.NET PROGRAMMING (C#)
(18 MCA 4 2 C)

UNIT – V

ADO.NET & ASP.NET

FACULTY:

Dr. K. Arthi

Assistant Professor,

Post Graduate And Research Department Of Computer Applications,

Government Arts College (Autonomous), Coimbatore 641 018.

CONTENT

- ADO.NET
- ASP.NET

ADO.NET

INTRODUCTION

.NET DATA ACCESS AND MANİPULATİON

OVERVİEW

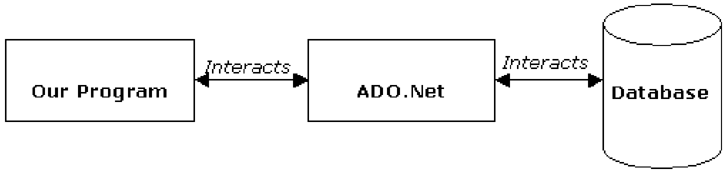
- What is ADO.NET?
- Disconnected vs. connected data access models
- ADO.NET Architecture
- ADO.NET Core Objects
- Steps of Data Access
- Advanced Techniques and UI Tools

WHAT İS ADO.NET?

- A data-access technology that enables applications to connect to data stores and manipulate data contained in them in various ways
- Former version was ADO (ActiveX Data Object)

WHAT İS ADO.NET?

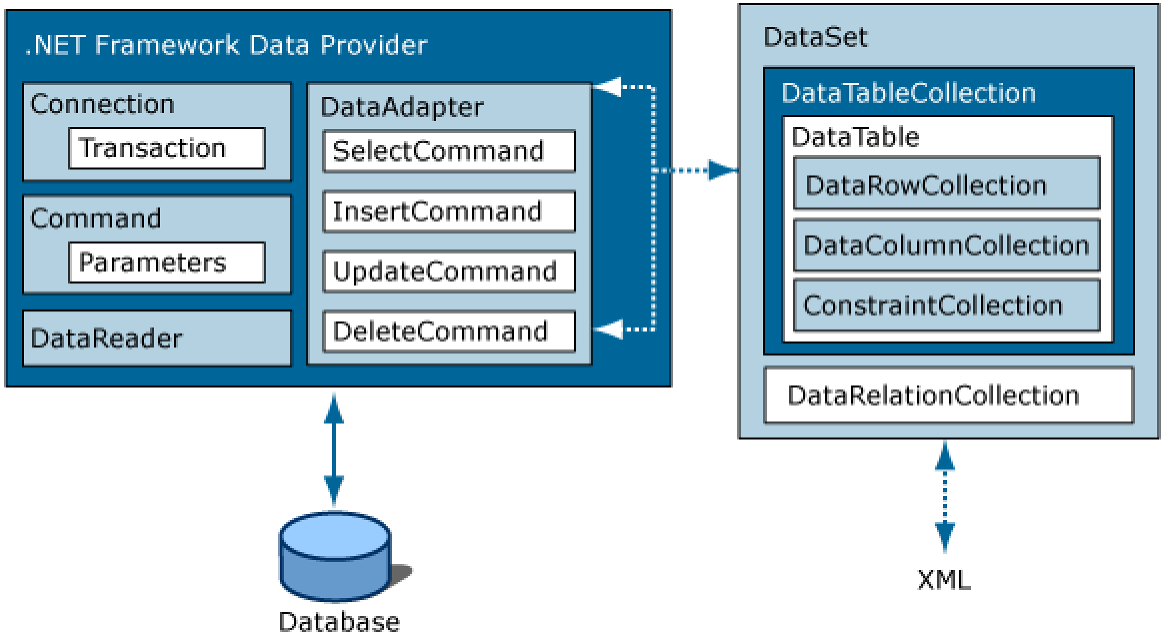
- An object oriented framework that allows you to interact with database systems



OBJECTİVE OF ADO.NET

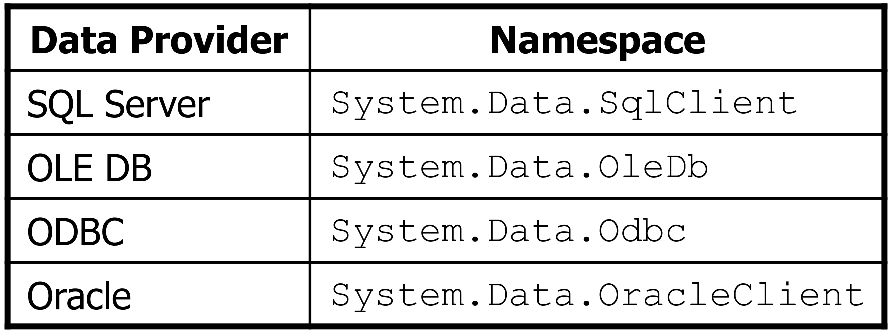
- Support disconnected data architecture,
- Tight integration with XML,
- Common data representation
- Ability to combine data from multiple and varied data sources
- Optimized facilities for interacting with a database

ADO.NET ARCHITECTURE



ADO.NET CORE OBJECTS

- Core namespace: System.Data
- .NET Framework data providers:

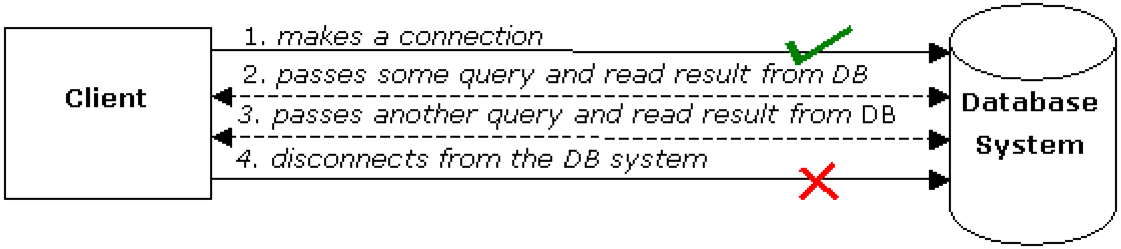


Data Provider	Namespace
SQL Server	`System.Data.SqlClient`
OLE DB	`System.Data.OleDb`
ODBC	`System.Data.Odbc`
Oracle	`System.Data.OracleClient`

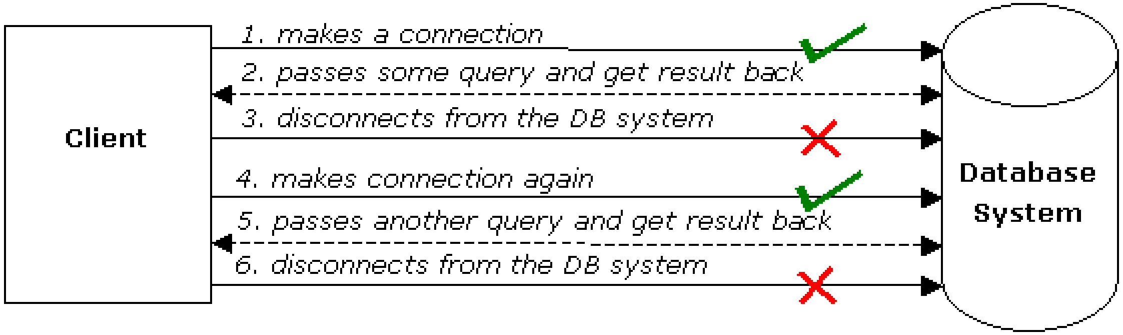
ADO.NET CORE OBJECTS

Object	Description
Connection	Establishes a connection to a specific data source. (Base class: DbConnection)
Command	Executes a command against a data source. Exposes Parameters and can execute within the scope of a Transaction from a Connection. (The base class: DbCommand)
DataReader	Reads a forward-only, read-only stream of data from a data source. (Base class: DbDataReader)
DataAdapter	Populates a DataSet and resolves updates with the data source. (Base class: DbDataAdapter)
DataTable	Has a collection of DataRows and DataColumns representing table data, used in disconnected model
DataSet	Represents a cache of data. Consists of a set of DataTables and relations among them

CONNECTED DATA ACCESS MODEL


Client
1. makes a connection
2. passes some query and read result from DB
3. passes another query and read result from DB
4. disconnects from the DB system
Database System

DİSCONNECTED DATA ACCESS MODEL


Client
1. makes a connection
2. passes some query and get result back
3. disconnects from the DB system
4. makes connection again
5. passes another query and get result back
6. disconnects from the DB system
Database System

PROS AND CONS

	Connected	Disconnected
Database Resources	-	+
Network Traffic	-	+
Memory Usage	+	-
Data Access	-	+

STEPS OF DATA ACCESS: DİSCONNECTED ENVİRONMENT

- Defining the connection string
- Defining the connection
- Defining the command
- Defining the data adapter
- Creating a new DataSet object
- SELECT -> fill the dataset object with the result of the query through the data adapter
- Reading the records from the DataTables in the datasets using the DataRow and DataColumn objects
- UPDATE, INSERT or DELETE -> update the database through the data adapter

EXAMPLE

```
using System;
using System.Data;
using System.Data.SqlClient;

namespace SampleClass
{
  class Program
  {
     static void Main(string[] args)
     {
       string connStr =
                      Properties.Settings.Default.connStr;
         SqlConnection conn = new SqlConnection(connStr);
       string queryString = "SELECT * from titles;";
         SqlDataAdapter da = new
                             SqlDataAdapter(queryString,conn);

          DataSet ds = new DataSet();
          da.fill(ds);
          // Work on the data in memory using
          // the DataSet (ds) object
       }
   }
}
```

DİSCONNECTED:
UPDATE, DELETE, INSERT

`SqlDataAdapter da = new SqlDataAdapter();`
`DataSet ds = new DataSet();`
`SqlCommandBuilder cmdBuilder = new SqlCommandBuilder(da);`
`da.Fill(ds);`
	INITIAL CODE

`DataRow dr = ds.Tables[0].Rows[0];`
`dr.Delete();`
`da.UpdateCommand = builder.GetUpdateCommand();`
`da.Update(ds);`
	DELETE

`DataRow dr = ds.Tables[0].Rows[0];`
`dr["CustomerName"] = "John";`
`da.UpdateCommand = builder.GetUpdateCommand();`
`da.Update(ds);`
	UPDATE

`DataRow dr = ds.Tables[0].NewRow();`
`dr["CustomerName"] = "John";`
`dr["CustomerSurName"] = "Smith";`
`ds.Tables[0].Rows.Add(dr);`
`da.UpdateCommand = builder.GetUpdateCommand();`
`da.Update(ds);`
	INSERT

STEPS OF DATA ACCES : CONNECTED ENVİRONMENT

- Create connection
- Create command (select-insert-update-delete)
- Open connection
- If SELECT -> use a DataReader to fetch data
- If UPDATE,DELETE, INSERT -> use command object’s methods
- Close connection

EXAMPLE

```
static void Main()
{
    string connectionString = Properties.Settings.Default.connStr;
    string queryString = "SELECT CategoryID, CategoryName FROM dbo.Categories;";
    SqlConnection connection = new SqlConnection(connectionString);
    SqlCommand command = new SqlCommand(queryString,connection);
    try
    {
        connection.Open();
        SqlDataReader reader = command.ExecuteReader();
        while (reader.Read())
        {
            Console.WriteLine("\t{0}\t{1}“,reader[0],reader[1]);
        }
        reader.Close();
            connection.close();
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
}
```

CONNECTED – UPDATE, DELETE, INSERT

- Command class core methods:
  - ExecuteNonQuery : Executes a SQL statement against a connection object
  - ExecuteReader: Executes the CommandText against the Connection and returns a DbDataReader
  - ExecuteScalar: Executes the query and returns the first column of the first row in the result set returned by the query

CONNECTED – UPDATE, DELETE, INSERT

```
string connString = Properties.Settings.Default.connStr;
SqlConnection conn = new SqlConnection(connString);
SqlCommand cmd = new SqlCommand("delete from
   Customers" + "where custID=12344", conn);
conn.Open();
cmd.ExecuteNonQuery();
conn.Close();
```

Can be an update or insert command

CHOOSİNG A DATAREADER OR A DATASET

- The type of functionality application requires should be considered
- Use a dataset to:
  - Cache data locally in your application so that you can manipulate it
  - Remote data between tiers or from an XML Web service
  - Interact with data dynamically such as binding to a Windows Forms control or combining and relating data from multiple sources
  - Perform extensive processing on data without requiring an open connection to the data source, which frees the connection to be used by other clients
- If readonly data is needed use DataReader to boost performance

BEST PRACTİCES

- Don’t create a new connection string for every code connecting to DB
- Use app.config file to keep your connection strings through the application scope
  - Right click on project and select properties
  - Select settings from the left tabbed menu
  - add the connection string to the table and save project, Name field is the name of the string to access at runtime
- Accessing settings at runtime:
- You can keep any other variable to reach at runtime using this technique

```
string connStr = Properties.Settings.Default.connStr;
```

AFTER .NET FRAMEWORK 2.0

- To minimize the code written by developers new UI tools and objects have been intoduced with .NET Framework 2.0

AFTER .NET FRAMEWORK 2.0

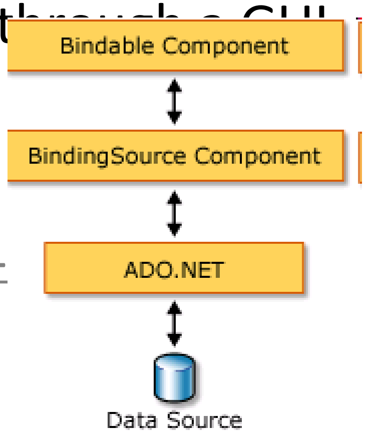
- Strongly Typed vs Untyped Datasets
  - Untyped: DataSet and DataTables included are created at runtime completely using code
  - Strongly Typed: Dataset is created at design time, it is defined by an xsd schema

AFTER .NET FRAMEWORK 2.0

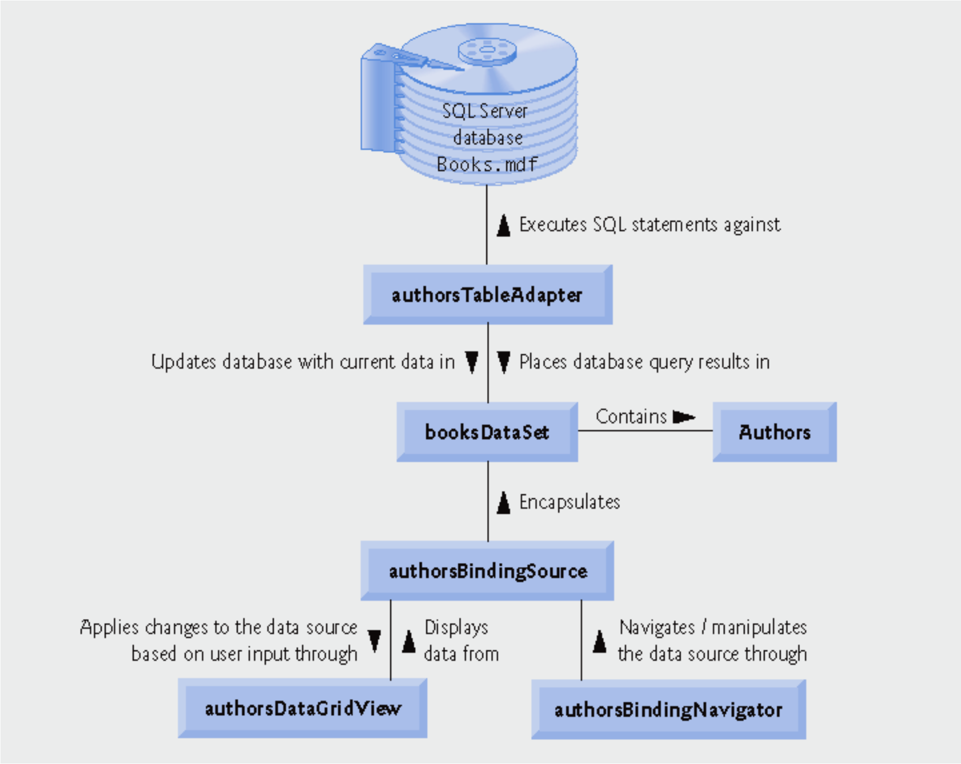
- TableAdapter
  - provides communication between your application and a database
  - Provides update/delete/insert functions
  - Encapsulates a SQLDataAdapter object
  - MSDN link:
  - http://msdn.microsoft.com/en-us/library/bz9tthwx(VS.80).aspx

AFTER .NET FRAMEWORK 2.0

- BindingSource
  - Binds UI components to a strongly typed Dataset
  - Ex: Binds a DataGridView to a DataTable
  - Sets a DataSet as a datasource and datamember as a dataset table
  - EndEdit() method: Applies changes made to data through a GUI control to the data source bound to that control
  - MSDN link:
  - http://msdn.microsoft.com/en-us/library/xxxf124



AFTER .NET FRAMEWORK 2.0


SQL Server
database
Books.mdf
Executes SQL statements against
authorsTableAdapter
Updates database with current data in
Places database query results in
booksDataSet
Contains
Authors
Encapsulates
authorsBindingSource
Applies changes to the data source based on user input through
Displays data from
Navigates / manipulates the data source through
authorsDataGridView
authorsBindingNavigator

An example of databinding model

AFTER .NET FRAMEWORK 2.0

- Binding Navigator
  - Used for creating a standardized means for users to search and change data on a Windows Form
  - Used with BindingNavigator with the BindingSource component to enable users to move through data records on a form and interact with the records
  - MSDN link:
  - http://msdn.microsoft.com/en-us/library/8zhc8d2f(VS.80).aspx

AFTER .NET FRAMEWORK 2.0

- TableAdapterManager
  - New component in Visual Studio 2008
  - Builds upon existing data features (typed datasets and TableAdapters) and provides the functionality to save data in related data tables.
  - Manages inserts/updates/deletes without violating the foreign-key constraints
  - MSDN link:
  - http://msdn.microsoft.com/en-us/library/bb384426.aspx

HANDS ON: CREATE A DB NAVİGATOR

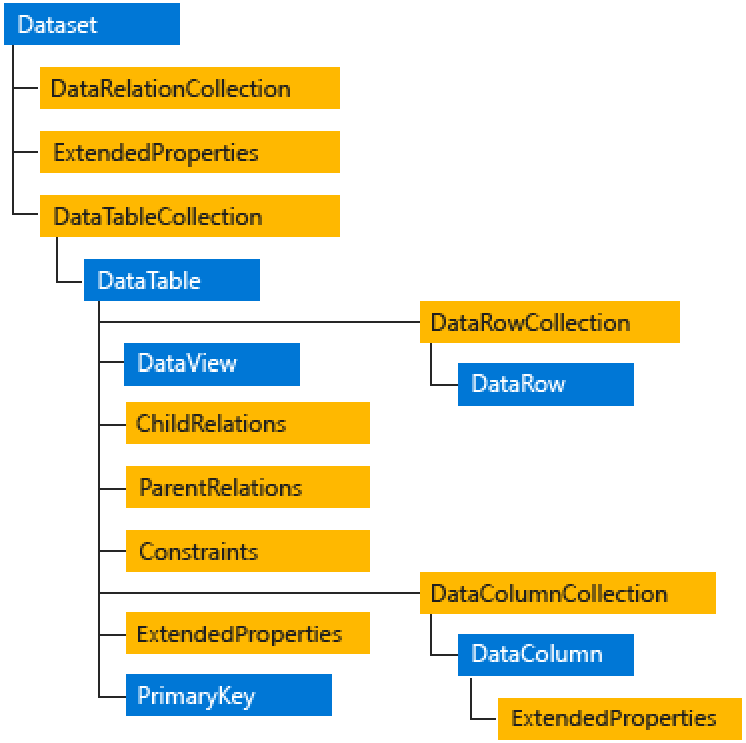
- Create a DB navigator with UI components and wizards

HANDS ON: CUSTOM QUERİES

- Create a filter mechanism on an DataGridView with using custom queries
- Manage datatables and TableAdapters

HANDS ON: MANAGİNG MULTİPLE TABLES

- Create a navigation system with using the relations between two tables



DATABASE CONNECTIVITY

CORE OBJECTS OF .NET FRAMEWORK PROVIDERS

OBJECTS	BASE CLASS	DESCRIPTION
Connection	DbConnection	Establishes a connection to a specific data source.
Command	DbCommand	Executed a command against a data source.
DataReader	DbDataReader	Reads a forward-only, read-only stream of data from a data source.
DataAdapter	DbDataAdapter	Populates a DataSet and resolves updates with the data source

ADO.NET DATA PROVIDERS

DATA PROVIDER	NAMESPACE
SQL Server	System.Data.SqlClient;
OLEDB	System.Data.OleDb;
ODBC	System.Data.Odbc;
ORACLE	System.Data.OracleClient;

DATABASE CONNECTIVITY

SQL CLIENT

SQL SERVER

- It provides access for Microsoft SQL Server
- Uses the “System.Data.SqlClient;” namespace.
- G.F.:

```
using System.Data.SqlClient;

class Program

{

-----

string connectionString = "Data Source=(local);Initial Catalog=Northwind;“ + "Integrated Security=true";

-----

}
```

Connecting to a database

```
using(SqlConnection Conn = new SqlConnection())

Conn.ConnectionString = “Server = [ServerName]; Database = [DatabaseName]; Trusted_Connection = true";
```

Adding data into a database

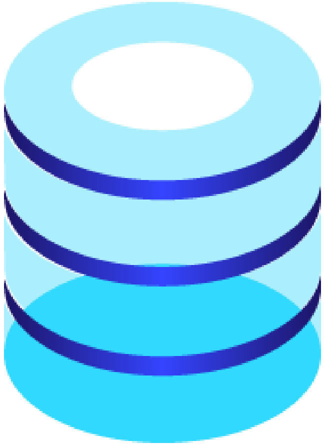
```
SqlCommand insertCommand = new SqlCommand (INSERT INTO Tablename ([Column1], [Column2],… VALUES (Value1,
Value2,… ));
```

Updating a record in Database

```
SqlCommand updateCommand = new SqlCommand (UPDATE TABLE Tablename SET ([FieldName] = Value,…) WHERE CONDITION;
```

Deleting a record from database

```
SqlCommand deleteCommand = new SqlCommand (DELETE FROM Tablename WHERE CONDITION;
```



DATABASE CONNECTIVITY

OLEDB

OLEDB

- It provides access for Microsoft Access Database.
- Uses the “System.Data.OleDb;” namespace.
- G.F.:

```
using System.Data.OleDb;

class Program

{

-----

OleDbConnection Connection_Variable = new
OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0; Data Source =
“Your_Database_Directory;";

-----

}
```

EXAMPLE FOR CONNECTING TO A DATABASE

```
using System.Data.OleDb;
class Program
{
OleDbConnection Con = new OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0; Data Source = MyDb.accdb;";
private void BtnSubmit_Click(object sender, EventArgs e)
{
OleDbCommand Ins = con.CreateCommand();
Ins.Open();
Ins.CommandText = "INSERT INTO Student(FirstName, LastName) VALUES (‘” + textBox1.Text + "','" + textBox2.Text + “‘)”;
Ins.Connection = Con;
Ins.ExecuteNonQuery();
MessageBox.Show("Record Submitted","Congrats");
Con.Close();
 }
}
```

DATABASE CONNECTIVITY

ODBC

ODBC

- It provides access for Microsoft Access Database
- Uses the “System.Data.Odbc;” namespace.
- G.F.:

```
using System.Data.Odbc;

class Program

{

-----

string connectionString = "Driver={Microsoft Access Driver (*.mdb)};“ + "DBQ = Databasedirectory.mdb;”;

-----

}
```

EXAMPLE FOR
CONNECTING TO A DATABASE
INSERTING DATA INTO A DATABASE

```
static private void InsertRow(string connectionString)
{
//Connecting to database
string connectionString = "Driver={Microsoft Access Driver (*.mdb)};“ + "Dbq = Database directory;”;
//Adding a record
string queryString = "INSERT INTO Customers (CustomerID, CompanyName) Values('NWIND', 'Northwind Traders')";
OdbcCommand command = new OdbcCommand(queryString);
using (OdbcConnection connection = new OdbcConnection(connectionString))
 {
  command.Connection = connection;
  connection.Open();
  command.ExecuteNonQuery();
  // The connection is automatically closed at the end of the Using block.
 }
}
```

DATABASE CONNECTIVITY

ORACLE CLIENT

ORACLE CLIENT

- It provides access for Oracle data sources.
- It requires ODAC, and Oracle Database X or Oracle Database XE.
- Uses the “System.Data.OracleClient;” namespace.
- G.F.:

```
using System.Data.OracleClient;

class Program

{

-----

string oradb = "Data Source = (DESCRIPTION =" +
"(ADDRESS = (PROTOCOL = TCP)(HOST = Your host name)(PORT = 1521))" +
"(CONNECT_DATA = " +
"(SERVER = DEDICATED)" +
"(SERVICE_NAME = XE))); +

“User Id=your_user_id; assword=******;";

-----

}
```

EXAMPLE FOR CONNECTING TO A DATABASE

```
string connectionString = "Data Source = ThisOracleServer; Integrated Security = yes;";
string queryString = "SELECT CUSTOMER_ID, NAME FROM DEMO.CUSTOMER";
using (OracleConnection connection = new OracleConnection(connectionString))
{
 OracleCommand command = connection.CreateCommand();
 command.CommandText = queryString;
 try
 {
  connection.Open();
  OracleDataReader reader = command.ExecuteReader();
  while (reader.Read())
  {
   Console.WriteLine("\t{0}\t{1}", reader[0], reader[1]);
  }
  reader.Close();
 }
 catch (Exception ex)
 {
  Console.WriteLine(ex.Message);
 }
}
```

CONNECTING TO A DATABASE

VISUAL STUDIO C#

STEP 1: OPEN VISUAL STUDIO

- Go to Start Menu.
  - Click All Programs.
    - Microsoft Visual Studio [yyyy]
      - Visual Studio

STEP 2: CREATE NEW C# PROJECT

- Select "Windows Forms Application" under Visual C# Group.
  - Provide a Suitable name for Project.
    - Click "OK".

