



Important Instructions to examiners:

- 1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
- 2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
- 3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills).
- 4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
- 5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
- 6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
- 7) For programming language papers, credit may be given to any other program based on equivalent concept.

1. (a) Attempt any THREE of the following :

Marks 12

i) What is data provider? Explain its components.

(Provider-1 Mark and Components -3 Marks)

Data Provider provides access to data source (SQL SERVER , ACCESS ,ORACLE etc.).

Data Provider is a set of components that facilitate data access. It provides objects to achieve functionalities like opening and closing connection ,retrieve data and update data.

It contains components like connection, command, dataReader, dataAdapter.

Connection object represents physical connection of the datasource. Its properties determine the data provider, data source and data base to which it will connect, and the string to be used during connection.

Command object represents a SQL statement or stored procedure to be executed at the data source.

DataReader object that provides connected, forward-only, read-only access to a database. DataReaders cannot be created directly in code, they are created by calling ExecuteReader method of Command object.

DataAdapter provides disconnected data access. It is the most complex object in a Data Provider. It provides the bridge between a Connection and a Dataset. It contains four Command objects as SelectCommand, UpdateCommand, InsertCommand and DeleteCommand. SelectCommand can be used to fill the Dataset and other commands can be used to transmit the changes to the database as per requirements.

**ii) Explain types of Error and what is error provider?**

(Types of error -1 Marks and Error provider-3 Marks)

Types of Errors:

Errors can be mainly classified as Syntactical and Logical errors.

The mistakes done in writing correct syntax of any statement in computer language causes syntactical error. It is generally identified at the time of compilation of program.

The errors caused by writing wrong logical steps of the program are logical errors. These are generally identified when output of program is examined with all possible inputs.

Error Provider:

The Windows Forms ErrorProvider component is used to validate user input on a form or control. It is typically used in conjunction with validating user input on a form, or displaying errors within a dataset. An error provider is a better alternative than displaying an error message in a message box, because once a message box is dismissed, the error message is no longer visible. TheErrorProvider component displays an error icon (🔴) next to the relevant control, such as a text box; when the user positions the mouse pointer over the error icon, a ToolTip appears, showing the error message string.

The ErrorProvider component's key properties are DataSource, ContainerControl, and Icon. The ContainerControl property must be set to the appropriate container (usually the Windows Form) in order for the ErrorProvider component to display an error icon on the form. When the component is added in the designer, the ContainerControl property is set to the containing form; if you add the control in code, you must set it yourself.

The Icon property can be set to a custom error icon instead of the default.

When the DataSourceproperty is set, the ErrorProvider component can display error messages for a dataset.

The key method of the ErrorProvider component is the SetError method, which specifies the error message string and where the error icon should appear.

```
Private Sub TextBox1_Validating(ByVal Sender As Object, _  
ByVal e As System.ComponentModel.CancelEventArgs) Handles _  
TextBox1.Validating
```

```
    If Not IsNumeric(TextBox1.Text) Then
```

```
        ErrorProvider1.SetError(TextBox1, "Not a numeric value.")
```



Else

' Clear the error.

ErrorProvider1.SetError(TextBox1, "")

End If

End Sub

iii) Differentiate checkbox and checklist box with an example

(1/2 Mark for each point, Example- 1 Mark each.)

The CheckBox control and the CheckBoxList control provide a way for users to specify yes/no (true/false) choices.

Differerence :

Sr.No	CheckBox	CheckBoxList
1	You add individual CheckBox controls to a page and work with them singly,	You can use the CheckBoxList control, which is a single control that acts as a parent control for a collection of check-box list items.
2	By using individual CheckBox controls, you get more control over the layout of the check boxes on the page than by using the CheckBoxList control.	You get less control over the layout of the check boxes on the page than by using the CheckBoxList control.
3	The CheckBox control is a better choice if you want to individually select an element.	The CheckBoxList control is a better choice if you want to create a series of check boxes from data in a data source.
4	Individual CheckBox controls raise the CheckedChanged event when users click the control.	The CheckBoxList control raises a SelectedIndexChanged event when users select any check box in the list.

CheckBox Example:

```
Private Sub CheckBox1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles  
    s CheckBox1.Click
```

```
If CheckBox1.Checked = True Then
```

```
    CheckBox1.Text = "Checked"
```



Else

 CheckBox1.Text = "Unchecked"

End If

End Sub

CheckBoxList Example:

Protected Sub CheckBoxList1_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles CheckBoxList1.SelectedIndexChanged

 Dim CheckCollection As String

 Dim i As Integer

 For i = 0 To CheckBoxList1.Items.Count - 1

 If (CheckBoxList1.Items(i).Selected = True) Then

 CheckCollection = CheckCollection + " " + CheckBoxList1.Items(i).Text

 End If

 Next

 Response.Write(CheckCollection)

End Sub

iv) How CDOSYS is different from CDONTS

(CDOSYS -2 Marks and CDONTS-2 Marks)

CDONTS provide only email services. If you have IIS SMTP service installed on your machine, you can send an email using the service and CDONTS. NewMail object. This object lets you send text or HTML emails (with attachment/images) by a simple way, but its performance is not so good. You can send only several few emails per second. It is mainly designed for Microsoft NT server.

CDOSYS is descendant of CDONTS. It is an easy way to send email from web pages. It is also a Microsoft technology which simplifies messaging application. It's a built in component of ASP.

v) What is global.asax file

(Explanation-4 Marks)

The Global.asax file, sometimes called the ASP.NET application file, provides a way to respond to application or module level events in one central location. You can use this file to implement



application security, as well as other tasks. The Global.asax file is in the root application directory. It's actually an optional file. The .asax file extension signals that it's an application file. The Global.asax file is so configured that any direct HTTP request (via URL) is rejected automatically, so users cannot download or view its contents. The ASP.NET page framework recognizes automatically any changes that are made to the Global.asax file. The framework reboots the application, which includes closing all browser sessions, flushes all state information, and restarts the application domain.

The Global.asax file, which is derived from the `HttpApplication` class, maintains a pool of `HttpApplication` objects, and assigns them to applications as needed.

It contains events such as `Application_Init`, `Application_Disposed`, `Application_Error`, `Application_Start`, `Application_End`, `Application_BeginRequest`, `Application_EndRequest` etc.

Global.asax can be created as

- Create new Web Site/ Web Application.
- In solution explorer, right click on project name -> Add new item->select 'Global Application class' option.
- Write required code in the events provided in global.asax.



b) Attempt any ONE of the following:

Marks 6

i) Compare request object and response object.

(1 Mark for each point)

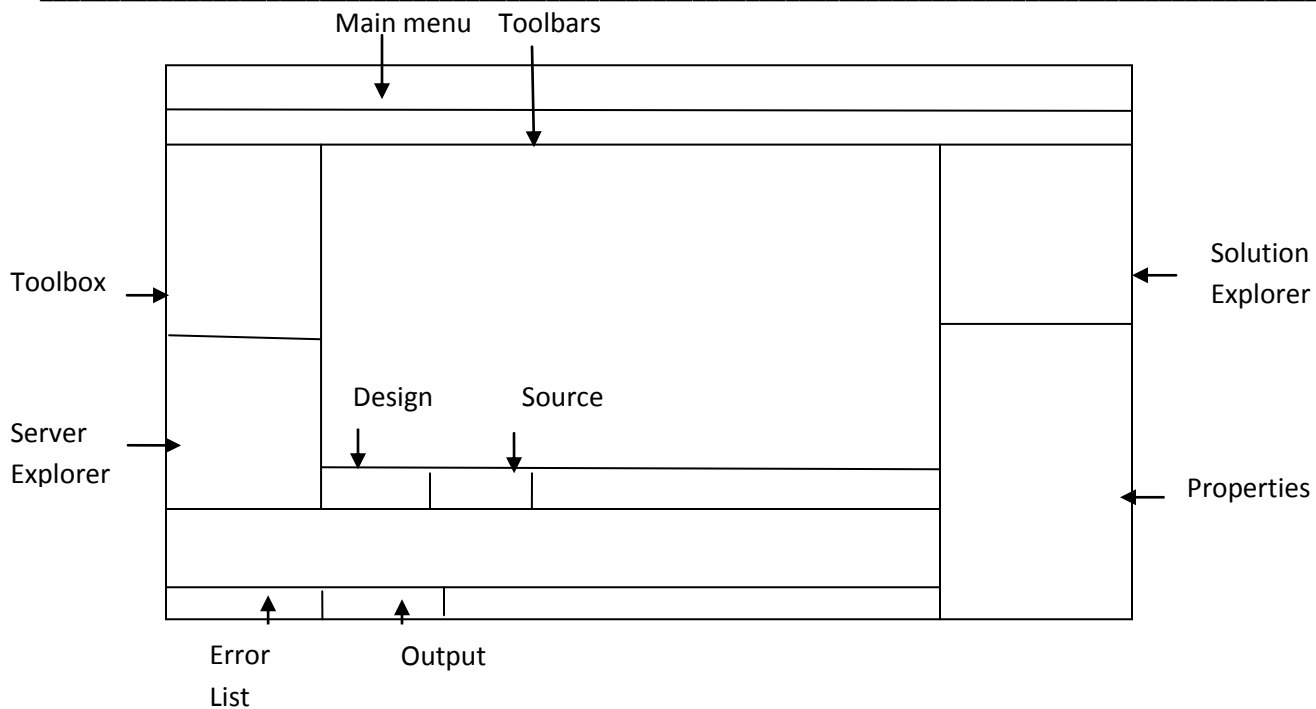
	Request Object	Response Object
1.	Request is sent from client to web server.	Response is sent from web server to the client.
2.	A request includes information like the browser who makes the request.	Inserts information into page being sent back to the client.
3.	Stores information in the form of cookies.	Sends information to the browser to create cookies on the client.
4.	Data input is by users	Output generated by web server is sent to client
5.	Some of the major properties are Request.QueryString Request.TotalBytes Request.ServerVariables	Some of the major properties are Response.Buffer Response.ContentType Response.CacheControl
6.	Method : Request.BinaryRead()	Method: Response.Write() Response.Redirect()

ii) Give brief idea of ASP.net IDE. How will you create web forms and what are web form controls.

