



Session 2

Working with
ASP.NET Web
Forms, Controls,
and Events

Session Overview

- Explain Web application development and Web Forms
- Describe event handling in ASP.NET
- List and describe various types of controls in ASP.NET
- Explain Web API and Web security

Introduction to Web Application Development and Web Forms

A Web application is a computer program consisting of HTML Pages with contents and hyperlinks (URL).

It resides on a remote server and passed over the network such as a browser application.

ASP.NET is an advanced technology enabling a fast Web development with lesser code.

ASP.NET offers three programming models for creating Websites - Web Forms, Web Pages, and Model-View-Controller (MVC) Applications.

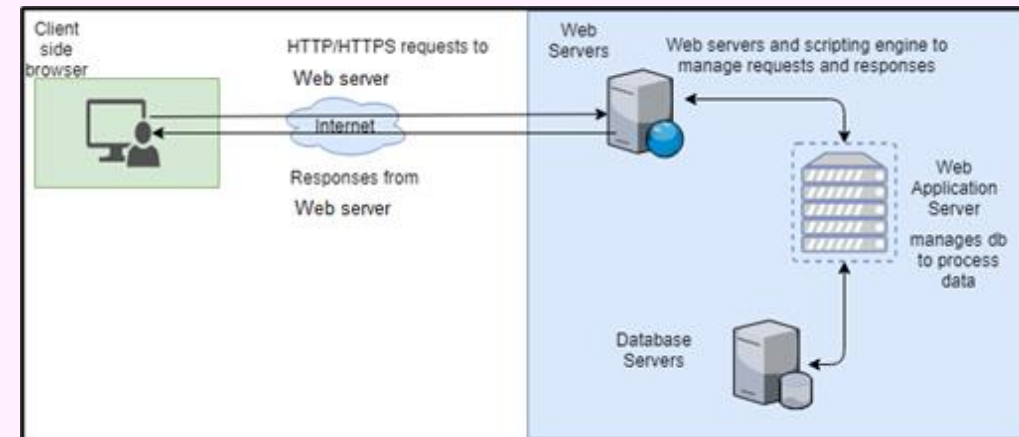


Figure 2.1: Web Application Process

ASP.NET Environment Setup

Microsoft Visual Studio is an exceptionally capable IDE for writing, compiling, and debugging code.

Provides a complete set of development tools for building ASP.NET Web applications, Web services, desktop applications, and mobile applications.

While installing it, one must select appropriate workloads so as to ensure correct templates.

