Contents

Interview Questions

DOT NET

.NET

Q-1 : What are the differences between .NET framework and core?

1.Cross-platform Vs windows-only

.Net framework : primarily assigned 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 : Open source and developed as a community project on GitHub

3- Modularity

.Net framework Monolithic framework with large installations footprint

.Net Core : 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 : 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 : 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 : Unified platform with broader set of APIs, making it easier to write cross-platform applications

7- Tooling

.NET framework : Uses VS for development

.NET Core : Continues to support VS but also introduces cross-platform development with tools like VS Code.

8-Versioning

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

.NET Core : introduces a new versioning scheme independent of windows release

9- Future development

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

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

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

- **Common Language Runtime (CLR):** The execution engine for .NET applications, managing code execution, memory, and system services.
- **Application Domain:** Isolated environments within a process for running applications, enhancing security and reliability.
- **Common Type System (CTS):** Defines how data types are declared and used, ensuring interoperability across .NET languages.

- **.NET Class Library:** A collection of reusable classes and APIs that provide a range of functionalities for applications.
- **.NET Framework:** The original implementation of .NET, providing a comprehensive environment for building and running applications on Windows.
- **Profiling:** Tools and APIs that allow developers to monitor application performance, memory usage, and other runtime behaviors.

Q-4 : 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 language.

Q-5 : What is CLR?

CLR stands for Common language run-time, 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

The **Common Language Specification (CLS)** is a set of rules and guidelines defined by Microsoft to ensure that different .NET languages (such as C#, VB.NET, F#) can interoperate with each other. it helps in reusing the code in other .NET compatible languages. This allows developers to create libraries in one .NET language that can be used in another.

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 machine code**

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
Boxing is the process of converting a value type to a reference type	Extracting the value type from the object
Obj myObect = i	I = (int) myObject

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

We can use the sealed keyword to prevent a class form 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..

- 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 objects in the memory.

The memory head is divided into 3 generations:

- Generation[0]: it stores short-lives 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 temporary in the memory so 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 increases its speed.

Types of cashing:

- Page caching
- Data Caching
- Fragment cashing

Q-14: what are features of MVC?

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

Q-15: Explain Localization and Globalization?

Localization	Globalization
It means changing to cater to a specific language or culture.	Globalization is the process of developing applications to support multiple languages.

Microsoft.Extentions.Localization is used to localize the application content	Existing applications can also be converted to support multiple language

Q-16: What is the delegate in .NET?

- A delegate in .NET is like 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 be passed in a program, which will call the referenced method.
- Can be used to create a custom event in a class

Q-17: Difference between interface and abstract class?

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

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 precise objects or data

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

Clint-side validation when the validation takes place on the clint-side.

Usually JavaScript is used for clint-side validations

Server-side Validation when the validation takes place on the server.

Server-side validation is considered as a secure form of validation because even if the user bypasses the clint-side validation we can still catch it in the server-side validation

Q-20: What are EXE and DLL?

- EXE: it is the executable file which run the application for which it is designed.
  When we build an application an EXE file is generated. , the assemblies are loaded directly when we run an exe.
  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 are the types of Validation controls?

1. Required field validator
2. Range validator
3. Compare validator
4. Regular expression validator
5. Custom validator
6. Validation summary

All happened in the client side

Q-22(web form) 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.

Q-24(web form) What is cross-page posting?

Whenever we click on submit 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 in page. Passport service then authenticated the details of the user on the sign in page and if they are valid, stores them on the client machine and then redirects the user to the requested page.

Q-26(web form) List all the templates of the Repeater control?

- Item Template
- AlternatingItem Template
- Separator Template
- Header Template
- Footer Template

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 clints 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:
  - **Definition:** Temporary cookies that are stored in the browser's memory and are deleted once the browser session ends (i.e., when the user closes the browser).
  - **Lifetime**: Exist only during the browser session.
- Persistent Cookie :
  - **Definition**: Cookies that are stored on the user's device (hard drive) and remain there even after the browser is closed, until they expire or are deleted.
  - **Lifetime**: Can have a specific expiration date set using the Expires or Max-Age attribute. If not deleted manually, they persist until the specified date.

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

In ADO.NET, both ExecuteScaler and ExecuteNonQuery are methods of the SqlCommand class used to execute SQL queries, but they differ in their purpose and the type of results they return.

