

Unit -4 Authentication, Authorization and Data Controls**4.1 Membership class:**

The ASP.NET Membership class is a part of the .NET Framework for application security that provides some features by default to create users, assign roles and assign authorization and authentication to user roles. It minimizes the effort of the developer in terms of coding because it provides the wizard to configure the membership settings. The ASP.NET Membership class provides the following features:

- i. Wizard-based configuration for user management capabilities.
- ii. Browser-based user management and access control configuration.
- iii. The Membership and Roles classes that you can use to access user management capabilities within your code.
- iv. A set of ASP.NET controls that provide users the ability to log on, log off, create new accounts and recover lost words and so on.
- v. It uses ASP.NET and a database for configuration settings.

➤ This membership class provides the following static methods:

- i. CreateUser
- ii. DeleteUser
- iii. FindUsersByEmail
- iv. FindUsersByName
- v. GetUser
- vi. GeneratePassword
- vii. GetAllUsers

- **CreateUser:** This method adds a user to the database. Use this method if you create a custom page to enable users or administrators to add new accounts.
- **DeleteUser:** This method removes a user from the data store. Use this method if you create custom user management tools.
- **FindUsersByEmail:** This method gets a collection of membership users with the specified email addresses.
- **FindUsersByName:** This method gets a collection of membership users with the specified user names.
- **GetUser:** This method returns a Membership User object representing the current logged-on user. Call this method any time you need to access the current user's account.
- **GeneratePassword:** Generates a random password of the specified length.

4.2 Role class:

In the *User-Based Authorization* tutorial we looked at using URL authorization to restrict certain users from a set of pages and explored declarative and programmatic techniques for adjusting an ASP.NET page's functionality based on the visiting user. Granting permission for page access or functionality on a user-by-user basis, however, can become a maintenance nightmare in scenarios where there are many user accounts or when users' privileges change often. Any time a user gains or loses authorization to perform a particular task, the administrator needs to update the appropriate URL authorization rules, declarative markup, and code. ASP.NET offers a Roles framework for defining roles and associating them with user accounts. With the Roles framework we can create and delete roles, add users to or remove users from a role, determine the set of users that belong to a particular role, and tell whether a user belongs to a particular role. Once the Roles framework has been configured, we can limit access to pages on a role-by-role basis through URL authorization rules and show or hide additional information or functionality on a page based on the currently logged on user's roles.

➤ Methods of Role Class:-

- i. AddUsersToRole
- ii. AddUserToRole
- iii. CreateRole
- iv. DeleteRole
- v. FindUserInRole
- vi. GetUsersInRole
- vii. IsUserInRole
- viii. RemoveUserFromRole(s)

- **AddUsersToRole :-** Adds the specified users to the specified role.
- **AddUserToRole:-** Adds the specified user to the specified role.
- **CreateRole:-** Adds a new role to the data source.
- **DeleteRole:-** Removes a role from the data source.
- **FindUserInRole:-** Gets a list of users in a specified role where the user name contains the specified user name to match.
- **GetUsersInRole:-** Gets a list of users in the specified role.
- **IsUserInRole:-** Gets a value indicating whether the currently logged-on user is in the specified role. The API is only intended to be called within the context of an ASP.NET request thread, and in that sanctioned use case it is thread-safe.

- **RemoveUserFromRole(s)** :- Removes the specified users from the specified role.

4.3 ProfileManager class:

The ASP.NET profile is used to store and retrieve user settings in a data source such as a database. The user profile is accessed using the Profile property of the current HttpContext. Profile information and property values are managed using a profile provider. The ProfileManager class is used to manage profile settings, search for user profiles, and delete user profiles that are no longer in use. The ProfileManager class provides static methods and properties that can be accessed by referencing the ProfileManager class in your application code. For an example, see the Example section in this topic and additional examples in the topics for members of the ProfileManager class.

➤ Methods of ProfileManager class:-

- i. DeleteInactiveProfiles
- ii. DeleteProfile
- iii. DeleteProfiles
- iv. FindProfileByUserName
- v. GetAllProfiles
- vi. GetNumberofProfile

- **DeleteInactiveProfiles**:- Deletes user profile data for which the last activity date and time occurred before the specified date and time.
- **DeleteProfile**:- Deletes the profile for the specified user name from the data source.
- **DeleteProfiles**:- Deletes profile properties and information from the data source for the supplied list of profiles.
- **FindProfileByUserName**:- Retrieves all profile information for profiles in which the user name matches the specified name.
- **GetAllProfiles**:- Retrieves user profile data for profiles in the data source.
- **GetNumberofProfile**:- Gets the number of profiles in the data source.

➤ Various Data Bound Controls Used in ASP.Net:

ASP.NET provides a wide variety of rich controls that can be bound to data.

Under the Data tab of the Visual Studio Toolbox, you can get several controls under the Data tab that could be used to display data from a data source, like a database or XML file.

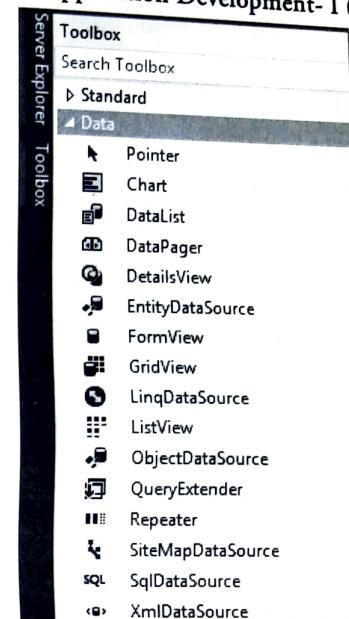


Figure 4.1: Create Access Rules

The standard ASP.NET data presentation controls are:

- i. Repeater
- ii. DataList
- iii. GridView
- iv. DetailsView
- v. FormView

GROUP 1	GROUP 2
1. DetailsView 2. FormView	1. Repeater 2. DataList 3. GridView 4. ListView

Description:
These two controls can only show individual record at the same time.

Description:
These controls show collection of records, and are commonly called data list controls.

Figure 4.2: ASP.NET data presentation controls

First of all, we go to the first group that includes the four controls Repeater, DataList, GridView and ListView. For demonstration purposes, we will use a table named StudentDetails in our sample database. Using this table, we will be able to display our data we stored in the table along with a different control. Let's now take a detailed look at all of the standard controls used in ASP.NET.

➤ Repeater Control:

The Repeater control was introduced with ASP.NET 1.0. The ASP.NET Repeater is a basic container control that allows you to create custom lists from any data available to the page. It provides a highly customized interface. It renders a read-only template; in other words, it supports only the ItemTemplate to define custom binding. The Repeater control is a Data Bind Control, also known as container controls. The Repeater control is used to display a repeated list of items that are bound to the control. This control may be bound to a database table, an XML file, or another list of items. It has no built-in layout or styles, so you must explicitly declare all layout, formatting and style tags within the controls templates. You would require writing an explicit code to do paging using this control. The Repeater repeats a layout of HTML you write, it has the least functionality of the rest of the three controls.

The Repeater control supports the following features:

- List format
- No default output
- More control and complexity
- Item as row
- Paging, Sorting and Grouping requires custom code writing
- only Web control that allows you to split markup tags across the templates
- no built-in selection capabilities
- no built-in support for edit, insert and delete capabilities
- no built-in support for paging, sorting and grouping capabilities
- no built-in layout or styles, need to declare all layout, formatting and style tags explicitly within the control's templates
- Strictly emits the markup specified in its templates, nothing more and nothing less.
- Using the data we stored in table Student of our sample database, the Repeater control will look like this.

Roll Number :1
Name:Vijay
City:Jamnagar
Roll Number :2
Name:Rajesh
City:Ahmedabad
Roll Number :3
Name:Akshay
City:Mumbai
Roll Number :4
Name:Chetan
City:Ahmedabad
Roll Number :5
Name:Sagar
City:Ahmedabad

4.4 DataList Control:

The DataList control was introduced with ASP.NET 1.0. DataList is the next step up from a Repeater; except you have very little control over the HTML that the control renders. DataList allows you to repeat columns horizontally or vertically. The DataList control renders data as a table and enables you to display data records in various layouts, such as ordering them in columns or rows. You can configure the DataList control to enable users to edit or delete a record in the table. We can use a DataList control where we need a single-column list. The DataList control works like the Repeater control, used to display the data in a repeating structure, such as a table. It displays data in a format that you can define using a template and styles. However, it arranges the data defined in the template within various HTML structures. This includes options for horizontal or vertical layout and it also allows you to set how the data should be repeated, as flow or table layout. The DataList control does not automatically use a data source control to edit data. Instead, it provides command events in which you can write your own code for these scenarios.

You can configure the DataList control where the user can edit or delete a record in the table.

The DataList control supports the following features:

- Support for binding data source controls such as SqlDataSource, LinqDataSource and ObjectDataSource
- Directional rendering
- Good for columns

- Item as cell
- Updatable
- Control over Alternate item
- Paging function needs handwriting.

rno: 1
name: Vijay
city: Jamnagar

rno: 2
name: Rajesh
city: Ahmedabad

rno: 3
name: Akshay
city: Mumbai

rno: 4
name: Chetan
city: Ahmedabad

rno: 5
name: Sagar
city: Ahmedabad

rno: 6
name: Ajay
city: Mumbai

rno: 7
name: Akshay
city: Baroda

➤ GridView Control:

ASP.NET provides a number of tools for showing tabular data in a grid, including the GridView control. It was introduced with ASP.NET 2.0. The GridView control is used to display the values of a data source in a table.

