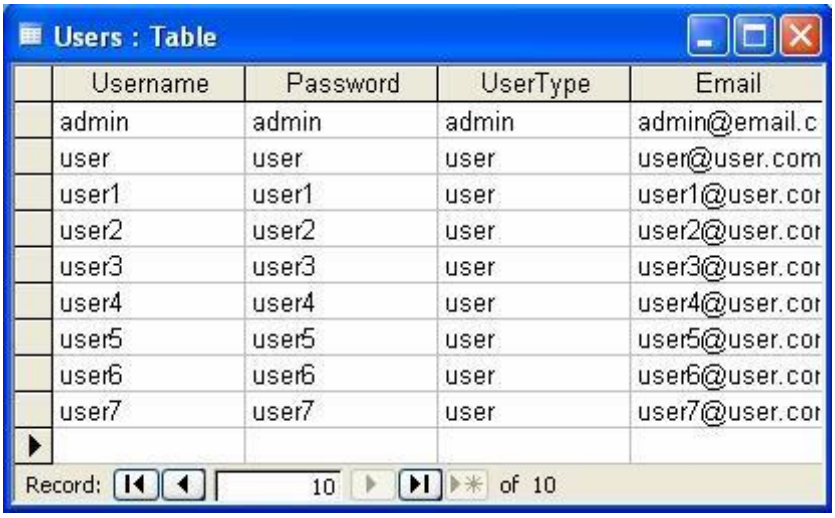
Lab 6 : Form Authentication

Authentication

- Authentication is the process of obtaining identification credentials such as name and password from a user and validating those credentials against some authority.
- If the credentials are valid, the entity that submitted the credentials is considered an authenticated identity.
- Once an identity has been authenticated, the authorization process determines whether that identity has access to a given resource.
- These tutorial consisting 3 parts:
    1. Creating a Login Page
    2. Creating a Members Only Page
    3. Creating a Admin Only Page
- Here is the table used to perform the user authentication.



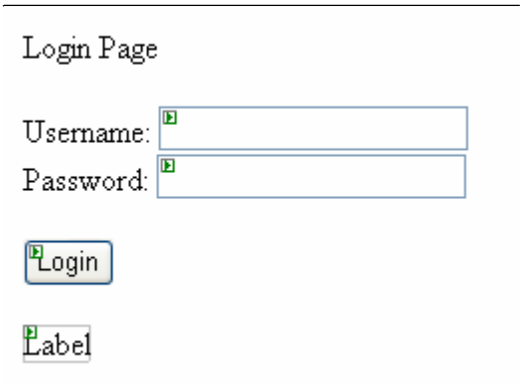
Creating a Login Page

- In this example, we are going to use the Users table from the Auth database.

Follow these steps:

1. Create a new Web Form and call the Login.aspx.
2. Switch to Design view, then open the Standard section of the toolbox and drag a 2 TextBox, 1 Label and 1 Button controls over to the page.
3. Open the Data section of the toolbox and drag a SqlDataSource control over to the page.
4. Click Configure Data Source in the Smart Tag menu. In the Configure Data Source Wizard, choose the database Auth.
5. Click **Next** in the Configure Data Source Wizard. The page that opens allows you to choose a table.
6. Select the Users table and click the * in the Columns.