ExecuteScaler	ExecuteNonQuery
Returns the output value	Does not return any value
Used for fetching a single value	Used to execute commands that modify data or database structure
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 bundling the data and the functions that access that data
  Key Concepts of Encapsulation:

1. ***Data Hiding**:* Restricting access to certain components of an object by making them private or protected and exposing only the necessary parts through public methods or properties.
2. ***Controlled Access**:* Providing methods or properties to read or modify the private fields. This ensures that data is accessed or changed in a controlled manner.
3. ***Improved Security and Maintainability**:* Encapsulation allows for safeguarding sensitive data and makes the system easier to maintain and extend.

- **Abstraction :** Handles complexity and allows the implementation of further complex logic without disclosing it to the user object.
  It allows developers to reduce complexity by breaking down systems into simpler, high-level components.
- **Polymorphism(many forms) :** Operation performed depends upon the context at runtime or compile time to facilitate easy integration.
  Polymorphism in .NET can be achieved in two main ways:
  Compile-time polymorphism (Static Binding) - Achieved through method overloading and operator overloading.
  Run-time polymorphism (Dynamic Binding) - Achieved through method overriding and implementing interfaces.
- **Inheritance :** Creation of classes in a hierarchy to enable a class to inherit behavior from its parent class allowing reuse of code

Q-31: What are the different types of JIT compilers?

Pre-JIT compiler : Compiles the entire application before execution using **Native Image Generator (Ngen.exe)**, improving startup time.

Normal JIT Compiler : Compiles methods **just before execution** and stores the compiled code in memory for reuse.

Econo JIT Compiler : Similar to Normal JIT but **does not store compiled code**, reducing memory usage but requiring recompilation on each execution.

Q-32: Discuss the difference between the constants and read-only variables?

Constant: fields are created using the const key word and their value remains the same through the program.

Constants are values that are **fixed at compile-time** and cannot change during program execution.

The Read-only: fields are created using readonly keyword and their value can be changed.

Read-only variables are assigned **once**, either at declaration or in a constructor, and cannot be modified afterward.

Const is a compile-time constant while read-only is a runtime constant.

Q-33: Explain the difference between value types and reference types?

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.

A Reference Type **store a reference (memory address)** to the actual data, which is stored in the **heap**.

Aspect	Value Types	Reference Types
Memory allocation	Stack(or within objects)	Heap
Storage	Store data directly	Stores a reference to data
Instances	Primitive types, structs	Classes, interfaces, strings, arrays
Assignment	Copies the data	Copies the reference
Size	Fixed size	Dynamic size
Copy behavior	Independent copies(pass by value)	Shared reference(pass by reference)

Q-34: Explain the difference between Stack and Queue?

Stack	Queue
(LIFO) Last-in-first-out	(FIFO) First-in-first-out
Elements are inserted and deleted from the top	Element are inserted from the rear end and deleted from the front end

Q-35: Explain the difference between StringBuilder and string?

String	StringBuilder
Immutable – fixed length	Mutable – variable length

Q-36: Explain the difference between Base class and derived class?

Base Class	Derived Class
The class whose members and functions can be inherited	The class that inherits those members and may also have additional properties

Q-37: 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 recompiling it isn't required.

Q-38: 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

Q-39: 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 a class inherits only the signatures of the functions from another class .

Q-40: 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 if the class is created.

Q-41: 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

- Requires the virtual keyword in the base class and the override keyword in the derived class (in C#).
- The method in the derived class **replaces** the base class method for all instances of the derived class.
- Works with **polymorphism**: If a base class reference points to a derived class object, the overridden method in the derived class is called.

Q-42: Define Method shadowing ?

Shadowing in .NET is a concept where a derived class defines a member (field, property, or method) with the same name as a member in the base class, effectively hiding the base class member. This is different from overriding because shadowing does not rely on polymorphism, and the base class member is still accessible if explicitly referenced.

- Uses the new keyword in the derived class (in C#).
- The base class member is **not overridden**; it is simply hidden in the derived class.
- Works with **compile-time binding**: The method called depends on the **reference type**, not the object type.

Q-43: 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

Q-44: Do we have multiple inheritance 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.

Q-45: What is the Diamond of Death?

It is an ambiguity that arises duo 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.

Q-46: What is business logic?

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

Q-47(web form) Differentiate between user controls and custom controls?

User and customer 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

Q-48: What is the .Net Reflection?

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

Q-49: 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

Q-50: Name design patterns in the .NET Framework?