Each column represents a field where each row represents a record. It can also display empty data. The GridView control provides many built-in capabilities that allow the user to sort, update, delete, select and page through items in the control. The GridView control can be bound to a data source control, in order to bind a data source control, set the DataSourceID property of the GridView control to the ID value of the data source control. It's considered a replacement for the DataGrid control from .NET 1.1. Therefore, it is also known as a super DataGrid. The GridView control offers improvements such as the ability to define multiple primary key fields, improved user

interface customization using bound fields and templates and a new model for handling or canceling events. Performance is slow compared to DataGrid and ListView.

The GridView control supports the following features:

- Improved data source binding capabilities
- Tabular rendering – displays data as a table
- Item as row
- Built-in sorting capability
- Built-in select, edit and delete capabilities
- Built-in paging capability
- Built-in row selection capability
- Multiple key fields
- Programmatic access to the GridView object model to dynamically set properties, handle events and so on
- Richer design-time capabilities
- Control over Alternate item, Header, Footer, Colors, font, borders, and so on.
- Slow performance as compared to Repeater and DataList control

city	eno	name
Jamnagar	Edit Delete Select 1	Vijay
Ahmedabad	Edit Delete Select 2	Rajesh
Mumbai	Edit Delete Select 3	Akshay
Ahmedabad	Edit Delete Select 4	Chetan
Ahmedabad	Edit Delete Select 5	Sagar
Mumbai	Edit Delete Select 6	Ajay
Baroda	Edit Delete Select 7	Akshay
Ahmedabad	Edit Delete Select 8	Rohit
Rajkot	Edit Delete Select 9	Mahesh
Baroda	Edit Delete Select 10	Nikunj

➤ DetailsView control:

The DetailsView control was introduced with ASP.NET 2.0. The DetailsView control uses a table-based layout where each field of the data record is displayed as a row in the control. Unlike the GridView control, the DetailsView control displays one row from a data source at a time by rendering an HTML table. The DetailsView supports both declarative and programmatic data binding. The DetailsView control is often used in master-detail scenarios where the selected record in a master control determines the record to display in the DetailsView control. It shows the details for the row in a separate space. We can customize the appearance of the DetailsView control using its style properties. Alternatively, we can also use Cascading Style Sheets (CSS) to provide

CC-301 Web Application Development- I (Using C#)
 styles to a DetailsView control. A DetailsView control appears as a form of recording and is provided by multiple records as well as insert, update and delete record functions.

The DetailsView control supports the following features:

- Tabular rendering
- Supports column layout, by default two columns at a time
- Optional support for paging and navigation.
- Built-in support for data grouping
- Built-in support for edit, insert and delete capabilities

rno	1
name	Vijay
city	Jamnagar
Edit Delete New	
1 2 3 4 5 6 7 8 9 10	

➤ FormView control:

The FormView was introduced with ASP.NET 2.0. The FormView control renders a single data item at a time from a data source, even if its data source exposes a multiple records data item from a data source. It allows for a more flexible layout when displaying a single record. The FormView control renders all fields of a single record in a single table row. In contrast, the FormView control does not specify a pre-defined layout for displaying a record. Instead, you create templates that contain controls to display individual fields from the record. The template contains the formatting, controls and binding expressions used to lay out the form. When using templates, we can place any control such as a dropdown list, checkbox and we can even place tables and rich controls like a GridView and so on. A FormView is a databound control used to insert, display, edit, update and delete data in ASP.NET that renders a single record at a time. A FormView control is similar to a DetailView in ASP.NET but the only difference is that a DetailsView has a built-in tabular rendering whereas a FormView requires a user-defined template to insert, display, edit, update and delete data.

The FormView control supports the following features:

1. Template driven
2. Supports column layout
3. Built-in support for paging and grouping
4. Built-in support for insert, edit and delete capabilities

rno: 1
name: Vijay
city: Jamnagar
Edit Delete New
1 2 3 4 5 6 7 8 9 10

4.5 ADO.NET:

ADO.NET is a set of classes (a framework) to interact with data sources such as databases and XML files. ADO is the acronym for ActiveX Data Objects. It allows us to connect to underlying data or databases. It has classes and methods to retrieve and manipulate data. The following are a few of the .NET applications that use ADO.NET to connect to a database, execute commands and retrieve data from the database:

- ✓ ASP.NET Web Applications
- ✓ Console Applications
- ✓ Windows Applications.

Various Connection Architectures:

There are the following two types of connection architectures:

1. **Connected architecture:** the application remains connected with the database throughout the processing.
2. **Disconnected architecture:** the application automatically connects/ disconnects during the processing. The application uses temporary data on the application side called a DataSet.

[Understanding ADO.NET and it's class library]

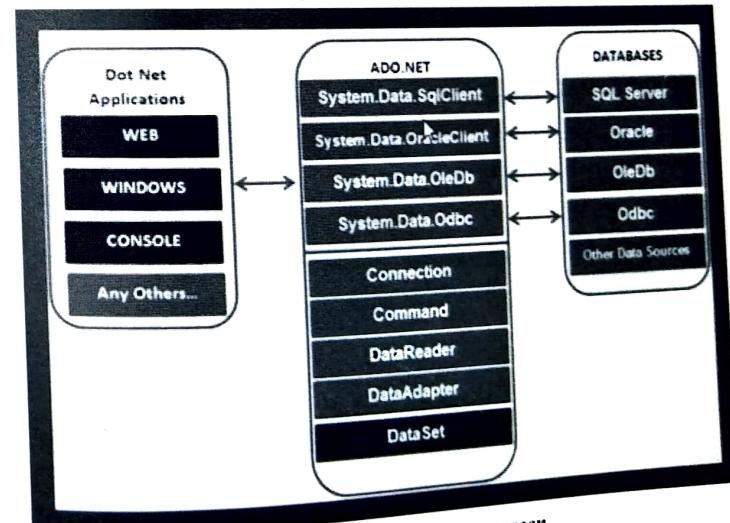


Figure 4.3: class library screen
 Computer World Publication

In this diagram, we can see that there are various types of applications (Web Application, Console Application, Windows Application and so on) that use ADO.NET to connect to databases (SQL Server, Oracle, OleDb, ODBC, XML files and so on).

Important Classes in ADO.NET

We can also observe various classes in the preceding diagram are as follows:

- i. Connection Class
- ii. Command Class
- iii. DataReader Class
- iv. DataAdaptor Class
- v. DataSet Class
- vi. CommandBuilder Class

i. Connection Class:

In ADO.NET, we use these connection classes to connect to the database. These connection classes also manage transactions and connection pooling. To learn more about connection classes.

Properties:

Property	Description
Connection String	Connection string is used to establish and create connection to data source by using server name, database name, user id and password.
State	By this property, we can check your current connection open or close before connection opening or closing

Methods:

Method	Description
Close()	Close method is used, when any current connection is open and finally its closed after completed execution.
Open()	Open method is used, if current connection is close then before execution started. First of all You have opened connection must.

ii. Command Class:

The Command class provides methods for storing and executing SQL statements and Stored Procedures. The following are the various commands that are executed by the Command Class.

Properties:

Property	Description
CommandText	The commandText property is used to set the SQL statement.
Connection	This property is used to get connection to the database source which is specified in SqlConnection.

Methods:

1. **ExecuteReader ()**: Returns data to the client as rows. This would typically be an SQL select statement or a Stored Procedure that contains one or more select statements. This method returns a DataReader object that can be used to fill a DataTable object or used directly for printing reports and so forth.
2. **ExecuteNonQuery()**: Executes a command that changes the data in the database, such as an update, delete, or insert statement, or a Stored Procedure that contains one or more of these statements. This method returns an integer that is the number of rows affected by the query.
3. **ExecuteScalar ()**: This method only returns a single value. This kind of query returns a count of rows or a calculated value.

iii. DataReader Class:

The DataReader is used to retrieve data. It is used in conjunction with the Command class to execute an SQL Select statement and then access the returned rows. A data reader provides an easy way for the programmer to read data from a database as if it were coming from a stream. The DataReader is the solution for forward streaming data through ADO.NET. The data reader is also called a firehose cursor or forward read-only cursor because it moves forward through the data. The data reader not only allows you to move forward through each record of database, but it also enables you to parse the data from each column. The DataReader class represents a data reader in ADO.NET.

Properties:

Item	Gets the value of a column in native format
------	---

Methods:

Read	Reads next record in the data reader.
------	---------------------------------------

iv. DataAdapter Class:

The DataAdapter is used to connect DataSets to databases. The DataAdapter is most useful when using data-bound controls in Windows Forms, but it can also be used to provide an easy way to manage the connection between your application and the underlying database tables, views and Stored Procedures.

The completely new object in the ADO.NET world is the DataAdapter. The purpose of the DataAdapter is embedded in its name: It performs the activities necessary to get the data from the data source on the server into the database that's held in the DataSet. To do that, the DataAdapter lets us specify the commands that should be carried out to retrieve and update data.