STEP 3: FORM DESIGN

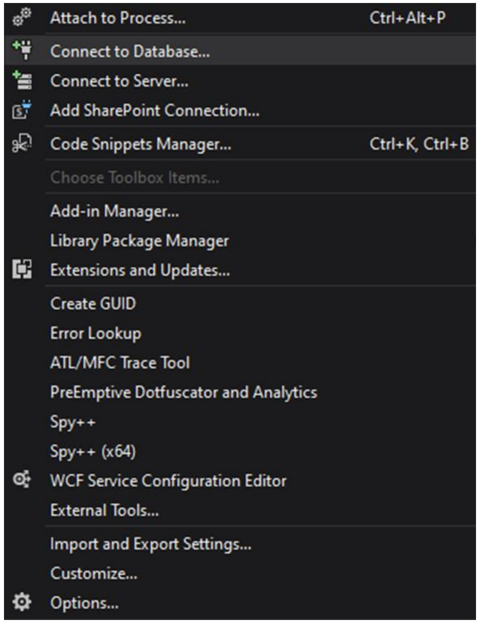
- Add some Controls from the Tool box. (i.e.)
  - 2 x Labels
  - 2 x Text boxes
  - 1 x Button

STEP 4: CONTROL PROPERTIES

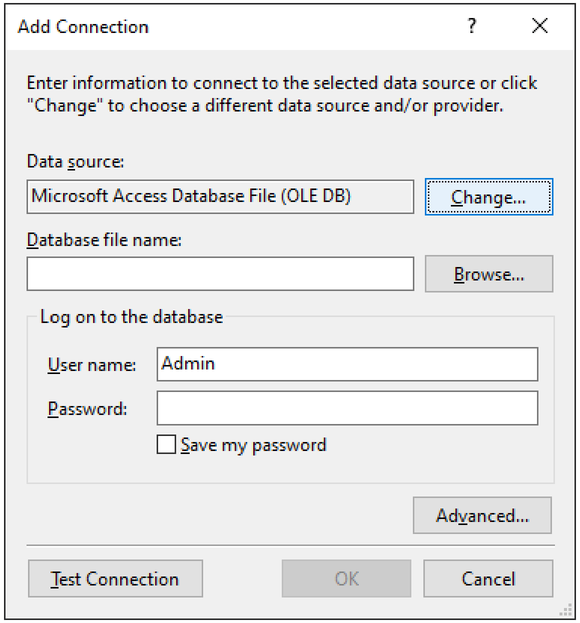
- Select the Controls and provide a valid name.
  - E.g.:
    - "Username", "Password" for Labels and Textboxes
    - "Submit" for Button.
- Change the appearances of controls if preferred.

STEP 5: CONNECT TO A DATABASE

- Click "TOOLS" Menu.
- Click "Connect to Database".


Attach to Process... Ctrl+Alt+P
Connect to Database...
Connect to Server...
Add SharePoint Connection...
Code Snippets Manager... Ctrl+K, Ctrl+B
Choose Toolbox Items...
Add-in Manager...
Library Package Manager
Extensions and Updates...
Create GUID
Error Lookup
ATL/MFC Trace Tool
PreEmptive Dotfuscator and Analytics
Spy++
Spy++ (x64)
WCF Service Configuration Editor
External Tools...
Import and Export Settings...
Customize...
Options...

Click "Change" button to change "DataSource" & "DataProvider"


Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft Access Database File (OLE DB)

Database file name:

Log on to the database

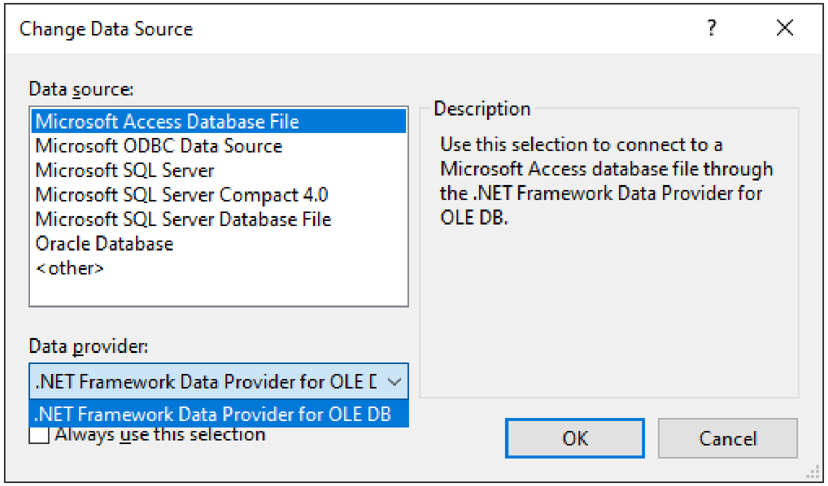
User name: Admin

Password:

☐ Save my password

Select preferred "DataSource" & "DataProvider".

Click "OK".



Change Data Source

Data source:

- Microsoft Access Database File
- Microsoft ODBC Data Source
- Microsoft SQL Server
- Microsoft SQL Server Compact 4.0
- Microsoft SQL Server Database File
- Oracle Database
- <other>

Description

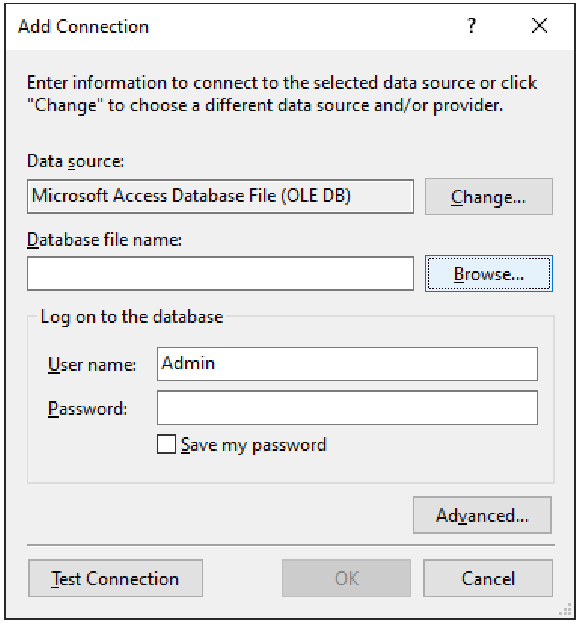
Use this selection to connect to a Microsoft Access database file through the .NET Framework Data Provider for OLE DB.

Data provider: .NET Framework Data Provider for OLE D

.NET Framework Data Provider for OLE DB

□ Always use this selection

Click "Browse" button to Select Database.



Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft Access Database File (OLE DB)

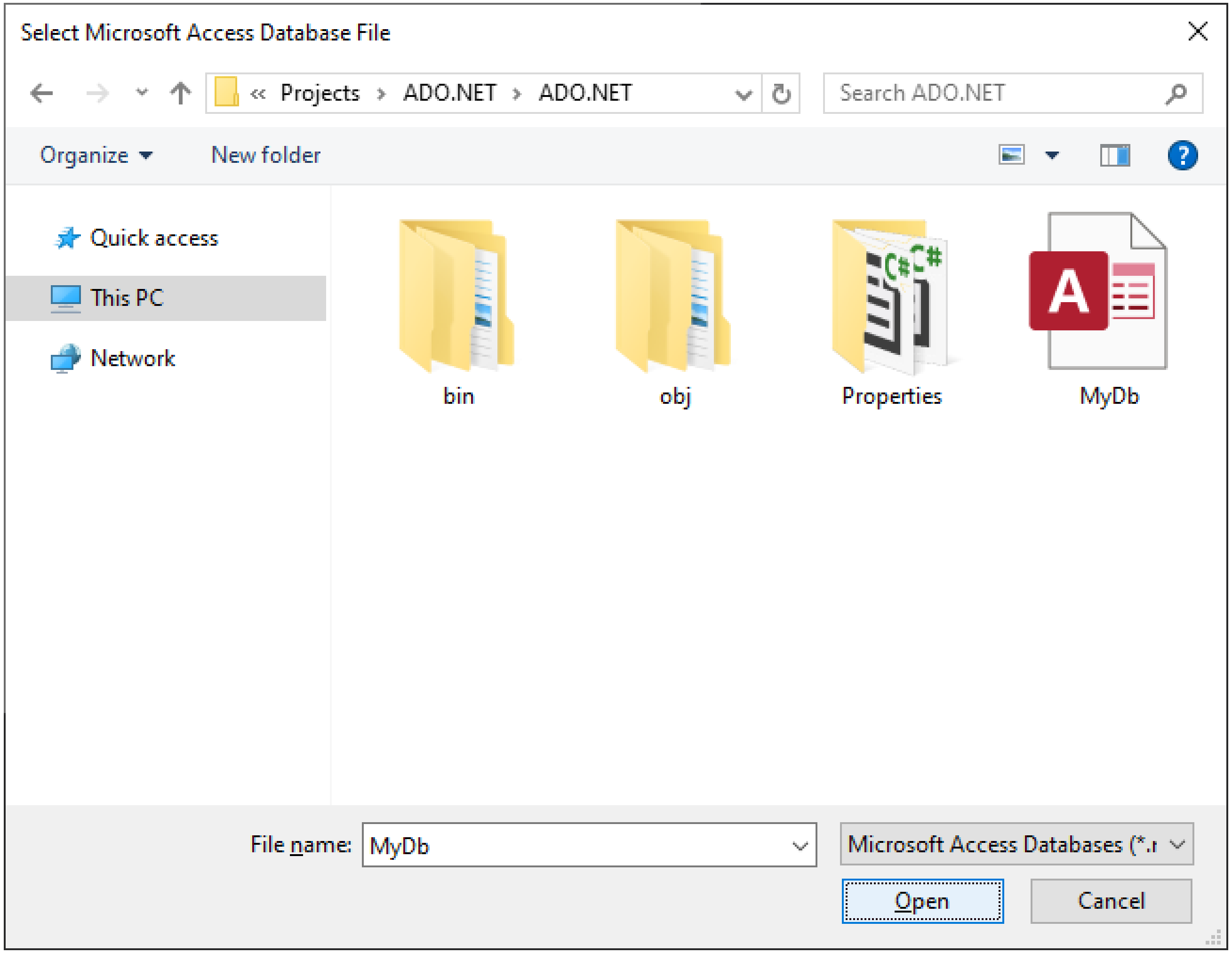
Database file name:

Log on to the database

User name: Admin

Password:

□ Save my password



Select Microsoft Access Database File

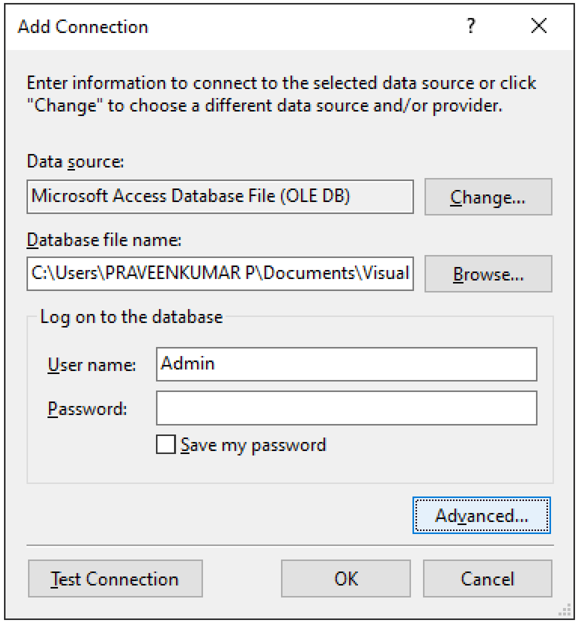
« Projects › ADO.NET › ADO.NET

File name: MyDb

Microsoft Access Databases (*.r

Navigate to the preferred Directory. Select the database and Click "OK".

Click "Advanced" to view Advanced Properties.



Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft Access Database File (OLE DB)

Database file name: C:\Users\PRAVEENKUMAR P\Documents\Visual

Log on to the database

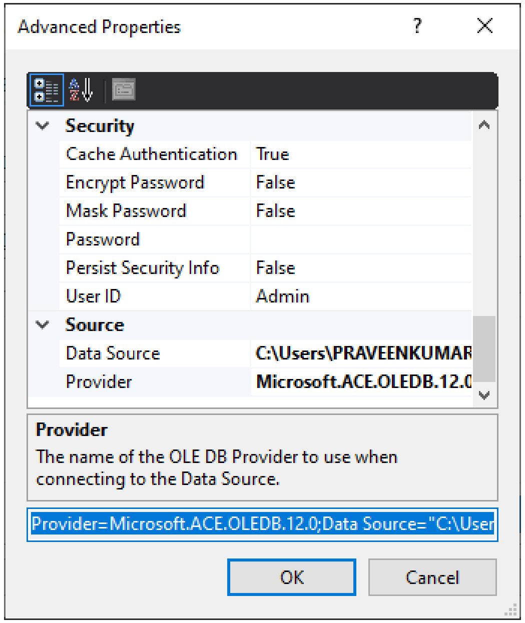
User name: Admin

Password:

□ Save my password

Make modifications if Needed.

Copy the Highlighted Text to Paste it on the Connection String. Click "OK".



Advanced Properties

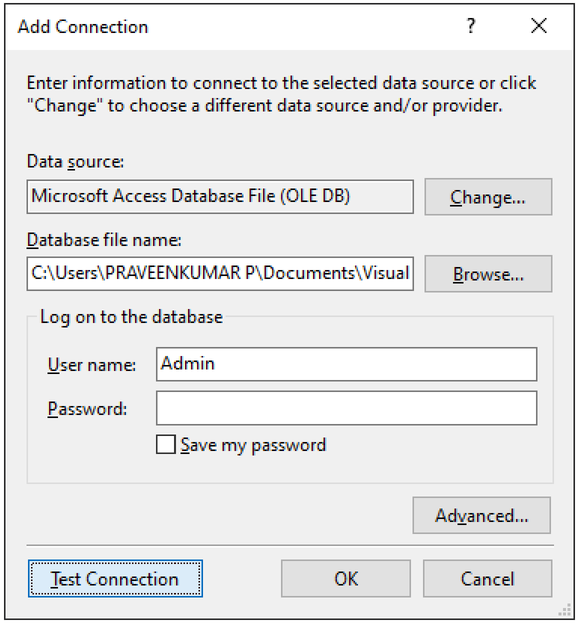
Security	
Cache Authentication	True
Encrypt Password	False
Mask Password	False
Password	
Persist Security Info	False
User ID	Admin
Source	
Data Source	**C:\Users\PRAVEENKUMAR**
Provider	**Microsoft.ACE.OLEDB.12.0**

Provider

The name of the OLE DB Provider to use when connecting to the Data Source.

Provider=Microsoft.ACE.OLEDB.12.0;Data Source="C:\User

Click “Test Connection” to verify that the database is connected correctly.



Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft Access Database File (OLE DB)

Database file name: C:\Users\PRAVEENKUMAR P\Documents\Visual

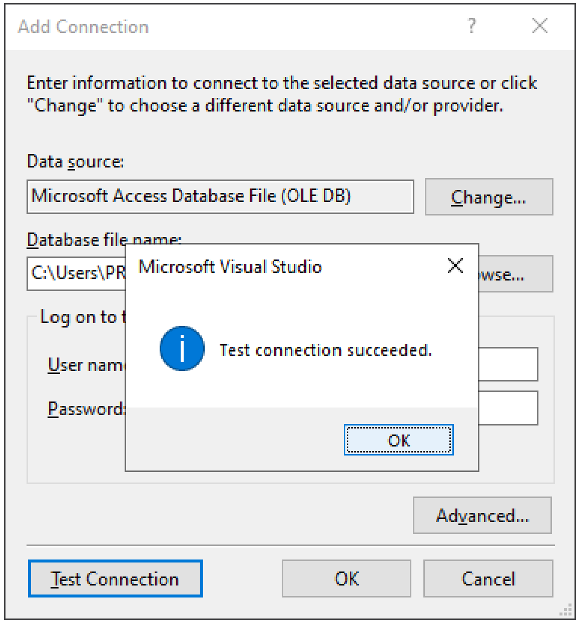
Log on to the database

User name: Admin

Password:

□ Save my password

Click “OK” to close the message box and Click “OK” again to Finish Connecting to Database.



Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft Access Database File (OLE DB)

Database file name: C:\Users\PR

Log on to t

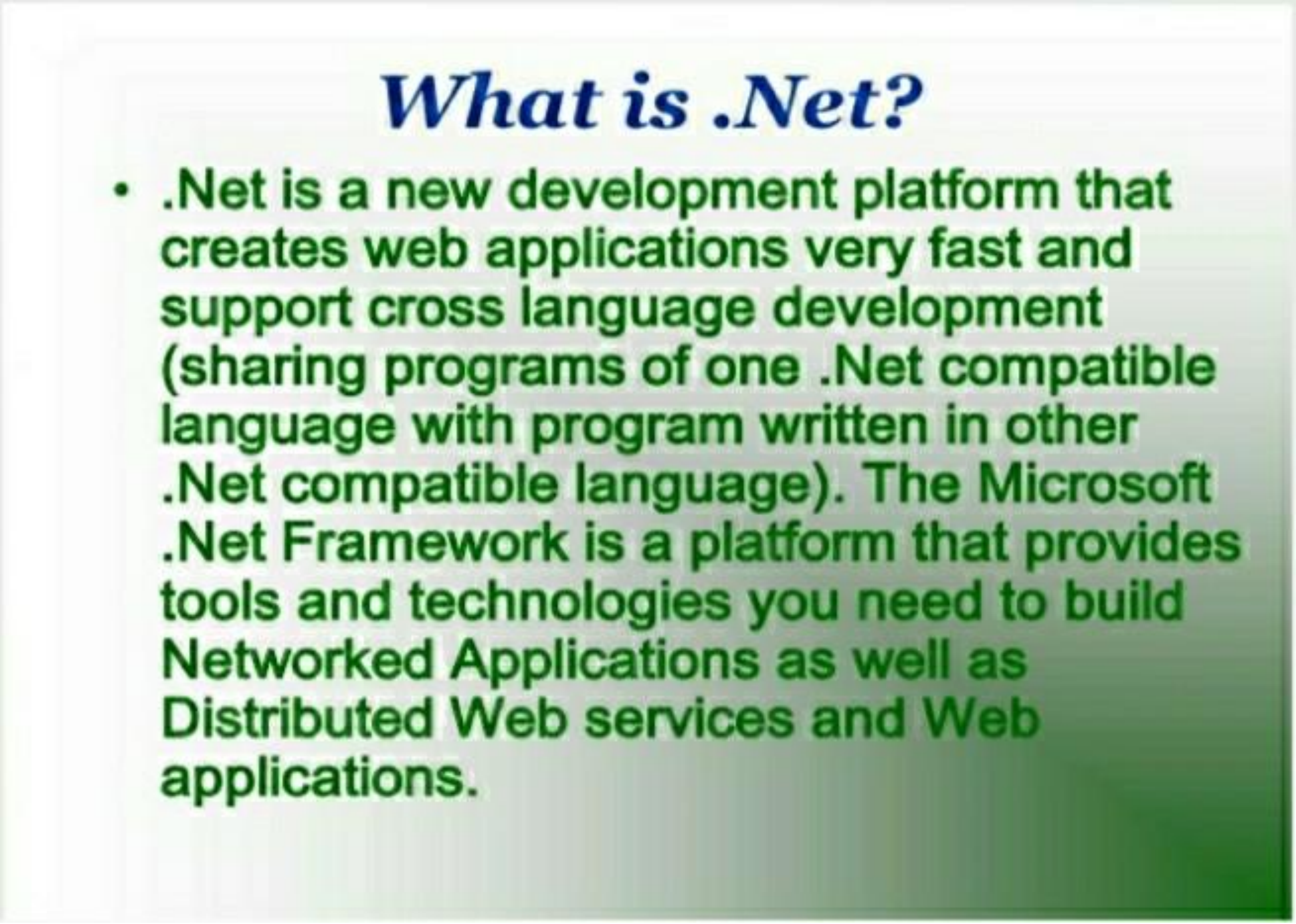
User nam

Password:

Microsoft Visual Studio

Test connection succeeded.

ASP.NET

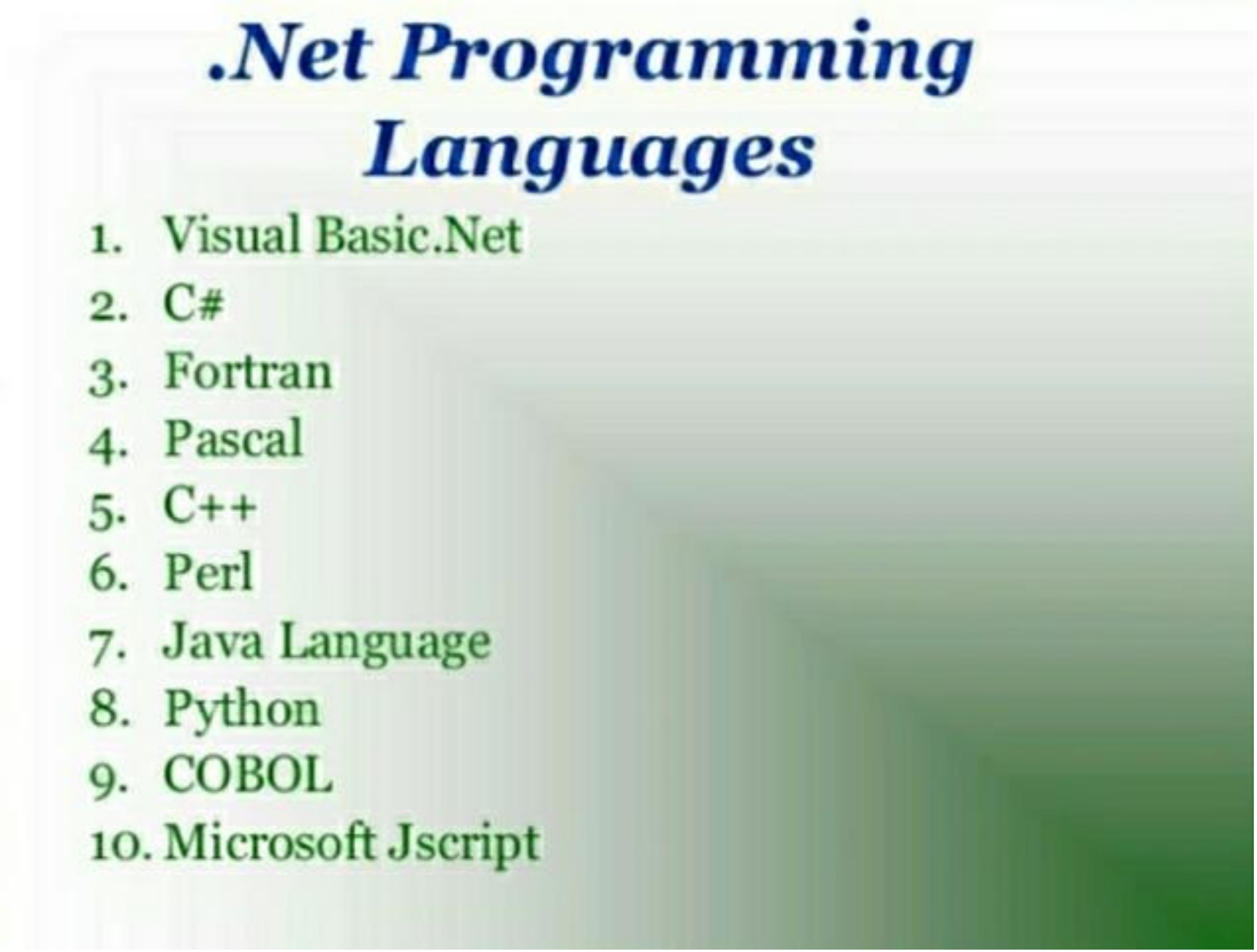


What is .Net?

- .Net is a new development platform that creates web applications very fast and support cross language development (sharing programs of one .Net compatible language with program written in other .Net compatible language). The Microsoft .Net Framework is a platform that provides tools and technologies you need to build Networked Applications as well as Distributed Web services and Web applications.

