**LAB # 12**

**Task # 01: Two factor authentication for login.**

**Solution:**

**Home Controller:**

namespace TwoFactorAuthentication.Controllers

{

public class HomeController : Controller

{

public ActionResult Index()

{

if (Session["Username"] == null || Session["IsValidTwoFactorAuthentication"] == null || !(bool)Session["IsValidTwoFactorAuthentication"])

{

return RedirectToAction("Login");

}

return View();

}

public ActionResult About()

{

if (Session["Username"] == null || Session["IsValidTwoFactorAuthentication"] == null || !(bool)Session["IsValidTwoFactorAuthentication"])

{

return RedirectToAction("Login");

}

ViewBag.Message = "Your application description page.";

return View();

}

public ActionResult Contact()

{

if (Session["Username"] == null || Session["IsValidTwoFactorAuthentication"] == null || !(bool)Session["IsValidTwoFactorAuthentication"])

{

return RedirectToAction("Login");

}

ViewBag.Message = "Your contact page.";

return View();

}

public ActionResult Login()

{

Session["UserName"] = null;

Session["IsValidTwoFactorAuthentication"] = null;

return View();

}

[HttpPost]

public ActionResult Login(LoginModel login)

{

bool status = false;

if (Session["Username"] == null || Session["IsValidTwoFactorAuthentication"] == null || !(bool)Session["IsValidTwoFactorAuthentication"])

{

string googleAuthKey = WebConfigurationManager.AppSettings["GoogleAuthKey"];

string UserUniqueKey = (login.UserName + googleAuthKey);

//Take UserName And Password As Static - Admin As User And 12345 As Password

if (login.UserName == "amjad5azx" && login.Password == "12345")

{

Session["UserName"] = login.UserName;

//Two Factor Authentication Setup

TwoFactorAuthenticator TwoFacAuth = new TwoFactorAuthenticator();

var setupInfo = TwoFacAuth.GenerateSetupCode("UdayDodiyaAuthDemo.com", login.UserName, ConvertSecretToBytes(UserUniqueKey, false), 300);

Session["UserUniqueKey"] = UserUniqueKey;

ViewBag.BarcodeImageUrl = setupInfo.QrCodeSetupImageUrl;

ViewBag.SetupCode = setupInfo.ManualEntryKey;

status = true;

}

}

else

{

return RedirectToAction("Index");

}

ViewBag.Status = status;

return View();

}

private static byte[] ConvertSecretToBytes(string secret, bool secretIsBase32) =>

secretIsBase32 ? Base32Encoding.ToBytes(secret) : Encoding.UTF8.GetBytes(secret);

public ActionResult TwoFactorAuthenticate()

{

var token = Request["CodeDigit"];

TwoFactorAuthenticator TwoFacAuth = new TwoFactorAuthenticator();

string UserUniqueKey = Session["UserUniqueKey"].ToString();

bool isValid = TwoFacAuth.ValidateTwoFactorPIN(UserUniqueKey, token, false);

if (isValid)

{

HttpCookie TwoFCookie = new HttpCookie("TwoFCookie");

string UserCode = Convert.ToBase64String(MachineKey.Protect(Encoding.UTF8.GetBytes(UserUniqueKey)));

Session["IsValidTwoFactorAuthentication"] = true;

return RedirectToAction("Index");

}

ViewBag.Message = "Google Two Factor PIN is expired or wrong";

return RedirectToAction("Login");

}

public ActionResult Logoff()

{

Session["UserName"] = null;

Session["IsValidTwoFactorAuthentication"] = null;

return RedirectToAction("Login");

}

}

}

**Login View:**

@model TwoFactorAuthentication.Models.LoginModel

@{

ViewBag.Title = "Login";

}

<center>

<h2>Login Page</h2>

@if (ViewBag.Status == null || !ViewBag.Status)

{

<div>@ViewBag.Message</div>

<div>

@using (Html.BeginForm())

{

<div class="form-group">

<label for="UserName">UserName : </label>

@Html.TextBoxFor(a => a.UserName, new { @class = "form-control" })

</div>

<div class="form-group">

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

@Html.TextBoxFor(a => a.Password, new { @class = "form-control", type = "password" })

