

ASP.NET Interview Question And Answers

Q1. What is ASP.NET?

ASP.NET Framework is a part of the .NET framework used to create a dynamic website, web application, and web services. It is a server-side technology that uses all .NET compatible languages such as [C#](#), VB.NET, J#, etc. which are compiled to Microsoft Intermediate Language (MSIL). [ASP.NET](#) uses server control to develop a rapid and interactive application in an easy way.

Any ASP.Net applications would also be written in multiple choice of .Net languages that include C#, VB.Net, and J# and it provides multiple development modes, that help to develop an application in an easy and better way.

Features of ASP.NET:

- It uses C# and VB.NET languages to build the website.
- It allows us to separate the HTML layout with server-side code.
- It allows us to make the same class name qualifying under a different namespace.
- [ASP.NET](#) pages are compiled, not interpreted.
- ASP.NET is a request processing engine. It takes an incoming request and passes it through its internal pipeline to an endpoint where a developer can attach code to process that request.

Q2. What is the difference between Web Site and a Web Application?

There are the following differences between these two :

Web Site

- On a website, you cannot add multiple projects.
- There is no project file (.csproj or .vbproj). All the files in a folder structure are automatically included on the site.
- By default, compilation produces multiple assemblies.
- The website is easy to create and deploy.
- You can use different .NET languages on a single web site such as VB.NET pages and C# pages can be used on a single website.
- You can edit a single page/file after deployment recompilation is not required.

- Choose a website where one developer will be responsible for creating and managing an entire website. Since decoupling is not possible on the website.
- You cannot establish dependencies on the website.

Web Application

- In a web application, you can add multiple projects.
- It has a Visual Studio project file (.csproj or .vbproj) that stores information about the project like as the list of files that are included in the project, and any project-to-project references.
- By default, compilation of code files (excluding .aspx and .ascx files) produces a single assembly.
- It is easy to develop compared to the website.
- You cannot use different .NET languages in a Web Application such as C# and VB.NET both cannot be used in a web application.
- You cannot edit a single file. Recompilation is required.
- The right choice for enterprise environments where multiple developers work for creating, testing, and deployment. Different groups work on various components independently like one group works on the domain layer, other works on the UI layer hence, decoupling is possible here.
- You can establish dependencies in a Web Application.

Q3. What is a round trip?

The trip of a Web page from the client to the server and then back to the client is known as a round trip. In ASP.NET, `Response.Redirect()` causes a round trip. In other words, any web page that travels from the client-side to the server-side will get processed on the server and travels back to the client, this whole process is called a round trip.

Q4. What is a Global.asax file?

The Global.asax file has been derived from the `HttpApplication` class. The Global.asax file is used to add application-level logic and processing. It does neither handle any UI-related processing, nor it does support individual requests. It basically controls the following events.

- `Application_Start`
- `Application_End`
- `Session_Start`

- Session_End
- **Following are the important aspect of the use of the Global.asax file.**
- The code in the Global.asax is compiled when the web application is built for the first time.
- Session level code and variables can be declared in the Application_Start event.
- Application level events are for the entire application and may be used by any user.
- Session level events are user-specific for the length of a session.

Q5. What is the difference between Web.config and Machine.config file?

Both files are used to define configurations for your ASP.NET application. There are the following differences between these two configuration files.

Web.config

- The Web.config is used to define application-level settings.
- If any setting is not specified in Web.config it inherits from Machine.config by default.
- Web.config is an XML-based configuration file for an ASP.NET application that includes the setting for Data Connection, Customizing Security, State Management, Memory Management, Error Handling, and much more.
- It is sometimes called an application and session level configuration file.
- You can have more than one Web.config file in your ASP.NET application.

Machine.config

- This file is at the highest level in the configuration hierarchy.
- You can find it at `<WinDir>\Microsoft.NET\Framework\<version>\config\machine.config`
- Machine.config is used for defining server level settings.

- It defines the supported configuration file section and ASP.NET work process.
- It registers providers that can be used for advanced features such as profiles membership and role-based security.
- It is automatically installed while installing Visual Studio.NET.
- It is sometimes called a machine level configuration file.
- Only one Machine.config file can be at the machine level.