There are 23 design patterns classified into 3 categories:

Creation Design Pattern

1. Factory method
2. Abstract factory
3. Builder
4. Prototype
5. Singleton

Structural Design Patterns

1. Adapter
2. Bridge
3. Composite
4. Decorator
5. Façade
6. Flyweight

Behavioral Design Patterns

1. Chain of responsibility
2. Command
3. Interpreter
4. Iterator
5. Mediator
6. Memento
7. Observer
8. State
9. Strategy
10. Visitor
11. Template Method

Q-51: What are the design principles used in .Net?

.Net uses the SOLID design principle which includes the following:

- Single responsibility principle(SRP)

  Every class should have only one reason to change, meaning it should have one, and only one, responsibility.

- Open-Closed principle(OCP)

Software entities (classes, modules, functions, etc.) should be open for extension but closed for modification.

- Liskov substitution principle(LSP)

  Objects of a superclass should be replaceable with objects of a subclass without affecting the correctness of the program.

- Interface segregation Principle(ISP)

  No client should be forced to depend on methods it does not use. It’s better to have multiple small, specific interfaces rather than a large, general-purpose interface.

- Dependency inversion Principle(DIP)

  High-level modules should not depend on low-level modules; both should depend on abstractions. Abstractions should not depend on details—details should depend on abstractions.

Q-52: What is Marshaling?

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

Q-53: What is the difference between the Server. Transfer and Response. Redirect?

These are used to navigate between pages

Feature	**Server.Transfer**
Mechanism	Transfers control to another page on the **server** without informing the
Client Involvement	Happens entirely on the server side; the client is unaware of the page change.
URL in Address Bar	The browser's address bar **does not change**; it still shows the original U

Feature	Server.Transfer
Performance	Faster because it avoids an additional request-response cycle.
HTTP Method	Retains the original HTTP method (e.g., POST).
State Preservation	Retains server-side state, such as form data and view state.
Cross-Application	Limited to the **same application**.
Error Handling	Errors in the transferred page may appear as errors in the original pag
Usage in SEO	Not SEO-friendly as the URL doesn't change.

Q-54: What is the difference between trace class and debug class?

The call to the Debug class is included in Debug mode only and it is used at the time of application development.

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

Q-55: What is 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.

Q-56: What is session object?

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

Q-57: 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 clint’s operations.
- It is stateless and firewall-friendly

Q-58: What is serialization and Deserialization?

Serialization is the process of converting a state of an object into new form (a stream of bytes) to be persisted or transported.

Deserialization converts a stream into an object and is the opposite of serialization.

Q-59: What is the difference between the dataset.clone and dataset.copy?

Dataset.clone copies only the structure of the Dataset which includes all Data Tables, schemas, relations, and constraints but doesn’t copy any data.

Dataset.copy is a deep copy of the Dataset the duplicates both its structure and data

Q-60: What is the difference between Task and Thread in .Net?

The Task and Thread classes in .NET are both used for managing asynchronous or concurrent operations, but they differ significantly in abstraction, functionality, and use cases.

Feature	Thread	Task
Abstraction Level	Low-level abstraction representing a single thread of execution.	High-level abstraction that represents a unit of work.
Execution Model	Directly maps to a system thread.	Runs on the ThreadPool by default (but can be configured otherwise).
Resource Management	Must be manually managed and explicitly created.	Automatically managed by the runtime. Efficiently uses threads from the ThreadPool.

Feature	Thread	Task
Creation	Explicitly created using new Thread().	Created using Task.Run() or Task.Factory.StartNew().
Return Value	Cannot directly return a value. Use Thread.Join() or shared variables to retrieve results.	Can return a value using Task<T> and supports async/await for easier result handling.
Exception Handling	Must handle exceptions manually.	Exceptions are captured and can be awaited or handled using Task.Wait() or Task.ContinueWith().
Synchronization	Requires explicit use of locks (e.g., Monitor, lock, Mutex).	Works seamlessly with high-level synchronization primitives or continuation tasks.
Concurrency Management	Only represents a thread of execution.	Provides additional concurrency features like continuations, chaining, and cancellation.
Performance	Can be expensive as it directly creates system threads.	More efficient as it leverages the ThreadPool for managing threads.
Use with async and await	Not compatible with async/await.	Fully supports async/await for asynchronous programming.
Cancellation Support	Requires custom implementation for cancellation.	Built-in support for cancellation using CancellationToken.

Q-61: What is multi-threading?

Multi-threading is a process that contains multiple threads each of which performs different activities within a single process.

.Net supports multi-threading in two ways:

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

Q-62: What is Difference between ASP and ASP .NET?