In ADO, if we used a client-side Recordset, our data was disconnected from the data source. This had lots of advantages, not the least of which was that we could close our application's connection to the data source and still work with the data. This would free up the connection so that it could be used by other applications, improving our application's scalability.

Properties:

PROPERTY	DESCRIPTION
DeleteCommand	Represents a DELETE statement or stored procedure for deleting records from the data source
InsertCommand	Represents an INSERT statement or stored procedure for inserting a new record to The data source
SelectCommand	Represents a SELECT statement or stored procedure can be used to select records from a data source
UpdateCommand	Represents an UPDATE statement or stored procedure for Updating recording in a data source

Methods:

METHOD	DESCRIPTION
Fill	This method fills data records from a DataAdapter to a DataSet object.
Update	This method stores data from a data set to the data source.

v. DataSet Class:

The DataSet is the heart of ADO.NET. The DataSet is essentially a collection of DataTable objects. In turn each object contains a collection of DataColumn and Computer World Publication

DataRow objects. The DataSet also contains a Relations collection that can be used to define relations among Data Table Objects. The ADO.NET DataSet contains DataTableCollection and their DataRelationCollection. It represents a collection of data retrieved from the Data Source. We can use Dataset in combination with DataAdapter Class. The DataSet object offers disconnected data source architecture. The Dataset can work with the data without knowing the source of the data coming from. That is, the Dataset can work with a disconnected mode from its Data Source. It gives a better advantage over DataReader, because the DataReader is working only with the connection oriented Data Sources. The Dataset contains the copy of the data we requested. The Dataset contains more than one Table at a time. The DataAdapter Object allows us to populate DataTables in a DataSet. We can use Fill method of the DataAdapter for populating data in a Dataset. The DataSet can be filled either from a data source or dynamically. A DataSet can be saved to an XML file and then loaded back into memory very easily. A DataSet can read and write data and schema as XML documents.

vi. CommandBuilder Class:

The DataAdapter is a part of the ADO.NET Data Provider. ADO.NET DataAdapter is used to manage four separate Command objects.

The InsertCommand ,the UpdateCommand and DeleteCommand properties of the SqlDataAdapter object update the database with the data modifications that are run on a DataSet object. The SqlCommand objects that are assigned to these properties can be created manually in code or automatically generated by using the SqlCommandBuilder object. The SqlCommandBuilder opens the Connection associated with the DataAdapter and makes a round trip to the server each and every time it's asked to construct the action queries. It closes the Connection when it's done. The following C# Source Code demonstrate how to use the SqlDataAdapter object to update a SQL Server database with data modifications that are run on a DataSet Object that is populated with data from a table in the database using SqlCommandBuilder object.



Exercises**Q-1 Answer the following questions in details :**

1. Explain GridView, Repeater and DataList control with its features.
2. Explain DetailsView and FormView control with its features.
3. What is ADO.NET? What is connected and disconnected environment?
4. What is connection and Command class? Explain properties and methods of it.
5. Explain DataAdapter and Dataset in detail.
6. Explain membership class and its method.
7. Explain Role class and its methods.
8. Explain ProfileManager class and its method.

Q-2 Answer the following questions in short :

1. Write any two methods of Role Class.
2. What is command object?
3. Write any two Gridview control properties.
4. What is use of DataReader?
5. Explain ExecuteReader() & ExecuteNonQuery() methods of command class.
6. Write difference between FormView Control and ListView Control.
7. What is ADO.NET?

Q-3 Answer the following MCQ :

1. Which of the following is DataReader class property?
(a) init (b) command (c) read (d) item
2. Which of the following is DataAdapter class method?
(a) insert (b) select (c) update (d) delete
3. Database is stored under which of the following Asp.net directory?
(a) App_Code (b) App_Data (c) Bin (d) App_LocalResources
4. Which programming model should you implement if you want to separate your server-side code from your client-side layout code in a Web page?
(a) Single-file model (b) Code-behind model
(c) Inline model (d) Client-server model
5. Which of the following command will return only single value?
 (a) ExecuteScalar (b) ExecuteReader

- (c) ExecuteNonQuery (d) None of above
6. Which are the Command Object Methods?
(a) ExecuteScalar (b) ExecuteReader
(c) ExecuteNonQuery (d) all of above
7. When you create a relationship between tables in ADO.NET DataSet?
(a) A UniqueConstraint is added to the parent table
(b) A ForeignKeyConstraint is added to the child table by default
 (c) Both a and b
(d) None
8. Which ado.net class provide disconnected environment?
(a) DataReader
 (b) DataSet
(c) Command
(d) None of above
9. The first record in a dataset has a position property of :
 (a) Zero
(b) One
(c) Any value defined by programmer
(d) All of above
10. Which of the following is not a DataReader method?
(a) NextResult
(b) GetName
(c) GetValue
(d) GetDataType

MCQ Answers:

- | | | | | |
|-------|-------|-------|-------|--------|
| (1) d | (2) c | (3) b | (4) b | (5) a |
| (6) d | (7) c | (8) b | (9) a | (10) d |



CC-304 Unit - 4 Web Application Development-I Practicals**Program : 1**

Create a database with two tables as StudentInfo, which contain rollno, name and year of a student and StudentScore which contain marks of three subjects for each student. Display name of all the students in a DropDownList and according to user's choice, particular student's score record should display in a FormView/DetailView control.

Default.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default3.aspx.cs" Inherits="Default3" %>

<!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></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

            <asp:DropDownList ID="DropDownList1"
runat="server" AutoPostBack="True"
                DataSourceID="SqlDataSource1"
                DataTextField="name" DataValueField="rno">
            </asp:DropDownList>
            <asp:SqlDataSource ID="SqlDataSource1"
runat="server" ConnectionString="<%$ ConnectionStrings:ConnectionString %>">
```

```
    SelectCommand="SELECT * FROM
[studinfo]"></asp:SqlDataSource>
<br />
<asp:FormView ID="FormView1" runat="server"
BackColor="White"
BorderColor="#3366CC" BorderStyle="None"
BorderWidth="1px" CellPadding="4"
DataKeyNames="rno"
DataSourceID="SqlDataSource2" GridLines="Both">
    <EditItemTemplate>
        rno:
        <asp:Label ID="rnoLabel1" runat="server"
Text='<%# Eval("rno") %>' />
        <br />
        php:
        <asp:TextBox ID="phpTextBox"
runat="server" Text='<%# Bind("php") %>' />
        <br />
        java:
        <asp:TextBox ID="javaTextBox"
runat="server" Text='<%# Bind("java") %>' />
        <br />
        python:
        <asp:TextBox ID="pythonTextBox"
runat="server" Text='<%# Bind("python") %>' />
        <br />
        <asp:LinkButton ID="UpdateButton"
runat="server" CausesValidation="True"
                CommandName="Update" Text="Update" />
        &nbsp;<asp:LinkButton
ID="UpdateCancelButton" runat="server"
                CausesValidation="False"
                CommandName="Cancel" Text="Cancel" />
    </EditItemTemplate>
    <EditRowStyle BackColor="#009999" Font-
Bold="True" ForeColor="#CCFF99" />
```

CC-301 Web Application Development- I (Using C#)

```
<FooterStyle BackColor="#99CCCC"
    ForeColor="#003399" />
    <HeaderStyle BackColor="#003399" Font-
Bold="True" ForeColor="#CCCCFF" />
    <InsertItemTemplate>
        rno:
            <asp:TextBox ID="rnoTextBox"
runat="server" Text='<%# Bind("rno") %>' />
            <br />
        php:
            <asp:TextBox ID="phpTextBox"
runat="server" Text='<%# Bind("php") %>' />
            <br />
        java:
            <asp:TextBox ID="javaTextBox"
runat="server" Text='<%# Bind("java") %>' />
            <br />
        python:
            <asp:TextBox ID="pythonTextBox"
runat="server" Text='<%# Bind("python") %>' />
            <br />
            <asp:LinkButton ID="InsertButton"
runat="server" CausesValidation="True"
                CommandName="Insert" Text="Insert" />
                &nbsp;<asp:LinkButton
ID="InsertCancelButton" runat="server"
                    CausesValidation="False"
CommandName="Cancel" Text="Cancel" />
            </InsertItemTemplate>
            <ItemTemplate>
                rno:
                    <asp:Label ID="rnoLabel" runat="server"
Text='<%# Eval("rno") %>' />
                    <br />
                php:
```

CC-301 Web Application Development- I (Using C#)