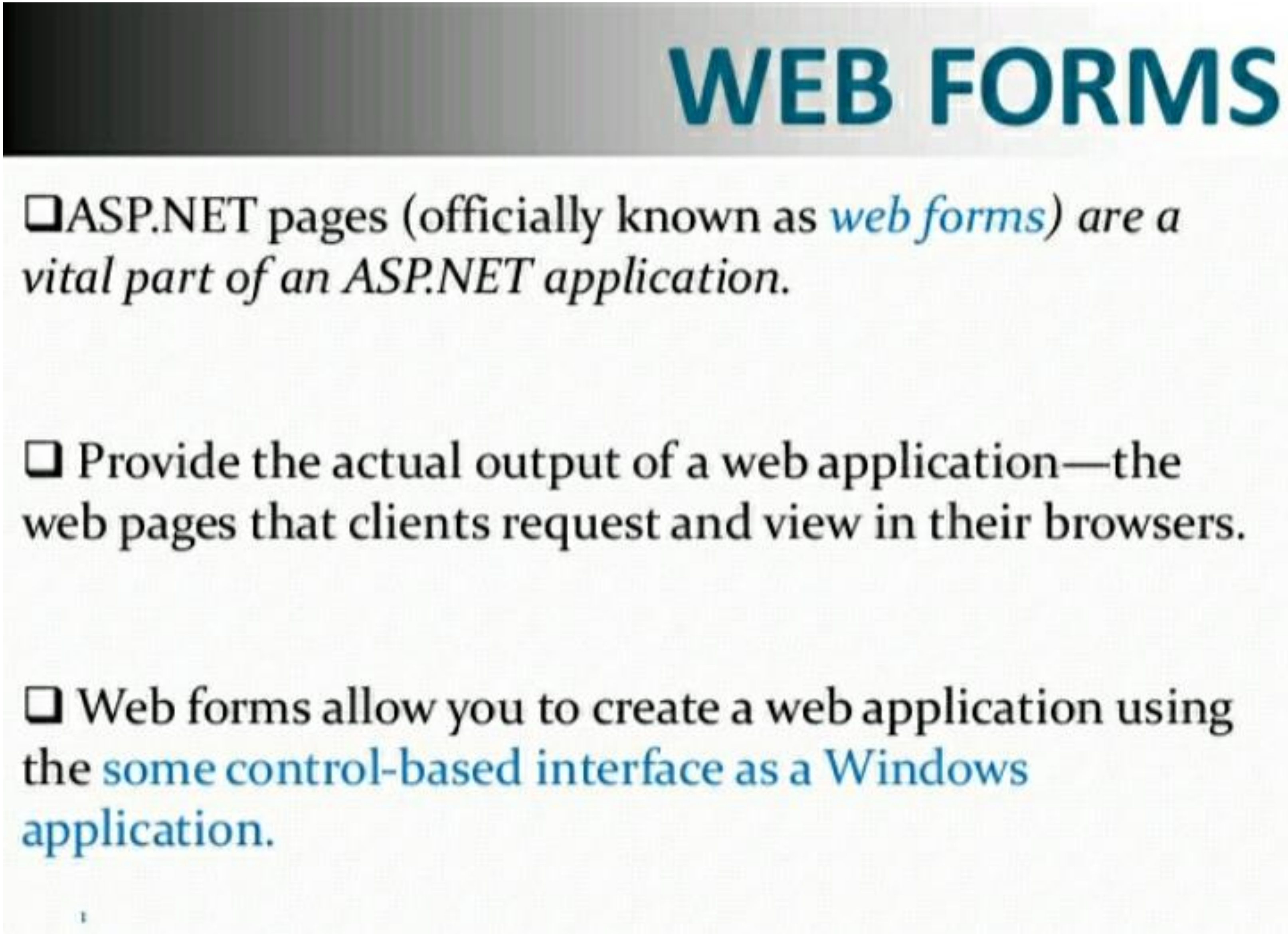
ABOUT ASP.NET

- ASP-active server pages.
- Introduced in 1998 as microsoft first server side scripting engine.
- Asp.net is an new asp generation.
- Asp.net ghave the extension .aspx,and normally written in vb or c#



.Net Programming Languages

1. Visual Basic.Net
2. C#
3. Fortran
4. Pascal
5. C++
6. Perl
7. Java Language
8. Python
9. COBOL
10. Microsoft Jscript



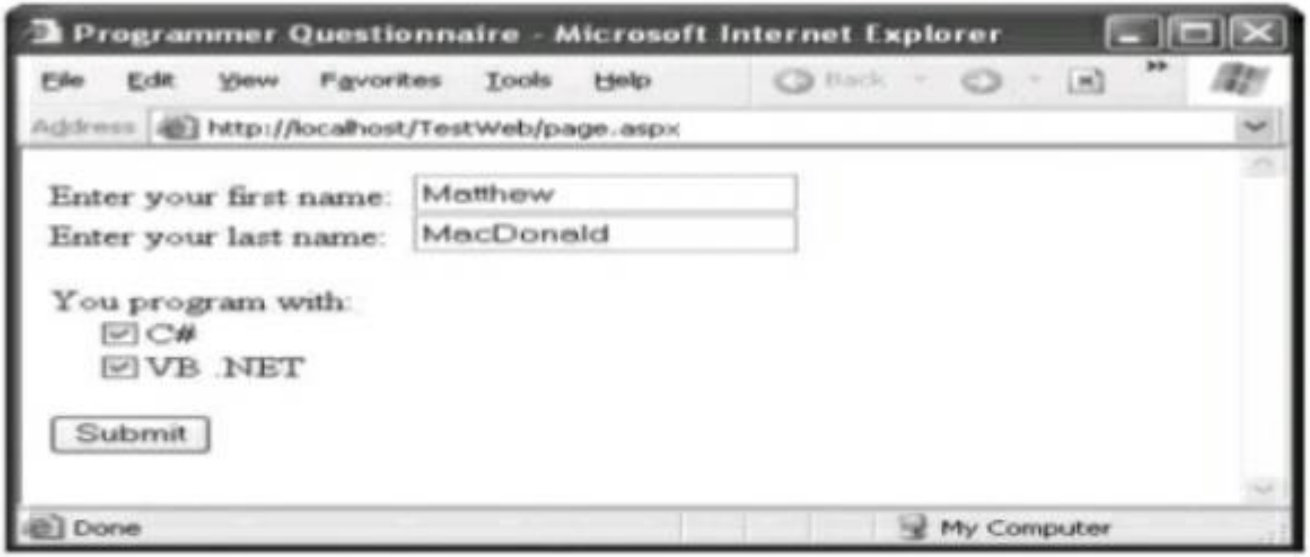
WEB FORMS

- ❑ASP.NET pages (officially known as web forms) *are a vital part of an ASP.NET application.*
- ❑ Provide the actual output of a web application—the web pages that clients request and view in their browsers.
- ❑ Web forms allow you to create a web application using the some control-based interface as a Windows application.

HTML Forms

- ❑ Simplest way to send client-side data to the server is using a <form> tag

- ❑ Inside the <form> tag, can place other <input> tags to represent basic user interface ingredients


Programmer Questionnaire - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address
Done
My Computer

http://localhost/TestWeb/page.aspx

Enter your first name: Matthew

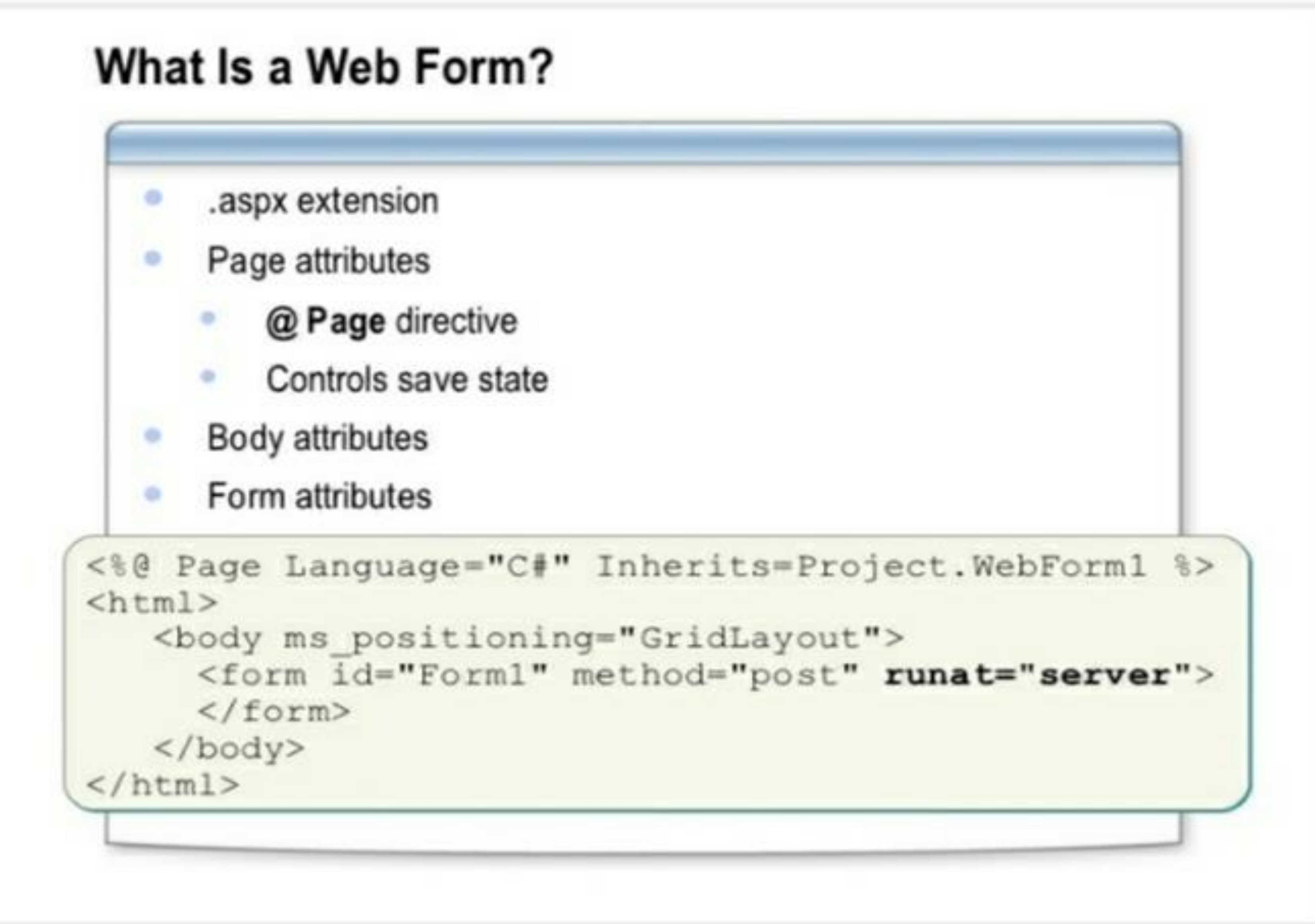
Enter your last name: MacDonald

You program with:

☑ C#

☑ VB .NET

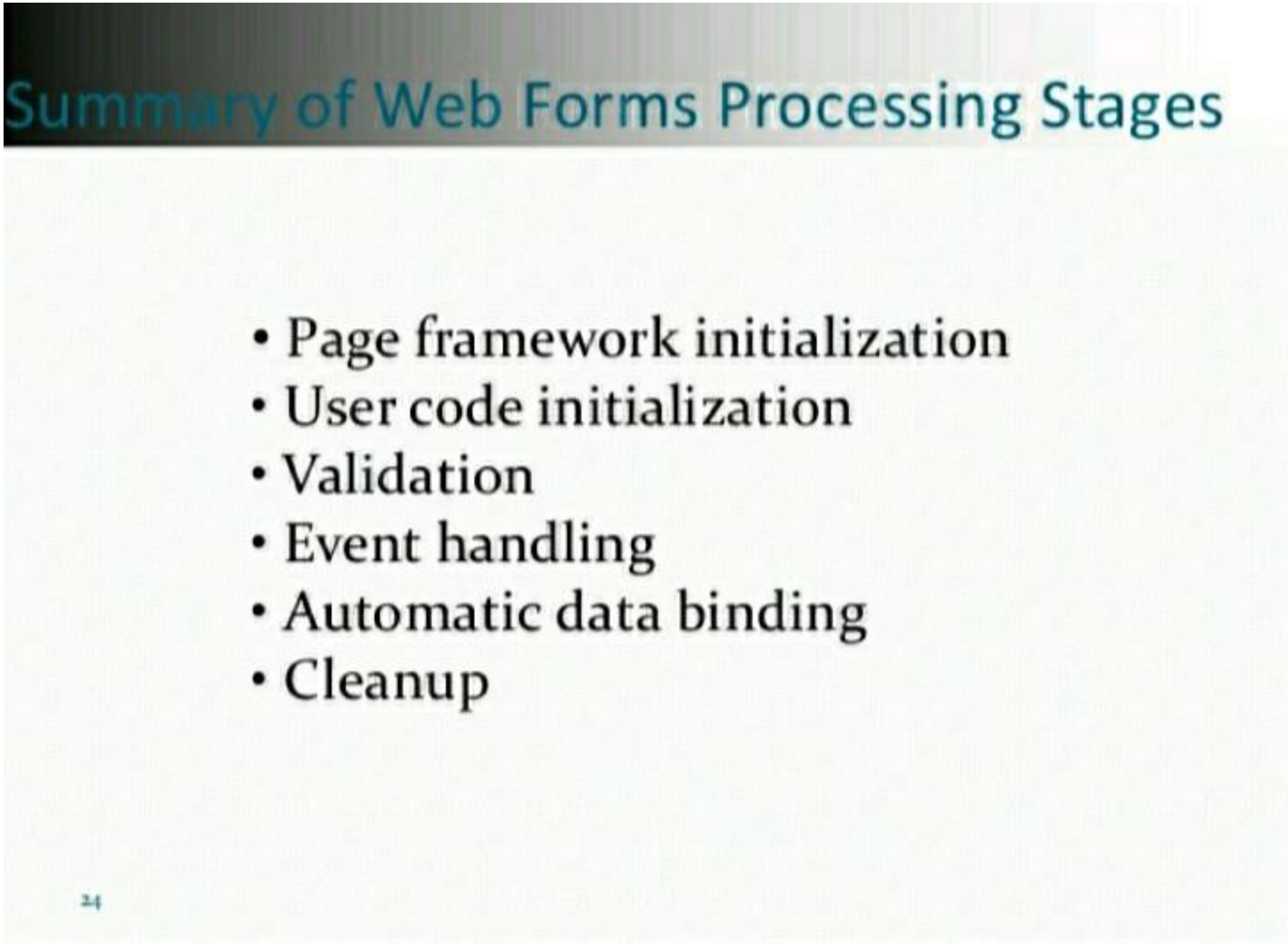
Submit



What Is a Web Form?

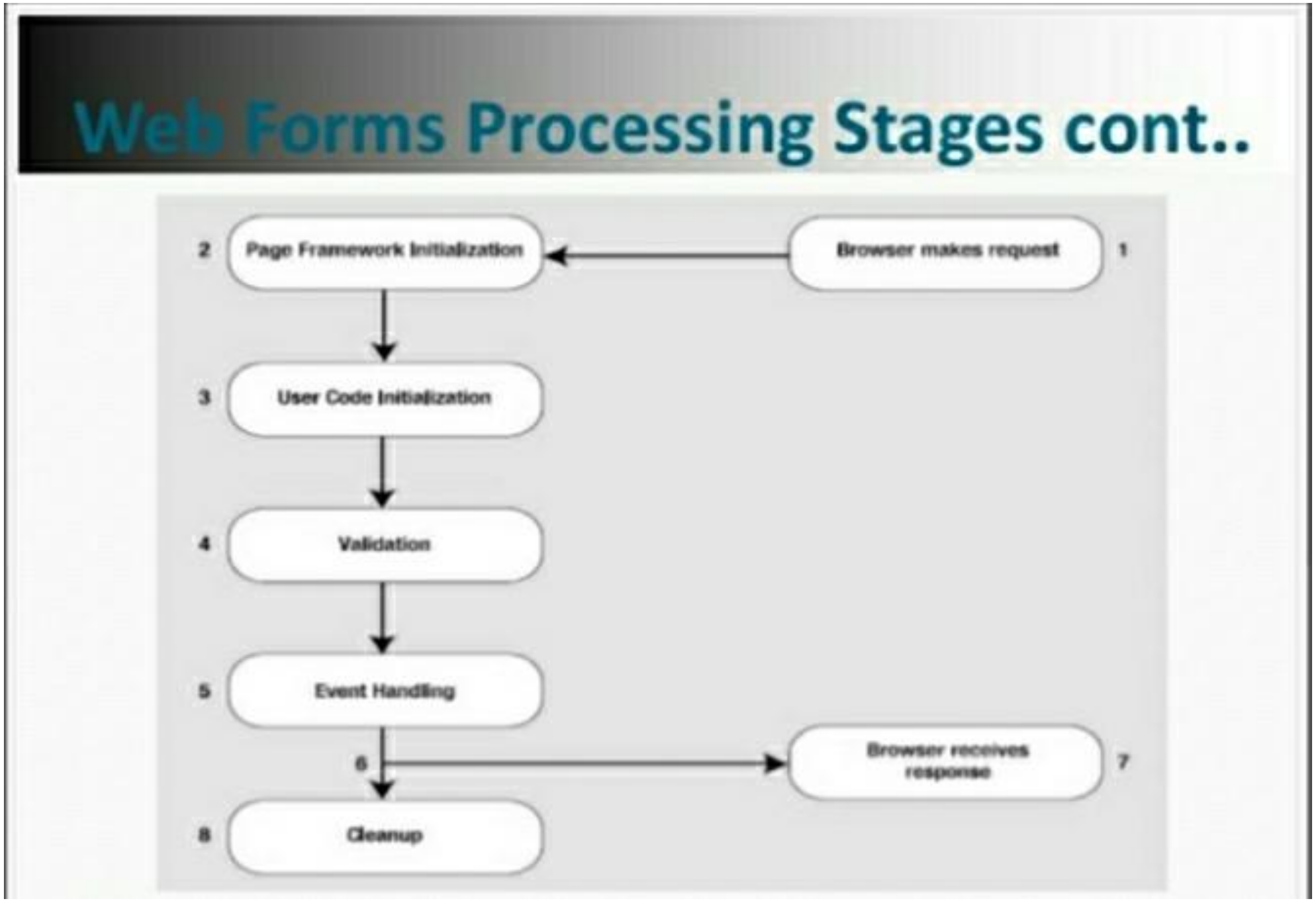
- .aspx extension
- Page attributes
  - **@ Page** directive
  - Controls save state
- Body attributes
- Form attributes

```
<%@ Page Language="C#" Inherits=Project.WebForm1 %>
<html>
   <body ms_positioning="GridLayout">
     <form id="Form1" method="post" runat="server">
     </form>
   </body>
</html>
```



Summary of Web Forms Processing Stages

- Page framework initialization
- User code initialization
- Validation
- Event handling
- Automatic data binding
- Cleanup



Web Forms Processing Stages cont..

1 Browser makes request

2 Page Framework Initialization

3 User Code Initialization

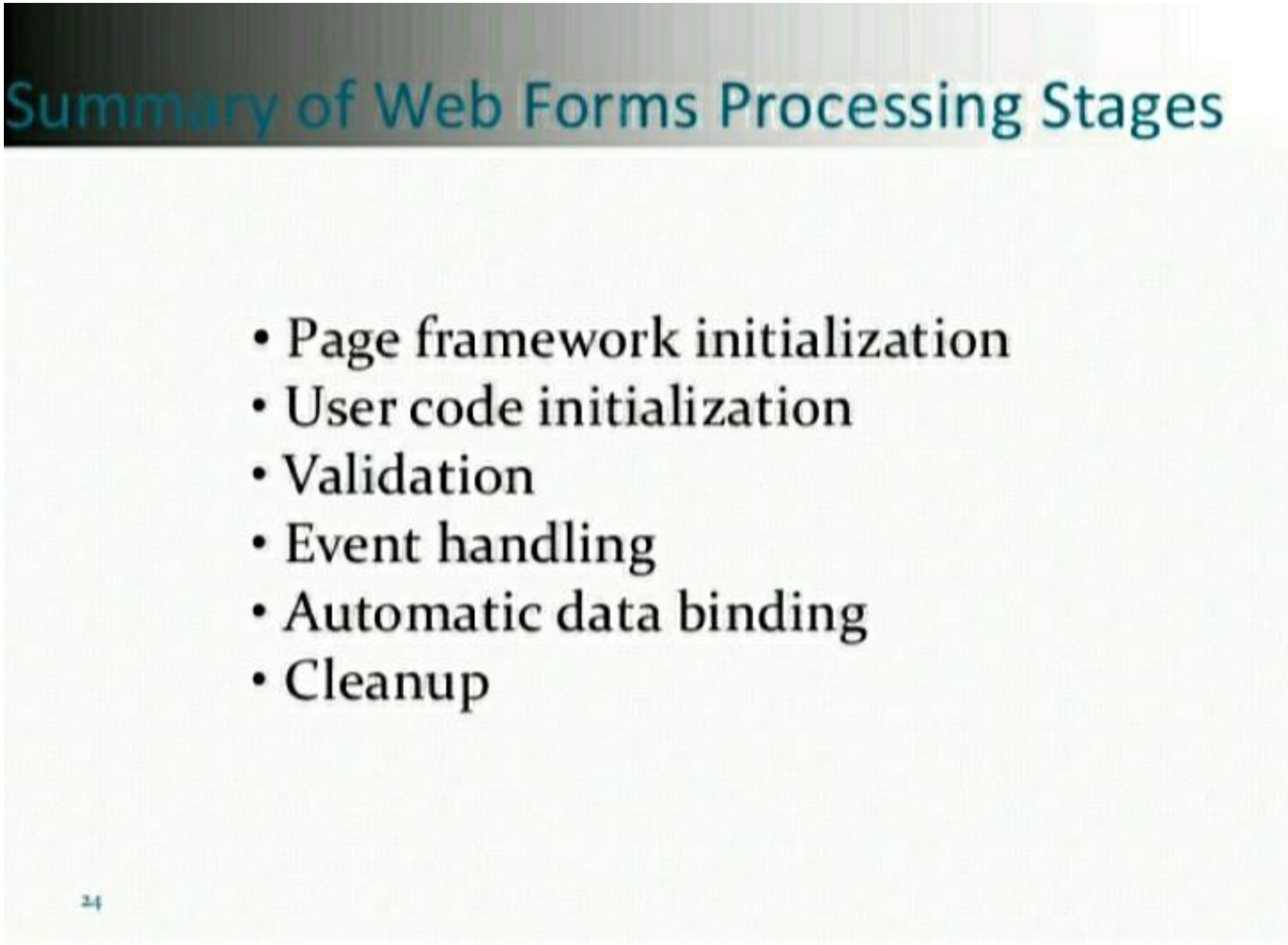
4 Validation

5 Event Handling

6

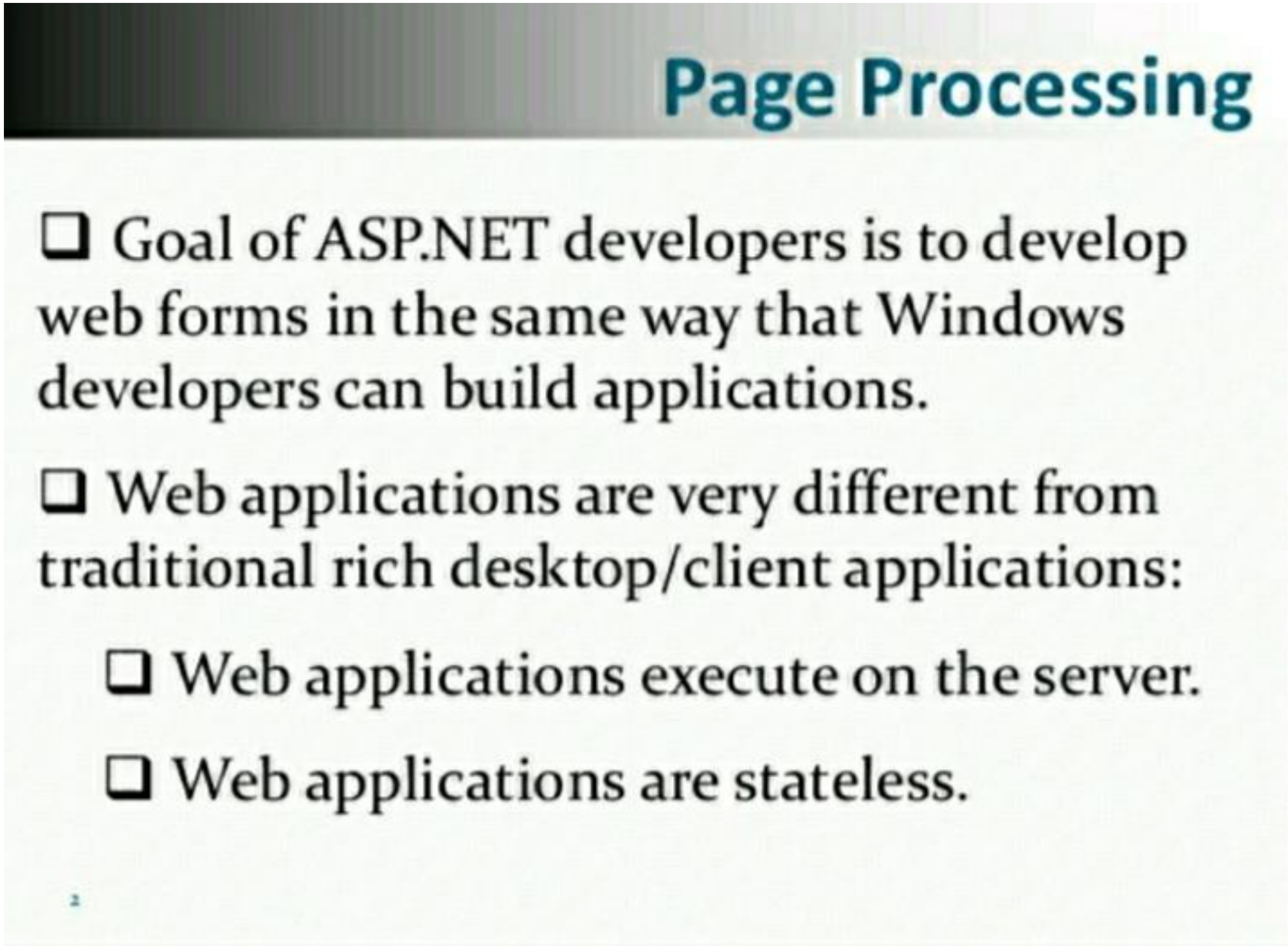
7 Browser receives response

8 Cleanup



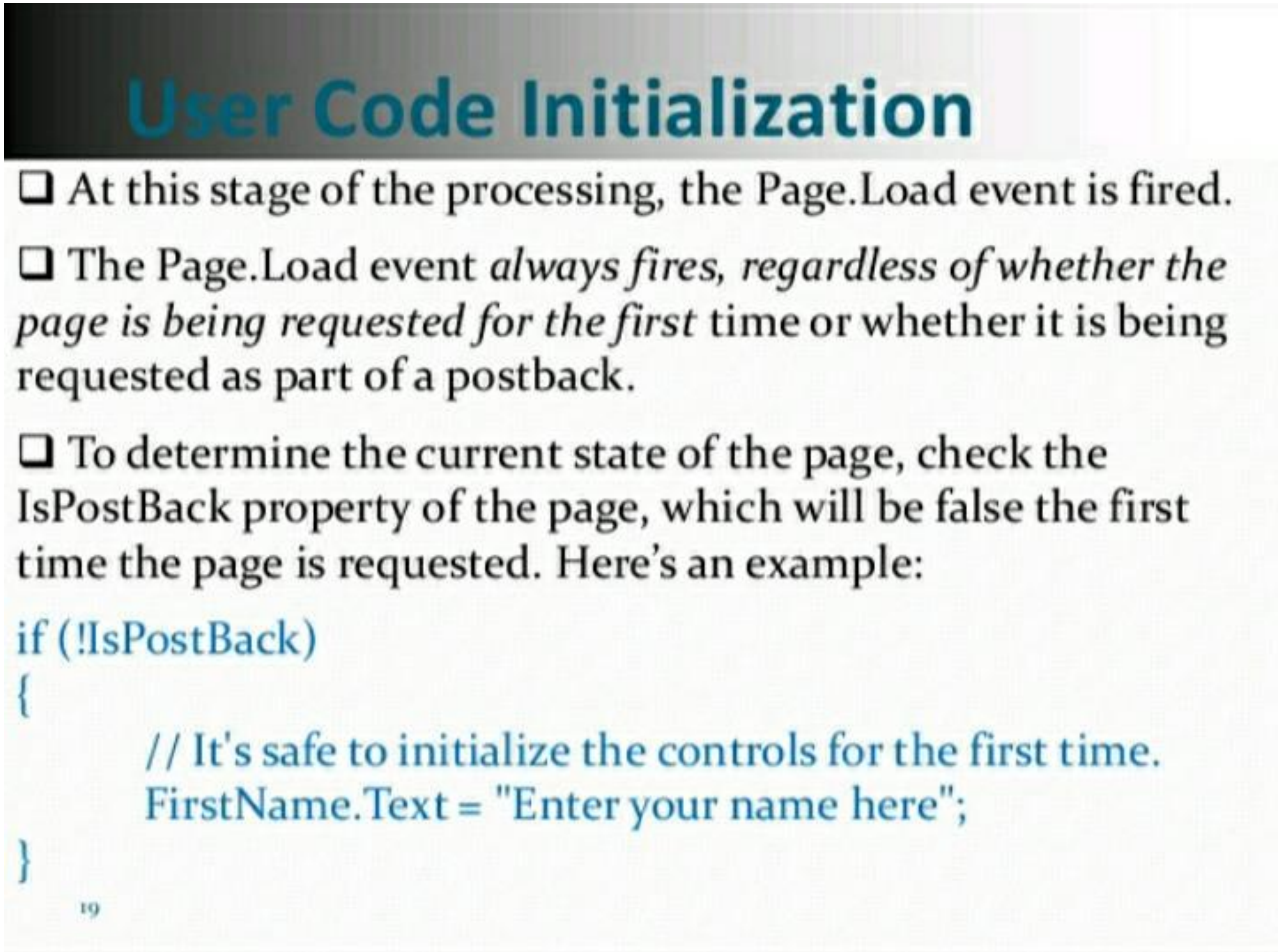
Summary of Web Forms Processing Stages

- Page framework initialization
- User code initialization
- Validation
- Event handling
- Automatic data binding
- Cleanup



Page Processing

- □ Goal of ASP.NET developers is to develop web forms in the same way that Windows developers can build applications.
- □ Web applications are very different from traditional rich desktop/client applications:
  - □ Web applications execute on the server.
  - □ Web applications are stateless.



