THE STATE UNIVERSITY OF ZANZIBAR

SUZA

DEPARTMENT OF INFORMATION AND COMMUNICATION
TECHNOLOGY WITH   ACCOUNTING

ASSIGMENT: MARKUP TEST

REG NUMBER: BITA/5/21/011/TZ.

STUDENT NAME: WEMA PANDE ACHANI

SUBJECT NAME: ADVANCE WBSITE

LECTURAL NAME: MR MASOUD

Employee Management System this project that to create the spring-boot by initializer on to search the google search that to initializer
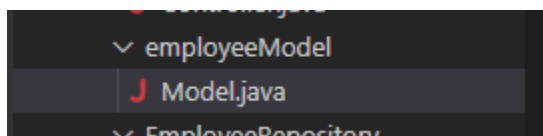
The following step using spring initializer or using maven commands

i.   Go to spring initializer

ii.   Fill in the details as per the requirements

iii.   Click on generate which will download the start project

iv.   Extract the zip file

v.   Create a java class with name controller model service repository

vi.   Go to postman and add url address and make put request

When to create mode are

Create mode code are:-

```java
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="employee")
public class Model {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private Long Id;
    @Column(nullable = false)
    private String firstname;
    @Column(nullable = false)
    private String lastname;
    @Column(nullable = false)
    private String email;

    public Model() {
    }

        public Model(Long Id, String firstname, String lastname, String email) {
        this.Id = Id;
        this.country = firstname;
        this.region = lastname;
```

```java
        this.district = email;
    }
public Long getId() {
        return Id;
    }

    public void setId(Long Idd) {
        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;
    }


}
```

The code for employee repository are:-

```java
package EployeeMnagementSystem;


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


import Online.school.management.system.Model.Model;


public interface Repository extends JpaRepository<Model, Long>{


}
```

The code of controller are :-

```java
package Employee.Mnagement.System.controller;
import java.util.List;


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

```java
import EmployeeMnagementSystem.Model.Model;
import EmployeeMnagementSystem.Repository.Repository;

@RestController
@RequestMapping("/api/s0")
@CrossOrigin
public class Controller {
    @Autowired
    private Repository repository;

    @PostMapping("/s2")
    public Model addBmiRecords(@RequestBody Model employee){
        return repository.save(employee);
    }

    @GetMapping("/s2")
    public List<Model> getAllEmployeeDB() {
        return repository.findAll();
    }

    @DeleteMapping("s2/{Id}")
    public void delete(@PathVariable long Id){
        repository.deleteById(Id);
    }

}
```

anEmployee Application class which will be used as Main Class.

```java
package com.examplel.EployeeMnagementSystem;

import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest
class EmployeeMnagementSystemApplicationTests {

    @Test
    void contextLoads() {

    }

}
```

Database

```
server.port=8090
spring.datasource.url=jdbc:mysql://localhost:3306/EmployeeDB?useSSL=false
spring.datasource.username=root
spring.datasource.password=

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect
spring.jpa.hibernate.ddl-auto=update
```