```
<asp:Label ID="phpLabel" runat="server"
Text='<%# Bind("php") %>' />
<br />
java:
<asp:Label ID="javaLabel" runat="server"
Text='<%# Bind("java") %>' />
<br />
python:
<asp:Label ID="pythonLabel" runat="server"
Text='<%# Bind("python") %>' />
<br />
</ItemTemplate>
<PagerStyle BackColor="#99CCCC"
ForeColor="#003399" HorizontalAlign="Left" />
<RowStyle BackColor="White"
ForeColor="#003399" />
</asp:FormView>
<asp:SqlDataSource ID="SqlDataSource2"
runat="server"
ConnectionString="<%$ ConnectionStrings:ConnectionString %>">
    SelectCommand="SELECT * FROM [subjectmark]
WHERE ([rno] = @rno)">
    <SelectParameters>
        <asp:ControlParameter
ControlID="DropDownList1" Name="rno"
PropertyName="SelectedValue"
Type="Decimal" />
    </SelectParameters>
</asp:SqlDataSource>
</div>
</form>
</body>
</html>
```

Program : 2

Develop a web application to reserve online in a hotel. The user should enter date of arrival, number of days, room type, number of persons etc. He would be able to confirm booking and allowed to pay advance on confirmation.

Default.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default3.aspx.cs" Inherits="Default3" %>

<!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>Online Hotel Booking</title>
    <style type="text/css">
        .style1
        {
            width:100%;
        }
        .style2
        {
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <table class="style1">
            <tr>
                <td colspan=2> Online Hotel Room Booking
System</td>
            </tr>
            <tr>

```

```
<td class="style2"> Check In Date:</td>
<td>
    <asp:TextBox ID="Textbox1" runat="server"></asp:TextBox>
</td>
</tr>
<tr>
    <td class="style2"> Number of Days:</td>
    <td>
        <asp:TextBox ID="Textbox2" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td class="style2"> Room Type:</td>
    <td>
        <asp:DropDownList ID="DropDownList1" runat="server"
AutoPostBack="true"></asp:DropDownList>
    </td>
</tr>
<tr>
    <td class="style2"> Number of Guest:</td>
    <td>
        <asp:TextBox ID="Textbox3" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td class="style2"> Advance Payment(Rs.):</td>
    <td>
        <asp:TextBox ID="Textbox4" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
```

CC-301 Web Application Development- I (Using C#)

```
<td class="style2" colspan="2">
    <asp:RadioButtonID="RadioButton1" runat="server"
        GroupName="confirm" text="yes"/>
    <asp:RadioButtonID="RadioButton2" runat="server"
        GroupName="confirm" text="no"/>
</td>
</tr>
<tr>
    <td class="style2" colspan="2">
        <asp:ButtonID="Button1" runat="server"
            onClick="Button1_Click" text="Submit"/>
    </td>
</tr>

<tr>
    <td class="style2" colspan="2"> <asp:TextBoxId="Textbox5"
        runat="server" Height="160px"
        TextMode="MultiLine"></asp:TextBox>
    </td>
</tr>
</table>
<div>
</div>
</form>
</body>
</html>
```

default.aspx.cs :

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

CC-301 Web Application Development- I (Using C#)

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

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        String s="Data Source=
        .\\SQLEXPRESS;AttachDbFilename=|DataDirectory|
        \\Database.mdf";
        OleDbConnection cn=new OleDbConnection(s);
        String str="select * from hotel";
        OleDbCommand cmd=new OleDbCommand(str,cn);
        OleDbDataReader dr;
        cn.open();
        dr=cmd.ExecuteReader();
        while(dr.read())
        {
            DropDownList1.Items.Add(dr[1].ToString());
        }
        cn.close();
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        String s="Data Source=
        .\\SQLEXPRESS;AttachDbFilename=|DataDirectory|
        \\Database.mdf";
        OleDbConnection cn=new OleDbConnection(s);
        String str="select price from hotel where
        hotelroomtype='"+DropDownList1.SelectedItem+"'";
        OleDbCommand cmd=new OleDbCommand(str,cn);
        int i;
        cn.open();
        i=Convert.ToInt32(cmd.ExecuteScalar());
```

```

        cn.Close();
        if(RadioButton1.Checked==True)
        {
            Textbox5.Text="Check In Date:" + Textbox1.Text;
            Textbox5.Text+="Number of Days:" + Textbox2.Text;
            Textbox5.Text+="Room Type:" + DropDownList1.SelectedItem;
            Textbox5.Text+="Number of Guest:" + Textbox3.Text;
            Textbox5.Text+="Advance Payment(Rs.):" + Textbox4.Text;
            Textbox5.Text+="Total Bill Amount(Rs.):" +
            Convert.ToString(Convert.ToInt32(Textbox2.Text)*i);
        }
        else
        {
            Textbox5.Text="Booking Cancelled...";
        }
    }
}

```

Program : 3

Create an application to display all records from an Employee table with proper formatting. (Use Repeater Control).

Default.aspx:

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default.aspx.cs" 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></title>
</head>
<body>
    <form id="form1" runat="server">

```

<div>

```

        <asp:Repeater ID="Repeater1" runat="server"
DataSourceID="SqlDataSource1">
        <ItemTemplate>
        <div class="c1">
            Employee Id:<%#Eval("empno") %><br />
            Employee Name:<%#Eval("name") %><br />
            City:<%#Eval("city") %><br />
        </div>
        </ItemTemplate>
        <SeparatorTemplate>
        <hr color="blue" />
        </SeparatorTemplate>
    </asp:Repeater>
    <hr color="blue" />
    <asp:SqlDataSource ID="SqlDataSource1" runat="server"
        ConflictDetection="CompareAllValues"
        ConnectionString="<%$ConnectionString:ConnectionString %>">
        DeleteCommand="DELETE FROM [emp] WHERE [empno]
        = @original_empno AND (([name] = @original_name) OR
        ([name] IS NULL AND @original_name IS NULL)) AND (([city]
        = @original_city) OR ([city] IS NULL AND @original_city IS
        NULL))"
        InsertCommand="INSERT INTO [emp] ([empno],
        [name], [city]) VALUES (@empno, @name, @city)"
        OldValuesParameterFormatString="original_{0}"
        SelectCommand="SELECT * FROM [emp]"
        UpdateCommand="UPDATE [emp] SET [name] =
        @name, [city] = @city WHERE [empno] = @original_empno AND
        ([name] = @original_name) OR ([name] IS NULL AND
        @original_name IS NULL) AND (([city] = @original_city) OR
        ([city] IS NULL AND @original_city IS NULL))">
        <DeleteParameters>

```

Program : 4

Create a product table having field (ProductCode, Name, Price, Description, CategoryName, ProductURL etc.). Design a web page on specific hyperlink pass that category. When user click ProductList.aspx. This page lists the entire product which belongs to the category selected by the user on the previous page. The page has small image of the product and price only. When user clicks on small product image then user will be redirected to ProductDetails.aspx page. Which shows all the details of the product with larger image?

Default.aspx:

```

CC-301 Web Application Development- I (Using C#)

<asp:Parameter Name="original_empno"
    Type="Decimal" />
    <asp:Parameter Name="original_name"
    Type="String" />
    <asp:Parameter Name="original_city"
    Type="String" />
</DeleteParameters>
<InsertParameters>
    <asp:Parameter Name="empno" Type="Decimal"
        >
            <asp:Parameter Name="name" Type="String"
                >
                    <asp:Parameter Name="city" Type="String"
                        >
                            <asp:Parameter Name="name" Type="String"
                                >
                                    <asp:Parameter Name="city" Type="String"
                                        >
                                            <asp:Parameter Name="original_empno"
                                                Type="Decimal" />
                                                <asp:Parameter Name="original_name"
                                                Type="String" />
                                                <asp:Parameter Name="original_city"
                                                Type="String" />
                                                <asp:Parameter Name="original_empno"
                                                    Type="Decimal" />
                                                    <asp:Parameter Name="original_name"
                                                    Type="String" />
                                                    <asp:Parameter Name="original_city"
                                                    Type="String" />
                                                    <asp:Parameter Name="original_name"
                                                        Type="String" />
                                                        <asp:Parameter Name="original_city"
                                                            Type="String" />
                                                            <asp:Parameter Name="original_name"
                                                                Type="String" />
                                                                <asp:Parameter Name="original_city"
                                                                    Type="String" />
                                                                    <asp:Parameter Name="original_name"
                                                                        Type="String" />
                                                                        <asp:Parameter Name="original_city"
                                                                            Type="String" />
                                                                            <asp:Parameter Name="original_name"
                                                                                Type="String" />
                                                                                <asp:Parameter Name="original_city"
                                                                                    Type="String" />
                                                                                    <asp:Parameter Name="original_name"
                                                                                        Type="String" />
                                                                                        <asp:Parameter Name="original_city"
                                                                                            Type="String" />
                                                                                            <asp:Parameter Name="original_name"
                                                                                                Type="String" />
                                                                                                <asp:Parameter Name="original_city"
                                                                                                    Type="String" />
                                                                                                    <asp:Parameter Name="original_name"
                                                                                                        Type="String" />
                                                                                                        <asp:Parameter Name="original_city"
                                                                                                            Type="String" />
                                                                                                            <asp:Parameter Name="original_name"
                                                                                                                Type="String" />
                                                                                                                <asp:Parameter Name="original_city"
                                                                                                                    Type="String" />
                                                                                                                    <asp:Parameter Name="original_name"
                                                                                                                        Type="String" />
                                                        </UpdateParameters>
</asp:SqlDataSource>

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

