**SPRING BOOT**

1. **Design a Spring Boot program to create a CRUD (Create, Read, Update, Delete) application using Hibernate for managing employee records. The program should allow users to perform the following operations on the employee database:**
   1. **Add a new employee: The user can enter details like employee name, department, and salary, and the program should add the employee to the database.**
   2. **Update employee details: The user can update the name, department, or salary of an existing employee based on their employee ID.**
   3. **Delete an employee: The user can delete an employee from the database based on their employee ID.**
   4. **Display all employees: The program should retrieve and display a list of all employees and their details from the database.**
   5. **Requirements:**
      1. **Use Spring Boot to create the application and Hibernate to manage the database.**
      2. **Implement JPA (Java Persistence API) for data access.**
      3. **Provide a RESTful API for performing CRUD operations on employees.**
      4. **Implement exception handling to handle possible errors during database interactions.**
      5. **Cover Spring Boot and Hibernate topics, such as entity classes, repositories, services, and controllers.**
   6. **Note: Before running the program, make sure you have set up the database and configured the connection in the application.properties file.**

**EmployeeController:**

package com.project44.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.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.project44.entity.Employee;

import com.project44.exception.ResourceNotFoundException;

import com.project44.repository.EmployeeRepository;

import com.project44.service.employee\_service;

@RestController

@RequestMapping("/api/v1/")

public class EmployeeController {

    @Autowired

    employee\_service employeeService;

    @Autowired

    private EmployeeRepository employeeRepository;

    public employee\_service getEmployeeService() {

        return employeeService;

    }

    public void setEmployeeService(employee\_service employeeService) {

        this.employeeService = employeeService;

    }

    @PutMapping("/employeeadd")

    public void addEmployee(@RequestBody Employee emp)

    {

        employeeService.addEmployee(emp);

    }

    @GetMapping("/employees")

    public List<Employee> getAllEmployee()

    {

        return employeeService.getAllEmployee();

    }

    @PutMapping("/employees/{id}")

    public ResponseEntity<Employee> updateEmployee(@PathVariable int id, @RequestBody Employee emp1)

    {

        Employee employee = employeeRepository.findById(id).orElseThrow(()->new ResourceNotFoundException("Employee not exist with given id" +id));

        employee.setEmpName(emp1.getEmpName());

        employee.setDeptName(emp1.getDeptName());

        employee.setSalary(emp1.getSalary());

        Employee updatedEmployees = employeeRepository.save(employee);

        return ResponseEntity.ok(updatedEmployees);

    }

    @DeleteMapping("/employees/{id}")

    public ResponseEntity<Map<String, Boolean>> deleteEmployee(@PathVariable int id)

    {

        Employee employee = employeeRepository.findById(id).orElseThrow(()->new ResourceNotFoundException("Employee not exist with given id" +id));

        employeeRepository.delete(employee);

        Map<String, Boolean> response = new HashMap<>();

        response.put("deleted", Boolean.TRUE);

        return ResponseEntity.ok(response);

    }

}

**entity:Employee.java**

package com.project44.entity;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

@Entity

@Table(name="Employee")

public class Employee {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private int id;

    @Column(name="employee\_Name")

    private String empName;

    @Column(name="dept\_Name")

    private String deptName;

    @Column(name="Salary")

    private long salary;

    public Employee() {

        super();

        // TODO Auto-generated constructor stub

    }

    public Employee(int empid, String empName, String deptName, long salary) {

        super();

        this.id = empid;

        this.empName = empName;

        this.deptName = deptName;

        this.salary = salary;

    }

    public Employee(String empName, String deptName, long salary) {

        super();

        this.empName = empName;

        this.deptName = deptName;

        this.salary = salary;

    }

    public int getId() {

        return id;

    }

    public void setId(int id) {

        this.id = id;

    }

    public String getEmpName() {

        return empName;

    }

    public void setEmpName(String empName) {

        this.empName = empName;

    }

    public String getDeptName() {

        return deptName;

    }

    public void setDeptName(String deptName) {

        this.deptName = deptName;

    }

    public long getSalary() {

        return salary;

    }

    public void setSalary(long salary) {

        this.salary = salary;