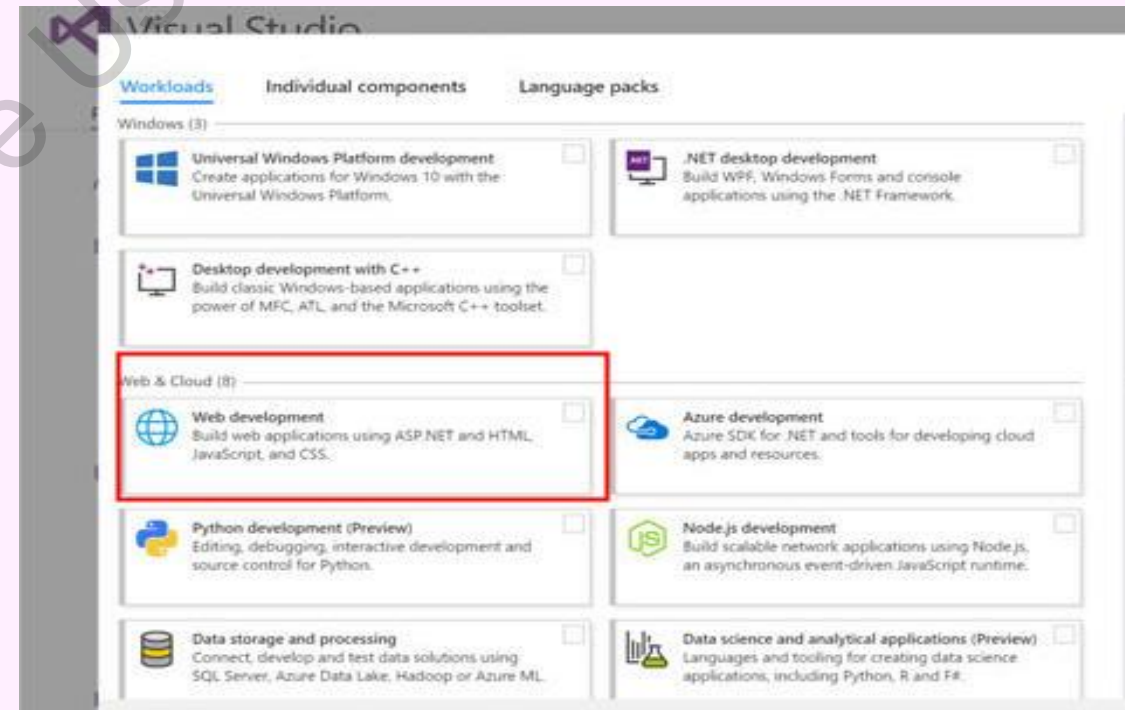


Figure 2.2: Selecting Workloads for Web Development

ASP.NET Web Forms

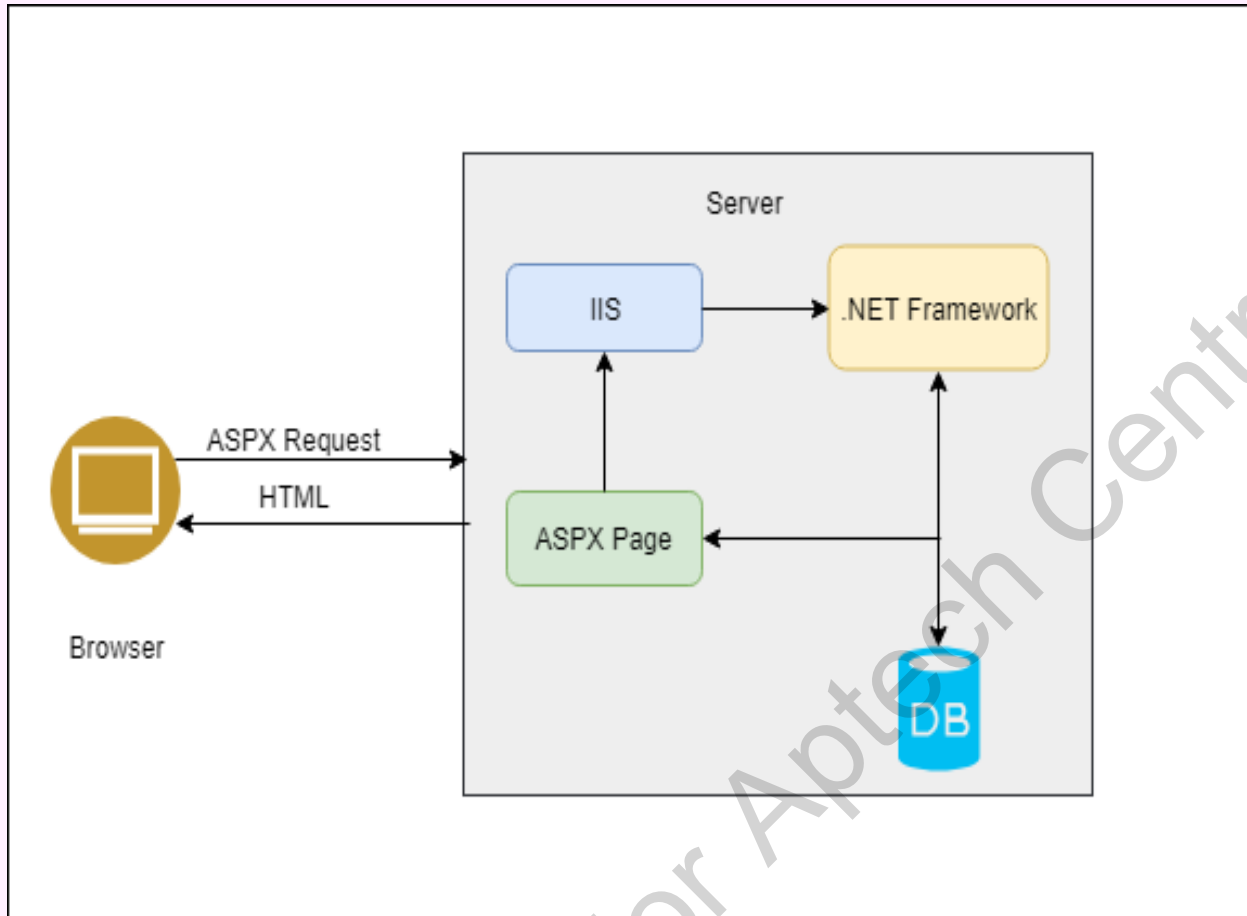


Figure 2.3: ASP.NET Web Form

ASPX

Is a visual based Web page or User Interface (UI).

Code-behind file

Contains server-side code of the Web page that specifies how the Web page and its visual controls should behave upon execution. In a C# based ASP.NET application, it will have .cs extension.

Creating an ASP.NET Web Form (1-4)

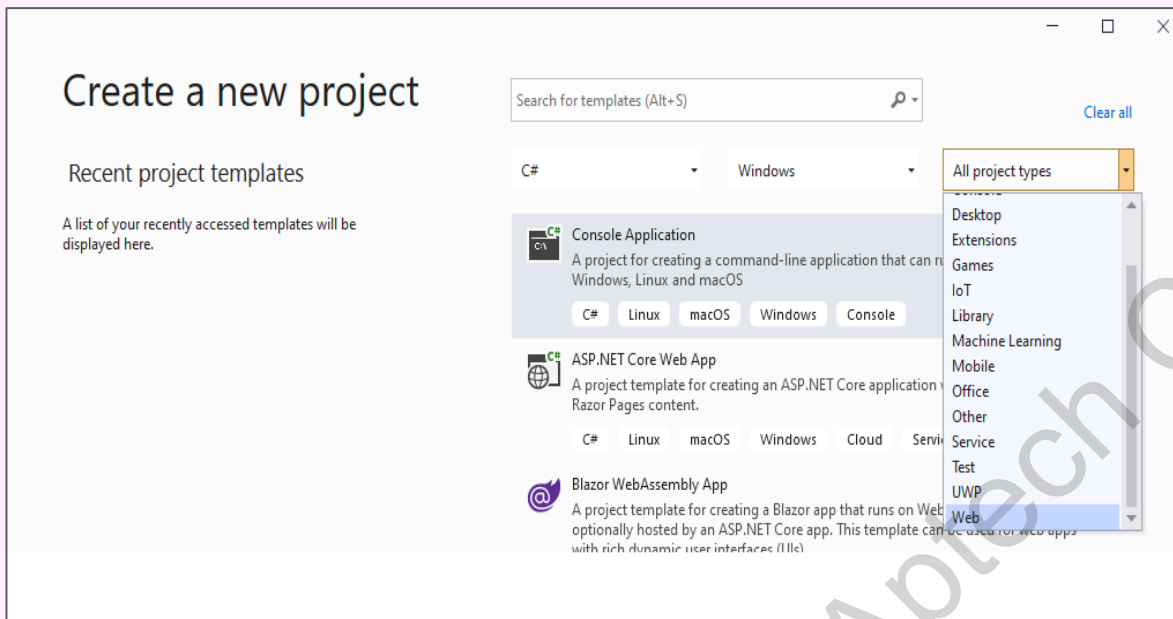


Figure 2.4: Create a New Project

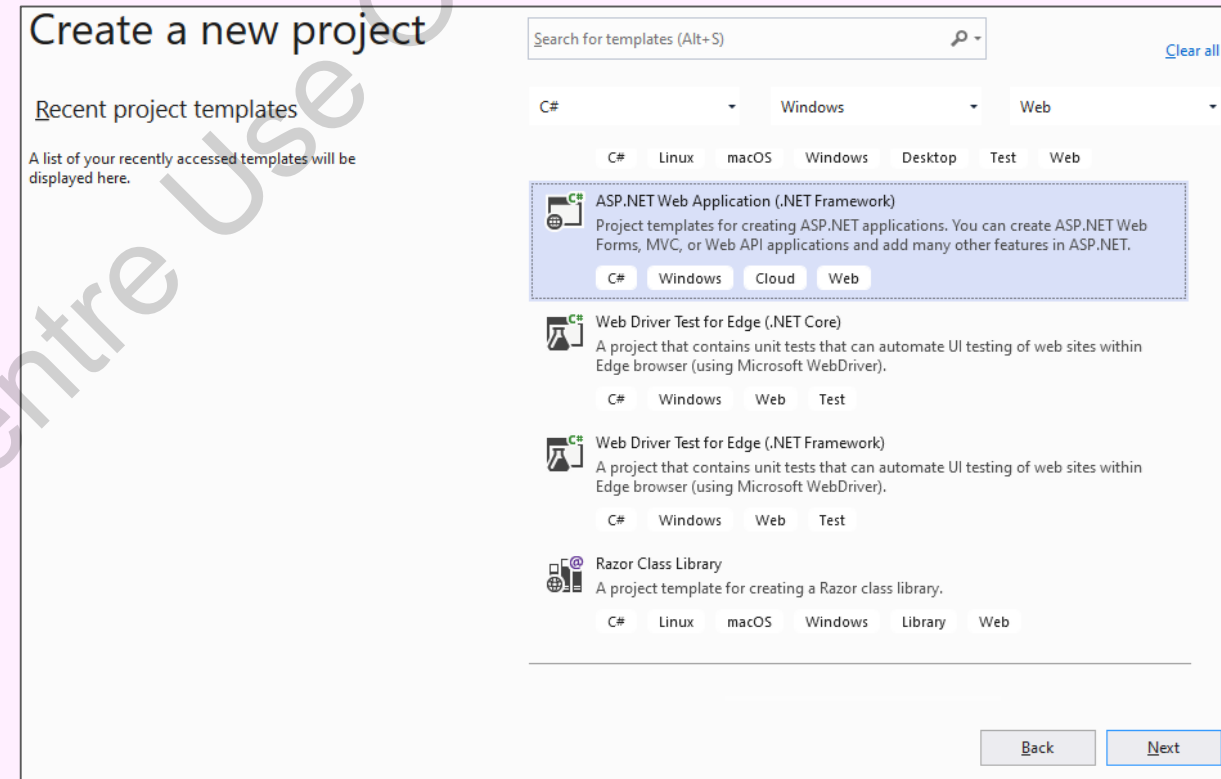


Figure 2.5: ASP.NET Web Application

Creating an ASP.NET Web Form (2-4)

The screenshot shows the 'Configure your new project' window. At the top, it says 'ASP.NET Web Application (.NET Framework)' with tabs for 'C#', 'Windows', 'Cloud', and 'Web'. The 'Project name' field contains 'MyFirstDemo'. The 'Location' field shows 'C:\Users\username\source\repos'. The 'Solution name' field also contains 'MyFirstDemo'. There is a checkbox for 'Place solution and project in the same directory' which is unchecked. The 'Framework' dropdown is set to '.NET Framework 4.7.2'. At the bottom, there are 'Back' and 'Create' buttons.