(ASP.net IDE-3 Marks , Steps for Web form-1 Mark and Any four Web controls -2 Marks)

Asp.net IDE :

ASP.net IDE is an Integrated Development Environment for developing Web sites and web applications. It is divided into different sections according to functionality as follows:



Main Menu : Displays major options to handle management of Web application.

Toolbars : Shows iconic representations of different options from main menu.

Design : Provides area for placement of controls on web pages.

Source : Provides area for code behind.

Toolbox : Provides tools like html controls, web controls, data controls etc.

Server Explorer : Explores databases to be used in the web applications.

Properties : Shows Properties of the selected control.

Solution Explorer : Shows list of files from the current web project.

Steps to create Web form :

Select File -> New -> Project -> Web -> provide name of the Web Application-> press Enter

It shows the design or source window for the default file named 'Default.aspx'.

List Web Form Controls:

Label, TextBox, Button, LinkButton, ImageButton, HyperLink, DropDownList, ListBox, CheckBox, CheckBoxList, RadioButton, RadioButtonList, Image, ADRotator, FileUpload.

**2. Attempt any TWO of the following :****Marks 16****a) How repeaters are used to binding the data on ASP.net***(Explanation-6 Marks, Example-2 Marks)*

The Repeater control provides two properties to support data binding. To bind data to any collection that implements the System.Collections. IEnumerable interface (such as System.Data. DataView, a System.Collections. ArrayList, or an array), or the IListSource interface, use the DataSource property to specify the data source. When you set the DataSource property, you must manually write the code to perform data binding. To automatically bind the Repeater control to a data source represented by a data source control, set the DataSourceID property to the ID of the data source control to use. When you set the DataSourceID property, the Repeater control automatically binds to the specified data source control on the first request. Therefore, you do not need to explicitly call the DataBind method unless you have changed data-related properties of the Repeater control. A Repeater control binds its ItemTemplate and Alternating ItemTemplate to either the data model declared and referenced by its DataSource property or the data source control specified by its DataSourceID property. The HeaderTemplate, FooterTemplate, and SeparatorTemplate are not data-bound. If the Repeater control's data source is set but no data is returned, the control renders the HeaderTemplate and FooterTemplate with no items. If the data source is null, the Repeater is not rendered.

Templates

At a minimum, every Repeater control must define an ItemTemplate. However, other optional templates described in the following table can be used to customize the appearance of the list.



Template name	Description
ItemTemplate	Defines the content and layout of items within the list.
AlternatingItemTemplate	If defined, determines the content and layout of alternating (zero-based odd-indexed) items.
SeparatorTemplate	If defined, is rendered between items (and alternating items). If not defined, a separator is not rendered.
HeaderTemplate	If defined, determines the content and layout of the list header. If not defined, a header is not rendered.
FooterTemplate	If defined, determines the content and layout of the list footer. If not defined, a footer is not rendered.

Example :

```
<% @ Page Language="vb" %>
<% @ Import Namespace="System.Data" %>
<% @ Import Namespace="System.Data.SqlClient" %>

<script runat="server">
Sub Page_Load(sender As Object, e As EventArgs)

    Dim cnn As SqlConnection = New SqlConnection("server=(local);" & _
        "database=pubs;Integrated Security=SSPI")

    Dim cmd As SqlDataAdapter = New SqlDataAdapter("select * from authors", cnn)
    Dim ds As DataSet = New DataSet()
    cmd.Fill(ds)
```



```
Repeater1.DataSource = ds

Repeater1.DataBind()
End Sub
</script>
<html>
<body>
  <form id="Form1" method="post" runat="server">
    <asp:Repeater id="Repeater1" runat="server">
      <ItemTemplate>
        <%# DataBinder.Eval(Container.DataItem,"au_id") %><br>
      </ItemTemplate>
    </asp:Repeater>
  </form>
</body>
</html>
```

b) What is multithreading and explain working of multithreading.

(Multithreading-2 Marks, Working -4 Marks and Example-2 Marks)

Multithreading is the process which allows .Net applications to do multiple tasks simultaneously. It improves performance and responsiveness of your applications. When two separate processes appear running simultaneously, they actually don't. In multithreading, the operating system divides the processor's execution time amongst different threads with a very small switching time so that they appear to run at a single time.

For managing multithreading .Net provides Thread class with various methods. Threading functionalities are available in System.Threading namespace.

A new thread object can be created as

```
Dim thread1 as new System.Threading.Thread(AddressOf <sub name>)
```

A Thread can be started as

```
thread1.start()
```

and can be stopped as



thread1.stop()

A thread can be taken to blocked/paused state with the help of following methods :

thread1.Sleep(100) – here thread1 can be blocked for 100 ms.

thread1.suspend() & thread1.resume() – another method to pause the thread is suspend(), which can be resumed back by resume() method.

It can be destroyed by thread1.Abort() method.

thread1.join() makes the current thread wait for another thread to finish.

Example :

```
Public Class Form1
```

```
    Dim i As Integer
```

```
    Dim i2 As Integer
```

```
    Dim thread As System.Threading.Thread
```

```
    Dim thread2 As System.Threading.Thread
```

```
    Private Sub countup()
```

```
        Do Until i = 10000
```

```
            i = i + 1
```

```
            Label1.Text = i
```

```
        Me.Refresh()
```

```
    Loop
```

```
End Sub
```

```
    Private Sub countup2()
```

```
        Do Until i2 = 10000
```

```
            i2 = i2 * 2
```

```
            Label2.Text = i2
```

```
            Me.Refresh()
```

```
    Loop
```

```
End Sub
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As  
        System.EventArgs) Handles Button1.Click
```

```
        thread = New System.Threading.Thread(AddressOf countup)
```



```
thread.Start()
```

```
End Sub
```

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button2.Click
```

```
thread2 = New System.Threading.Thread(AddressOf countup2)
```

```
thread2.Start()
```

```
End Sub
```

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Load
```

```
End Sub
```

```
End Class
```

c) Differentiate VB and VB.Net

(Any Eight point ,1- Mark each Point)

	VB	VB.NET
1.	Object based.	Object oriented.
2.	Can be used with only one language.	Provides common platform for different languages
3.	Uses VB runtime environment.	Uses .Net Common Language runtime
4.	Usually suitable for standalone applications	Can be used for both, stand alone as well as web applications.
5.	No concept of exception handling other than on error goto	Provides try catch mechanism for error handling.
6.	No multithreading procedures available	Multithreading available.
7.	COM component architecture	Uses assembly as its architecture.
8.	Less readymade controls for developing GUI as compared to VB.net	Variety of user friendly controls available for developing GUI
9.	Limited controls given for database connectivity	Database connectivity can be done in connected or disconnected way.



3. Attempt any **TWO** of the following:

Marks 16

a) Explain built in ASP.NET objects in details.

(List all built in ASP. NET object-1 Mark , All object description -5 Marks ,list method and properties-2 Marks)

Server Object:

- It is used to provide a set of essential properties and methods that enables a user to perform action on a server.
- Set the timeout so that long tasks can be completed.encode text so that characters Not allowed in a browser parameter can be used without incident.
- Transforms text that are html tags into the correct codes to display properly.
- Properties of server object is scripttimeout.
- Methods of server object are
 - 1) createobject
 - 2) execute
 - 3) htmlencode
 - 4) htmldecode
 - 5) mappath
 - 6) urlencode
 - 7) urldecode.

Response Object:

- Response object is used to send output to the user from the server.
- Halt the page being served in mid stream if required.
- Transfer the page being served.send cookies to the browser.
- Properties of response object are
 - 1)buffer,
 - 2)cachecontrol
 - 3)charset
 - 4)contenttype
 - 5)expires



6)expiresabsolute

7)status

8)isclientconnected

9)cookies.

- Methods of response object are

1) write,

2) redirect

3) flush

4) end.

Request Object:

- Retrieve any form values that a page submits to the script
- When a browser asks for a page from a server, it is called a request. The Request object is used to get information from a visitor.
- Retrieve any form values that a page submits to the script.
- Retrieve some vital server and script .
- Properties of request object are
 - 1)total bytes.
- Methods of request object are
 - 1)binary read
- collection of request object are
 - 1)server variables
 - 2)querystring
 - 3)charsetform
 - 4)cookies
 - 5)client certificate

Application Object:

- Use to store information that many scripts need to access globally.
- Can be used to store objects that many scripts will access.



- Application variables might be affected adversely if task switching occurs. It offers a LOCK and UNLOCK capability to force such operations to not be interrupted.
- Properties of application object are
 - 1) total bytes.
- Methods of application object are
 - 1) set
 - 2) add
 - 3) count
 - 4) lock
 - 5) unlock
 - 6) remove all
 - 7) remove
 - 8) removeat
- collection of request object are
 - 1) contents
 - 2) static objects

Session Object:

- Used to store data attached to a specific user/browser with minimal effort on the programmer's part.
- Can be used to store data on the server that can only be accessed by scripts but never is stored on the client's cookies.
- Properties of session object are
 - 1) timeout
 - 2) sessioID
 - 3) LCID
 - 4) codepage
- Methods of session object are
 - 1) binary read
- collection of session object are
 - 1) contents.removeall
 - 2) contents.remove



3)abandon

4) contents.removeat

- b) Describe with example, how database connectivity can be achieved in VB.Net using SQL or MS-Access with dataset and data reader codes to display content of database.

