Web Application Deployed via Jenkins

Practice Project

Source code:

Pizza.cs:

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

namespace BAL
{
    public class Pizza
    {
        public int Order_id { get; set; }
        public string Food_name { get; set; }

        public int Cost { get; set; }



    }
}
```

DAL1.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 DAL1
    {
        public bool Insert(Pizza school)
        {
            SqlConnection cn = new SqlConnection("Data Source=DESKTOP-
5GL4B5D\\SQLEXPRESS1;Initial Catalog=Order_pizza;Integrated Security=True");

            SqlCommand cmdInsert = new SqlCommand("insert into
Pizza(Order_id,Food_name,Cost) values(@Order_id,@Food_name,@Cost)", cn);
            cmdInsert.Parameters.AddWithValue("@Order_id", school.Order_id);
            cmdInsert.Parameters.AddWithValue("@Food_name", school.Food_name);
            cmdInsert.Parameters.AddWithValue("@cost", school.Cost);
```

```
            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(Pizza school)
        {
            SqlConnection cn = new SqlConnection("Data Source=DESKTOP-
5GL4B5D\\SQLEXPRESS1;Initial Catalog=Order_pizza;Integrated Security=True");



            SqlCommand cmdUpdate = new SqlCommand("[dbo].[Update]", cn);

            cmdUpdate.CommandType = System.Data.CommandType.StoredProcedure;
            cmdUpdate.Parameters.AddWithValue("@p_id", school.Order_id);
            cmdUpdate.Parameters.AddWithValue("@p_name", school.Food_name);
            cmdUpdate.Parameters.AddWithValue("@p_cost", school.Cost);


            cn.Open();
            int s = cmdUpdate.ExecuteNonQuery();
            bool statusd = false;
            if (s == 1)
            {
                statusd = true;
            }
            cn.Close();//finally
            cn.Dispose();//finally
            return statusd;

        }

        public Pizza Find(int id)
        {
            SqlConnection cn = new SqlConnection("Data Source=DESKTOP-
5GL4B5D\\SQLEXPRESS1;Initial Catalog=Order_pizza;Integrated Security=True");

            SqlCommand cmdSelect = new SqlCommand("[dbo].sp_Find", cn);
            cmdSelect.CommandType = System.Data.CommandType.StoredProcedure;
            cmdSelect.Parameters.AddWithValue("@p_id", id);
```

```
            SqlParameter p1 = new SqlParameter();
            p1.ParameterName = "@p_name";
            p1.SqlDbType = System.Data.SqlDbType.NVarChar;
            p1.Size = 10;
            p1.Direction = System.Data.ParameterDirection.Output;
            cmdSelect.Parameters.Add(p1);


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




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

           Pizza found = new Pizza();

            found.Food_name = p1.Value.ToString();
            found.Cost = Convert.ToInt32(p2.Value);




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


            return found;



        }
        public List<Pizza> List()
        {

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


            SqlCommand cmdlist = new SqlCommand("select Order_id,Food_name,Cost from
Pizza", cn);

            cn.Open();
            SqlDataReader dr = cmdlist.ExecuteReader();
            List<Pizza> emplist = new List<Pizza>();
            if (dr.HasRows)
            {
                while (dr.Read())
                {
                    Pizza bal = new Pizza();
```

```
                    bal.Order_id = Convert.ToInt32(dr["Order_id"]);
                    bal.Food_name = dr["Food_name"].ToString();
                    bal.Cost = Convert.ToInt32(dr["Cost"]);


                    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=Order_pizza;Integrated Security=True");

            SqlCommand cmdDelete = new SqlCommand("[dbo].sp_Delete", 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;

        }
    }
    }
```

Helperclass.cs:

```
using BAL;
using DAL;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Helper
{
    public class Helperclass
    {
        DAL1 dal = null;
        public Helperclass()
        {
            dal = new DAL1();
        }
```

```
        public bool AddE(Pizza school)
        {
            return dal.Insert(school);

        }

        public bool Edit(Pizza school)
        {
            return dal.Update(school);
        }

        public Pizza search(int id)
        {
            return dal.Find(id);
        }
        public List<Pizza> List()
        {
            return dal.List();
        }
        public bool remove(int id)
        {
            return dal.Delete(id);
        }
    }
}
```

Homecontroller:

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

namespace JenkinsWebApplication.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult About()
        {
            ViewBag.Message = "Pizza Hub.";

            return View();
        }

        public ActionResult Contact()
        {
            ViewBag.Message = "Contact details.";

            return View();
        }
    }
```

```
}
```

pizzaController:

