THE STATE UNIVERSITY OF ZANZIBAR (SUZA)



SCHOOL OF BUSINESS (SoB)

DEPAERTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

BACHELOR DEGREE IN INFORMATION TECHNOLOGY WITH ACCOUNTING

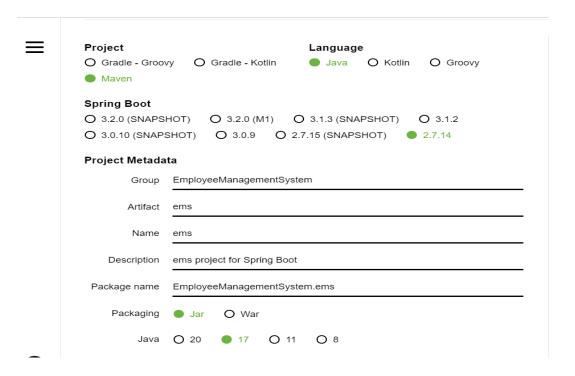
STUDENT NAME: IBTISAM ABDULLA RASHID

REGISTRATION: BITA/5/21/013/TZ

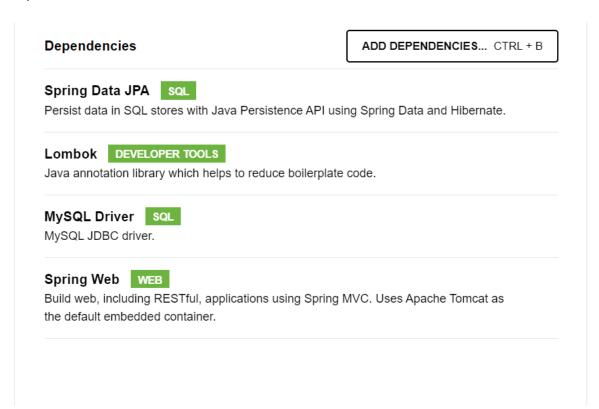
TASK: MARKUP TEST

QUESTION 1:

a. Project



b. Dependencies



c. Application.properties

```
spring.datasource.url=jdbc:mysql://localhost:3306/ems_db
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. Model
- e. EmployeeModel

```
package EmployeeManagementSystem.ems.model;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;
@Entity
@Table(name = "employees")
public class Employee {
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;
    @Column(name = "first name")
    private String first_name;
    @Column(name = "last_name")
    private String last_name;
    @Column(name = "email_id")
    private String email_id;
    public Employee() {
```

```
}
public Employee(String first_name, String last_name, String email_id) {
    super();
    this.first_name = first_name;
    this.last_name = last_name;
    this.email id = email id;
public long getId() {
    return id;
public void setId(long id) {
    this.id = id;
public String getFirst_name() {
    return first_name;
public void setFirst_name(String first_name) {
    this.first_name = first_name;
public String getLast_name() {
    return last_name;
public void setLast_name(String last_name) {
    this.last_name = last_name;
public String getEmail_id() {
    return email_id;
public void setEmail_id(String email_id) {
   this.email_id = email_id;
```

f. EmployeeRepository

```
package EmployeeManagementSystem.ems.repositoty;
import org.springframework.data.jpa.repository.JpaRepository;
import EmployeeManagementSystem.ems.model.Employee;
public interface EmployeeRepository extends JpaRepository<Employee, Long > {
}
```

g. Service

```
Get All Employees
   @GetMapping("/employees")
    public List<Employee> getAllEmployee(){
        return employeeRepository.findAll();
    // Create Employees rest api
   @PostMapping("/employees")
   public Employee createEmployee(@RequestBody Employee employee) {
        return employeeRepository.save(employee);
   // Get Employee by Id rest api
   @GetMapping("/employees/{id}")
    public ResponseEntity<Employee> getEmployeeById(@PathVariable Long id) {
        Employee employee = employeeRepository.findById(id)
                .orElseThrow(() -> new ResourceNotFoundException("Employee Not
Exist with Id :" + id));
        return ResponseEntity.ok(employee);
    // Delete Employee rest api
    @DeleteMapping("/employee/{id}")
```

h. Controller

```
package EmployeeManagementSystem.ems.controller;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.DeleteMapping;
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 EmployeeManagementSystem.ems.exception.ResourceNotFoundException;
import EmployeeManagementSystem.ems.model.Employee;
import EmployeeManagementSystem.ems.repositoty.EmployeeRepository;
@CrossOrigin
@RestController
@RequestMapping("/api/v1/")
public class EmployeeController {
    @Autowired
    private EmployeeRepository employeeRepository;
```

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

