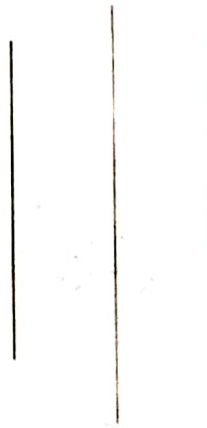


sagarmatha

COLLEGE OF SCIENCE & TECHNOLOGY

TRIBHUVAN UNIVERSITY AFFILIATE



Lab Report On: NCC

Lab Report No: 03

Submitted By:

Bijay Shrestha

Roll no: 11

Group: BSc.CSIT

Submitted To:

CSIT Department

Date: *Nov 28*

1) WAP to show the use ADD.NET in your .NET Program with either ASP.NET Web Forms Project or ASP.NET Core MVC Project.

a) Create Database & Table in SQL Server

→ The query to create database is :-

```
CREATE database Lab3dbdb
```

→ The query to create table is :-

```
CREATE TABLE dbo.Employee (
```

```
Id int IDENTITY (1,1) NOT NULL,
```

```
EmployeeName nvarchar(100) NULL,
```

```
Department nvarchar(50) NULL,
```

```
Salary nvarchar(50) NULL,
```

```
JoinedOn date NULL,
```

```
)
```

b.) Create a form to get user's data input and insert into table

→ In Controller folder a new controller is created named as EmployeeController.cs.

→ A constructor is created with a field member as :-

```

public class EmployeeController : Controller
{
    public IConfiguration Configuration { get; }
    public EmployeeController ( IConfiguration
        configuration )
    {
        Configuration = configuration;
    }
    :
}

```

→ To insert into table, at first a action method
Create is created as :-

```

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

```

```

[HttpPost]
public IActionResult Create (Employee employee)
{
    if (ModelState.IsValid)
    {
        string connectionString = Configuration ["Connect
            ionStrings: DefaultConnection"];
        using (SqlConnection connection = new SqlConnection
            (connectionString))
    }
}

```

```

{
    string sql = $"Insert Into Employee (
        EmployeeName, Department, Salary, JoinedOn)
    Values (' {employee.EmployeeName} ', '{employee
        Department} ', '{employee.Salary} ', '{employee.
        JoinedOn} ' ) ;
    using (SqlCommand command = new SqlCommand
        (sql, connection))

```

```

{
    command.CommandType = CommandType.Text ;
    connection.Open ();
    command.ExecuteNonQuery ();
    connection.Close ();

```

```

    }
    return RedirectToAction ("Index");
}
else
    return View ();

```

```

}

```

→ The View ~~Index~~ ^{Create}.cshtml is created with the code as :-

```

@model Employee

```

```

@{
    ViewData ["Title"] = "Create" ;

```

```

    var title = "Add Employee" ;

```

```

    layout = " ~/Views/Shared/_Layout.cshtml" ;
}

```



```
<style>
    .input-validation-error {
        border-color : red;
    }
</style>
```

```
<h2> @title </h2>
```

```
<div asp-validation-summary = "ModelOnly" class = "text-danger">
```

```
</div>
```

```
<form class = "m-1 p-1" method = "post">
```

```
    <div class = "form-group">
```

```
        <label asp-for = "EmployeeName"> 2) label>
```

```
        <input asp-for = "EmployeeName" class = "form-control">
```

```
        <span asp-validation-for = "EmployeeName" class = "text-
            danger"> </span>
```

```
    </div>
```

```
    <div class = "form-group">
```

```
        <label asp-for = "Department"> </label>
```

```
        <input asp-for = "Department" class = "form-control">
```

```
        <span asp-validation-for = "Department" class = "text-danger"
            > </span>
```

```
    </div>
```

```
    <div class = "form-group">
```

```
        <label asp-for = "Salary"> </label>
```

```
        <input asp-for = "Salary" type = "text" class = "form-control">
```

```
        <span asp-validation-for = "Salary" class = "text-danger"
            > </span>
```

```
    </div>
```

```
    <button type = "submit" class = "btn btn-primary"> Submit
```

```
    </button>
```

```
</form>
```

c.) Show Data from your table in your list page.

