**WEEK 4 WEB API HANDSON**

**Exercise 3:**

**Department.cs:** namespace EmployeeApi.Models

{

public class Department

{

public int Id { get; set; }

public string Name { get; set; }

}

}

**Employee.cs**:

using System;

using System.Collections.Generic;

namespace EmployeeApi.Models

{

public class Employee

{

public int Id { get; set; }

public string Name { get; set; }

public int Salary { get; set; }

public bool Permanent { get; set; }

public Department Department { get; set; }

public List<Skill> Skills { get; set; }

public DateTime DateOfBirth { get; set; }

}

}

**Skills.cs:**

namespace EmployeeApi.Models

{

public class Skill

{

public int Id { get; set; }

public string Name { get; set; }

}

}

**EmployeeController.cs:** using Microsoft.AspNetCore.Mvc;

using EmployeeApi.Models;

using Microsoft.AspNetCore.Authorization;

using EmployeeApi.Filters;

[ApiController]

[Route("[controller]")]

[Authorize(Roles = "POC")]

//[ServiceFilter(typeof(CustomAuthFilter))]

[AllowAnonymous]

public class EmployeeController : ControllerBase

{

private readonly List<Employee> \_employees;

public EmployeeController()

{

\_employees = GetStandardEmployeeList();

}

[HttpGet("GetStandard")]

//[AllowAnonymous]

[ProducesResponseType(typeof(List<Employee>), 200)]

[ProducesResponseType(500)]

public ActionResult<List<Employee>> GetStandard()

{

return \_employees;

}

[HttpPost]

public ActionResult AddEmployee([FromBody] Employee emp)

{

\_employees.Add(emp);

return Ok(new

{

message = "Employee added",

employee = emp

});

}

[HttpPut]

public ActionResult<Employee> UpdateEmployee([FromBody] Employee emp)

{

if (emp.Id <= 0)

return BadRequest("Invalid employee id");

var existingEmp = \_employees.FirstOrDefault(e => e.Id == emp.Id);

if (existingEmp == null)

return BadRequest("Invalid employee id");

existingEmp.Name = emp.Name;

existingEmp.Salary = emp.Salary;

existingEmp.Permanent = emp.Permanent;

existingEmp.Department = emp.Department;

existingEmp.Skills = emp.Skills;

existingEmp.DateOfBirth = emp.DateOfBirth;

return Ok(existingEmp);

}

[HttpDelete("{id}")]

public ActionResult DeleteEmployee(int id)

{

if (id <= 0)

return BadRequest("Invalid employee id");

var empToDelete = \_employees.FirstOrDefault(e => e.Id == id);

if (empToDelete == null)

return BadRequest("Invalid employee id");

\_employees.Remove(empToDelete);

return Ok($"Employee with ID {id} has been deleted.");

}

private List<Employee> GetStandardEmployeeList()

{

return new List<Employee>

{

new Employee

{

Id = 1,

Name = "Emma",

Salary = 60000,

Permanent = true,

Department = new Department { Id = 1, Name = "HR" },

Skills = new List<Skill>

{

new Skill { Id = 1, Name = "C#" },

new Skill { Id = 2, Name = "SQL" }

},

DateOfBirth = new DateTime(1997, 5, 12)