    }

}

**Exception: ResourceNotFoundException**

package com.project44.exception;

import org.springframework.http.HttpStatus;

import org.springframework.web.bind.annotation.ResponseStatus;

@ResponseStatus(value = HttpStatus.NOT\_FOUND)

public class ResourceNotFoundException extends RuntimeException{

    private static final long serialVersionUID = 1L;

    public ResourceNotFoundException(String message) {

        super(message);

    }

}

**repository:EmployeeRepository**

package com.project44.repository;

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

import org.springframework.stereotype.Repository;

import com.project44.entity.Employee;

@Repository

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

**service:employee\_service**

package com.project44.service;

import java.util.List;

import com.project44.entity.Employee;

public interface employee\_service {

    public void addEmployee(Employee emp);

    public void updateEmployee(Employee emp, int empid);

    public void deleteEmployee(int empid);

    public List<Employee> getAllEmployee();

}

**employee\_service\_Impl**

package com.project44.service;

import java.util.List;

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

import org.springframework.stereotype.Service;

import com.project44.entity.Employee;

import com.project44.repository.EmployeeRepository;

@Service

public class employee\_service\_Impl implements employee\_service {

    @Autowired

    EmployeeRepository employeeRepository;

    public EmployeeRepository getEmployeeRepository() {

        return employeeRepository;

    }

    public void setEmployeeRepository(EmployeeRepository employeeRepository) {

        this.employeeRepository = employeeRepository;

    }

    @Override

    public void addEmployee(Employee emp) {

        employeeRepository.save(emp);

    }

    @Override

    public void updateEmployee(Employee emp, int empid) {

        employeeRepository.save(emp);

    }

    @Override

    public void deleteEmployee(int empid) {

        employeeRepository.deleteById(empid);

    }

    @Override

    public List<Employee> getAllEmployee() {

        return employeeRepository.findAll();

    }

}

**EmployeeManagementSystemAssignment1Application.class**

package com.project44;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class EmployeeManagementSystemAssignment1Application {

    public static void main(String[] args) {

        SpringApplication.run(EmployeeManagementSystemAssignment1Application.class, args);

    }

}

**EmployeeManagementSystemAssignment1ApplicationTests**

package com.project44;

import org.junit.jupiter.api.Test;

