***<https://github.com/23rashmi/Phase4Project1.git>***

***WebAppTestSteps.cs======***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using TechTalk.SpecFlow;

using OpenQA.Selenium.Chrome;

using OpenQA.Selenium;

using NUnit.Framework;

namespace BankDetailsProject1.StepDefinitions

{

[Binding]

public class WebAppTestSteps

{

private readonly LoginService loginService;

private string? username;

private string? password;

private string? loginResult;

public WebAppTestSteps(LoginService loginService)

{

this.loginService = loginService;

}

[Given(@"a user with valid username ""([^""]\*)"" and valid password ""([^""]\*)""")]

public void GivenAUserWithValidUsernameAndValidPassword(string validUsername, string validPassword)

{

username = validUsername;

password = validPassword;

}

[When(@"the Login method is called")]

public void WhenTheLoginMethodIsCalled()

{

loginResult=loginService.Login(username, password);

}

[Then(@"the result should be Login Success")]

public void ThenTheResultShouldBeLoginSuccess()

{

Assert.AreEqual("Login Success", loginResult);

}

[Then(@"the result should be Login Failed")]

public void ThenTheResultShouldBeLoginFailed()

{

Assert.AreEqual("Login Failed", loginResult);

}

}

}

***WebAppTest.Feature====***

Feature: WebAppTest

@mytag

Scenario: Successful User Login

Given a user with valid username "rashmi123" and valid password "rashmi@123"

When the Login method is called

Then the result should be Login Success

@mytag

Scenario: Unsuccessful User Login

Given a user with valid username "invalid\_user" and valid password "invalid"

When the Login method is called

Then the result should be Login Failed

***LoginService.cs======***

namespace BankDetailsProject1

{

public class LoginService

{

public string Login(string username, string password)

{

if (IsValidUser(username, password))

{

return "Login Success";

}

else

{

return "Login Failed";

}

}

private bool IsValidUser(string username, string password)

{

return username == "rashmi123" && password == "rashmi@123";

}

}

}

***AccountController.cs========***

using Microsoft.AspNetCore.Identity;

using Microsoft.AspNetCore.Mvc;

using Project1Phase4.Models;

using System.ComponentModel.DataAnnotations;

namespace Project1Phase4.Controllers

{

public class AccountController : Controller

{

[HttpGet]

public IActionResult Login()

{

return View();

}

[HttpPost]

public IActionResult Login(Models.User user)

{

if(ModelState.IsValid)

{

if (user.Username == "rashmi" && user.Password == "rashmi@123")

{

TempData["Email"] = GetEmailByUsername(user.Username);

return RedirectToAction("Dashboard", "Account");

}

else if (user == null)

{

return RedirectToAction("Login");

}

else

{

ModelState.AddModelError(string.Empty, "Invalid username or password");

}

}

return View(user);

}

public IActionResult Dashboard()

{

string? email = TempData["Email"] as string;

ViewData["Email"] = email;

return View();

}

public string GetEmailByUsername(string username)

{

return "user@example.com";

}

public IActionResult Logout()

{

return View();

}

[HttpGet]

public IActionResult UpdateEmail()

{

return View();

}

[HttpPost]

public IActionResult UpdateEmail(string newEmail)

{

if (IsValidEmail(newEmail))

{

ViewBag.SuccessMessage = "Email updated successfully!";

}

else

{

ViewBag.ErrorMessage = "Invalid email format. Please enter a valid email address.";

}

return RedirectToAction("Dashboard", "Account");

}

private bool IsValidEmail(string email)

{

return new EmailAddressAttribute().IsValid(email);

}

[HttpGet]

public IActionResult ChangePassword()

{

return View();

}

[HttpPost]

public IActionResult ChangePassword(string currentPassword, string newPassword, string confirmPassword)

{

if (IsValidPasswordChange(currentPassword, newPassword, confirmPassword))