Q6. How to define a connection string in the Web.config file?

Following is the way to define a connection string in Web.config for SQL Server database.

```
<connectionStrings>
<add name="strcon" connectionString="Data Source=DonetApp; Initial Catalog=dotnetDB; Persist
Security Info=True; User ID=asif@123; Password=123@asif" providerName="System.Data.SqlClient"/>
</connectionStrings>
```

Q7. What are various page events in ASP.NET?

Following are the page level events in ASP.NET:

- **PreInit:** This is the first event of a page used to set values such as a master page.
- **Init:** This event fires after each control have been initialized. You can use this event to change the initialized value of controls.
- **InitComplete:** This event is raised after all initializations of a page and its controls have been completed.
- **PreLoad:** This event fires before the view state has been loaded for a page and its controls and before page postback processing. This event is useful when you need to write code after the page is initialized.

- **Load(PageLoad):** The pageload event is generally used to check postback and then sets control properties appropriately. After this event, the load event of child control is called.
- **ControlEvents(Postback):** This event is called when a page or its controls causes postback such as ButtonClick event, SelectedIndexChanged event, CheckedChanged events, etc.
- **LoadComplete:** At this event, all controls are loaded even after additional processing can be done here.
- **PreRender:** This event is taking place before the view state is saved also it allows final changes to the page or its control.
- **SaveStateComplete:** Any changes to the page's controls at this point are ignored here. This event is useful if you need to write processing that requires the view state to be set.
- **Render:** In reality, it is not an event this is actually a method of a page object and its controls. At this point, controls are rendered in client-side [HTML](#), DHTML, and scripts for the browser.
- **Unload:** This event is used to clean up code you can use it to manually release resources.

Q8. What are server controls in ASP.NET?

The ASP.NET server controls are objects on the ASP.NET pages that run when the Web page is requested. Many server controls, such as buttons and text boxes, are similar to the HTML controls. In addition to the HTML controls, there are many controls, which include complex behavior, such as the controls used to connect to data sources and display data.

Q9. What is the difference between Hyperlink and LinkButton?

Hyperlink control does not have the Click and Command events; whereas the LinkButton control has these events, which can be handled in the code-behind file of the Web page.

Q10. What are navigation controls?

Navigation controls help us to navigate in a Web application easily. These controls store all the links in a hierarchical or drop-down structure; thereby facilitating easy navigation in a Web application.

Q11. How many navigation controls are in ASP.NET?

There are three navigation controls in ASP.NET:

- SiteMapPath
- Menu
- TreeView

Q12. What is the difference between client-side and server-side validations?

Client-side validations work at the client end with the help of scripting languages such as JavaScript or jQuery and VBScript. On the other hand, server-side validations work at the server end with the help of programming languages like C#, VB, F#, etc. Server validations work when you submit or send data to the server.

Q13. What is the difference between User Control and Custom Control?

The differences between User Control and Custom Control are given below-

User Control
Built into .ascx file.
Built into a redistributable assembly (i.e. a DLL).

Generally used for static content.

Generally used for dynamic content.

It is easy to create and use as it inherits from server controls.

Comparatively difficult because you have to develop from scratch.

Can be only used with current applications.

Can be used with any number of applications.

Language dependent.

Language independent, a control created in C# can be used in VB.NET.

Cannot be added to the Visual Studio toolbox.

Can be added to the Visual Studio toolbox.

Q14. What are Globalization and Localization?

Globalization is the process of designing and developing an application that functions for multiple cultures or locales. In other words, Globalization is the process of designing and developing an application in such a way that it can be used by users of multiple cultures. Globalization makes your application ready for international markets. This process involves:

- Identifying the culture and locale that must be supported by the application.
- Designing features to support those cultures and locales.
- Writing code that functions equally well with all the supported cultures and locales.

Localization is the process of customizing your application for a given culture or locale. In other words, Localization is the process of customizing your application in such a way that it behaves as per your current culture or locale. Typically, Localization translates your application UI into the current culture or locale.

Q15. What is a Resource File?

