

User Registration Crud Application

1) Create Table Users

```
CREATE TABLE Users (  
    Id INT PRIMARY KEY IDENTITY(1,1),  
    FirstName VARCHAR(30) NOT NULL,  
    LastName VARCHAR(30) NOT NULL,  
    DOB DATE NOT NULL,  
    Gender VARCHAR(20) NOT NULL,  
    PhoneNumber VARCHAR(15) NOT NULL,  
    Email VARCHAR(50) UNIQUE NOT NULL,  
    Address VARCHAR(100) NOT NULL,  
    City VARCHAR(40) NOT NULL,  
    UserName VARCHAR(30) UNIQUE NOT NULL,  
    Password NVARCHAR(255) NOT NULL  
);
```

2) Create Procedure

----- INSERT USER DETAILS-----

```
CREATE PROCEDURE usp_InsertUser  
    @Id int ,  
    @FirstName VARCHAR(30) ,  
    @LastName VARCHAR(30) ,  
    @DOB DATE ,  
    @Gender VARCHAR(20),  
    @PhoneNumber VARCHAR(15) ,  
    @Email VARCHAR(50) ,
```

```
@Address VARCHAR(100) ,  
@City VARCHAR(50) ,  
@UserName VARCHAR(30) ,  
@Password NVARCHAR(255)
```

```
AS
```

```
BEGIN
```

```
INSERT INTO
```

```
Users(FirstName,LastName,DOB,Gender,PhoneNumber,Email,Address,Ci  
ty,UserName,Password)
```

```
VALUES(@FirstName, @LastName, @DOB, @Gender,@PhoneNumber  
,@Email,@Address,@City,@UserName,@Password);
```

```
END
```

```
----- GET ALL USERS -----
```

```
CREATE PROCEDURE usp_GetAllUsers
```

```
AS
```

```
BEGIN
```

```
    SELECT * FROM Users;
```

```
END
```

```
-----GET USERS BY ID-----
```

```
CREATE PROCEDURE usp_GetUserById
```

```
@Id int
```

```
AS
```

```
BEGIN
```

```
    SELECT * FROM Users WHERE Id=@Id;
```

```
END
```

----- **UPDATE USER DETAILS** -----

```
CREATE PROCEDURE usp_UpdateUser
    @Id INT,
    @FirstName VARCHAR(30) ,
    @LastName VARCHAR(30) ,
    @DOB DATE ,
    @Gender VARCHAR(20),
    @PhoneNumber VARCHAR(15) ,
    @Email VARCHAR(50) ,
    @Address VARCHAR(100) ,
    @City VARCHAR(50) ,
    @UserName VARCHAR(30) ,
    @Password NVARCHAR(255)
AS
BEGIN
    UPDATE Users SET
    FirstName=@FirstName,LastName=@LastName,DOB=@DOB,Gender=@
    Gender,PhoneNumber=@PhoneNumber,

    Email=@Email,Address=@Address,City=@City,UserName=@UserName,P
    assword=@Password
    WHERE Id=@Id;
END
```

-----**DELETE USER** -----

```
CREATE PROCEDURE usp_DeleteUser
    @Id INT

AS
BEGIN
    DELETE FROM Users WHERE Id=@Id;
END
```

3) Create Model

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

```
namespace UserRegistrationCrud.Models
```

```
{
```

```
    public class User
```

```
    {
```

```
        [Key]
```

```
        public int Id { get; set; }
```

```
            [Required]
```

```
            public String FirstName { get; set; }
```

```
        [Required]
```

```
        public String LastName { get; set; }
```

```
        [Required]
```

```
        public DateTimeOffset DOB { get; set; }
```

```
        [Required]
```

```
        public String Gender { get; set; }
```

```
        [Required]
```

```
        [MaxLength(15)]
```

```
        public String PhoneNumber { get; set; }
```

```
        [Required]
```

```
        [EmailAddress]
```

```
        public String Email { get; set; }
```

```
    [Required]
```

```

[StringLength(50)]
public String Address { get; set; }

[Required]
public String City { get; set; }

[Required]
[MaxLength(40)]
public String UserName { get; set; }

[Required]
[MaxLength(255)] //hash password
public String Password { get; set; }
}
}

