

## Lab 6 Registration – Web Form and Table Design

1. Create a new table as shown in Figure 1 and named it as **userTable**. For **Id**, set the **Identity Specification** to **True** in **Properties** window.


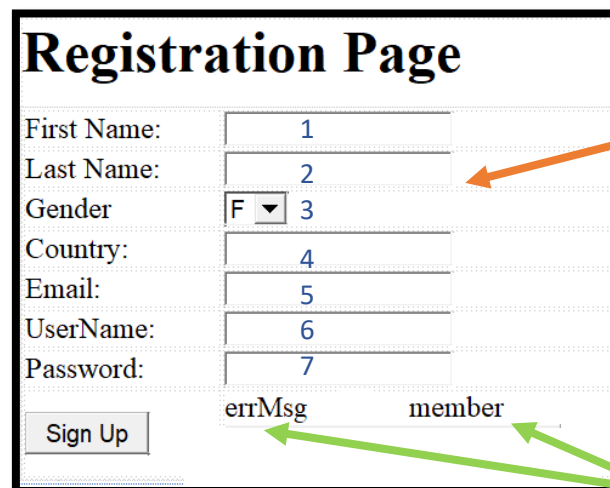
	Name	Data Type	Allow Nulls	Default
	Id	int	<input type="checkbox"/>	
	fname	nvarchar(50)	<input checked="" type="checkbox"/>	
	lname	nvarchar(50)	<input checked="" type="checkbox"/>	
	gender	nchar(10)	<input checked="" type="checkbox"/>	
	country	nvarchar(50)	<input checked="" type="checkbox"/>	
	email	nvarchar(50)	<input checked="" type="checkbox"/>	
	username	nvarchar(50)	<input checked="" type="checkbox"/>	
	password	nvarchar(50)	<input checked="" type="checkbox"/>	
	usertype	nchar(10)	<input checked="" type="checkbox"/>	

Figure 1: Table Design

2. Create a Web Form as shown in Figure 2 and named it as **memberRegistration.aspx**.



The figure shows a web form titled "Registration Page". It contains the following fields and controls:

- First Name: Text box with ID 1
- Last Name: Text box with ID 2
- Gender: Dropdown menu with ID 3, currently showing 'F'
- Country: Text box with ID 4
- Email: Text box with ID 5
- UserName: Text box with ID 6
- Password: Text box with ID 7
- Sign Up: Button
- errMsg: Label with ID 2 (indicated by a green arrow from the errMsg label in the form)
- member: Label with ID 2 (indicated by a green arrow from the member label in the form)

- 1 Id = fname
- 2 Id = lname
- 3 Id = gender
- 4 Id = country
- 5 Id = email
- 6 Id = username
- 7 Id = pwd

2 labels with Id = errMsg and usertype

These 2 labels are used for feedback and programming purposes.

Figure 2: Registration Page

3. Type in the following code in the *class* **Button1\_Click** as shown in Figure 4:-

```
try
{
    SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString);
    con.Open();

    string query = "select count(*) from userTable where username = '" + username.Text + "'";
    SqlCommand cmd = new SqlCommand(query, con);
    int check = Convert.ToInt32(cmd.ExecuteScalar().ToString());

    if (check > 0)
    {
        errMsg.Visible = true;
        errMsg.ForeColor = System.Drawing.Color.Red;
        errMsg.Text = "Username has been taken!";
    }
    else
    {
        //create record in a table called userTable
        string query1 = "insert into userTable (fname, lname, gender, country, email, username, password, usertype) values (@firstName, @lastName, @gender, @country, @email, @username, @password, @usertype) ";
        SqlCommand cmd1 = new SqlCommand(query1, con);

        cmd1.Parameters.AddWithValue("@firstName", fname.Text);
        cmd1.Parameters.AddWithValue("@lastName", lname.Text);
        cmd1.Parameters.AddWithValue("@gender", gender.SelectedItem.ToString());
        cmd1.Parameters.AddWithValue("@country", country.Text);
        cmd1.Parameters.AddWithValue("@email", email.Text);
        cmd1.Parameters.AddWithValue("@username", username.Text);
        cmd1.Parameters.AddWithValue("@password", pwd.Text);
        cmd1.Parameters.AddWithValue("@usertype", usertype.Text);
        cmd1.ExecuteNonQuery();
        Response.Redirect("login.aspx");
    }
    con.Close();
}
catch (Exception ex)
{
    errMsg.Visible = true;
    errMsg.ForeColor = System.Drawing.Color.Red;
    errMsg.Text = "Registration not successful!" + ex.ToString();
}
```

Figure 4: Insert Query

4. **Save > Build Solution > Run IIS** to test it.
5. Create a Web Form as shown in Figure 1 and named it as **adminRegistration.aspx**.
6. Change the **Text** value to **admin** for Label Id = **usertype**. Repeat step 3 to 5.

**ExecuteNonQuery** used for executing queries that does not return any data. It is used to execute the sql statements like update, insert, delete etc. ExecuteNonQuery executes the command and returns the number of rows affected.