**Practice Project**

**ASP.NET WEB API Application**

**Source Code:**

**BAL Library:**

**marks.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace BAL

{

public class marks

{

public int student\_id { get; set; }

public string student\_name { get; set; }

public int subject\_marks { get; set; }

}

}

**DAL Library:**

**Student\_DAL.cs:**

using BAL;

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DAL

{

public class Student\_DAL

{

public bool Insert(marks school)

{

SqlConnection cn = new SqlConnection("Data Source=DESKTOP-5GL4B5D\\SQLEXPRESS1;Initial Catalog=School;Integrated Security=True");

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(marks school)

{

SqlConnection cn = new SqlConnection("Data Source=DESKTOP-5GL4B5D\\SQLEXPRESS1;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;

}

public marks Find(int id)

{

SqlConnection cn = new SqlConnection("Data Source=DESKTOP-5GL4B5D\\SQLEXPRESS1;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 = 30;

p1.Direction = System.Data.ParameterDirection.Output;

cmdSelect.Parameters.Add(p1);

SqlParameter p2 = new SqlParameter();

p2.ParameterName = "@p\_marks\_submarks";

p2.SqlDbType = System.Data.SqlDbType.Int;

p2.Size = 20;

p2.Direction = System.Data.ParameterDirection.Output;

cmdSelect.Parameters.Add(p2);

cn.Open();

cmdSelect.ExecuteNonQuery();

marks found = new marks();

found.student\_name = p1.Value.ToString();

found.subject\_marks = Convert.ToInt32(p2.Value);

cn.Close();

cn.Dispose();

return found;

}

public List<marks> List()

{

SqlConnection cn = new SqlConnection("Data Source=DESKTOP-5GL4B5D\\SQLEXPRESS1;Initial Catalog=School;Integrated Security=True");

SqlCommand cmdlist = new SqlCommand("select student\_id,student\_name,subject\_marks from marks", cn);

cn.Open();

SqlDataReader dr = cmdlist.ExecuteReader();

List<marks> emplist = new List<marks>();

if (dr.HasRows)

{

while (dr.Read())

{

marks bal = new marks();

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 stuid)

{

SqlConnection cn = new SqlConnection("Data Source=DESKTOP-5GL4B5D\\SQLEXPRESS1;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", stuid);

cn.Open();

int i = cmdDelete.ExecuteNonQuery();

bool status = false;

if (i == 1)

{

status = true;

}

cn.Close();//finally

cn.Dispose();//finally

return status;

}

}

}

**Student\_Helper:**

using BAL;

using DAL;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Helper

{

public class Student\_Helper

{

Student\_DAL dal = null;

public Student\_Helper()

{

dal = new Student\_DAL();

}

public bool AddE(marks school)

{

return dal.Insert(school);

}

public bool Edit(marks school)

{

return dal.Update(school);

}

public marks search(int id)

{

return dal.Find(id);

}

public List<marks> BList()

{

return dal.List();

}

public bool remove(int id)

{

return dal.Delete(id);

}

}

}

**SubController.cs:**

using API.Models;

using BAL;

using Helper;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

namespace API.Controllers

{

public class SubController : ApiController

{

Student\_Helper obj = null;

public SubController()

{

obj = new Student\_Helper();

}

// [Route("GetAllMarks")]

[HttpGet]

public List<sub\_mark> GetMarkList()

{

List<marks> empbal = new List<marks>();

empbal = obj.BList();

List<sub\_mark> emps = new List<sub\_mark>();

foreach (var item in empbal)

{

//Employees emp = new Employees();

emps.Add(new sub\_mark { student\_id = item.student\_id, student\_name = item.student\_name, subject\_marks = item.subject\_marks });

}

return emps;

}

// GET api/<controller>/5

// [Route("~/FindE/{id}")]

// [Route("FindById/{id:int:min(1)}")]

[Route("FindById/{id:int?}")]

public sub\_mark GetMarkByID(int id = 1)

{

marks empbal = new marks();

empbal = obj.search(id);

sub\_mark emp = new sub\_mark();

//emp.Id = empbal.Id;

emp.student\_id = id;

emp.student\_name = empbal.student\_name;

emp.subject\_marks = empbal.subject\_marks;

return emp;

}

// POST api/<controller>

public HttpResponseMessage PostMarks([FromBody] sub\_mark empdata)

{

marks empbal = new marks();

empbal.student\_id = empdata.student\_id;

empbal.student\_name = empdata.student\_name;

empbal.subject\_marks = empdata.subject\_marks;

bool ans = obj.AddE(empbal);

if (ans)

{

return Request.CreateResponse(HttpStatusCode.OK);