(Explanation of database connectivity -3 Marks, Program-5 Marks)

[Note : Other examples showing use of dataset and data reader also can be considered]

The Sql Connection Object is Handling the part of physical communication between the application and the SQL Server Database. An instance of the SqlConnection class in .NET Framework is supported the Data Provider for SQL Server Database The four Objects from the .Net Framework provide the functionality of Data Providers in ADO.NET. They are Connection Object, Command Object , DataReader Object and DataAdapter Object.

Connection Object The Connection Object provides physical connection to the Data Source. Connection Object connect to the specified Database and open a connection between the application and the Database . When the connection is established , SQL Commands may be executed, with the help of the Connection Object, to retrieve or manipulate data in the Database

Data Reader The DataReader Object is a stream-based , forward-only, read-only retrieval of query results from the Data Source, which do not update the data. DataReader Object in ADO.NET is a stream-based , forward-only, read-only retrieval of query results from the Data Source, which do not update the data. The DataReader cannot be created directly from code, they created only by calling the ExecuteReader method of a Command Object. **DataReader = Command.ExecuteReader()** DataReader Object provides a connection oriented data access to the data Sources. A Connection Object can contain only one DataReader at a time and the connection in the DataReader remains open and cannot be used for any other purpose while data is being accessed. When started to read from a DataReader it should always be open and positioned prior to the first record. The Read() method in the DataReader is used to read the rows from DataReader and it always moves forward to a new valid row, if any row exist. **DataReader.Read()** There are two types of DataReader in ADO.NET. They are SqlDataReader and the OleDbDataReader. The System.Data.SqlClient and System.Data.OleDb are containing these DataReaders respectively

DataSet stores many DataTables in VB.NET programs. A DataSet is conceptually a set of DataTables and other information about those tables. It is an abstraction that makes programs



simpler to develop. This is a container for multiple DataTables. You can use it to create XML. It is a useful abstraction for simplifying program.

[Note: same example can be considered for MS-access database, in that case instead of sqlclient, oledb can be used]

example using Dataset

Imports System.Data.SqlClient

Public Class Form1

Dim con As SqlConnection

Dim da As SqlDataAdapter

Dim ds As DataSet

Dim dv As DataView

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Handles Button1.Click

con = New SqlConnection("Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Documents
and Settings\vaishali\My Documents\mydb.mdf;Integrated Security=True;Connect
Timeout=30;User Instance=True")

con.Open()

da = New SqlDataAdapter("select * from student", con)

ds = New DataSet

da.Fill(ds, "student")

dv = New DataView(ds.Tables("student"))

DataGridView1.DataSource = dv

End Sub

End Class

Example using DataReader

Imports System.Data.SqlClient

Public Class Form2

Dim con As SqlConnection

Dim cmd As SqlCommand

Dim dr As SqlDataReader

Private Sub buttonshow_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Handles buttonshow.Click



```
con = New SqlConnection("Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Documents
and Settings\vaishali\My Documents\mydb.mdf;Integrated Security=True;Connect
Timeout=30;User Instance=True")
con.Open()
cmd = New SqlCommand("select * from student", con)
Dim b As Boolean
dr = cmd.ExecuteReader()
b = dr.Read()
If b Then txtrollno.Text = dr(0).ToString
txtname.Text = dr(1)
txtyear.Text = dr(2)
txtcourse.Text = dr(3)
End If
End Sub
End Class
```

rollno	sname
101	abc
102	xyz
103	pqr
104	lmn
105	hjk

show data

Example using DataReader

Imports System.Data.SqlClient

Public Class Form2

Dim con As SqlConnection



```
Dim cmd As SqlCommand
```

```
Dim dr As SqlDataReader
```

```
Private Sub buttonshow_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles buttonshow.Click
```

```
con = New SqlConnection("Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Documents  
and Settings\vaishali\My Documents\mydb.mdf;Integrated Security=True;Connect
```

```
Timeout=30;User Instance=True")
```

```
con.Open()
```

```
cmd = New SqlCommand("select * from student", con)
```

```
Dim b As Boolean
```

```
dr = cmd.ExecuteReader()
```

```
b = dr.Read()
```

```
If b Then txtrollno.Text = dr(0).ToString
```

```
txtname.Text = dr(1)
```

```
txtyear.Text = dr(2)
```

```
txtcourse.Text = dr(3)
```

```
End If
```

```
End Sub
```

```
End class
```

Form2

Rollno 101

Nmae abc

Year First

Course CO

Show data



c) What is .Net framework? Explain the component of .Net framework.

(Correct diagram – 2 Marks, component explanation -6 Marks)

- .NET framework is part of Microsoft .net technology. Its purpose is to create and run next generation applications and xml web services.
- The .NET Framework consists of
 - i. CLR,
 - ii. .NET Framework Class Library,
 - iii. Common Language Specification (CLS),
 - iv. a number of .NET languages, and
 - v. Visual Studio.NET
 - vi. Common Type system(CTS)

Common Language Runtime

A runtime provides services to executing programs. Traditionally, different programming environments have different runtimes. Examples of runtimes include the standard C library, MFC, the Visual Basic runtime, and the Java Virtual Machine (JVM). The runtime environment provided by .NET, the CLR, manages the execution of code and provides useful services. The services of the CLR are exposed through programming languages. The syntax for these services varies from language to language, but the underlying execution engine providing the services is the same. Not all languages expose all the features of the CLR. The language with the best mapping to the CLR is the new language C#. VB.NET, however, does an admirable job of exposing the functionality.

.NET Framework Class Library

The .NET Framework class library is huge, comprising more than 2,500 classes. All this functionality is available to all the .NET languages.

The library consists of four main parts:

1. Base class library (which includes networking, security, diagnostics, I/O, and other types of operating system services)
2. Data and XML classes
3. Windows UI
4. Web services and Web UI

**Common Language Specification**

An important goal of the .NET Framework is to support multiple languages. But all languages are not created equal, so it is important to agree upon a common subset that all languages will support. The CLS is an agreement among language designers and class library designers about those features and usage conventions that can be relied upon. CLS rules apply to public features that are visible outside the assembly where they are defined. (An assembly can be thought of as a logical EXE or DLL and will be discussed later in this chapter.) For example, the CLS requires that public names do not rely on case for uniqueness, because some languages are not case sensitive

Languages in .NET

A language is a CLS-compliant *consumer* if it can use any CLS-compliant type—that is, if it can call methods, create instances of types, and so on. (A type is basically a class in most object-oriented languages, providing an abstraction of data and behavior, grouped together.) A language is a CLS-compliant extender if it is a consumer and can also extend any CLS-compliant base class, implement any CLS-compliant interface, and so on. Microsoft itself is providing four CLS-compliant languages. VB.NET, C#, and C++ with Managed Extensions are extenders. JScript.NET is a consumer. Third parties are providing additional languages (more than a dozen so far). Active-State is implementing Perl and Python. Fujitsu is implementing COBOL. It should be noted that at present some of these languages are not .NET languages in the strict sense. For example, ActiveState provides a tool called PerlNET that will create a .NET component from a Perl class. This facility enables .NET applications to call the wealth of Perl modules, but it does not make Perl into either a consumer or an extender. For more information on PerlNET, see the book *Programming Perl in the .NET Environment* (Saltzman, Oberg), another book in The Integrated .NET Series



VB.NET	C#	Jscript NET	MORE .NET LANGUAGES
COMMON LANGUAGE SPECIFICATION(CLS)			
COMMON TYPE SYSTEM(CTS)			
.NET FRAMEWORK LASS LIBRARY(FCL)			
ASP.NET		WINDOWS FORMS	CONSOLE
ADO.NET		.NET REMOTING	
COMMON LANGUAGE RUNTIME(CLR)			
COMMON LANGUAGE INFRASTRUCTURE(CLI)			
OPERATING SYSTEM			

Fig. An overview of the .NET framework

The common Type System (CTS)

To support multiple programming in a Common Type system (CTS), which defines the entire core data types and data mechanisms used in .NET programs. This includes all numeric, string and Boolean value types. It also defines the object, the core data storage unit in .NET. The CTS divides all data objects into two buckets. The first bucket, called value types. The other bucket contains reference types. When you look in this bucket, you see a map that tells you where to find the actual data somewhere else in the computer's memory.

4. a) Attempt any THREE of the following :

Marks 12

i) Explain various ASP.Net states with neat diagram.

(Client side-2 Marks, Server side-2 Marks)

[Note: No specific diagram for asp.net states]

A new instance of the Web page class is created each time the page is posted to the server. In traditional Web programming, this would typically mean that all information associated with the page and the controls on the page would be lost with each round trip. For example, if a user enters information into a text box, that information would be lost in the round trip from the browser or client device to the server.



To overcome this inherent limitation of traditional Web programming, ASP.NET includes several options that help you preserve data on both a per-page basis and an application-wide basis. These features are as follows:

- View state
- Control state
- Hidden fields
- Cookies
- Query strings
- Application state
- Session state
- Profile Properties

View state, control state, hidden fields, cookies, and query strings all involve storing data on the client in various ways. However, application state, session state, and profile properties all store data in memory on the server. Each option has distinct advantages and disadvantages, depending on the scenario.

Client-Based State Management Options :

The following sections describe options for state management that involve storing information either in the page or on the client computer. For these options, no information is maintained on the server between round trips.

View State

The ViewState property provides a dictionary object for retaining values between multiple requests for the same page. This is the default method that the page uses to preserve page and control property values between round trips.

Control State

The ControlState property allows you to persist property information that is specific to a control and cannot be turned off like the ViewState property.