```

4) **Create Repository**

```

using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Security.Cryptography.X509Certificates;
using System.Web;
using UserRegistrationCrud.Models;

namespace UserRegistrationCrud.Repository
{
    public class UserRepository
    {
        //configure database conection.....

        String constring =
        ConfigurationManager.ConnectionStrings["UserDbConnection"].ToString(
        );
    }
}

```

// GET ALL USER DETAILS.....

```
public List<User> GetAllUsers()
{
    List<User> userList = new List<User>();
    using (SqlConnection con = new SqlConnection(constring))
    {
        SqlCommand cmd = new SqlCommand("usp_GetAllUsers", con);
        cmd.CommandType = CommandType.StoredProcedure;
        con.Open();
        SqlDataReader reader = cmd.ExecuteReader();
        while (reader.Read())
        {
            userList.Add(new User
            {
                Id = Convert.ToInt32(reader["Id"]),
                FirstName = reader["FirstName"].ToString(),
                LastName = reader["LastName"].ToString(),
                DOB = Convert.ToDateTime(reader["DOB"]),
                Gender = reader["Gender"].ToString(),
                PhoneNumber = reader["PhoneNumber"].ToString(),
                Email = reader["Email"].ToString(),
                Address = reader["Address"].ToString(),
                City = reader["City"].ToString(),
                UserName = reader["UserName"].ToString(),
                Password = reader["Password"].ToString()
            });
        }
    }
    return userList;
}
```

// GET USER DETAILS BY ID.....

```
public User GetUserById(int id)
{
    using(SqlConnection con=new SqlConnection(constring))
    {
        User user = null;
        SqlCommand cmd = new SqlCommand("usp_GetUserById", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@Id", id);

        con.Open();

        SqlDataReader reader = cmd.ExecuteReader();

        if (reader.Read())
        {
            user = new User{
                FirstName = reader["FirstName"].ToString(),
                LastName = reader["LastName"].ToString(),
                DOB = Convert.ToDateTime(reader["DOB"]),
                Gender = reader["Gender"].ToString(),
                PhoneNumber = reader["PhoneNumber"].ToString(),
                Email = reader["Email"].ToString(),
                Address = reader["Address"].ToString(),
                City = reader["City"].ToString(),
                UserName = reader["UserName"].ToString(),
                Password = reader["Password"].ToString()

            };
        }
        return user;
    }
}
```

// INSERT USER DETAILS.....

```
public void InsertUser(User user)
{
    using (SqlConnection con = new SqlConnection(constring))
    {
        SqlCommand cmd = new SqlCommand("usp_InsertUser", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@Id", user.Id);
        cmd.Parameters.AddWithValue("@FirstName", user.FirstName);
        cmd.Parameters.AddWithValue("@LastName", user.LastName);
        cmd.Parameters.AddWithValue("@DOB", user.DOB);
        cmd.Parameters.AddWithValue("@Gender", user.Gender);
        cmd.Parameters.AddWithValue("@PhoneNumber", user.PhoneNumber);
        cmd.Parameters.AddWithValue("@Email", user.Email);
        cmd.Parameters.AddWithValue("@Address", user.Address);
        cmd.Parameters.AddWithValue("@City", user.City);
        cmd.Parameters.AddWithValue("@UserName", user.UserName);
        cmd.Parameters.AddWithValue("@Password", user.Password);

        con.Open();

        cmd.ExecuteNonQuery()
    }
}
```

// UPDATE USER DETAILS.....

```
public void UpdateUser(User user)
{
    using (SqlConnection con = new SqlConnection(constring))
    {
```



```

        SqlCommand cmd = new SqlCommand("usp_UpdateUser", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@Id", user.Id);
        cmd.Parameters.AddWithValue("@FirstName", user.FirstName);
        cmd.Parameters.AddWithValue("@LastName", user.LastName);
        cmd.Parameters.AddWithValue("@DOB", user.DOB);
        cmd.Parameters.AddWithValue("@Gender", user.Gender);
        cmd.Parameters.AddWithValue("@PhoneNumber", user.PhoneNumber);
        cmd.Parameters.AddWithValue("@Email", user.Email);
        cmd.Parameters.AddWithValue("@Address", user.Address);
        cmd.Parameters.AddWithValue("@City", user.City);
        cmd.Parameters.AddWithValue("@UserName", user.UserName);
        cmd.Parameters.AddWithValue("@Password", user.Password);
        con.Open();
        cmd.ExecuteNonQuery();
    }
}