Figure 2.6: Configure Your New Project Window

The screenshot shows the 'Create a new ASP.NET Web Application' window. It lists four templates: 'Empty', 'Web Forms', 'MVC', and 'Web API'. The 'Web Forms' template is selected and highlighted. To the right, there are sections for 'Authentication' (set to 'No Authentication'), 'Add folders & core references' (with checkboxes for 'Web Forms', 'MVC', and 'Web API'), and 'Advanced' (with checkboxes for 'Configure for HTTPS', 'Docker support', and 'Also create a project for unit tests'). At the bottom, there are 'Back' and 'Create' buttons.

Figure 2.7: Selecting a Web Form

Creating an ASP.NET Web Form (3-4)

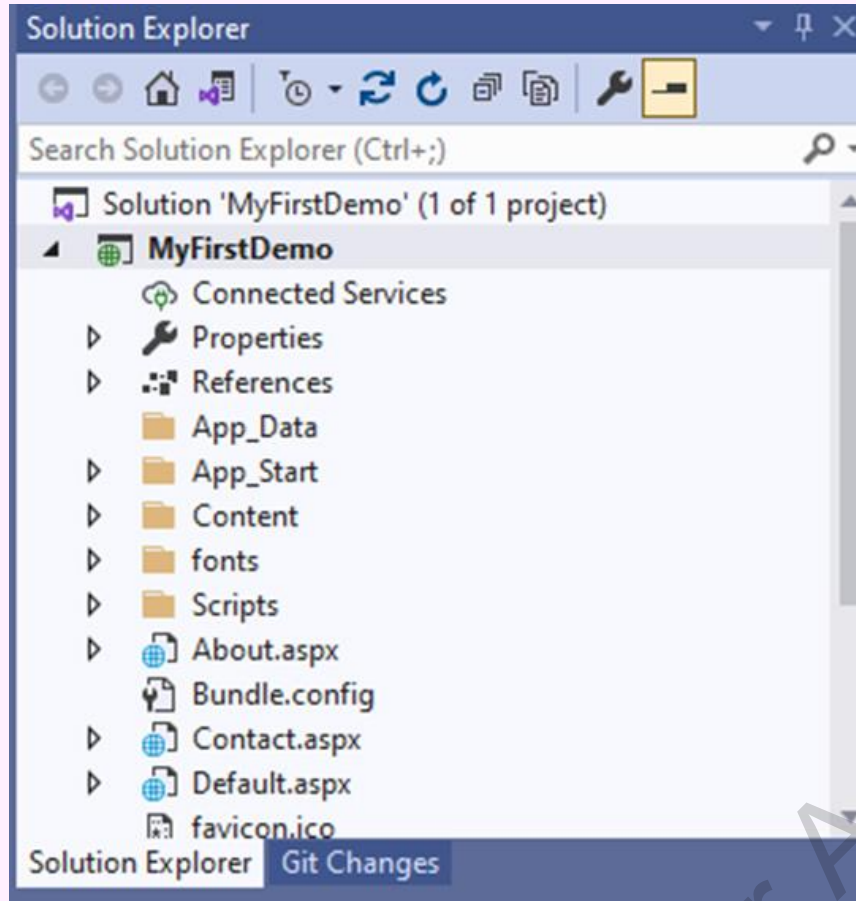


Figure 2.8: Solution Explorer

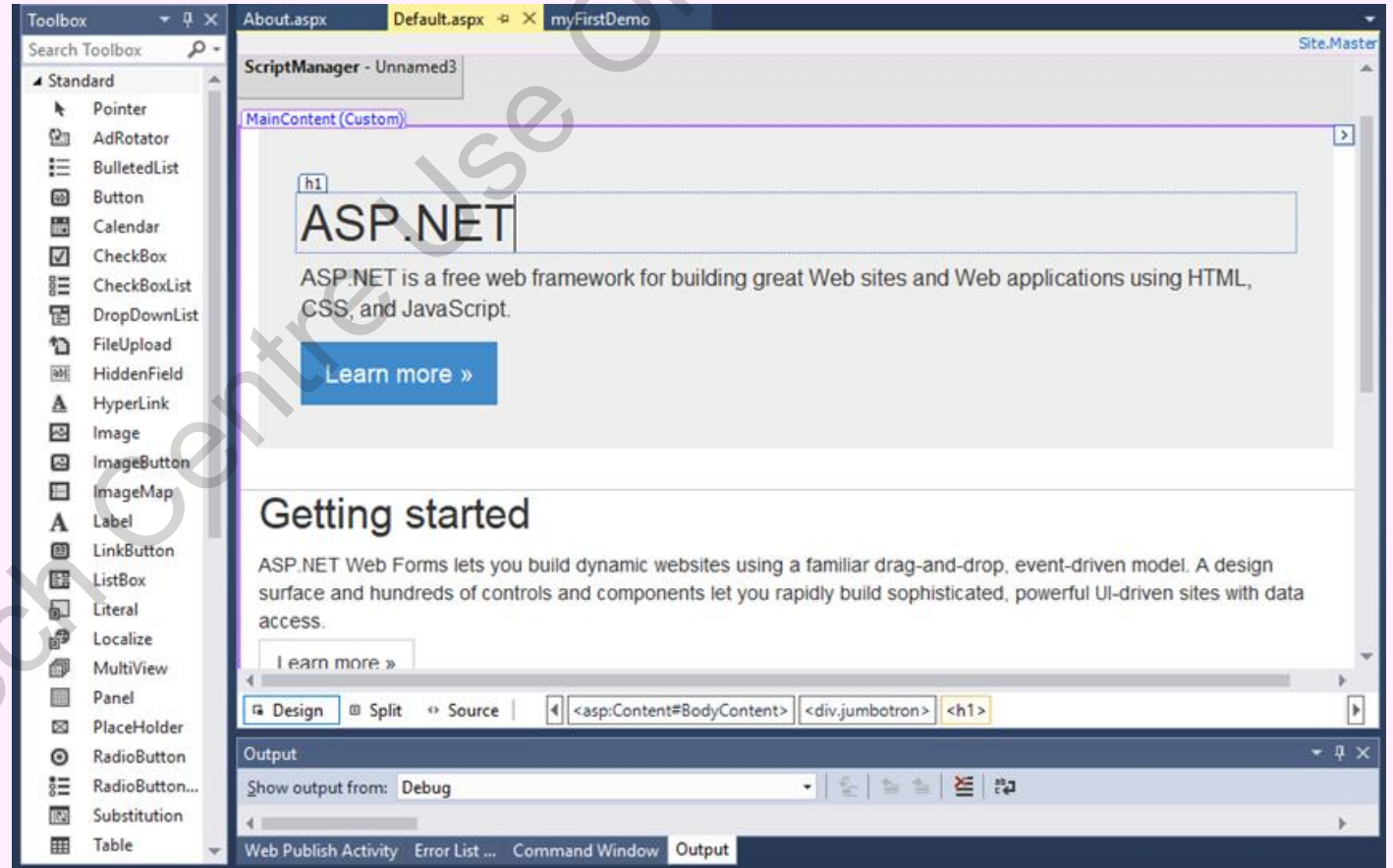


Figure 2.9: Design Output

Creating an ASP.NET Web Form (4-4)