Hidden Fields

ASP.NET allows you to store information in a HiddenField control, which renders as a standard HTML hidden field. A hidden field does not render visibly in the browser, but you can set its properties just as you can with a standard control.



Cookies

A cookie is a small amount of data that is stored either in a text file on the client file system or in-memory in the client browser session. It contains site-specific information that the server sends to the client along with page output. Cookies can be temporary (with specific expiration times and dates) or persistent.

Query Strings

A query string is information that is appended to the end of a page URL. A typical query string might look like the following example:

`http://www.contoso.com/listwidgets.aspx?category=basic&price=100`

In the URL path above, the query string starts with a question mark (?) and includes two attribute/value pairs, one called "category" and the other called "price."

Server-Based State Management Options :

ASP.NET offers you a variety of ways to maintain state information on the server, rather than persisting information on the client. With server-based state management, you can decrease the amount of information sent to the client in order to preserve state, however it can use costly resources on the server. The following sections describe three server-based state management features: application state, session state, and profile properties.

Application State

Application state is stored in a key/value dictionary that is created during each request to a specific URL. You can add your application-specific information to this structure to store it between page requests.

Session State

Session state is similar to application state, except that it is scoped to the current browser session. If different users are using your application, each user session will have a different session state. In addition, if a user leaves your application and then returns later, the second user session will have a different session state from the first.

**ii) Explain how to create (data tables) SQL connection in ASP.Net**

(Explanation-1 Marks , Program 3-Marks)

The DataTable is a central object in the ADO.NET library. A DataTable has a collection of rows and columns represented by the Rows Property and Columns property. A DataTable can be built by programming, consisting records by adding DataColumn to the columns.collection and

DataRow to the rows collection. Other objects that use the DataTable include the DataSet and the DataView. When accessing DataTable objects, they are conditionally case sensitive. For example, if one DataTable is named “myTable” and another is named as “MyTable” a string used to search for one of the tables is regarded as case sensitive. However if “myTable exists and MyTable does not the search string is regarded as case sensitive.

To add new DataTable one can follow syntax below:

```
Dim dt As DataTable = New DataTable (“AWT”)
```

```
<%@Page Language=”VB” AutoEventWireup=”false” CodeFile= “Default.aspx.vb”
```

```
Inherits=”_Default” %>
```

```
<%@Import Namespace=”system.data.sqlclient” %>
```

```
<!DOCTYPE html PUBLIC”-//W3C//DTD XHTML 1.0
```

```
Transitional//EN””http://www.w3.org/TR/xhtml1-transitional.dtd”>
```

```
<html xmlns=”http://www.w3.org/1999/xhtml”>
```

```
<head runat=”server”>
```

```
<title>Untitled Page</title>
```

```
</head> <script runat=”server”>
```

```
Sub page_load(ByVal sender As Object, ByVal e As EventArgs) Handles Me.Load
```

```
Dim con as new SqlConnection("Data
```

```
Source=.\SQLEXPRESS;AttachDbFilename=C:\Documents and Settings\vaishali\My
```

```
Documents\mydb.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True")
```

```
Con.open()
```

```
Dim Table1 As DataTable
```

```
Table1=New DataTable(“Customers”)
```

```
Dim Row1, Row2, Row3 As DataRow
```

```
Try
```

```
Dim id As DataColumn = New DataColumn(“id”)
```



```
Id.DataType = System.Type.GetType("System.String")
Table1.Columns.Add(id)
Dim Product As DataColumn = New DataColumn("Product")
Product.DataType = System.Type.GetType("System.String")
Table1.Column.Add(Product) Row1 = Table1.NewRow()
Row1.Item("id") = "A101"
Row1.Item("Product") = "Sugar"
Table1.Rows.Add(Row1)
Row2 = Table1.NewRow()
Row2.Item("id") = "B333"
Row2.Item("Product") = "Soap"
Table1.Rows.Add(Row2)
Row3 = Table1.NewRow()
Row3.Item("id") = "C450"
Row3.Item("Product") = "fruits"
Table1.Rows.Add(Row3)
Catch
End Try
Dim ds As New DataSet
ds = New DataSe
t ds.Tables.Add(Table1)
dg1.DataSource = ds
dg1.DataBind()
End Sub
</script>
<body>
<form id="form1" runat="server">
<div>
<h2>Shopping Mall</h2>
<asp:DataGrid ID="dg1" runat="server"></asp:DataGrid>
</div>
</form>
```



</body>

</html>

iii) Explain Synchronization in Threads:*(Multithreading -1 Marks, Synchronization explanation-2 Marks and Program -1 Marks)***Multithreading**

It is a feature provided by the operating system that enables your application to have more than one execution path at the same time. We are all used to Windows' multitasking abilities, which allow us to execute more than one application at the same time. While writing in Microsoft Word, listening to favorite songs in WinAmp and downloading a new song using the Internet Download Manager and document is sent for printing. In a similar manner, we may use multithreading to run different methods of our program at the same time. The .NET Framework, and thus VB.NET provides full support for multiple execution threads in a program. You can add threading functionality to your application by using the System.Threading namespace. A thread in .NET is represented by the System.Threading.Thread class. We can create multiple threads in our program by creating multiple instances (objects) of this class. A thread starts its execution by calling the specified method and terminates when the execution of that method gets completed. We can specify the method name that the thread will call when it starts by passing a delegate of the ThreadStart type in the Thread class constructor. The delegate System.Threading.ThreadStart may reference any method which has the void return type and which takes no arguments.

synchronization Threads:

- There may be a situation when two threads are working with the same data and you might not want the second thread to work with that data until the first thread is finished with it.
- So in this situation synchronization is necessary. You can achieve synchronization by, SyncLock statement and join method. SyncLock gains an exclusive lock to be an object reference that is passed to it.
- By gaining this exclusive lock you can ensure that multiple threads are not accessing shared data or that the code is executing on multiple threads.
- This code will provide the locking functionality:



Program:

Imports System

Imports System.Threading

Imports system.data

Imports system.text

Module module1

Dim text as new stringbuilder()

Public sub main()

Dim firstThread As New Thread(New ThreadStart(AddressOf Fun1))

Dim secondThread As New Thread(New ThreadStart(AddressOf Fun2))

firstThread.Start()

secondThread.Start()

firstThread.join()

secondThread.join()

console.writeline("text is: {0} {1}",vbCrLf,text.ToString()))

consolr.read()

End Sub

Public Sub Fun1()

Dim i As Integer

Synclock text

For i = 1 To 20

Thread.sleep(10)

Text.append(i.ToString()+" ")

Next

End Synclock

End Sub

Public Sub Fun2()

Dim i As Integer

Synclock text

For i =2 1 To 30

Thread.sleep(2)

Text.append(i.ToString()+" ")



Next

End Synclock

End Sub

End Module

iv) What is assembly? Explain private and shared assembly.

(Assembly -2 Marks, Private -1 Mark and Shared-1 Mark)

Assemblies:

- 1) It is collection of types and resources that are built to work together and form a logical unit of functionality.
 - 2) Assembly refers to both a logical construct and set of physical files.
 - 3) Purpose of assembly is to specify a logical unit or building blocks for .net applications that encapsulation certain properties.
 - 4) .Net applications consist of one or more assemblies. logically speaking an assembly is just a set of specification as follows.
 - i) An assembly specifies (MSIL)code that is associated with the assembly. This is associated with the assembly. This code lies in a portable executable (PE) file.
 - ii) An assembly specifies security permission for itself.
 - iii) An assembly specifies a list of data types and provides scoping for those types.
 - iv) An assembly specifies rules for resolving external types and external references.
 - v) An assembly specifies which of its parts are exposed outside the assembly and which are private to the assembly itself.
 - vi) With the help of assembly we eliminate DLL Hell problem.
 - vii) There are two types of assemblies we can create.
- Private:
 - Can be used by only one application.
 - It is default style of on assembly in .NET
 - It deployed in the application's own directory.
 - It does not required strong name, but they need version number and assembly name.
 - Assemblies are self describing no registry entries are required.
 - Shared assemblies :
 - It is used by multiple applications on target machine.

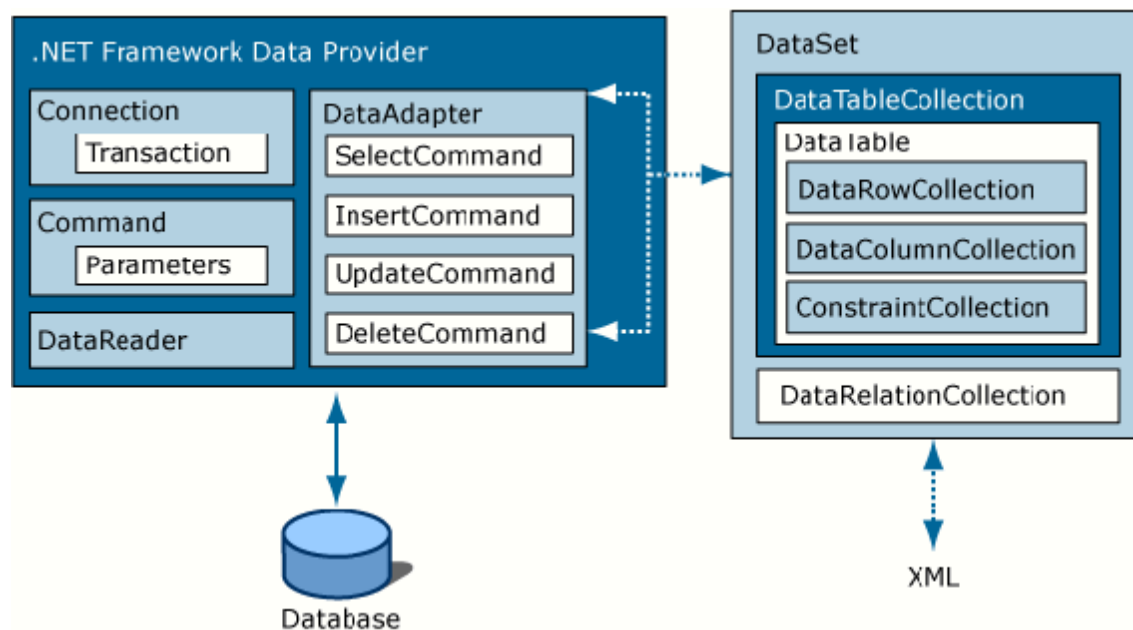
- All .NET assemblies which come with .NET framework are shared so you can use those assemblies in your can use those assemblies in your project.
- Shared assembly must have a globally unique name which is known as “Strong name”
- Shared assemblies are stored in GAC(global assembly cache).Using gacutil.exe
- We can register multiple versions of the same shared assembly in GAC.
- We can register multiple versions of the same shared assembly in GAC.
- We required to register assembly to make strong name in GAC.

b) Attempt any ONE of the following :

Marks 6

i) Draw and explain ADO.NET architecture.

