ASP.NET Interview Questions

Mostafa Ahmed

Q-1: What are the difference between .NET framework and Core?

1. Cross-platform vs. Windows-only

   .NET Framework: Primarily designed for Windows applications.

   .NET Core and .NET 5+: Designed to be cross-platform, supporting Windows, macOS, and Linux.

2. Open Source

   .NET Framework: Closed-source and primarily controlled by Microsoft.

   .NET Core and .NET 5+: Open source and developed as a community project on GitHub.

3. Modularity

   .NET Framework: Monolithic framework with a large installation footprint.

   .NET Core and .NET 5+: More modular, allowing developers to include only the necessary components, resulting in a smaller installation footprint.

4. Deployment

.NET Framework: Requires the installation of the framework on the target machine.

.NET Core and .NET 5+: Supports self-contained deployment, meaning applications can include the necessary runtime components, making deployment more straightforward.

5. Performance

.NET Framework: Generally has good performance but may not be as optimized as .NET Core or later versions.

.NET Core and .NET 5+: Optimized for better performance, including improvements in speed and resource utilization.

6. API Compatibility

.NET Framework: Has a large set of APIs specific to Windows.

.NET Core and .NET 5+: Unified platform with a broader set of APIs, making it easier to write cross-platform applications.

7. Tooling

.NET Framework: Uses Visual Studio for development.

.NET Core and .NET 5+: Continues to support Visual Studio but also introduces cross-platform development with tools like Visual Studio Code.

8. Versioning

.NET Framework: Follows a versioning pattern tied to Windows releases.

.NET Core and .NET 5+: Introduces a new versioning scheme independent of Windows releases.

9. Future Development

.NET Framework: Limited future development; most new features are introduced in .NET Core and later versions.

.NET Core and .NET 5+: Represents the future of the .NET platform, with ongoing updates and improvements.

Q-2: What are the different components of .NET?

Following are the components of .NET

- Common Language run-time
- Application Domain
- Common Type System
- .NET Class Library
- .NET Framework
- Profiling

Q-3: What do you know about CTS?

CTS stands for Common Type System. It follows certain rules according to which a data type should be declared and used in the program code. CTS also describes the data types that are going to be used in the application. We can even make our own classes and functions following the rules in the CTS, it helps in calling the data type declared in one program language by other programming languages.

Q-4: What is CLR?

CLR stands for common language run-time, it is an important component of the .NET framework. We can use CLR as a building block of various applications and provides a secure execution environment for applications.

Whenever an application written in C# is compiled, the code is converted into an intermediate language. After this, the code is targeted to CLR which then performs several operations like memory management, security checks, loading assemblies, and thread management.

Q-5: Explain CLS.

Common language specification helps the developers to use the components that are inter-language compatible with certain rules that come with CLS. It then helps in reusing the code in other .NET compatible languages.

Q-6: What do you know about JIT?

JIT is a compiler which stands for Just In Time. It is used to convert the intermediate code into the native language. During the execution, the intermediate code is converted into the native language.

Q-7: What is the difference between managed and unmanaged code?

Managed code	Unmanaged code
Managed code is managed by **CLR**	Any code that is not managed by **CLR**
.NET framework is necessary to execute managed code	Independent of .NET framework
CLR manages memory management through garbage collection	Own runtime environment for compilation and execution

Q-8: What do you know about boxing and unboxing?

Boxing	Unboxing
Implicit	Explicit
Converting a value type to the type object	Extracting the value type from the object
eg : obj myObject = i;	eg : i = (int)myObject;

Q-9: How do you prevent a class from being inherited?

In C#, we can use the sealed keyword to prevent a class from being inherited.

Q-10: What are the different types of constructors in c#?

- Default Constructor
- Parameterized constructor
- Copy Constructor
- Static Constructor
- Private Constructor

Q-11: What are MDI and SDI?

- MDI(Multiple Document Interface): An MDI lets you open multiple windows, it will have one parent window and as many child windows. The components are shared from the parent window like menubar, toolbar, etc.
- SDI(Single Document Interface): It opens each document in a separate window. Each window has its own components like menubar, toolbar, etc. Therefore it is not constrained to the parent window.