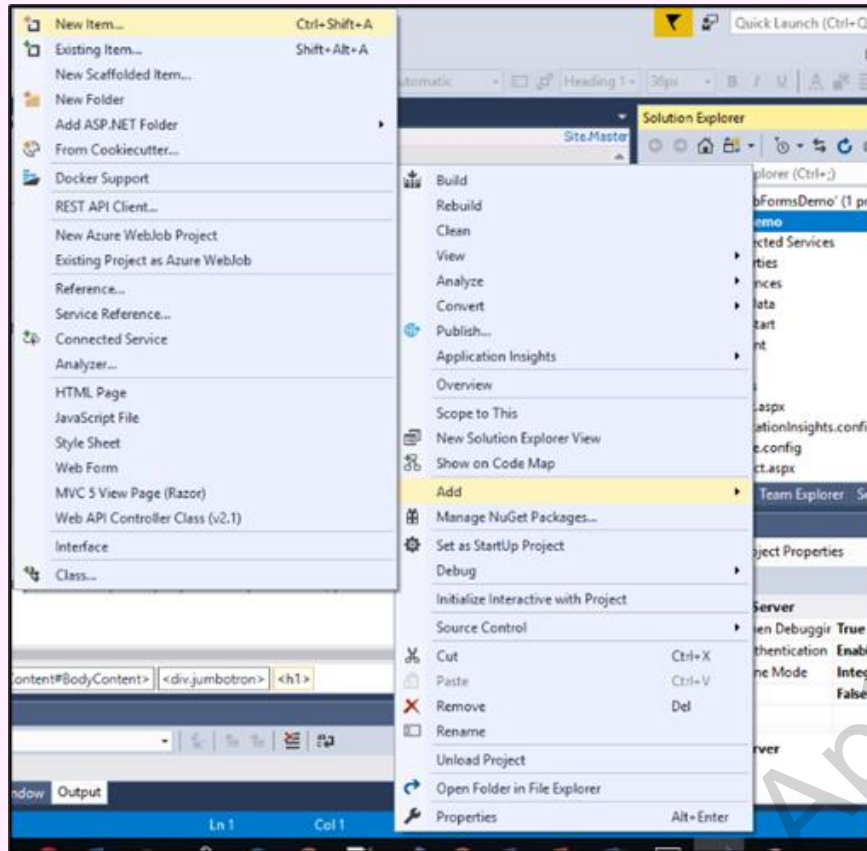


Figure 2.10: Adding a New Item

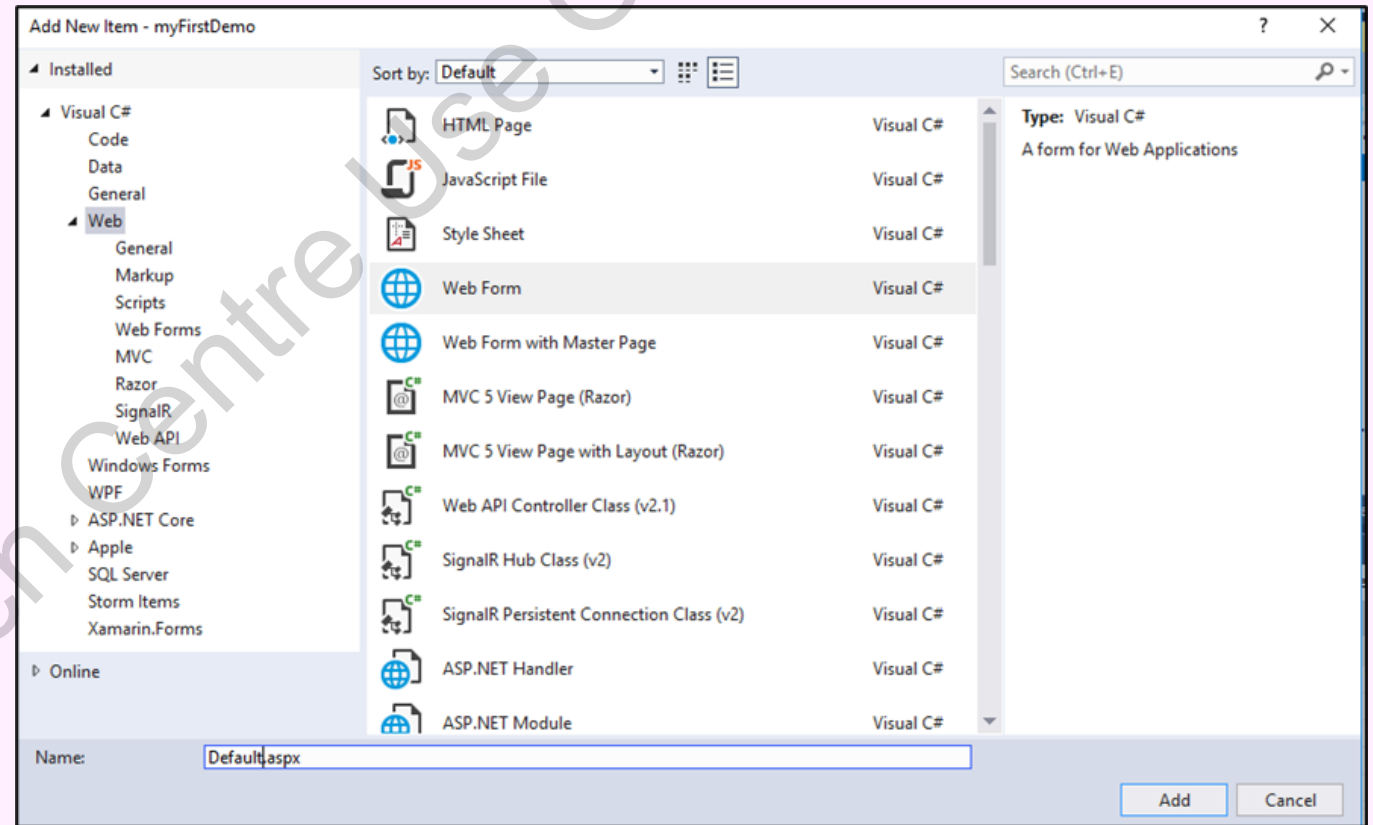


Figure 2.11: Selecting Web Form

Adding Controls to the Form (1-2)