(Diagram -2 Marks and Explanation of all objects of diagram-4 Marks)



The architecture of ADO.net in which data retrieved from database can be accessed even when connection to database was closed is called as disconnected architecture.

ADO.NET Components

- There are two components of ADO.NET that you can use to access and manipulate data:

.NET Framework data providers

The DataSet

**ADO.NET objects**

- The four Objects from the .Net Framework provide the functionality of Data Providers in ADO.NET. They are

- **Connection Object:**

It provides a connection to the database or data source. The Connection Object provides physical connection to the Data Source. Connection Object connects to the specified Database and opens a connection between the application and the Database. When the connection is established, SQL Commands may be executed, with the help of the Connection Object, to retrieve or manipulate data in the Database.

- **Command Object:**

It is used to execute a command and enables access to database commands to return data, modify data, run stored procedures, and send or retrieve parameter information.

- **DataReader object:**

It provides a forward-only, read only, connected recordset. The DataReader Object is a stream-based, forward-only, read-only retrieval of query results from the Data Source, which do not update the data. DataReader Object in ADO.NET is a stream-based, forward-only, read-only retrieval of query results from the Data Source, which do not update the data. The DataReader cannot be created directly from code, they are created only by calling the ExecuteReader method of a Command Object. ***DataReader = Command.ExecuteReader()*** DataReader Object provides a connection oriented data access to the data Sources. A Connection Object can contain only one DataReader at a time and the connection in the DataReader remains open and cannot be

used for any other purpose while data is being accessed. When started to read from a DataReader it should always be open and positioned prior to the first record. The Read() method in the DataReader is used to read the rows from DataReader and it always moves forward to a new valid row, if any row exists. ***DataReader.Read()*** There are two types of DataReader in ADO.NET. They are SqlDataReader and the OleDbDataReader. The System.Data.SqlClient and System.Data.OleDb are containing these DataReaders respectively.

- **DataAdapter object:**

DataAdapter populates a disconnected DataSet with data and performs update and it provides DataAdapter provides the bridge between the **DataSet** object and the data source. The



DataAdapter uses **Command** objects to execute SQL commands at the data source to both load the **DataSet** with data, and reconcile changes made to the data in the **DataSet** back to the data source.

.NET Framework Data Providers

- The .NET Framework Data Providers are components that have been explicitly designed for data manipulation and fast, forward-only, read-only access to data. The

- **DataSet**

The ADO.NET DataSet is explicitly designed for data access independent of any data source. As a result, it can be used with multiple and differing data sources, used with XML data, or used to manage data local to the application. The **DataSet** contains a collection of one or more **DataTable** objects made up of rows and columns of data, as well as primary key, foreign key, constraint, and relation information about the data in the **DataTable** objects. The following diagram illustrates the relationship between a .NET Framework data provider and a **DataSet**.

DataSet stores many DataTables in VB.NET programs. A DataSet is conceptually a set of DataTables and other information about those tables. It is an abstraction that makes programs simpler to develop. This is a container for multiple DataTables. You can use it to create XML. It is a useful abstraction for simplifying programs

- **XML and ADO.NET**

ADO.NET leverages the power of XML to provide disconnected access to data. ADO.NET was designed hand-in-hand with the XML classes in the .NET Framework; both are components of a single architecture. ADO.NET and the XML classes in the .NET Framework converge in the **DataSet** object. The **DataSet** can be populated with data from an XML source, whether it is a file or an XML stream. The **DataSet** can be written as World-Wide Web Consortium (W3C) compliant XML, including its schema as XML Schema definition language (XSD) schema, regardless of the source of the data in the **DataSet**. Because of the native serialization format of the **DataSet** is XML, it is an excellent medium for moving data between tiers, making the **DataSet** an optimal choice for remoting data and schema context to and from an XML Web services.

**ii) Explain any two server components.**

(Explanation of each component -3 Marks)

1) AdRotator :

- Adrotator Use the AdRotator control to display a randomly or sequentially selected advertisement banner on the Web page. The displayed advertisement changes whenever the page refreshes. Advertisement information is stored in a separate XML file. The XML file
- allows you to maintain a list of advertisements and their associated attributes. Attributes included are as follows
- The AdRotator is a special control in ASP.NET that is used to display flashing banner ads. The control is capable of displaying ads randomly or sequentially as set by the developer.
- Each time is refreshed or reload a new ad can be displayed to the used. Also, we can assign priorities in such a way that certain ads are displayed frequently than others. Properties of AdRotator Control: - **AdvertisementFile**: This file holds the image files which are to be displayed as a banner. **KeywordFilter**: - The „KeywordFilter“ property enables you to filter advertisement using a keyword. If your Advertisement file contains different kinds of ads (banner, leaderboard, skyscraper etc.),
- you can use this property to filter out different ads on different sections of the page. If you observe, the ads.xml file also contains a property called „Keyword“ which binds that ad with the AdRotator that contains the same KeywordFilter, in our case „small“ .
- Example: - Code for ads.XML

```
<?xml version="1.0" encoding="utf-8" ?> <Advertisements> <Ad>
  <ImageUrl>~/images/dnc.jpg</ImageUrl>
  <NavigateUrl>http://www.dotnetcurry.com</NavigateUrl>
  <AlternateText>DotNetCurry Home Page</AlternateText>
  <Impressions>40</Impressions>
  <Keyword>small</Keyword>
</Ad>
<Ad>
  <ImageUrl>~/images/ssc.jpg</ImageUrl>
  <NavigateUrl>http://www.sqlservercurry.com</NavigateUrl>
  <AlternateText>SQL Server Curry Home Page</AlternateText>
  <Impressions>20</Impressions>
```



```
<Keyword>small</Keyword>
</Ad>
<Ad>
<ImageUrl>~/images/writeforus.jpg</ImageUrl>
<Width>300</Width>
<Height>50</Height>
<NavigateUrl>http://www.dotnetcurry.com/writeforus.aspx</NavigateUrl>
<AlternateText>dotnetcurry.com Write For Us</AlternateText>
<Impressions>40</Impressions>
<Keyword>small</Keyword>
</Ad>
</Advertisements>
```

- **Code for ad.aspx**

```
<form id="form1" runat="server">
<div>
<asp:ScriptManager ID="ScriptManager1" runat="server" />
<asp:Timer ID="Timer1" Interval="2000" runat="server" />
<asp:UpdatePanel ID="up1" runat="server">
<Triggers> <asp:AsyncPostBackTrigger ControlID="Timer1" EventName="Tick" />
</Triggers> <ContentTemplate> <asp:AdRotator id="AdRotator1"
AdvertisementFile="~/App_Data/Ads.xml" KeywordFilter="small" Runat="server" />
</ContentTemplate>
</asp:UpdatePanel>
</div>
</form>
```

2) Content linking :

The content linking component manages a list of URLs so that you can treat the pages in your web site like the pages in a book. The content Linking Component can be used to automatically generate and update tables of contents and navigational links to previous and subsequent web pages. This is ideal for application such as online newspapers and forum messages listings.

Structure Of Content Linking Component: - The Content linker has two parts:



The Index file: An ASCII text file that holds a list of .asp pages in the order that they should be presented to the viewer. The index file can easily be changed to add new pages, revise the order of pages or substitute newer pages for older pages

The ASP pages: - Each .asp file contains two parts. The first part consists of the normal content of the page, such as the text and graphics. The second part is the ASP code that uses the content Linker to give the user options for the next, first, previous and last page.

Content Linking Index File: - The content linking component depends on an external file kept on the server for its list of navigable content. An index file will be text file that looks like as follows

HTMLPage.html	HTMLPage1	Intro
HTMLPage2.html	HTMLPage2	Variables
HTMLPage3.html	HTMLPage3	Procedures

Each line in the above file includes three things The relative URL of the content page The Display text A Comment describing the content

Code in Default.aspx file

```
<% @Page Language="VB" AutoEventWireup="false" CodeFile="Default.aspx.vb"
Inherits="_Default" %> <html xmlns="http://www.w3.org/1999/xhtml">
  <head runat="server"> <title>Untitled Page</title> </head> <script runat="server">
<body>
  <Form id="forma"runat="server">
    <h3>Contents List :</h3>
    <ul>
<%Dim nl = Server.createObject("MSWC.NextLink")
  Dim intCount = nl.GetListCount("textFile.txt")
  Dim intLoop
  For intLoop = 1 To intCount%>
<li>
  <a href="<% = nl.GetNthURL("TextFile.txt" , intLoop)%>">
    <% =n=nl.GetNthDeascription("TextFile.txt",intLoop)%>
  </a>
</li>
<% Next%>
```



</form></body></html>

Program:

<!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><title>Untitled Page</title></head>

<body><h1>page1</h1></body></html>



3) Browser capabilities :

ASP.NET determines the capabilities of the browser with which a user is browsing your site using a feature called 'browser capabilities'. Following are the properties of the Browser capabilities

There are two ways to implement browser capabilities

1. Using sever object (used in ASP and ASP.net both)
2. Using request object (Used in ASP.NET)

When a client requests a page from the server, the HTTP header includes a user agent ASCII string that specifies the browser software name and version. Browser capabilities component search for this string in the Browser capabilities component and the server adopts the properties of the browser.