ASP	ASP .NET
Only two Languages available for scripting VBScript and JavaScript	Any fully compliant .NET language can now be used with ASP .NET, including C# and VBNET
ASP is interpreted	ASP .NET is compiled
Classic ASP uses a technology called ADO to connect and work with database	ASP.NET uses ADO.NET technology
ASP has Mixed GTML and coding logic	Html and coding part are separated by code behind files
ASP is partially object oriented	ASP.NET purely object oriented
ASP not in-built support for XML	ASP.NET full XML support for easy data exchange

Q-63 (web form): When all controls is fully loaded and when View state is called ?

All controls are fully loaded in Page_Load event

View state is called after the initialize event .. and before load event

Q-64: What are virtual functions?

Virtual functions are functions that are present in the parent class and are overridden by the subclass. These functions are used to achieve runtime polymorphism.

Q-65: What are pure virtual functions?

Pure virtual functions or abstract functions are functions that are only declared in the base class. This means that they do not contain any definition in the base class and need to be redefined in the subclass.

Q-65: What is a destructor?

A destructor is a method that is automatically invoked when an object is destroyed. The destructor also recovers the heap space that was allocated to the destroyed object. Closes the files and database connections of the object.

Q-66: What is an exception?

An exception is a kind of notification that interrupts the normal execution of a program. Exceptions provide a pattern to the error and transfer the error to the exception handler to resolve it. The state of the program is saved as soon as an exception is raised.

Q-67: What is Difference Error and Exception?

Feature	**Errors**	**Exceptions**
Severity	Critical system-level issues	Application-level issues
Recoverable	Usually not recoverable	Often recoverable
Causes	Hardware failures, JVM/CLR issues	Invalid input, resource issues, or logic errors
Handled by Code	No, typically unhandled	Yes, with try-catch blocks
Examples	OutOfMemoryError, StackOverflowError	FileNotFoundException, NullReferenceException

Q-68: What are the limitations of OOPs?

- Usually not suitable for small problems
- Requires intensive testing
- Takes more time to solve the problem
- Requires proper planning
- The programmer should think of solving a problem in terms of objects

Q-69: what is the visual studio technology that is used to test an API?

Technology	**Purpose**	**Complexity**	**Best For**
HTTP REPL (Read-Eval-Print Loop)	Manual API exploration	Easy	Quick API checks

MSTest, NUnit, xUnit	Unit and integration testing	Moderate	Automated testing
Swagger UI	Interactive API testing	Easy	Testing Swagger-based APIs
Web Performance Tests	Load and performance testing	Complex	Stress and performance tests
REST Client / Thunder	Manual API testing in VS Code	Easy	Lightweight testing

Q-70:Whay use [From body]?

1. **Binding Request Body Data**:
   - HTTP request bodies typically contain the data in a serialized format (like JSON or XML). [FromBody] tells the framework to deserialize the body content into the specified object.
2. **Flexible Data Passing**:
   - For large or complex objects, sending them in the request body is more practical than using query parameters or headers.
   - Example: Sending a JSON payload with nested objects.
3. **Clear Source Indication**:
   - By using [FromBody], you explicitly indicate that the data comes from the request body. This improves code clarity and avoids ambiguity when multiple data sources are involved (e.g., query, form, or header).

Q-71:What is the default compiler for the visual studio and what is using for

Visual Studio used Roslyn as its default compiler since 2015,

- **Compile Code:** Roslyn is the default C# and VB.NET compiler in Visual Studio, turning your source code into Intermediate Language (IL) code.
- **Provide Real-Time Code Analysis:** It powers features like IntelliSense, error detection, and warnings by continuously analyzing your syntax and semantics.
- **Enable Code Refactoring and Fixes:** The rich APIs exposed by Roslyn allow Visual Studio to offer automated refactoring suggestions and code fixes as you type.
- **Support Custom Extensions:** Developers can build custom analyzers and code generators (like source generators) that integrate directly into the IDE.

Q-72:what is the difference between Viewdata, ViewBag and TempData

In ASP.NET MVC, ViewData, ViewBag, and TempData are mechanisms for passing data between controllers and views, each with distinct characteristics and use cases.