→ Action method to show data is :-

```
public IActionResult Index ()
{
    List <Employee> employeeList = new List <Employee> ();
    string connectionString = Configuration [ "ConnectionStrings :
        Default Connection " ];
    using (SqlConnection connection = new SqlConnection (
        connectionString))
    {
        connection.Open ();
        string sql = "Select * From Employee ";
        SqlCommand command = new SqlCommand (sql, connection);
        using (SqlDataReader dataReader = command.ExecuteReader ())
        {
            while (dataReader.Read ())
            {
                Employee employee = new Employee ();
                employee.Id = Convert.ToInt32 (dataReader ["Id"]);
                employee.EmployeeName = Convert.ToString (dataReader [
                    "Employee Name"]);
                employee.DepartmentSalary = Convert.ToDecimal (dataReader [
                    "Salary"]);
                employee.Department = Convert.ToString (dataReader [
                    "Department"]);
                employee.JoinedOn = Convert.ToDateTime (dataReader [
                    "Joined On"]);
                employeeList.Add (employee);
            }
        }
        connection.Close ();
        return View (employeeList);
    }
}
```

→ View for Index ie Index.cshtml is :-

```
@model IEnumerable<Employee>
```

```
@{ ViewData["Title"] = "Employee";
```

```
var title = "Employee List";
```

```
layout = "~/Views/Shared/_Layout.cshtml";
```

```
}
```

```
</h2>
```

```
@title (Asp.net core 3.1 crud operation with  
Addo.net
```

```
)
```

```
</h2>
```

```
<h3> <a asp-action = "Create" class = "btn btn-sm  
btn-secondary" > Create </a> </h3>
```

```
<table class = "table table-bordered table-sm table-  
striped">
```

```
<thead>
```

```
<tr> <th>Id</th> <th>Name</th> <th>Department</th>
```

```
<th>Salary</th> <th>Joining On</th> <th>Update
```

```
</th> <th>Delete</th> </tr>
```

```
</thead>
```

```
<tbody>
```

```
@if (Model == null)
```

```
{ <tr> <td colspan = "7" class = "text-center">  
No Model Data </td> </tr>
```

```
}
```

```
else
```

```
{ @foreach (var p in Model)
```

```
{ <tr>
```

```
<td> @p.Id </td>  
<td> @p.EmployeeName </td>  
<td> @p.Department </td>  
<td> Rs. @p.Salary </td>  
<td> @string.Format("{0:dddd, dd MMMM yyyy}",  
    p.JoinedOn) </td>
```

```
<td> <a asp-action = "Update" asp-route-id = "  
    @p.Id"> Update </a> </td>
```

```
<td>  
    <form asp-action = "Delete" method = "post"  
        asp-route-id = "@p.Id">  
        <button> Delete </button>
```

```
</form>
```

```
</td>
```

```
</tr>
```

```
<tbody>
```

```
</tbody>
```

```
</table>
```

```
</table>
```


d.) Update Data through your edit page.

→ Action method for Update is :-

```
public IActionResult Update (int id)
```

```
{ string connectionString = Configuration ["ConnectionStrings: Default Connection"];
```

```
Employee employee = new Employee ();
```

```
using (SqlConnection connection = new SqlConnection (connectionString))
```

```
{ string sql = $"Select * From Employee where Id = {id}";
```

```
SqlCommand command = new SqlCommand (sql, connection);
```

```
connection.Open ();
```

```
using (SqlDataReader dataReader = command.ExecuteReader ())
```

```
{ while (dataReader.Read ())
```

```
{ employee.Id = Convert.ToInt32 (dataReader ["Id"]);
```

```
employee.EmployeeName = Convert.ToString (dataReader ["EmployeeName"]);
```

```
employee.Department = Convert.ToString (dataReader ["Department"]);
```

```
employee.Salary = Convert.ToDecimal (dataReader ["Salary"]);
```

```
employee.JoinedOn = Convert.ToDateTime (dataReader ["JoinedOn"]);
```

```
}
```

```
connection.Close (); return View (employee); }
```