}

else

{

return Request.CreateResponse(HttpStatusCode.NotAcceptable);

}

}

// PUT api/<controller>/5

public HttpResponseMessage PutMarks([FromBody] sub\_mark empdata)

{

marks empbal = new marks();

empbal.student\_id = empdata.student\_id;

empbal.student\_name = empdata.student\_name;

empbal.subject\_marks = empdata.subject\_marks;

bool ans = obj.Edit(empbal);

if (ans)

{

return Request.CreateResponse(HttpStatusCode.OK);

}

else

{

return Request.CreateResponse(HttpStatusCode.NotAcceptable);

}

}

// DELETE api/<controller>/5

public HttpResponseMessage DeleteProduct(int id)

{

bool ans = obj.remove(id);

if (ans)

{

return Request.CreateResponse(HttpStatusCode.OK);

}

else

{

return Request.CreateResponse(HttpStatusCode.NotAcceptable);

}

}

}

}

**API:**

**Sub\_mark.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace API.Models

{

public class sub\_mark

{

public int student\_id { get; set; }

public string student\_name { get; set; }

public int subject\_marks { get; set; }

}

}

**Clientdemo:**

**Markcontroller:**

using Clientdemo.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Http;

using System.Web;

using System.Web.Mvc;

namespace Clientdemo.Controllers

{

public class markController : Controller

{

// GET: mark

public ActionResult Index()

{

List<mark> emplist = new List<mark>();

using (var client = new HttpClient())

{

client.BaseAddress = new Uri("https://localhost:44300/api/");

var responseTask = client.GetAsync("Sub");

responseTask.Wait();

var result = responseTask.Result;

if (result.IsSuccessStatusCode)

{

var readData = result.Content.ReadAsAsync<mark[]>();

readData.Wait();

var empdata = readData.Result;

foreach (var item in empdata)

{

emplist.Add(new mark

{

student\_id = item.student\_id,

student\_name = item.student\_name,

subject\_marks = item.subject\_marks

});

}

}

}

return View(emplist);

}

public ActionResult Create()

{

return View();

}

[HttpPost]

public ActionResult Create(mark empmodel)

{

using (var client = new HttpClient())

{

client.BaseAddress = new Uri("https://localhost:44300/api/Sub");

var emp = new mark

{

student\_id = empmodel.student\_id,

student\_name = empmodel.student\_name,

subject\_marks = empmodel.subject\_marks

};

var postTask = client.PostAsJsonAsync<mark>(client.BaseAddress, emp);

postTask.Wait();

var result = postTask.Result;

if (result.IsSuccessStatusCode)

{

var readtaskResult = result.Content.ReadAsAsync<mark>();

readtaskResult.Wait();

var dataInserted = readtaskResult.Result;

}

}

return RedirectToAction("Index");

}

}

}

**Mark.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace Clientdemo.Models

{

public class mark

{

public int student\_id { get; set; }

public string student\_name { get; set; }

public int subject\_marks { get; set; }

}

}

**Views:**

**Index.cshtml:**

@model IEnumerable<Clientdemo.Models.mark>

@{

ViewBag.Title = "Index";

}

<h2>Index</h2>

<p>

@Html.ActionLink("Create New", "Create")

</p>

<table class="table">

<tr>

<th>

@Html.DisplayNameFor(model => model.student\_id)

</th>

<th>

@Html.DisplayNameFor(model => model.student\_name)

</th>

<th>

@Html.DisplayNameFor(model => model.subject\_marks)

</th>

<th></th>

</tr>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item.student\_id)

</td>

<td>

@Html.DisplayFor(modelItem => item.student\_name)

</td>

<td>

@Html.DisplayFor(modelItem => item.subject\_marks)

</td>

<td>

@Html.ActionLink("Edit", "Edit", new { id=item.student\_id }) |

@Html.ActionLink("Details", "Details", new { id=item.student\_id }) |

@Html.ActionLink("Delete", "Delete", new { id=item.student\_id })

</td>

</tr>

}

</table>

**Create.cshtml:**

@model Clientdemo.Models.mark

@{

ViewBag.Title = "Create";

}

<h2>Create</h2>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-horizontal">

<h4>mark</h4>

<hr />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.student\_id, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.student\_id, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.student\_id, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.student\_name, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.student\_name, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.student\_name, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.subject\_marks, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.subject\_marks, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.subject\_marks, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<input type="submit" value="Create" class="btn btn-default" />

</div>

</div>

</div>

}

<div>

@Html.ActionLink("Back to List", "Index")

</div>

@section Scripts {

@Scripts.Render("~/bundles/jqueryval")

}