AS-4232

B. Tech. (Seventh Semester) Examination, 2013

(Information Technology Engg. Branch)

INTRODUCTION to .NET TECHNOLOGY

(IT4102)

Time Allowed : Three hours

Maximum Marks : 60

***Note** : The paper consists of two sections. Section-A and Section-B. Section-A is compulsory. In Section-B attempt any one question from each unit.*

Section-A

***Note :** All questions are compulsory. Each question carries equal marks.*

1. Objective type questions : 10×1=10

(i) MSIL is ..... .

(a) Microsoft Intermediate Language

(b) Microsoft Intermeta Language

(c) Microsoft Intermemory Language

(d) None of these

(ii) LINO in .NET is ..... .

(a) Language Integrated Query

(b) Language Interrelated Query

(c) Language Inform Query

(d) None of these

(iii) Generic Collection Classes Derives from ..... .

(a) System.Collections

(b) System.Data

(c) System.Array

(d) None of these

(iv) UDDI stands for ..... .

(a) Universal Description, Discovery and Integration

(b) Universal Drive, Discovery and Integration

(c) Universal Driver, Discovery and Integration

(d) None of these

(v) Which is not a main component of the Visual Studio IDE?

(a) Solution Explorer

(b) Start Menu

(c) Designer Window

(d) Properties Window

(vi) .NET doesn't have Automatic Memory Management Feature.

(True/False)

(vii) Default Document is the HTTP features of IIS.

(True/False)

(viii) We can't change the .NET Framework Version in IIS.

(True/False)

(ix) Impersonation feature in not present in IIS.

(True/False)

(x) Details View Control is not available with ASP.NET.

(True/False)

2. Short answer type : **5×2=10**

(i) What is FCL.NET? (Framework class lib)

(ii) What is IIS Express?

(iii) What is C#?

(iv) What is Config file in .NET?

(v) What is UDDI?

Section-B **5×8=40**

Note : Attempt any one question from each unit. Each question carries equal 8 marks.

Unit-I

3. Discuss the various components of a .NET IDE.

Or

4. How .NET framework helps towards Web and Desktop Programming?

Unit-II

5. What are Collections in VB.NET? Discuss its various types in brief.

Or

6. Discuss various events in VB.NET with one complete example.

Unit-III

7. Explain how connection and command objects helps ADO.NET. Discuss with proper example.

Or

8. Write a program in VB.NET to access the Data from MS SQL Server using Dataset.

Unit-IV

9. Explain various available HTML controls in ASP.NET.

Or

10. Write a complete web application code in ASP.NET for login module.

Unit-V

11. Write in brief on AJAX and XML Serialization.

Or

12. What is OLAP? Explain it with one complete suitable working code.

SOLUTION

B.TECH (7TH SEM) EXAM 2013

(INFORMATION TECHNOLOGY ENGG. BRANCH)

SUBJECT : INTRODUCTION TO .NET TECHNOLOGY

(AS-4232)

Q(1.) Objective type questions Answer :

i) Ans : A

ii) Ans: A

iii) Ans: A

iv) Ans: A

v) Ans: B

vi) Ans: FALSE

vii) Ans: TRUE

viii) Ans: FALSE

ix) Ans: FALSE

x) Ans: FALSE

Q(2) Short answer type :

i) Ans : The **Framework Class Library** (**FCL**) is a standard library and one of two core components of Microsoft .NET Framework. The FCL is a collection of reusable classes, interfaces and value types. The Base Class Library is a part of FCL and provides the most fundamental functionality, which includes classes in namespaces System, System.CodeDom, System.Collections, System.Diagnostics, System.Globalization, System.IO, System.Resources and System.Text

ii) Ans: IIS Express is a lightweight, self-contained version of IIS optimized for developers. IIS Express makes it easy to use the most current version of IIS to develop and test websites. It has all the core capabilities of IIS 7 and above as well as additional features designed to ease website development including:

- It doesn't run as a service or require administrator user rights to perform most tasks.
- IIS Express works well with ASP.NET and PHP applications.
- Multiple users of IIS Express can work independently on the same computer.

iii) Ans : C# is an elegant and type-safe object-oriented language that enables developers to build a variety of secure and robust applications that run on the .NET Framework. You can use C# to create Windows client applications, XML Web services, distributed components, client-server applications, database applications, and much, much more. Visual C# provides an advanced code editor, convenient user interface designers, integrated debugger, and many other tools to make it easier to develop applications based on the C# language and the .NET Framework.

iv) Ans : The .NET framework provides a rich set of classes, and techniques, to simplify application configuration. Essentially, all of these classes make it easy to read and write that configuration information from an XML configuration file.

The configuration file includes a number of standard sections, some custom sections for common .NET features, and also allows the developer to create their own custom configuration sections.

The standard sections have evolved over time. Initially, standard configuration was done mostly through theappSettings section, which contains name / value pairs for each setting. Over time, transparent, type-safe support was provided via a generated C# Settings class and the corresponding applicationSettings and userSettingsconfiguration sections. The machine configuration file lives with the, not so easily found, .NET framework files. The location of the configuration file is dependent on the version of .NET and type of platform (e.g. 64 bit) used by your application.

A typical example, might be: C:\Windows\Microsoft.NET\Framework\v4.0.30319\CONFIG\machine.confi

(V) Ans : UDDI is an XML-based standard for describing, publishing, and finding Web services.

- UDDI stands for Universal Description, Discovery and Integration.
- UDDI is a specification for a distributed registry of Web services.
- UDDI is platform independent, open framework.
- UDDI can communicate via SOAP, CORBA, Java RMI Protocol.
- UDDI uses WSDL to describe interfaces to web services.
- UDDI is seen with SOAP and WSDL as one of the three foundation standards of web services.
- UDDI is an open industry initiative enabling businesses to discover each other and define how they interact over the Internet.

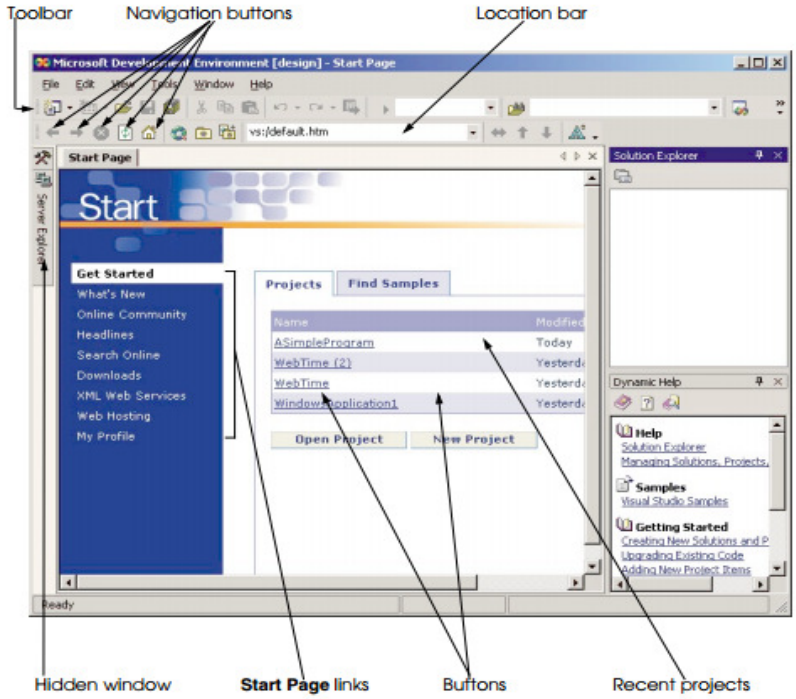
UDDI has two parts:

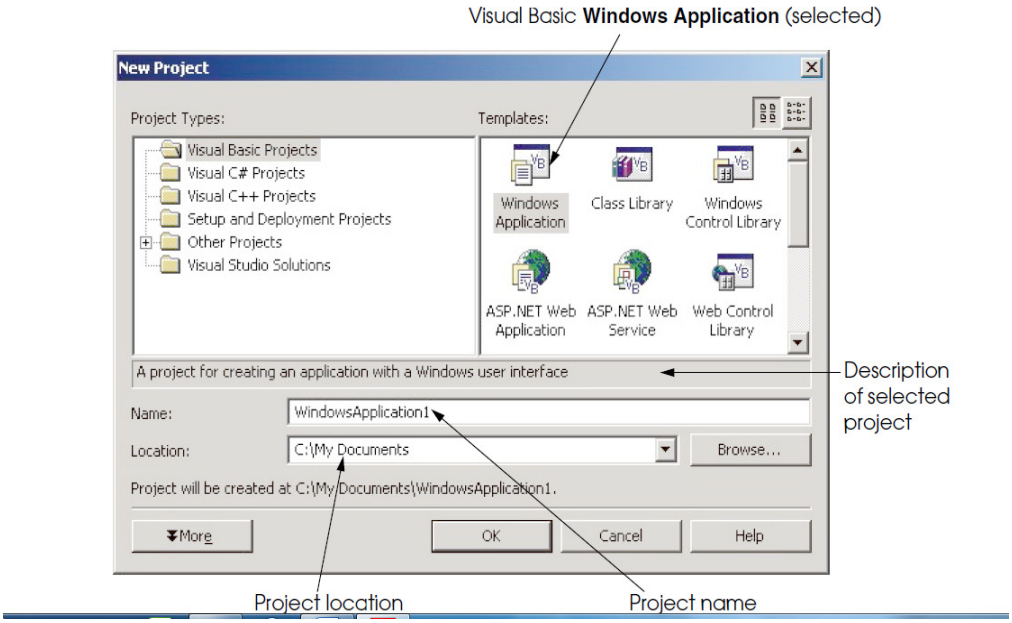
- A registry of all a web service's metadata including a pointer to the WSDL description of a service
- A set of WSDL port type definitions for manipulating and searching that registry

Section – B

Unit – I

Q(3) Ans :


Toolbar
Navigation buttons
Location bar
Start Page
Get Started
What's New
Online Community
Headlines
Search Online
Downloads
XML Web Services
Web Hosting
My Profile
Projects
Find Samples
Open Project
New Project
Hidden window
Start Page links
Buttons
Recent projects


Visual Basic Windows Application (selected)
New Project
Project Types:
Templates:
Visual Basic Projects
Visual C# Projects
Visual C++ Projects
Setup and Deployment Projects
Other Projects
Visual Studio Solutions
Windows Application
Class Library
Windows Control Library
ASP.NET Web Application
ASP.NET Web Service
Web Control Library
A project for creating an application with a Windows user interface
Description of selected project
Name:
WindowsApplication1
Location:
C:\My Documents
Browse...
Project will be created at C:\My Documents\WindowsApplication1.
More
OK
Cancel
Help
Project location
Project name

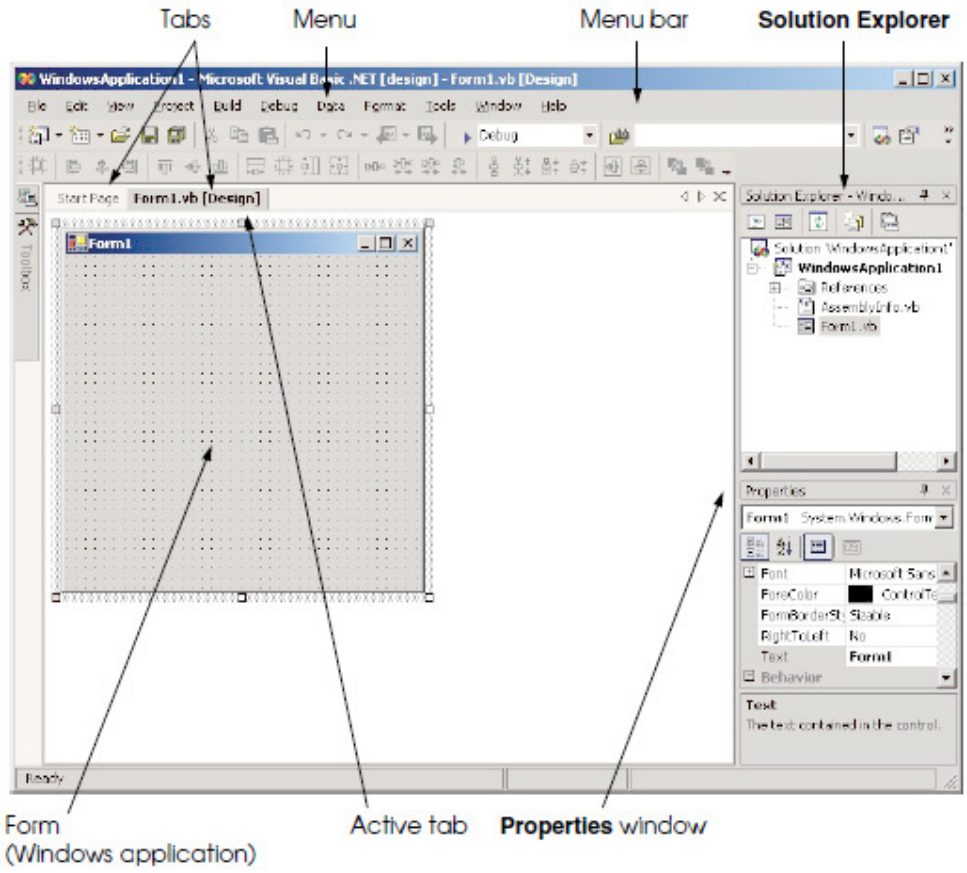


Fig. 2.3 **Design** view of Visual Studio .NET IDE.

Fig. 2.4 Visual Studio .NET IDE menu bar.

