Source Code Of Practice Project 4 :

Github Link:

1. Bal Library

Bal.cs

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Bal
{
    public class BAL
    {
        public int student_id { get; set; }
        public string student_name { get; set; }
        public int subject_marks { get; set; }
    }
}
```

2. Dal Library
Dal.cs

```csharp
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Bal;

namespace Dal
{
    public class DAL
    {
        public bool Insert(BAL school)
        {
            // SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["NorthCnString"].ConnectionString);
            SqlConnection cn = new SqlConnection("Data Source=DESKTOP-9E6TH3I\\SQLEXPRESS;Initial
Catalog=school;Integrated Security=True");
```

```csharp
        SqlCommand cmdInsert = new SqlCommand("insert into marks(student_id,student_name,subject_marks)
values(@student_id,@student_name,@subject_marks)", cn);
        cmdInsert.Parameters.AddWithValue("@student_id", school.student_id);
        cmdInsert.Parameters.AddWithValue("@student_name", school.student_name);
        cmdInsert.Parameters.AddWithValue("@subject_marks", school.subject_marks);


        cn.Open();
        int i = cmdInsert.ExecuteNonQuery();

        bool status = false;

        if (i == 1)
        {
            status = true;
        }

        cn.Close();//finally
        cn.Dispose();//finally
        return status;


    }




    public bool Update(BAL school)
    {

        //  SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["NorthCnString"].ConnectionString);
        SqlConnection cn = new SqlConnection("Data Source=DESKTOP-9E6TH3I\\SQLEXPRESS;Initial
Catalog=school;Integrated Security=True");
        SqlCommand cmdUpdate = new SqlCommand("[dbo].[Updatemarks]", cn);

        cmdUpdate.CommandType = System.Data.CommandType.StoredProcedure;
        cmdUpdate.Parameters.AddWithValue("@p_studid", school.student_id);
        cmdUpdate.Parameters.AddWithValue("@p_studname", school.student_name);
        cmdUpdate.Parameters.AddWithValue("@p_submarks", school.subject_marks);
        cn.Open();
        int s = cmdUpdate.ExecuteNonQuery();
        bool statusd = false;
        if (s == 1)
        {
            statusd = true;
        }
        cn.Close();//finally
        cn.Dispose();//finally
        return statusd;

    }
```

```csharp
        public BAL Find(int id)
        {
            //  SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["NorthCnString"].ConnectionString);
            SqlConnection cn = new SqlConnection("Data Source=DESKTOP-9E6TH3I\\SQLEXPRESS;Initial
Catalog=school;Integrated Security=True");
            SqlCommand cmdSelect = new SqlCommand("[dbo].sp_Findmarks", cn);
            cmdSelect.CommandType = System.Data.CommandType.StoredProcedure;
            cmdSelect.Parameters.AddWithValue("@p_studid", id);
            SqlParameter p1 = new SqlParameter();
            p1.ParameterName = "@p_mark_studname";
            p1.SqlDbType = System.Data.SqlDbType.NVarChar;
            p1.Size = 20;
            p1.Direction = System.Data.ParameterDirection.Output;
            cmdSelect.Parameters.Add(p1);


            SqlParameter p2 = new SqlParameter();
            p2.ParameterName = "@p_marks_submarks";
            p2.SqlDbType = System.Data.SqlDbType.NVarChar;
            p2.Size = 20;
            p2.Direction = System.Data.ParameterDirection.Output;
            cmdSelect.Parameters.Add(p2);




            cn.Open();
            cmdSelect.ExecuteNonQuery();

            BAL found = new BAL();

            found.student_name = p1.Value.ToString();
            found.subject_marks = Convert.ToInt32(p2.Value);




            cn.Close();
            cn.Dispose();


            return found;


        }
        public List<BAL> list()
        {
            //  SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["NorthCnString"].ConnectionString);
            //   SqlConnection cn = new SqlConnection("Data Source =DESKTOP-LUAVTH3; Initial Catalog = school;
Integrated Security = True");
            SqlConnection cn = new SqlConnection("Data Source=DESKTOP-9E6TH3I\\SQLEXPRESS;Initial
Catalog=school;Integrated Security=True");
            SqlCommand cmdlist = new SqlCommand("select * from marks", cn);
```

```csharp
            cn.Open();
            SqlDataReader dr = cmdlist.ExecuteReader();
            List<BAL> emplist = new List<BAL>();
            if (dr.HasRows)
            {
                while (dr.Read())
                {
                    BAL bal = new BAL();
                    bal.student_id = Convert.ToInt32(dr["student_id"]);
                    bal.student_name = dr["student_name"].ToString();
                    bal.subject_marks = Convert.ToInt32(dr["subject_marks"]);

                    emplist.Add(bal);
                }
            }
            cn.Close();
            cn.Dispose();
            return emplist;