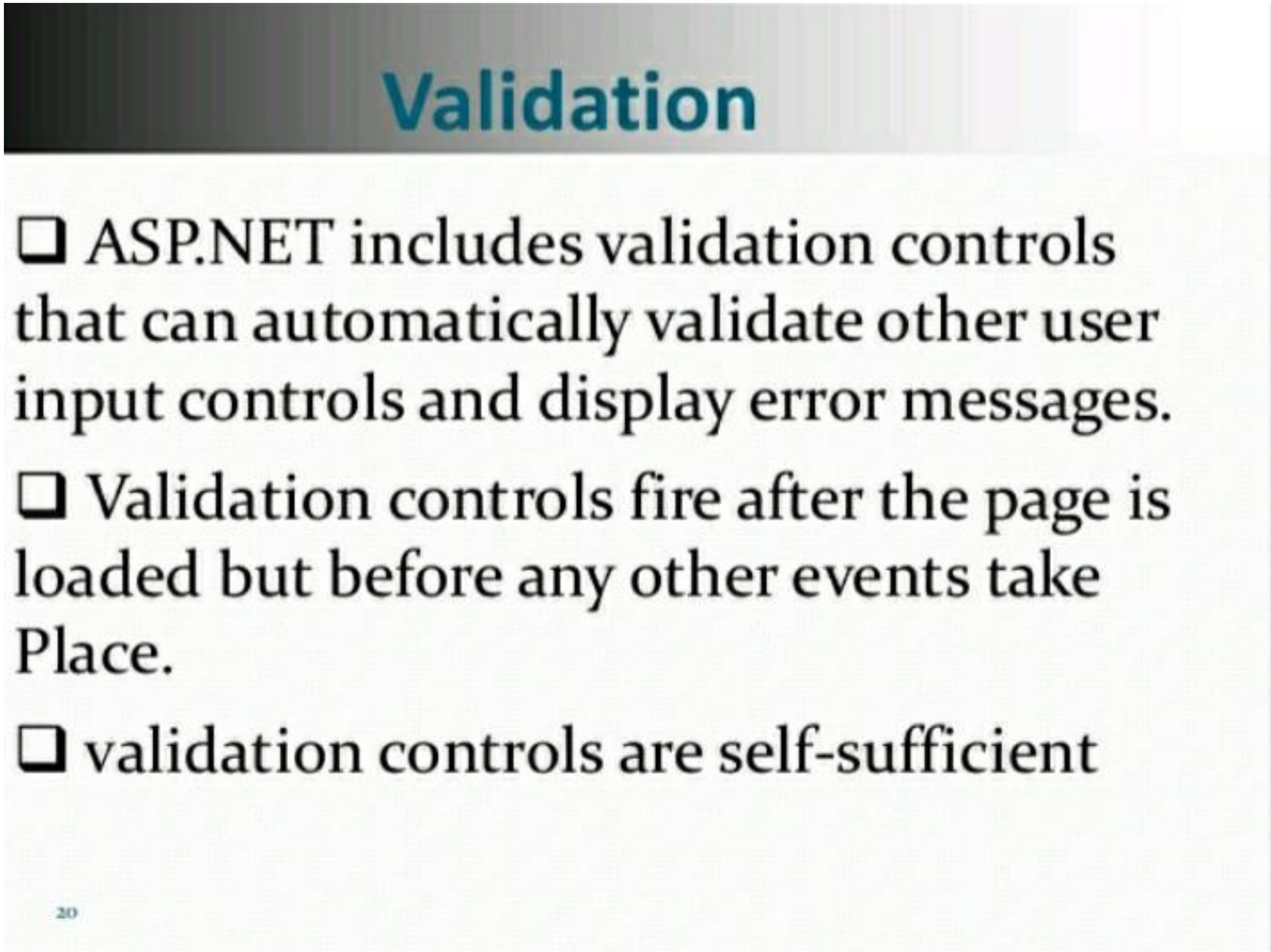
User Code Initialization

❑ At this stage of the processing, the Page.Load event is fired.

❑ The Page.Load event *always fires, regardless of whether the page is being requested for the first* time or whether it is being requested as part of a postback.

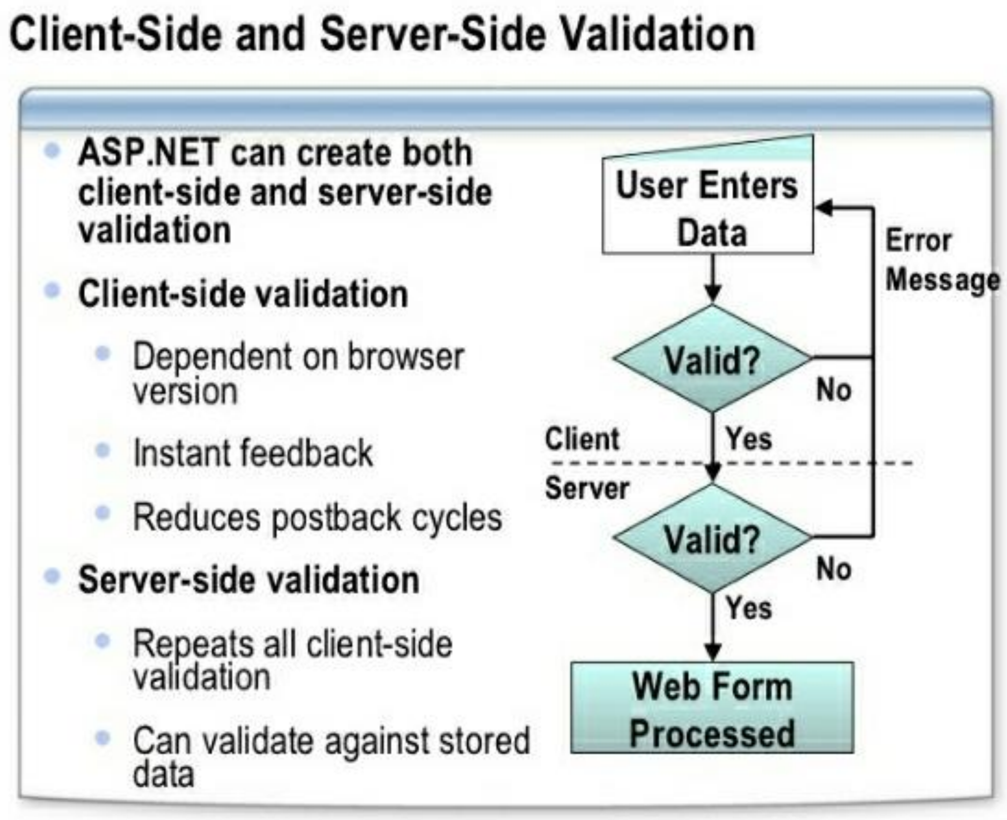
❑ To determine the current state of the page, check the IsPostBack property of the page, which will be false the first time the page is requested. Here's an example:

```
if (!IsPostBack)
{
        // It's safe to initialize the controls for the first time.
        FirstName.Text = "Enter your name here";
}
```



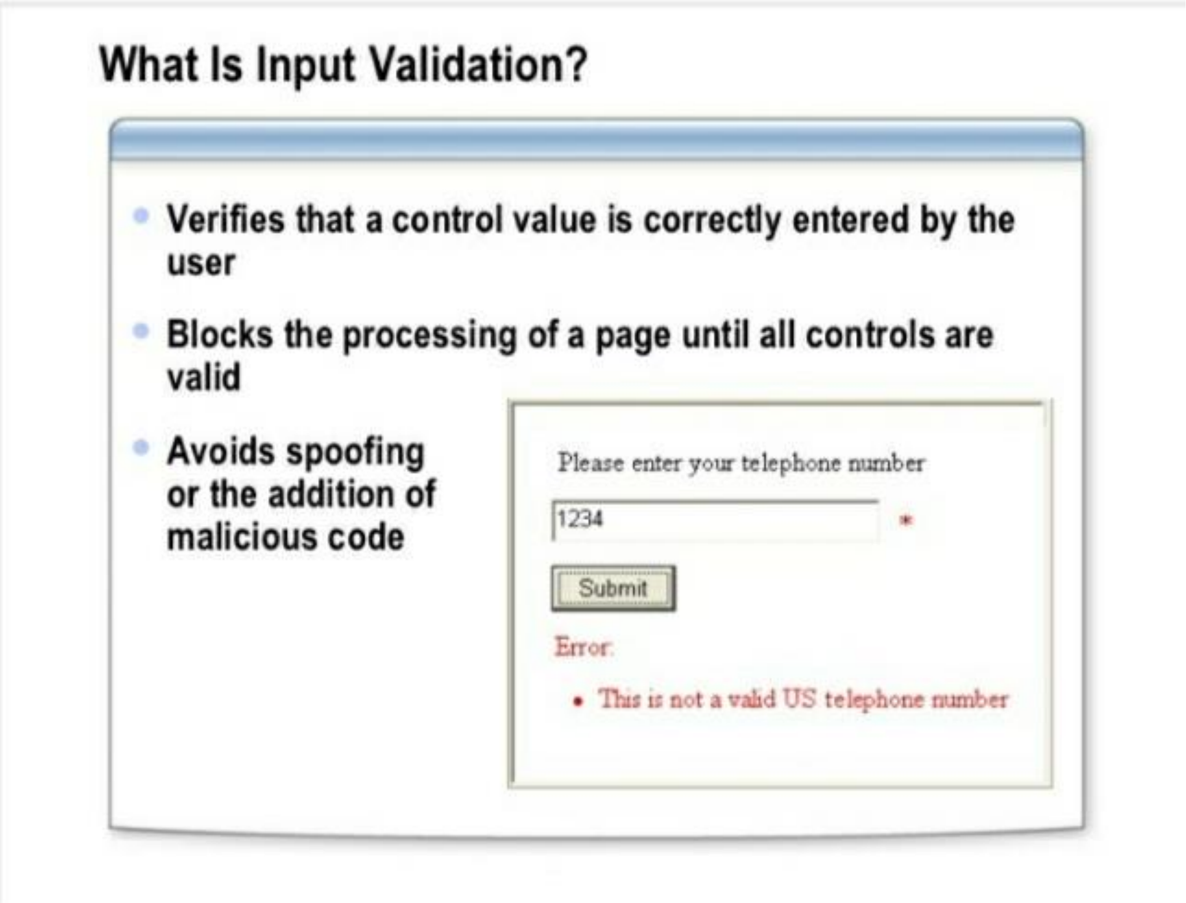
Validation

- ❑ ASP.NET includes validation controls that can automatically validate other user input controls and display error messages.
- ❑ Validation controls fire after the page is loaded but before any other events take Place.
- ❑ validation controls are self-sufficient



Client-Side and Server-Side Validation

- ASP.NET can create both client-side and server-side validation
- Client-side validation
  - Dependent on browser version
  - Instant feedback
  - Reduces postback cycles
- Server-side validation
  - Repeats all client-side validation
  - Can validate against stored data



What Is Input Validation?

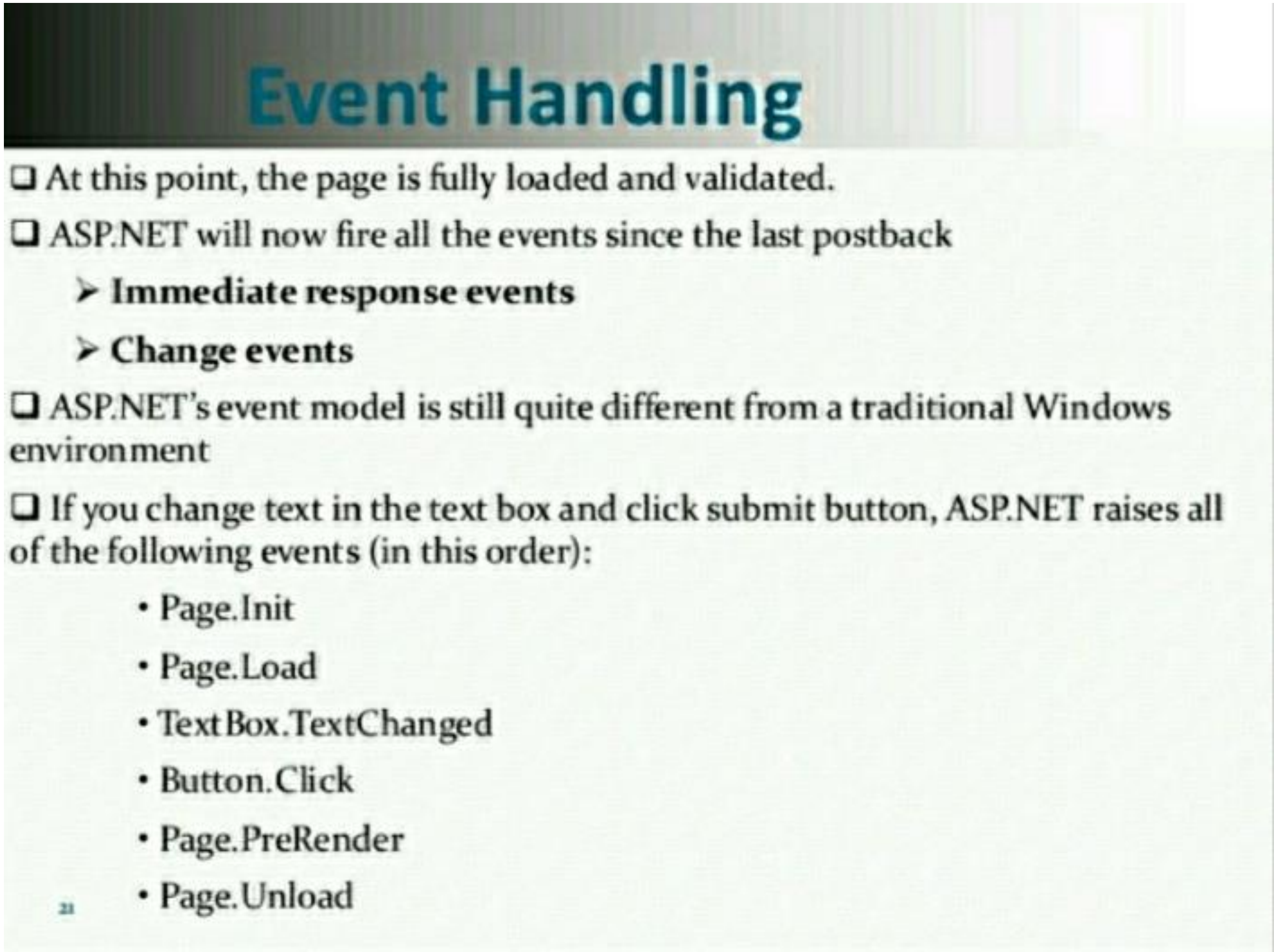
- Verifies that a control value is correctly entered by the user
- Blocks the processing of a page until all controls are valid
- Avoids spoofing or the addition of malicious code

Please enter your telephone number

1234 *

Error:

- This is not a valid US telephone number



Event Handling

❑ At this point, the page is fully loaded and validated.

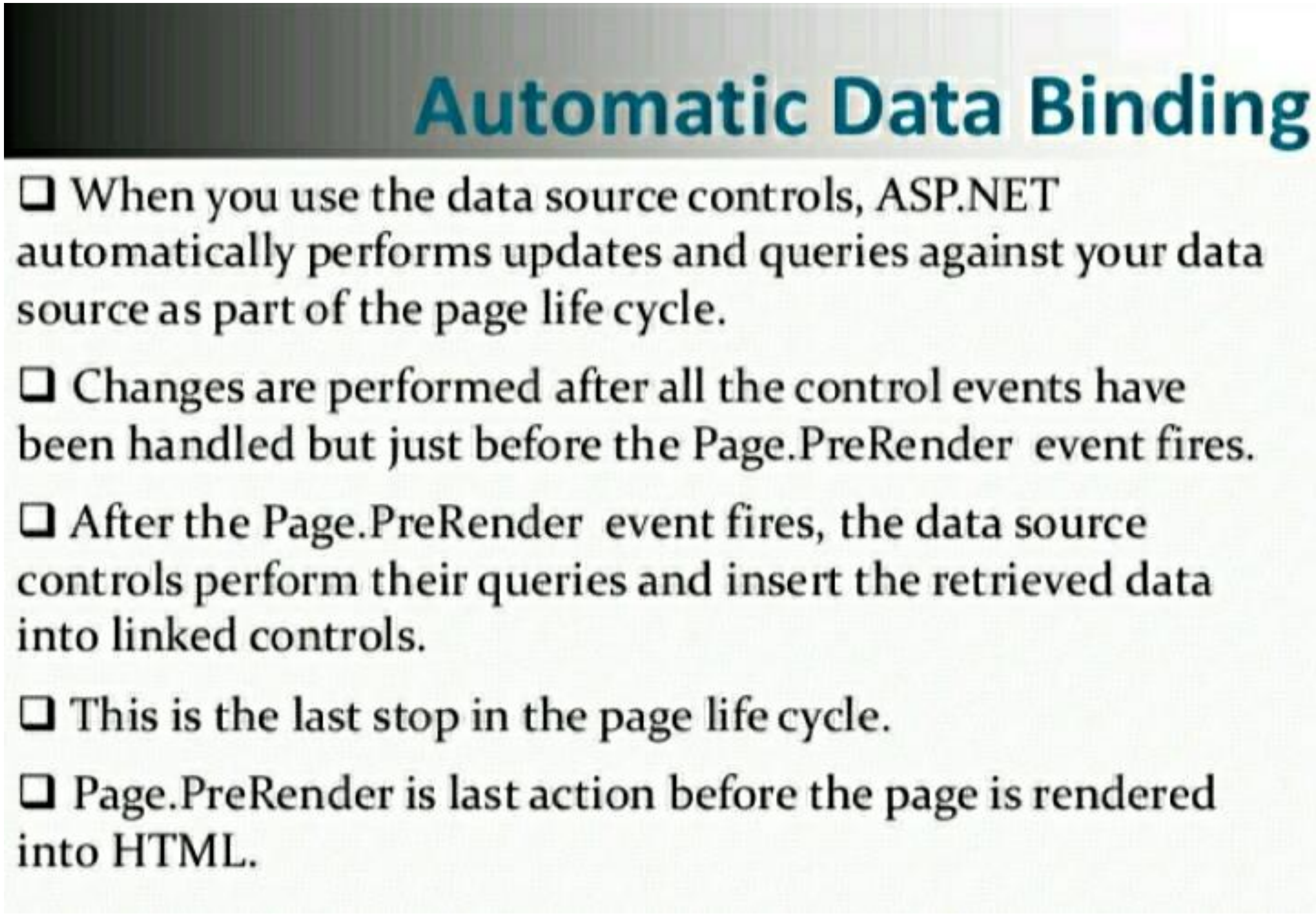
❑ ASP.NET will now fire all the events since the last postback

- ➢ **Immediate response events**
- ➢ **Change events**

❑ ASP.NET's event model is still quite different from a traditional Windows environment

❑ If you change text in the text box and click submit button, ASP.NET raises all of the following events (in this order):

- Page.Init
- Page.Load
- TextBox.TextChanged
- Button.Click
- Page.PreRender
- Page.Unload



Automatic Data Binding

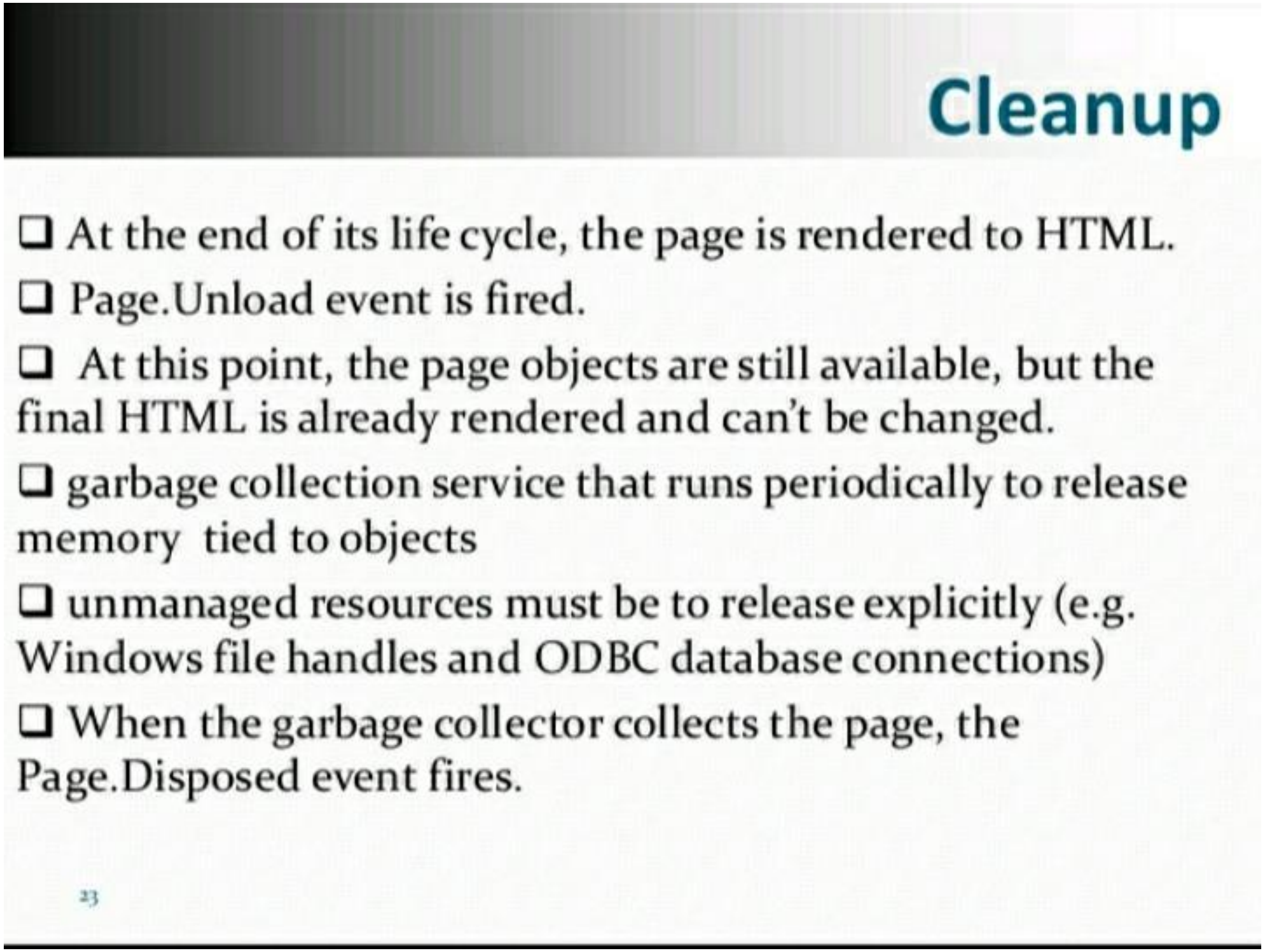
□ When you use the data source controls, ASP.NET automatically performs updates and queries against your data source as part of the page life cycle.

□ Changes are performed after all the control events have been handled but just before the Page.PreRender event fires.

□ After the Page.PreRender event fires, the data source controls perform their queries and insert the retrieved data into linked controls.

□ This is the last stop in the page life cycle.

□ Page.PreRender is last action before the page is rendered into HTML.



Cleanup

□ At the end of its life cycle, the page is rendered to HTML.