Q-12: What is a garbage collector?

Garbage collector feature in .NET frees the unused code objects in the memory. The memory head is divided into 3 generations:

- Generation 0: It stores short-lived objects.
- Generation 1: This is for medium-lived objects.
- Generation 2: It stores long-lived objects.

Collection of garbage refers to the collection of objects stored in the generations.

Q-13: What is caching?

Caching simply means storing the data temporarily in the memory so that the data can be accessed from the memory instead of searching for it in the original location. It increases the efficiency of the application and also increases its speed.

Following are the types of caching:

- Page caching
- Data caching
- Fragment caching

Q-14: What are Features of MVC?

- Separation of concern Model – View – Controller
- URL Mapping (Restful)
- Using Razor pages

Q-15: Explain localization and globalization

Localization	Globalization
It means changing the already globalized application to cater to a specific language or culture.	Globalization is the process of developing applications to support multiple languages.
Microsoft.Extensions.Localization is used to localize the application content.	Existing applications can also be converted to support multiple languages.

Q-16: What is delegate in .NET?

A delegate in .NET is similar to a function pointer in other programming languages like C or C++. A delegate allows the user to encapsulate the reference of a method in a delegate object. A delegate object can then be passed in a program, which will call the referenced method. We can even use a delegate method to create a custom event in a class.

Q-17: Difference between interface and abstract class in .NET?

Interface	Abstract Class
An interface merely declares a contract or behavior that implementing classes should have.	An abstract class provides a partial implementation for a functionality that must be implemented by the inheriting entities.
An interface may declare only properties, methods and events with no access modifier.	An abstract class declares fields too.

Q-18: What is the difference between a stack and a heap?

Stack	Heap
Stored value type	Stored reference type
A stack is responsible for keeping track of each executing thread and its location.	The heap is responsible for keeping track of the more precise objects or data.

Q-19: What are the different validators in ASP.NET?

- **Client-side validation –** When the validation takes place on the client-side browser, it is called client-side validation. Usually, JavaScript is used for client-side validation.
- **Server-side validation –** When the validation takes place on the server then it is called server-side validation. Server-side validation is considered as a secure form of validation because even if the user bypasses the client-side validation we can still catch it in server-side validation.

Q-20: What are EXE and DLL?

EXE: It is an executable file that runs the application for which it is designed. When we build an application, an exe file is generated. Therefore the assemblies are loaded directly when we run an exe. But an exe file cannot be shared with other applications.

DLL: It stands for dynamic link library that consists of code that needs to be hidden. The code is encapsulated in this library, an application can have many DLLs and can also be shared with other applications.

Q-21: What is the difference between function and stored procedure?

Function	Stored Procedure
Must return a single value	Always used to perform a specific task
It can only have the input parameter	It can have both input and output parameters
Exception handling is not possible using a try-catch block	Exception handling can be done using a try-catch block
A stored procedure cannot be called from a function	A function can be called from a procedure

Q-22: List the events in the page life cycle.

- Page_PreInit
- Page_Init
- Page_InitComplete
- Page_PreLoad
- Page_Load
- Page_LoadComplete
- Page_PreRender
- Render

Q-23: Explain role-based security.

Role-based security is used to implement security measures based on the role assigned to the users in the organization. Then we can authorize users based on their roles in the organization. For example, windows have role-based access like user, administrators, and guests.

Q-24: What is cross-page posting?

Whenever we click on a submit button on a page, the data is stored on the same page. But if the data is stored on a different page, it is known as a cross-page posting.

Cross-page posting can be achieved by POSTBACKURL property which causes the postback.

FindControl method can be used to get the values that are posted on this page to which the page has been posted.

Q-25: Explain passport authentication.

During the passport authentication, it first checks the passport authentication cookie, if the cookie is not available the application redirects to the passport sign on page. Passport service then authenticates the details of the user on the sign on page and if they are valid, stores them on the client machine and then redirects the user to the requested page.

Q-26: List all the templates of the Repeater control.

- ItemTemplate
- AlternatingItemTemplate
- SeparatorTemplate
- HeaderTemplate
- FooterTemplate

Q-27: What is MIME?