Menu	Description
File	Contains commands for opening projects, closing projects, printing project data, etc.
Edit	Contains commands such as cut, paste, find, undo, etc.
View	Contains commands for displaying IDE windows and toolbars.
Project	Contains commands for managing a project and its files.
Build	Contains commands for compiling a program.
Debug	Contains commands for *debugging* (i.e., identifying and correcting problems in a program) and running a program.
Data	Contains commands for interacting with *databases* (i.e., files that store data, which we discuss in Chapter 19, Databases, SQL and ADO .NET).
Format	Contains commands for arranging and changing the appearance of a form's controls.
Tools	Contains commands for accessing additional IDE tools and options that enable customization of the IDE.
Windows	Contains commands for arranging and displaying windows.
Help	Contains commands for accessing the IDE's help features.

2.4 Visual Studio .NET IDE Windows

The IDE provides windows for accessing project files and customizing controls. In this section, we introduce several windows that are essential in the development of Visual Basic applications. These windows can be accessed via the toolbar icons (Fig. 2.8) or by selecting the name of the desired window in the **View** menu.

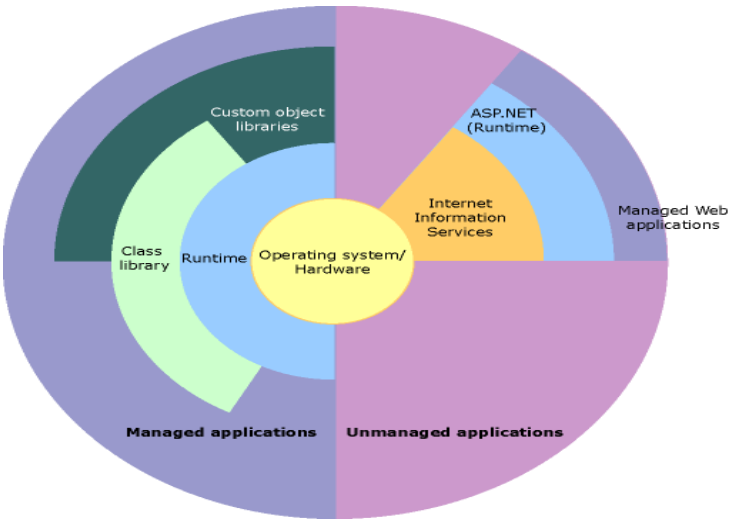
Q(4.) Ans :

.NET Framework (pronounced *dot net*) is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a largelibrary and provides language interoperability (each language can use code written in other languages) across several programming languages. Programs written for .NET Framework execute in a software environment (as contrasted to hardware environment), known as the Common Language Runtime (CLR), an application virtual machine that provides services such as security, memory management, and exception handling. The class library and the CLR together constitute .NET Framework.

.NET Framework's Base Class Library provides user interface, data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications. Programmers produce software by combining their own source code with .NET Framework and other libraries. .NET Framework is intended to be used by most new applications created for the Windows platform. Microsoft also produces anintegrated development environment largely for .NET software called Visual Studio.

The .NET Framework is a technology that supports building and running the next generation of applications and XML Web services. The .NET Framework is designed to fulfill the following objectives:

- To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
- To provide a code-execution environment that minimizes software deployment and versioning conflicts.
- To provide a code-execution environment that promotes safe execution of code, including code created by an unknown or semi-trusted third party.
- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
- To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
- To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.



2. .Net Framework 2.0 Features

- ADO.NET 2.0
- SQL Server data provider (SqlClient)
- XML
- .NET Remoting
- ASP.NET 2.0

3. .Net Framework 3.0/3.5 Features

- Windows Presentation Foundation (WPF)
- Windows Communication Foundation (WCF)
- Windows Workflow Foundation (WWF)
- Windows Card Space (WCS)
- Core New Features and Improvements:

    - Auto Implemented
    - Implicit Typed local variable
    - Implicitly Typed Arrays
    - Anonymous Types

- Extension Methods (3.5 new feature)
- Object and Collection Initializers
- Lambda Expressions

4. .Net Framework 4.0 Features

- Application Compatibility and Deployment
- Core New Features and Improvements
  - BigInteger and Complex Numbers
  - Tuples
  - Covariance and Contravariance
  - Dynamic Language Runtime
- Managed Extensibility Framework
- Parallel Computing
- Networking
- Web
- Client
- Data
- Windows Communication Foundation
- Windows Workflow Foundation

5. .Net Framework 4.5 Features

- .NET for Windows Store Apps
- Portable Class Libraries
- Core New Features and Improvements
- Tools
- Parallel Computing
- Web
  Windows Presentation Foundation (WPF)

- Windows Communication Foundation (WCF)
- Windows Workflow Foundation (WF)

Unit – II

Q(5.) Ans :

In general terms, a *collection* is an object used for grouping and managing related objects. For example, every Form has a collection of controls. (You can access this collection through the form's Controls property.) This collection is an object that represents all the controls on that form. It allows you to retrieve a control in the collection by its index, and to loop through the elements of the collection using a For Each...Next Statement (Visual Basic).
However, there are several kinds of collections, and they differ from each other in several ways.

Collection classes are specialized classes for data storage and retrieval. These classes provide support for stacks, queues, lists, and hash tables. Most collection classes implement the same interfaces.

Collection classes serve various purposes, such as allocating memory dynamically to elements and accessing a list of items on the basis of an index, etc. These classes create collections of objects of the Object class, which is the base class for all data types in VB.Net.

Various Collection Classes and Their Usage

The following are the various commonly used classes of the **System.Collection** namespace. Click the following links to check their details.

Class	Description and Useage
ArrayList	It represents ordered collection of an object that can be **indexed** individually. It is basically an alternative to an array. However, unlike array, you can add and remove items from a list at a specified position using an **index** and the array resizes itself automatically. It also allows dynamic memory allocation, add, search and sort items in the list.
Hashtable	It uses a **key** to access the elements in the collection. A hash table is used when you need to access elements by using key, and you can identify a useful key value. Each item in the hash table has a **key/value** pair. The key is used to access the items in the collection.

SortedList	It uses a **key** as well as an **index** to access the items in a list. A sorted list is a combination of an array and a hash table. It contains a list of items that can be accessed using a key or an index. If you access items using an index, it is an ArrayList, and if you access items using a key, it is a Hashtable. The collection of items is always sorted by the key value.
Stack	It represents a **last-in, first out** collection of object. It is used when you need a last-in, first-out access of items. When you add an item in the list, it is called **pushing** the item, and when you remove it, it is called**popping** the item.
Queue	It represents a **first-in, first out** collection of object. It is used when you need a first-in, first-out access of items. When you add an item in the list, it is called **enqueue**, and when you remove an item, it is called**deque**.
BitArray	It represents an array of the **binary representation** using the values 1 and 0. It is used when you need to store the bits but do not know the number of bits in advance. You can access items from the BitArray collection by using an **integer index**, which starts from zero.

A **collection** is an object. It contains references to other objects. In this way VB.NET combines many class instances and values together. With collections we construct object models that mirror real-world problems.

