

Project Name: Department Management CRUD App

(functionality: Functionality: Manage department name.

Endpoints: POST /departments GET /departments PUT /departments/{id} DELETE
/departments/{id} using spring boot)

Github link: <https://github.com/pratham19092006/Department-Management-System>

NAME : Pratham Mishra

Roll No. : 2400291520179

Department: CSEAI

Date: 29-12-2025

CODE:

DemoManagementApplication.java:

```
package com.example.deptmanagement;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class DeptManagementApplication {

    public static void main(String[] args) {
        SpringApplication.run(DeptManagementApplication.class, args);
    }
}
```

DeptController.java:

```
package com.example.deptmanagement.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import com.example.deptmanagement.model.Dept;
import com.example.deptmanagement.service.DeptService;

@RestController
@RequestMapping("/depts")
public class DeptController {

    @Autowired
    private DeptService deptService;

    // POST /depts
    @PostMapping

```

```

public Dept createDept(@RequestBody Dept dept) {
    return deptService.addDept(dept);
}

// GET /depts
@GetMapping
public List<Dept> getAllDepts() {
    return deptService.getAllDepts();
}

// PUT /depts/{id}
@PutMapping("/{id}")
public Dept updateDept(
    @PathVariable int id,
    @RequestBody Dept dept) {
    return deptService.updateDept(id, dept);
}

// DELETE /depts/{id}
@DeleteMapping("/{id}")
public String deleteDept(@PathVariable int id) {
    return deptService.deleteDept(id);
}

```

DeptService.java:

```

package com.example.deptmanagement.service;

import java.util.ArrayList;
import java.util.List;

import org.springframework.stereotype.Service;

import com.example.deptmanagement.model.Dept;

@Service

```

```
public class DeptService {  
  
    private List<Dept> depts = new ArrayList<>();  
  
    // CREATE  
    public Dept addDept(Dept dept) {  
        depts.add(dept);  
        return dept;  
    }  
  
    // READ  
    public List<Dept> getAllDepts() {  
        return depts;  
    }  
  
    // UPDATE  
    public Dept updateDept(int id, Dept updatedDept) {  
        for (Dept d : depts) {  
            if (d.getId() == id) {  
                d.setName(updatedDept.getName());  
                return d;  
            }  
        }  
        return null;  
    }  
  
    // DELETE  
    public String deleteDept(int id) {  
        depts.removeIf(d -> d.getId() == id);  
        return "Dept deleted successfully";  
    }  
}
```

Dept.java:

```
package com.example.deptmanagement.service;

import java.util.ArrayList;
import java.util.List;
import org.springframework.stereotype.Service;
import com.example.deptmanagement.model.Dept;

@Service
public class DeptService {

    private List<Dept> depts = new ArrayList<>();

    // CREATE
    public Dept addDept(Dept dept) {
        depts.add(dept);
        return dept;
    }

    // READ
    public List<Dept> getAllDepts() {
        return depts;
    }

    // UPDATE
    public Dept updateDept(int id, Dept updatedDept) {
        for (Dept d : depts) {
            if (d.getId() == id) {
                d.setName(updatedDept.getName());
                return d;
            }
        }
    }
}
```

```

        return null;
    }

// DELETE

public String deleteDept(int id) {
    deptList.removeIf(d -> d.getId() == id);
    return "Dept deleted successfully";
}

}

```

Pom.xml:

```

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <!--  CORRECT Spring Boot Parent -->

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.2.5</version>
        <relativePath/>
    </parent>

    <groupId>com.example</groupId>
    <artifactId>dept-management-system</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>dept-management-system</name>
    <description>Dept Management CRUD Application</description>

    <properties>
        <java.version>17</java.version>
    
```

```
</properties>

<dependencies>

    <!--  REQUIRED FOR REST CONTROLLERS -->

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <!-- Testing -->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>

</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>

</project>
```