Aspect	**ViewData**	**ViewBag**	**TempData**
Type	Dictionary (ViewDataDictionary)	Dynamic property	Dictionary (TempDataDictionary)
Syntax	ViewData["Key"] = value;	ViewBag.Key = value;	TempData["Key"] = value;
Typecasting	Required when retrieving data	Not required	Required when retrieving data
Null Checking	Necessary to avoid exceptions	Not necessary	Necessary to avoid exceptions
Scope	Current request	Current request	Current and subsequent requests

Aspect	ViewData	ViewBag	TempData
Persistence	Data is lost after redirection	Data is lost after redirection	Data persists after redirection until read in the subsequent request
Use Case	Pass data from controller to view within the same request	Pass data from controller to view within the same request	Pass data between controller actions, especially during redirects

Q-73:what is the difference between AddSingleton, AddScoped, and AddTransient

Aspect	AddSingleton	AddScoped	AddTransient
Description	A single instance is created once and shared throughout the entire application lifetime.	A new instance is created for each client request (or scope).	A new instance is created each time the service is requested.
Instance	Created once on first request and reused for all	New instance per HTTP request (or defined scope).	New instance every

Aspect	AddSingleton	AddScoped	AddTransient
Creation	subsequent requests.		time the service is injected or requested.
Typical Use Cases	Shared state services like caching, configuration, or logging.	Request-specific operations, e.g., Entity Framework’s DbContext, where consistency during a single request is important.	Lightweight, stateless services where no shared state is required.
Example Registration Code	services.AddSingleton<IService, ServiceImplementation>();	services.AddScoped<IService, ServiceImplementation>();	`services.AddTransient<IService, ServiceImplementation>();`

Q-74: What is Dependency Injection

Dependency Injection (DI) is a design pattern, and a technique used to achieve Inversion of Control (IoC) between classes and their dependencies. Instead of a class instantiating its dependencies internally, they are "injected" from the outside, usually by a framework or an external container.

Q-75: What is difference between IQueryable and IEnumerable?

Aspect	IEnumerable	IQueryable
Execution Location	Query execution occurs in memory. The entire collection is loaded, and operations are performed on that in-memory data.	Query execution is deferred and typically occurs on the data source. The query is translated (e.g., into SQL) and executed on the server.
Deferred Execution	Supports deferred execution, but once the data is loaded, further operations are done in memory.	Fully supports deferred execution, enabling the translation of the entire query into the native query language of the data source.
Performance	Suitable for in-memory collections and small data sets. May be less efficient for large or remote data sets due to data loading.	More efficient for remote data sources, as filtering, sorting, and other operations are performed by the database server before the data is sent over the network.

Query Flexibility	Limited to LINQ-to-Objects, meaning operations are performed on loaded objects.	Offers more advanced querying capabilities with LINQ providers (e.g., LINQ-to-SQL, Entity Framework), translating expression trees into optimized queries for the data source.

Q-76: What is CORS?

CORS stands for **Cross-Origin Resource Sharing**. It's a security mechanism implemented by browsers that enables controlled access to resources located outside the domain from which the request originated.

Q-77: What is Middleware?

Is a component that process requests and responses in the application pipeline

There are three methods of Middleware:

- App.Run() : This method terminates the pipeline. Once it's invoked, no further middleware components are executed.
- App.Use() : This method processes part of the request and then passes control to the next middleware component in the pipeline.
- App.Map() : This method branches the middleware pipeline based on a specified request path, so that only requests matching that path are handled by the branch.

Q-78: What is different between eager loading and lazy loading?

Aspect	**Eager Loading**	**Lazy Loading**
When Data is Loaded	At the time of the initial query.	When the related property is accessed for the first time.
Number of Database Calls	Typically, one complex query (joins) to retrieve everything.	Multiple queries; one for the main entity and additional ones for each related entity.

Definition	Defined in System.Collections (and System.Linq for LINQ queries). It represents a sequence of in-memory objects.	Def que

Aspect	**Eager Loading**	**Lazy Loading**
Use Case	When you know you'll need the related data immediately.	When you want to delay loading related data until it is needed.
Potential Performance Issues	Can lead to larger result sets and memory usage if too much data is loaded.	Can lead to the "N+1" query problem, causing many small queries.

.Linked Lists

Q-1: Mention what is Linked list?

A linked list is a data structure that can store a collection of items. In other words, linked lists can be utilized to store several objects of the same type. Each unit or element of the list is referred to as a node. Each node has its own data and the address of the next node. Its like a chain. Linked lists are used to create graphs or trees.

Q-2: What type of memory allocation is referred to for Linked lists?

Dynamic memory allocation is referred for Linked lists.

Q-3: Mention what is traversal in linked lists?

Term Traversal is used to refer to the operation of processing each element in the list.

Q-4: Describe what is the Node in the link list? And name the types of Linked Lists?