{

ViewBag.SuccessMessage = "Password changed successfully!";

}

else

{

ViewBag.ErrorMessage = "Invalid password change. Please check your inputs and try again.";

}

return RedirectToAction("Dashboard", "Account");

}

private bool IsValidPasswordChange(string currentPassword, string newPassword, string confirmPassword)

{

return !string.IsNullOrEmpty(currentPassword) &&

!string.IsNullOrEmpty(newPassword) &&

newPassword == confirmPassword;

}

public IActionResult Faq()

{

return View();

}

public IActionResult Documentation()

{

return View();

}

}

}

***User.cs=====***

using System.ComponentModel.DataAnnotations;

namespace Project1Phase4.Models

{

public class User

{

[Required(ErrorMessage ="Username is required")]

public string? Username { get; set; }

[Required(ErrorMessage ="Password is required")]

[DataType(DataType.Password)]

public string? Password { get; set; }

[Required(ErrorMessage = "Email is required")]

public string? Email { get; set; }

}

}

***ChangePassword.cshtml=====***

@model User

@{

ViewData["Title"] = "ChangePassword";

}

<h1>ChangePassword</h1>

<h2>Change Password</h2>

@if (ViewBag.SuccessMessage != null)

{

<div class="alert alert-success" role="alert">

@ViewBag.SuccessMessage

</div>

}

@if (ViewBag.ErrorMessage != null)

{

<div class="alert alert-danger" role="alert">

@ViewBag.ErrorMessage

</div>

}

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

<div class="form-group">

<label for="currentPassword">Current Password:</label>

<input type="password" class="form-control" id="currentPassword" name="currentPassword" required>

</div>

<div class="form-group">

<label for="newPassword">New Password:</label>

<input type="password" class="form-control" id="newPassword" name="newPassword" required>

</div>

<div class="form-group">

<label for="confirmPassword">Confirm Password:</label>

<input type="password" class="form-control" id="confirmPassword" name="confirmPassword" required>

</div>

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

</**form**>

***Dashboard.cshtml=====***

@using System.Security.Claims

<h1>Dashboard</h1>

@model User

@\* <h2>Welcome, @ViewData["Username"]</h2>

<h2>Welcome, @ViewData["Email"]</h2> \*@

<!DOCTYPE html>

<html>

<head>

@\* <title>Dashboard</title> \*@

<!-- Include any necesaary style or scripts. -->

<style>

/\* Modal styles \*/

.modal{

display: none;

position: fixed;

z-index: 1;

left: 0;

top: 0;

width: 100%;

height: 100%;

overflow: auto;

background-color: rgb(0,0,0);

background-color: rgba(0,0,0,0.4);

padding-top: 60px;

}

.modal-content {

background-color: #fefefe;

margin: 5% auto;

padding: 20px;

border: 1px solid #888;

width: 80%;

}

.close {

color: #aaa;

float: right;

font-size: 28px;

font-weight: bold;

}

.close:hover,

.close:focus {

color: black;

text-decoration: none;

cursor: pointer;

}

body{

background-color:#f0f0f0;

color:#333;

}

.dashboard-container{

max-width:800px;

margin:0 auto;

padding:20px;

border:1px solid #ccc;

background-color:#fff;

}

.task-list{

list-style-type:none;

padding:0;

}

.task-item{

margin-bottom:10px;

}

.completed-task{

text-decoration:line-through;

color:#2158;