```

```

<asp:HyperLinkID="H1" runat="server"
NavigateUrl='<%#Eval("category"), "ProductDetail.aspx?i={0}"%>' Text='<%#Eval("category")%></asp:Hyperlink>
</ItemTemplate>
<asp:TemplateField>
</Columns>
<asp:GridView>
<asp:DataSourceID="SqlDataSource1" runat="server"
DataFile="~/Database.mdf"
SelectCommand="SELECT DISTINCT [category] FROM [product]">
</asp:SqlDataSource>
</div>
</form>
</body>
</html>

```

Default2.aspx:

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default2.aspx.cs" 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></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:DetailsViewID="DetailsView1" runat="server"
                AllowPaging="True" AutoGenerateRows="False"
                DataSourceID="SqlDataSource1" Height="80px" width="160px">
            </Fields>
        </div>
    </form>
</body>
</html>

```

```

<asp:BoundFieldDataField="price" HeaderText="price"
SortExpression="Price"/>
<asp:TemplateFieldHeaderText="producturl">
<EditItemTemplate>
<asp:TextBoxID="TextBox1" runat="server"
Text='<%#Bind("producturl")%>'><asp:TextBox>
<InsertItemTemplate>
<asp:TextBoxID="TextBox1" runat="server"
Text='<%#Bind("producturl")%>'><asp:TextBox>
</InsertItemTemplate>
<ItemTemplate>
<asp:ImageButtonID="ImageButton1" runat="server"
Height="50px" Imageurl='<%#Eval("producturl")%>'
PostBackUrl='<%#Eval("producturl", "Default3.aspx?i={0}")%>'>
</ItemTemplate>
</asp:TemplateField>
</Fields>
<asp:DetailsView>
<asp:DataSourceID="SqlDataSource1" runat="server"
DataFile="~/Database.mdf"
SelectCommand="SELECT [price], [producturl] FROM [product]
WHERE ([category]=?)">
<Selectparameters>
<asp:QueryStringParameterName="category"
QueryStringField="i" Type="String"/>
</Selectparameters>
</asp:SqlDataSource>
</div>
</form>
</body>
</html>

```

Default3.aspx:

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default3.aspx.cs" 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></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

<asp:DetailsViewID="DetailsView1" runat="server"
AllowPaging="True" DataKeyNames="PCODE"
AutoGenerateRows="False" DataSourceID="SqlDataSource1"
Height="80px" width="160px">
<Fields>
    <asp:BoundFieldDataField="PCODE" HeaderText="Product Code"
ReadOnly="True" SortExpression="PCODE"/>
    <asp:BoundFieldDataField="PNAME" HeaderText="Product Name"
ReadOnly="True" SortExpression="PNAME"/>
    <asp:BoundFieldDataField="CATEGORY" HeaderText="Category"
ReadOnly="True" SortExpression="Category"/>
    <asp:BoundFieldDataField="PRICE" HeaderText="Product Price"
ReadOnly="True" SortExpression="PRICE"/>
    <asp:BoundFieldDataField="DESC" HeaderText="Product Description"
ReadOnly="True" SortExpression="DESC"/>
<asp:TemplateFieldHeaderText="ProductURL">
<EditItemTemplate>
    <asp:TextBoxID="TextBox1" runat="server"
Text='<%#Bind("ProductURL")%>'><asp:TextBox>
<InsertItemTemplate>
    <asp:TextBoxID="TextBox1" runat="server"
Text='<%#Bind("ProductURL")%>'><asp:TextBox>
</InsertItemTemplate>
<ItemTemplate>

```

```

<asp:ImageID="Image1" runat="server"
Imageurl='<%#Eval("ProductURL")%>' />
</ItemTemplate>
</asp:TemplateField>
</Fields>
</asp:DetailsView>
<asp:DataSourceID="SqlDataSource1" runat="server"
DataFile="~/Database.mdf"
SelectCommand="SELECT * FROM [product] WHERE
([productURL]=?)">
<SelectParameters>
    <asp:QueryStringParameterName="ProductURL"
QueryStringField="i" Type="String"/>
</SelectParameters>
</asp:SqlDataSource>
</div>
</form>
</body>
</html>

```

Program : 5

Create a product table having field (ProductCode, Name, Price, Description, CategoryName, ProductURL etc.). Design a webpage so the Admin can enter the new product details. Use file upload control for the ProductURL field which also upload the product picture to some specific folder called 'ProductImages'. After successful insertion redirect user to the 'ProductList' page where all product details displayed in the GridView with small product image.

Default.aspx:

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="_Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

```

```

    <asp:TextBox ID="txtpcode" runat="server" Width="87px"></asp:TextBox>
    </td>
    <td rowspan="5" style="vertical-align: top;">
        <asp:Image ID="Image1" runat="server" Height="122px" Width="116px" />
    </td>
</tr>
<tr>
    <td class="style2" align="right">
        Name</td>
    <td class="style3">
        <asp:TextBox ID="txtname" runat="server" Width="296px"></asp:TextBox>
    </td>
</tr>
<tr>
    <td class="style2" align="right">
        Photo</td>
    <td class="style3">
        <asp:FileUpload ID="FileUpload1" runat="server" />
    </td>
</tr>
<tr>
    <td class="style2" align="right">
        Price</td>
    <td class="style3">
        <asp:TextBox ID="txtprice" runat="server" Width="295px"></asp:TextBox>
    </td>
</tr>

```

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <style type="text/css">
        .style1 {
            width: 89%;
        }
        .style2 {
            width: 237px;
        }
        .style3 {
            width: 300px;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <div>

            <table class="style1" width="50%">
                <tr>
                    <td colspan="3">
                        <center> <asp:Label ID="LblTitle" runat="server" Text="Product Detail Form" Font-Bold="True" Font-Size="X-Large"></asp:Label></center>
                    </td>
                </tr>
                <tr>
                    <td class="style2" align="right">
                        Productcode</td>
                    <td class="style3">

```

```

<tr>
    <td class="style2" align="right">
        Description</td>
    <td class="style3">
        <asp:TextBox ID="txtdescription"
runat="server" Width="88px"></asp:TextBox>
    </td>

    </tr>
    <tr>
        <td class="style2" align="right">
            Category Name</td>
        <td class="style3">
            <asp:TextBox ID="txtcatname"
runat="server"></asp:TextBox>
        </td>

        </tr>
        <tr>
            <td colspan="3">
                <center>
                    <asp:Label ID="lblMessage"
runat="server" Font-Bold="True" Font-Size="Medium"
                        ForeColor="Blue" Text="Data
Successfully Saved..." Visible="False"></asp:Label>
                    <asp:Button ID="btnSubmit"
runat="server" Text="Submit" onclick="btnSubmit_Click"
                        />
                    <asp:Button ID="btnClear"
runat="server" Text="Clear" />
                    <asp:Button ID="btnDisplay"
runat="server" onclick="btnDisplay_Click"
                        Text="Display" />
                    <asp:HyperLink ID="HyperLink1"
runat="server"

```

NavigateUrl="~/DisplayAllStudInfo.aspx">Display All
Students Information</asp:HyperLink>

```

                </center></td>
            </tr>
        </table>
    
```

```

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

```

Default.aspx.cs:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.IO;
public partial class _Default : System.Web.UI.Page
{
    SqlConnection cn = new SqlConnection("Data
Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\Database.mdf;Integrated Security=True;User Instance=True");
    //static int recptr = 0;
    static DataTable dt = new DataTable();
    SqlDataAdapter da = new SqlDataAdapter();
    DataSet ds = new DataSet();
    protected void Page_Load(object sender, EventArgs e)
    {
    }
}

```

CC-301 Web Application Development- I (Using C#)

```
protected void btnSubmit_Click(object sender,
EventArgs e)
{
    // insert record in table
    FileUpload1.SaveAs(Server.MapPath("~/images/") +
Path.GetFileName(FileUpload1.FileName));
    string img = "images/" +
Path.GetFileName(FileUpload1.FileName);
    string str;
    cn.Open();
    str = "insert into product values(" +
txtpcode.Text + "','" + txtname.Text + "','" + img + "','" +
+ txtprice.Text + "','" + txtdescription.Text + "','" +
txtcatname.Text + ")";
    SqlCommand cmd = new SqlCommand(str, cn);
    cmd.ExecuteNonQuery();
    cn.Close();
    lblMessage.Visible = true;
    Image1.ImageUrl = "images/" +
Path.GetFileName(FileUpload1.FileName);
}

protected void btnDisplay_Click(object sender,
EventArgs e)
{
    String str;
    cn.Open();
    str = "select * from product where rno=" +
txtpcode.Text;
    SqlCommand cmd = new SqlCommand(str, cn);
    SqlDataReader dr;
    dr = cmd.ExecuteReader();
    if (dr.Read())
    {
        txtpcode.Text = dr[0].ToString();
        txtname.Text = dr[1].ToString();
    }
}
```

CC-301 Web Application Development- I (Using C#)

```
Image1.ImageUrl = dr[2].ToString();
txtprice.Text = dr[3].ToString();
txtdescription.Text = dr[4].ToString();
txtcatname.Text = dr[5].ToString();
}
cn.Close();
}
```

DisplayAllInfo.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="DisplayAllStudInfo.aspx.cs"
Inherits="DisplayAllStudInfo" %>

<!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></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:GridView ID="GridView1" runat="server"
AutoGenerateColumns="False"
CellPadding="4" DataKeyNames="rno"
DataSourceID="SqlDataSource1"
AllowPaging="True" BackColor="White"
BorderColor="#CC9966" BorderStyle="None"
BorderWidth="1px">
                <Columns>
                    <asp:BoundField DataField="PCODE"
HeaderText="Product Code" ReadOnly="True"
SortExpression="PCODE" />