Q(6.) Ans :

An event is a signal that informs an application that something important has occurred. For example, when a user clicks a control on a form, the form can raise a**Click** event and call a procedure that handles the event. Events also allow separate tasks to communicate. Say, for example, that your application performs a sort task separately from the main application. If a user cancels the sort, your application can send a cancel event instructing the sort process to stop.

1.) Click

2.) Key

3.) Mouse

4.) Custom

5.) Form

Events are basically a user action like key press, clicks, mouse movements, etc., or some occurrence like system generated notifications. Applications need to respond to events when they occur.

Clicking on a button, or entering some text in a text box, or clicking on a menu item, all are examples of events. An event is an action that calls a function or may cause another event.

Event handlers are functions that tell how to respond to an event.

VB.Net is an event-driven language. There are mainly two types of events:

- Mouse events
- Keyboard events

Handling Mouse Events

Mouse events occur with mouse movements in forms and controls. Following are the various mouse events related with a Control class:

- **MouseDown** - it occurs when a mouse button is pressed
- **MouseEnter** - it occurs when the mouse pointer enters the control
- **MouseHover** - it occurs when the mouse pointer hovers over the control
- **MouseLeave** - it occurs when the mouse pointer leaves the control
- **MouseMove** - it occurs when the mouse pointer moves over the control
- **MouseUp** - it occurs when the mouse pointer is over the control and the mouse button is released
- **MouseWheel** - it occurs when the mouse wheel moves and the control has focus

The event handlers of the mouse events get an argument of type **MouseEventArgs**. The MouseEventArgs object is used for handling mouse events. It has the following properties:

- **Buttons** - indicates the mouse button pressed
- **Clicks** - indicates the number of clicks
- **Delta** - indicates the number of detents the mouse wheel rotated
- **X** - indicates the x-coordinate of mouse click
- **Y** - indicates the y-coordinate of mouse click

Handling Keyboard Events

Following are the various keyboard events related with a Control class:

- **KeyDown** - occurs when a key is pressed down and the control has focus
- **KeyPress** - occurs when a key is pressed and the control has focus
- **KeyUp** - occurs when a key is released while the control has focus

The event handlers of the KeyDown and KeyUp events get an argument of type **KeyEventArgs**. This object has the following properties:

- **Alt** - it indicates whether the ALT key is pressed/p>
- **Control** - it indicates whether the CTRL key is pressed
- **Handled** - it indicates whether the event is handled
- **KeyCode** - stores the keyboard code for the event
- **KeyData** - stores the keyboard data for the event
- **KeyValue** - stores the keyboard value for the event
- **Modifiers** - it indicates which modifier keys (Ctrl, Shift, and/or Alt) are pressed
- **Shift** - it indicates if the Shift key is pressed

The event handlers of the KeyDown and KeyUp events get an argument of type **KeyEventArgs**. This object has the following properties:

- **Handled** - indicates if the KeyPress event is handled
- **KeyChar** - stores the character corresponding to the key pressed

Unit- III

Q(7.) Ans :

DO.NET provides consistent access to data sources such as SQL Server and XML, and to data sources exposed through OLE DB and ODBC. Data-sharing consumer applications can use ADO.NET to connect to these data sources and retrieve, handle, and update the data that they contain.

ADO.NET separates data access from data manipulation into discrete components that can be used separately or in tandem. ADO.NET includes .NET Framework data providers for connecting to a database, executing commands, and retrieving results. Those results are either processed directly, placed in an ADO.NET DataSet object in order to be exposed to the user in an ad hoc manner, combined with data from multiple sources, or passed between tiers. The **DataSet** object can also be used independently of a .NET Framework data provider to manage data local to the application or sourced from XML.

The ADO.NET classes are found in System.Data.dll, and are integrated with the XML classes found in System.Xml.dll. For sample code that connects to a database, retrieves data from it, and then displays that data in a console window, see ADO.NET Code Examples.

ADO.NET provides functionality to developers who write managed code similar to the functionality provided to native component object model (COM) developers by ActiveX Data Objects (ADO). We recommend that you use ADO.NET, not ADO, for accessing data in your .NET applications.

Connection & Command Objects

The Connection Object is a part of ADO.NET Data Provider and it is a unique session with the Data Source. In .Net Framework the Connection Object is Handling the part of physical communication between the application and the Data Source. Depends on the parameter specified in the Connection String , ADO.NET 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. Once the Database activity is over , Connection should be closed and release the resources . In ADO.NET the type of the Connection is depend on what Database system you are working with. The following are the commonly using the connections in the ADO.NET

SqlConnection

OleDbConnection

OdbcConnection

Command : Command is used to execute almost any SQL command from within the .net application. The SQL command like insert, update, delete, select, create, alter, drop can be executed with command object and you can also call stored procedures with the command object. Command object has the following important properties.

- Connection : used to specify the connection to be used by the command object.
- CommandType : Used to specify the type of SQL command you want to execute. To assign a value to this property, use the enumeration CommandType that has the members Text, StoredProcedure and TableDirect. Text is the default and is set when you want to execute ant SQL command with command object. StoredProcedure is set when you want to call a stored procedure or function and**TableDirect** is set when you want to retrieve data from the table directly by specifying the table name without writing a select statement.
- CommandText : Used to specify the SQL statement you want to execute.
- Transaction : Used to associate a transaction object to the command object so that the changes made to the database with command object can be committed or rollback.

ExecuteNonQuery() : Used to execute an SQL statement that doesn't return any value like insert, update and delete. Return type of this method is int and it returns the no. of rows effected by the given statement.

- ExecuteScalar() : Used to execute an SQL statement and return a single value. When the select statement executed by executescalar() method returns a row and multiple rows, then the method will return the value of first column of first row returned by the query. Return type of this method is object.
- **ExecuteReader() :** Used to execute a select a statement and return the rows returned by the select statement as a DataReader. Return type of this method is **DataReader.**

Example :

isting 1. Using a SqlConnection

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

/// <summary>
/// Demonstrates how to work with SqlConnection objects
/// </summary>
class SqlConnectionDemo
{
    static void Main()
    {
        // 1. Instantiate the connection
        SqlConnection conn = new SqlConnection(
            "Data Source=(local);Initial Catalog=Northwind;Integrated Security=SSPI");

        SqlDataReader rdr = null;

        try
        {
            // 2. Open the connection
            conn.Open();

            // 3. Pass the connection to a command object
            SqlCommand cmd = new SqlCommand("select * from Customers", conn);

            //
            // 4. Use the connection
            //

            // get query results
```

```
            rdr = cmd.ExecuteReader();

            // print the CustomerID of each record
            while (rdr.Read())
            {
                Console.WriteLine(rdr[0]);
            }
        }
        finally
        {
            // close the reader
            if (rdr != null)
            {
                rdr.Close();
            }

            // 5. Close the connection
            if (conn != null)
            {
                conn.Close();
            }
        }
    }
}
```

Q(8.) Ans :