}

</style>

</head>

<body>

<h2>Dashboard</h2>

<!-- View Account Information section -->

<div>

<h3>Account Information</h3>

<p>Username: @(User.Identity?.Name)</p>

<p>Email: @(User.FindFirst(ClaimTypes.Email)?.Value)</p>

<p>Account Number : 1234567890</p>

<p>Account Balance: $1500.00</p>

</div>

<!-- Transaction History section -->

<div>

<h3>Transaction History</h3>

<!--Display a list of recent transactions-->

<ul>

<li>

<span>Date: 2024-01-21</span>

<span>Amount: $100.00</span>

<span>Description: Purchase at ABC store</span>

</li>

<li>

<span>Date: 2023-05-25</span>

<span>Amount: $500.00</span>

<span>Description: Online Payment</span>

</li>

</ul>

</div>

<!-- Financial Overview section -->

<div>

<h3>Financial Overview</h3>

<!-- Display a summary of financial data -->

<div>

<p>Account Balance: $1,500.00</p>

<p>Available Funds: $1,200.00</p>

<p>Investment Portfolio Value: $50,000.00</p>

<p>Credit Card Debt: -$500.00</p>

</div>

</div>

<!-- Personalization Settings section -->

<div>

<h3>Personalization Settings</h3>

<!-- Form for personalization settings -->

<form id="personalizationForm">

<label for=" backgroundColor">Background Color: </label>

<input type="color" id="backgroundColor" name="backgroundColor" />

<label for="textColor">Text Color : </label>

<input type="color" id="textColor" name="textColor" />

<button type="button" onclick="applyPersonalization()">Apply Personalization</button>

</form>

</div>

<!-- Support and Help section -->

<div>

<h3>Support and Help</h3>

<!-- Links/buttons for accessing help resources -->

<ul>

<li><a href="/account/faq">FAQs</a></li>

<li><a href="/account/documentation"> Documentation </a></li>

<li><button onclick="contactSupport()">Contact Support</button></li>

</ul>

</div>

<!-- Modal for contacting support -->

<div id="supportModal" class="modal">

<div class="modal-content">

<span class="close" onclick="closeSupportModal()">&times;</span>

<h2>Contact Support</h2>

<!-- Include a form or chat interface for contacting support -->

<p>This is a placeholder for your support form or chat interface.</p>

</div>

</div>

<!-- Integration with Other Services section -->

<div>

<h3>Integration with Other Services</h3>

<!-- Third-party widgets or services -->

<div>

<!--Example: News Feed-->

<iframe src="https://www.example.com/news-feed" width="700" height="300" frameborder="0"></iframe>

<!-- Example: Weather Widget -->

<script src="https://www.example.com/weather-widget.js" defer></script>

</div>

</div>

@\* <!-- Logout button -->

<div>

<form asp-controller="Account" asp-action="Logout" method="post">

<button type="submit">Logout</button>

</form>

</div> \*@

<!-- Task Management section -->

<div>

<h3>Task Management</h3>

<!--Task input from-->

<form onsubmit="addTask(); return false;">

<label for="taskInput">New Task ; </label>

<input type="text" id="taskInput" required />

<button type="submit">Add Task</button>

</form>

<!--Task List-->

<ul class="task-list" id="taskList">

<!-- Task will be added dynamically here.... -->

</ul>

</div>

<!-- Account Settings section -->

<div>

<h3>Account Settings</h3>

<!-- Change Password form -->

<form asp-controller="Account" asp-action="ChangePassword" method="post">

<div>

<label for="CurrentPassword">Current Password: </label>

<input type="password" name="CurrentPassword" id="CurrentPassword" required />

</div>

<div>

<label for="NewPassword">New Password: </label>

<input type="password" name="NewPassword" id="NewPassword" required />

</div>

<div>

<button type="submit">Change Password</button>

</div>

</form>

<!-- Update Email form -->

<form asp-controller="Account" asp-action="UpdateEmail" method="post">

<div>

<label for=" NewEmail">New Email: </label>

<input type="email" name="NewEmail" id="NewEmail" required />

</div>

<div>

<button type="submit">Update Email</button>

</div>

</form>

</div>

<!-- Account-related activities buttons -->

<!-- Logout button -->

<form asp-controller="Account" asp-action="Logout" method="post">

<button type="submit">Logout</button>

</form>

<script>

function contactSupport() {

// Open the support modal

var modal = document.getElementById("supportModal");

modal.style.display = "block";