```

```

<asp:BoundField DataField="name"
HeaderText="Product Name"
    SortExpression="name" />
<asp:BoundField DataField="price"
HeaderText="price"
    SortExpression="price" />
<asp:BoundField DataField="description"
HeaderText="description"
    SortExpression="description" />
<asp:BoundField DataField="Category"
HeaderText="Category"
    SortExpression="Category" />
<asp:ImageField DataImageUrlField="photo"
HeaderText="Product Image">
    <ControlStyle Height="100px"
Width="100px" />
    </asp:ImageField>
</Columns>
<FooterStyle BackColor="#FFFFCC"
ForeColor="#330099" />
<HeaderStyle BackColor="#990000" Font-
Bold="True" ForeColor="#FFFFCC" />
<PagerStyle BackColor="#FFFFCC"
ForeColor="#330099" HorizontalAlign="Center" />
<RowStyle BackColor="White"
ForeColor="#330099" />
<SelectedRowStyle BackColor="#FFCC66" Font-
Bold="True" ForeColor="#663399" />
<SortedAscendingCellStyle BackColor="#FEFCEB" />
<SortedAscendingHeaderStyle
BackColor="#AF0101" />
<SortedDescendingCellStyle BackColor="#F6F0C0" />
<SortedDescendingHeaderStyle
BackColor="#7E0000" />
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1"
runat="server"

```

```

ConnectionString="<%$ ConnectionStrings:ConnectionString %>" SelectCommand="SELECT * FROM [product]"></asp:SqlDataSource>

```

```

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

```

Program : 6

Create two tables Doctor and Patient. Display details of doctor in first grid view. When user selects doctor from the first grid view, display the details of all patients of that doctor in second grid view.

Default.aspx:

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default.aspx.cs" 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></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:GridView ID="GridView1" runat="server"
AllowPaging="True"
                AllowSorting="True"
                AutoGenerateColumns="False" DataKeyNames="doctorid"
                DataSourceID="SqlDataSource1">
                <Columns>

```

```

        <asp:CommandField ShowSelectButton="True" />
        <asp:BoundField DataField="doctorid"
HeaderText="doctorid" ReadOnly="True"
                SortExpression="doctorid" />
        <asp:BoundField DataField="doctorname"
HeaderText="doctorname"
                SortExpression="doctorname" />
    </Columns>
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1"
runat="server"
    ConnectionString="<%$.ConnectionStrings:ConnectionString %>"
        SelectCommand="SELECT * FROM
[doctor]"></asp:SqlDataSource>
<br />
<asp:GridView ID="GridView2" runat="server"
AllowPaging="True"
    AllowSorting="True"
DataSourceID="SqlDataSource2">
    <Columns>
        <asp:CommandField ShowSelectButton="True" />
    </Columns>
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource2"
runat="server"
    ConnectionString="<%$.ConnectionStrings:ConnectionString %>"
        SelectCommand="SELECT FROM patient INNER JOIN
doctor ON patient.doctorid = doctor.doctorid">
</asp:SqlDataSource>
</div>
</form>
</body>
</html>

```

Write C#.NET code for following using membership and role and other ACL classes:

- Registration page
- Login Form
- Page to add user in the specific role
- List all user details in the GridView

web.config :

```

<?xml version="1.0"?>
<!--
    For more information on how to configure your ASP.NET
application, please visit
    http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
    <system.web>
        <authentication mode="Forms"/>
        <compilation debug="true" targetFramework="4.0"/>
        <profile enabled="true">
            <properties>
                <add name="Fname"/>
                <add name="FClr"/>
                <add name="ProfilePic"/>
            </properties>
        </profile>
    </system.web>
</configuration>

```

Default.aspx :

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default.aspx.cs" 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></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

            Welcome,
            <asp:LoginName ID="LoginName1" runat="server" />
            <br />
            <asp:LoginStatus ID="LoginStatus1" runat="server" />
        </div>
        <asp:Label ID="Label1" runat="server"
            Text="Label"></asp:Label>
    </form>
    <p>
        <a href="SecurePage/ProfilePage.aspx">ProfilePage.aspx</a></p>
    </body>
</html>

```

Login.aspx :

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Login.aspx.cs" Inherits="Login" %>

<!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></title>
</head>
<body>

```

CC-301 Web Application Development- I (Using C#)

```

<form id="form1" runat="server">
    <div>

        <asp:Login ID="Login1" runat="server" BackColor="#F7F7DE" BorderColor="#CCCC99" BorderStyle="Solid" BorderWidth="1px" Font-Names="Verdana" Font-Size="10pt" TitleTextStyle BackColor="#6B696B" Font-Bold="True" ForeColor="#FFFFFF" />
    </asp:Login>
    <br />
    <a href="SignUP.aspx">SignUP.aspx</a>
</form>
</body>
</html>

```

Signup.aspx:

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="SignUP.aspx.cs" Inherits="SignUP" %>

<!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></title>
    <style type="text/css">
        .style1
        {
            width: 100%;
        }
        .style2
        {
            height: 20px;
        }
    </style>
</head>
<body>

```

CC-301 Web Application Development- I (Using C#)

```

        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <div>

            <asp:CreateUserWizard ID="CreateUserWizard1"
runat="server" BackColor="#F7F7DE"
                BorderColor="#CCCC99" BorderStyle="Solid"
BorderWidth="1px"
                    Font-Names="Verdana" Font-Size="10pt"
Height="223px"

oncontinuebuttonclick="CreateUserWizard1_ContinueButtonClick"
onfinishbuttonclick="CreateUserWizard1_FinishButtonClick"
Width="335px">

                <ContinueButtonStyle BackColor="#FFFBBF"
BorderColor="#CCCCCC"
                    BorderStyle="Solid" BorderWidth="1px"
Font-Names="Verdana"
                        ForeColor="#284775" />
                <CreateUserButtonStyle BackColor="#FFFBBF"
BorderColor="#CCCCCC"
                    BorderStyle="Solid" BorderWidth="1px"
Font-Names="Verdana"
                        ForeColor="#284775" />
                <TitleTextStyle BackColor="#6B696B" Font-
Bold="True" ForeColor="#FFFFFF" />
                <WizardSteps>
                    <asp:CreateUserWizardStep runat="server" />
                    <asp:WizardStep runat="server"
Title="Personal Info:>
                        <table class="style1">
                            <tr>

```

CC-301 Web Application Development- I (Using C#)

```

                            First Name:</td>
                        <td>
                            <asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
                        </td>
                        <td>
                            &nbsp;</td>
                        </tr>
                        <tr>
                            <td class="style2">
                                Fav Color:</td>
                            <td class="style2">
                                <asp:TextBox ID="TextBox2"
runat="server"></asp:TextBox>
                            </td>
                            <td class="style2">
                                <asp:FileUpload
ID="FileUpload1" runat="server" />
                            </td>
                            <td>
                                &nbsp;</td>
                        </tr>
                        <tr>
                            <td>
                                Profile Picture:</td>
                            <td>
                                <asp:FileUpload
ID="FileUpload2" runat="server" />
                            </td>
                            <td>
                                &nbsp;</td>
                        </tr>
                    </table>
                </asp:WizardStep>
                <asp:CompleteWizardStep runat="server" />
            </WizardSteps>
            <HeaderStyle BackColor="#6B696B" Font-
Bold="True" ForeColor="#FFFFFF" />
        </HeaderTemplate>
    </asp:FormView>

```

```

        HorizontalAlign="Center" />
    <NavigationButtonStyle BackColor="#FFFBBFF"
    BorderColor="#CCCCCC"
        BorderStyle="Solid" BorderWidth="1px"
    Font-Names="Verdana"
        ForeColor="#284775" />
    <SideBarButtonStyle BorderWidth="0px" Font-
    Names="Verdana"
        ForeColor="#FFFFFF" />
    <SideBarStyle BackColor="#7C6F57"
    BorderWidth="0px" Font-Size="0.9em"
        VerticalAlign="Top" />
    <StepNavigationTemplate>
        <asp:Button ID="StepPreviousButton"
    runat="server" BackColor="#FFFBF0"
        BorderColor="#CCCCCC"
    BorderStyle="Solid" BorderWidth="1px"
        CausesValidation="False"
    CommandName="MovePrevious" Font-Names="Verdana"
        ForeColor="#284775" Text="Previous" />
        <asp:Button ID="StepNextButton"
    runat="server" BackColor="#FFFBF0"
        BorderColor="#CCCCCC"
    BorderStyle="Solid" BorderWidth="1px"
        CommandName="MoveNext" Font-
    Names="Verdana" ForeColor="#284775" Text="Next" />
    </StepNavigationTemplate>
    <StepStyle BorderWidth="0px" />
</asp:CreateUserWizard>
<br />

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

```

ProfilePage.aspx :

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="ProfilePage.aspx.cs"
Inherits="SecurePage_ProfilePage" %>

<!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></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