7. Click Next to test out the query. You will see a table that shows the rows from the table.

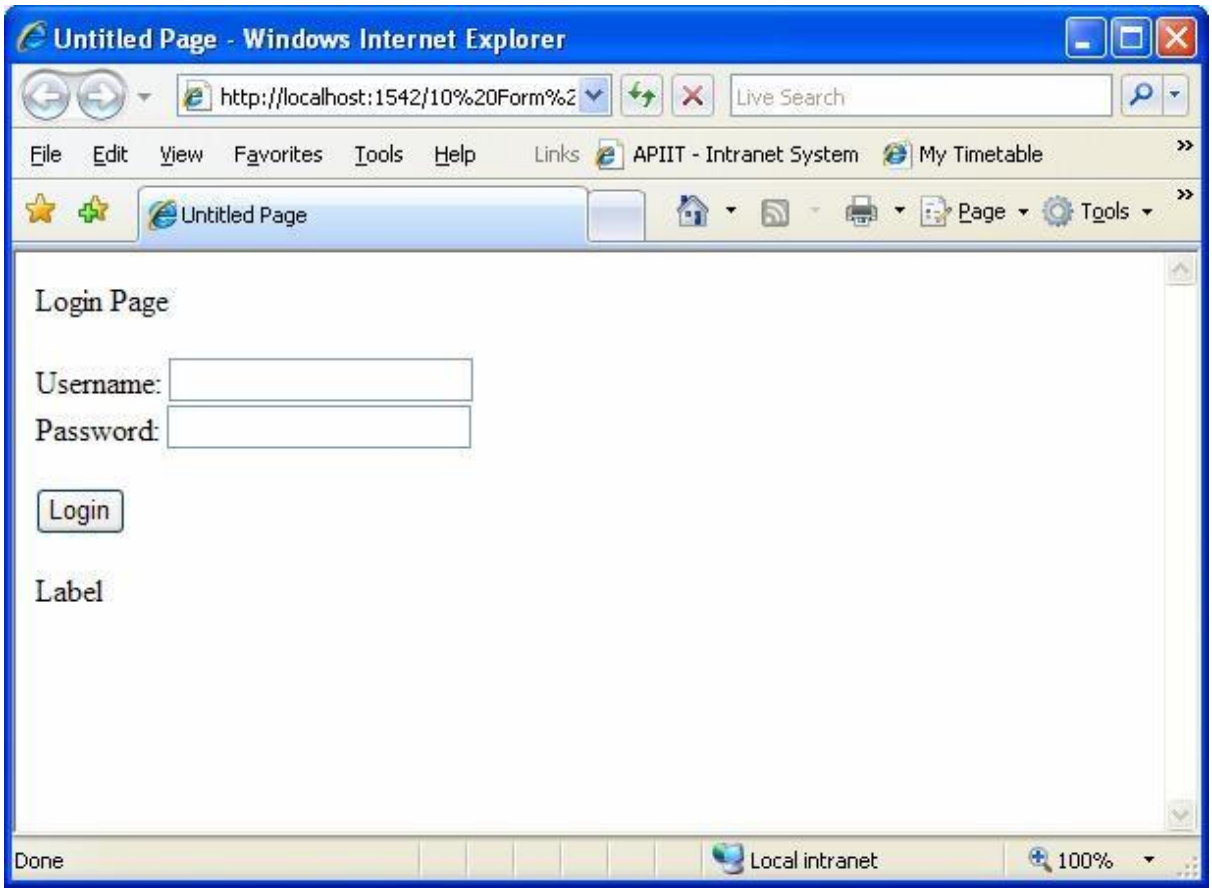


3. Double click the Login button and enter the following statements.

```
protected void Button1_Click(object sender, EventArgs e)
    {
        SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["LoginConnectionString"].Connection
String);
        con.Open();
        SqlCommand cmd = new SqlCommand("select count(*) from Users where Username = '"
+ TextBox1.Text + "' and Password = '" +TextBox2.Text+"'", con);
        int count = Convert.ToInt32(cmd.ExecuteScalar().ToString());

        if (count > 0)
        {
            SqlCommand cmdType = new SqlCommand("select UserType from Users where
Username = '" + TextBox1.Text + "'", con);
            string type = cmdType.ExecuteScalar().ToString().Replace(" ", "");
            Session["UserType"] = type;
            if (type == "admin")
                Response.Redirect("AdminOnlyPage.aspx");
            else if (type == "user")
                Response.Redirect("MemberOnlyPage.aspx");
        }
        else
        {
            this.Label3.ForeColor = System.Drawing.Color.Red;
            this.Label3.Text = "Login Failed!";
            return;
        }
        con.Close();
    }
```

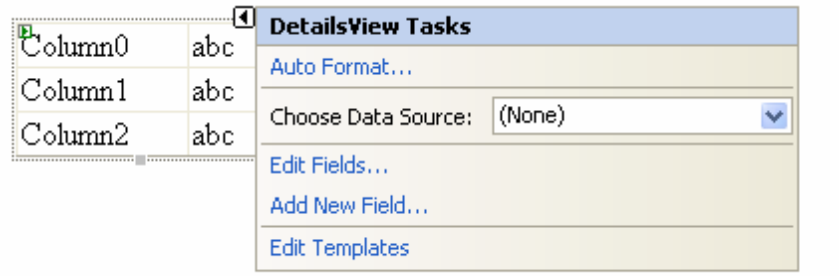
4. Open this page in the Web browser; it will look like figure below.