Property/Method	Type	Purpose
ActiveXControls	Bool	Support for ActiveX controls
AOL	Bool	America Online browser
BackgroundSounds	Bool	Support for background sounds
Beta	Bool	Beta version
Browser	String	Name of the browser
ClrVersion	Version	Version number of the latest .NET Framework installed on the client
Cookies	Bool	Support for cookies (though not whether cookies are disabled)
Crawler	Bool	Client is a search engine Web crawler
Frames	Bool	Support for frames
JavaApplets	Bool	Support for Java applets
JavaScript	Bool	Support for JavaScript blocks
MajorVersion	Integer	Major version of the browser
MinorVersion	Integer	Minor version of the browser
MSDomVersion	Version	Version of the Microsoft HTML (MSHTML) Document Object Model (DOM) is supported by the client
Platform	String	Operating system running the browser
Tables	Bool	Support for tables
Type	String	Browser name and major version number
VBScript	Bool	Support for VBScript
Version	String	Full version number of the browser

**Program:****Note: Any four properties can be considered in description and example****Example of Browser Capabilities**

```
<%@ Page Language="vb" %>
<%@ Import Namespace="System.Web" %>

<html><head>
<title>Determining Browser Capabilities in ASP.NET: Demo</title></head>
<body>
    <%

        Dim objBrowser As HTTPBrowserCapabilities
        objBrowser = Request.Browser

    %>
    <p align = center>
    <b>Determining Browser Capabilities in ASP.NET: Demo</b></p>
    <TABLE>
    <% With objBrowser %>
    <tr>
    <td><b>Browser type:</b></td><td><%=.Type %>
    </td></tr>
    <tr>
    <td><b>Browser version:</b></td><td><%=.Version%>
    </td></tr>
    <tr>
    <td><b>Major version:</b></td><td><%=.MajorVersion%>
    </td></tr>
    <tr>
    <td><b>Minor version:</b></td><td><%=.MajorVersion%></td></tr>
    <tr>
    <td><b>Platform:</b></td><td><%=.Platform%></td>
    </tr>
    <tr>
    <td><b>This is a Win16 Platform:</b></td><td><%=iif(.Win16, "Yes", "No")%></td>
    </tr>
    <tr>
    <td><b>This is a Win32 Platform:</b></td><td><%=iif(.Win32, "Yes", "No")%></td>
    </tr>
    <tr>
    <td><b>Tables Supported:</b></td><td><%=iif(.Tables, "Yes", "No")%></td>
    </tr>
    <tr>
    <td><b>Frames Supported:</b></td><td><%=iif(.Frames, "Yes", "No")%></td>
    </tr>
    <tr>
    <td><b>Background Sounds Supported:</b></td><td><%=iif(.BackgroundSounds, "Yes",
    "No")%></td>
    </tr>
    <tr>
    <td><b>Cookies Supported:</b></td><td><%=iif(.Cookies, "Yes", "No")%></td>
    </tr>
```



```

<tr>
<td><b>Java Script Supported:</b></TD><td> <%=iif(.JavaScript, "Yes", "No")%></td>
</tr>
<tr>
<td><b>Java Applets Supported:</b></TD><td> <%=iif(.JavaApplets, "Yes", "No")%></td>
</tr>
<tr>
<td><b>VB Script Supported:</b></TD><td> <%=iif(.VBScript, "Yes", "No")%></td>
</tr>
<tr>
<td><b>Active X Controls Supported:</b></TD><td> <%=iif(.ActiveXControls, "Yes",
"No")%></td>
</tr>
<tr>
<td><b>AOL Client:</b></TD><td> <%=iif(.AOL, "Yes", "No")%></td>
</tr>
<tr>
<td><b>Browser is a Search-Engine Crawler:</b></TD><td> <%=iif(.Crawler, "Yes",
"No")%></td>
</tr>
<tr>
<td><b>Browser is a Beta Version:</b></TD><td> <%=iif(.Beta, "Yes", "No")%></td>
</tr>
<tr>
<td><b>Channel Definition Format Supported:</b></TD><td> <%=iif(.CDF, "Yes",
"No")%></td>
</tr>
<%End With %>
</table>

</body>
</html>

```

5. Attempt any **FOUR** of the following:

Marks 16

a) List and explain any two data bound controls.

(List-1 Mark, any two data bound controls – 1 ½ marks each)

Data bound web server controls are controls that can be bound to a data source control to make it easy to display and modify data in your web application.

List of data bound controls:

1. Repeater Control
2. DataList Control



3. FormView Control
4. GridView Control
5. DetailView Control

1. Repeater control

- The repeater control displays data items in a repeating list. Similar to DataList, the content and layout of list items in repeater is defined using templates.
- At a minimum, every repeater must define an item Template: Unlike DataList, repeater has no built-in layout or styles.
- You must explicitly declare all HTML layout, formatting and style tags within the templates of the control.

Example:

```
<html>
<body>
<form id="Form1"runat="server">
<asp:Repeater ID="rptStudents"Runat="Server">
<Item Template><%#Container.DataItem("Student_name")%></Item Template>
</asp:Repeater>
</form>
</body>
</html>
```

2. DataList control

The dataList control displays data items in a repeating list and optionally supports selecting and editing the items.

the content and layout of list items in DataList is defined using templates. The following table describes those templates. The following table describes those templates.

Example : Creating DataList in a tabular form.

```
<form id="form1"runat="server">
<div>
<% Dim con As sqlconnection
Dim dtr As sqlDataReader
Dim cmd As sqlcommand
```




```
con=New SqlConnection("data Source=Vaishali\\sqlExpress;Initial Catalog=tempdb;integrated
Security=True")
cmd=New SqlCommand(";select* From Employee",con)
con.Open( )
dtr=cmd.ExecuteReader()
datalist1.DataSource=dtr
datalist1.DataBind()
dtr.close()
con.close()%>
</div>
<asp:DataList ID="dataList1"runat="server">
<header template>
<tr>
<th align="right">Name</th>
<th align="right">Id</th>
</tr>
</header template>
<Item template>
<tr>
<td><%#Container.dataItem("Name")%></td>
<td><%#Container.DataItem.DataItem("Id")%></td>
</tr>
</Item template>
</asp:DataList>
</form>
</body>
</html>
```

3. FormView Control

- Fromview is a data bound user interface control that render a single record at a time from its associated data source, optionally providing paging buttons to navigate between records.
- Binding to data source control,such as SqlDataSource and ObjectDataSource.
- The Features of FormView are as follows:



1. Built-in inserting, updating and deleting capabilities.
2. Built-in paging capabilities.
3. Customizable appearance through user-defined templates, themes and styles.

Example:

Default.aspx

```
<%@Page Language="VB" CodeFile="Default.aspx.vb" Inherits="_Default"%>
<html xmlns=http://www.w3.org/1999/xhtml>
<head runat="server"> <title>Untitled page</title>/>
<body>
<form id="Form1" runat="server">
<asp:Label ID="label1" runat="server" text="FormViewConrol"/><br/>
<asp:FormViewID="FormView1" runat="server" Allowpaging="True" BorderColor="Black"
BorderWidth="1px" dataKeyNames="Student_no" DataSourceID="sqlDataSource1">
<insertItemTemplate>
    Student_no:<asp:TextBoxID="TextBoxID" runat="server" Text="<%=Eval(Employee_ID)"
)%"/><br/>
    Student_name:<asp:TextBoxID="TextBox2" runat="server
text="<%=Bind("Employee_name")%"/><br/>
<insertitemtemplate>
<itemTemplate>
    Student_no:<asp:Label ID="Lable3" runat="server" text="<%=Eval
("Student_no")%"/><br/>
    Student_name:<asp:Label ID="Lable4" runat="server" text="<%=Bind
("Student_no")%"/><br/>
</Item Template>
</asp:FromView>
<asp:sqlDataSource ID="sqldataSource1" runat="Server" ConnectionString="dataSource=.;initial
catalog=northwind;user ID=sa;password=sa" SelectCommand="SELECT *FROM
Student" providename="System.Data.SqlClient"/>
</from></body></html>
```



4. GridView Control

- The GridView control is used to display the values of a data source in a table. Each Column represents a field, while each row represents a records. It is a successor of datagrid control.
- The GridView control supports the Following features:
 1. Binding to data source control, such as sqlDataSource.
 2. Built-in sorting, updating, deleting, paging capabilities.
 3. Built –in row selecting capabilities.
 4. Multiple key fields.
 5. Multiple data field for the hyperlink columns.
 6. Customizable appearance through themes and styles.

Example:

```
<%@Page Language="VB"
AutoEventWireup="false"codeFile="Default.aspx.vb"Inherits="_Default"%>
<!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>Untitled page</title></head>
<body>
<form id="Form1"runat="server">
<asp:GridView ID="GridView1"dataSourceID="sqlDataSource1"runat="server"/>
<asp:sqlDataSourceID="sqlDataSource1"Runat="Server"ConnectionString="Data
Source=.;Initial Catalog=northwind;UID=sa;Password=sa"SelectCommand="SELECT*
FROM Student "ProviderName="System.Data.SqlClient"/>
</form></body></html>
```

5. DetailView control

It is a data-bound user interface control that renders a single record at a time from its associated data source,optionally providing paging buttons to navigate between records.