}

function closeSupportModal() {

// Close the support modal

var modal = document.getElementById("supportModal");

modal.style.display = "none";

}

// Close the modal if the user clicks outside of it

window.onclick = function (event) {

var modal = document.getElementById("supportModal");

if (event.target == modal) {

modal.style.display = "none";

}

}

function performActivity(activity) {

alert("Performing: " + activity);

}

//Task management functions

function addTask() {

var taskInput = document.getElementById("taskInput");

var taskList = document.getElementById("taskList");

if (taskInput.value.trim() !== "") {

var taskItem = document.createElement("li");

taskItem.className = "task-item";

taskItem.innerHTML = '<input type="checkbox" onchange="completeTask(this)"> ' + '<span>' + taskInput.value + '</span>' + '<button onclick="deleteTask(this)">Delete</button>';

taskList.appendChild(taskItem);

taskInput.value = "";

}

}

function completeTask(checkbox) {

var taskText = chekbox.nextElementSibling;

if (checkbox.checked) {

taskText.classList.add("completed-task");

}

else {

taskText.classList.remove("completed-task");

}

}

function deleteTask(button) {

var taskItem = button.parentNode;

taskItem.parentNode.removeChild(taskItem);

}

function performActivity1() {

alert("Perform Account Activity 1");

}

function performActivity2() {

alert("Perform Account Activity 2");

}

function applyPersonalization() {

var backgroundColor = document.getElementById("backgroundColor").value;

var textColor = document.getElementById("textColor").value;

document.body.style.backgroundColor = backgroundColor;

document.body.style.color = textColor;

}

function contactSupport() {

alert("Contacting Support");

}

</script>

</body>

</html>

***Documentation.cshtml=====***

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>Documentation</title>

</head>

<body>

<h2>Documentation</h2>

<section>

<h3>Getting Started</h3>

<p>Welcome to our application! To get started, follow these steps:</p>

<ol>

<li>Create an account by clicking on the "Sign Up" button.</li>

<li>Log in with your credentials.</li>

</ol>

</section>

<section>

<h3>Features</h3>

<h4>1. Account Management</h4>

<p>This section covers account-related features such as updating email, changing passwords, and viewing account information.</p>

<ul>

<li><a href="/Account/UpdateEmail">Update Email</a></li>

<li><a href="/Account/ChangePassword">Change Password</a></li>

</ul>

<h4>2. FAQ</h4>

<p>Visit our <a href="/Account/Faq">FAQ page</a> for answers to common questions.</p>

</section>

</body>

</html>

***Faq.cshtml=====***

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>Faq</title>

</head>

<body>

<h2>Frequently Asked Questions</h2>

<dl>

<dt>Q: How do I create an account?</dt>

<dd>A: To create an account, click on the "Sign Up" button on the homepage and follow the registration process.</dd>

<dt>Q: Can I reset my password?</dt>

<dd>A: Yes, you can reset your password by clicking on the "Forgot Password" link on the login page and following the instructions.</dd>

<dt>Q: How can I contact customer support?</dt>

<dd>A: You can contact our customer support team by emailing support@example.com or using the contact form on the "Contact Us" page.</dd>

</dl>

<a href="/Account/Dashboard">Go back to Dashboard</a>

</body>

</html>

***Login.cshtml======***

@model Project1Phase4.Models.User

<!DOCTYPE html>

<html>

<head>

<title>Login Pge</title>

</head>

<body>

<h2>Login</h2>

<**form** **asp-controller**="Account" **asp-action**="Login" method="post" >

<div>

<**label** **asp-for**="Username">Username: </**label**>

<**input** **asp-for**="Username"/>

<**span** **asp-validation-for**="Username"></**span**>

</div>

<div>

<**label** **asp-for**="Password">Password: </**label**>

<**input** **asp-for**="Password" **type**="password"/>

<**span** **asp-validation-for**="Password"></**span**>

</div>

<div>

<**label** **asp-for**="Email">Email: </**label**>

<**input** **asp-for**="Email" **type**="email" />

<**span** **asp-validation-for**="Email"></**span**>

</div>

<button type="submit">Login</button>

</**form**>

</body>

</html>

***Logout.cshtml=====***

@{

ViewData["Title"] = "Logout";