```
Imports System
Imports System.Data
Imports System.Data.SqlClient

Module Module1

    Sub Main()
        Dim sConnectionString As String
        sConnectionString = "Password=<strong password>;User ID=<username>;" & _
                            "Initial Catalog=pubs;" & _
                            "Data Source=(local)"

        Dim objConn As New SqlConnection(sConnectionString)
        objConn.Open()
```

```
        Dim daAuthors As _
            New SqlDataAdapter("Select * From Authors", objConn)

        Dim dsPubs As New DataSet("Pubs")
        daAuthors.FillSchema(dsPubs, SchemaType.Source, "Authors")
        daAuthors.Fill(dsPubs, "Authors")

        Dim tblAuthors As DataTable
        tblAuthors = dsPubs.Tables("Authors")

        Dim drCurrent As DataRow
        For Each drCurrent In tblAuthors.Rows
            Console.WriteLine("{0} {1}", _
                drCurrent("au_fname").ToString, _
                drCurrent("au_lname").ToString)
        Next
        Console.ReadLine()
    End Sub

End Module
```

Unit – IV

Q(9.)

Ans :

HTML Controls

HTML elements in ASP.NET files are, by default, treated as text. To make these elements programmable, add a runat="server" attribute to the HTML element. This attribute indicates that the element should be treated as a server control.

Note: All HTML server controls must be within a <form> tag with the runat="server" attribute!

Note: ASP.NET requires that all HTML elements must be properly closed and properly nested.

HTML Server Control	Description
HtmlAnchor	Controls an <a> HTML element
HtmlButton	Controls a <button> HTML element

HtmlForm	Controls a <form> HTML element
HtmlGeneric	Controls other HTML element not specified by a specific HTML serve <body>, , , etc.
HtmlImage	Controls an <image> HTML element
HtmlInputButton	Controls <input type="button">, <input type="submit">, and <input HTML elements
HtmlInputCheckBox	Controls an <input type="checkbox"> HTML element
HtmlInputFile	Controls an <input type="file"> HTML element
HtmlInputHidden	Controls an <input type="hidden"> HTML element
HtmlInputImage	Controls an <input type="image"> HTML element
HtmlInputRadioButton	Controls an <input type="radio"> HTML element
HtmlInputText	Controls <input type="text"> and <input type="password"> HTML elemer
HtmlSelect	Controls a <select> HTML element
HtmlTable	Controls a <table> HTML element
HtmlTableCell	Controls <td>and <th> HTML elements
HtmlTableRow	Controls a <tr> HTML element
HtmlTextArea	Controls a <textarea> HTML element

Q(10 .) Ans:

C #

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
```

```
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Data.SqlClient;

public partial class sigin_in : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {

   }
   SqlConnection con = new SqlConnection("Data Source=KRISHN-PC;Initial Catalog=sign;Persist Security Info=True;User ID=sa;Password=**********");
   protected void Button1_Click(object sender, EventArgs e)
   {
      SqlDataAdapter obj = new SqlDataAdapter("select from login where userid = '"+TextBox1.Text+"'and password = '"+TextBox2.Text +"'",con);
      DataSet a = new DataSet();
      obj.Fill(a);
      //obj.Update(a);
      if (a.Tables[0].Rows.Count > 0)
      {
         Response.Redirect("home.aspx");
      }
      else
      {
         Label.Text = " wrong pswd";
      }
   }

}
```

Vb.NET

```
Imports System.Data
Imports System.Data.SqlClient
Imports System.Configuration

Partial Class _Default
Inherits System.Web.UI.Page
Protected Sub Page_Load(ByVal sender As Object, ByVal e As EventArgs)
End Sub
Protected Sub btnSubmit_Click(ByVal sender As Object, ByVal e As EventArgs)
Dim con As NewSqlConnection(ConfigurationManager.ConnectionStrings("dbconnection").ConnectionString)
```

```
con.Open()
Dim cmd As New SqlCommand("select * from UserInformation where UserName =@username and Password=@password", con)
cmd.Parameters.AddWithValue("@username", txtUserName.Text)
cmd.Parameters.AddWithValue("@password", txtPWD.Text)
Dim da As New SqlDataAdapter(cmd)
Dim dt As New DataTable()
da.Fill(dt)
If dt.Rows.Count > 0 Then
Response.Redirect("Details.aspx")
Else
ClientScript.RegisterStartupScript(Page.[GetType](), "validation", "<script language='javascript'>alert('Invalid Username and Password')</script>")
End If
End Sub
End Class
```

Unit- V

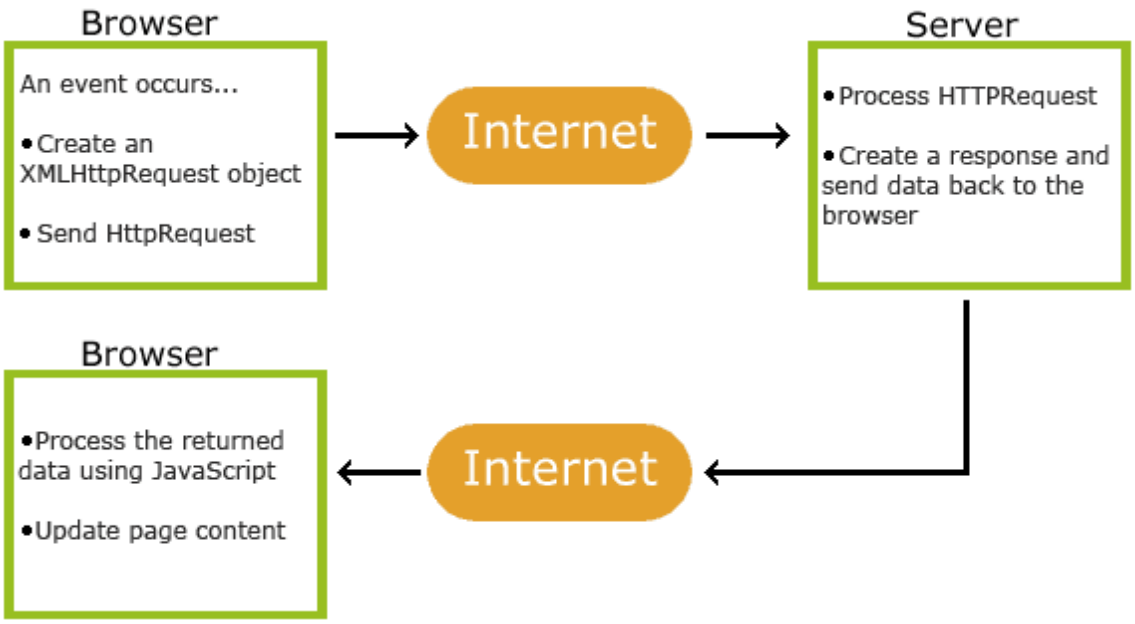
Q(11.) Ans :

AJAX

AJAX = Asynchronous JavaScript and XML.

AJAX is not a new programming language, but a new way to use existing standards.

