﻿@{

Layout = null;

}

<head>

<link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />

<style>

.form-container {

max-width: 400px;

margin: 0 auto;

padding: 20px;

border: 1px solid #ccc;

border-radius: 5px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

body {

background-image: url("../../wwwroot/images2/7up.jpg");

background-size: cover;

background-repeat: no-repeat;

background-attachment: fixed;

}

</style>

</head>

<div class="form-container mt-5">

<h1>login</h1>

<form asp-action="Login">

<div class="form-group">

<label class="control-label"> user Name</label>

<input name="na" class="form-control" />

</div>

<div class="form-group">

<label for="pa" class="control-label"> user Password</label>

<input name="pa" class="form-control" /> <br />

</div>

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

<p style="color:red"> @ViewData["message"]</p>

<label>Not registered?</label>

<a asp-action="Create">create account</a>

</form>

</div>

-----------------------

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Mail;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Rendering;

using Microsoft.Data.SqlClient;

using Microsoft.EntityFrameworkCore;

using project.Data;

using project.Models;

namespace project.Controllers

{

public class UserallsController : Controller

{

private readonly projectContext \_context;

public UserallsController(projectContext context)

{

\_context = context;

}

public async Task<IActionResult> Index()

{

ViewData["role"] = HttpContext.Session.GetString("Role");

return View();

}

public async Task<IActionResult> email(int? id)

{

string ss = HttpContext.Session.GetString("Role");

if (ss == "admin")

{

return View();

}

else

return RedirectToAction("Warning", "Home");

}

[HttpPost, ActionName("email")]

[ValidateAntiForgeryToken]

public IActionResult email(string address, string subject, string body)

{

SmtpClient SmtpServer = new SmtpClient("smtp.gmail.com");

var mail = new MailMessage();

mail.From = new MailAddress("homm9807@gmail.com");

mail.To.Add(address); // receiver email address

mail.Subject = subject;

mail.IsBodyHtml = true;

mail.Body = body;

SmtpServer.Port = 587;

SmtpServer.UseDefaultCredentials = false;

SmtpServer.Credentials = new System.Net.NetworkCredential("homm9807@gmail.com", "mydjptatmcmypudn");

SmtpServer.EnableSsl = true;

SmtpServer.Send(mail);

ViewData["Message"] = "Email sent.";

return View();

}

public async Task<IActionResult> admin(int? id)

{

string ss = HttpContext.Session.GetString("Name");

ViewData["admin"] = ss;

return View();

}

public IActionResult customer()

{

string ss = HttpContext.Session.GetString("Name");

ViewData["customer"] = ss;

return View();

}

public IActionResult login()

{

return View();

}

[HttpPost, ActionName("login")]

[ValidateAntiForgeryToken]

public async Task<IActionResult> login(string na, string pa)

{

var builder = WebApplication.CreateBuilder();

string conStr = builder.Configuration.GetConnectionString("projectContext");

SqlConnection conn1 = new SqlConnection(conStr);

string sql;

sql = "SELECT \* FROM userall where name ='" + na + "' and password ='" + pa + "' ";

SqlCommand comm = new SqlCommand(sql, conn1);

conn1.Open();

SqlDataReader reader = comm.ExecuteReader();

if (reader.Read())

{

string id = Convert.ToString((int)reader["Id"]);

string na1 = (string)reader["name"];

string ro = (string)reader["role"];

HttpContext.Session.SetString("userid", id);

HttpContext.Session.SetString("Name", na1);

HttpContext.Session.SetString("Role", ro);

reader.Close();

conn1.Close();

if (ro == "admin")

{

return RedirectToAction(nameof(admin));

}

return RedirectToAction("Index", "item");

}

else

{

ViewData["Message"] = "wrong user name password";

return View();

}

}

// GET: Useralls

// GET: Useralls/Details/5

public async Task<IActionResult> Details(int? id)

{

if (id == null || \_context.Userall == null)

{

return NotFound();

}

var userall = await \_context.Userall

.FirstOrDefaultAsync(m => m.Id == id);

if (userall == null)

{

return NotFound();

}

return View(userall);

}

// GET: Useralls/Create

public IActionResult CreateCustomer()

{

return View();

}

// POST: Useralls/Create

// To protect from overposting attacks, enable the specific properties you want to bind to.

// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> CreateCustomer([Bind("Id, name, password, role, RegistDate")] Userall userall)

{

if (ModelState.IsValid)

{

if (\_context.Userall.Any(u => u.name == userall.name))

{

ModelState.AddModelError("name", "Name already exists. Please choose a different name.");

return View(userall);

}

\_context.Add(userall);

await \_context.SaveChangesAsync();

return RedirectToAction(nameof(login));

}

return View(userall);

}

public IActionResult CreateAdmin()

{

return View();

}

// POST: Useralls/Create

// To protect from overposting attacks, enable the specific properties you want to bind to.

// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

[HttpPost]

public async Task<IActionResult> CreateAdmin([Bind("Id, name, password, confirmPassword, role, RegistDate")] Userall userall)

{

// Check if the password and confirmed password match

if (userall.password != userall.password)

{

ModelState.AddModelError("confirmPassword", "The password and confirmation password do not match.");

}

// Check if the name already exists in the database

if (\_context.Userall.Any(u => u.name == userall.name))

{

ModelState.AddModelError("name", "Name already exists. Please choose a different name.");

}

if (ModelState.IsValid)

{

// The rest of your code here, e.g., adding to context and saving changes

\_context.Add(userall);

await \_context.SaveChangesAsync();

return RedirectToAction(nameof(login));

}

// If ModelState is not valid, return to the view with validation errors

return View(userall);

}

// GET: Useralls/Edit/5

public async Task<IActionResult> Edit(int? id)

{

if (id == null || \_context.Userall == null)

{

return NotFound();

}

var userall = await \_context.Userall.FindAsync(id);

if (userall == null)

{

return NotFound();

}

return View(userall);

}

// POST: Useralls/Edit/5

// To protect from overposting attacks, enable the specific properties you want to bind to.

// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Edit(int id, [Bind("Id,name,password,role,RegistDate")] Userall userall)

{

if (id != userall.Id)

{

return NotFound();

}

if (ModelState.IsValid)

{

try

{

\_context.Update(userall);

await \_context.SaveChangesAsync();

}

catch (DbUpdateConcurrencyException)

{

if (!UserallExists(userall.Id))

{

return NotFound();

}

else

{

throw;

}

}

return RedirectToAction(nameof(Index));

}

return View(userall);

}

// GET: Useralls/Delete/5

public async Task<IActionResult> Delete(int? id)

{

if (id == null || \_context.Userall == null)

{

return NotFound();

}

var userall = await \_context.Userall

.FirstOrDefaultAsync(m => m.Id == id);

if (userall == null)

{

return NotFound();

}

return View(userall);

}

// POST: Useralls/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public async Task<IActionResult> DeleteConfirmed(int id)

{

if (\_context.Userall == null)

{

return Problem("Entity set 'projectContext.Userall' is null.");

}

var userall = await \_context.Userall.FindAsync(id);

if (userall != null)

{

\_context.Userall.Remove(userall);

}

await \_context.SaveChangesAsync();

return RedirectToAction(nameof(Index));

}

private bool UserallExists(int id)

{

return (\_context.Userall?.Any(e => e.Id == id)).GetValueOrDefault();

}

}

}

{

public int Id { get; set; }

public string name { get; set; }

public string password { get; set; }

public string role { get; set; }

[BindProperty, DataType(DataType.Date)]

public DateTime RegistDate { get; set; }

}

Admin view

@{

ViewData["Title"] = "admin";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

<head>

<**link** rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />

<style>

.form-container {

max-width: 400px;

margin: 0 auto;

padding: 20px;

border: 1px solid #ccc;

border-radius: 5px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

</style>

</head>

<body>

<h1 class="form-container">Welcome @ViewData["admin"]</h1>

</body>

Create admin view

@model project.Models.Userall

@{

ViewData["Title"] = "CreateAdmin";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

<h1>CreateAdmin</h1>

<h4>Userall</h4>

<hr />

<div class="row">

<div class="col-md-4">

<**form** **asp-action**="CreateAdmin">

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

<div class="form-group">

<**label** **asp-for**="name" class="control-label"></**label**>

<**input** **asp-for**="name" class="form-control" />

<**span** **asp-validation-for**="name" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="password" class="control-label"></**label**>

<**input** **asp-for**="password" **type**="password" class="form-control" />

<**span** **asp-validation-for**="password" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="role" class="control-label"></**label**>

<**input** **asp-for**="role" class="form-control" **value**="admin" readonly />

<**span** **asp-validation-for**="role" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="RegistDate" class="control-label"></**label**>

<**input** **asp-for**="RegistDate" class="form-control" />

<**span** **asp-validation-for**="RegistDate" class="text-danger"></**span**>

</div>

<div class="form-group">

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

</div>

</**form**>

</div>

</div>

<div>

<**a** **asp-action**="Index">Back to List</**a**>

</div>

@section Scripts {

@{await Html.RenderPartialAsync("\_ValidationScriptsPartial");}

}

Create customer view

@model project.Models.Userall

@{

ViewData["Title"] = "Create";

}

<h1>Create</h1>

<h4>Userall</h4>

<hr />

<div class="row">

<div class="col-md-4">

<**form** **asp-action**="CreateCustomer">

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

<div class="form-group">

<**label** **asp-for**="name" class="control-label"></**label**>

<**input** **asp-for**="name" class="form-control" />

<**span** **asp-validation-for**="name" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="password" class="control-label"></**label**>