Together(data + link) is referred to as the node.

Types of Linked Lists are.

- Singly Linked List
- Doubly Linked List
- Multiply Linked List
- Circular Linked List

Q-5: What are the Singly Linked lists?

Singly Linked lists are a type of data structure. In a singly linked list, each node in the list stores the content of the node and a reference or pointer to the next node in the list. It doesn't store any reference or pointer to the pervious node.

Q-6: What is Difference between Linear Array and Linked List?

Arrays	Linked List
Deletion and insertions are difficult	Deletion and insertions can be done easy
For deletion and insertions, it needs movements	For deletion and insertions, it doesn't require movement of nodes
In it space is wasted	In it space in not wasted
It is expensive	It is not expensive
It cannot be reduced or extended according to requirements	It can be reduced or extended according to requirements
To avail each element the same amount of time is required	To avail each element a different amount of time is required
In consecutive memory locations elements are stored	Elements may or may not be stored in consecutive memory locations
We can reach there directly if we must go to a particular element	To reach a particular node, you need to go through all those nodes that come before that node

Q-7: Mention what are the application of Linked Lists?

Applications of Linked Lists are,

- Linked lists are used to implement queues, stacks, graphs
- In Linked Lists you don't need to know the size in advance.
- Linked lists let you insert elements at the beginning and end of the list

Q-8: Mention what are the difference between singly and doubly linked lists?

A doubly Linked List nodes contain three fields:

- An integer value
- Two links to other nodes, one to point to the pervious and other to the next node

Whereas a singly linked list contains points only to the next node.

Q-9: what is the biggest benefit of linked lists?

The biggest benefit of linked lists is that you do not specify a fixed size for your list. The more elements you add to the chain, the bigger the chain gets.

****

SQL Server

Q-1: What do you understand about Database?

Database is an organized collection of related data where the data is stored and organized to serve some specific purpose.

Q-2: Define DBMS?

DBMS stands for Database Management System. It is a collection of application programs which allow the user to organize, restore and retrieve information about data efficiently and as effectively as possible.

Q-2: Enlist the advantages of DBMS?

- Data is stored in a structured way and hence redundancy is controlled
- Validates the data entered and provides restrictions on unauthorized access to the database.
- Provides backup and recovery of the data when required
- It provides multiple user interfaces

Q-4: What do you understand about Data Redundancy?

Duplication of data in the data base is known as data redundancy. Duplicated data is present at multiple locations; hence it leads to wastage of the storage space and the integrity of the database is destroyed.

Q-5: What are the various types of relationships in Database?

One-to-one: A single record in one table is associated with **only one record** in another table, and vice versa.

One-to-many: A single record in one table can be associated with **one or more records** in another table, but a record in the second table is associated with only one record in the first table.

Many-to-many: A single record in one table can be associated with **many records** in another table, and vice versa.

Q-6: Explain Normalization and De-Normalization?

Normalization: is the process of removing redundant data from the database by splitting the table in well-defined manner to maintain data integrity. This process saves much of the storage space.

De-normalization: is the process of add redundant data on the table to speed up the complex queries and thus achieve better performance

Q-7: What are the different types of normalization?

- **First Normal Form (1nf):** A relation is said to be in 1NF only when all the columns of the table contain unique or atomic values.

- **Second Normal Form(2NF):** A relation is said to be in 2NF only if it is in 1NF and all the non-key attributes of the table are fully dependent on the primary key.
- **Third Normal Form(2NF):** A relation is said to be in 3NF only if it is in 2NF and there is **no transitive dependency** (i.e., non-key attributes should not depend on other non-key attributes).

Q-8:What is SQL?

Structured Query language. SQL is an ANSI(American National Standard institute) standard programming language that is designed specifically for storing and managing the data in the relational database management system(RDBMS) using all kinds of data operations.

Q-9:How many SQL statements are used? Define them.

SQL statements are basically divided into four categories, DDL, DML, TCL, and DCL.

Data Definition Language(DDL): commands are used to define the structure that holds the data. These commands are auto-committed i.e. changes done by the DDL commands on the data base are saved permanently.

(Create-Alter-Drop-Rename-Truncate-Comment)

Data Manipulation Language(DML): commands are used to manipulate the data of the database. These commands are not auto-committed and can be rolled back .

(Select-Insert-Update-Delete-Merge-Call-Explain Plan-Lock table)

Data Control Language(DCL): commands are used to control the visibility of the data in the database like revoke access permission for using data in the database.