import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest

class EmployeeManagementSystemAssignment1ApplicationTests {

    @Test

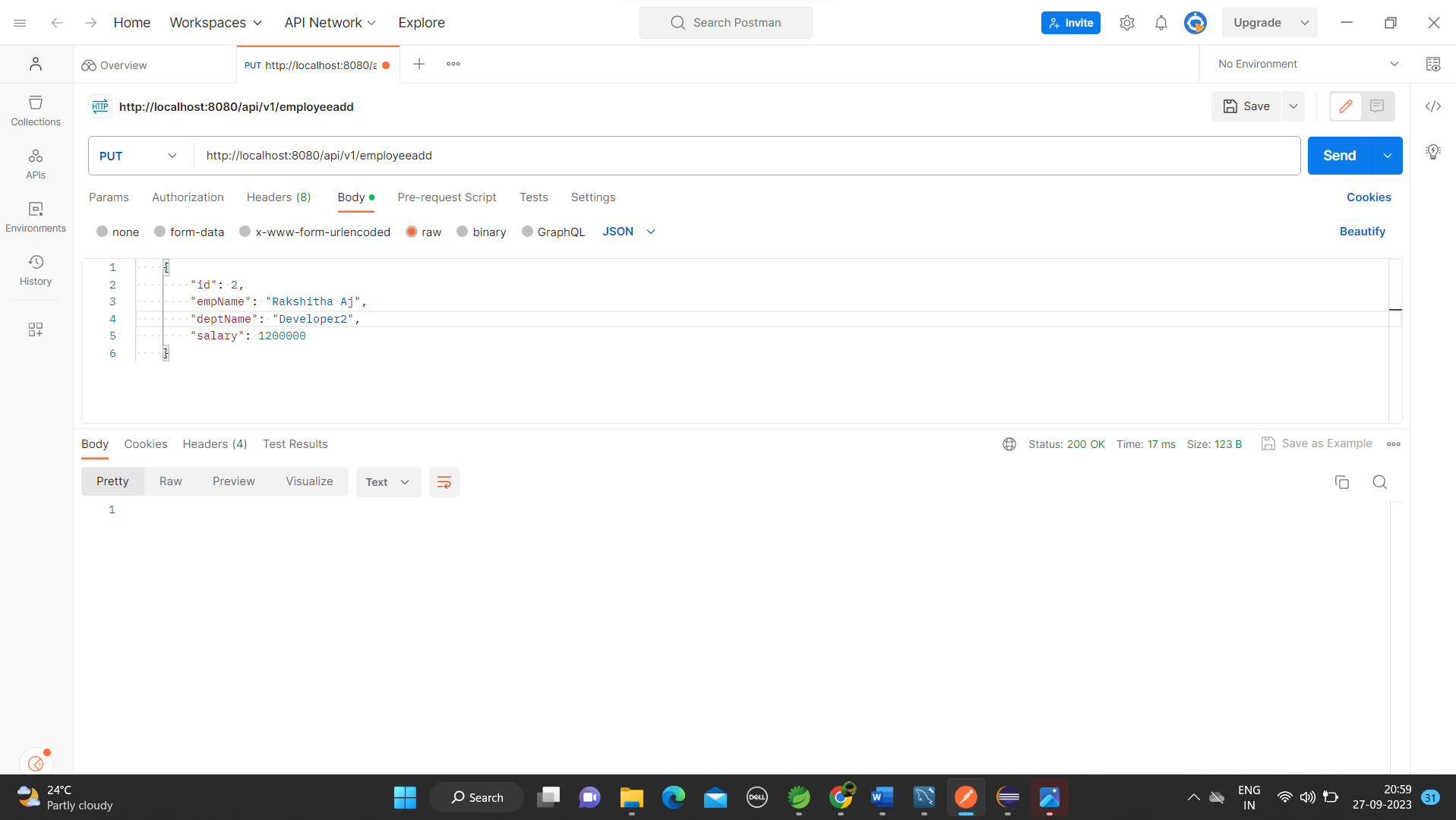
    void contextLoads() {

    }

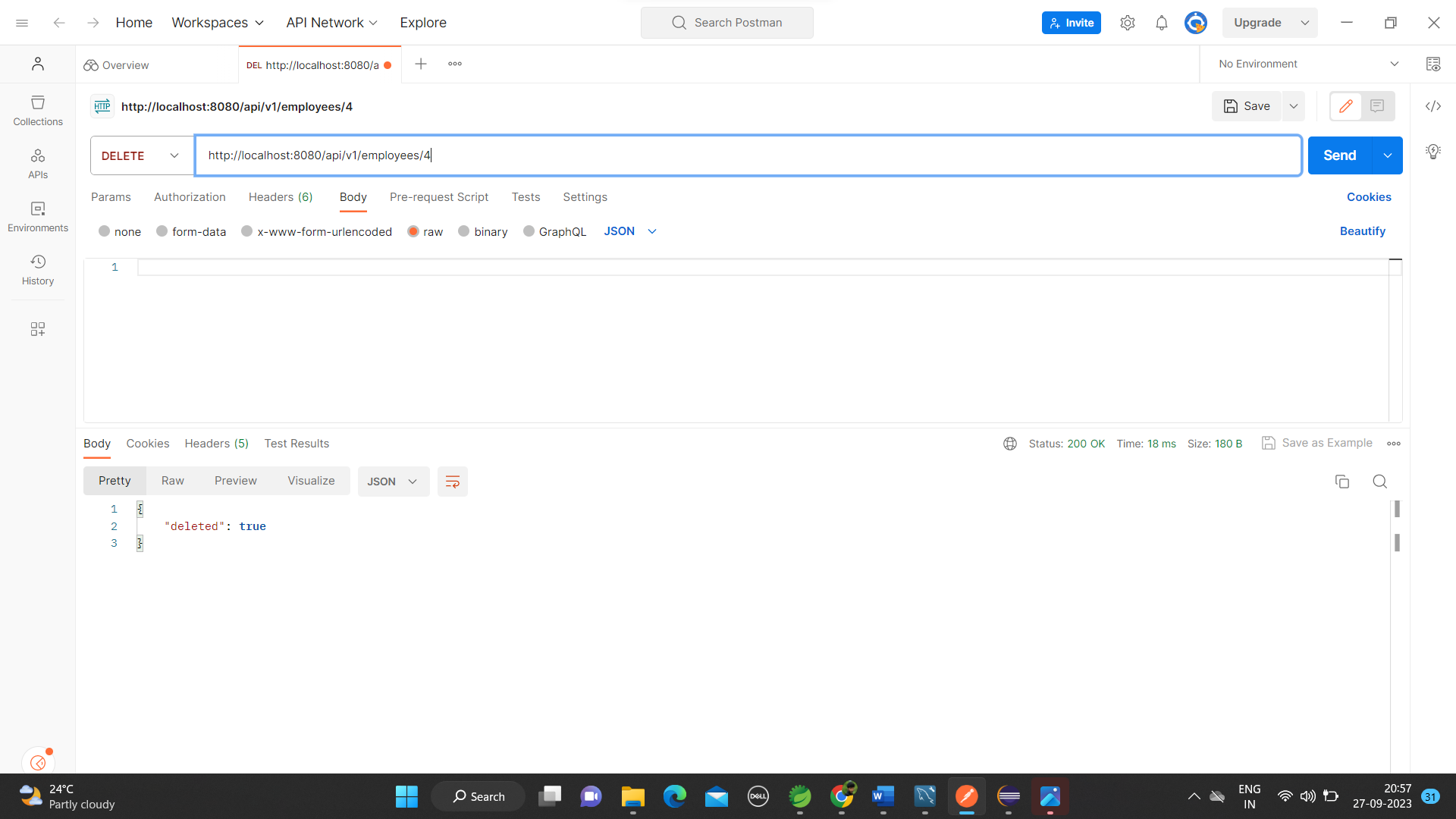
}

**Output:**