<**input** **asp-for**="password" class="form-control" />

<**span** **asp-validation-for**="password" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="role" class="control-label"></**label**>

<**input** **asp-for**="role" class="form-control" **value**="Customer" readonly />

<**span** **asp-validation-for**="role" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="RegistDate" class="control-label"></**label**>

<**input** **asp-for**="RegistDate" class="form-control" />

<**span** **asp-validation-for**="RegistDate" class="text-danger"></**span**>

</div>

<div class="form-group">

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

</div>

<p style="color:red"> @ViewData["message"]</p>

</**form**>

</div>

</div>

<div>

<**a** **asp-action**="Index">Back to List</**a**>

</div>

@section Scripts {

@{await Html.RenderPartialAsync("\_ValidationScriptsPartial");}

}

Customer view

@model IEnumerable<project.Models.item>

@{

ViewData["Title"] = "customer";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

<head>

<**link** rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />

<style>

.form-container {

max-width: 400px;

margin: 0 auto;

padding: 20px;

border: 1px solid #ccc;

border-radius: 5px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

</style>

</head>

<h1 class="form-container">Welcome @ViewData["customer"]</h1>

<div class="container-fluid">

<div class="row">

@foreach (var item in Model.Where(item => item.discount == "yes"))

{

<div class="col-md-3">

<div class="card">

<div class="card-header">

@item.name

</div>

<div class="card-body">

<**img** src="~/images/@item.imagefilename" height="150" />

</div>

<div class="card-footer">

<**a** **asp-action**="Details" **asp-route-id**="@item.Id">Details</**a**> |

</div>

</div>

</div>

}

</div>

</div>

Email view

@{

ViewData["Title"] = "email";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

<html>

<head>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<style>

.form-container {

max-width: 400px;

margin: 0 auto;

padding: 20px;

border: 1px solid #ccc;

border-radius: 5px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

body {

background-image: <<url(../../wwwroot/images2/7up.jpg)>>;

background-size: cover;

background-repeat: no-repeat;

background-attachment: fixed;

}

</style>

</head>

<body>

<div class="form-container">

<**form** **asp-action**="email" method="post">

<div class="form-group">

<label for="address" class="control-label">Address</label>

<input id="address" name="address" class="form-control" />

</div>

<div class="form-group">

<label for="body" class="control-label">Body</label>

<input id="body" name="body" class="form-control" />

</div>

<div class="form-group">

<label for="subject" class="control-label">Subject</label>

<textarea id="subject" name="subject" rows="4" cols="50"></textarea>

</div>

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

</**form**>

<p>@ViewData["message"]</p>

</div>

</body>

</html>

Login view

@{

ViewData["Title"] = "login";

Layout = null;

}

<head>

<**link** rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />

<style>

.form-container {

max-width: 400px;

margin: 0 auto;

padding: 20px;

border: 1px solid #ccc;

border-radius: 5px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

body {

background-image: url("../../wwwroot/images2/7up.jpg");

background-size: cover;

background-repeat: no-repeat;

background-attachment: fixed;

}

</style>

</head>

<div class="form-container mt-5">

<h2>login</h2>

<**form** **asp-action**="Login">

<div class="form-group">

<label class="control-label"> user Name</label>

<input name="na" class="form-control" />

</div>

<div class="form-group">

<label for="pa" class="control-label"> user Password</label>

<input name="pa" class="form-control" /> <br />

</div>

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

<p style="color:red"> @ViewData["message"]</p>

<label>Not registered?</label>

<**a** **asp-action**="CreateCustomer">create account</**a**>

</**form**>

</div>

Regi view

@model project.Models.Userall

@{

ViewData["Title"] = "registration";

Layout = "~/Views/Shared/\_LayoutCustomer.cshtml";

}

<h1>registration</h1>

<h4>Userall</h4>

<hr />

<div class="row">

<div class="col-md-4">

<**form** **asp-action**="registration">

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

<div class="form-group">

<**label** **asp-for**="name" class="control-label"></**label**>

<**input** **asp-for**="name" class="form-control" />

<**span** **asp-validation-for**="name" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="password" class="control-label"></**label**>

<**input** **asp-for**="password" class="form-control" />

<**span** **asp-validation-for**="password" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="role" class="control-label"></**label**>

<**input** **asp-for**="role" class="form-control" />

<**span** **asp-validation-for**="role" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="RegistDate" class="control-label"></**label**>

<**input** **asp-for**="RegistDate" class="form-control" />

<**span** **asp-validation-for**="RegistDate" class="text-danger"></**span**>

</div>

<div class="form-group">

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

</div>

</**form**>

</div>

</div>

<div>

<**a** **asp-action**="Index">Back to List</**a**>

</div>