MIME stands for multipurpose internet mail extensions, it is the extension of the e-mail protocol which lets users use the protocol to exchange files over the internet.

Servers insert the MIME header at the beginning of the web transmission. Then the clients use this header to select an appropriate 'player' for the type of data that the header indicates. Some of these players are built into the web browser.

Q-28: What are the different types of cookies in ASP.NET?

- **Session Cookie:** It resides on the client machine for a single session until the user logs out.
- **Persistent Cookie:** Resides on the user machine for a period specified for its expiry. It may be an hour, a month or never.

Q-29: What is the difference between ExecuteScalar and ExecuteNonQuery?

ExecuteScalar	ExecuteNonQuery
Returns the output value	Does not return any value
Used for fetching a single value	Used to execute insert and update statements
Does not return the number of affected rows	Returns the number of affected rows.

Q-30: What are the basic features of OOP?

- **Encapsulation:** Creation of self-contained modules that bind together the data and the functions that access that data.
- **Abstraction:** Handles complexity and allows the implementation of further complex logic without disclosing it to the user object.

- **Polymorphism:** Operation performed depends upon the context at runtime to facilitate easy integration.
- **Inheritance:** Creation of classes in a hierarchy to enable a class to inherit behavior from its parent class allowing reuse of code.

Q31: What are the different types of JIT Compilers?

Pre-JIT compiler: It compiles all the source code into the machine code in a single compilation cycle, i.e. at the application deployment time.

Normal JIT Compiler: The source code methods required at run-time are compiled into machine code and stored in the cache to be called later.

Econo JIT Compiler: The methods required only at run-time are compiled using this compiler and they are not stored for future use.

Q32: Discuss the difference between constants and read-only variables?

Constant fields are created using the const keyword and their value remains the same throughout the program.

The Read-only fields are created using a read-only keyword and their value can be changed. Const is a compile-time constant while Read-only is a runtime constant.

Q33: Explain the difference between value type and reference type?

Types in .NET Framework are either Value Type or Reference Type. A Value Type is stored in the stack and it holds the data within its own memory allocation. While a Reference Type is

stored in the heap and it contains a pointer to another memory location that holds the real data.

Q34: What is the difference between Stack and Queue?

The values in a stack are processed following the **LIFO** (Last-In, First-Out) principle, so all elements are inserted and deleted from the top end. But a queue lists items on a **FIFO** (First-In, First-Out) basis in terms of both insertion and deletion. The elements are inserted from the rear end in a queue and deleted from the front end.

Q35: What are the differences between systems. StringBuilder and system. string?

System. a .string is **immutable** and fixed-length, whereas StringBuilder is **mutable** and variable length. The size of the .string cannot be changed, but that of the .stringbuilder can be changed.

Q36: What are a base class and derived class?

The base class is a class whose members and functions can be inherited, and the derived class is the class that inherits those members and may also have additional properties.

Q37: What is the extension method for a class?

The extension method is used to add new methods in the existing class or the structure without modifying the source code of the original type. Special permission from the original type or re-compiling it isn’t required.

Q38: What is inheritance?

Inheritance is a method for creating hierarchies of objects wherein one class, called a subclass, is based on another class, called a base class.

Q39: What are implementation inheritance and interface inheritance?

Implementation inheritance is when a class inherits all members of the class from which it is derived. Interface inheritance is when the class inherits only signatures of the functions from another class.

Q40: What is a constructor in C#?

A constructor is a special method of the class that contains a collection of instructions and gets automatically invoked when an instance of the class is created.

Q41: Define Method Overriding?

Method Overriding is a process that allows using the same name, return type, argument, and invoking the same functions from another class (base class) in the derived class.

Q42: What is Shadowing?

Shadowing **makes the method of the parent class available to the child class without using the override keyword.** It is also known as Method Hiding.

Q43: What is the difference between shadowing and overriding?

Shadowing is used to provide a new implementation for the base class method and helps protect against subsequent base class

modification. Overriding allows you to rewrite a base class function with a different definition and achieve polymorphism.

Q44: Do we have multiple inheritances in .NET? Why?