It is similar to the form view of an Access database and is that DetailsView has a built-in tabular rendering, whereas form View requires a user-defined template for its rendering.

Example:



<% @ Page

Language="VB" AutoEventWireup="false" CodeFile="Default.aspx.vb" Inherits="_Default"%>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title>Untitled Page</title>

</head>

<body.

<form id="Form1" runat="server">

<asp:DetailsView ID="DetailsView1" runat="server" AllowPaging="true" AutoGenerate

Rows="false" DataSourceID="sqlDataSource1" Header Text="Rollno SortExpression="Rollno"/>

</Field>

</asp:DetailView>

</asp:DetailView>

<asp:sqlDtasource ID="SqlDataSource1" runat="server" ConnectionString="data source=lab-311

\SQLEXPRESS;Initial catalog=tempdb;Integrated Security=true"

ProviderName="System.Data.SqlClient" SelectCommand="SELECT* FROM[student]">

</asp:sqlDataSource>

</body>

</html>

**b) Explain different data types available in VB.Net.**

(Any 8 Data types – ½ mark each)

The data types available in VB.NET along with their size, type and description are summarized in below table.

Data Type	Size	Description	Type
Byte	1	8-bit unsigned integer	System.Byte
Char	2	16-bit Unicode characters	System.Char
Integer	4	32-bit signed integer	System.Int32
Double	8	64-bit floating point	System.Double
Long	8	64-bit signed integer	System.Int64
Short	2	16-bit signed integer	System.Int16
Single	4	32-bit floating point	System.Single
String	Varies	Non-Numeric Type	System.String
Data	8	Data Type	System.Data
Boolean	2	Non-Numeric Type	System.Boolean
Object	4	Non-Numeric Type	System.Object
Decimal	16	128-bit floating point	System.Decimal

c) What is XML File? Explain Read XML. Write XML with example.

(Explanation of XML File-1 Mark, Read XML with example-1 ½ Mark, Write XML with example- 1 ½ Mark)

XML is a meta-markup language which means that it lets us create our own markup language (our own tags). XML is popular for the following reasons:

1. XML allows easy data exchange.
2. It allows us to customize markup languages.
3. Makes the data in the document self-describing.
4. XML allows for structured and integrated data.

Example: Save the following code with a .XML extension.

```
<?xml version="1.0"encoding="UTF-8"?>
```



<DOCUMENT>

<WELCOME>Hello World

</WELCOME>

</DOCUMENT>

Read XML Files:

- Read XML is a method used to fill a Dataset with data from XML. It can read data from a file, a stream or an XmlReader.
- The syntax of this method is as follows:
Public Function ReadXml(fileName As String) As XmlReadMode
Example:

Code for XMLFile1.xml

```
<?xml version="1.0"encoding="UTF-8"?>
```

```
<Table>
<Product>
<Product_id>1</Product_id>
<Product_name>Product 1</Product_name>
<Product_price>1000</Product_price>
</Product>
<Product>
<Product_id>2</Product_id>
<Product_name>Product 2</Product_name>
<Product_price>2000</Product_price>
</Product>
<Product>
<Product_id>3</Product_id>
<Product_name>Product 3</Product_name>
<Product_price>3000</Product_price>
</Product>
<Product>
<Product_id>4</Product_id>
<Product_name>Product 4</Product_name>
<Product_price>4000</Product_price>
</Product>
</Table>
```

Code for Windowapplication1

```
Public class form3
```

```
Private Sub Button1_Click(By Val sender As System.Object,ByVal e As System.Event Args)
```

```
handles Button1.Click
```



```
Dim d1 As New DataSet()
d1.ReadXml("C:\Users\Temporary Projects\WindowsApplication1\XMLFile1.xml")
DataGrid1.DataSource=d1
DataGrid1.DataMember="product"
End Sub
End Class
```

Write XML Files:

- WriteXML is a method used to write the XML representation of the DataSet into a file, a stream, an XmlWriter object or a string. Its syntax is:

```
Public Sub WriteXml(filename As string)
```

Example:

```
Public class form3
Private Sub Form3_Load(By Val sender As System.Object,ByVal e As System.Event Args)
handles MyBase.Load
Dim ds As New DataSet("Contacts")
Dim dt As New DataTable("Contacts")
Dim dr As DataRow
dt.Columns.Add("FirstName",System.Type.GetType("System.String"))
dt.Columns.Add("LastName",System.Type.GetType("System.String"))
ds.Tables.Add(dt)
dr=ds.Tables("Contact").NewRow
dr(FirstName)="vinay"
dr(LastName)="manik"
dt.Rows.Add(dr)
ds.WriteXml("c:\XMLFile2.xml")
MsgBox("XMLFile2.xml is created on c drive")
End Sub
End Class
```



- d) Create a console application that will show the comparison between two values.

(Correct logic – 2 Marks, Correct syntax- 2 Marks)

[Note: any other logic should also be consider)

```
Imports System.Console
Module Module1
Sub Main ()
Dim no1, no2 As Integer
WriteLine("Enter any two numbers")
no1 = Val(ReadLine())
no2 = Val(ReadLine())
If no1 = no2 then
WriteLine(no1 & " is equal to " & no2)
ElseIf no1 < no2 then
WriteLine(no1 & " is less than " & no2)
Else
WriteLine(no1 & " is greater than " & no2)
End If
Read ()
End Sub
End Module
```

- e) How ASP.net is different from ASP?

(Any 8 point, 1/2 mark each)

ASP.NET is not simply the next version of asp it's a completely redesigned technology that takes the best aspects of ASP and merge them with the power of pure Object-Oriented Programming (OOP), a powerful development framework, to give it a vast range of functionality and the advantages of a fully complied execution environment.



ASP	ASP.NET
ASP uses VBScript	ASP.NET uses languages like C#, VB.NET, etc.
It uses interpreter when page is executed	It uses Compiler, i.e code is compiled to MSIL
It is not language independent	ASP.NET is a part of .Net framework and so it is language independent.
ASP use ADO to connect with database.	ASP.NET uses ADO.NET for database connectivity.
ASP does not allow to separate the code from page content.	ASP.NET allows to separate the code from page content by placing all the code in a separate file.
User input validation is hard to achieve by code in ASP	ASP.NET provides a set of server controls that checks the user input.
Depends on developer to implement own security.	ASP.NET features built-in security configuration in its application setting files. It has two type of Security, viz window based and web based
Server controls are not used in ASP.	Server control concept is introduced in ASP.NET
ASP does not provide User Input validation.	ASP.NET provides a set of server controls that check whether the information users have entered is in correct form.
Event-Driven Programming: This type of event driven programming was not available with classic ASP.	Event-Driven Programming: This type of event driven programming is available with classic ASP.NET.

6. Attempt any FOUR of the following:**Marks 16****a) Write a program to send attachment with mail.***(Procedure – 2 Marks, Code example – 2 Marks)*

A screen can be designed to fill the parameters such as from. To, cc, bcc subject message body required in email. (***)Layout of the screen can be drawn(***)Ensuring that SMTP server is set to send Emails outside, following code can be written in code behind:

```
Imports System.Web.Mail
Dim Mail as Object
Set Mail =Server.CreateObject("CDONTS.NewMail")
Mail.To=Request.form("malfrom")
Mail.From= Request.form("mailto")
```



```
Mail.Cc= Request.form("mailcc")
Mail.Bcc= Request.form("mailBcc")
Mail.Subject= Request.form("mailsub")
Mail.Body= Request.form("mailbody")
Mail.AddAttachment("c:\mydocuments\test.txt")
Mail.send
End Sub
```

b) How exception handling is done in OOP.

(Explanation – 4 marks)

Structured exception Handling

- Structured exception handling is code designed to detect and respond to errors during execution by combining a control structure, (similar to select Case or while) with executions, protected blocks of code and filters.
- using the Try....Catch....Finally statement, you can protect blocks of code that have the potential to raise errors. the Try block encloses an expression that might generate an exception.
- The Try block has one or more associated Catch statements, which filter exceptions and match them with a particular action.
- If the exception occurs, the runtime stop normal execution and starts searching for a Catch block that can catch the exception, based on its type.
 - i. If an appropriate Catch block is not found in the immediate procedure, the runtime proceeds up the call stack searching for the calling procedure.
 - ii. if an appropriate catch block is not found there, the runtime searches for the procedure that called procedure and so on, until it finds an appropriate Catch block.
 - iii. If a Catch block is found, the exception is considered to have been caught and execution restarts, beginning with the body of the Catch block.
 - iv. If no Catch block is found, an error message appears and application is closed.
- The following code shows the structure of a simple Try-Catch statement:

```
Try
    'Starts a structured execution handler
Catch[option filter]
    'This code runs if the statements listed in the try block
```



‘fail and the filter on the Catch statement is true

End try

- The code in the Try block always executes. If an error occurs during execution of any of the code in the try section, Visual Basic examines that occurred to the name of the exception specified in the Catch statement and then transfers control to the first line of code in the Catch statement. If no appropriate Catch block is found, an error is produced.
- The Finally block is an optional block of code, which if included, is always executed.
- The Finally block is most often used to clean up resources used in your code when a method fails.
- The following code prods an example of using a Finally block to close an open file.

Example:

The Try block always executes.

If there is a FileNotFoundException error, code in the first Catch block is execution, followed by the code in the the Finally block.

If there is some other type of error, the code in the second Catch block is executed, followed by the code in the Finally block.

If there is no error, then code in the Finally block executes after the code in the try block.