            First Name:
            <asp:TextBox ID="TextBox1"
    runat="server"></asp:TextBox>
            <br />
            Fav. Color:
            <asp:TextBox ID="TextBox2"
    runat="server"></asp:TextBox>
            <br />
            <br />
            <asp:Image ID="Image1" runat="server"
Width="200px" />

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

```

Program : 8

Design a webpage which allow user to perform Select, Insert, Update and Delete record operation to the database table using ADO.NET code. Also provide to see First, Last, Next and Previous record in textbox.

Default.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default.aspx.cs" 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></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

            <asp:GridView ID="GridView1" runat="server"
CellPadding="4" ForeColor="#333333"
                GridLines="None">
                <AlternatingRowStyle BackColor="White" />
                <EditRowStyle BackColor="#2461BF" />
                <FooterStyle BackColor="#507CD1" Font-
Bold="True" ForeColor="White" />
                <HeaderStyle BackColor="#507CD1" Font-
Bold="True" ForeColor="White" />
                <PagerStyle BackColor="#2461BF"
ForeColor="White" HorizontalAlign="Center" />
                <RowStyle BackColor="#EFF3FB" />
                <SelectedRowStyle BackColor="#D1DDF1" Font-
Bold="True" ForeColor="#333333" />
            <SortedAscendingCellStyle BackColor="#F5F7FB" />
```

CC-301 Web Application Development- I (Using C#)

```
<SortedAscendingHeaderStyle BackColor="#6D95E1" />
<SortedDescendingCellStyle BackColor="#E9EBEF" />
<SortedDescendingHeaderStyle
BackColor="#4870BE" />
</asp:GridView>
<br />
<asp:Label ID="Label1" runat="server" Text="Roll
No:></asp:Label>
<asp:TextBox ID="txtrno"
runat="server"></asp:TextBox>
<br />
<asp:Label ID="Label2" runat="server"
Text="Name:></asp:Label>
<asp:TextBox ID="txtname"
runat="server"></asp:TextBox>
<br />
<asp:Label ID="Label3" runat="server"
Text="City:></asp:Label>
<asp:TextBox ID="txtcity"
runat="server"></asp:TextBox>
<br />
<asp:Button ID="btninsert" runat="server"
onclick="btninsert_Click"
Text="Insert" />

<asp:Button ID="btnupdate" runat="server"
onclick="btnupdate_Click"
Text="Update" />
<asp:Button ID="btndelete" runat="server"
onclick="btndelete_Click"
Text="Delete" />
<asp:Button ID="btnSelect" runat="server"
onclick="btnSelect_Click"
Text="Select" />
```

```

<br />
<asp:Button ID="btnFirst" runat="server"
Text="First"
    onclick="btnFirst_Click" />
<asp:Button ID="btnNext" runat="server"
Text="Next" onclick="btnNext_Click" />
<asp:Button ID="btnPrevious" runat="server"
Text="Previous"
    onclick="btnPrevious_Click" />
<asp:Button ID="btnLast" runat="server"
Text="Last" onclick="btnLast_Click" />

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

```

Default.aspx.cs:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;

public partial class _Default : System.Web.UI.Page
{
    SqlConnection cn = new SqlConnection("Data
Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\Dat
abase.mdf;Integrated Security=True;User Instance=True");
    static int recptr = 0;
    static DataTable dt = new DataTable();
    SqlDataAdapter da = new SqlDataAdapter();

```

```

DataSet ds = new DataSet();
protected void Page_Load(object sender, EventArgs e)
{
    //select record(display) record from table to
    Gridview
    if (!IsPostBack)
    {
        recptr = 0;
        fillTextbox();
        fillGrid();
    }
}

public void fillGrid()
{
    //select record(display record from table
    cn.Open();
    SqlCommand cmd = new SqlCommand("select * from
student", cn);
    da.SelectCommand = cmd;
    da.Fill(ds);
    GridView1.DataSource = ds.Tables[0];
    GridView1.DataBind();
    cn.Close();
}

public void clearTextbox()
{
    txtrno.Text = "";
    txtname.Text = "";
    txtcity.Text = "";
}

protected void btninsert_Click(object sender,
EventArgs e)
{
    // insert record in table
}

```

```

String str;
cn.Open();
str = "insert into student values('" + txtrno.Text
+ "','" + txtname.Text + "','" + txtcity.Text + "')";
SqlCommand cmd = new SqlCommand(str, cn);
cmd.ExecuteNonQuery();

cn.Close();
fillGrid();
}

protected void btnupdate_Click(object sender,
EventArgs e)
{
    // Update record in table
    String str;
    cn.Open();
    str = "update student set name='" + txtname.Text +
"',city='" + txtcity.Text + "' where rno=" + txtrno.Text ;
    SqlCommand cmd = new SqlCommand(str, cn);
    cmd.ExecuteNonQuery();
    cn.Close();
    fillGrid();
    clearTextbox();
}

protected void btndelete_Click(object sender,
EventArgs e)
{
    //Delete record from table
    String str;
    cn.Open();
}

```

```

str = "delete from student where rno=" +
txtrno.Text;
SqlCommand cmd = new SqlCommand(str, cn);
cmd.ExecuteNonQuery();
cn.Close();
fillGrid();
clearTextbox();
}

protected void btnSelect_Click(object sender,
EventArgs e)
{
    //select record(display) record from table to
    Textbox
}

```

```

String str;
cn.Open();
str = "select * from student where rno=" +
txtrno.Text;
SqlCommand cmd = new SqlCommand(str, cn);
SqlDataReader dr;
dr=cmd.ExecuteReader();
if (dr.Read())
{
    txtrno.Text = dr[0].ToString();
    txtname.Text = dr[1].ToString();
    txtcity.Text = dr[2].ToString();
}

cn.Close();
}

public void fillTextbox()
{
    dt.Clear();
    da = new SqlDataAdapter("select * from student", cn);
}

```

```

da.Fill(dt);
txtrno.Text = dt.Rows[recptr][0].ToString();
txtname.Text = dt.Rows[recptr][1].ToString();
txtcity.Text = dt.Rows[recptr][2].ToString();
}
protected void btnFirst_Click(object sender, EventArgs e)
{
    recptr = 0;
    fillTextbox();
}
protected void btnNext_Click(object sender, EventArgs e)
{
    if (recptr < dt.Rows.Count - 1)
    {
        recptr++;
        fillTextbox();
    }
}

protected void btnPrevious_Click(object sender,
EventArgs e)
{
    if (recptr > 0)
    {
        recptr--;
        fillTextbox();
    }
}
protected void btnLast_Click(object sender, EventArgs e)
{
    recptr = dt.Rows.Count - 1;
    fillTextbox();
}
}

```

Program : 9

Design a webform to enter new Employee details includes EmpCode, Name, Address, DeptCode, StateCode and CityCode. EmpCode should be automatically generated like E001, E002, E010 and so on. When user press insert button. Use DropDownList and fetch codes based on user selection. On the click of the insert button store these details in the Employee table.

Default.aspx:

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default.aspx.cs" 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></title>
    <style type="text/css">
        .style1
        {
            width: 100%;
        }
        .style2
        {
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <table class="style1">
            <tr>
                <td class="style2">
                    Employee Code:</td>
                <td>

```

CC-301 Web Application Development- I (Using C#)

```
<asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td class="style2">
Employee Name:</td>
<td>
<asp:TextBox ID="TextBox2"
runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td class="style2">
Address:</td>
<td>
<asp:TextBox ID="TextBox3" runat="server"
Height="63px" TextMode="MultiLine"></asp:TextBox>
</td>
</tr>
<tr>
<td class="style2">
Select Department:</td>
<td>
<asp:DropDownList ID="DropDownList1"
runat="server">
</asp:DropDownList>
</td>
</tr>
<tr>
<td class="style2">
Select State:</td>
<td>
<asp:DropDownList ID="DropDownList2"
runat="server">
```

CC-301 Web Application Development- I (Using C#)

```
</asp:DropDownList>
</td>
</tr>
<tr>
<td class="style2">
Select City:</td>
<td>
<asp:DropDownList ID="DropDownList3"
runat="server">
</asp:DropDownList>
</td>
</tr>
<tr>
<td class="style2" colspan="2">
<asp:Button ID="Button1" runat="server"
Text="Submit" />
<asp:Button ID="Button2" runat="server"
Text="Cancel" />
</td>
</tr>
</table>
<div>
</div>
</form>
</body>
</html>
```

Default.aspx.cs:

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

```

using System.Data.SqlClient;

public partial class _Default : System.Web.UI.Page
{
    SqlConnection cn = new SqlConnection("Data
Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\Dat
base.mdf;Integrated Security=True;User Instance=True");
    SqlCommand cmd;
    SqlDataAdapter da = new SqlDataAdapter();
    DataSet ds = new DataSet();
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            cn.Open();
            DropDownList1.Items.Clear();
            String str = "select * from department";
            cmd = new SqlCommand(str, cn);
            SqlDataReader dr = cmd.ExecuteReader();
            while (dr.Read())
            {
                DropDownList1.Items.Add(dr[1].ToString());
            }
            cn.Close();
            DropDownList2.Items.Clear();
            cn.Open();
            String str2 = "Select * from state";
            cmd = new SqlCommand(str2, cn);
            SqlDataReader dr2 = cmd.ExecuteReader();
            while (dr2.Read())
            {
                DropDownList2.Items.Add(dr[1].ToString());
            }
            cn.Close();
            cn.Open();
        }
    }
}