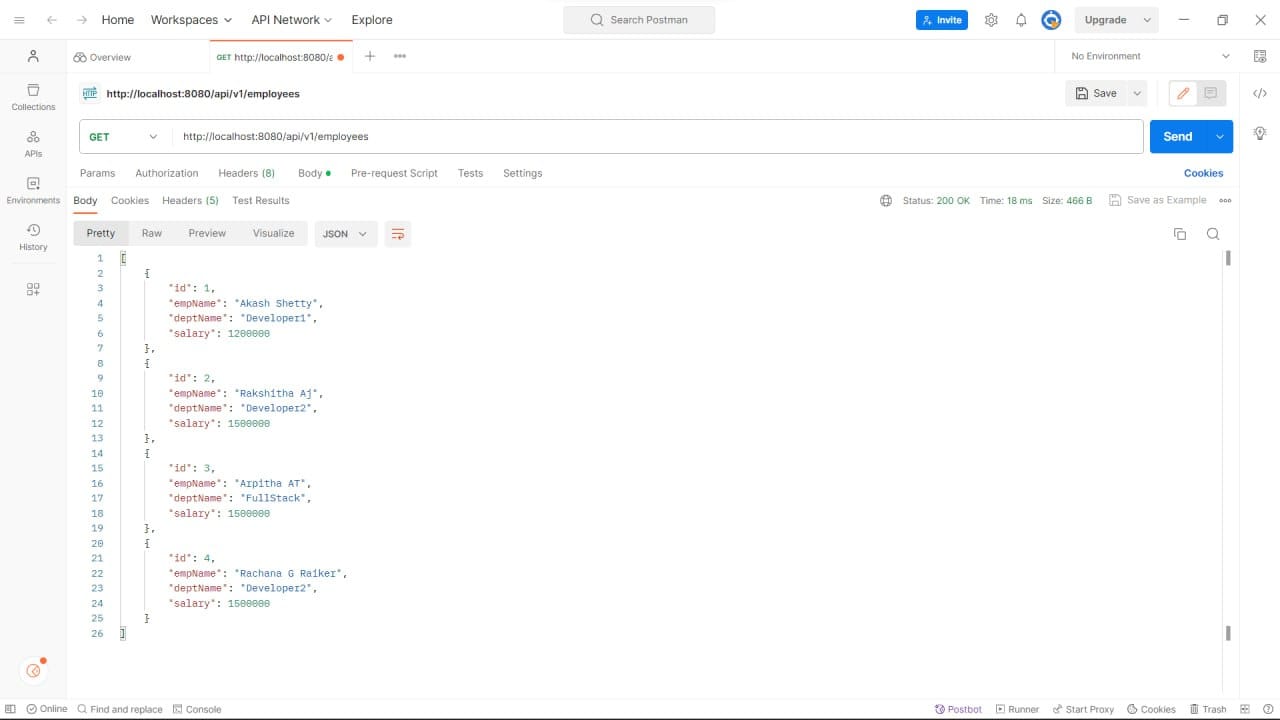
**1.Adding Employee**

****

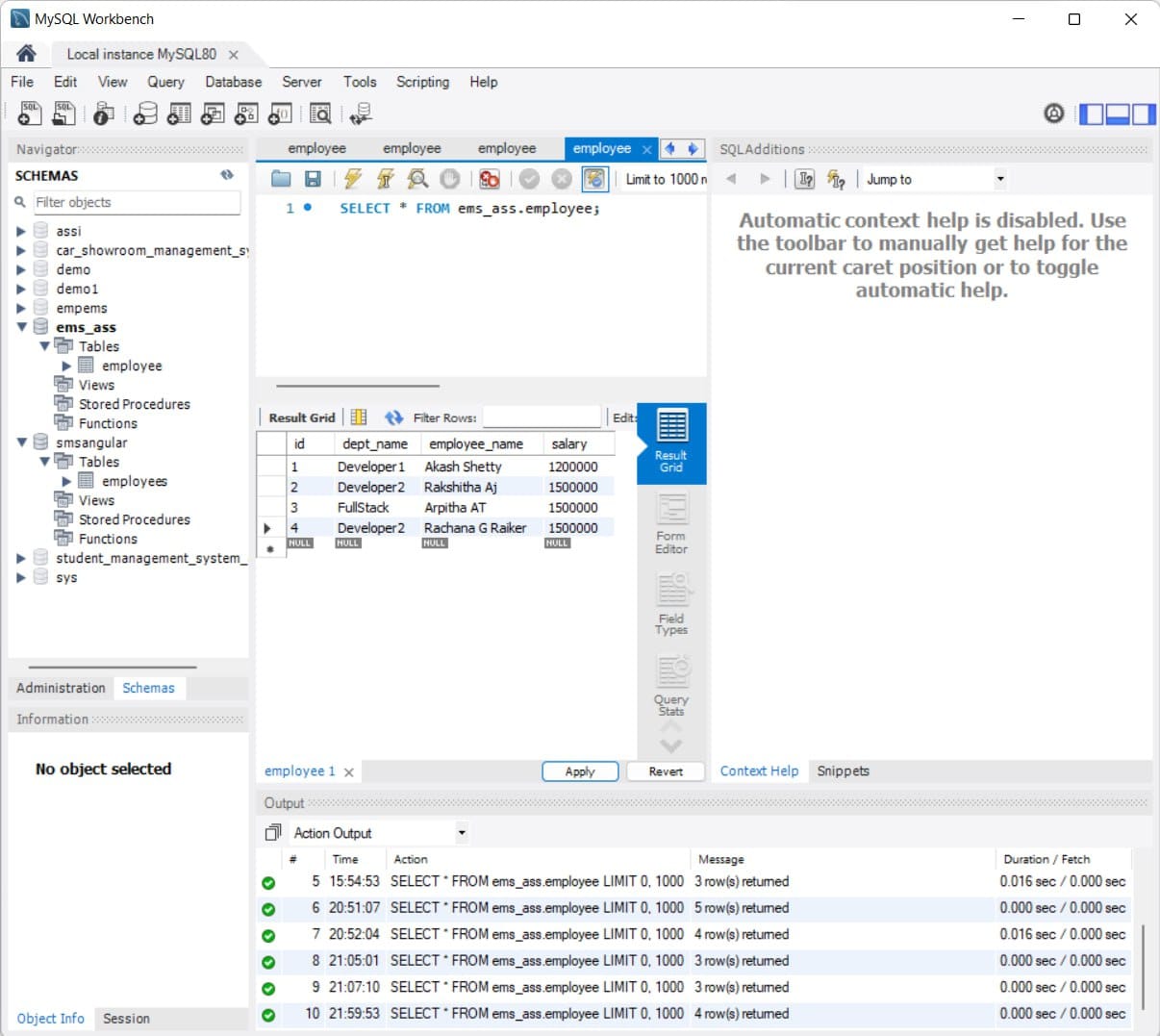
**2.Delete Employee**

****