Try

```
fs=NewFileStream("data.txt",FileMode.Open)
```

```
Catch ex As FileNotFoundException
```

```
    MessageBox.Show("Data File Missing")
```

```
Catch ex As Exception
```

```
    MessageBox.Show(ex.Message)
```

finally

```
    If Not(fs Is Nothing)Then
```

```
        fs.close()
```

End Try

c) Explain Data Binding advantages.

(Any 4 advantages- 1 Mark each)

- data binding means binding controls to information stored in a data source. This implies binding any control property to almost any kind of data store.



- A data store can be simple as a public property on a page, or as complex as a database stored on a server
- This broad choice among data stores provides high flexibility, and thus enables us to bind a control to any data store based on our need.
- The best example of a data binding is Dropdown List control to an array by using ArrayList class.
- Thus, we can bind control to properties, methods or collection. As in the above example, ArrayList is a collection so it is valid.

d) Enlist different scripting runtime library functions.

(Scripting.FileSystemObject Object-2 Marks, Scripting.Dictionary Object – 1 Mark, Scripting.TextStream Object- 1 Mark)

Scripting.FileSystemObject Object

- The FileSystemObject object provides us with access to the underlying file system on the server.
- This object can manipulate files, folders and directory paths. It is also possible to retrieve file system information with this object.
- This object contains three objects :
 1. **The Drive object:** Provides access to all the drives available on the machine.
 2. **The Folder object:** Provides access to the folder on a drive.
 3. **The File object:** Provides access to the file on a drive.

1. Scripting.Dictionary Object

- The Dictionary object is used to store information in name/value pairs (referred to as key and item).
- The Dictionary object is similar to Arrays. However, the Dictionary object is a more desirable solution to manipulate related data.
- **Comparing Dictionaries and Arrays :**
 - Key are used to identify the items in a Dictionary object.
 - You do not have to call ReDim to change the size of the Dictionary object.
 - When deleting an item from a Dictionary, the remaining items will automatically shift up.
 - Dictionaries cannot be multidimensional ; Arrays can be multidimensional.
 - Dictionaries have more built-in functions than Arrays.



- Dictionaries work better than arrays on accessing random elements frequently.

2. Scripting.TextStream Object

- The TextStream object is used to access the contents of text files.
- To create an instance of the TextStream object you can use the CreateTextFile or openTextFile methods of the FileSystemObject object or you can use the OpenAsTextStream method of the File object.

The following table describes the objects contained in the Scripting Runtime library.

Object	Collection	Description
Dictionary		Top-level object. Similar to the VBA Collection object.
Drive	Drives	Refers to a drive or collection of drives on the system.
File	Files	Refers to a file or collection of files in the file system.
FileSystemObject		Top-level object. Use this object to access drives, folders, and files in the file system.
Folder	Folders	Refers to a folder or collection of folders in the file system.
TextStream		Refers to a stream of text that is read from, written to, or appended to a text file.



e) What is web application and why it is used?

(Explanation of Web application – 2 Marks, Use – 2 Marks)

- A Web application (Web app) is an application program that is stored on a remote server and delivered over the Internet through a browser interface.
- Web application commonly uses a combination of server side script and client side script to develop the application. The client side is to develop the presentation of the information while server side script deals with the storing and retrieving information.
- A web application is a software which is accessed from a web browser via the HTTP protocol.
- A web application is a self-contained sub-tree of the web site. It uses different functionality provided by ASP.Net code to provide a response to a client that makes an HTTP request.
- Web application commonly uses a combination of server-script and client-side script to develop an application.

Uses of web application :

- Web Applications run through a web browser like IE, Mozilla which are familiar to the users. Web applications interact with users requesting and responding to users. Most common example is online shopping.
- Web application typically uses a database at the backend to permanently store information. They provide many business benefits as compared to office based solutions.

Some of the Advantages of web pages are :

- No special configuration is required.
- Less efforts are required to tell about the products and sell them.
- Data is centrally stored and is secure and easy to backup.
- Updates are quick and easy.
- Users are familiar with browser which is used in web applications.
- People located at different places can have direct access for current information.

f) What is data list control, data grid control, from view control, and detail view control

(explanation of each control – 1 Mark)

1. DataList control

The dataList control displays data items in a repeating list and optionally supports selecting and editing the items.



the content and layout of list items in DataList is defined using templates. The following table describes those templates. The following table describes those templates.

Example : Creating DataList in a tabular form.

Default.aspx

```
<% @ Page Language="VB" CodeFile="Default.aspx.vb" Inherits="_Default"%>
<% @ Import namespace="System.Data.SqlClient"%>
<% @ Import Namespace="System.Data"%>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>Untitled page</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<% Dim con As SqlConnection
Dim dtr As SqlDataReader
Dim cmd As SqlCommand
con=New SqlConnection("data Source=Vaishali\sqlExpress;Initial Catalog=tempdb;integrated
Security=True")
cmd=New SqlCommand("select * From Employee", con)
con.Open()
dtr=cmd.ExecuteReader()
datalist1.DataSource=dtr
datalist1.DataBind()
dtr.close()
con.close()
%>
</div>
<asp:DataList ID="dataList1" runat="server">
<header template>
<tr>
```



```
<th align="right">Emp_Name</th>
<th align="right">Emp_Id</th>
</tr>
</header template>
<Item template>
<tr>
<td><%#Container.dataItem("Name")%></td>
<td><%#Container.DataItem.DataItem("Rollno")%></td>
</tr>
</Item template>
</asp:DataList>
</form>
</body>
</html>
```

2.GridView Control

- The GridView control is used to display the values of a data source in a table. Each Column represents a field, while each row represents a records. It is a successor of datagrid control.
- The GridView control supports the Following features:
 7. Binding to data source control, such as sqlDataSource.
 8. Built-in sorting, updating, deleting, paging capabilities.
 9. Built –in row selecting capabilities.
 10. Multiple key fields.
 11. Multiple data field for the hyperlink columns.
 12. Customizable appearance through themes and styles.

Example:

```
<%@Page Language="VB"
AutoEventWireup="false"codeFile="Default.aspx.vb"Inherits="_Default"%>
<!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>Untitled page</title>/head>
<body>
<form id="Form1" runat="server">
<asp: GridView ID="GridView1" DataSourceID="sqlDataSource1" runat="server"/>
<asp: aqlDatasourceID="sqlDataSource1" Runat="Server"ConnectionString="Data
Source=.;Initial Catalog=northwind;UID=sa;Password=sa"Select Command="SELECT *
FROM Employee" ProviderName="System.Data.Aqlclient"/>
</form></body></html>
```

3. FormView Control

- Fromview is a data bound user interface control that render a single record at a time from its associated data source, optionally providing paging buttons to navigate between records.
- Binding to data source control,such as SqlDataSource and ObjectDataSource.
- The Features of FormView are as follows:
 1. Built-in inserting,updating and deleting capabilities.
 2. Built-in paging capabilities.
 3. Customizable appearance through user-defined templates, themes and styles.

Example:

Default.aspx

```
<%@Page Language="VB" CodeFile="Default.aspx.vb" Inherits="_Default"%>
<html xmlns=http://www.w3.org/1999/xhtml>
<head runat="server"> <title>Untitled page</title>/head>
<body>
<form id="Form1" runat="server">
<asp:Lable ID ="lable1" runat="server" text="FormViewConrol"/xbr/><br/>
<asp:FormViewID="FormView1" runnat="server" Allowpaging="True" BorderColor="Black"
BorderWidth="1px" dataKeyNames="Student_Rollno" DtaSourceID="sqlDataSource1">
<insertItemTemplate.
Empolyee_ID:>asp:TextBoxID="TextBoxID" runat="server" Text='<%=#Eval(Employee_I
D)%' /><br/>
```



```
Student _name:<asp:TextBoxID=""TextBox2"runat=""server
text='<%#Bind(""Student _name")%>' /><br/>
<insertitemtemplate>
<itemTemplate>
    Student_Rollno:<asp:Lable ID=""Lable3 runat=""server" text='<%# Eval
    ("Student_Rollno")%>' /><br/>
    Student _name:<asp:Lable ID=""Lable4"runat =server" text='<%# Bind
    ("Student_Rollno")%>' /><br/>
</Item Template>
</asp:FromView>
<asp:sqlDataSource ID=""sqldataSource1" runat=""Server" ConnectionString=""dtaSource=.;intial
catalog =northwind;user ID=sa;password=sa" SelectCommand=""SELECT *FROM Student"
providername=""System.Data.Sqlulclient"/>
</from></body></html>
```

4. DetailView control

It is a data-bound user interface control that renders a single record at a time from its associated data source, optionally providing paging buttons to navigate between records.

It is similar to the form view of an Access database and is that DetailsView has a built-in tabular rendering, whereas formView requires a user-defined template for its rendering.

Example:

```
<% @ Page
Language="VB"AutoEventWireup="false"CodeFile="Default.aspx.vb"Inherits=""_Default"%>
<html xmlns-"http://www.w3.org/1999/xhtml">
<head runat="server">
<title>Untitled Page</title>
</head>
<body.
<form id=Form1 "runat="server">
<asp:DetailsView ID="DetailsView1"runat="server" AllowPaging="true"AutoGenerate
Rows="false" dataSource ID=""sqlDataSource1"Header Text="Emp_Id"
SortExpression="Emp_Id"/>
```



</Field>

</asp:DetailView>

</asp:DetailView>

<asp:sqlDtasource ID="SqlDataSource1"runat="server" ConnectionString="data source=lab-311

\SQLEXPRESS;Initial catalog=tempdb;Integrated Security=true"

ProviderName="System.Data.SqlClient"SelectCommand="SELECT * FROM [Employee]">

</asp:sqlDataSource>

</body>

</html>