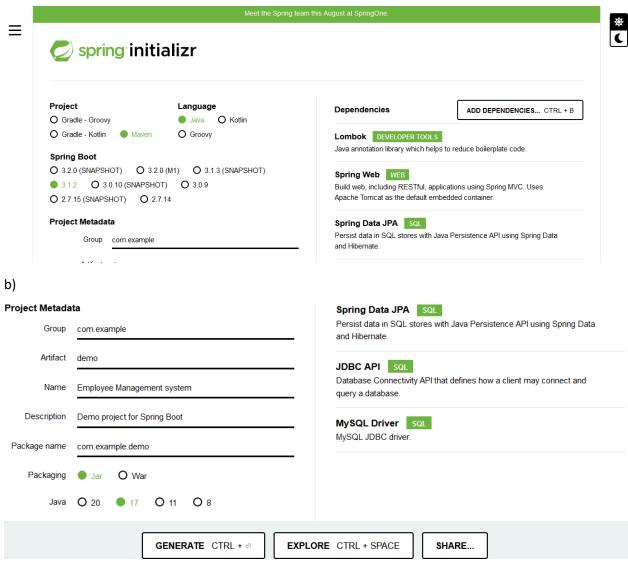
## BITA/5/21/036/TZ

## **QUESTION 2**

## REPORT AND SOURCE CODE

a)



c)

d)

```
package com.Employee Management System;
import lombok.Data;
import javax.persistence.*;
@Table
@Entity
@Data
public class Employee 🛚
   @GeneratedValue(strategy = GenerationType.IDENTITY)
   private int employee_number;
   private String first_name;
   private String last_name;
   private String email;
   public Customer() {
   public Customer(int customer_ID, String phone_no, String cus_name, String food_name, String email) {
       employee_number = employee_number;
       this.first_name = first_name;
       this.last_name = last_name
        this.email = email;
```

```
public Customer() {
public Customer(int customer_ID, String phone_no, String cus_name, String food_name, String email) {
    employee_number = employee_number;
    this.first_name = first_name;
    this.last_name = last_name
    this.email = email;
public int employee_number() {
    return employee_number;
public void set employee_number(int employee_number) {
loyee_number = employee_number }
public String get first_name() {
    return first_name;
public void set first_name(String first_name) {
    this.first_name = first_name;
public String get last_name() {
    return last_name;
public void set last_name(String last_name) {
    this.last_name = last_name;
```

```
public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email = email;
}
```

f)

```
package com.Employee Management System.Repository;

import com.Resta.RestaManagement.Module.Employee
import org.springframework.data.jpa.repository.JpaRepository;

public interface CustomerRepository extends JpaRepository<Customer,Integer> {
}
```

h)

```
package com.Employee Management System.Controller;
import com.Resta.Employee Management System.ResourceNotFoundException;
import com.Resta.Employee Management System.Module.employee;
{\color{blue} \textbf{import com.}} \textbf{Resta.Employee Management system.} \textbf{Repository.employeeRepository;}
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/emp")
public class employee Controller {
    @Autowired
    public employeeRepository employee
    @GetMapping("/employee")
    private List<Employee>viewEmployee(){
        return employeeRepository.findAll();
    @PostMapping("/employee")
    public Employee addEmployee(@RequestBody Employee employee){
        return employeeRepository.save(employee);
    @GetMapping("/employee/{id}")
    public ResponseEntity<Employee> getEmployeeByID(@PathVariable int id) {
        Employee employee = Employee.findById(id)
```

```
@GetMapping("/employee/{id}")
public ResponseEntity<Employee> getEmployeeByID(@PathVariable int id) {
    Employee employee = Employee.findById(id)
            .orElseThrow(() -> new ResourceNotFoundException("Invalid Id: " + id));
    return ResponseEntity.ok(employee);
@DeleteMapping("/employee/{id}")
public ResponseEntity<Map<String,Boolean>> deleteEmployee(@PathVariable int id)
    Employee employee = EmployeeRepository.findById(id)
            // throw resource not found exception if not found
            .orElseThrow(() -> new ResourceNotFoundException("Invalid Id: "+id));
    employeeRepository.delete(employee);
    Map<String,Boolean> response = new HashMap<>();
    response.put("employee was Deleted : ",Boolean.TRUE);
    return ResponseEntity.ok(response);
@PutMapping("/employee/{id}")
public ResponseEntity<Employee> updateEmployee(@PathVariable("id") int employee_id, @RequestBody Employee
    Employee employee1 = employeeRepository.findById(employee number)
            .orElseThrow(() -> new ResourceNotFoundException("Invalid Id: "+employee_id));
```

```
employee1setEmployee_number(employee.getEmployeeByID());
employee1.setEmail(employee.getEmail());
employee1setFirst_name(employee.getFirst_name());
employee1setLast_name(employee.getLast_name());

// saving the new values

return ResponseEntity.ok(employeeRepository.save(employee1));
}
```