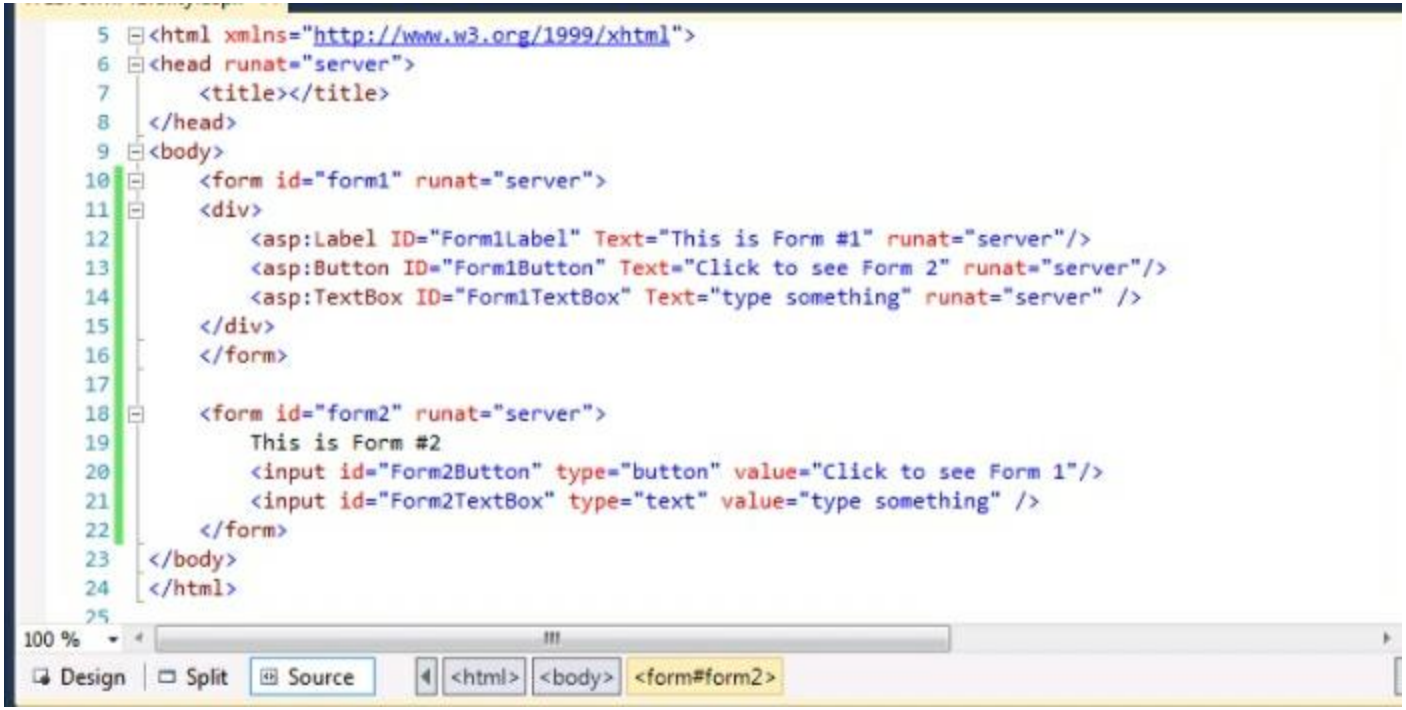
□ Page.Unload event is fired.

□ At this point, the page objects are still available, but the final HTML is already rendered and can't be changed.

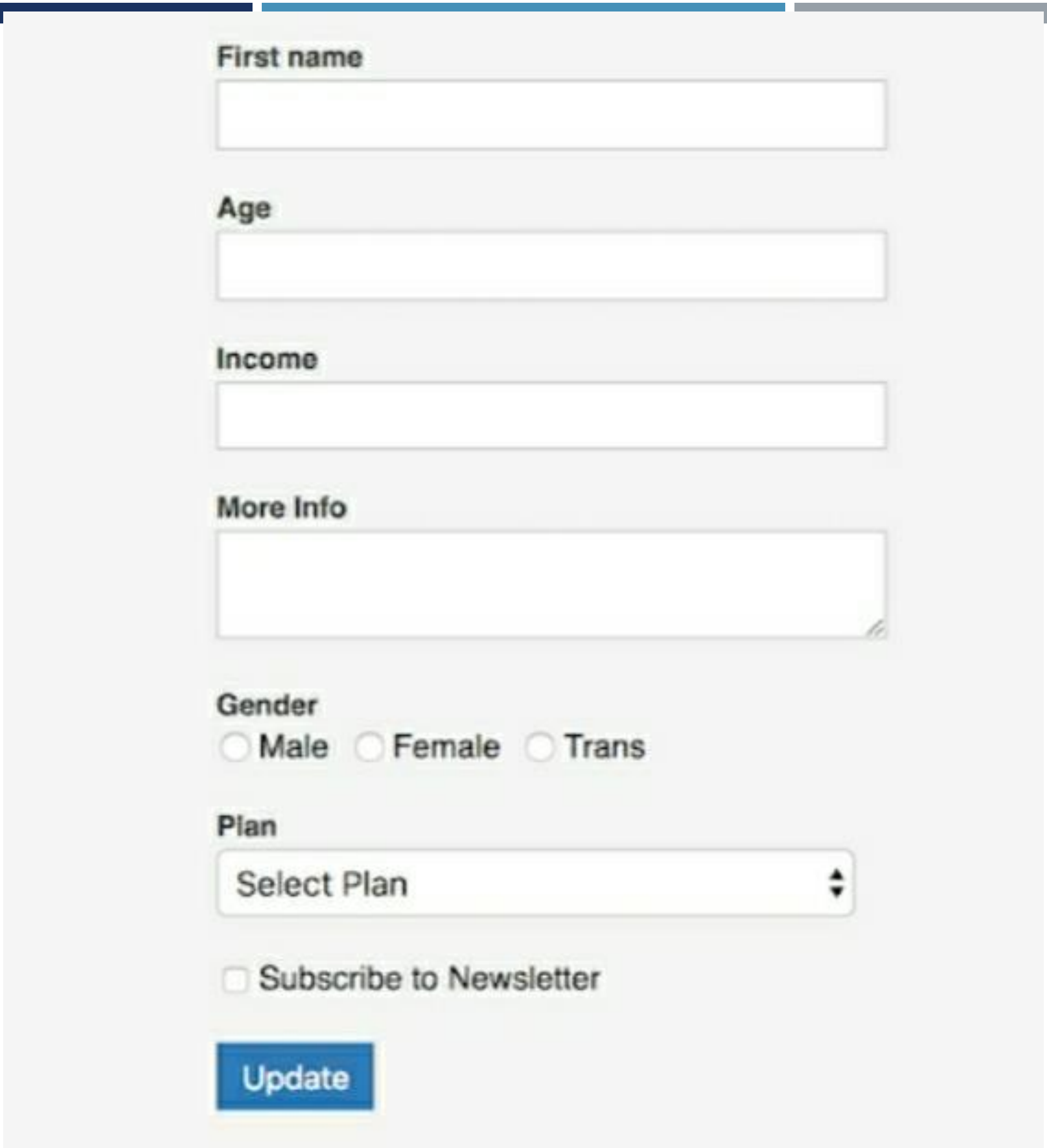
□ garbage collection service that runs periodically to release memory tied to objects

□ unmanaged resources must be to release explicitly (e.g. Windows file handles and ODBC database connections)

□ When the garbage collector collects the page, the Page.Disposed event fires.



```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
        <asp:Label ID="Form1Label" Text="This is Form #1" runat="server"/>
        <asp:Button ID="Form1Button" Text="Click to see Form 2" runat="server"/>
        <asp:TextBox ID="Form1TextBox" Text="type something" runat="server" />
    </div>
    </form>

    <form id="form2" runat="server">
        This is Form #2
        <input id="Form2Button" type="button" value="Click to see Form 1"/>
        <input id="Form2TextBox" type="text" value="type something" />
    </form>
</body>
</html>
```



First name

Age

Income

More Info

Gender

○ Male ○ Female ○ Trans

Plan

Select Plan

□ Subscribe to Newsletter

Update

SERVER CONTROLS

- Organized into logical families
  - HTML controls
  - Web control

SERVER CONTROLS
HTML CONTROLS

- Works well with existing HTML designers
- Properties map 1:1 with HTML
  - table.bgcolor ="red“;
- Can specify client-side event handlers
- Good when quickly converting existing pages
- Derived from System.Web.UI.HtmlControls.HtmlControl
- Supported controls have custom class, others derive from HtmlGenericControl

SERVER CONTROLS
HTML CONTROLS

- Supported controls
    - <a>
    - <img>
    - <form>
    - <table>
    - <tr>
    - <td>
    - <th>
    - <select>

<textarea>

<button>

<input type=text>

<input type=file>

<input type=submit>

<input type=button>

<input type=reset>

<input type=hidden>

SERVER CONTROLS
HTML CONTROLS

- Can use controls two ways:
  - Handle everything in action events (e.g. button click)
    - Event code will read the values of other controls (e.g. text, check boxes, radio buttons, select lists)
  - Handle change events as well as action events

SERVER CONTROLS
WEB CONTROLS

```
Label1.BackColor = Color.Red;
Table.BackColor = Color.Blue;
```

- Consistent object model

Richer functionality

E.g. AutoPostBack, additional methods

Strongly-typed; no generic control

Enables better compiler type checking

SERVER CONTROLS
WEB CONTROLS

- Web controls appear in HTML markup as namespaced tags
- Web controls have an asp: prefix

```
<asp:button onclick="button1_click‟ runat=server>
<asp:textbox onchaged="text1_changed‟ runat=server>
```

- Defined in the System.Web.UI.WebControls namespace
- This namespace is automatically mapped to the asp: prefix

SERVER CONTROLS
WEB CONTROLS

- Web Controls provide extensive properties to control display and format, e.g.
  - Font
  - BackColor, ForeColor
  - BorderColor, BorderStyle, BorderWidth
  - Style, CssClass
  - Height, Width
  - Visible, Enabled

SERVER CONTROLS
WEB CONTROLS

- Four types of Web Controls
  - Intrinsic controls
  - List controls
  - Rich controls
  - Validation controls

SERVER CONTROLS
INTRINISIC CONTROLS

- Correspond to HTML controls
- Supported controls
  - <asp:button>
  - <asp:imagebutton>
  - <asp:linkbutton>
  - <asp:hyperlink>
  - <asp:textbox>
  - <asp:checkbox>

```
<asp:radiobutton>
<asp:image>
<asp:label>
<asp:panel>
<asp:table>
```

SERVER CONTROLS
INTRINISIC CONTROLS

- TextBox, ListControl, CheckBox and their subclasses don’t automatically do a postback when their controls are changed
- Specify AutoPostBack=true to make change events cause a postback

SERVER CONTROLS
LIST CONTROLS

```
<asp:dropdownlist>
<asp:listbox>
<asp:radiobuttonlist>
<asp:checkboxlist>
<asp:repeater>
<asp:datalist>
<asp:datagrid>
```

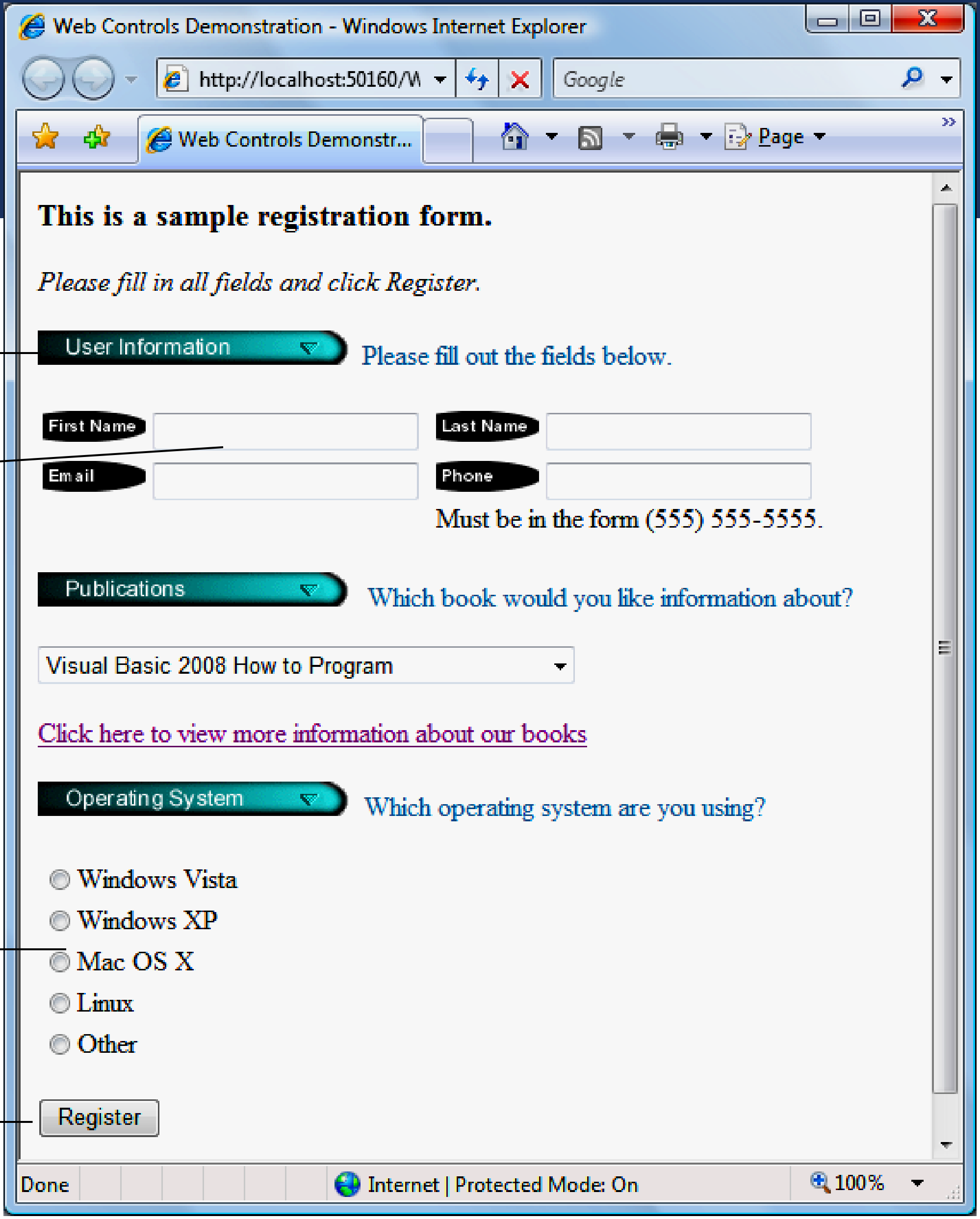
- Controls that handle repetitionSupported controls

- `Repeater`, `DataList` and `DataGrid` controls expose templates for customization
  - More about these controls and templates later

SERVER CONTROLS
CHECKBOXLIST & RADIOBUTTONLIST

```
<asp:CheckBoxList id=Check1 runat="server">
  <asp:ListItem>Item 1</asp:ListItem>
  <asp:ListItem>Item 2</asp:ListItem>
  <asp:ListItem>Item 3</asp:ListItem>
  <asp:ListItem>Item 4</asp:ListItem>
  <asp:ListItem>Item 5</asp:ListItem>
</asp:CheckBoxList>
```

WebControls3 Example



Web Controls Demonstration - Windows Internet Explorer

http://localhost:50160/W

This is a sample registration form.

Please fill in all fields and click Register.

Please fill out the fields below.

First Name:

Last Name:

Email:

Phone:

Must be in the form (555) 555-5555.

Which book would you like information about?

Visual Basic 2008 How to Program

Click here to view more information about our books

Which operating system are you using?

○ Windows Vista
○ Windows XP
○ Mac OS X
○ Linux
○ Other

Register

Done | Internet | Protected Mode: On | 100%

Image control

TextBox control

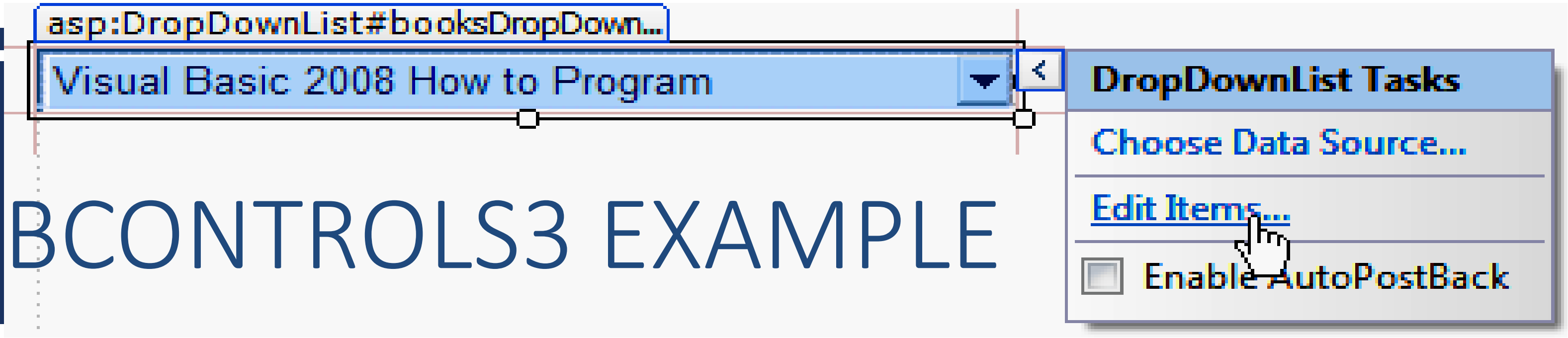
DropDownList control

HyperLink control

RadioButtonList

Button control

WEBCONTROLS3 EXAMPLE



- A DropDownList does not allow users to type text.
- Each item in the drop-down list is defined by a ListItem element.
- You can add items to a DropDownList using the ListItem Collection Editor. This process is similar to customizing a ListBox in a Windows application.
- Visual Studio displays smart-tag menus for many ASP.NET controls to facilitate common tasks.

WEBCONTROLS3 EXAMPLE

- Add a HyperLink control.
- The NavigateUrl property of this control specifies the resource that is requested when a user clicks the hyperlink.
- Setting the Target property to _blank specifies that the requested web page should open in a new window or tab.
- The RadioButtonList control provides a series of radio buttons from which the user can select only one.
- Like options in a DropDownList, individual radio buttons are defined by ListItem elements.
- A Button web control represents a button that triggers an action when clicked, and typically maps to an XHTML input element of type "button".

Simple form for gathering user input

WebControls.aspx

(1 of 6)

```
<%-- WebControls3.aspx --%>
<%-- Registration form that demonstrates web controls. --%>
<%@ Page Language="C#" AutoEventWireup="true"
   CodeFile="WebControls3.aspx.cs" Inherits="WebControls3" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>Web Controls Demonstration</title>
   <style type="text/css">
      .style1
      {
         color: #006699;
      }
```

In the head element of your .aspx file, the **style** element defines **embedded style sheets**.

Web Form that demonstrates web controls. (Part 1 of 6.)

```
      .style2
      {
         width: 100%;
      }
   </style>
</head>
<body>
   <form id="form1" runat="server">
   <div>
      <h3>This is a sample registration form.</h3>
      <p><i>Please fill in all fields and click Register.</i></p>
      <p>
         <asp:Image ID="userInformationImage" runat="server"
            ImageUrl="~/Images/user.png" />
         <span class="style1">Please fill out the fields below.</sp
      </p>
      <table class="style2">
         <tr>
            <td valign="top" style="width: 225px">
               <asp:Image ID="firstNameImage" runat="server"
                  ImageUrl="~/Images/fname.png" />
               <asp:TextBox ID="firstNameTextBox" runat="server">
               </asp:TextBox>
            </td>
```

In the `head` element of your `.aspx` file, the **style** element defines **embedded style sheets**.

An `Image` control inserts an image into a web page.

A **TextBox** control allows you to obtain text from the user and display text to the user.

Fig. | Web Form that demonstrates web controls. (Part 2 of 6.)

```
      <p>
         <asp:Image ID="publicationsImage" runat="server"
            ImageUrl="~/Images/publications.png" />&nbsp;
         <span class="style1">
            Which book would you like information about?</span>
      </p>
      <p>
         <asp:DropDownList ID="booksDropDownList" runat="server">
            <asp:ListItem>Visual Basic 2008 How to Program</asp:ListItem>
            <asp:ListItem>Visual C# 2008 How to Program</asp:ListItem>
            <asp:ListItem>Java How to Program 6e</asp:ListItem>
            <asp:ListItem>C++ How to Program 5e</asp:ListItem>
            <asp:ListItem>C How to Program 5e</asp:ListItem>
            <asp:ListItem>Internet and World Wide Web How to Program
               </asp:ListItem>
         </asp:DropDownList>
      </p>
      <p>
         <asp:HyperLink ID="booksHyperLink" runat="server"
            NavigateUrl="http://www.deitel.com" Target="_blank">
            Click here to view more information about our books
         </asp:HyperLink>
      </p>
```

The **DropDownList** control provides a list of values from which t he user can select only one.

The **HyperLink** c ontrol adds a hyperl ink to a web page.

Fig. | Web Form that demonstrates web controls. (Part 4 of 6.)

```
        <p>
           <asp:Image ID="operatingSystemImage" runat="server"
              ImageUrl="~/Images/os.png" />&nbsp;
           <span class="style1">Which operating system are you using?
              </span>
        </p>
        <p>
           <asp:RadioButtonList ID="operatingSystemRadioButtonList"
              runat="server">
              <asp:ListItem>Windows Vista</asp:ListItem>
              <asp:ListItem>Windows XP</asp:ListItem>
              <asp:ListItem>Mac OS X</asp:ListItem>
              <asp:ListItem>Linux</asp:ListItem>
              <asp:ListItem>Other</asp:ListItem>
           </asp:RadioButtonList>
        </p>
        <p>
           <asp:Button ID="registerButton" runat="server"
              Text="Register" />
        </p>
     </div>
     </form>
</body>
</html>
```

The `RadioButtonList` control provides a series of radio buttons from which the user can select only one.

A `Button` web control represents a button that triggers an action when clicked, and typically maps to an XHTML `input` element of `type "button"`.

Fig. | Web Form that demonstrates web controls. (Part 5 of 6.)

SERVER CONTROLS
RICH CONTROLS

- ASP.NET provides large set of controls. These controls are divided into different categories, depends upon their functionalities. The followings control comes under the rich controls category.
  - FileUpload control
  - Calendar control
  - AdRotator control
  - MultiView control
  - Wizard control

FILE UPLOAD CONTROL:

- FileUpload control is used to browse and upload files. After the file is uploaded, you can store the file on any drive or database. FileUpload control is the combination of a browse button and a text box for entering the filename.

FILE UPLOAD CONTROL:

- The FileUpload control supports the following important properties.
- FileBytes: It returns the contents of uploaded file as a byte array
- FileContent: You can get the uploaded file contents as a stream.
- FileName: Provides the name of uploaded file.
- HasFile: It is a Boolean property that checks whether particular file is available or not.
- PostedFile: Gets the uploaded file wrapped in the HttpPostedFile object.

CALENDAR CONTROL:

- Calendar control provides you lots of property and events. By using these properties and events you can perform the following task with calendar control.
-
- Select date.
- Selecting a day, a week or a month.
- Customize the calendar's appearance.

THE CALENDAR CONTROL SUPPORTS THREE IMPORTANT EVENTS:

- SelectionChanged-This event is fired when you select a day, a week or an entire month.
- DayRender-This event is fired when each data cell of the calendar control is rendered.
- VisibleMonthChanged -It is raised when user changes a month.
-
- Calendar control supports SelectionMode property that allows you to select a single day, week, or entire month.

ADROTATOR CONTROL:

- AdRotator control is used to display different advertisements randomly in a page. The list of advertisements is stored in either an XML file or in a database table. Lots of websites uses AdRotator control to display the advertisements on the web page.

Important properties of AdRotator control:

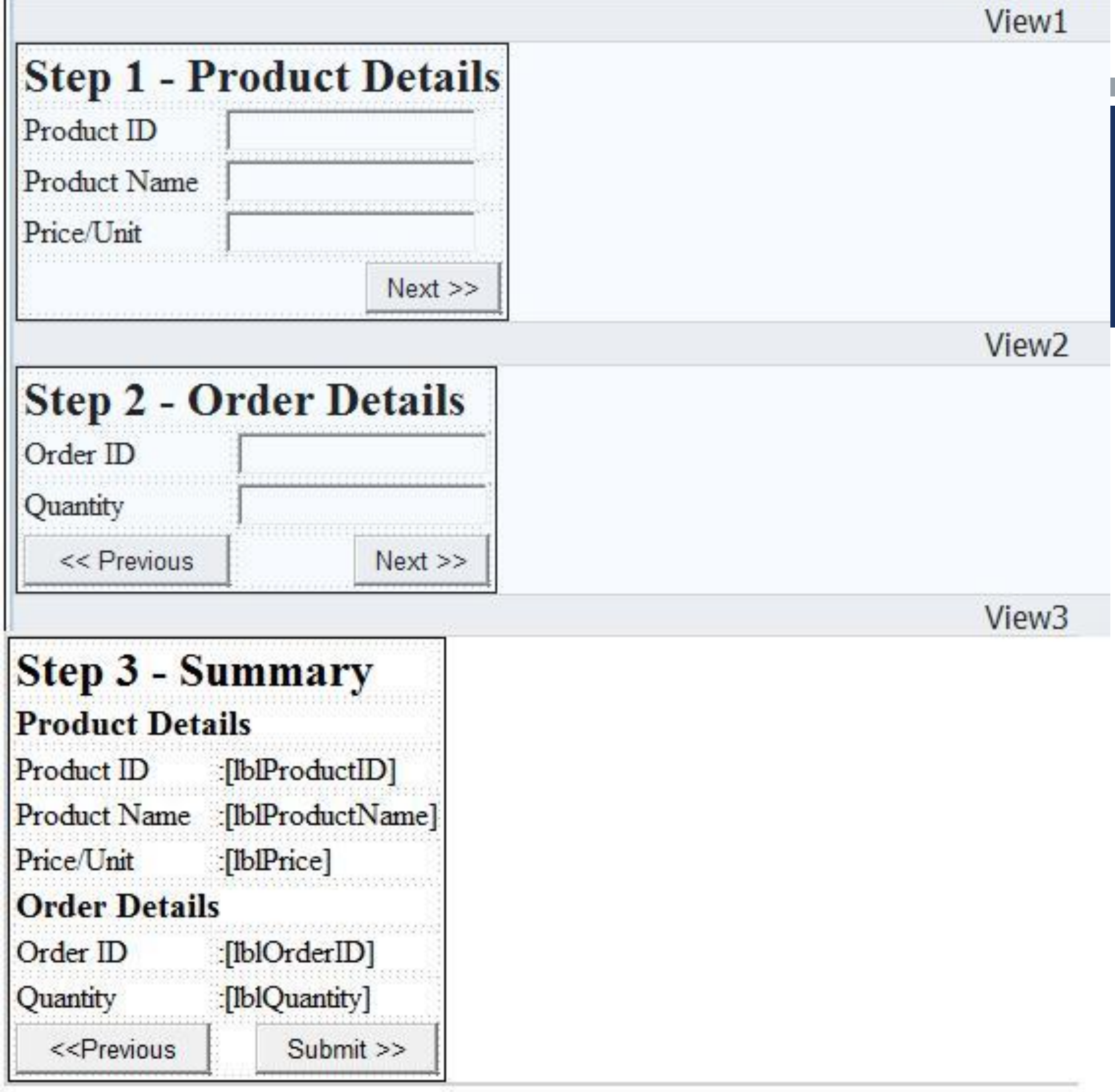
ImageUrl: The URL of the image that will be displayed through AdRotator control.
NavigateUrl: If the user clicks the banner or ad then the new page is opened according to given URL.
AlternateText: It is used for displaying text instead of the picture if picture is not displayed. It is also used as a tooltip.
Impressions: It is a number that sets how frequently an advertisement will appear.
Keyword: It is used to filter ads or identifies a group of advertisement

MultiView control:

MultiView control can be used when you want to create a tabbed page. In many situations, a web form may be very long, and then you can divide a long form into multiple sub forms. MultiView control is made up of multiple view controls. You can put multiple ASP.NET controls inside view controls. One View control is displayed at a time and it is called as the active view. View control does not work separately. It is always used with a Multiview control.

example, in Multiview control, we have taken three separ ate View control.