(Grant-revoke)

Transaction Control Language(TCL): commands are used to manage and control transactions in a relational database. These commands ensure that the database remains consistent, even in situations like system crashes or multiple concurrent operations.

(Commit-Roll Back- Save point-Set Transaction)

Q-10:Enlist the advantages of SQL.

- Simple SQL queries can be used to retrieve large amount of data from the database very quickly and efficiency.
- SQL is easy to learn and almost every DBMS supports SQL.
- It is easier to manage the database using SQL as no large amount of coding is required.

Q-11:Explain the terms ‘Record’, ‘Field’, ‘Table’ in terms of database

Table: is the collection of records of specific types.

Record: is a collection of values or fields of a specific entity.

Field: refers to an area within a record that is reserved for a specific data.

Q-12: What are advantages and disadvantages of views in the database?

View is a virtual table that don't have its data on its own rather the data is defined from one or more underlying tables

Advantages of Views:

- As there is no physical location where the data in the view is stored. It generates output without wasting resources.
- Data access is restricted as it doesn't allow commands like insertion, updating, and deletion

Disadvantages of Views:

- The view becomes irrelevant if we drop a table related to that view
- Much memory space is occupied when the view is created for large tables

Q-13: What do you know about Functional dependency?

A relation is said to be in functional dependency when one attribute uniquely defines another attribute.

Q-14: What do you understand by the E-R model?

The **Entity-Relationship (E-R) Model** is a conceptual tool used in database design to visually represent the structure of a database. It helps in understanding the relationships between different entities (objects or concepts) and their attributes. The E-R model is widely used during the **database design phase** to create a blueprint of the database before implementation.

Q-15: Define Entity, Entity type, and Entity set

Entity is a real-world object or concept that can be uniquely identified and has a distinct existence.

Entity type refers to a category or class of entities that share common attributes.

Entity set in the database represents a collection of entities having a particular entity type

Q-16: Define the weak Entity set.

Weak Entity set is the one whose primary key comprises its partial key as well as the primary key of its parent entity. This is the case because the entity set may not have sufficient attributes to form a primary key

Q-17:Explain the terms 'Attribute' and 'Relation' .

Attribute is described as the properties or characteristics of an entity.

Relation is a two-dimensional table containing a number of rows and columns where every row represents a record in the relation. Here, a records are also known as 'Tuples' and columns are known as 'Attributes'.

Q-18:What is database transaction?

Sequence of operations performed which change the consistent state of the database to another is known as the database transaction. After the completion of the transaction, either the successful completion is reflected in the system, or the transaction fails, and no change is reflected

Q-19:What do you understand by join ?

A **join** is an operation in relational databases that combines rows from two or more tables based on a **related column** (usually a primary key or foreign key). It allows you to retrieve data from multiple tables as if they were a single table, enabling you to derive relationships between tables.

Q-20:What are the disadvantages of query?

- **Lack of Indexes**
- Stored procedures are excessively compiled.
- Difficulty in interfacing

Q-21:Define join tables?

a) **Inner Join** : an INNER JOIN combines rows from two tables based on a related column,

b) **Natural JOIN**: This is a type of inner JOIN that returns results from both the tables having the same name and data values in the columns of both the tables to be joined

c) **Cross JOIN:** Cross JOIN returns the result as all records where each row from the first table is combined with each row of the second table.

d) **Right JOIN**: Right JOIN is also known as Right Outer JOIN. This returns all the rows as a result from the right table even if the JOIN condition does not match any records in the left table.

e) **Left JOIN**: Left JOIN is also known as Left Outer JOIN. This returns all the rows as a result from the left table even if the JOIN condition does not match any records in the right table.

f) **Outer/Full JOIN:** Full JOIN return results in combining the result of both the left JOIN and right JOIN

Q-22: Explain the Data Dictionary?

Data dictionary is a set of information describing the content and structure of the tables and database objects. The job of the information stored in the data dictionary is to control, manipulate and access the relationship between database elements

Q-23: Explain the Primary Key and Composite Key?

Primary Key is that column of the table whose every row data is uniquely identified. Each row in the table must have primary key and no two rows can have the same primary key. Primary key value can never be null, nor can it be modified or updated.

Composite key is a form of the candidate key where a set of columns will uniquely identify every row in the table.

Q-24: What do you understand about Unique key?

A **Unique key** is the same as primary key whose every row data is uniquely identified with a difference of null value .i.e. Unique key allows one value as a NULL value

Q-25: Name the different data models that are available for database systems?