ASP.NET provides resource files to implement globalization and localization. The resource file is an XML file having extension resx. Each resource in the resx file is in the form of a key-value pair. For each culture that your application needs to support create a separate resource file. For example, WebResources.en.resx is a resource file for the English language, and WebResources.hi.resx is a resource file for the Hindi language.

Q16. What are Local Resources and Global Resources?

There are two types of resource files as given below :

Local Resources: Local resources are specific to a single web page and are used for providing versions of a web page in different languages. These are stored in the App_LocalResources folder.

Global Resources: Global resources are common for the whole web application and can be accessed by all the web pages. These are stored in the App_GlobalResources folder.

Q17. What are Neutral Culture, Culture, and Language?

A neutral culture is a culture that is associated with a language but not with a country or region. For example, "en" for English and in for Hindi. Culture consists of language and the country or region. It is denoted by culture code which contains two lowercase letters denoting the language and two uppercase letters denoting the country or region like as en-US for English in the US, en-GB for the UK, etc. A language is any spoken language like English (en), Hindi (hi), and German (de), etc.

Q18. What are the differences between GridView and DataGrid?

The [differences between GridView and DataGrid](#) are given below :

GridView

DataGrid

It was introduced with ASP.NET 2.0

It was introduced with ASP.NET 1.0

Built-in supports for Paging and Sorting.

For sorting, you need to handle the SortCommand event and rebind grid required and for paging, you need to handle the PageIndexChanged event and rebind grid required

Built-in supports for Update and Delete operations.

Need to write code for implementing Update and Delete operations.

Supports auto format or style features.

This feature is not supported.

Performance is slow as compared to DataGrid

Performance is fast as compared to GridView.

Q19. What are the differences between ListView and Repeater?

The differences between [ListView and Repeater](#) are given below :

ListView

Repeater

It was introduced with ASP.NET 3.5

It was introduced with ASP.NET 1.0

Built-in supports for paging and sorting.

Need to write custom code.

Built-in supports for Data grouping.

Need to write custom code.

Built-in supports for insert update and delete operation.

Need to write custom code.

Performance is slow as compared to Repeater

Performance is fast as compared to ListView

Q20. What are the different modes for the Session state in ASP.NET?

There are following the different modes of session states which can be defined in a Web.config file.

Off: Specify that Session state is not enabled.

InProc: This is the default mode of the session. In this mode session state is stored on the web server i.e. IIS where an application is running.

OutProc: The OutProc mode can be handled by using the following ways -

- **State Server:** In this mode, the session is stored in a separate process called ASP.NET state service. This ensures that the session state will be available in the web application and is restarted. This way also makes the session state available to multiple web servers in a web farm.
- **SQL Server:** In this mode, the session is stored in a [SQL Server](#) database. This also ensures that the session state will be available in the web application is restarted. This way also makes the session state available to multiple web servers in a web farm.

Note: Session_End event of Global.asax is fired only in InProc session mode. The object stored in the Session state must be serializable if the session mode is set to OutProc.

Q21. When to opt for classic ASP.NET over ASP.NET Core?

Although it's a better choice in nearly all the characteristics, you need not have to switch to ASP.NET Core if you intend to maintain a legacy ASP.NET application that is no longer actively developed. It is better to choose ASP.NET over ASP.NET Core if you:

- Require a stable environment to work within
- Don't require cross-platform support for your Web app
- Have closer release schedules
- Are previously continued working on an existing app and expanding its functionality
- Already owns a team with ASP.NET skills

Q22. What is the MVC pattern in ASP.NET?

The Model-View-Controller (MVC) represents an architectural pattern that splits an application into 3 key logical components. The names of these components are the model, the view, and the controller. Keep in mind that this pattern is not related to the layered pattern. Essentially, the MVC pattern runs on the software side.

In a particular application that implements the MVC pattern, every component comes with its role dedicated. To understand better, for example, model classes merely hold the data as well as the business logic. These classes don't deal with the HTTP requests but they view only display information. Moreover, the controllers deal with and respond to user input and determine which model to be passed to which view. Doing this makes an application simple to develop and maintain as it expands in complexity.

Q23. What is the NuGet package manager?