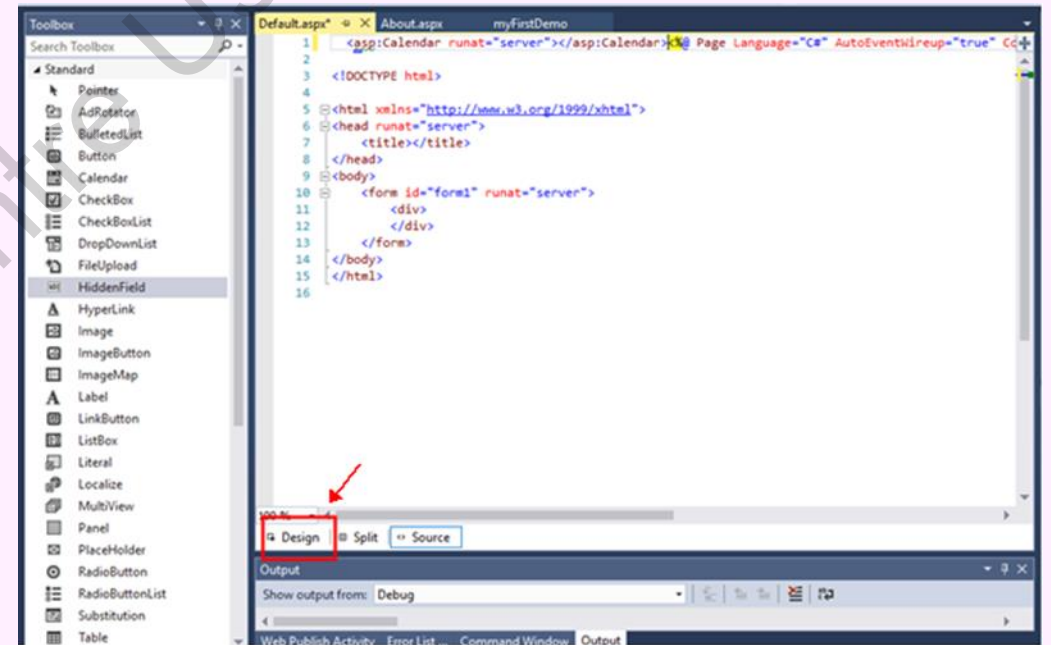
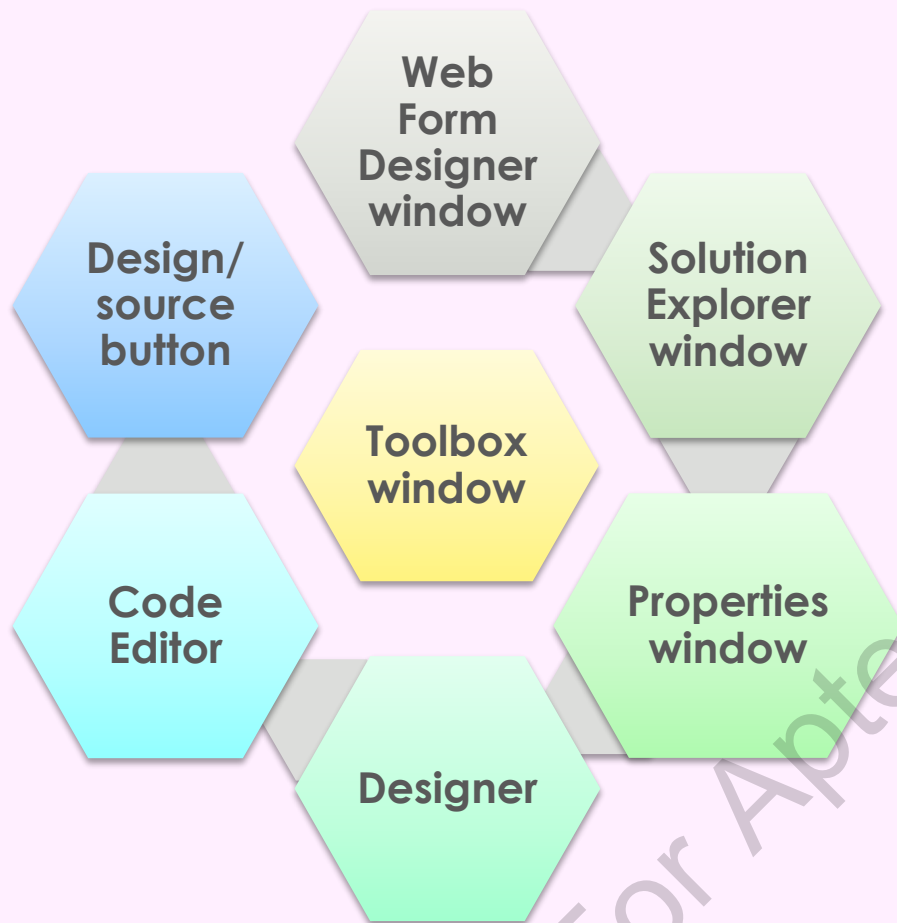


Figure 2.12: Design Tab

Adding Controls to the Form (2-2)