1. In First step we will design to capture Product details
2. In Second step we will design to capture Order details
3. Next we will show summary for confirmation.



View1

Step 1 - Product Details

Product ID

Product Name

Price/Unit

Next >>

View2

Step 2 - Order Details

Order ID

Quantity

<< Previous

Next >>

View3

Step 3 - Summary

Product Details

Product ID	:[lblProductID]
Product Name	:[lblProductName]
Price/Unit	:[lblPrice]

Order Details

Order ID	:[lblOrderID]
Quantity	:[lblQuantity]

<<Previous

Submit >>

Wizard Control:

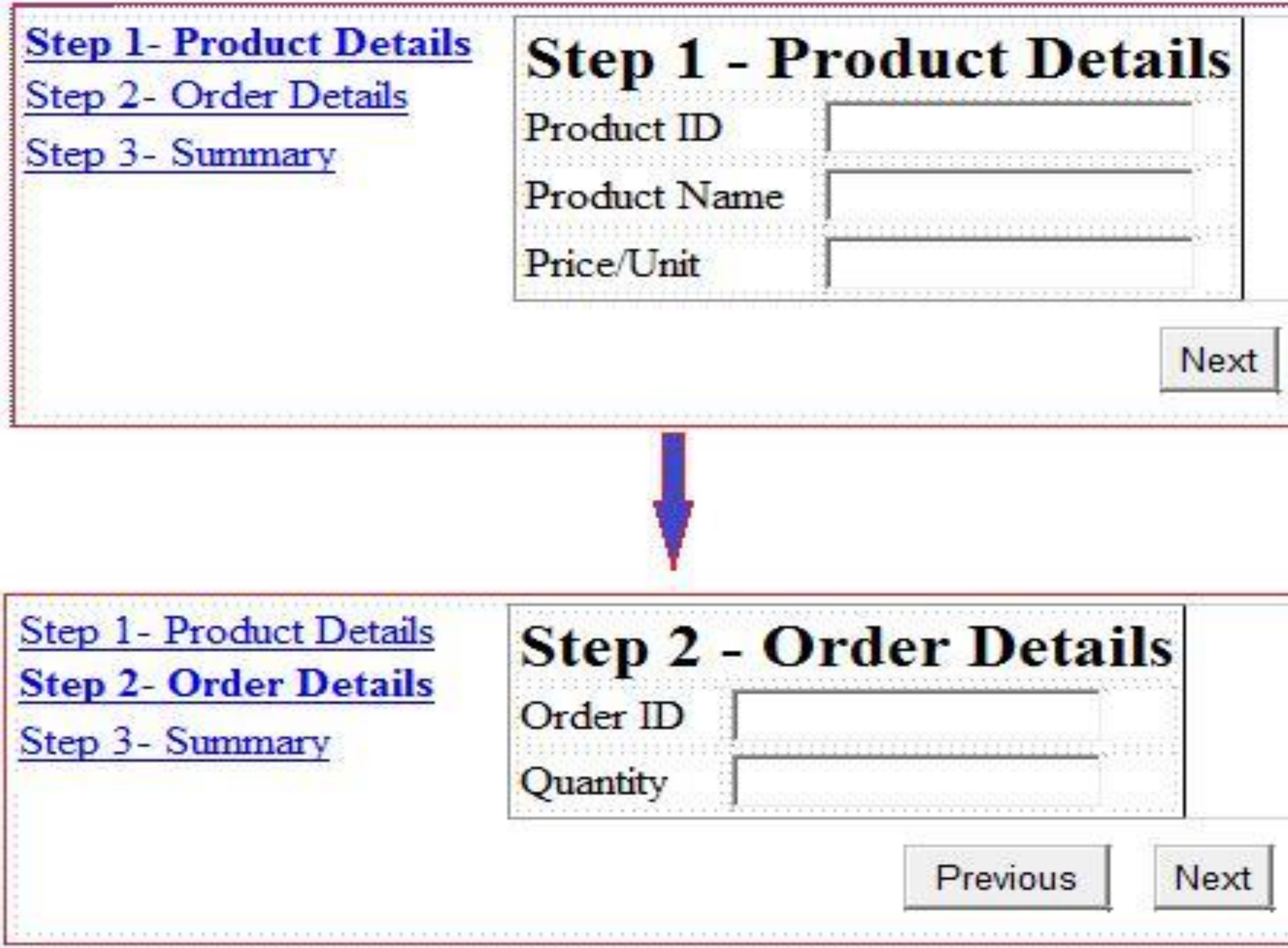
This control is same as MultiView control but the main difference is that, it has inbuilt navigation buttons.

The wizard control enables you to design a long form in such a way that you can work in multiple sub form. You can perform the task in a step by step process. It reduces the work of developers to design multiple forms. It enables you to create multi step user interface. Wizard control provides with built-in previous/next functionality.

The Wizard control can contains one or more WizardStep as child controls. Only one WizardStep is displayed at a ti me. WizardStep control has an important property called as StepType.

The StepTypes are:

Start
Step
Finish
Complete
Auto


Step 1- Product Details
Step 2- Order Details
Step 3- Summary
Step 1 - Product Details
Product ID
Product Name
Price/Unit
Next
Step 1- Product Details
Step 2- Order Details
Step 3- Summary
Step 2 - Order Details
Order ID
Quantity
Previous
Next

ASP.NET validation controls validate the user input data to ensure that useless, unauthenticated, or contradictory data don't get stored.

ASP.NET provides the following validation controls:

RequiredFieldValidator
RangeValidator
CompareValidator
RegularExpressionValidator
CustomValidator
ValidationSummary

RequiredFieldValidator Control

The RequiredFieldValidator control ensures that the required field is not empty. It is generally tied to a text box to force input into the text box.

The syntax of the control is as given:

```
<asp:RequiredFieldValidator ID="rfvcandidate"
   runat="server" ControlToValidate ="ddlcandidate"
   ErrorMessage="Please choose a candidate"
   InitialValue="Please choose a candidate">
</asp:RequiredFieldValidator>
```

RangeValidator Control

The RangeValidator control verifies that the input value falls within a predetermined range.

It has three specific properties:

Type-It defines the type of the data. The available values are: Currency, Date, Double, Integer, and String.
MinimumValue-It specifies the minimum value of the range.
MaximumValue-   It specifies the maximum value of the range.
The syntax of the control is as given:

```
<asp:RangeValidator ID="rvclass" runat="server" ControlToValidate="txtclass"
   ErrorMessage="Enter your class (6 - 12)" MaximumValue="12"
   MinimumValue="6" Type="Integer">

</asp:RangeValidator>Example
```

The following example describes a form to be filled up by all the students of a school, divided into four houses, for electing the school president. Here, we use the validation controls to validate the user input.
This is the form in design view:
form in Design view

CompareValidator Control:

The CompareValidator control compares a value in one control with a fixed value or a value in another control.

It has the following specific properties:

Properties	Description
Type	It specifies the data type.
ControlToCompare	It specifies the value of the input control to compare with.
ValueToCompare	It specifies the constant value to compare with.
Operator	It specifies the comparison operator, the available values are: Equal, NotEqual, GreaterThan, GreaterThanEqual, LessThan, LessThanEqual, and DataTypeCheck.

The basic syntax of the control is as follows:

```
<asp:CompareValidator ID="CompareValidator1" runat="server"
   ErrorMessage="CompareValidator">

</asp:CompareValidator>
```

RegularExpressionValidator

The RegularExpressionValidator allows validating the input text by matching against a pattern of a regular expression. The regular expression is set in the ValidationExpression property.

The following table summarizes the commonly used syntax constructs for regular expressions:

Character Escapes	Description
\b	Matches a backspace.
\t	Matches a tab.
\r	Matches a carriage return.
\v	Matches a vertical tab.
\f	Matches a form feed.
\n	Matches a new line.
\	Escape character.

The syntax of the control is as given:

```
<asp:RegularExpressionValidator ID="string" runat="server" ErrorMessage="string"
   ValidationExpression="string" ValidationGroup="string">

</asp:RegularExpressionValidator>
```

ValidationSummary:

The ValidationSummary control does not perform any validation but shows a summary of all errors in the page. The summary displays the values of the ErrorMessage property of all validation controls that failed validation.

The following two mutually inclusive properties list out the error message:

ShowSummary : shows the error messages in specified format.

ShowMessageBox : shows the error messages in a separate window.

The syntax for the control is as given:

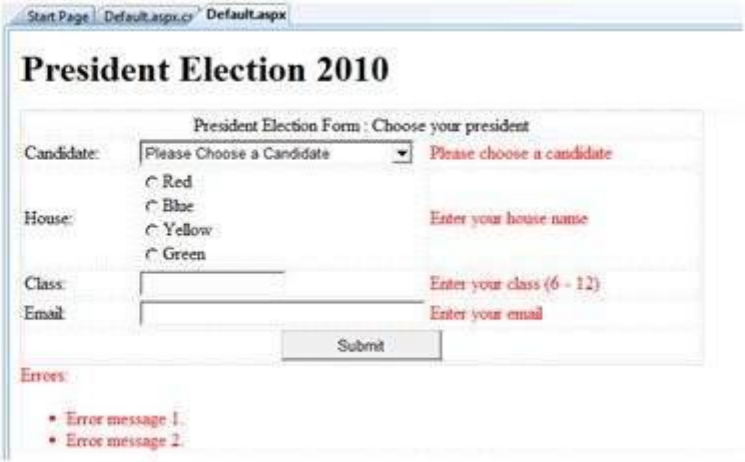
```
<asp:ValidationSummary ID="ValidationSummary1" runat="server"
   DisplayMode = "BulletList" ShowSummary
```

Validation Groups:

Complex pages have different groups of information provided in different panels. In such situation, a need might arise for performing validation sep arately for separate group. This kind of situation is handled using validati on groups.

To create a validation group, you should put the input controls and the val idation controls into the same logical group by setting their ValidationGro up property.

Example
The following example describes a form to be filled up by all the students of a school, divided into four houses, for electing the school president. Here, we use the validation controls to validate the user input.



President Election 2010

	President Election Form : Choose your president	
Candidate:	Please Choose a Candidate	Please choose a candidate
House:	○ Red ○ Blue ○ Yellow ○ Green	Enter your house name
Class:		Enter your class (6 - 12)
Email:		Enter your email
	Submit	

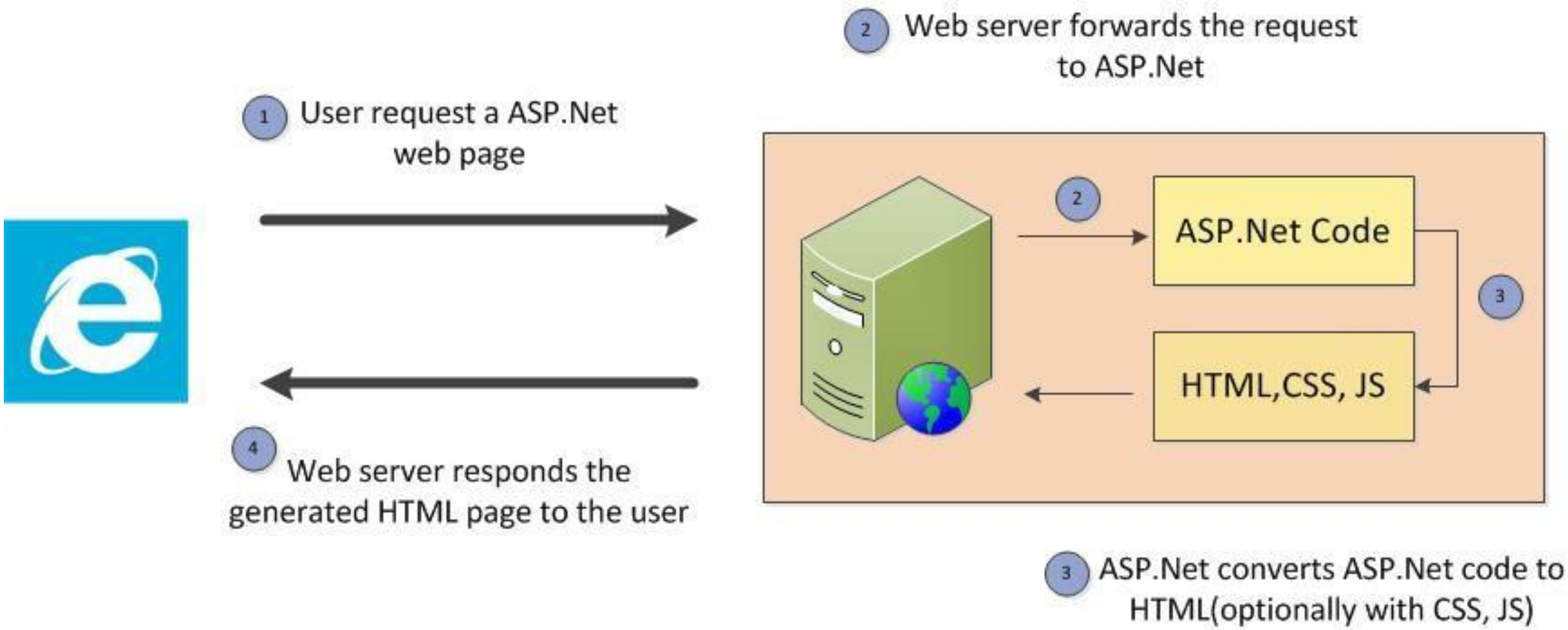
Errors:

- Error message 1.
- Error message 2.

ASP STANDS FOR ACTIVE SERVER PAGES.

- ASP.NET is an open source server-side web-application.
- It's a framework designed for web development.
- It's used to produce dynamic web pages.
- It was developed by Microsoft.
- It allows programmers to build dynamic web sites, applications and services.

PROCESS OF ASP.NET



1 User request a ASP.Net web page

2 Web server forwards the request to ASP.Net

3 ASP.Net converts ASP.Net code to HTML(optionally with CSS, JS)

4 Web server responds the generated HTML page to the user

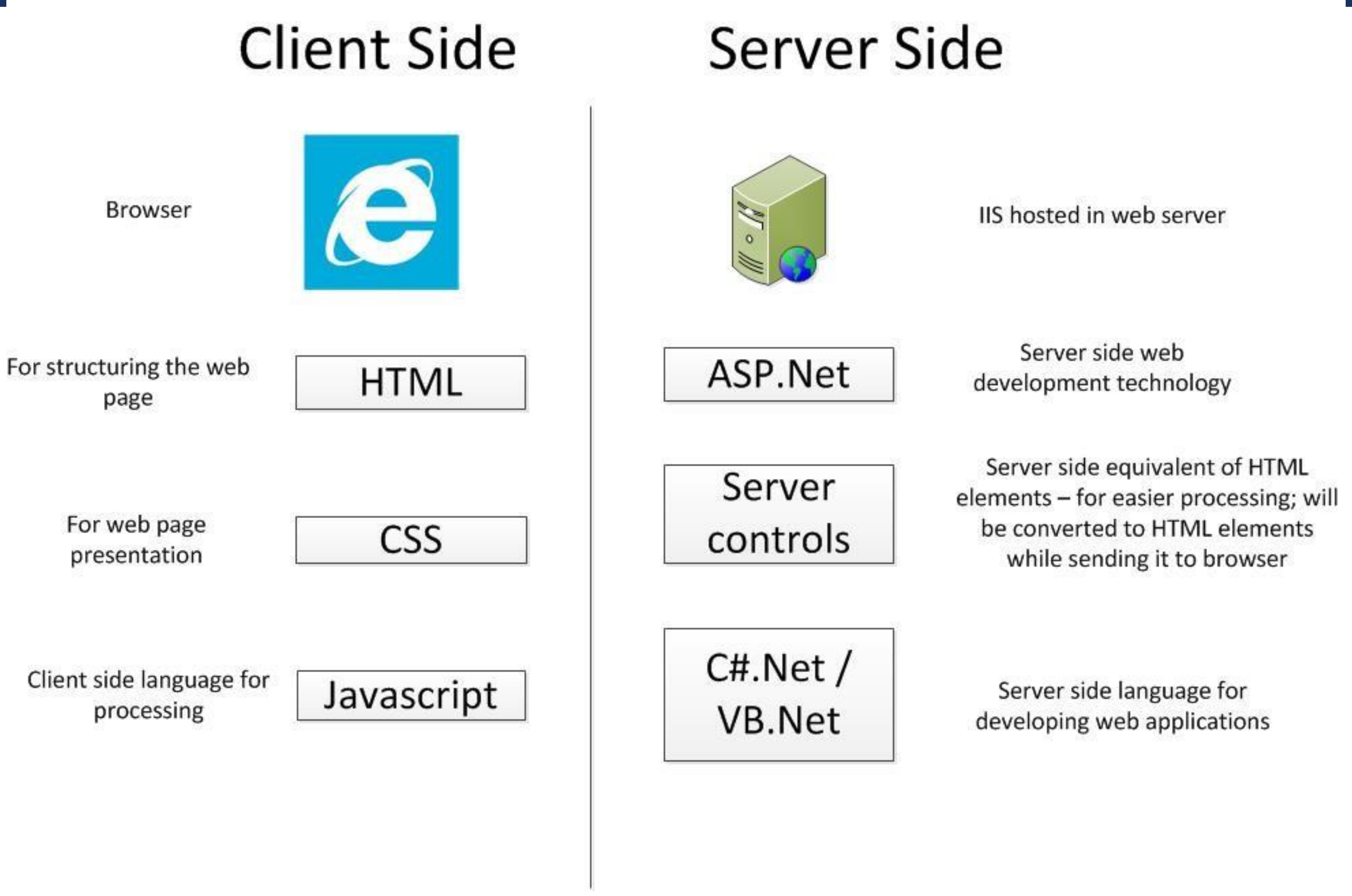
Developer(s)	Microsoft
Initial Release	January 5,2002,18 years ago
Stable Release	4.8 / April 18,2019, 9 months ago
Written In	.NET Languages
Operating System	Microsoft Windows, Linux, MacOS
Platform	Cross-Platform
Type	Web Framework
License	Apache License2.0
Website	Dotnet.microsoft/.com/apps/asp.net
Filename Extension	.aspx, .cshtml, .vbhtml
Internet Media Type	Text/HTML

- ASP.NET is build on the Common Language Runtime(CLR).
- It's allowing programmers to write ASP.NET code using any supported .NET languages.
- ASP.NET Simple Object Access Protocol extension (SOAP) framework allows ASP.NET components to process SOAP messages.
- ASP.NET's successor is ASP.NET Core.

- It’s a re-implementation of ASP.NET as a modular framework, together with other framework like Entity Framework.
- The new framework uses the new open source .NET Compiler Platform.
- ASP.NET MVC, ASP.NET WEB API, and ASP.NET Web Pages have merged into a unified MVC 6.

PROGRAMMING MODELS

- ASP.NET supports a number of programming models for building web applications:
- ASP.NET Web Forms - a framework for building modular pages out of components, with UI events being processed server-side.
- ASP.NET MVC - allows for building web pages using the model–view–controller design pattern.
- ASP.NET Web Pages - a lightweight syntax for adding dynamic code and data access directly inside HTML markup.
- ASP.NET Web API - a framework for building Web APIs on top of the .NET Framework.
- ASP.NET WebHooks - implements the Webhook pattern for subscribing to and publishing events via HTTP.
- SignalR - a real-time communications framework for bi-directional communication between client and server.



Client Side

Browser

For structuring the web page — HTML

For web page presentation — CSS

Client side language for processing — Javascript

Server Side

IIS hosted in web server

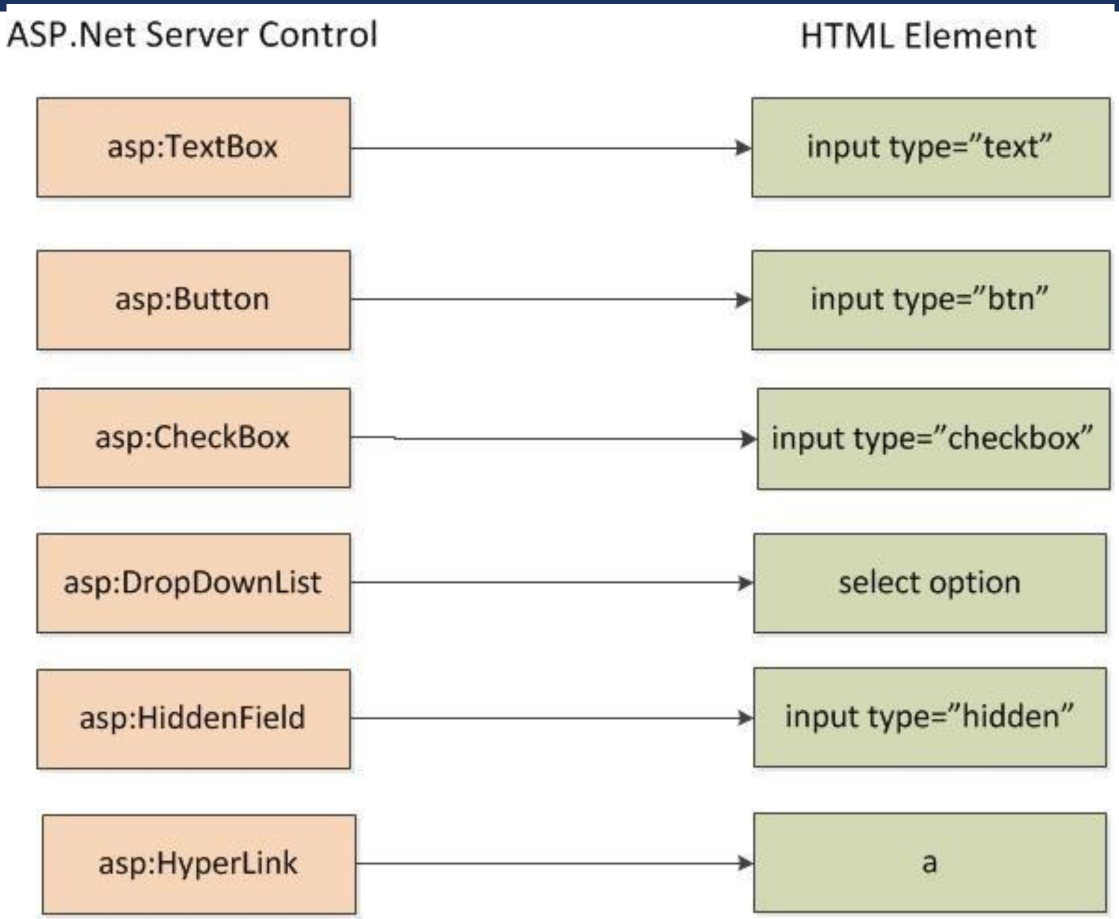
ASP.Net — Server side web development technology

Server controls — Server side equivalent of HTML elements – for easier processing; will be converted to HTML elements while sending it to browser

C#.Net / VB.Net — Server side language for developing web applications

OTHER ASP.NET EXTENSIONS INCLUDE:

- ASP.NET Handler: Are components that implement the System.Web.IHttpHandler interface. Unlike ASP.NET Pages, they have no HTML-markup file, no events and other supporting.
- All they have is a code-file (written in any .NET-compatible language) that writes some data to the server HTTP response. HTTP handlers are similar to ISAPI extensions.
- ASP.NET AJAX: An extension with both client-side as well as server-side components for writing ASP.NET pages that incorporate Ajax functionality. ASP.NET Dynamic Data: A scaffolding extension to build data driven web applications