NuGet is a package manager dedicated to the .NET ecosystem. Essentially, Microsoft developed it to offer access to thousands of packages being written by .NET developers. Also, you can use it for sharing the code you already wrote with others.

A classic web application being developed through ASP.NET depends on several open-source NuGet packages to work. One of the famous NuGet package managers is Newtonsoft.Json. It is used for working with JSON data in .NET.

Q24. How is the Program class useful in ASP.NET?

Program.cs class signifies the entry point of our application. Any ASP.NET application begins identically as a console application, directly from a static void Main() function. It is this class that configures the web host which would serve the requests.

Q25. How is the Startup class useful in ASP.NET?

This class deals with two significant aspects of your application. They are service registration and middleware pipeline.

Two methods involved in startup class are ConfigureServices() and Configure().

Q26. What is a Kestrel?

Kestrel is an open-source type cross-platform web server uniquely designed for ASP.NET Core. It is incorporated and enabled by default in ASP.NET Core. Moreover, it can be utilized as a web server processing requests straight from a network.

The Kestrel is a cross-platform web server built for the ASP.NET Core based on "libuv" that is a cross-platform asynchronous I/O library which is a extremely fast library.

Q27. What is dependency injection?

Dependency injection represents a design pattern that assists to develop loosely coupled code. It is this pattern that is widely used in ASP.NET. The purpose of using Dependency injection is to provide the objects with what they need in the particular object's constructor instead of needing that object to construct them.

Dependency injection decreases and usually removes needless unnecessary dependencies between those objects that don't require knowing one another. Also, it assists in testing by hitting out the dependencies at runtime.

For implementing dependency injection, we can try multiple patterns such as constructor injection pattern or property injection patterns to make it work effectively.

Q28. What are the various validators in ASP.NET?

RequiredFieldValidator: This validator is used when you don't want the container to stay empty. It examines whether the control possesses any value or not.

RangeValidator: It finds out whether the value in validated control falls the specific range or not. **CompareValidator:** It tests if the value in controls matches certain specific values or not. **RegularExpressionValidator:** Tests if the particular value matches an explicit regular expression or not.

CustomValidator: Useful for defining User Defined validation.

Summary Validation: This shows a summary of all existing validation errors over an ASP.NET page.

Q29. Explain the state management in ASP.NET:

State management is a method to manage a state of a particular object on various requests. The HTTP protocol is the elementary protocol of the World Wide Web. It is a stateless protocol which means that each request is generated from a new user in accordance with a web server. In any web application, maintaining the state is vital.

There are majorly four types of state management strategies available which are given below:

- View State
- Control State
- Session State
- Application State

Above are the way to manage the state and all these are part of two categories which are given below.

2 types of state management systems existing in ASP.NET are:

1. Client-side state management
2. Server-side state management

Q30. What are the components of the web form in ASP.NET?

Here is the brief of components of web forms in ASP.NET:

Server controls: They are Hypertext Markup Language (HTML) elements that contain a `runat=server` attribute. These components provide automatic state management as well as server-side events and react to the user events by implementing an event handler on the server. **HTML controls:** They also respond to the related user events but the events processing takes place on the client machine.

Data controls: Data controls enable connection to the database, run commands, and access data from a database.

System components: They offer access to system-level events that take place on the server.

Q31. Explain the ASP.NET page lifecycle.

The ASP.NET page lifecycle is the sequence of events that happen in an ASP.NET web page, that is, from creation to its rendering. There are many stages involved in the process which are as follows:

1. **Initialization**- At this stage, the initialization of controls and the setting of their properties happen.
2. **Loading**- Here, control events are processed and they have a ViewState data which is restored.
3. **Postback event handling**- The Postback events are handled at this stage.
4. **Rendering**- At this stage the page is sent to the client after it is generated.
5. **Unloading**- Finally, the page is unloaded and it releases all its resources.

Q32. Differentiate between ASP.NET Web Forms and ASP.NET MVC.

ASP.NET Web Forms	ASP.NET MVC
It is a framework that is used to build web applications with the help of drag-and-drop, event-driven model.	It is a web application framework and it follows the Model-View-Controller pattern.
It uses traditional practices using controls with built-in mechanism for managing state.	It gives a more lightweight and control-centric approach.
It is widely used for projects where the focus is on maintaining state and handling complex UI.	It is widely used for projects which need more control over the structure of the application.

Q33. Explain the use of master pages in ASP.NET.

In ASP.NET, the master pages, basically, creates a common structure, layout, elements which are usually shared among multiple content pages and provides a template on the basis of that structure for consistent layout and design on multiple pages in a web application.

Q34. Explain the concept of caching in ASP.NET.

In ASP.NET, Caching is a concept which refers to storing the data that is frequently accessed or web page output in memory which helps in improving performance and reducing server load. There are several kinds of caching mechanisms such as:

1. Output Caching
2. Data Caching
3. Fragment Caching
4. SQL Cache Dependency
5. Custom Caching

Q35. What does ASP stands for?

ASP stands for Active Server Pages. It is a server side scripting language used to create dynamic and interactive web pages.

Q36. What is the difference between session state and view state?

Session State	View State
It is used for storing the data that is user-specific across multiple requests during user's session.	It is primarily used to keep up the control state information in between postbacks within one page.
It is stored on the server.	It is stored in the page in the form of a hidden field.
It maintains user authentication status, shopping cart contents and user preferences by storing any serializable data.	It maintains the user input and control state by storing values and properties of controls across postbacks.

Q37. What is authentication and authorization in ASP.NET?

Authentication- It is the process in which the user's identity is verified who are accessing an ASP.NET application. It is done by validating their credentials like usernames and passwords. There are several kinds of authentication mechanisms in ASP.NET such as forms authentication, authentication with external identity providers like OAuth.

Authorization- It is the process of determining those actions which are allowed for authenticated users to perform within an ASP.NET application. It basically controls the access to resources depending upon what roles and permissions are assigned to the users

Q38. What is the difference between Response.Redirect and Server.Transfer in ASP.NET?

Response.Redirect	Server.Transfer
With the help of Response.Redirect, the user is redirected to a new page.	With the help of Server.Transfer, the user is completely transferred to a new page on the same server.
It requires a round trip.	A single request to the server is only needed.
The URL changes which can be seen on the browser's address bar.	It shows no URL change.

Q39. Explain the concept of routing in ASP.NET MVC.

In ASP.NET MVC, Routing is a concept that refers to a mechanism which is used to define URL patterns for the application and then, those URL patterns are mapped onto specific controller actions. This routing system also does

interpretation of the incoming URL requests and tells which controller and action is required to handle them.

Q40. How do you deploy an ASP.NET application?

To deploy an ASP.NET application, you need to follow the following steps:

1. **Preparing the Application for Deployment-** Make sure that the application is successfully developed, tested and ready for deployment.
2. **Choosing a Hosting Environment-** Choose a hosting environment like Azure app Service or AWS Elastic Beanstalk.
3. **Setting up of Production Environment-** Make sure that all the hardware and software requirements are met by the server.
4. **Publishing the Application-** Publish the application using Visual Studio or .NET CLI.
5. **Deploying to the Server-** Transfer it to the production server with the help of FTP, SSH or Git.
6. **Configuring IIS (if applicable)-** Configure IIS settings if the application is hosted on an IIS server.
7. **Testing the Environment-** Test the deployed application thoroughly to make sure that all the functions are working correctly.
8. **Monitoring and Maintenance-** Monitor the performance of the application and regularly update and maintain the application with enhancements required.
9. **Backup and Disaster recovery-** Set up backup and disaster recovery techniques to protect the application and its data.

Q41. What are the different types of authentication modes in ASP.NET?

The different types of authentication modes in ASP.NET are as follows:

1. **Windows Authentication**
2. **Forms Authentication**
3. **Token-Based Authentication**
4. **Client Certificate Authentication**

Q42. What is the role of the @ symbol in ASP.NET syntax?

In ASP.NET syntax, The role of the @ symbol is to denote different types of constructs and features. There are some common uses of the @ symbol that include:

- Escape Character
- Directive Prefix
- Inline Code Blocks(Razor Syntax)
- Variable prefix(Razor Syntax)
- HTML Attribute Prefix(Razor Syntax)
- Email Address Obfuscation(Razor Syntax)

Q43. Explain the concept of partial views in ASP.NET MVC.

In ASP.NET, the concept of partial views can be defined as the components that are reusable and are used to represent a part of a view or user interface. In simple words, it breaks down the complex view into its smaller components which makes it easier to manage them as they can be modified independently and reused across other views within the application as well.

Q44. What is ActionResult in ASP.NET MVC?

ActionResult in ASP.NET is actually a base class for those classes who perform encapsulation of the result which is received by the execution of an action method. It has various derived classes representing different types of results. Some of them are:

1. ViewResult,
2. PartialViewResult,
3. RedirectResult,
4. JsonResult,
5. ContentResult,
6. FileResult, etc

Q45. How does ASP.Net Works?

The working of ASP.NET is as follows:

1. The client sends a request to the server when a user accesses an application on ASP.NET.
2. The process of routing maps the incoming request with the controller and action method it needs.
3. The request passed onto the controller where application logic is executed.
4. The action method within the controller handles the request.
5. If there is input data involved, the incoming request is bound to the parameters of the action method.
6. A view is selected and rendered where the data will be displayed.
7. A response is generated to be sent back to the client.
8. After receiving the response, the clients modifies it to display the requested web page by the user.
9. At this stage, the several mechanisms comes into action that help in managing state and session data.
10. Developers are able to hook into several lifecycle events with the help of middleware components and filters.

Q46. What is the extension for ASP.Net pages?

The extension for the ASP.NET pages differ with the technology you are using such as:

- For AS.NET web forms, it is **.aspx**.
- For ASP.NET MVC applications, the extension can be **.cshtml** or **.vbhtml**.
- For ASP.NET Razor pages, it will be **.cshtml**.

Q47. What are the different types of session state modes in ASP.NET?

In ASP.NET, session state are used to store and retrieve user specific data over many HTTP requests. The different types of session state modes are:

1. In-Process (InProc)
2. State Server
3. SQL Server
4. Custom

Q48. What is ViewState and why is it used in ASP.NET Web Forms?

ViewState typically refers to a mechanism which is used in ASP.NET web forms to maintain the state of server-side controls across postbacks. Its main uses are to maintain these states across postbacks, to simplify development and state management. It also helps in improving user experience by retaining user input and control state.

Q49. How do you implement URL routing in ASP.NET Web Forms?

To implement URL routing in ASP.NET Web Forms, you must the following steps:

1. Firstly, enable routing in your ASP.NET web forms application.
2. Define routes in the 'RouteConfig' class or other suitable location.
3. Handle Routed Requests using 'Page.RouteData' to access and 'Page.Routedata.Values' to retrieve route values.
4. Use routing in hyperlimks with the help of 'RouteUrl1' method.
5. Define a default route to handle requests which do not match your custom routes. This step is optional.

Q50.Explain the concept of bundling and minification in ASP.NET.

In ASP.NET, the concept of bundling and minification refers to the techniques that help in optimizing the performance of web application by reducing the number of HTTP requests and minimizing the size of resourceslike CSS, JavaScript.

FAQs

Q1. How do I prepare for a .NET interview?

You can prepare for a .NET interview by:

- Properly understanding the main concepts like object oriented programming, ASP.NET and ADO.NET.
- Practicing more and more coding and preparing for common interview questions.
- Be well prepared with your previous .NET projects.

Q2. What are some additional skills that would be helpful for an ASP.NET developer?

Some additional skills that would be helpful for an ASP.NET developer are knowledge of:

- front-end technologies such as HTML, CSS, JavaScript
- database management systems like SQL Server, MySQL
- version control systems like Git

Q3. What resources can help me prepare for ASP.NET interviews?

You can prepare for ASP.NET interview with the help of resources like official Microsoft documentation, tutorials that you can get online for free and practice exercises on coding platforms.

Q4. Is it important to practice coding exercises for a .NET interview?

Yes, you should definitely practice more and more coding exercises and get familiarized with algorithms and data structures.