Creating a Members Only Page

1. Create a new Web Form and call the MembersOnlyPage.aspx.

2. Switch to Design view, then open the Data section of the toolbox and drag a SqlDataSource control over to the page.
3. Click Configure Data Source in the Smart Tag menu. This summons the familiar Configure Data Source Wizard dialog box.
4. In the Configure Data Source Wizard, choose LoginConnectionString.
5. Click **Next** in the Configure Data Source Wizard. The page that opens allows you to choose a table.
6. Select the Users table and click the * in the Columns.
7. Click Next to test out the query. You will see a table that shows the rows from the table.
8. Click Finish. Now the data source has been configured.
9. Drop a DetailView control on your form; expand its Smart Tag by clicking the little arrow in the upper-right corner, as shown in figure below.



10. Click the Choose Data Source drop-down list, and select SqlDataSource1, follow by checking the Enable Paging option only.
11. Double click the any blank space on the form button and enter the following statements for the page load event.

```
protected void Page_Load(object sender, EventArgs e)
{
      // Return to Login page if the session variable not exist if
      (Session["UserType"] == null ||
      !Session["UserType"].Equals("user"))
            Response.Redirect("Login.aspx");
}
```

12. Drag a LinkButton control on the page. Double click the LinkButton and enter the following statements.

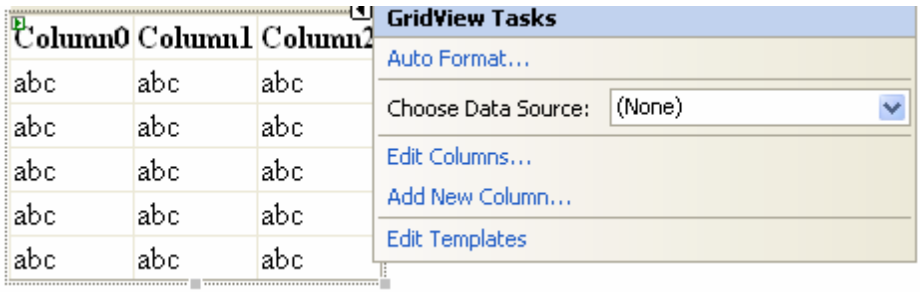
```
protected void LinkButton1_Click(object sender, EventArgs e)
{
      // Cancel the current session
      Session.Abandon();
      Response.Redirect("Login.aspx");
}
```

13. Open this page in the Web browser it will redirect to Login Page. Only successful login can view the page.

Creating an Admin Only Page

1. Create a new Web Form and call the AdminOnlyPage.aspx.

2. Switch to Design view, then open the Data section of the toolbox and drag SqlDataSource control over to the page.
3. Click Configure Data Source in the Smart Tag menu. This summons the familiar Configure Data Source Wizard dialog box.
4. In the Configure Data Source Wizard, choose LoginConnectionString.
5. Click **Next** in the Configure Data Source Wizard. The page that opens allows you to choose a table.
6. Select the Users table and click the * in the Columns. Finally, click Next to test out the query.
7. In the Test Query page, you can test the page if you would like by clicking the Test Query button.
8. Click Finish. Now the data source has been configured.
9. Drop a GridView control on your form; expand its Smart Tag by clicking the little arrow in the upper-right corner, as shown in figure below.



10. Click the Choose Data Source drop-down list, and select SqlDataSource1, follow by checking the Enable Paging option.

11. Double click the any blank space on the form button and enter the following statements for the page load event.

```
protected void Page_Load(object sender, EventArgs e)
{
      // Return to Login page if the session variable not exist if
      (Session["UserType"] == null ||
      !Session["UserType"].Equals("admin"))
            Response.Redirect("Login.aspx");
}
```

12. Drag a LinkButton control on the page. Double click the LinkButton and enter the following statements.

```
protected void LinkButton1_Click(object sender, EventArgs e)
{
      // Cancel the current session
      Session.Abandon();
      Response.Redirect("Login.aspx");
}
```

13. Open this page in the Web browser it will redirect to Login Page. Only successful login can view the page.