[HttpPost]

[ActionName("Update")]

public ActionResult UpdatePost(Employee employee)

{
 string connectionString = Configuration["ConnectionStrings"]
 : DefaultConnection "];

using (SqlConnection connection = new SqlConnection(
 connectionString))

{
 string sql = \$"Update Employee SET EmployeeName
 = '{employee.EmployeeName}', Department = '{
 employee.Department}', Salary = '{employee.
 Salary}', Where Id = '{employee.Id}'";

using (SqlCommand command = new SqlCommand(
 sql, connection))

{
 connection.Open();
 command.ExecuteNonQuery();
 connection.Close();

return RedirectToAction("Index");

}

→ View page for update i.e. Update.cshtml is :-

```
@model Employee
```

```
@{ ViewData["Title"] = "Update";
```

```
var title = "Update Employee";
```

```
layout = "~/Views/Shared/_Layout.cshtml";
```

```
}
```

```
<h2>@title </h2>
```

```
<div asp-validation-summary = "ModelOnly" class = "text-danger">
```

```
</div>
```

```
<form class = "m-1 p-1" method = "post">
```

```
<div class = "form-group">
```

```
<label asp-for = "Id"></label>
```

```
<input asp-for = "Id" type = "text" readonly class = "form-control" />
```

```
</div>
```

```
<div class = "form-group">
```

```
<label asp-for = "EmployeeName"></label>
```

```
<input asp-for = "EmployeeName" type = "text" class = "form-control" />
```

```
<span asp-validation-for = "EmployeeName" class = "text-danger"></span>
```

```
</div>
```

```
<div class = "form-group">
```

```
<label asp-for = "EmployeeName Department"></label>
```

```
<input asp-for = "Department" type = "text" class = "form-control" />
```

```
<span asp-validation-for = "Department" class = "text-danger"></span>
```

```
</div>
```

```

<div class = "form-group">
  <label asp-for = "Salary"> </label>
  <input asp-for = "Salary" type = "text" class =
    "form-control" />
  <span asp-validation-for = "Salary" class = "text-danger">
    </span>

```

```

</div>

```

```

<div class = "form-group">
  <label asp-for = "JoinedOn"> </label>
  <input asp-for = "JoinedOn" type = "text" class =
    "form-control" asp-format = "{0:d5}" />

```

```

</div>

```

```

<button type = "submit" class = "btn btn-primary">
  Submit </button>

```

```

</form>

```

e.) Delete a row of data from your given input data.

→ In a list of ~~ro~~data of employee, there is a button which on click deletes a row of data.

→ Action method of it is given below :-

```

[HttpPost]

```

```

public IActionResult Delete (int id)

```

```

{

```



```
string connectionString = configuration["connectionStrings"]  
: DefaultConnection "I";
```

```
using (SqlConnection connection = new SqlConnection  
(connectionString))
```

```
{ string sql = $"Delete From Employee Where Id =  
    '{Id}'";
```

```
using (SqlCommand command = new SqlCommand  
(sql, connection))
```

```
{ connection.Open();
```

```
    try
```

```
    { command.ExecuteNonQuery();
```

```
    }  
    catch (SqlException ex)
```

```
    { ViewBag.Result = "Operation got error:" + ex.Message;
```

```
    }  
    connection.Close();
```

```
}
```

```
return RedirectToAction("Index");
```

```
}
```

The model class Employee is

```
public class Employee
{
    public int Id { get ; set ; }
    [Required]
    [Display (Name = " Employee Name ")]
    public string EmployeeName { get ; set ; }
    [Required]
    public string Department { get ; set ; }
    [Required]
    public decimal Salary { get ; set ; }
    public DateTime JoinedOn { get ; set ; }
}
```

In appSettings.json, connection string is added as

```
{
  "allowedHosts" : "*",
  "connectionStrings" : {
    "Default Connection" : "Data Source = . ; Initial
                           Catalog = lab3db ; Integrated Security = True"
  }
}
```