ASP.NET - SECURITY

http://www.tutorialspoint.com/asp.net/asp.net_security.htm

Implementing security in a site has the following aspects:

- **Authentication** : It is the process of ensuring the user's identity and authenticity. ASP.NET allows four types of authentications:
    - Windows Authentication
    - Forms Authentication
    - Passport Authentication
    - Custom Authentication
- **Authorization** : It is the process of defining and allotting specific roles to specific users.
- **Confidentiality** : It involves encrypting the channel between the client browser and the web server.
- **Integrity** : It involves maintaining the integrity of data. For example, implementing digital signature.

Forms-Based Authentication

Traditionally, forms-based authentication involves editing the web.config file and adding a login page with appropriate authentication code.

The web.config file could be edited and the following codes written on it:

```
<configuration>

<system.web>
   <authentication mode="Forms">
      <forms loginUrl ="login.aspx"/>
   </authentication>

   <authorization>
      <deny users="?"/>
   </authorization>
</system.web>
...
...
</configuration>
```

The login.aspx page mentioned in the above code snippet could have the following code behind file with the usernames and passwords for authentication hard coded into it.

```
protected bool authenticate(String uname, String pass)
{
   if(uname == "Tom")
   {
      if(pass == "tom123")
         return true;
   }

   if(uname == "Dick")
   {
      if(pass == "dick123")
         return true;
   }

   if(uname == "Harry")
   {
      if(pass == "har123")
```

```
         return true;
   }

   return false;
}

public void OnLogin(Object src, EventArgs e)
{
   if (authenticate(txtuser.Text, txtpwd.Text))
   {
      FormsAuthentication.RedirectFromLoginPage(txtuser.Text, chkrem.Checked);
   }
   else
   {
      Response.Write("Invalid user name or password");
   }
}
```

Observe that the FormsAuthentication class is responsible for the process of authentication.

However, Visual Studio allows you to implement user creation, authentication, and authorization with seamless ease without writing any code, through the Web Site Administration tool. This tool allows creating users and roles.

Apart from this, ASP.NET comes with readymade login controls set, which has controls performing all the jobs for you.

Implementing Forms-Based Security

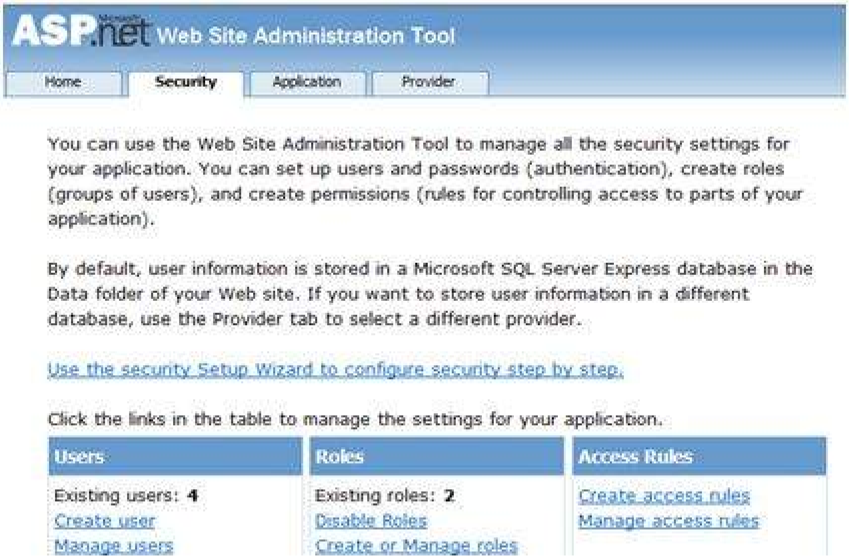
To set up forms-based authentication, you need the following:

- A database of users to support the authentication process
- A website that uses the database
- User accounts
- Roles
- Restriction of users and group activities
- A default page, to display the login status of the users and other information.
- A login page, to allow users to log in, retrieve password, or change password

To create users, take the following steps:

Step 1 : Choose Website -> ASP.NET Configuration to open the Web Application Administration Tool.