No, .NET supports only single inheritance due to the diamond problem. Also, it would add complexity when used in different languages. However, multiple interfaces can solve the purpose.

Q45: What is the Diamond of Death?

It is an ambiguity that arises due to multiple inheritances in C#. Two classes B and C inherit from A, and D inherits from both B and C but doesn't override the method defined in A. The Diamond Problem arises when class B or C has overridden the method differently and D cannot decide to inherit from either B or C.

Q46: What is business logic?

It is the application processing layer that coordinates between the User Interface Layer and Data Access Layer.

Q47: Differentiate between user controls and custom controls?

User and Custom controls inherit from different levels in the inheritance tree. Custom control is designed for use by a single application while user control can be used by more than one application.

Q48: What is .Net Reflection?

Reflection objects are used for creating type instances and obtaining type information at runtime. The classes in the System.Reflection namespace gives access to the metadata of a running program.

Q49: What is a Hashtable?

The Hashtable class is a **collection that stores key-value pairs**. It organizes the pairs based on the hash code of each key and uses it to access elements in the collection.

Q50: Name design patterns in the .NET Framework?

There are 23 design patterns classified into 3 categories:

Creational Design Pattern

i. Factory Method

ii. Abstract Factory

iii. Builder

iv. Prototype

v. Singleton

Structural Design Patterns

i. Adapter

ii. Bridge

iii. Composite

iv. Decorator

v. Façade

vi. Flyweight

vii. Proxy

Behavioral Design Patterns

i. Chain of Responsibility

ii. Command

iii. Interpreter

iv. Iterator

v. Mediator

vi. Memento

vii. Observer

viii. State

ix. Strategy

x. Visitor

xi. Template Method

Q51: What are the design principles used in .NET?

Net uses the SOLID design principle which includes the following:

- Single responsibility principle (**SRP**)
- Open-Closed Principle (**OCP**)
- Liskov substitution principle (**LSP**)
- Interface segregation principle (**ISP**)
- Dependency inversion principle (**DIP**)

Q52: What is Marshaling?

Marshaling is the process of transforming types in the managed and unmanaged code.

Q53: What is the difference between Server. Transfer and Response. Redirect?

These are used to redirect a user from one web page to the other one. The Response. The redirect method requests a new URL and specifies the new URL. The Server. The transfer method terminates the execution of the current page and starts the execution of a new page.

Q54: What is the difference between trace class and debug class?

The call to **Debug** class is included in Debug mode only and it is used at the time of application development. While the call to **Trace** class will be included in Debug as well as Release mode also and it is used at the time of application deployment.

Q55: What is the application object?

The Application object is used to share information among all users of an application. You can tie a group of ASP files that work together to perform some purpose.

Q56: What is the session object?

A Session object stores information and variables about a user and retains it through the session.

Q57: What are the advantages of Web Services?

The advantages of Web Services are:

- It is **simple to build** and supported by a variety of platforms.
- It **can extend its interface and add new methods** without affecting the client's operations.
- **It is stateless and firewall-friendly.**

Q58: What is Serialization?

Serialization is the process of converting the state of an object into a form (a stream of bytes) to be persisted or transported. **Deserialization** converts a stream into an object and is the opposite of serialization. These processes allow data to be stored and transferred.

Q59: What is the difference between dataset.clone and dataset. copy?

Dataset.clone copies only the structure of the DataSet which includes all DataTable schemas, relations, and constraints but it does not copy any data. Dataset. copy is a deep copy of the DataSet that duplicates both its structure and data.

Q60: Differentiate between Task and Thread in .NET?

The **thread** represents an actual OS-level thread, with its own stack and kernel resources, and allows the highest degree of control. You can choose to Abort() or Suspend() or Resume() a thread, and set thread-level properties, like the stack size, apartment state, or culture. While a Task class from the Task Parallel Library is executed by a **TaskScheduler** to return a result and allows you to find out when it finishes.

Q61: What is Multithreading?

Multi-threading is a process that contains multiple threads each of which performs different activities within a single process. .NET supports multithreading in two ways:

1. Starting threads with **ThreadStart** delegates.
2. Using the **ThreadPool** class with asynchronous methods.