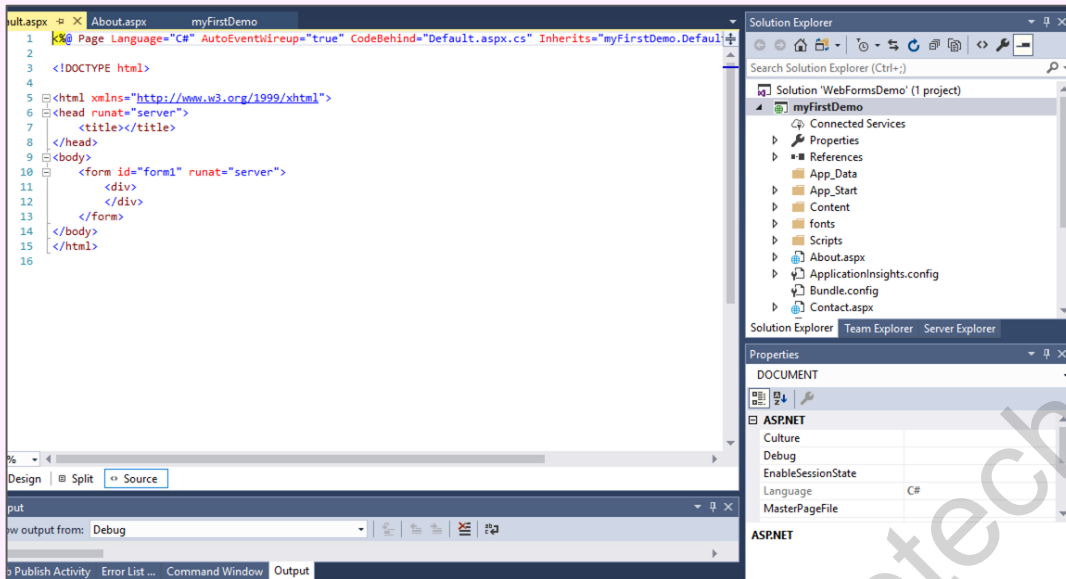


Figure 2.13: Selecting a Control

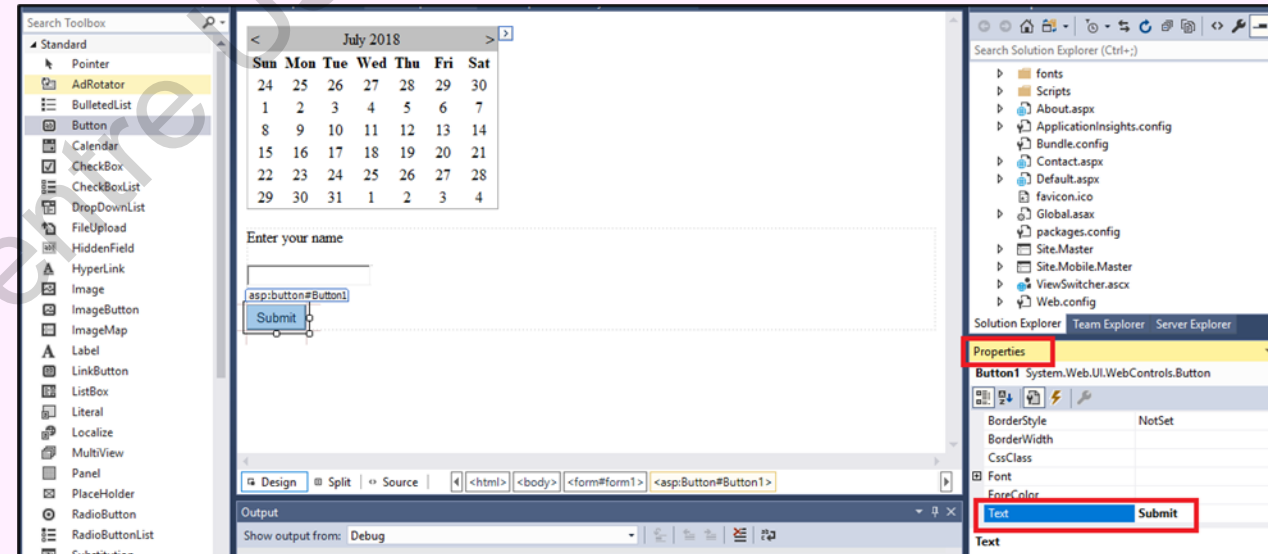


Figure 2.14: Creating a Web Form

Controls in ASP.NET

HTML Server Controls - Traditional HTML tags

Web Server Controls - ASP.NET tags

Validation Server Controls - For input validation

HTML Server Controls (1-4)

Advantages of HTML Server Controls over HTML Elements:

Compiled into the assembly with the `runat="server"` attribute

Map to the corresponding HTML tags

Retain their values, whenever ASP.NET page is reloaded

OnserverEvent is included in most controls for commonly used events

HTML Server Controls (2-4)

Control Name	HTML Tag with Description
HtmlHead	<head>element Used to store page Title and CSS, JavaScript links.
HtmlInputButton	<input type = button submit reset> Similar to HTML Submit Button.
HtmlInputCheck	<input type = checkbox> Used to Check/Uncheck multiple options.
HtmlInputFile	<input type = file> Used to browse and upload files.
HtmlInputHidden	<input type = hidden> Used to store temporary value.
HtmlInputImage	<input type = image> Used to load and display image.

HTML Server Controls (3-4)

Control Name	HTML Tag with Description
HtmlInputpassword	<code><input type = password></code> Textbox that masks password to *.
HtmlInputRadioButton	<code><input type = radio></code> Used to select single option from multiple options.
HtmlInputreset	<code><input type = reset></code> HTML Form reset button.
HtmlText	<code><input type = Text></code> Similar to HTML textbox.
HtmlImage	<code></code> element Used to display image.
HtmlLink	<code><link></code> element Similar to HTML hyperlink.

HTML Server Controls (4-4)

Control Name	HTML Tag with Description
HtmlAnchor	<code><a></code> element Similar to HTML hyperlink.
HtmlButton	<code><button></code> element Similar to HTML button.
HtmlForm	<code><form></code> element HTML form used to add input controls.
HtmlTable	<code><table></code> element HTML table to show data/control in tabular format.

Web Server Controls


Web Server Control	Description
AdRotator	Displays a sequence of images.
Button	Displays a push button.
Calendar	Displays a calendar.
CheckBox	Displays a check box.
CheckBoxList	Creates a multi-selection check box group.
DataGrid	Displays fields of a data source in a grid.
DataList	Displays items from a data source by using templates.
Image	Displays an image.
ImageButton	Displays a clickable image.
Label	Displays static text that displays information as a response to an action or description of how a control will behave when clicked.
ListBox	Creates a single or multi-selection drop-down list.
Table	Creates a table.

ASP.NET Validation Server Controls

Validation Server Control	Description
CompareValidator	Compares the value of one input control to the value of another input control or to a fixed value.
CustomValidator	Allows writing a method to handle the validation of the value entered.
RangeValidator	Checks that the user enters a value that falls between two values.
RegularExpressionValidator	Ensures that the value of an input control matches a specified pattern.
RequiredFieldValidator	Makes an input control a required field.
ValidationSummary	Displays a report of all validation errors that occurred in a Web page.

Event Handling in ASP.NET (1-4)

Sequence of steps while handling Button Click event are:

- 
- User clicks a Button in the client (Browser)
 - Click event is raised by the application
 - Browser publishes this event to the server
 - The event handler is executed by server
 - Notifications generated by the system to the user (Browser)

Event Handling in ASP.NET (2-4)

Application and Session Events

Application_Start

- Raised when the application/Website is started.

Application_End

- Raised when the application/Website is closed.

Session_Start

- Raised when a user requests a page from the application.

Session_End

- Raised when the session ends.

Event Handling in ASP.NET (3-4)

Application State

- Indicates data storage, available for all the classes.

Session State

- Enables a user to store and access values as one browses the pages.

Page and Control Events

PreInit

Init

InitComplete

PreLoad

Load

PreRender

LoadComplete

Render

Unload

Event Handling in ASP.NET (4-4)

Event	Attribute	Control
Click	OnClick	Button, Image Button, Link Button, and IImage map
Command	OnCommand	Button, Image Button, and Link button
TextChanged	OnTextChanged	Text box
SelectedIndexChanged	OnSelectedIndexChanged	Drop-down List, List Box, Radio Button List, and Check Box List
CheckedChanged	OnCheckedChanged	Check box and Radio Button

Exception Handling in ASP.NET (1-2)

Exceptions are run-time errors that disrupt the execution flow of instructions in a program.

Upon encountering an exception, the application terminates.

Unhandled Exceptions are exceptions that occur unexpectedly and are out of scope of the try/catch/finally block.

Exception Handling in ASP.NET (2-2)

Working with Custom Errors

RemoteOnly

If a custom error page exists, it is displayed to the remote user when any exception occurs.

On

The detailed ASP.NET error page is not shown to local users as well. This page is displayed when a custom error page is available.

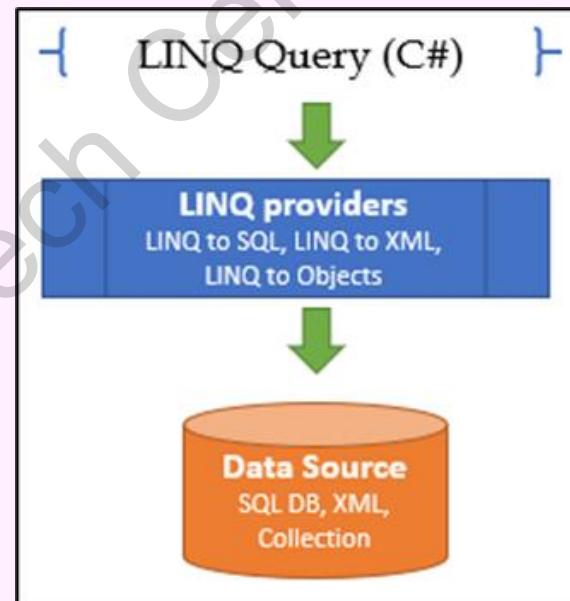
Off

The detailed ASP.NET error page is displayed always, even if a custom error page exists.