Step 2 : Click on the Security tab.



You can use the Web Site Administration Tool to manage all the security settings for your application. You can set up users and passwords (authentication), create roles (groups of users), and create permissions (rules for controlling access to parts of your application).

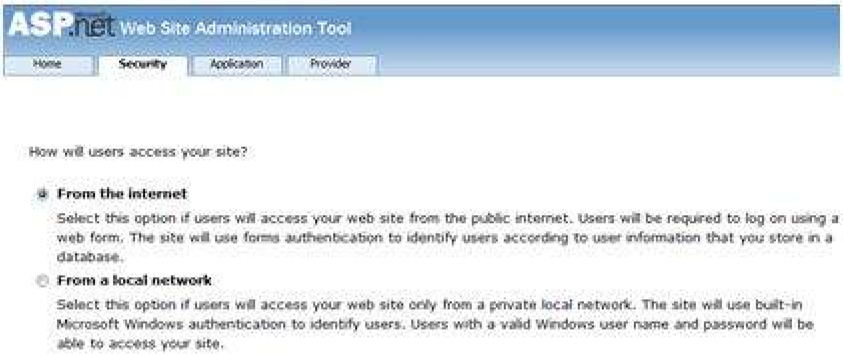
By default, user information is stored in a Microsoft SQL Server Express database in the Data folder of your Web site. If you want to store user information in a different database, use the Provider tab to select a different provider.

Use the security Setup Wizard to configure security step by step.

Click the links in the table to manage the settings for your application.

Users	Roles	Access Rules
Existing users: 4 Create user Manage users	Existing roles: 2 Disable Roles Create or Manage roles	Create access rules Manage access rules

Step 3 : Select the authentication type to 'Forms based authentication' by selecting the 'From the Internet' radio button.



How will users access your site?

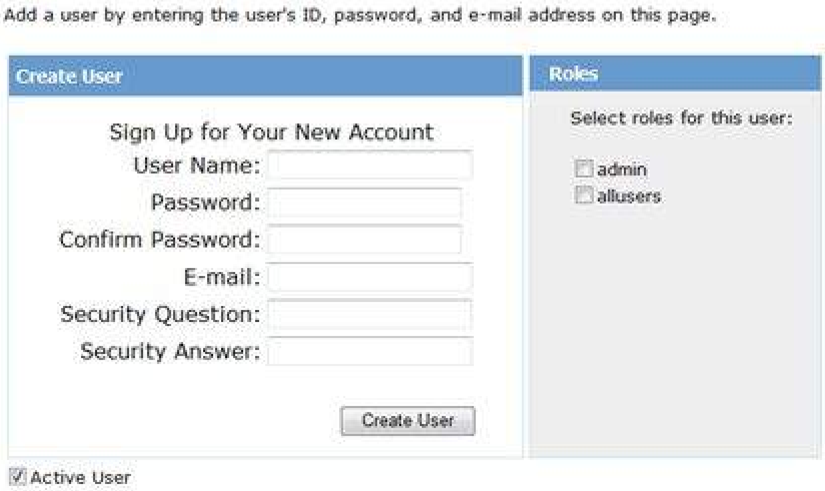
● **From the internet**

Select this option if users will access your web site from the public internet. Users will be required to log on using a web form. The site will use forms authentication to identify users according to user information that you store in a database.

○ **From a local network**

Select this option if users will access your web site only from a private local network. The site will use built-in Microsoft Windows authentication to identify users. Users with a valid Windows user name and password will be able to access your site.

Step 4 : Click on 'Create Users' link to create some users. If you already had created roles, you could assign roles to the user, right at this stage.



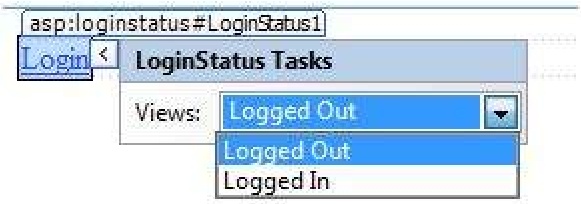
Add a user by entering the user's ID, password, and e-mail address on this page.