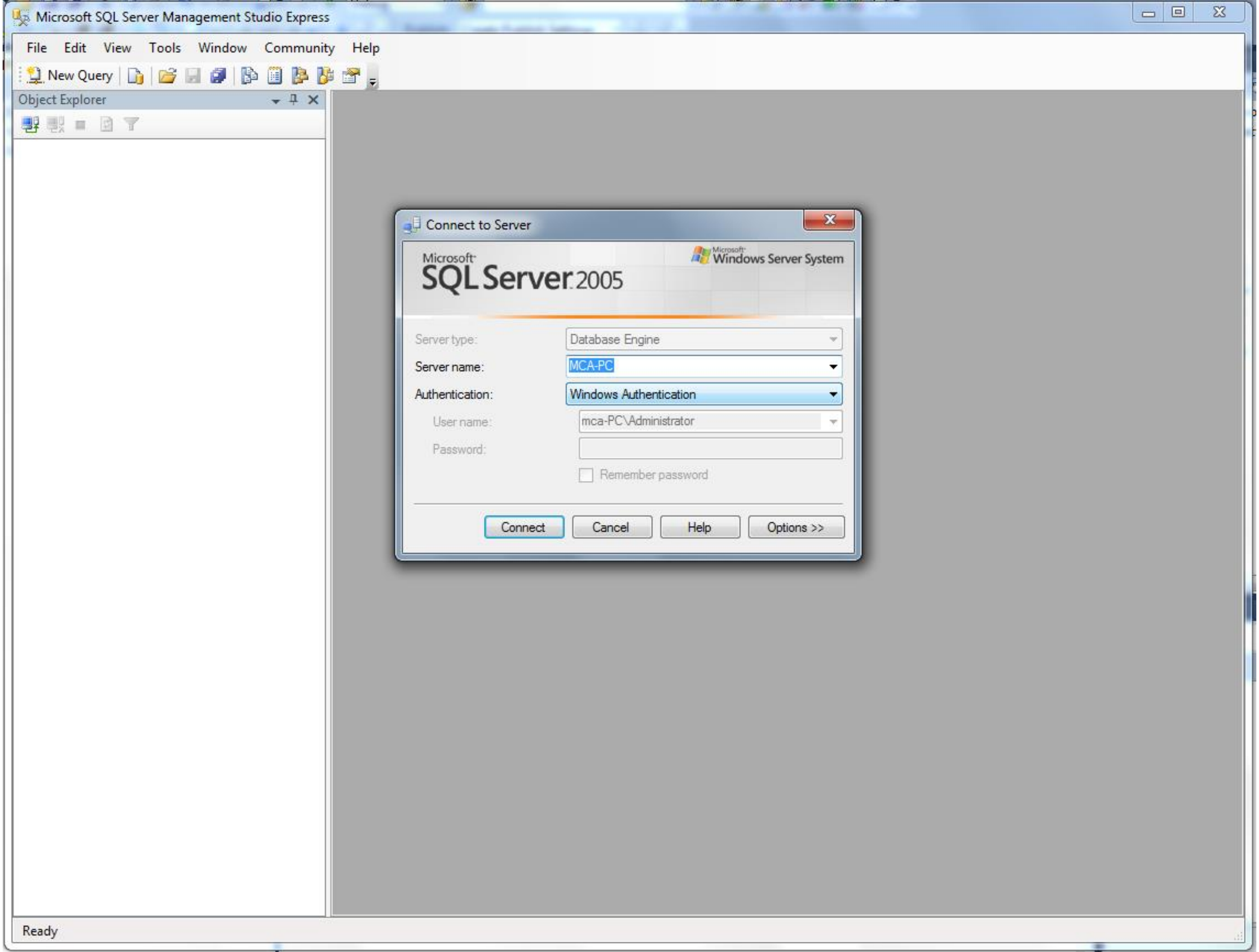
ASP.Net Server Control	HTML Element
asp:TextBox	input type="text"
asp:Button	input type="btn"
asp:CheckBox	input type="checkbox"
asp:DropDownList	select option
asp:HiddenField	input type="hidden"
asp:HyperLink	a

.NET PROGRAMMING(C#) [ASP.NET]

- SQL SERVER Database Connection

SQL SERVER

- SQL(Structured Query Language). Is a database server by MicroSoft



Microsoft SQL Server Management Studio Express

Connect to Server

Microsoft SQL Server 2005

Server type: Database Engine

Server name: MCA-PC

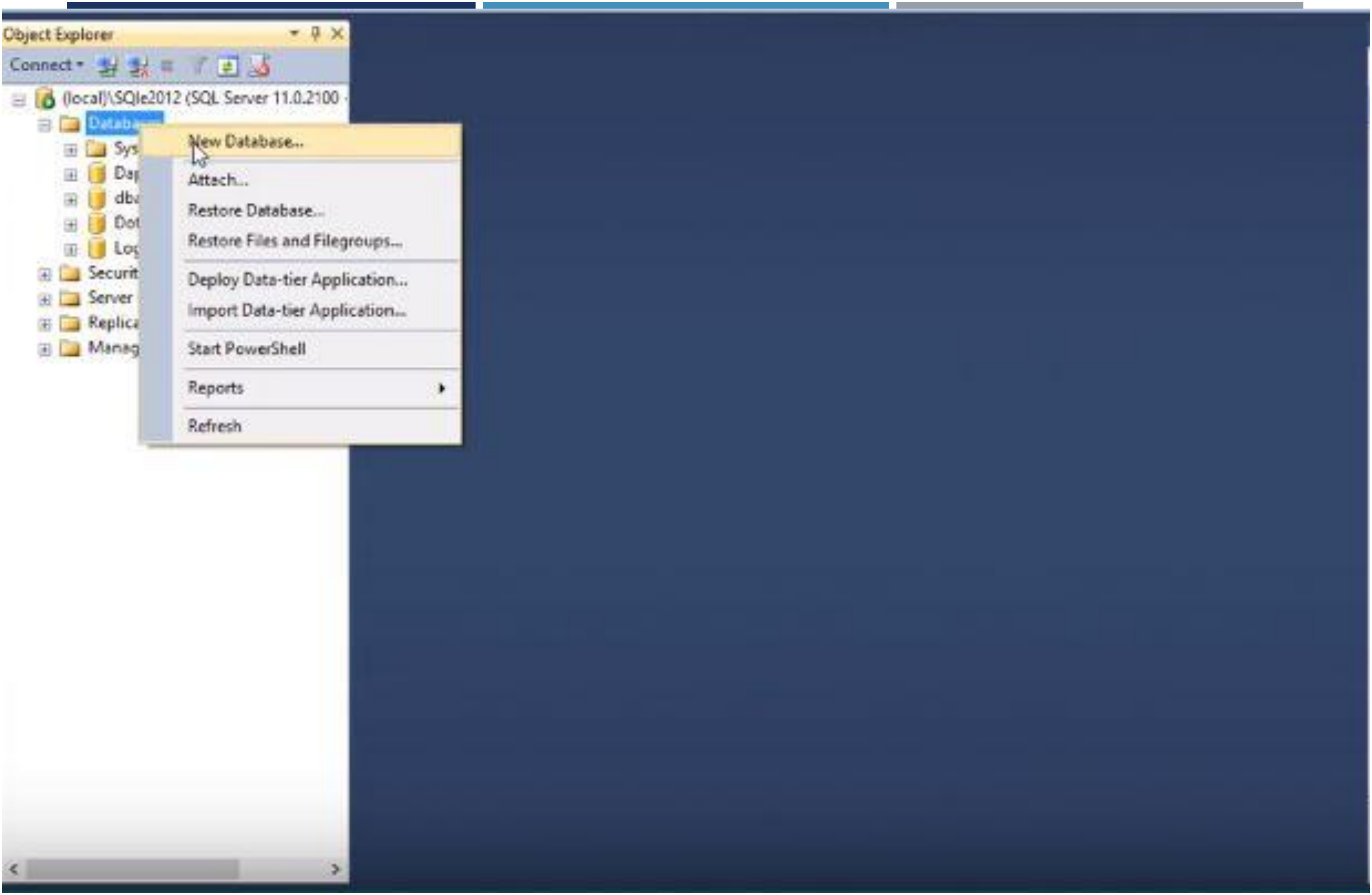
Authentication: Windows Authentication

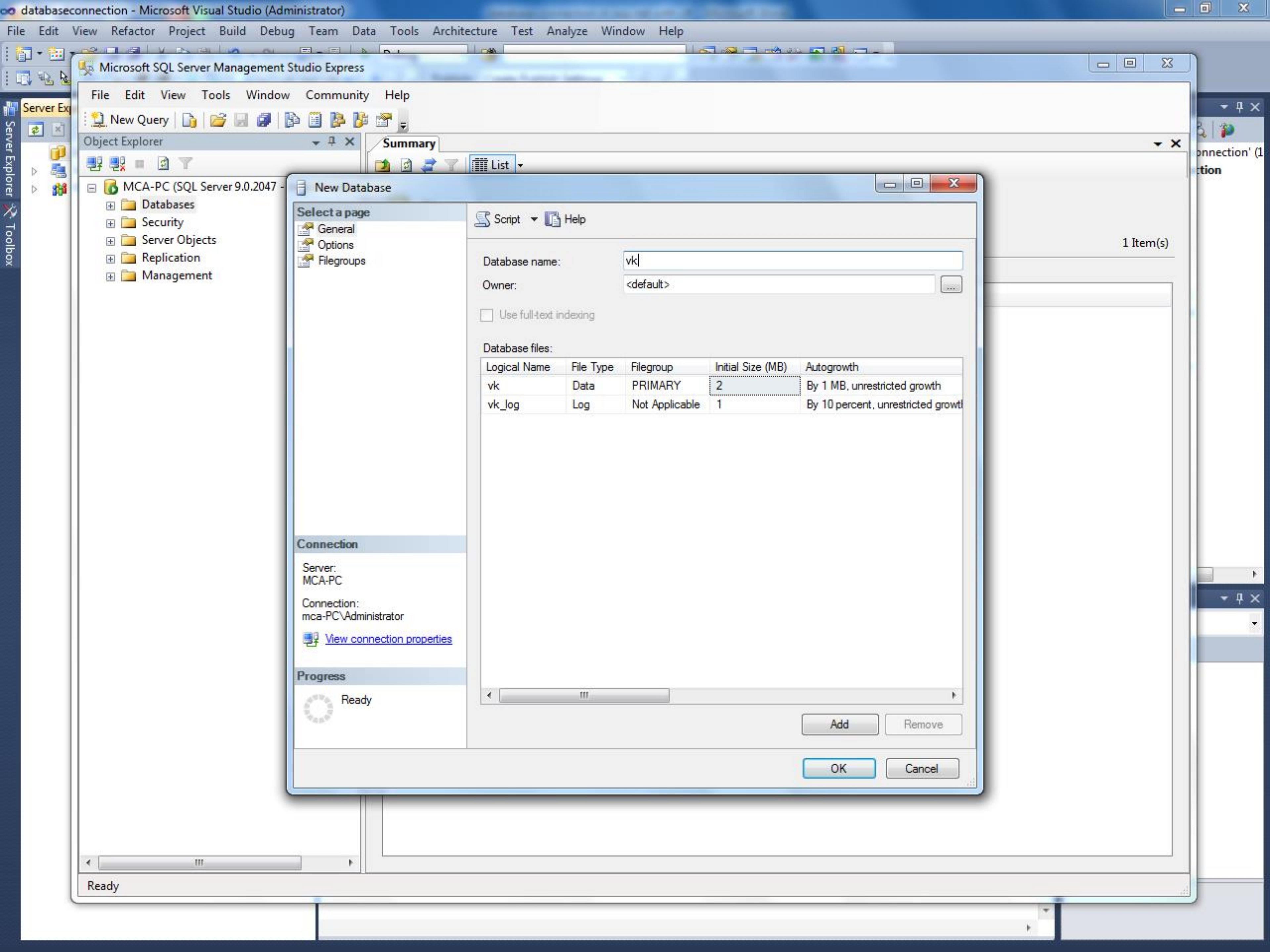
User name: mca-PC\Administrator

Password:

□ Remember password

Ready


Object Explorer
Connect
(local)\SQle2012 (SQL Server 11.0.2100
Databa
Sys
Dat
dba
Dot
Log
Securit
Server
Replica
Manag
New Database...
Attach...
Restore Database...
Restore Files and Filegroups...
Deploy Data-tier Application...
Import Data-tier Application...
Start PowerShell
Reports
Refresh


databaseconnection - Microsoft Visual Studio (Administrator)
File Edit View Refactor Project Build Debug Team Data Tools Architecture Test Analyze Window Help
Server Explorer
Toolbox
Microsoft SQL Server Management Studio Express
File Edit View Tools Window Community Help
New Query
Object Explorer
MCA-PC (SQL Server 9.0.2047 -
Databases
Security
Server Objects
Replication
Management
Summary
List
1 Item(s)
New Database
Select a page
General
Options
Filegroups
Script
Help
Add
Remove
OK
Cancel
View connection properties
Ready

Database name: vk

Owner: <default>

☐ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth
vk	Data	PRIMARY	2	By 1 MB, unrestricted growth
vk_log	Log	Not Applicable	1	By 10 percent, unrestricted growt

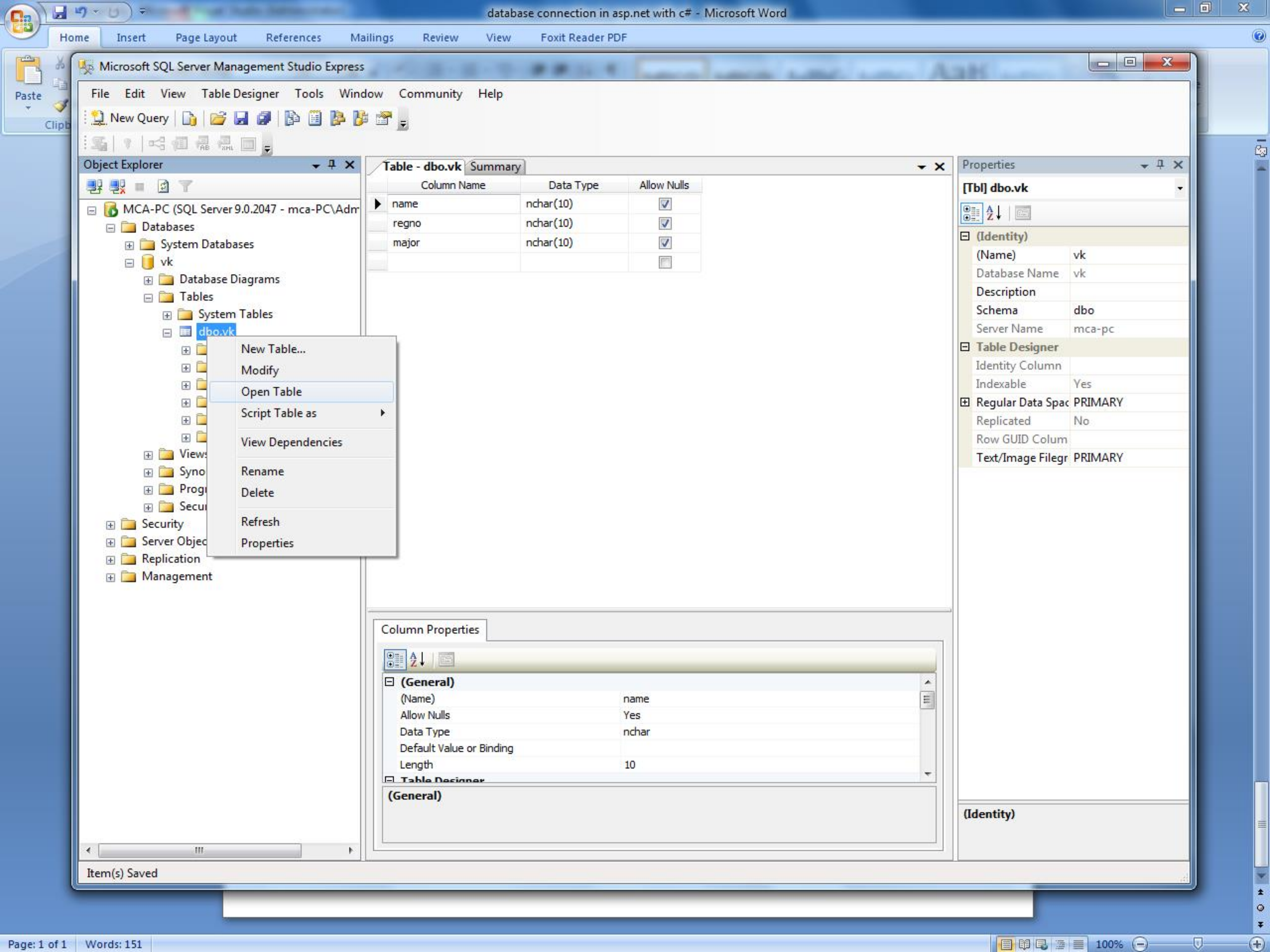
Connection

Server:
MCA-PC

Connection:
mca-PC\Administrator

Progress

Ready



Column Name	Data Type	Allow Nulls
name	nchar(10)	☑
regno	nchar(10)	☑
major	nchar(10)	☑
		□

Column Properties

(General)

(Name)	name
Allow Nulls	Yes
Data Type	nchar
Default Value or Binding	
Length	10

Table Designer

(General)

Properties

[Tbl] dbo.vk

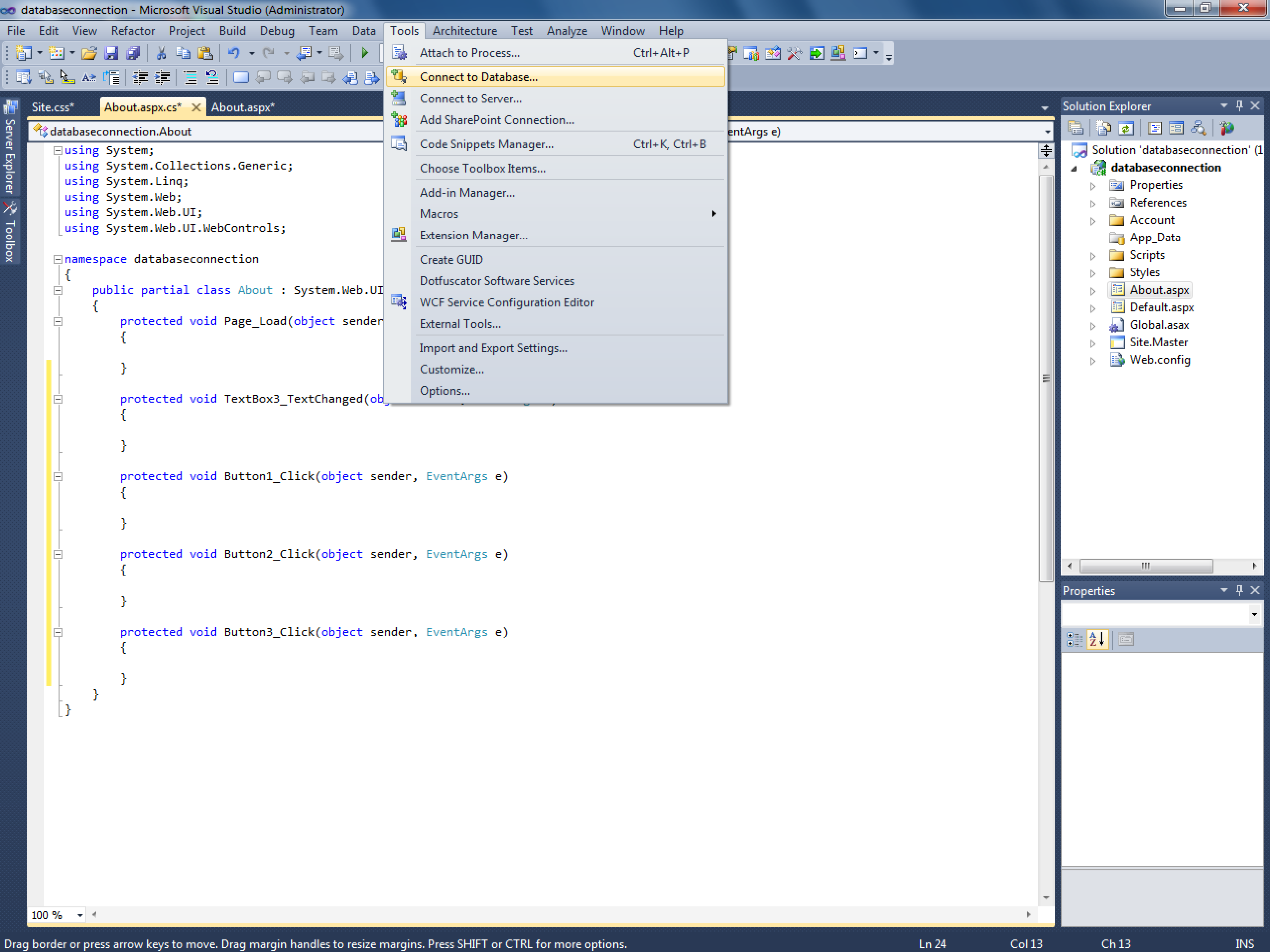
(Identity)

(Name)	vk
Database Name	vk
Description	
Schema	dbo
Server Name	mca-pc

Table Designer

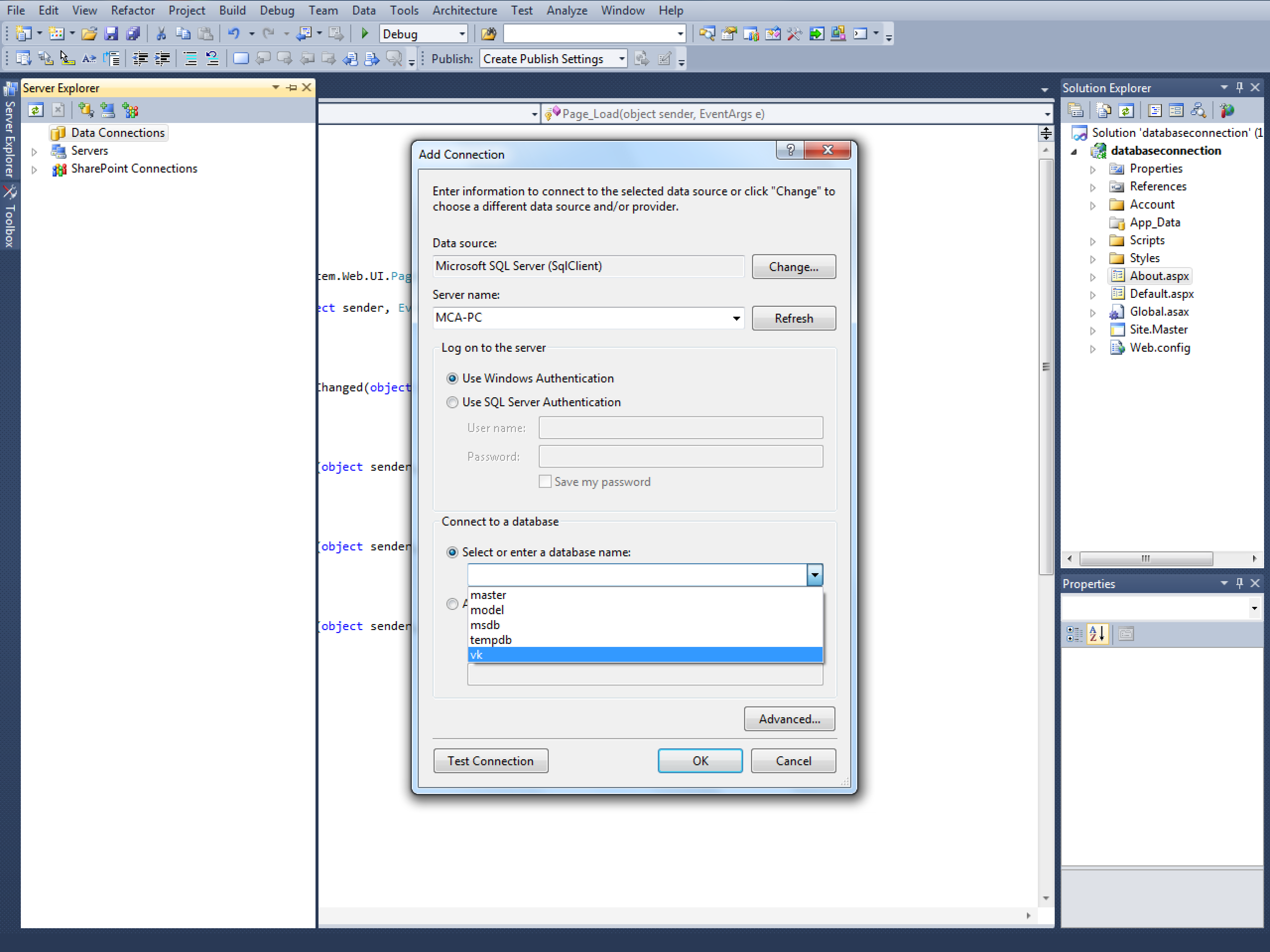
Identity Column	
Indexable	Yes
Regular Data Spac	PRIMARY
Replicated	No
Row GUID Colum	
Text/Image Filegr	PRIMARY

(Identity)



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace databaseconnection
{
    public partial class About : System.Web.UI
    {
        protected void Page_Load(object sender
        {

        }

        protected void TextBox3_TextChanged(ob

        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {

        }

        protected void Button2_Click(object sender, EventArgs e)
        {

        }

        protected void Button3_Click(object sender, EventArgs e)
        {

        }
    }
}
```

Drag border or press arrow keys to move. Drag margin handles to resize margins. Press SHIFT or CTRL for more options.