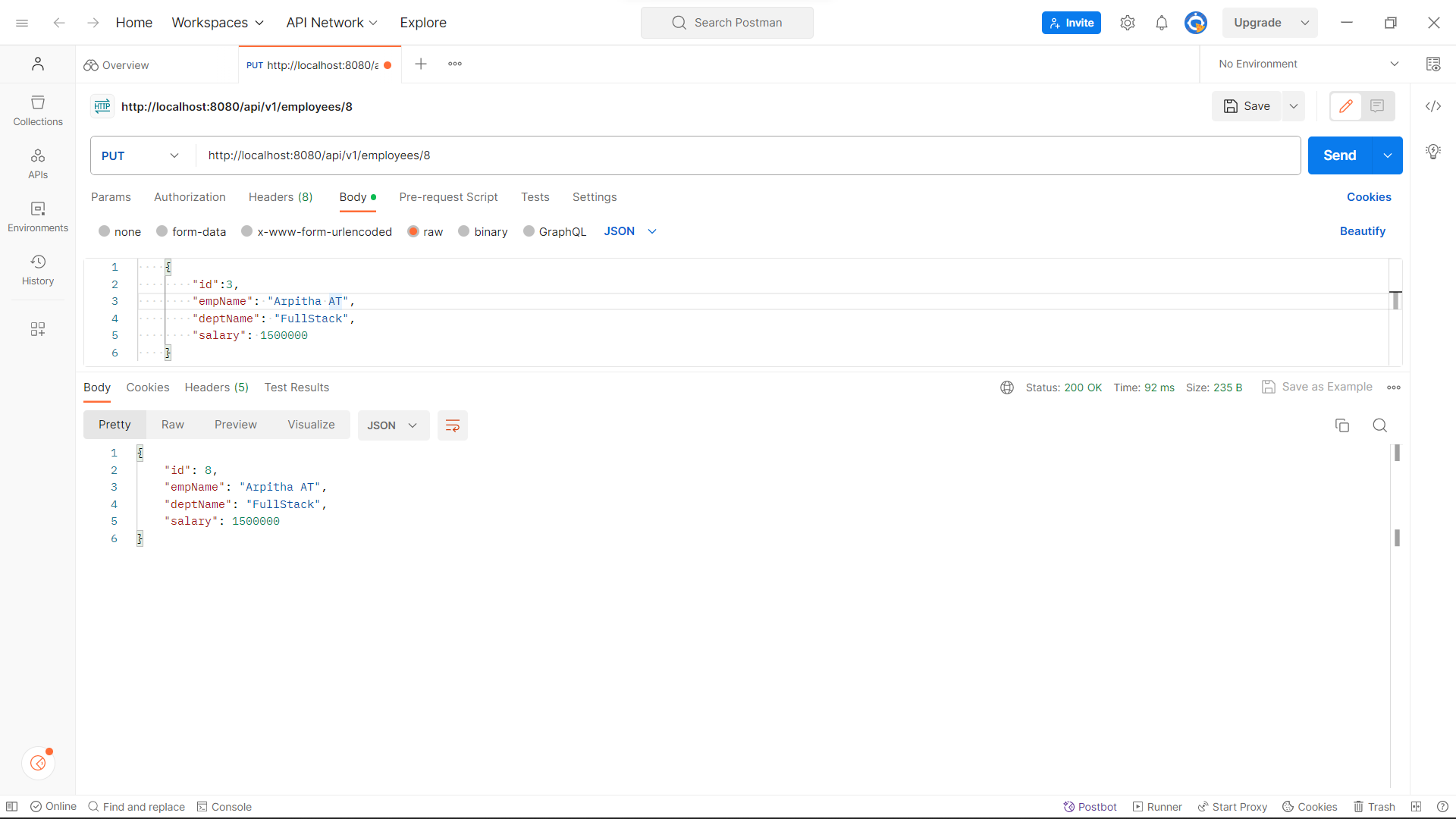
**3. Displaying all the employee**

****

**4. Employee Stored in database**

****

**5. Updating Employee**

****