Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

\_\_\_13\_\_\_

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **01** | Create Reusable Code/Software for generating Marksheet of a student. (Hint: Use Project 1 for calculation and Grading purpose) |
| **02** | Consume Google Maps Api in Html Webpage. |
|  |  |
|  |  |

Submitted On

20-06-2023

(Date: DD/MM/YY)

**Task No. 1:** Create webapi that uses all verbs of Rest Api’s for employee management Create new employee, Update Current employee, Get Employees, Delete employees with their respective Id’s -> (Optional)For best professional practices use layered architecture for getting data from data access layer and applies logic with business logic layer then controller will get data and http will consume it.

**Solution:**

**Define the Data Models:**

namespace EmployeeManagement.Models{

public class Employee{

public int Id { get; set; }

public string Name { get; set; }

public int Age { get; set; }

public string Position { get; set; }}}

**Implement the Data Access Layer:**

using System.Collections.Generic;

using System.Linq;

using EmployeeManagement.Models;

namespace EmployeeManagement.DataAccess{

public class EmployeeRepository{

private List<Employee> employees;

public EmployeeRepository(){

employees = new List<Employee>();}

public IEnumerable<Employee> GetEmployees(){

return employees;}

public Employee GetEmployeeById(int id){

return employees.FirstOrDefault(e => e.Id == id);}

public void AddEmployee(Employee employee){

employee.Id = employees.Count + 1;

employees.Add(employee);}

public void UpdateEmployee(Employee employee){

var existingEmployee = employees.FirstOrDefault(e => e.Id == employee.Id);

if (existingEmployee != null){

existingEmployee.Name = employee.Name;

existingEmployee.Age = employee.Age;

existingEmployee.Position = employee.Position;}}

public void DeleteEmployee(int id){

var employee = employees.FirstOrDefault(e => e.Id == id);

if (employee != null){

employees.Remove(employee);}}}}

**Implement the Business Logic Layer:**

using System.Collections.Generic;

using EmployeeManagement.DataAccess;

using EmployeeManagement.Models;

namespace EmployeeManagement.BusinessLogic{

public class EmployeeService{

private EmployeeRepository employeeRepository;

public EmployeeService(){

employeeRepository = new EmployeeRepository();}

public IEnumerable<Employee> GetEmployees(){

return employeeRepository.GetEmployees();}

public Employee GetEmployeeById(int id){

return employeeRepository.GetEmployeeById(id);}

public void AddEmployee(Employee employee){

employeeRepository.AddEmployee(employee);}

public void UpdateEmployee(Employee employee){

employeeRepository.UpdateEmployee(employee);}

public void DeleteEmployee(int id){

employeeRepository.DeleteEmployee(id);}}}

**Implement the API Controller:**

using System.Collections.Generic;

using EmployeeManagement.BusinessLogic;

using EmployeeManagement.Models;

using Microsoft.AspNetCore.Mvc;

namespace EmployeeManagement.API.Controllers{

[Route("api/employees")]

[ApiController]

public class EmployeeController : ControllerBase{

private EmployeeService employeeService;

public EmployeeController(){

employeeService = new EmployeeService();}

[HttpGet]

public ActionResult<IEnumerable<Employee>> GetEmployees(){

var employees = employeeService.GetEmployees();

return Ok(employees);}

[HttpGet("{id}")]

public ActionResult<Employee> GetEmployeeById(int id){

var employee = employeeService.GetEmployeeById(id);

if (employee == null){

return NotFound();}

return Ok(employee);}

[HttpPost]

public IActionResult CreateEmployee([FromBody] Employee employee){

employeeService.AddEmployee(employee);

return CreatedAtAction(nameof(GetEmployeeById), new { id = employee.Id }, employee);}

[HttpPut("{id}")]

public IActionResult UpdateEmployee(int id, [FromBody] Employee employee){

if (id != employee.Id){

return BadRequest();}

employeeService.UpdateEmployee(employee);

return NoContent();}

[HttpDelete("{id}")]

public IActionResult DeleteEmployee(int id){

employeeService.DeleteEmployee(id);

return NoContent();}}}

**Task No. 2:** Create webapi for restaurant named Alibaba the restaurant can be able to Post new item in menu, Get All items that restaurant offers, Put existing items, Delete items

**Solution:**

namespace \_13t1.Service{

[WebService(Namespace = "http://tempuri.org/")]

[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]

[System.ComponentModel.ToolboxItem(false)]

public class WebService1 : System.Web.Services.WebService{

[WebMethod]

public string HelloWorld(){

return "Hello World";}

[WebMethod]

public int Square(int num){

return num \* num;}

public int Add(int num1, int num2){

return num1 + num2;}

public int Multiply(int num1, int num2){

return num1 \* num2;}

public int Subtract(int num1, int num2){

if (num1 > num2){

return num1 - num2;}

return num2 - num1;}

public float Division(float num1, float num2){

return num1 / num2;}}

**Output:**

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated