ASP.NET Questions and Answers

Q1. Explain the role of Web Configuration files in ASP.NET. How does web.config influence the behavior of the web application? (5 marks)

Answer:

The web.config file is a configuration file used in ASP.NET applications to define various settings such as security, session state, error handling, database connections, and more. It is an XML-based file that is automatically created when you set up an ASP.NET project.

Key Functions:

1. Connection Strings: Stores connection strings for databases.
2. Authentication & Authorization: Configures security policies for the application.
3. Error Handling: Specifies custom error pages or detailed error messages.
4. Session State: Configures session state modes (InProc, StateServer, SQLServer).
5. AppSettings: Holds key-value pairs for application-wide settings.

Example of a connection string:

```
<connectionStrings>
    <add name="MyDB" connectionString="Data Source=ServerName;Initial Catalog=DatabaseName;User ID=Username;Password=Password" />
</connectionStrings>
```

Q2. Discuss different types of error pages in ASP.NET. How can you configure custom error pages in the web.config file? (4 marks)

Answer:

In ASP.NET, error pages provide a way to handle errors gracefully. There are two types of error pages:

1. Custom Error Pages: These are user-defined pages displayed to the user when an error occurs. This is useful for hiding technical error details from the user.
2. Default Error Pages: ASP.NET shows a detailed error page by default, but these pages are more suitable for debugging rather than production environments.

Configuring custom error pages:

Custom error pages can be configured in the web.config file like this:

```
<configuration>
  <system.web>
    <customErrors mode="On" defaultRedirect="ErrorPage.aspx">
      <error statusCode="404" redirect="PageNotFound.aspx" />
      <error statusCode="500" redirect="ServerError.aspx" />
    </customErrors>
  </system.web>
</configuration>
```

Q3. Explain ASP.NET Caching. Discuss different types of caching mechanisms used in ASP.NET with examples. (5 marks)

Answer:

ASP.NET caching is a technique used to store frequently used data in memory, allowing faster access to that data and improving performance. There are three types of caching:

1. Output Caching: Caches the dynamic page's output. This allows serving a cached version of the

page for subsequent requests.

<%@ OutputCache Duration="60" VaryByParam="None" %> This caches the output of the page for 60 seconds.

2. Data Caching: Caches specific data in memory, which can be reused across multiple requests. It uses the Cache object.

Cache["Data"] = myData;

3. Fragment Caching: Caches portions of a page (user controls) rather than the entire page.

<%@ OutputCache Duration="60" VaryByParam="None" %> This caches a user control for 60 seconds.

Q4. Differentiate between Common Web Server Controls and Specialized Web Server Controls. Provide two examples of each. (4 marks)

Answer:

- Common Web Server Controls are generic controls that are used to accept user input and provide basic functionality.

  Examples:

  1. TextBox: Used to accept user input.

  2. Button: Triggers postbacks and events on the server.

- Specialized Web Server Controls offer additional features or are tailored to specific tasks.

  Examples:

  1. FileUpload: Allows users to upload files to the server.

  2. Calendar: Displays a graphical calendar for date selection.

Q5. What is the use of the FileUpload control in ASP.NET? Provide a code example showing how to upload a file and save it on the server. (4 marks)

Answer:

The FileUpload control in ASP.NET allows users to select and upload files from their local machine to the server.

Example:

```
<asp:FileUpload ID="FileUpload1" runat="server" />
<asp:Button ID="Button1" Text="Upload" runat="server" OnClick="Button1_Click" />
```

Code-behind:

```
protected void Button1_Click(object sender, EventArgs e)
{
    if (FileUpload1.HasFile)
    {
        string filename = Path.GetFileName(FileUpload1.FileName);
        FileUpload1.SaveAs(Server.MapPath("~/Uploads/" + filename));
        Response.Write("File uploaded successfully!");
    }
}
```

Q6. Explain ASP.NET AJAX and its importance in building responsive web applications. Discuss the role of the UpdatePanel control in AJAX operations. (5 marks)

Answer:

ASP.NET AJAX allows asynchronous communication between the browser and server, enabling parts of the page to update without reloading the entire page, improving responsiveness and user experience.

The UpdatePanel control is central to AJAX operations. It wraps parts of the page (like controls) and allows them to perform partial postbacks instead of full postbacks. The ScriptManager control is required to enable AJAX functionality.

Example:

```
<asp:ScriptManager ID="ScriptManager1" runat="server" />
<asp:UpdatePanel ID="UpdatePanel1" runat="server">
    <ContentTemplate>
        <asp:Label ID="Label1" runat="server" Text="Current Time: " />
        <asp:Button ID="Button1" Text="Update Time" runat="server" OnClick="Button1_Click" />
    </ContentTemplate>
</asp:UpdatePanel>
```

Q7. What is Data Binding in ASP.NET? Describe the difference between Single-view, Repeated-value, and Grid View data controls. (5 marks)

Answer:
Data Binding refers to the process of binding a data source to a control so that the control can display data automatically.

1. Single-view controls: Displays a single item from the data source.
   Example: FormView.

2. Repeated-value controls: Displays multiple items from the data source, repeating them.

Example: Repeater.

3. Grid View: Displays data in a tabular format, providing built-in support for paging and sorting.

   Example: GridView control.