Assignment 4

# Objective

This assignment demonstrates the use of Web API to perform Create, Update, and Delete operations using a custom model class.   
It specifically focuses on updating employee data using the PUT action method, validating input ID, and returning appropriate responses.

# Requirements

- Used [FromBody] attribute to extract JSON data from request body  
- Used a hardcoded list of employees  
- Validated the employee ID:  
 - If ID ≤ 0, return BadRequest with message 'Invalid employee id'  
 - If ID > 0 but not found in the list, return BadRequest with same message  
 - If ID is valid, update the employee and return the updated object  
- Test using Swagger and Postman

# Updated EmployeeController.cs

The following code replaces the existing EmployeeController and includes updated PUT logic for assignment 4:

[HttpPut("{id}")]  
[ProducesResponseType(StatusCodes.Status200OK)]  
[ProducesResponseType(StatusCodes.Status400BadRequest)]  
public ActionResult<Employee> UpdateEmployee(int id, [FromBody] Employee emp)  
{  
 if (id <= 0)  
 {  
 return BadRequest("Invalid employee id");  
 }  
  
 var existing = \_employees.FirstOrDefault(e => e.Id == id);  
 if (existing == null)  
 {  
 return BadRequest("Invalid employee id");  
 }  
  
 // Update properties  
 existing.Name = emp.Name;  
 existing.Salary = emp.Salary;  
 existing.Permanent = emp.Permanent;  
 existing.Department = emp.Department;  
 existing.Skills = emp.Skills;  
 existing.DateOfBirth = emp.DateOfBirth;  
  
 return Ok(existing); // return updated employee  
}

# Testing the PUT API

The PUT endpoint was tested using both Swagger UI and Postman.

# Output Screenshots:



# 4.2