```
tem.Web.UI.Pag
ect sender, Ev
Changed(object
(object sender
(object sender
(object sender
```

Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft SQL Server (SqlClient)

Server name: MCA-PC

Log on to the server

● Use Windows Authentication

○ Use SQL Server Authentication

User name:

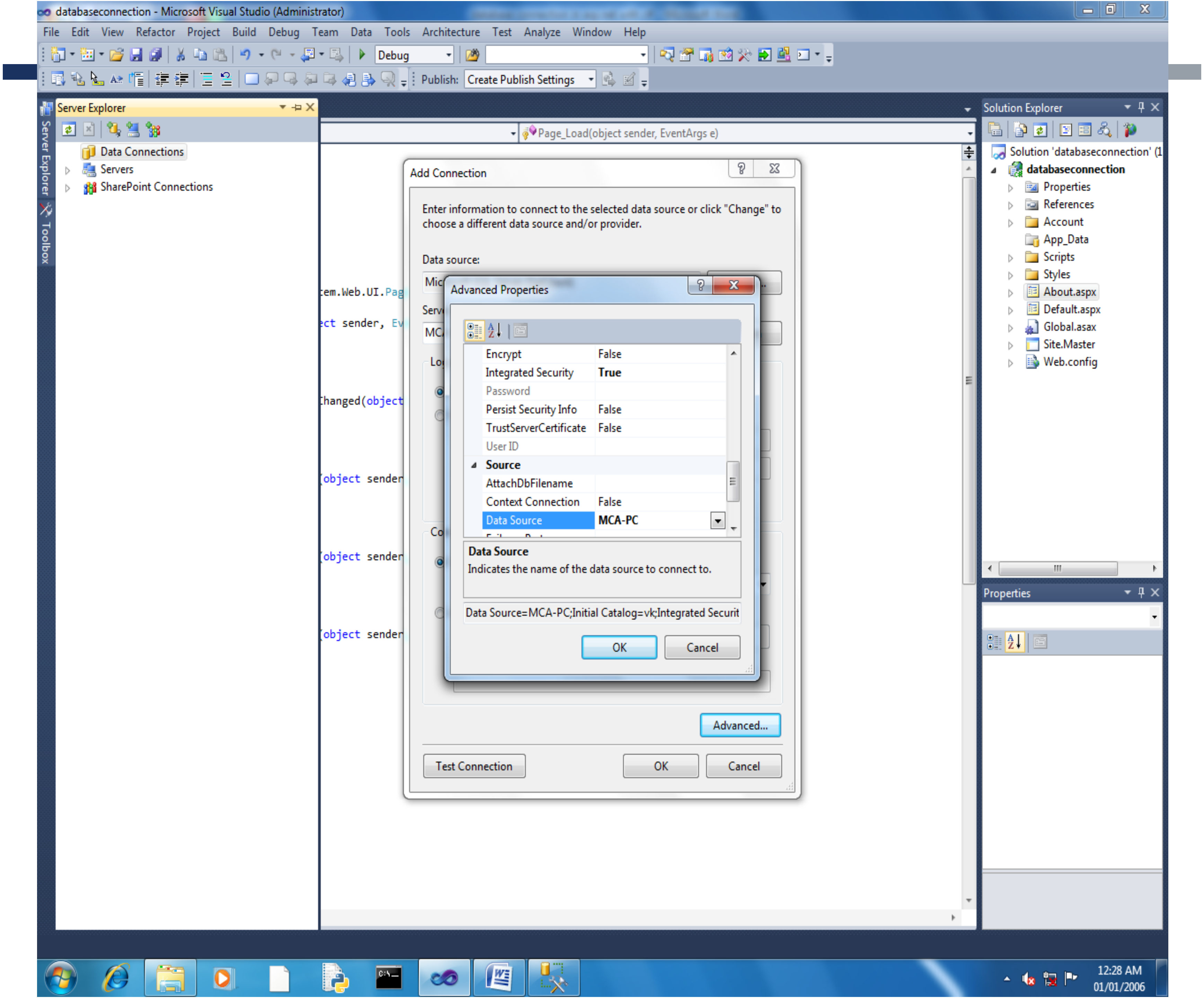
Password:

□ Save my password

Connect to a database

● Select or enter a database name:

master
model
msdb
tempdb
vk



Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:

Mic

Serv

MC

Lo

Co

```
em.Web.UI.Pag
ect sender, Ev
hanged(object
(object sender
(object sender
(object sender
```

Encrypt	False
Integrated Security	**True**
Password	
Persist Security Info	False
TrustServerCertificate	False
User ID	
Source	
AttachDbFilename	
Context Connection	False
Data Source	**MCA-PC**

Data Source

Indicates the name of the data source to connect to.

Data Source=MCA-PC;Initial Catalog=vk;Integrated Securit

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
namespace seminar
{
  public partial class _Default : Page
     {
        SqlConnection con = new SqlConnection(@"Data
Source=.\SQLEXPRESS;AttachDbFillname=c:\users\Dell\Documents\testing.mdf;Integrateds
Security=True;");
     protected void Page_Load(object sender, EventArgs e)
        {         dis_data();      }
        protected void Button1_Click(object sender, EventArgs e)
        {
            con.Open();
            SqlCommand cmd = con.CreateCommand();
            cmd.CommandType = CommandType.text;
            cmd.CommandText = "insert into vk values('" + TextBox1.Text + "','" +
TextBox2.Text + "','" + TextBox3.Text + "')";
cmd.ExecuteNonQuery();
            dis_data();
            TextBox1.Text = "";
            TextBox2.Text = "";
            TextBox3.Text = "";
            dis_data();
     }
 public void dis_data()
        {
                      SqlCommand cmd=con.CreateCommand();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText="select * from vk";
            cmd.ExecuteNonQuery();
            DataTable dt = new DataTable();
            SqlDataAdapter da = new SqlDataAdapter();
            da.Fill(dt);
            GridView1.DataSource = dt;
            GridView1.DataBind();
 }
 protected void Button4_Click(object sender, EventArgs e)
        {
            dis_data();
        }
 protected void Button3_Click(object sender, EventArgs e)
        {
            SqlCommand cmd = con.CreateCommand();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = "delete from vk where name='"+TextBox1.Text+"'";
            cmd.ExecuteNonQuery();
            dis_data();
        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            SqlCommand cmd = con.CreateCommand();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = "update vk set 
name='"+TextBox1.Text+"',major='"+TextBox3.Text+"'where regno='"+TextBox2.Text+"'";
                cmd.ExecuteNonQuery();
                TextBox1.Text = "";
                TextBox2.Text = "";
                TextBox3.Text = "";
                dis_data();
        }
    }
}
```

.NET PROGRAMMING(C#) [WIN FORM APPLICATIONS]

CREATING A WINDOWS APPLICATIONS PROJECT IN VISUAL STUDIO

INTRODUCTION

- Windows Form is a GUI(Graphical User Interface) class library that is bundled in .NET Framework.
- It provides an easier interface to develop applications for
  - Desktop, etc.

INTRODUCTION(CONT'D)

- Also Known as 'WinForms'.
- The applications developed by 'WinForms' or 'Windows Form' are known as the 'Windows Forms Application' that runs on specific platforms.

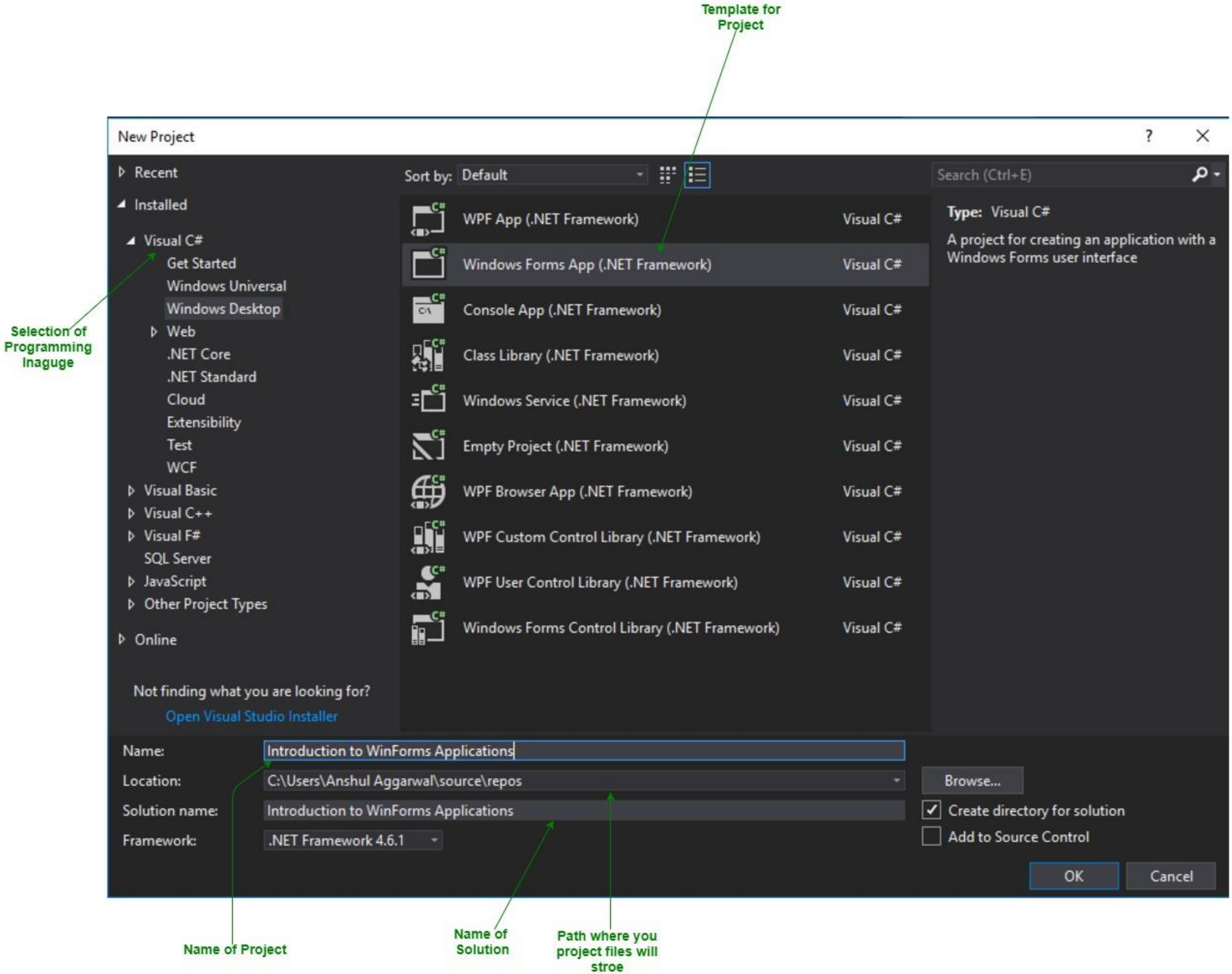
INTRODUCTION(CONT'D)

- WinForms can contain various controls like the following.,
  - Labels
  - TextBox
  - Button
  - ListBox
  - Etc.

CREATING WINDOWS FORM APPLICATIONS

VISUAL STUDIO

- Open Visual Studio(any version).
- Create a new project by,
  - Click **File** Menu
    - Click **New**
      - Click Project.
      - A Window with built-in templates will occur.
- In the templates window,
  - Select '**Windows Forms Application**' from Visual C# group.
    - Provide a suitable name for the project.
      - Click 'Ok'.



Template for Project

Selection of Programming lnaguge

Name of Project

Name of Solution

Path where you project files will stroe

WPF App (.NET Framework)	Visual C#
Windows Forms App (.NET Framework)	Visual C#
Console App (.NET Framework)	Visual C#
Class Library (.NET Framework)	Visual C#
Windows Service (.NET Framework)	Visual C#
Empty Project (.NET Framework)	Visual C#
WPF Browser App (.NET Framework)	Visual C#
WPF Custom Control Library (.NET Framework)	Visual C#
WPF User Control Library (.NET Framework)	Visual C#
Windows Forms Control Library (.NET Framework)	Visual C#

Type: Visual C#

A project for creating an application with a Windows Forms user interface

Name: Introduction to WinForms Applications

Location: C:\Users\Anshul Aggarwal\source\repos

Solution name: Introduction to WinForms Applications

Framework: .NET Framework 4.6.1

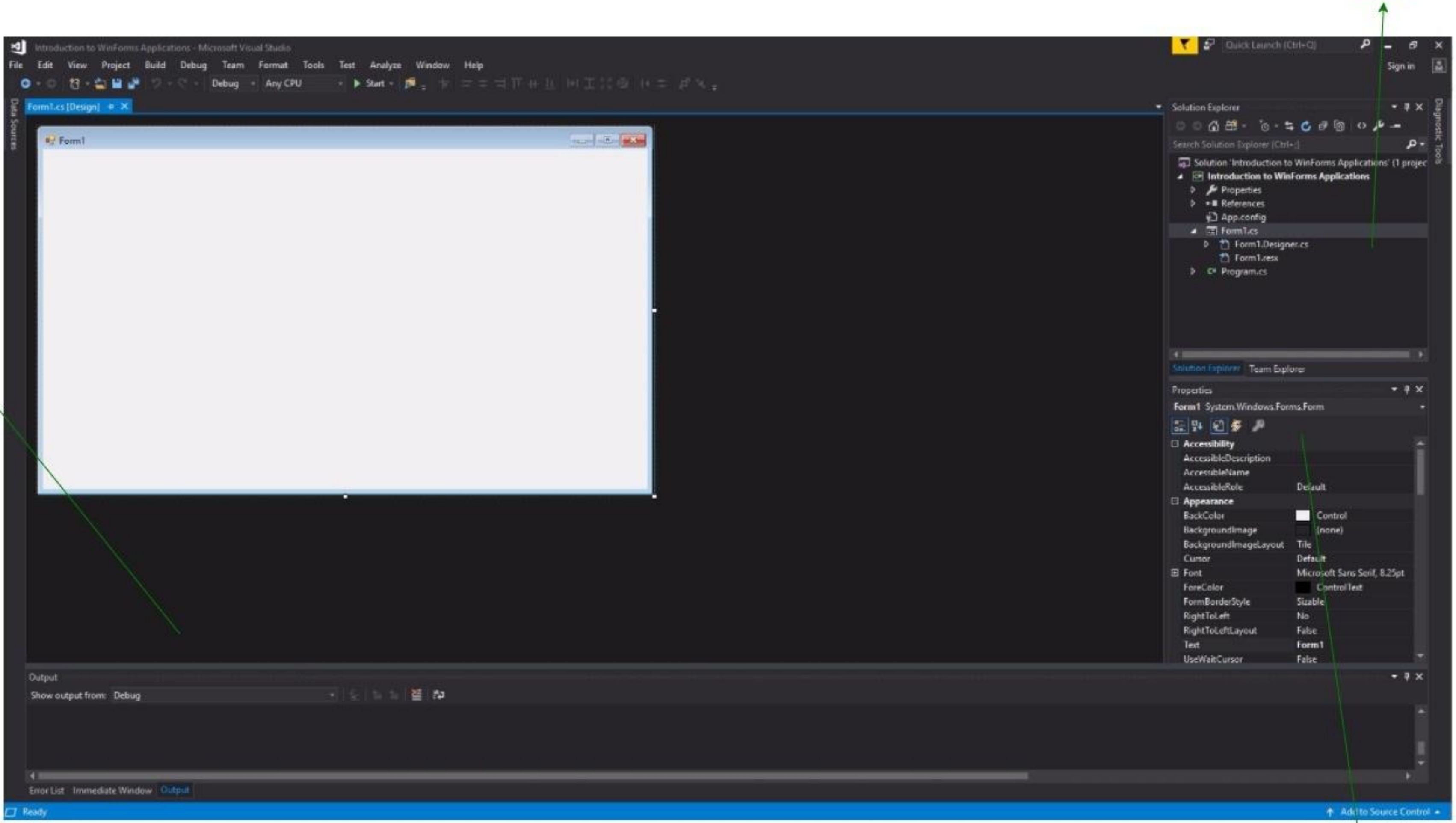
☑ Create directory for solution

☐ Add to Source Control

THE VISUAL STUDIO PROJECT WINDOW

- After creating a project, The following 3 windows will occur.
  - **Editor (or) Main Window:** A Place for design and code.
  - **Solution Explorer Window:** Navigation between project elements such as Forms, Code, etc.
  - **Properties Window:** Contains a set of properties for each controls on the form.

Solution Explorer Window



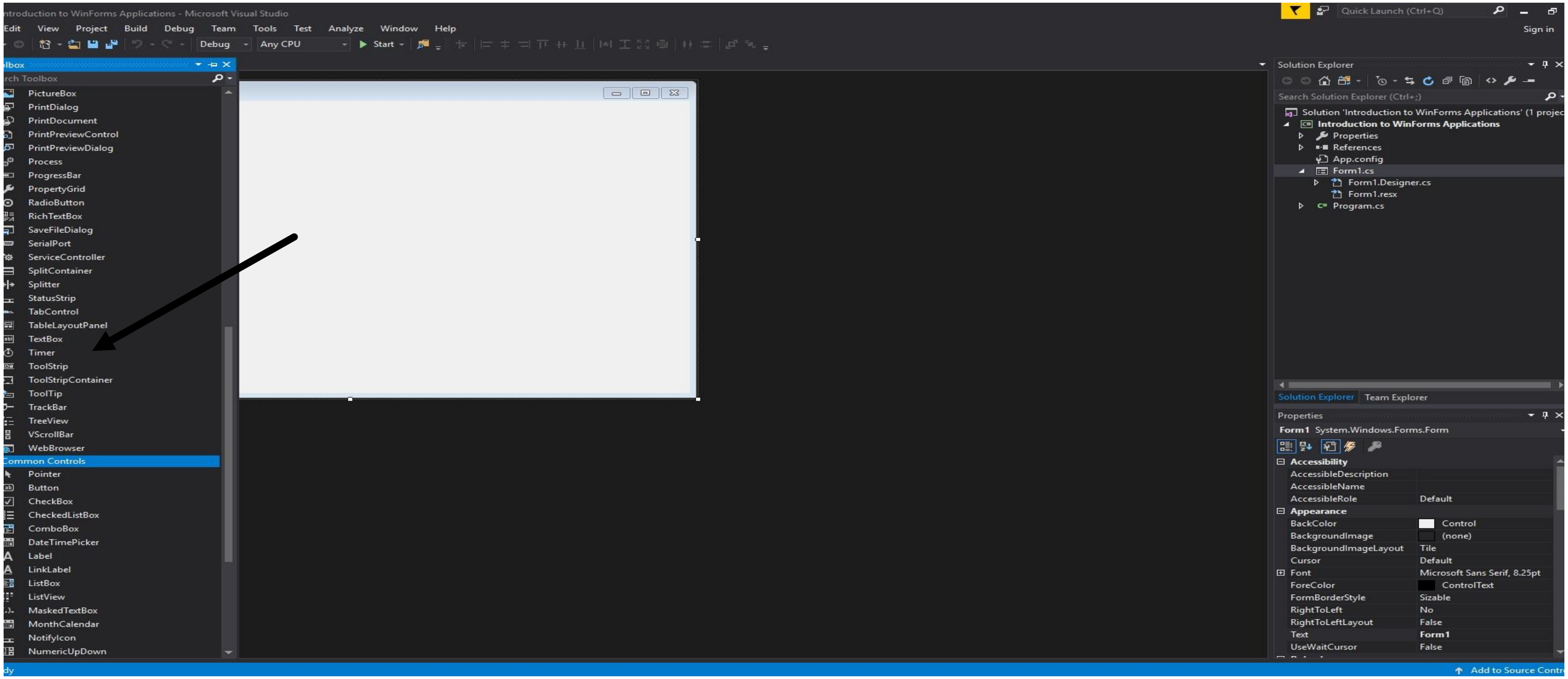
Solution 'Introduction to WinForms Applications' (1 projec
- Introduction to WinForms Applications
  - Properties
  - References
  - App.config
  - Form1.cs
    - Form1.Designer.cs
    - Form1.resx
  - Program.cs

Form1 System.Windows.Forms.Form

Accessibility	
AccessibleDescription	
AccessibleName	
AccessibleRole	Default
Appearance	
BackColor	Control
BackgroundImage	(none)
BackgroundImageLayout	Tile
Cursor	Default
Font	Microsoft Sans Serif, 8.25pt
ForeColor	ControlText
FormBorderStyle	Sizable
RightToLeft	No
RightToLeftLayout	False
Text	Form1
UseWaitCursor	False

Editor Window

Properties Window

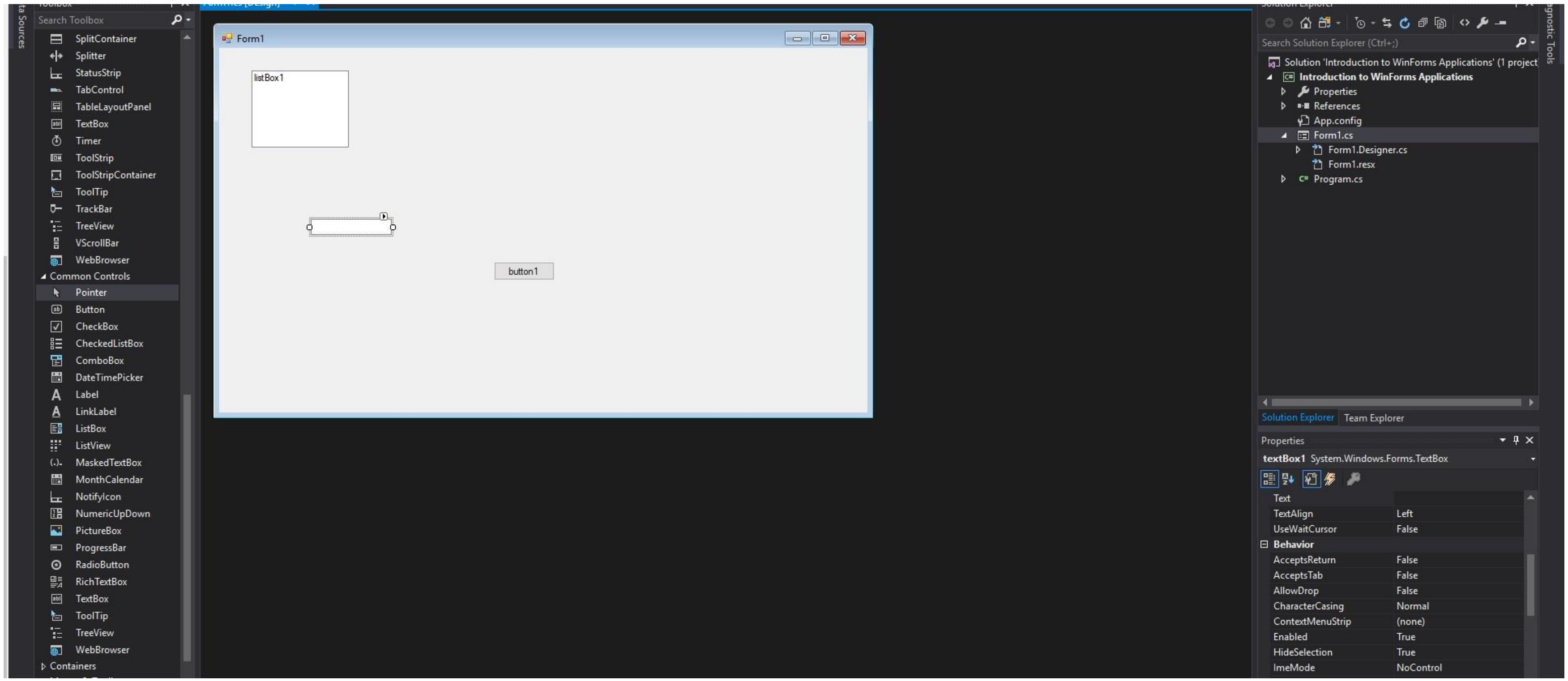


Accessibility	
AccessibleDescription	
AccessibleName	
AccessibleRole	Default
Appearance	
BackColor	Control
BackgroundImage	(none)
BackgroundImageLayout	Tile
Cursor	Default
Font	Microsoft Sans Serif, 8.25pt
ForeColor	ControlText
FormBorderStyle	Sizable
RightToLeft	No
RightToLeftLayout	False
Text	Form1
UseWaitCursor	False

TOOLBOX

One Another window is a 'ToolBox'.

ToolBox contains several control needed for form.



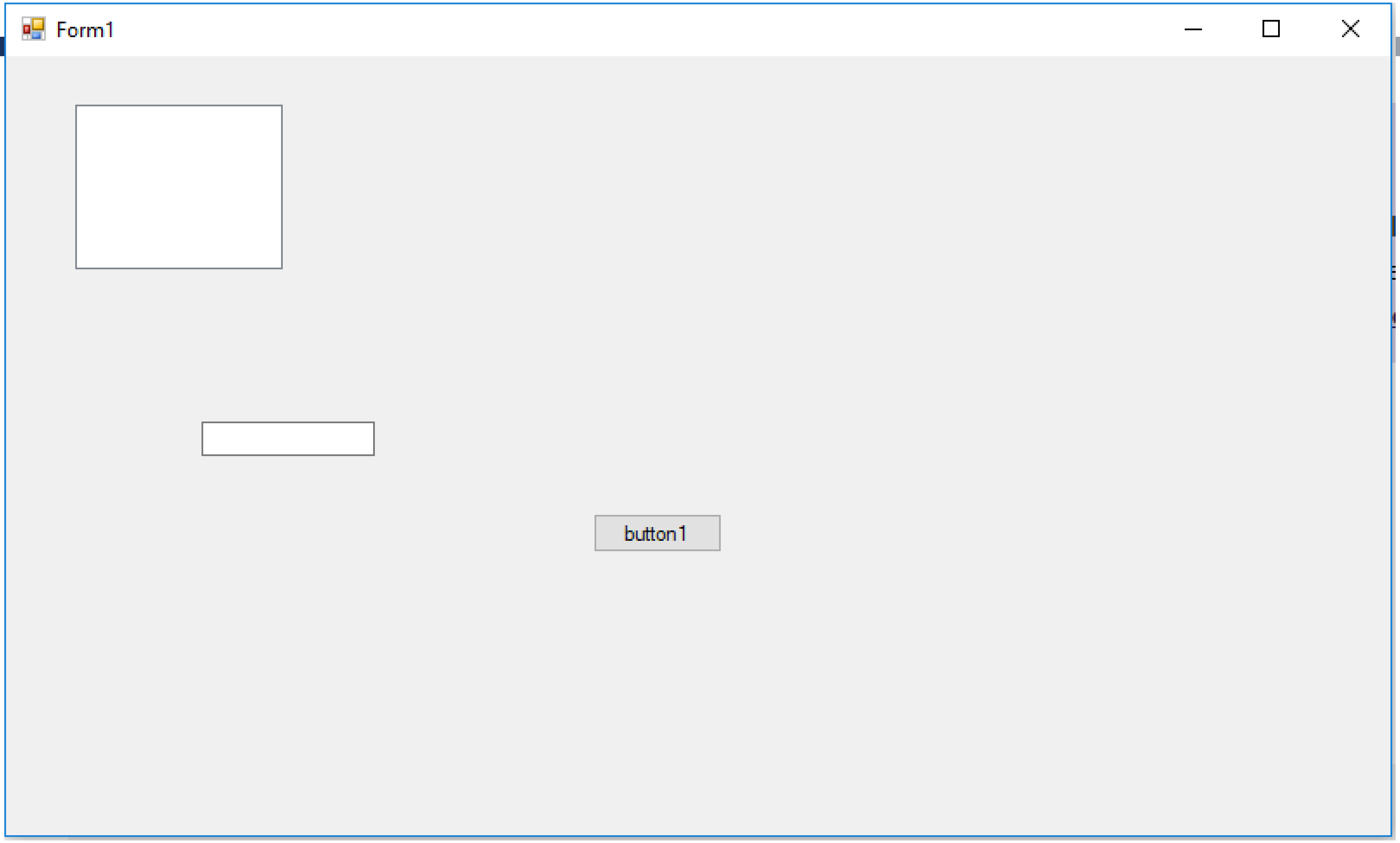
textBox1 System.Windows.Forms.TextBox

Text	
TextAlign	Left
UseWaitCursor	False
Behavior	
AcceptsReturn	False
AcceptsTab	False
AllowDrop	False
CharacterCasing	Normal
ContextMenuStrip	(none)
Enabled	True
HideSelection	True
ImeMode	NoControl

Controls can be added to the form by,

Double clicking on the controls in the toolbox or

Dragging and Dropping them from Toolbox.



IF THE PROJECT IS COMPLETED, CLICK ‘**F5**’ TO RUN IT.

THANKYOU

THE CONTENTS IN THIS E-MATERIAL IS TAKEN FROM THE TEXTBOOKS AND REFERENCE BOOKS GIVEN IN THE SYLLABUS