Step 5 : Create a web site and add the following pages:

- Welcome.aspx
- Login.aspx
- CreateAccount.aspx
- PasswordRecovery.aspx
- ChangePassword.aspx

Step 6 : Place a LoginStatus control on the Welcome.aspx from the login section of the toolbox. It has two templates: LoggedIn and LoggedOut.

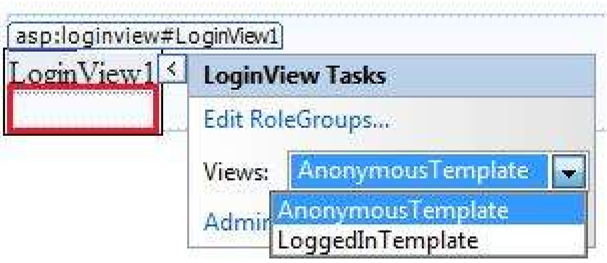
In LoggedOut template, there is a login link and in the LoggedIn template, there is a logout link on the control. You can change the login and logout text properties of the control from the Properties window.



Step 7 : Place a LoginView control from the toolbox below the LoginStatus control. Here, you can

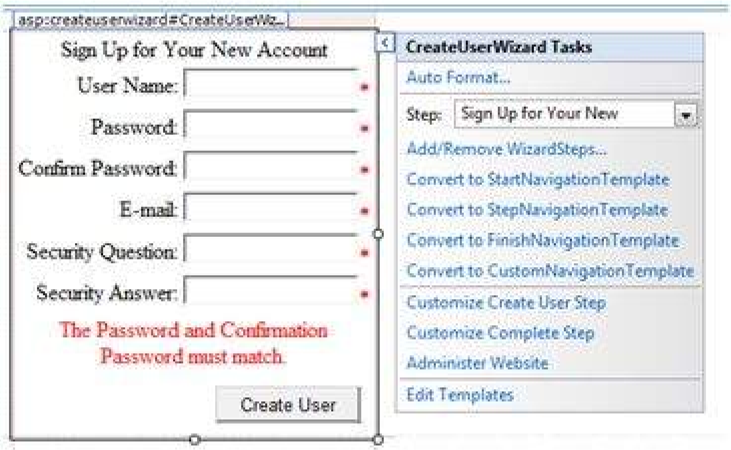
put texts and other controls *hyperlinks, buttonsetc.* , which are displayed based on whether the user is logged in or not.

This control has two view templates: Anonymous template and LoggedIn template. Select each view and write some text for the users to be displayed for each template. The text should be placed on the area marked red.



Step 8 : The users for the application are created by the developer. You might want to allow a visitor to create a user account. For this, add a link beneath the LoginView control, which should link to the CreateAccount.aspx page.

Step 9 : Place a CreateUserWizard control on the create account page. Set the ContinueDestinationPageUrl property of this control to Welcome.aspx.


asp:createuserwizard#CreateUserWiz...
Sign Up for Your New Account
User Name:
Password:
Confirm Password:
E-mail:
Security Question:
Security Answer:
The Password and Confirmation Password must match.
Create User
CreateUserWizard Tasks
Auto Format...
Step: Sign Up for Your New
Add/Remove WizardSteps...
Convert to StartNavigationTemplate
Convert to StepNavigationTemplate
Convert to FinishNavigationTemplate
Convert to CustomNavigationTemplate
Customize Create User Step
Customize Complete Step
Administer Website
Edit Templates

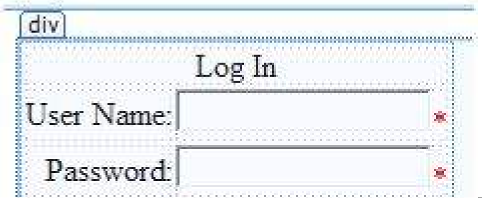
Step 10 : Create the Login page. Place a Login control on the page. The LoginStatus control automatically links to the Login.aspx. To change this default, make the following changes in the web.config file.

