4. WebApi\_Handson Objectives Answers

# Objective:

Demonstrate creation of an Action method to perform data create, update & delete operation

# Explanation:

- Use the [FromBody] attribute to accept incoming JSON data in POST, PUT, and DELETE methods.  
- Extract data into a custom model class like 'Employee' using FromBody.  
- Use a hardcoded list of Employee objects to simulate a database.  
- Use Swagger UI and Postman to test the endpoints:  
 • POST: To add a new employee  
 • PUT: To update an existing employee  
 • DELETE: To remove an employee by id

# Example Methods in EmployeeController:

## 1. POST – Add Employee

[HttpPost]  
public ActionResult<Employee> CreateEmployee([FromBody] Employee newEmp)  
{  
 \_employees.Add(newEmp);  
 return Ok(newEmp);  
}

## 2. PUT – Update Employee

[HttpPut("{id}")]  
public ActionResult<Employee> UpdateEmployee(int id, [FromBody] Employee updatedEmp)  
{  
 if (id <= 0) return BadRequest("Invalid employee id");  
 var emp = \_employees.FirstOrDefault(e => e.Id == id);  
 if (emp == null) return BadRequest("Invalid employee id");  
  
 emp.Name = updatedEmp.Name;  
 emp.Salary = updatedEmp.Salary;  
 emp.Permanent = updatedEmp.Permanent;  
 emp.Department = updatedEmp.Department;  
 emp.Skills = updatedEmp.Skills;  
 emp.DateOfBirth = updatedEmp.DateOfBirth;  
 return Ok(emp);  
}

## 3. DELETE – Remove Employee

[HttpDelete("{id}")]  
public ActionResult DeleteEmployee(int id)  
{  
 var emp = \_employees.FirstOrDefault(e => e.Id == id);  
 if (emp == null) return NotFound("Employee not found");  
 \_employees.Remove(emp);  
 return Ok("Employee deleted");  
}