```
using BAL;
using Helper;
using JenkinsWebApplication.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace JenkinsWebApplication.Controllers
{
    public class pizzaController : Controller
    {
        // GET: pizza
        Helperclass helper = null;
        public pizzaController()
        {
            helper = new Helperclass();
        }

        public ActionResult Index()
        {
            var stulist = helper.List();
            List<Pizza1> modelsList = new List<Pizza1>();
            foreach (var item in stulist)
            {
                modelsList.Add(new Pizza1
                {
                    Order_id = item.Order_id,
                   Food_name = item.Food_name,
                    Cost = item.Cost,

                });
            }

            return View(modelsList);
        }
        public ActionResult Pay()
        {
            return View();
        }
        public ActionResult Details(int id)
        {
            var data = helper.search(id);
            Pizza1 emp = new Pizza1();
            emp.Order_id = id;
            emp.Food_name = data.Food_name;
            emp.Cost = data.Cost;


            return View(emp);
```

```
}


public ActionResult Create()
{
    return View();
}


[HttpPost]
public ActionResult Create(FormCollection collection)
{
    Pizza bal = new Pizza();
    bal.Order_id = Convert.ToInt32(Request["Order_id"]);
    bal.Food_name = Request["Food_name"].ToString();
    bal.Cost = Convert.ToInt32(Request["Cost"]);




    bool ans = helper.AddE(bal);
    if (ans)
    {
        return RedirectToAction("Index");
    }
    else
    {
        return View();
    }
}

public ActionResult Edit(int id)
{
    var emp = helper.search(id);
    Pizza1 model = new Pizza1();
    model.Order_id= id;
    model.Food_name = emp.Food_name;
    model.Cost = emp.Cost;



    return View(model);
}


[HttpPost]
public ActionResult Edit(int id, FormCollection collection)
{
    try
    {
        var emp = helper.search(id);
        emp.Order_id = Convert.ToInt32(Request["Order_id"]);
        emp.Food_name = Request["Food_name"].ToString();
        emp.Cost = Convert.ToInt32(Request["Cost"]);

        bool ans = helper.Edit(emp);
```

```
        if (ans)
        {
            return RedirectToAction("Index");
        }
        else
        {
            return View();
        }


    }
    catch
    {
        return View();
    }
}


public ActionResult Delete(int id)
{
    var emp = helper.search(id);
   Pizza1 model = new Pizza1();
    model.Order_id = id;
    model.Food_name = emp.Food_name;
    model.Cost = emp.Cost;



    return View(model);
}


[HttpPost]
public ActionResult Delete(int id, FormCollection collection)
{
    try
    {
        var dataFound = helper.search(id);
        if (dataFound != null)
        {
            bool ans = helper.remove(id);
            if (ans)
            {
                return RedirectToAction("Index");
            }
            else
            {
                return View();
            }
        }

        return RedirectToAction("Index");
    }
    catch
    {
        return View();
    }
}
```

```
    }
}
```

Model:

Class1.cs:

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

namespace JenkinsWebApplication.Models
{
    public class Pizza1
    {
        public int Order_id { get; set; }
        public string Food_name { get; set; }

        public int Cost { get; set; }



    }
}
```

Views:

Create.cshtml:

```
@model JenkinsWebApplication.Models.Pizza1

@{
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Add Order</h2>


@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>Pizza1</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.Order_id, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Order_id, new { htmlAttributes = new
{ @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Order_id, "", new { @class
= "text-danger" })
            </div>
```

```
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.Food_name, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Food_name, new { htmlAttributes = new
{ @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Food_name, "", new {
@class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.Cost, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Cost, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Cost, "", new { @class =
"text-danger" })
            </div>
        </div>

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Place Order" class="btn btn-default" />
            </div>
        </div>
    </div>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

Delete.cshtml:

```
@model JenkinsWebApplication.Models.Pizza1

@{
    ViewBag.Title = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Delete Order</h2>

<h3>Are you sure you want to delete this?</h3>
<div>
    <h4>Pizza1</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>
```

```
            @Html.DisplayNameFor(model => model.Order_id)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Order_id)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.Food_name)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Food_name)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.Cost)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Cost)
        </dd>

    </dl>

    @using (Html.BeginForm()) {
        @Html.AntiForgeryToken()

        <div class="form-actions no-color">
            <input type="submit" value="Delete" class="btn btn-default" /> |
            @Html.ActionLink("Back to List", "Index")
        </div>
    }
</div>
```

Edit.cshtml:

```
@model JenkinsWebApplication.Models.Pizza1

@{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Update My Order Details</h2>


@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>Pizza1</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
```

```
            @Html.LabelFor(model => model.Order_id, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Order_id, new { htmlAttributes = new
{ @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Order_id, "", new { @class
= "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.Food_name, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Food_name, new { htmlAttributes = new
{ @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Food_name, "", new {
@class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.Cost, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Cost, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Cost, "", new { @class =
"text-danger" })
            </div>
        </div>

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Save" class="btn btn-default" />
            </div>
        </div>
    </div>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

Details.cshtml:

```
@model JenkinsWebApplication.Models.Pizza1

@{
    ViewBag.Title = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
```

```
<h2>Order History</h2>

<div>
    <h4>Pizza1</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.Order_id)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Order_id)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.Food_name)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Food_name)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.Cost)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Cost)
        </dd>

    </dl>
</div>
<p>
    @Html.ActionLink("Edit", "Edit", new { id = Model.Order_id }) |
    @Html.ActionLink("Back to List", "Index")
</p>
```

Index.cshtml:

```
@model IEnumerable<JenkinsWebApplication.Models.Pizza1>

@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Pizza Hub</h2>

<p>
    @Html.ActionLink("Place order", "Create")
</p>
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.Order_id)
        </th>
        <th>
```

```
            @Html.DisplayNameFor(model => model.Food_name)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Cost)
        </th>
        <th></th>
    </tr>

@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.Order_id)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Food_name)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Cost)
        </td>
        <td>
            @Html.ActionLink("Edit", "Edit", new {  id=item.Order_id }) |
            @Html.ActionLink("Details", "Details", new { id = item.Order_id }) |
            @Html.ActionLink("Delete", "Delete", new { id = item.Order_id })
        </td>
    </tr>
}

</table>
```

Home view:

Index.cshtml:

```
@{
    ViewBag.Title = "Home Page";
}

<div class="jumbotron">
    <h1>Hurry up for your next meal !!</h1>
    <div>
        <img src="~/Images/pizza.jpg" style="width:500px;height:400px;">
        <img src="~/Images/pizza2.jpg" style="width:500px;height:400px;" >



    </div>
    <p class="lead">Enjoy !! New deals every week - Place your order and sit and
entertain the weekend </p>
    <p><a href="https://asp.net" class="btn btn-primary btn-lg">See more
&raquo;</a></p>
</div>

<div class="row">
    <div class="col-md-4">
        <h2>Order Pizza Now..</h2>
        <p>
```

```
            If you, like everyone else in the world, can never say no to pizza, then
this list is just what you've been looking for. Pizza, as we know, can help us
through heartbreaks, make Netflix binges more delightful and also is just such a
great dish to share with friends when you are hanging out together. Hence, it only
makes sense for us to tell you where to get the best pizzas in Chennai because
friends help friends find the best of the best.
            Read on.
            You don't have to thank us.
        </p>
        <p><a class="btn btn-default"
href="https://go.microsoft.com/fwlink/?LinkId=301865">See more &raquo;</a></p>
    </div>
    <div class="col-md-4">
        <h2>
            Tuscana Pizzeria
        </h2>
        <p>
            This restaurant boasts a theme inspired by The Godfather, which in
itself makes it attractive.
            And then you take a look at their pizza offerings, and you're sold.
            They've got seafood pizzas, for those who are feeling adventurous, as
well as dessert pizzas because those are now a thing and we love it! But if you're a
traditional pizza eater, don't worry, we've got you covered because they have many
options to choose from in that category as well. Give the Pollo E Pesto or the Peri
Peri Veg a try, and you won't be sorry.
            Glutton for food? Try their Monster 30-inch pizzas!
        </p>
        <p><a class="btn btn-default"
href="https://go.microsoft.com/fwlink/?LinkId=301866">Learn more &raquo;</a></p>
    </div>
    <div class="col-md-4">
        <h2>Piccante</h2>
        <p>
            Ever craved a pizza but found it to be too much to finish it all by
yourself?
            Head to Chicago Pizza which sells single slice pizzas from an 18-inch
round piece with your favourite toppings.
            You could go for their non-veg overload, which features different kinds
of spicy, meaty toppings or you could go for the simple old margarita. The place is
mostly for take-away though, so you could grab a few slices to head home and enjoy
the lot with your favourite episode of FRIENDS.
        </p>
        <p><a class="btn btn-default"
href="https://go.microsoft.com/fwlink/?LinkId=301867">Learn more &raquo;</a></p>
    </div>
```