AJAX is the art of exchanging data with a server, and updating parts of a web page - without reloading the whole page.



AJAX is Based on Internet Standards

AJAX is based on internet standards, and uses a combination of:

- XMLHttpRequest object (to exchange data asynchronously with a server)
- JavaScript/DOM (to display/interact with the information)
- CSS (to style the data)
- XML (often used as the format for transferring data)
- AJAX Example Explained
- The AJAX application above contains one div section and one button.
- The div section will be used to display information returned from a server. The button calls a function named loadXMLDoc(), if it is clicked:

```
<!DOCTYPE html>
<html>
<body>

<div id="myDiv"><h2>Let AJAX change this text</h2></div>
<button type="button" onclick="loadXMLDoc()">Change Content</button>

</body>
</html>
```

- Next, add a <script> tag to the page's head section. The script section contains the loadXMLDoc() function:

```
<head>
<script>
function loadXMLDoc()
{
.... AJAX script goes here ...
}
</script>
</head>
```

XML Serialization

Serialization is the process of converting an object into a form that can be readily transported. For example, you can serialize an object and transport it over the Internet using HTTP between a client and a server. On the other end, deserialization reconstructs the object from the stream.
XML serialization serializes only the public fields and property values of an object into an XML stream. XML serialization does not include type information. For example, if you have a **Book** object that exists in the **Library** namespace, there is no guarantee that it is deserialized into an object of the same type.

The central class in XML serialization is the XmlSerializer class, and the most important methods in this class are the **Serialize** and **Deserialize** methods. The XmlSerializercreates C# files and compiles them into .dll files to perform this serialization. In .NET Framework 2.0, the XML Serializer Generator Tool (Sgen.exe) is designed to generate these serialization assemblies in advance to be deployed with your application and improve startup performance. The XML stream generated by the **XmlSerializer** is compliant with the World Wide Web Consortium (www.w3.org) XML Schema definition language (XSD) 1.0 recommendation. Furthermore, the data types generated are compliant with the document titled "XML Schema Part 2: Datatypes."
The data in your objects is described using programming language constructs like classes, fields, properties, primitive types, arrays, and even embedded XML in the form of**XmlElement** or **XmlAttribute** objects. You have the option of creating your own classes, annotated with attributes, or using the XML Schema Definition tool to generate the classes based on an existing XML Schema.
If you have an XML Schema, you can run the XML Schema Definition tool to produce a set of classes that are strongly typed to the schema and annotated with attributes. When an instance of such a class is serialized, the generated XML adheres to the XML Schema. Provided with such a class, you can program against an easily manipulated object model while being assured that the generated XML conforms to the XML schema. This is an alternative to using other classes in the .NET Framework, such as the **XmlReader**and **XmlWriter** classes, to parse and write an XML stream. For more information, see XML Documents and Data. These classes allow you to parse any XML stream. In contrast, use the **XmlSerializer** when the XML stream is expected to conform to a known XML Schema.

Attributes control the XML stream generated by the **XmlSerializer** class, allowing you to set the XML namespace, element name, attribute name, and so on, of the XML stream. For more information about these attributes and how they control XML serialization, see Controlling XML Serialization Using Attributes. For a table of those attributes that are used to control the generated XML, see Attributes That Control XML Serialization.

The **XmlSerializer** class can further serialize an object and generate an encoded SOAP XML stream. The generated XML adheres to section 5 of the World Wide Web Consortium document titled "Simple Object Access Protocol (SOAP) 1.1." For more information about this process, see How to: Serialize an Object as a SOAP-Encoded XML Stream. For a table of the attributes that control the generated XML, see Attributes That Control Encoded SOAP Serialization.

The **XmlSerializer** class generates the SOAP messages created by, and passed to, XML Web services. To control the SOAP messages, you can apply attributes to the classes, return values, parameters, and fields found in an XML Web service file (.asmx). You can use both the attributes listed in "Attributes That Control XML Serialization" and "Attributes That Control Encoded SOAP Serialization" because an XML Web service can use either the literal or encoded SOAP style. For more information about using attributes to control the XML generated by an XML Web service, see XML Serialization with XML Web Services. For more information about SOAP and XML Web services, seeCustomizing SOAP Messages.

Security Considerations for XmlSerializer Applications

When creating an application that uses the **XmlSerializer**, you should be aware of the following items and their implications:

- The **XmlSerializer** creates C# (.cs) files and compiles them into .dll files in the directory named by the TEMP environment variable; serialization occurs with those DLLs.

  Note:

  These serialization assemblies can be generated in advance and signed by using the SGen.exe tool. This other words, it is only for client use and for manual serialization.

- The code and the DLLs are vulnerable to a malicious process at the time of creation and compilation. When using a computer running Microsoft Windows NT 4.0 or later, it might be possible for two or more users to share the TEMP directory. Sharing a TEMP directory is dangerous if the two accounts have different security privileges and the higher-privilege account runs an application using the **XmlSerializer**. In this case, one user can breach the computer's security by replacing either the .cs or .dll file that is compiled. To eliminate this concern, always be sure that each account on the computer has its own profile. By default, the TEMP environment variable points to a different directory for each account.

- If a malicious user sends a continuous stream of XML data to a Web server (a denial of service attack), then the **XmlSerializer** continues to process the data until the computer runs low on resources.

This kind of attack is eliminated if you are using a computer running Internet Information Services (IIS), and your application is running within IIS. IIS features a gate that does not process streams longer than a set amount (the default is 4 KB). If you create an application that does not use IIS and deserializes with the **XmlSerializer**, you should implement a similar gate that prevents a denial of service attack.

- The **XmlSerializer** serializes data and runs any code using any type given to it.

  There are two ways in which a malicious object presents a threat. It could run malicious code or it could inject malicious code into the C# file created by the**XmlSerializer**. In the first case, if a malicious object tries to run a destructive procedure, code access security helps prevent any damage from being done. In the second case, there is a theoretical possibility that a malicious object may somehow inject code into the C# file created by the **XmlSerializer**. Although this issue has been examined thoroughly, and such an attack is considered unlikely, you should take the precaution of never serializing data with an unknown and untrusted type.

- Serialized sensitive data might be vulnerable.

  After the **XmlSerializer**has serialized data, it can be stored as an XML file or other data store. If your data store is available to other processes, or is visible on an intranet or the Internet, the data can be stolen and used maliciously. For example, if you create an application that serializes orders that include credit card numbers, the data is highly sensitive. To help prevent this, always protect the store for your data and take steps to keep it private.