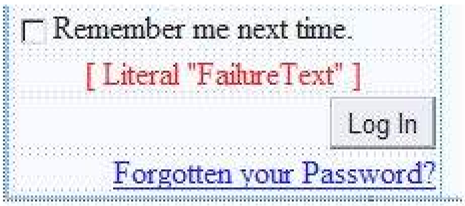
For example, if you want to name your log in page as signup.aspx, add the following lines to the <authentication> section of the web.config:

```
<configuration>
   <system.web>
      <authentication mode="Forms">
         <forms loginUrl ="signup.aspx" defaultUrl = â€œWelcome.aspxâ€ />
      </authentication>
   </system.web>
</configuration>
```

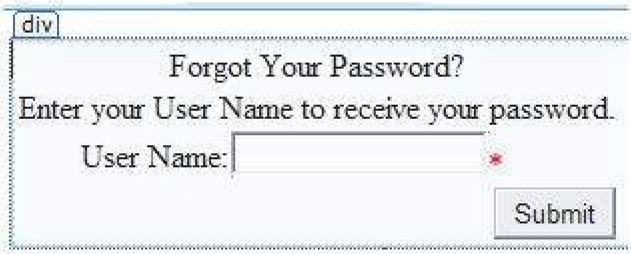
Step 11 : Users often forget passwords. The PasswordRecovery control helps the user gain access to the account. Select the Login control. Open its smart tag and click 'Convert to Template'.

Customize the UI of the control to place a hyperlink control under the login button, which should link to the PassWordRecovery.aspx.



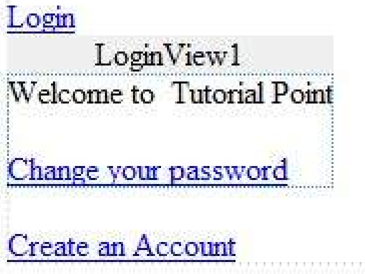


Step 12 : Place a PasswordRecovery control on the password recovery page. This control needs an email server to send the passwords to the users.

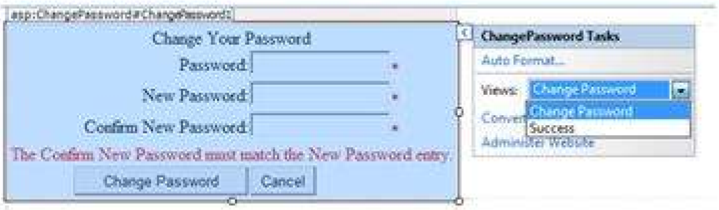

Forgot Your Password?

Enter your User Name to receive your password.

Step 13 : Create a link to the ChangePassword.aspx page in the LoggedIn template of the LoginView control in Welcome.aspx.



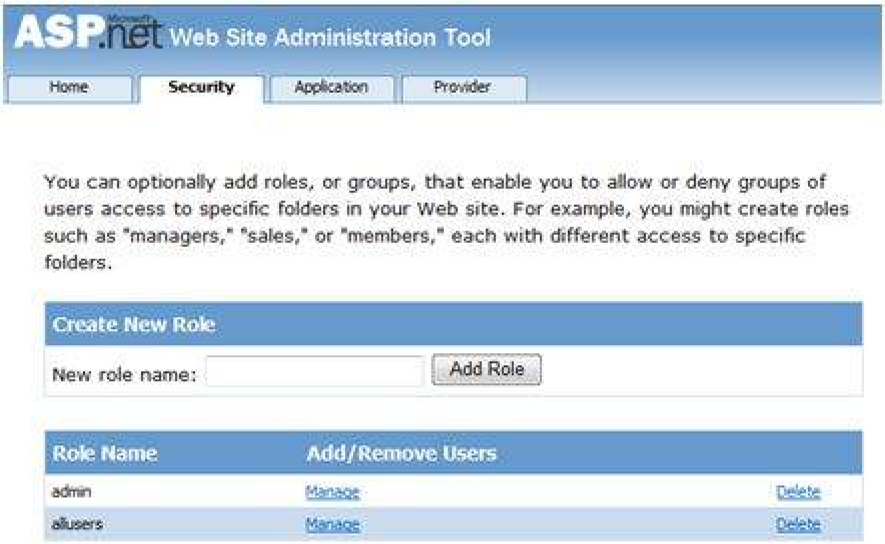
Step 14 : Place a ChangePassword control on the change password page. This control also has two views.


