARKUP TEST**

**QUESTION ONE**

**A)**

****

B)Dependence



C)APPLICATION.PROPERTIES

spring.datasource.url=jdbc:mysql://localhost:3306/emplo  
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
spring.datasource.username=root  
spring.datasource.password=  
spring.jpa.show-sql=true  
spring.jpa.hibernate.ddl-auto=update

D)EMPLOYEE MODEL

package Model;  
  
  
import lombok.Data;  
  
import javax.persistence.\*;  
  
@Data  
@Table (name="employee")  
@Entity  
  
public class Employee {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private long id;  
  
 @Column(name = "firstName")  
 private String firstName;  
  
 @Column(name = "lastName")  
 private String lastName;  
  
 @Column(name = "email")  
 private String email;  
  
 public Employee() {  
  
 }  
  
 public Employee(String firstName, String lastName, String email) {  
 this.firstName = firstName;  
 this.lastName = lastName;  
 }  
  
 public long getId() {  
 return id;  
 }  
  
 public void setId(long id) {  
 this.id = id;  
 }  
  
 public String getFirstName() {  
 return firstName;  
 }  
  
 public void setFirstName(String firstName) {  
 this.firstName = firstName;  
 }  
  
 public String getLastName() {  
 return lastName;  
 }  
  
 public void setLastName(String lastName) {  
 this.lastName = lastName;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public void setEmail(String email) {  
 this.email = email;  
 }  
 }

F)EMPLOYEEREPOSITORY

package com.Employementmamagement.emplyeemanagementsystem.Repository;  
  
import Model.Employee;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface EmpRepository extends JpaRepository<Employee, Long> {  
}

G)SERVICE

@GetMapping("/employees")  
 public List<Employee> getAllEmployee(){  
 return empRepository.findAll();  
 }  
  
 @PostMapping("/employees")  
 public Employee createEmployee(@RequestBody Employee employee) {  
 return empRepository.save(employee);  
 }  
 @GetMapping("/employees/{id}")  
 public ResponseEntity<Employee> getEmployeeById(@PathVariable Long id) {  
 Employee employee = empRepository.findById(id)  
 .orElseThrow(() -> new ResourceNotFoundException("Employee Not Exist with Id :" + id));  
 return ResponseEntity.*ok*(employee);  
 }  
 @DeleteMapping("/employee/{id}")  
 public ResponseEntity<Map<String, Boolean>> deleteCustomer(@PathVariable Long id){  
 Employee employee = empRepository.findById(id)  
 .orElseThrow(() -> new ResourceNotFoundException("Employee Not Exist With Id :" + id));  
  
 empRepository.delete(employee);  
 Map<String, Boolean> response = new HashMap<>();  
 response.put("deleted", Boolean.*TRUE*);  
 return ResponseEntity.*ok*(response);  
 }  
}

H)CONTROLLER FOR GETTING AND POSTING

package com.Employementmamagement.emplyeemanagementsystem.Controller;  
  
  
import Model.Employee;  
import com.Employementmamagement.emplyeemanagementsystem.Exception.ResourceNotFoundException;  
import com.Employementmamagement.emplyeemanagementsystem.Repository.EmpRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
@CrossOrigin  
@RestController  
@RequestMapping("/api/v1/")  
  
public class EmpController {  
 @Autowired  
 private EmpRepository empRepository;  
 @GetMapping("/employees")  
 public List<Employee> getAllEmployee(){  
 return empRepository.findAll();  
 }  
  
 @PostMapping("/employees")  
 public Employee createEmployee(@RequestBody Employee employee) {  
 return empRepository.save(employee);  
 }  
 @GetMapping("/employees/{id}")  
 public ResponseEntity<Employee> getEmployeeById(@PathVariable Long id) {  
 Employee employee = empRepository.findById(id)  
 .orElseThrow(() -> new ResourceNotFoundException("Employee Not Exist with Id :" + id));  
 return ResponseEntity.*ok*(employee);  
 }  
 @DeleteMapping("/employee/{id}")  
 public ResponseEntity<Map<String, Boolean>> deleteCustomer(@PathVariable Long id){  
 Employee employee = empRepository.findById(id)  
 .orElseThrow(() -> new ResourceNotFoundException("Employee Not Exist With Id :" + id));  
  
 empRepository.delete(employee);  
 Map<String, Boolean> response = new HashMap<>();  
 response.put("deleted", Boolean.*TRUE*);  
 return ResponseEntity.*ok*(response);  
 }  
}

I)API