Q(12.) Ans :

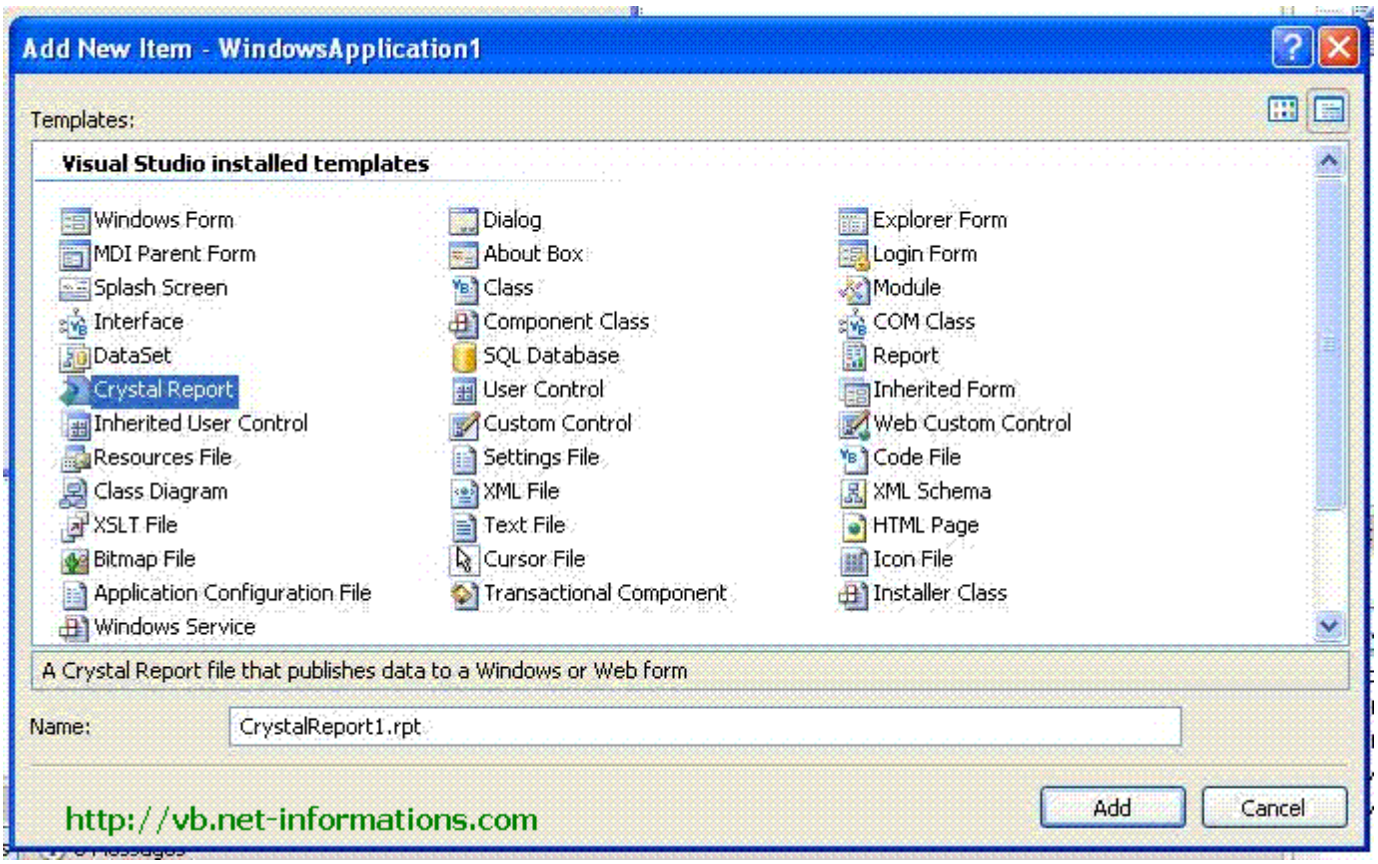
OLAP : In computing, **online analytical processing**, or **OLAP** /ˈoʊlæp/, is an approach to answering multi-dimensional analytical (MDA) queries swiftly.[1] OLAP is part of the broader category ofbusiness intelligence, which also encompasses relational database, report writing and data mining.[2] Typical applications of OLAP include business reporting for sales, marketing, management reporting, business process management (BPM),[3] budgeting and forecasting, financial reporting and similar areas, with new applications coming up, such as agriculture.[4] The term *OLAP* was created as a slight modification of the traditional database term OLTP (Online Transaction Processing).[5]

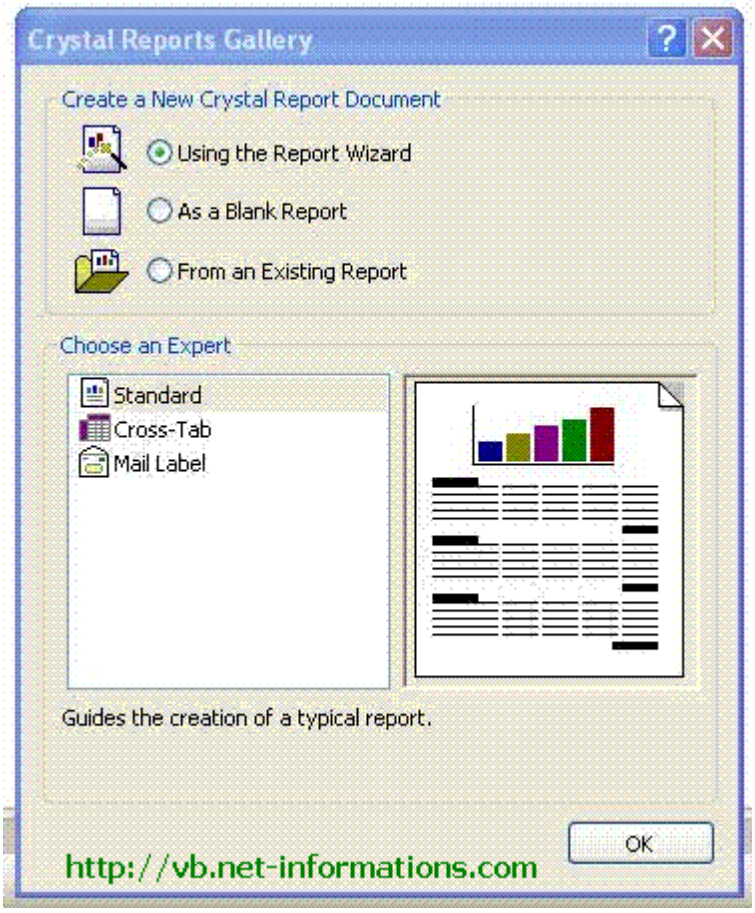
Example :