Change Your Password

The Confirm New Password must match the New Password entry.

Now run the application and observe different security operations.

To create roles, go back to the Web Application Administration Tools and click on the Security tab. Click on 'Create Roles' and create some roles for the application.



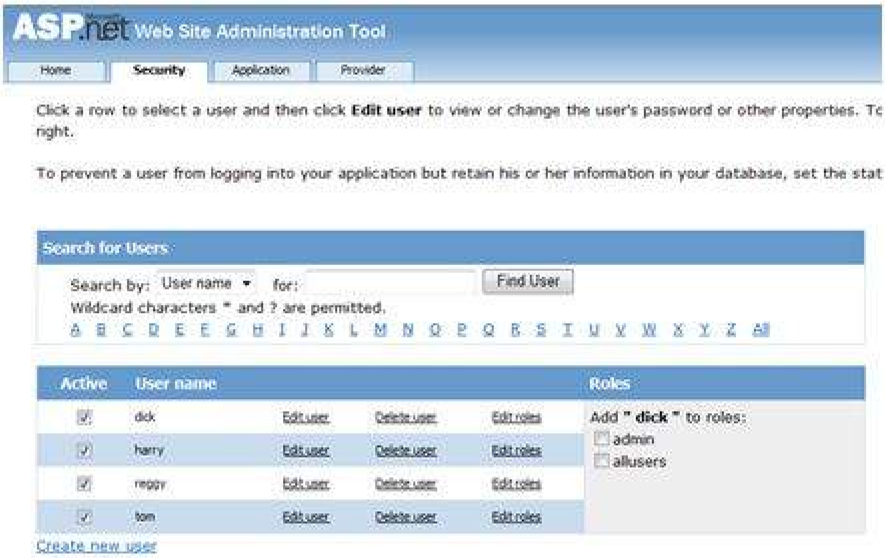
You can optionally add roles, or groups, that enable you to allow or deny groups of users access to specific folders in your Web site. For example, you might create roles such as "managers," "sales," or "members," each with different access to specific folders.

Create New Role

New role name: Add Role

Role Name	Add/Remove Users	
admin	Manage	Delete
allusers	Manage	Delete

Click on the 'Manage Users' link and assign roles to the users.



ASP.net Web Site Administration Tool
Home
Security
Application
Provider

Click a row to select a user and then click **Edit user** to view or change the user's password or other properties. To right.

To prevent a user from logging into your application but retain his or her information in your database, set the stat

Search for Users

Search by: User name ▾ for: Find User

Wildcard characters * and ? are permitted.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z All

Active	User name				Roles
☑	dick	Edit user	Delete user	Edit roles	Add " **dick** " to roles: □ admin □ allusers
☑	harry	Edit user	Delete user	Edit roles	
☑	reggy	Edit user	Delete user	Edit roles	
☑	tom	Edit user	Delete user	Edit roles	

IIS Authentication: SSL

The Secure Socket Layer or SSL is the protocol used to ensure a secure connection. With SSL enabled, the browser encrypts all data sent to the server and decrypts all data coming from the server. At the same time, the server encrypts and decrypts all data to and from browser.

The URL for a secure connection starts with HTTPS instead of HTTP. A small lock is displayed by a browser using a secure connection. When a browser makes an initial attempt to communicate with a server over a secure connection using SSL, the server authenticates itself by sending its digital certificate.

To use the SSL, you need to buy a digital secure certificate from a trusted Certification Authority CA and install it in the web server. Following are some of the trusted and reputed certification authorities:

- www.verisign.com
- www.geotrust.com
- www.thawte.com

SSL is built into all major browsers and servers. To enable SSL, you need to install the digital certificate. The strength of various digital certificates varies depending upon the length of the key generated during encryption. More the length, more secure is the certificate, hence the connection.

Strength	Description
40 bit	Supported by most browsers but easy to break.
56 bit	Stronger than 40-bit.
128 bit	Extremely difficult to break but all the browsers do not support it.