Working with LINQ (1-3)



Figure 2.15: LINQ Providers



Working with LINQ (2-3)

Creating an Application with LINQ from XML

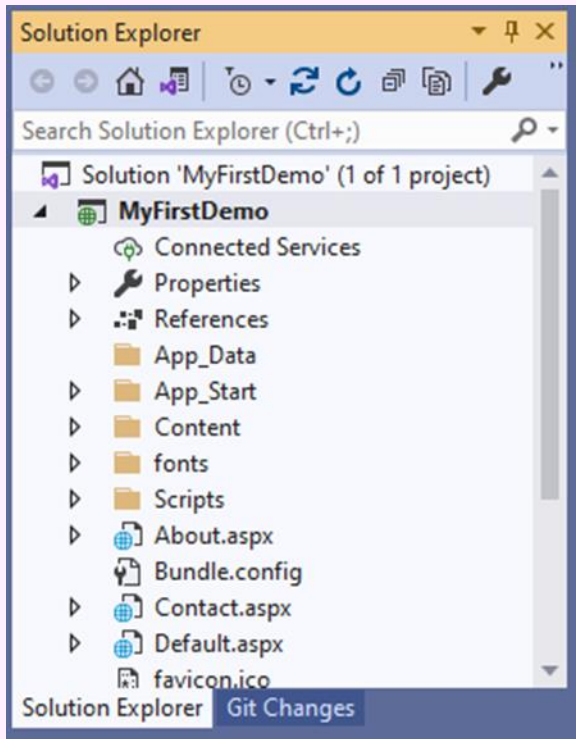


Figure 2.16: MyFirstDemo Solution

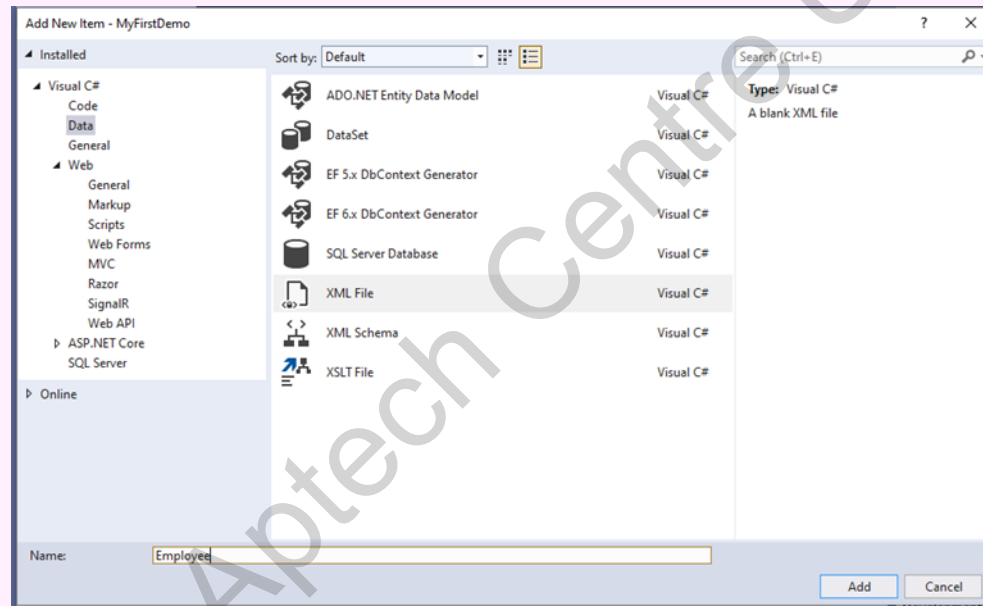


Figure 2.17: Naming the XML File

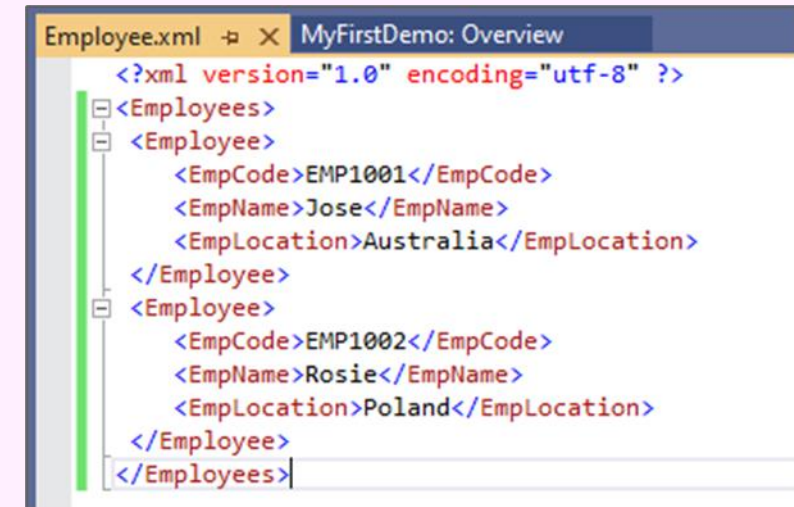


Figure 2.18: Employee.xml

Working with LINQ (3-3)



```
9
10 <form id="form1" runat="server">
11 <div>
12 <%
13 XElement xelementEmployee = XElement.Load(@"C:\Users\username\source\repos\MyFirstDemo\MyFirstDemo\Employee.xml");
14 var result = from list in xelementEmployee.Elements("Employee")
15              where (string)list.Element("EmpLocation") == "Australia"
16              select list;
17 foreach (XElement xEle in result)
18 {
19 <%
20 <%=xEle%>
21 <%
22 }
23 <%
24 </div>
25 </form>
26
```