}

<h1>Logout</h1>

<p>Thanks for using the software.....</p>

***UpdateEmail.cshtml=====***

@model User

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>UpdateEmail</title>

</head>

<body>

<h2>Update Email</h2>

@if(ViewBag.SuccessMessage != null)

{

<div class="alert alert-success" role="alert">

@ViewBag.SuccessMessage

</div>

}

@if (ViewBag.ErrorMessage != null)

{

<div class="alert alert-danger" role="alert">

@ViewBag.ErrorMessage

</div>

}

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

<div class="form-group">

<label for="newEmail">New Email:</label>

<input type="email" class="form-control" id="newEmail" name="newEmail" required>

</div>

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

</**form**>

</body>

</html>

***Program.cs======***

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllersWithViews();

var app = builder.Build();

if (!app.Environment.IsDevelopment())

{

app.UseExceptionHandler("/Home/Error");

}

app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.MapControllerRoute(

name: "default",

pattern: "{controller=Account}/{action=Login}/{id?}");

app.MapControllerRoute(

name: "default",

pattern: "{controller=Account}/{action=Logout}");

app.MapControllerRoute(

name: "default",

pattern: "{controller=Account}/{action=UpdateEmail}");

app.MapControllerRoute(

name: "default",

pattern: "{controller=Account}/{action=ChangePassword}");

app.MapControllerRoute(

name: "default",

pattern: "{controller=Account}/{action=Faq}");

app.MapControllerRoute(

name: "default",

pattern: "{controller=Account}/{action=Documentation}");

app.Run();

***Layout.cshtml***

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"] - Project1Phase4</title>

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

<**link** rel="stylesheet" **href**="~/css/site.css" **asp-append-version**="true" />

<**link** rel="stylesheet" **href**="~/Project1Phase4.styles.css" **asp-append-version**="true" />

</head>

<body>

<header>

<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">

<div class="container-fluid">

<**a** class="navbar-brand" **asp-area**="" **asp-controller**="Home" **asp-action**="Index">Project1Phase4</**a**>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse" aria-controls="navbarSupportedContent"

aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">

<ul class="navbar-nav flex-grow-1">

<li class="nav-item">

<**a** class="nav-link text-dark" **asp-area**="" **asp-controller**="Account" **asp-action**="Login">Login</**a**>

</li>

<li class="nav-item">

<**a** class="nav-link text-dark" **asp-area**="" **asp-controller**="Account" **asp-action**="Dashboard">Dashboard</**a**>

</li>

<li class="nav-item">

<**a** class="nav-link text-dark" **asp-area**="" **asp-controller**="Account" **asp-action**="UpdateEmail">UpdateEmail</**a**>

</li>

<li class="nav-item">

<**a** class="nav-link text-dark" **asp-area**="" **asp-controller**="Account" **asp-action**="ChangePassword">ChangePassword</**a**>

</li>

</ul>

</div>

</div>

</nav>

</header>

<div class="container">

<main role="main" class="pb-3">

@RenderBody()

</main>

</div>

<footer class="border-top footer text-muted">

<div class="container">

&copy; 2024 - Project1Phase4 - <**a** **asp-area**="" **asp-controller**="Home" **asp-action**="Privacy">Privacy</**a**>

</div>

</footer>

<**script** src="~/lib/jquery/dist/jquery.min.js"></**script**>

<**script** src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></**script**>

<**script** **src**="~/js/site.js" **asp-append-version**="true"></**script**>

@await RenderSectionAsync("Scripts", required: false)

</body>

</html>