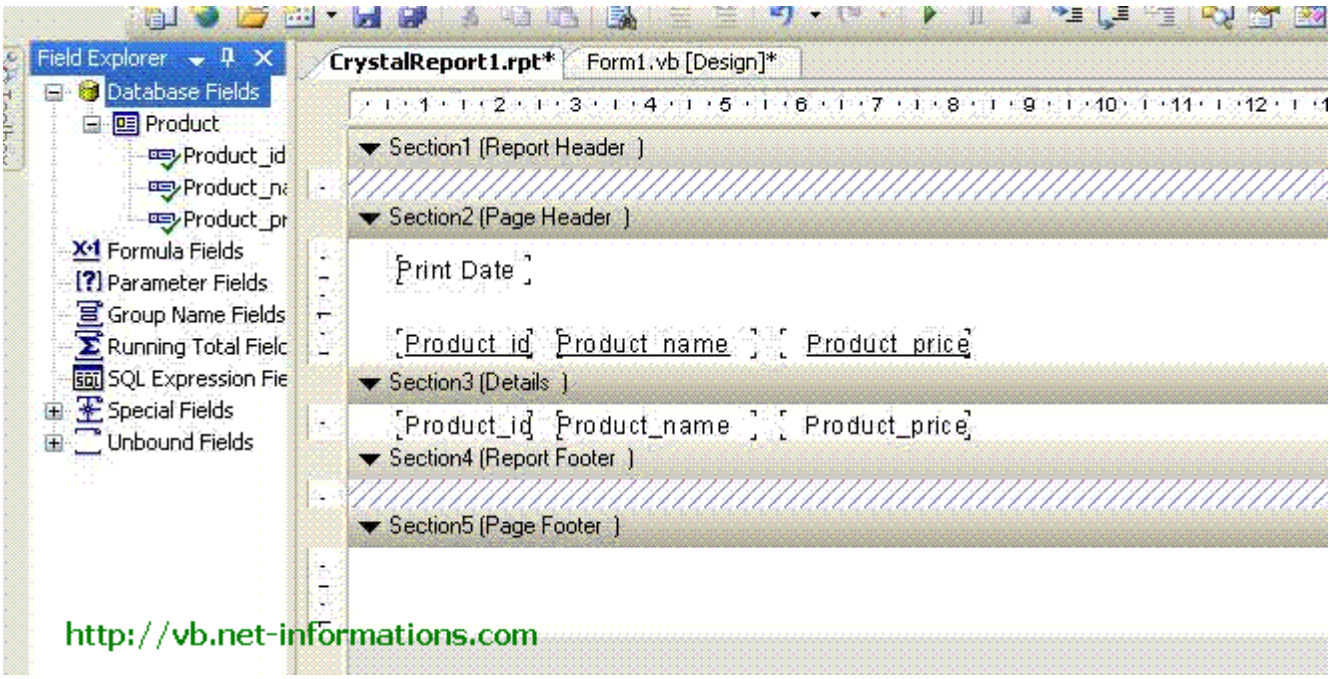
```
Imports CrystalDecisions.CrystalReports.Engine
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object,
```

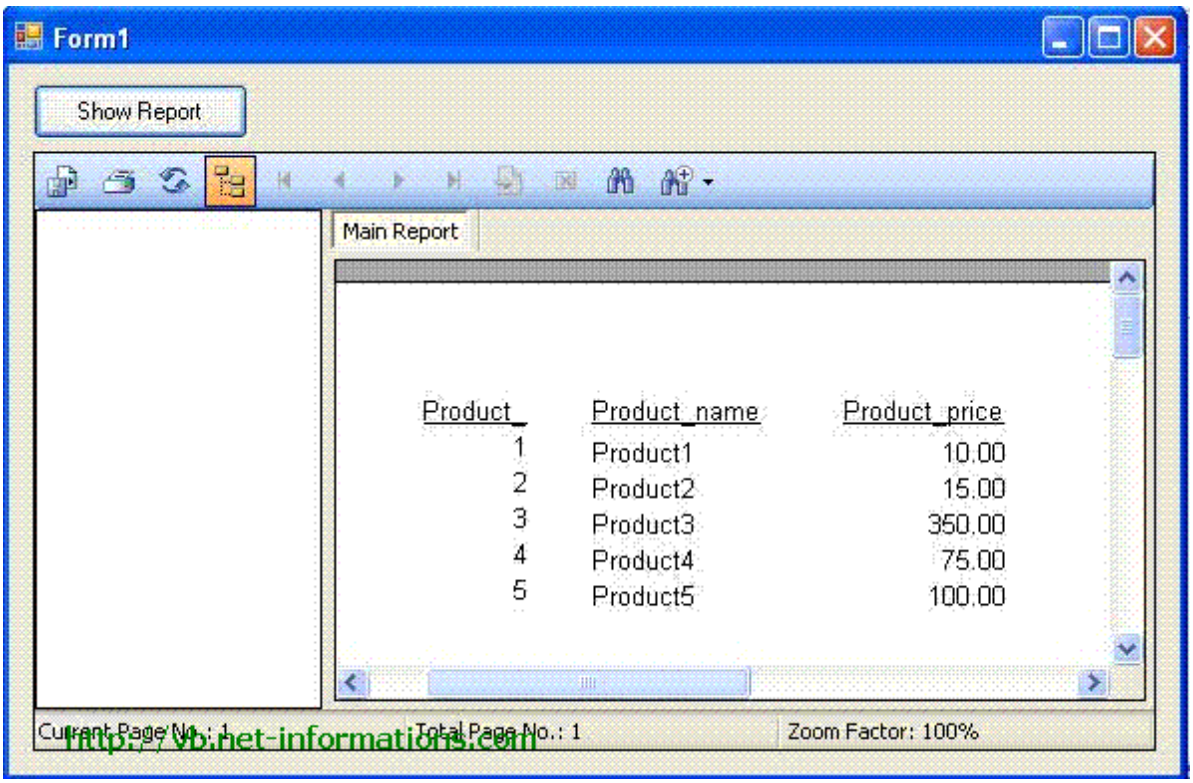
```
        ByVal e As System.EventArgs) Handles Button1.Click
    Dim cryRpt As New ReportDocument
    cryRpt.Load("PUT CRYSTAL REPORT PATH HERE\CrystalReport1.rpt")
    CrystalReportViewer1.ReportSource = cryRpt
    CrystalReportViewer1.Refresh()
  End Sub
End Class
```

cryRpt.Load("PUT CRYSTAL REPORT PATH HERE\CrystalReport1.rpt")




Crystal Reports Gallery
Create a New Crystal Report Document
Using the Report Wizard
As a Blank Report
From an Existing Report
Choose an Expert
Standard
Cross-Tab
Mail Label
Guides the creation of a typical report.
OK
http://vb.net-informations.com


Field Explorer
Database Fields
Product
Product_id
Product_na
Product_pr
Formula Fields
Parameter Fields
Group Name Fields
Running Total Fielc
SQL Expression Fie
Special Fields
Unbound Fields
CrystalReport1.rpt*
Form1.vb [Design]*
Section1 (Report Header)
Section2 (Page Header)
Print Date
Product_id
Product_name
Product_price
Section3 (Details)
Product_id
Product_name
Product_price
Section4 (Report Footer)
Section5 (Page Footer)
http://vb.net-informations.com


Form1
Show Report
Main Report
Product_
Product_name
Product_price
1
Product1
10.00
2
Product2
15.00
3
Product3
350.00
4
Product4
75.00
5
Product5
100.00
Current Page No.: 1
Total Page No.: 1
Zoom Factor: 100%
http://vb.net-informations.com