```

// DELETE USER DETAILS.....

```

public void DeleteUser(int id)
{
    using (SqlConnection con = new SqlConnection(constring))
    {
        SqlCommand cmd = new SqlCommand("usp_DeleteUser", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@Id", id);

        con.Open();

        cmd.ExecuteNonQuery();
    }
}

```

```
}
```

5) Configure Database Connection

```
<connectionStrings>
    <add name="UserDbConnection" connectionString="data
source=SHILPA\SQLEXPRESS ; database=UserDB; integrated
security=True;" />
</connectionStrings>
```

6) Create Controller

```
using System.Web.Mvc;
using UserRegistrationCrud.Models;
using UserRegistrationCrud.Repository;

namespace UserRegistrationCrud.Controllers
{
    public class UserController : Controller
    {
        private readonly UserRepository UserRepos = new UserRepository();

        // GET: User
        public ActionResult Index()
        {
            var users = UserRepos.GetAllUsers();
            return View(users); // Pass users to View
        }

        // GET: User/Details/5
        public ActionResult Details(int id)
        {
            var user = UserRepos.GetUserById(id);
            if (user == null) return HttpNotFound();
            return View(user);
        }
    }
}
```

```
// GET: User/Create
public ActionResult Create()
{
    return View();
}
```

```
// POST: User/Create
[HttpPost]
public ActionResult Create(User user)
{
    if (ModelState.IsValid)
    {
        UserRepos.InsertUser(user);
        return RedirectToAction("Index");
    }
    return View(user);
}
```

```
// GET: User/Edit/5
public ActionResult Edit(int id)
{
```

```
    var user = UserRepos.GetUserById(id);
    if (user == null) return HttpNotFound(); // Ensure user exists
```

```
    return View(user); // Pass user data to Edit.cshtml
}
```

```
// POST: User/Edit/5
[HttpPost]
public ActionResult Edit(User user)
{
    if (ModelState.IsValid)
    {
        UserRepos.UpdateUser(user);
    }
}
```

```

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

// GET: User/Delete/5
public ActionResult Delete(int id)
{
    var user = UserRepos.GetUserById(id);
    if (user == null) return HttpNotFound();
    return View(user);
}

// POST: User/Delete/5
[HttpPost, ActionName("Delete")]
public ActionResult DeleteConfirmed(int id)
{
    UserRepos.DeleteUser(id);
    return RedirectToAction("Index");
}
}
}

```

7) Create view

I. Index.cshtml

```

@model IEnumerable<UserRegistrationCrud.Models.User>

@{
    ViewBag.Title = "User List";
}

<h2>Index</h2>

```

```
<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.FirstName)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.LastName)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.DOB)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Gender)
        </th>
        <th>
            @Html.DisplayNameFor(model =>
model.PhoneNumber)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Email)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Address)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.City)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.UserName)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Password)
        </th>
        <th></th>
```

```
</tr>
```