```

CC-301 Web Application Development- I (Using C#)

```

sid=1";
}

Drop.DropDownList3.Items.Add(dr[1].ToString());
}
cn.Close();
}

protected void
DropDownList2_SelectedIndexChanged(object sender,
EventArgs e)
{
    cn.Open();
    Drop.DropDownList3.Items.Clear();
    String str = "select * from city c,state s where
c.sid=s.sid and s.name='" +
DropDownList2.SelectedItem.ToString() + "'";
    cmd = new SqlCommand(str, cn);
    SqlDataReader dr = cmd.ExecuteReader();
    while (dr.Read())
    {
        Drop.DropDownList3.Items.Add(dr[1].ToString());
    }
    cn.Close();
}

protected void Button1_Click(object sender, EventArgs e)
{
    cn.Open();
    Int32 sid, citycode, deptcode;
}

```

```

String str = "select citycode from city where
cityname=''" + DropDownList3.SelectedItem + "'";
cmd = new SqlCommand(str, cn);
citycode = Convert.ToInt32(cmd.ExecuteScalar());
cn.Close();
cn.Open();
String str2 = "select deptcode from department
where deptname=''" + DropDownList1.SelectedItem + "'";
cmd = new SqlCommand(str2, cn);
deptcode = Convert.ToInt32(cmd.ExecuteScalar());
cn.Close();
cn.Open();
String str3 = "select sid from state where
sname=''" + DropDownList2.SelectedItem + "'";
cmd = new SqlCommand(str3, cn);
sid = Convert.ToInt32(cmd.ExecuteScalar());
cn.Close();
cn.Open();
String empcode = "";
String str4 = "select max(empcode) from employee";
cmd = new SqlCommand(str4, cn);
empcode = Convert.ToString(cmd.ExecuteScalar());
cn.Close();
if (empcode == "")
{
    empcode = "E001";
}
else
{
    Int32 newcode;
    newcode =
Convert.ToInt32(empcode.Substring(1));
    newcode += 1;
    if (newcode.ToString().Length < 2)
{

```

```

    empcode = "E00" + newcode;
}
else
{
    empcode = "E0" + newcode;
}
}
cn.Open();
String str5 = "insert into employee values('" +
empcode + "','" + TextBox2.Text + "', '" + TextBox3.Text +
"', '" + deptcode + "','" + citycode + "','" + sid + "')";
cmd = new SqlCommand(str5, cn);
Response.Write(str5);
cmd.ExecuteNonQuery();
cn.Close();
Response.Write("Record saved successfully");
}
}

```

Program : 10**Create four tables as given:**

- Customer (CustomerCode, Name, Address)
- Product (ProductCode, ProductName, Price, Qty)
- SalesMaster (InvoiceNumber, CustomerCode, DateofInvoice)
- SalesDetails (InvoiceNumber, ProductCode, Qty, UnitPrice)

Design a webpage to generate Invoice details in which Customer name, Product name, and Line total (Unit price * Quantity) is shown using FormView and GridView.

Invoice.aspx:

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Invoice.aspx.cs" Inherits="Invoice" %>

<!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></title>
    <style type="text/css">
        .style1
        {
            width: 100%;
        }
        .style2
        {
            width: 357px;
            height: 123px;
        }
        .style3
        {
            height: 25px;
        }
        .style4
        {
            height: 123px;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Label ID="Label1" runat="server"
                Text="Generate Invoice"></asp:Label>
            <br />
            Order Id:<asp:Label ID="Label2" runat="server"
                Text="Label"></asp:Label>
            <br />
            <asp:Panel ID="Panel1" runat="server">
                <table class="style1" border="1">
                    <tr>

```

```

                <td style="text-align: center"
                    class="style3">
                    Retail Invoice</td>
                </tr>
                <tr>
                    <td>
                        Order No:-<asp:Label ID="Label3"
                            runat="server"></asp:Label>
                        <br />
                        Order Date:-<asp:Label ID="Label4"
                            runat="server"></asp:Label>
                        <br />
                        <asp:Button ID="btnDownload"
                            runat="server" BackColor="#FFFF99" Height="47px"
                            Text="Download Invoice" />
                        onclick="btnDownload_Click"
                    </td>
                </tr>
                <tr>
                    <td>
                        <table class="style1" border="1">
                            <tr>
                                <td class="style2">
                                    Buyer Address:<br />
                                    <asp:Label ID="Label5"
                                        runat="server"></asp:Label>
                                </td>
                                <td class="style4">
                                    Seller Address:<br />
                                    <asp:Label ID="Label6"
                                        runat="server"></asp:Label>
                                </td>
                            </tr>
                            </table>
                        </td>
                    </tr>

```

```

<tr>
    <td>
        <asp:GridView ID="GridView1"
            runat="server" Height="190px" Width="839px"
                AutoGenerateColumns="False">
            <Columns>
                <asp:BoundField
                    DataField="srno" HeaderText="Sr. No.">
                    <ItemStyle
                        HorizontalAlign="Center" />
                </asp:BoundField>
                <asp:BoundField
                    DataField="productid" HeaderText="Product Id">
                    <ItemStyle
                        HorizontalAlign="Center" />
                </asp:BoundField>
                <asp:BoundField
                    DataField="productname" HeaderText="Product Name">
                    <ItemStyle
                        HorizontalAlign="Center" />
                </asp:BoundField>
                <asp:BoundField
                    DataField="qty" HeaderText="Qty">
                    <ItemStyle
                        HorizontalAlign="Center" />
                </asp:BoundField>
                <asp:BoundField
                    DataField="price" HeaderText="Price">
                    <ItemStyle
                        HorizontalAlign="Center" />
                </asp:BoundField>
                <asp:BoundField
                    DataField="totalprice" HeaderText="Total Price">
                    <ItemStyle
                        HorizontalAlign="Center" />
                </asp:BoundField>
            </Columns>

```

```

                </asp:GridView>
            </td>
        </tr>
        <tr>
            <td>
                Grand Total:<asp:Label ID="Label1"
                    runat="server"></asp:Label>
                <asp:FormView ID="FormView1"
                    runat="server"
                    onpageindexchanging="FormView1_PageIndexChanging">
                    </asp:FormView>
                </td>
            </tr>
            <tr>
                <td>
                    <strong style="color:
#FF0000">Declaration : This is a Computer Generated
Invoice and does not required
signature.</strong></td>
                </tr>
            </table>
        </asp:Panel>
    </div>
</form>
</body>
</html>
invoice.aspx.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;

```

```

using System.Data.SqlClient;
using System.IO;
using iTextSharp.text;
using iTextSharp.text.html.simpleparser;
using iTextSharp.text.pdf;

public partial class Invoice : System.Web.UI.Page
{
    String str;
    protected void Page_Load(object sender, EventArgs e)
    {
        SqlConnection cn = new SqlConnection("Data
Source=.\SQLExpress;AttachDbFilename=|DataDirectory|\Database.mdf;Integrated Security=True;User Instance=True");
        str = "select c.custname,p.pname, sd.unitprice *
sd.qty as 'Total' from customer c, product p,salesmaster
sm, salesdetail sd where sm.invoiceno=sd.invoiceno and
sd.pcode=p.pcode and sm.custcode=c.custcode";
        SqlCommand cmd = new SqlCommand(str, cn);
        SqlDataAdapter da = new SqlDataAdapter(cmd);
        DataSet ds = new DataSet();
        da.Fill(ds);
        FormView1.DataSource = ds.Tables[0];
        FormView1.DataBind();
        GridView1.DataSource = ds.Tables[0];
        GridView1.DataBind();
    }

    protected void FormView1_PageIndexChanging(object
sender, FormViewPageEventArgs e)
    {
        FormView1PageIndex = e.NewPageIndex;
    }
}

```



Time: 2:30 Hours]

[Max. Marks: 70]

1. (A) Answer the following :

- (1) Explain features of IDE used to develop ASP.Net Web Application. 7
(2) Explain Label control with its properties. Also, State how it differs from Literal control. 7

OR

Answer the following:

- (1) Explain Application Configuration files in detail.
(2) Explain Panel control. Also, specify how to add a control at runtime in Panel control.

2. (B) Attempt the following: (any four) 4

- (1) Code Behind refers to the code for an ASP.NET Web page that is written in a script tag block in aspx file. (True / False)
(2) CLR stands for Common Language Runtime
(3) Literal control have 3 type(s) of Modes. (One/Two/Three).
(4) TextMode property is used to specify how a TextBox control is displayed on the page. (True / False)
(5) Bin directory contains compiled class file (.dll). (True / False)
(6) The process in which webpage sends data back to the same page on the server is called PostBack/GetPost. (PostBack/GetPost)

2. (A) Answer the following :

- (1) Explain the following methods with example :
(a) QueryString
(b) Session
(2) Explain ImageMap control with its properties.

OR

Answer the following

- (a) Explain FileUpload control with its properties.
(b) Explain Calendar control with its properties.