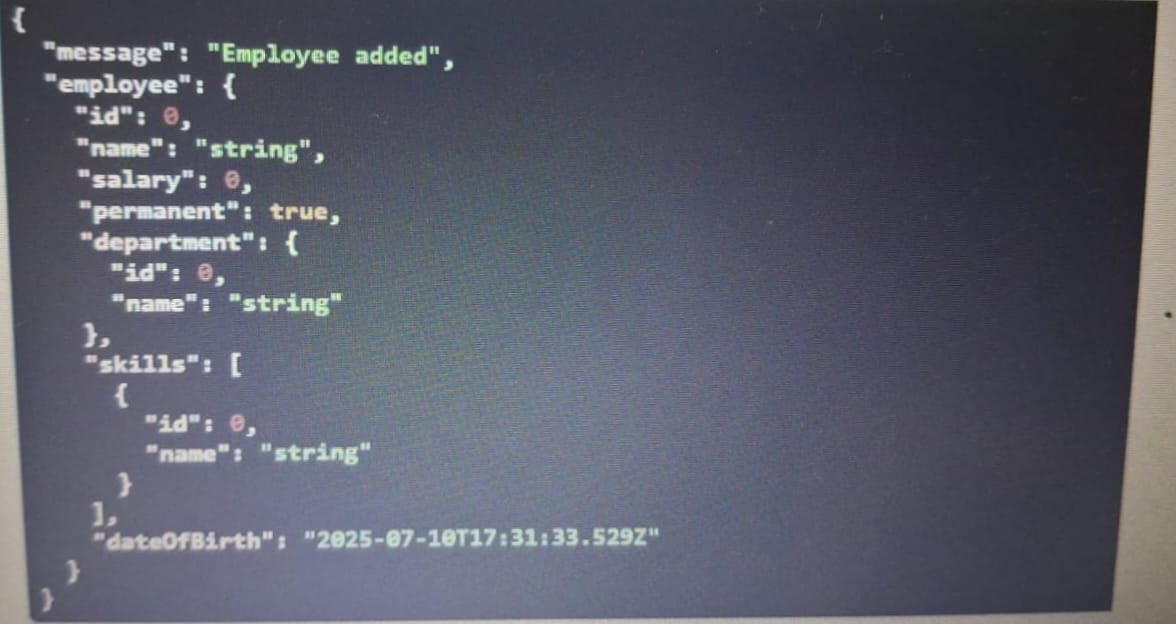
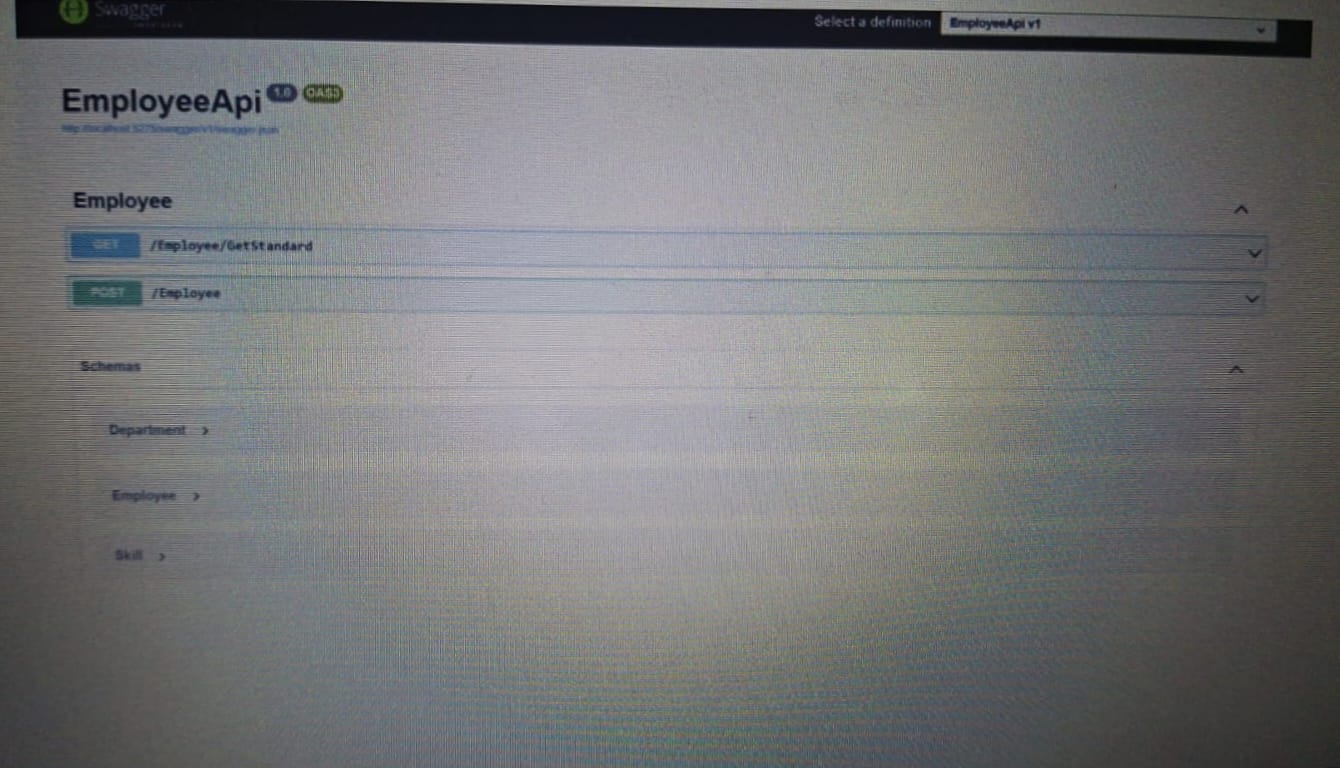
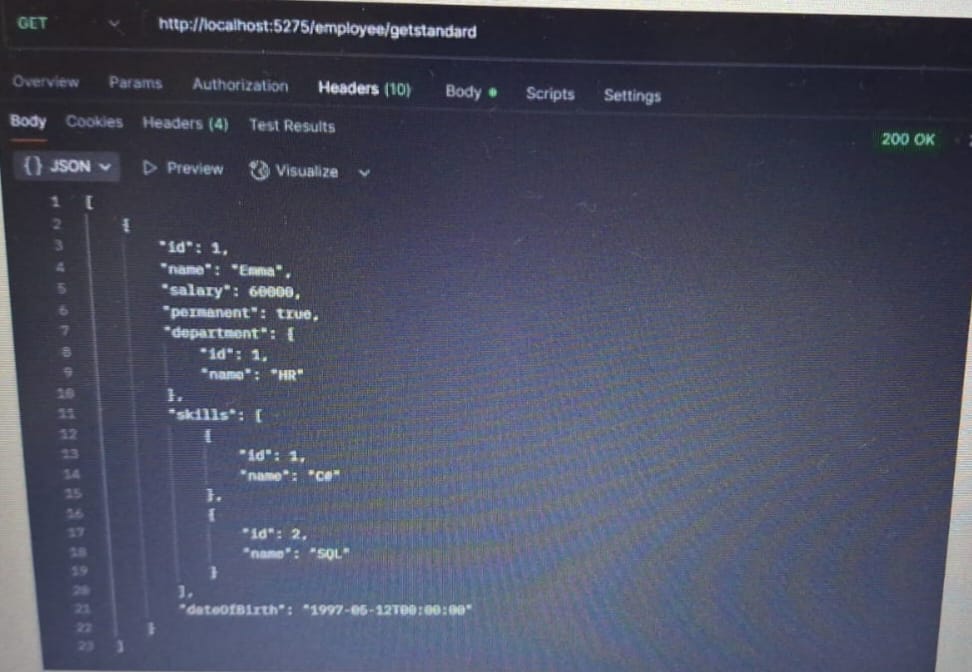
}

};

}

}

**OUTPUT:**

****