</div>

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

}

</div>

}

else

{

<div>@ViewBag.Message</div>

<div>

<img src="@ViewBag.BarcodeImageUrl" width="300" height="300" />

</div>

<div>

Manual Setup Code : @ViewBag.SetupCode

</div>

<div>

@using (Html.BeginForm("TwoFactorAuthenticate", "Home", FormMethod.Post))

{

<input type="text" name="CodeDigit" />

<input type="submit" class="btn btn-success" />

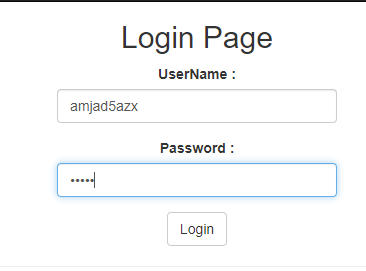
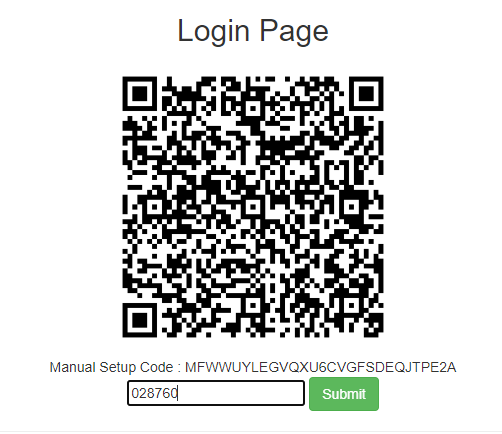
}

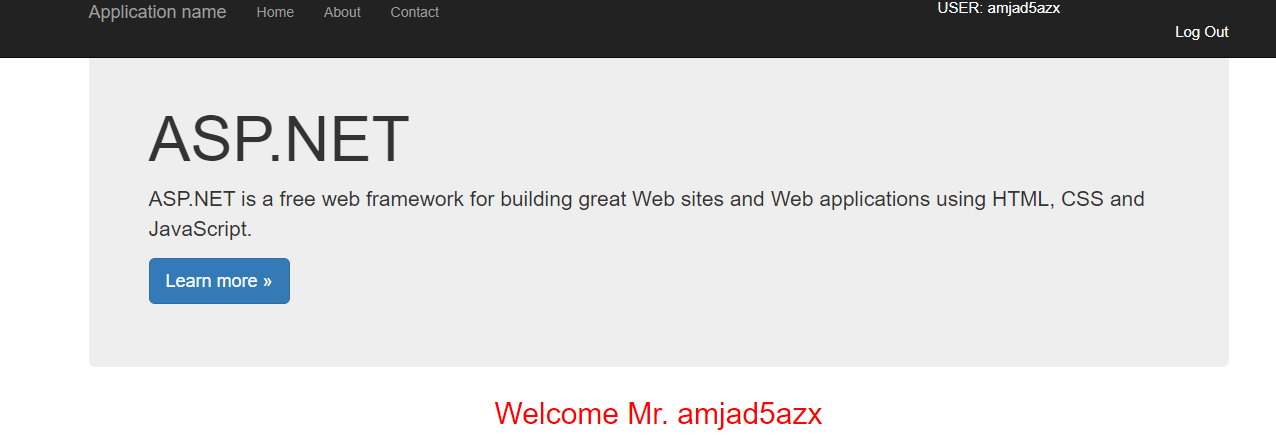
</div>

}

</center>

**Output:**



**Task # 02: Two factor authentication for sign up.**

**Solution:**

**Home Controller:**

public class HomeController : Controller

{

public ActionResult Index()

{

if (Session["Username"] == null || Session["IsValidTwoFactorAuthentication"] == null || !(bool)Session["IsValidTwoFactorAuthentication"])

{

return RedirectToAction("SignUp");

}

return View();

}

public ActionResult About()

{

if (Session["Username"] == null || Session["IsValidTwoFactorAuthentication"] == null || !(bool)Session["IsValidTwoFactorAuthentication"])

{

return RedirectToAction("SignUp");

}

ViewBag.Message = "Your application description page.";

return View();