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

```
<tr>
```

```
<td>
```

```
    @Html.DisplayFor(modelItem => item.FirstName)
```

```
</td>
```

```
<td>
```

```
    @Html.DisplayFor(modelItem => item.LastName)
```

```
</td>
```

```
<td>
```

```
    @Html.DisplayFor(modelItem => item.DOB)
```

```
</td>
```

```
<td>
```

```
    @Html.DisplayFor(modelItem => item.Gender)
```

```
</td>
```

```
<td>
```

```
    @Html.DisplayFor(modelItem =>
```

```
item.PhoneNumber)
```

```
</td>
```

```
<td>
```

```
    @Html.DisplayFor(modelItem => item.Email)
```

```
</td>
```

```
<td>
```

```
    @Html.DisplayFor(modelItem => item.Address)
```

```
</td>
```

```
<td>
```

```
    @Html.DisplayFor(modelItem => item.City)
```

```
</td>
```

```
<td>
```

```
    @Html.DisplayFor(modelItem => item.UserName)
```

```
</td>
```

```
<td>
```

```
    @Html.DisplayFor(modelItem => item.Password)
```

```
</td>
```

```
<td>
```

```
    @Html.ActionLink("Edit", "Edit", new { id=item.Id }) |
```

```

                @Html.ActionLink("Details", "Details", new {
id=item.Id }) |
                @Html.ActionLink("Delete", "Delete", new {
id=item.Id })
            </td>
        </tr>
    }

</table>

```

II. Create.cshtml

```
@model UserRegistrationCrud.Models.User
```

```
@{
    ViewBag.Title = "Create";
}
```

```
<h2>Create</h2>
```

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

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

```

```
        @Html.ValidationMessageFor(model =>
model.FirstName, "", new { @class = "text-danger" })
    </div>
</div>
```

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

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

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



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

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

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

```
<div class="form-group">
    @Html.LabelFor(model => model.City,
htmlAttributes: new { @class = "control-label col-md-2" })
```

```

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

```

```

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

```

```

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

```

```

}

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

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

```

III. Details.cshtml

```

@model UserRegistrationCrud.Models.User

@{
    ViewBag.Title = "Details";
}

<h2>Details</h2>

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

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

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

        <dd>

```

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

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

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

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

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

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

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

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

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

```
<dt>
```

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

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

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

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

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

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

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

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

</dl>
</div>
<p>
```

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

IV. Edit.cshtml

```
@model UserRegistrationCrud.Models.User
```

```
@{
    ViewBag.Title = "Edit";
}
```

```
<h2>Edit</h2>
```

```
@using (Html.BeginForm())
```

```
{
    @Html.AntiForgeryToken()
```

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

```
<h4>User</h4>
```

```
<hr />
```

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

```
@Html.HiddenFor(model => model.Id)
```

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

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

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

```
@Html.EditorFor(model => model.FirstName, new
{ htmlAttributes = new { @class = "form-control" } })
```

```
@Html.ValidationMessageFor(model =>
model.FirstName, "", new { @class = "text-danger" })
```

```
</div>
```

```
</div>
```

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

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

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

```
<div class="form-group">
    @Html.LabelFor(model => model.PhoneNumber,
htmlAttributes: new { @class = "control-label col-md-2" })
```

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

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

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

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



```
        @Html.ValidationMessageFor(model => model.City,
"", new { @class = "text-danger" })
    </div>
</div>
```

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

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

```
</div>
```

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

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

V. Delete.cshtml

```
@model UserRegistrationCrud.Models.User
```

```
@{
    ViewBag.Title = "Delete";
}
```

```
<h2>Delete</h2>
```

```
<h3>Are you sure you want to delete this?</h3>
```

```
<div>
```

```
    <h4>User</h4>
```

```
    <hr />
```

```
    <dl class="dl-horizontal">
```

```
        <dt>
```

```
            @Html.DisplayNameFor(model => model.FirstName)
```

```
        </dt>
```

```
        <dd>
```

```
            @Html.DisplayFor(model => model.FirstName)
```

```
        </dd>
```

```
        <dt>
```

```
            @Html.DisplayNameFor(model => model.LastName)
```

```
        </dt>
```

```
        <dd>
```

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

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

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

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

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

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

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

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

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

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

</dt>

<dd>

@Html.DisplayFor(model => model.Address)

</dd>

<dt>

@Html.DisplayNameFor(model => model.City)

</dt>

<dd>

@Html.DisplayFor(model => model.City)

</dd>

<dt>

@Html.DisplayNameFor(model => model.UserName)

</dt>

<dd>

@Html.DisplayFor(model => model.UserName)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Password)

</dt>

<dd>

@Html.DisplayFor(model => model.Password)

</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>
```

8) Run Application