Figure 2.19: LINQ Code

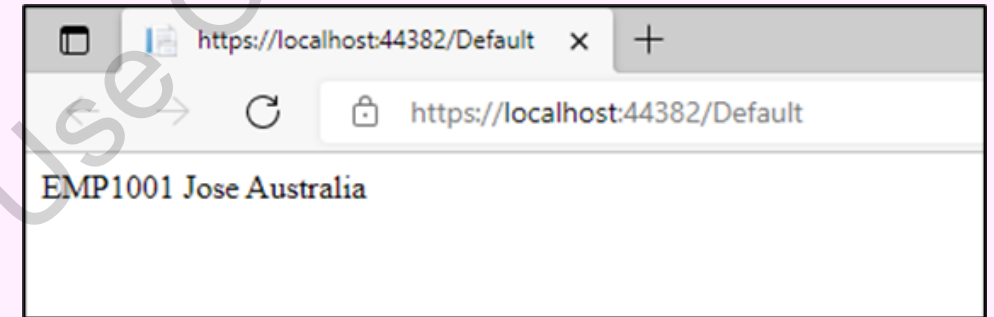


Figure 2.20: LINQ Code Output

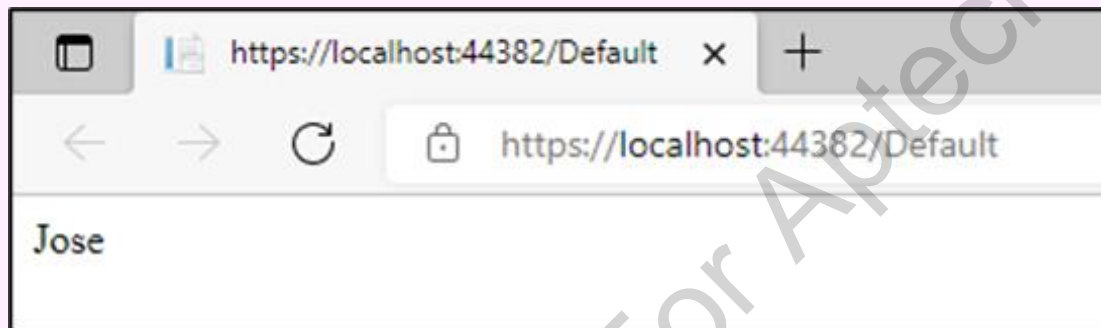
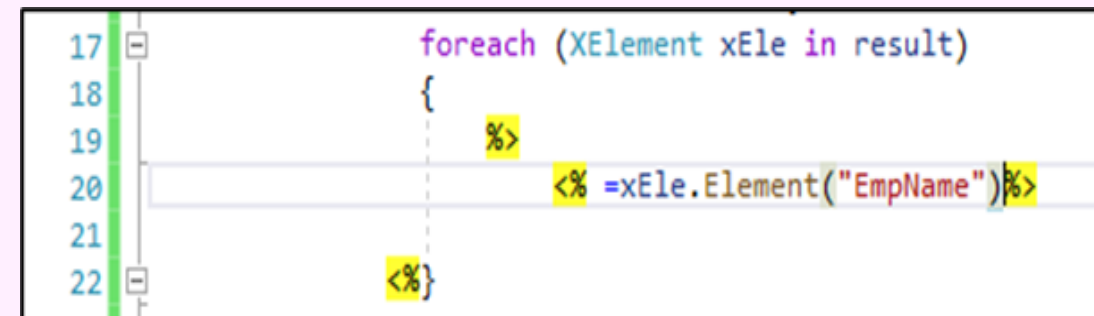


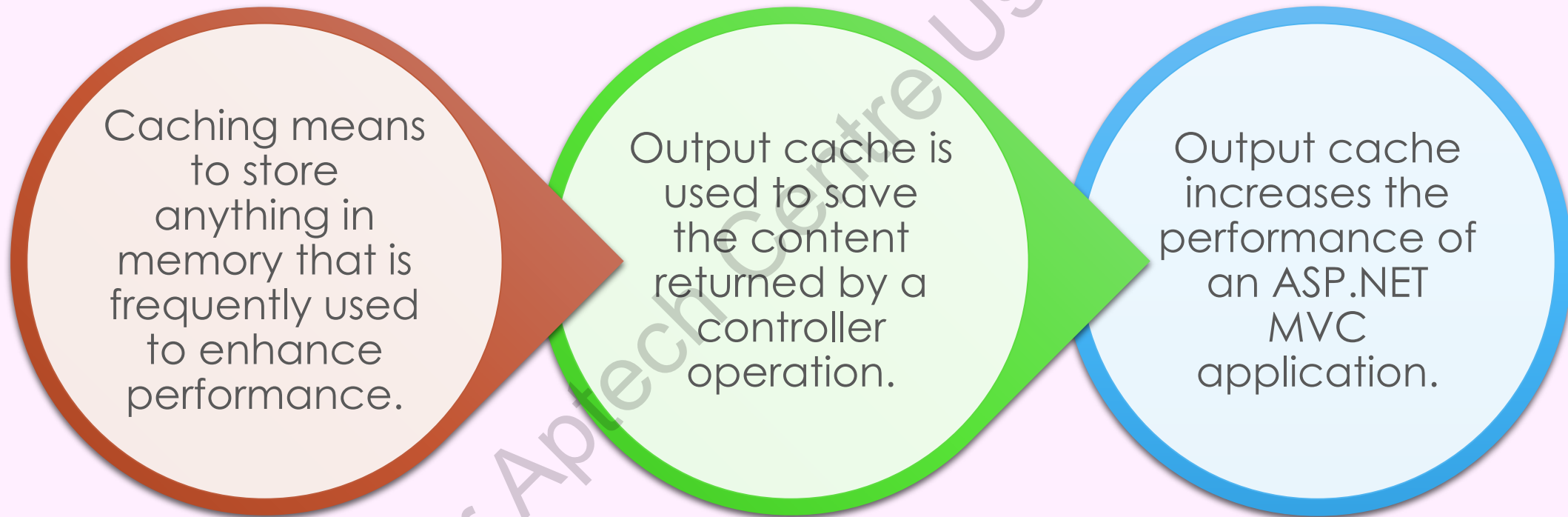
Figure 2.22: Modified Output



```
17 foreach (XElement xEle in result)
18 {
19 <%
20 <%=xEle.Element("EmpName")%>
21 <%
22 }
```

Figure 2.21: Using xEle.Element

Data Caching



Web API

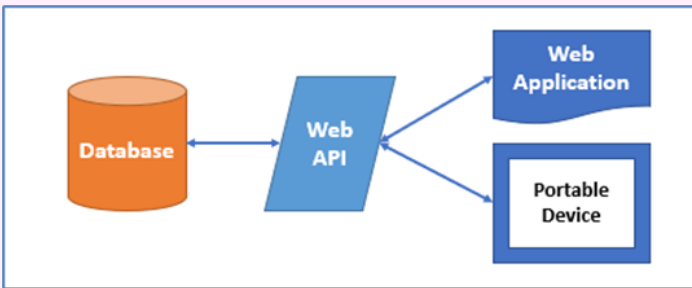


Figure 2.23: Web API Framework

Supports convention-based Create, Read, Update, and Delete (CRUD) actions

Includes an accept header and HTTP status code to the responses

Does formatting of responses into JSON, XML, or any format by Web MediaTypeFormatter, to add as a MediaTypeFormatter

Becomes simple and robust using routing, controllers, action results, and filter. Other features include model binders, IOC container, or dependency injection

Accepts and generates content, such as images, PDF files, and so on

Supports Open Data Protocol (OData) automatically

Hosts within the application or on Internet Information Server (IIS)

Web Security

Windows Authentication

Used for Intranet applications, wherein an organization can extend the functionality of users available over intranet.

Forms Authentication

Used for Internet applications, where different user credentials have been stored in database or other means of data storage.

Passport Authentication

Paid Service by Microsoft that offers a single logon or Single Sign On.

Summary

- ASP.NET Web Forms is one of the best methods to create ASP.NET Websites and Web-based applications. A Web Form can be created using Visual Studio and Web Form templates.
- The types of server controls for input validation are namely, HTML server controls (traditional HTML tags), Web server controls (new ASP.NET tags), and validation server controls.
- Events are handled at the server end when raised at the client end in ASP.NET.
- The keywords try-catch-finally are used to implement exception handling in ASP.NET.
- Output cache is used to increase the performance of an ASP.NET MVC application.
- ASP.NET Web API is a unique technology that makes it easy to create distributed applications.