}

public ActionResult Contact()

{

if (Session["Username"] == null || Session["IsValidTwoFactorAuthentication"] == null || !(bool)Session["IsValidTwoFactorAuthentication"])

{

return RedirectToAction("SignUp");

}

ViewBag.Message = "Your contact page.";

return View();

}

public ActionResult SignUp()

{

Session["UserName"] = null;

Session["IsValidTwoFactorAuthentication"] = null;

return View();

}

[HttpPost]

public ActionResult SignUp(User user)

{

bool status = false;

if (Session["Username"] == null || Session["IsValidTwoFactorAuthentication"] == null || !(bool)Session["IsValidTwoFactorAuthentication"])

{

string googleAuthKey = WebConfigurationManager.AppSettings["GoogleAuthKey"];

string UserUniqueKey = (user.Username + googleAuthKey);

Session["UserName"] = user.Username;

//Two Factor Authentication Setup

TwoFactorAuthenticator TwoFacAuth = new TwoFactorAuthenticator();

var setupInfo = TwoFacAuth.GenerateSetupCode("UdayDodiyaAuthDemo.com", user.Username, ConvertSecretToBytes(UserUniqueKey, false), 300);

Session["UserUniqueKey"] = UserUniqueKey;

ViewBag.BarcodeImageUrl = setupInfo.QrCodeSetupImageUrl;

ViewBag.SetupCode = setupInfo.ManualEntryKey;

status = true;

}

else

{

return RedirectToAction("Index");

}

ViewBag.Status = status;

return View();

}

private static byte[] ConvertSecretToBytes(string secret, bool secretIsBase32) =>

secretIsBase32 ? Base32Encoding.ToBytes(secret) : Encoding.UTF8.GetBytes(secret);

public ActionResult TwoFactorAuthenticate()

{

var token = Request["CodeDigit"];

TwoFactorAuthenticator TwoFacAuth = new TwoFactorAuthenticator();

string UserUniqueKey = Session["UserUniqueKey"].ToString();

bool isValid = TwoFacAuth.ValidateTwoFactorPIN(UserUniqueKey, token, false);

if (isValid)

{

HttpCookie TwoFCookie = new HttpCookie("TwoFCookie");

string UserCode = Convert.ToBase64String(MachineKey.Protect(Encoding.UTF8.GetBytes(UserUniqueKey)));

Session["IsValidTwoFactorAuthentication"] = true;

return RedirectToAction("Index");

}

ViewBag.Message = "Google Two Factor PIN is expired or wrong";

return RedirectToAction("SignUp");

}

public ActionResult Logoff()

{

Session["UserName"] = null;

Session["IsValidTwoFactorAuthentication"] = null;

return RedirectToAction("SignUp");

}

}

**SignUp View:**

@model TwoFactorAuthentication.Models.User

@{

ViewBag.Title = "Sign Up";

}

<center>

<h2>SignUp Page</h2>

@if (ViewBag.Status == null || !ViewBag.Status)

{

<div>@ViewBag.Message</div>

<div>

@using (Html.BeginForm())

{

<div class="form-group">

<label for="FirstName">FirstName : </label>

@Html.TextBoxFor(a => a.Firstname, new { @class = "form-control" })

</div>

<div class="form-group">

<label for="UserName">LastName : </label>

@Html.TextBoxFor(a => a.Lastname, new { @class = "form-control" })

</div>

<div class="form-group">

<label for="UserName">UserName : </label>

@Html.TextBoxFor(a => a.Username, new { @class = "form-control" })

</div>

<div class="form-group">

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

@Html.TextBoxFor(a => a.Password, new { @class = "form-control", type = "password" })

</div>

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

}

</div>

}

else

{

<div>@ViewBag.Message</div>

<div>

<img src="@ViewBag.BarcodeImageUrl" width="300" height="300" />

</div>

<div>

Manual Setup Code : @ViewBag.SetupCode

</div>

<div>

@using (Html.BeginForm("TwoFactorAuthenticate", "Home", FormMethod.Post))

{

<input type="text" name="CodeDigit" />

<input type="submit" class="btn btn-success" />

}

</div>

}

</center>

**Output:**