        }
        public bool Delete(int id)
        {
            //      SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["NorthCnString"].ConnectionString);
            SqlConnection cn = new SqlConnection("Data Source=DESKTOP-9E6TH3I\\SQLEXPRESS;Initial
Catalog=school;Integrated Security=True");
            SqlCommand cmdDelete = new SqlCommand("[dbo].sp_Deletemarks", cn);
            cmdDelete.CommandType = System.Data.CommandType.StoredProcedure;
            cmdDelete.Parameters.AddWithValue("@p_id", id);
            cn.Open();
            int i = cmdDelete.ExecuteNonQuery();
            bool status = false;
            if (i == 1)
            {
                status = true;
            }
            cn.Close();//finally
            cn.Dispose();//finally
            return status;

        }


    }
        }



### 3. Helper Library
#### Helper.cs

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Bal;
```

```csharp
using Dal;

namespace Helper
{
    public class Helper1
    {
        DAL dal = null;
        public Helper1()
        {
            dal = new DAL();
        }


        public bool Addmarks(BAL employee)
        {
            return dal.Insert(employee);

        }

        public bool Editmarks(BAL employee)
        {
            return dal.Update(employee);
        }

        public BAL searchmarks(int empid)
        {
            return dal.Find(empid);
        }
        public List<BAL> listmarks()
        {
            return dal.list();
        }
        public bool removemarks(int employee_id)
        {
            return dal.Delete(employee_id);
        }
    }
        }
```

4. In SimpliPractice4

ValuesController----- Controller

```csharp
using Bal;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;
using Helper;
using Simpli_Practice4.Models;


namespace Simpli_Practice4.Controllers
{
    public class ValuesController : ApiController
    {
```

```csharp
// GET api/values----receive
Helper1 obj = null;
public ValuesController()
{
    obj = new Helper1();
}
[HttpGet]
public List<marksmodel> marklist()
{
    //return new string[] { "value1", "value2" };

    List<BAL> empbal = new List<BAL>();
    empbal = obj.listmarks();
    List<marksmodel> emps = new List<marksmodel>();
    foreach (var item in empbal)
    {
        //Employees emp = new Employees();
        emps.Add(new marksmodel
        {
            student_id = item.student_id,
            student_name = item.student_name,
            subject_marks = item.subject_marks
        });
    }
    return emps;
}
public marksmodel marksbyid(int id)
{
    BAL empbal = new BAL();
    empbal = obj.searchmarks(id);
    marksmodel emp = new marksmodel();
    emp.student_id = empbal.student_id;
    emp.student_name = empbal.student_name;
    emp.subject_marks = empbal.subject_marks;

    return emp;
    //return "value";
}

// POST api/<controller> --- Update
public HttpResponseMessage Postmarks([FromBody] marksmodel empdata)
{
    BAL empbal = new BAL();
    empbal.student_id = empdata.student_id;
    empbal.student_name = empdata.student_name;
    empbal.subject_marks = empdata.subject_marks;



    bool ans = obj.Addmarks(empbal);
    if (ans)
    {
        return Request.CreateResponse(HttpStatusCode.OK);
    }
    else
    {
        return Request.CreateResponse(HttpStatusCode.NotAcceptable);
    }
```

```csharp
        }

        // PUT api/<controller>/5 -- add
        [HttpPut]


        public HttpResponseMessage Putmarks([FromBody] marksmodel empdata)
        {

            BAL empbal = new BAL();
            empbal.student_id = empdata.student_id;
            empbal.student_name = empdata.student_name;
            empbal.subject_marks = empdata.subject_marks;

            bool ans = obj.Editmarks(empbal);
            if (ans)
            {
                return Request.CreateResponse(HttpStatusCode.OK);
            }
            else
            {
                return Request.CreateResponse(HttpStatusCode.NotAcceptable);
            }
        }

        // DELETE api/<controller>/5
        public HttpResponseMessage Deletemarks(int id)
        {
            bool ans = obj.removemarks(id);
            if (ans)
            {
                return Request.CreateResponse(HttpStatusCode.OK);
            }
            else
            {
                return Request.CreateResponse(HttpStatusCode.NotAcceptable);
            }

        }
    }
}
```

Marksmodel.cs

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace Simpli_Practice4.Models
{
    public class marksmodel
    {
```

```csharp
    public int student_id { get; set; }
    public string student_name { get; set; }
    public int subject_marks { get; set; }
}
        }
```

Used Stored Procedure in this projects ----- Follow Github Link