- Relational Model
- Network model
- Hierarchical model

Q-26: differentiate between 'Delete', 'Truncate', and 'Drop' commands?

After the execution of **DELETE** operation, COMMIT and ROLLBACK statements can be performed to retrieve the lost data

After the execution of **TRUNCATE** operation, COMMIT and ROLLBACK statements cannot be performed to retrieve the lost data

DROP command is used to drop the table or key like the primary/foreign key

Q-27: What is Difference between Primary key and foreign key?

Primary Key	**Foreign Key**
Uniquely identify each row	Refer to a primary key in another table
Doesn't allow null value	Allow null value
Cannot be duplicated	Can be duplicated
Table have single primary key	Table can have multiple foreign keys

Q-28: What is the difference between function and stored procedure?

Function	Stored Procedure
Must return a single value	Can return zero or n values
Have only input parameters for it	It can have both input and output parameters
Exception handling is not possible using try-catch block	Exception handling can be done using try-catch block
Can be called from a stored procedure.	Cannot be called from a function.
Allow only select statements	Allows DML in it (Data manipulation language) -select, insert, update, delete
Can be embedded in select statement	Cannot be embedded in select statement

Q-28: What is the difference between Union and Union All?

In SQL, UNION and UNION ALL are set operations used to combine the results of two or more SELECT queries. They have similarities but also key differences in how they handle duplicate rows.

Feature	UNION	UNION ALL
Duplicates	Removes duplicates (distinct rows only)	Includes duplicates
Performance	Slower due to sorting for duplicate removal	Faster (no sorting or duplicate check)
Use Case	When you need unique rows	When duplicates are acceptable

Q-29: What is the trigger?

A trigger a special type of stored procedure in a database that is automatically executed(or "triggered") in response to specific events on a table or view, such as INSERT, UPDATE, DELETE

Triggers are tied to specific events:

- AFTER Trigger: Executes after the event.
- INSTEAD OF Trigger: Replaces the event.
- BEFORE Trigger (some databases): Executes before the event.

Q-30: What is the two virtual tables in trigger?

Inserted: Contains the rows that are being inserted.

Deleted: Contains the rows that are being deleted.

Q-31: What is the different between virtual table and trigger?

Feature	Virtual Table	Trigger
Definition	Logical table derived from a query or operation.	Special procedure executed on table events.
Examples	Views, temporary tables, or INSERTED/DELETED in triggers.	Automatically logs, validates, or enforces rules.
Persistence	Exists temporarily or logically.	Tied to specific table operations.

Feature	Virtual Table	Trigger
Use Case	Used for querying or intermediate data.	Used for auditing, enforcing rules, or automation.

Q-32: What is the different between where and having?

Feature	Where	Having
Usage	Clause is used to filter rows from the result set based on conditions applied to individual rows before any aggregate functions are calculated.	Clause is used to filter groups of rows(aggregated data) based on conditions applied to the result of aggregate functions.
Applied To	Is applied to individual rows of the original data before any grouping or aggregation occurs	Is applied to groups of rows resulting from GROUP BY clause, after aggregation has been performed
Position in the query	Typically appears before the GROUP BY clause (if used) and is followed by GROUP BY clause	Appears after the GROUP BY clause and any aggregate functions in the SELECT statement.
Filtering Conditions	You can use conditions to filter individual rows based on column values	You can use conditions to filter aggregated data based on the result of aggregate functions.

Q-33: What is the different Clustered and Non-Clustered Index?

Aspect	Clustered Index	Non-Clustered Index
Data Storage	Data rows are stored in order of the index key.	A separate structure holds index keys and pointers to the data rows.
Physical Order	Determines the physical order of the table data.	Does not affect the physical order of the table data.
Number per Table	Only one per table, since the data can only be sorted in one way.	Multiple non-clustered indexes can be created on a single table.

Aspect	Clustered Index	Non-Clustered Index
Performance	Excellent for range queries and sorting because the data is stored sequentially.	Ideal for random lookups; may require extra pointer lookups to fetch data.
Impact on Inserts/Updates	Changing the clustered key can be expensive due to data reordering.	Typically, less costly; only the index structure is updated.

Q-33: What is indexed view?

An indexed view is a view that has a unique clustered index created on it, which causes the view's result set to be stored (or materialized) on disk. This materialization allows SQL Server to quickly retrieve

Precomputed results rather than recalculating them each time the view is required.

Key Points:

- **Materialization:** Unlike regular views, which are just stored queries and are re-executed on demand, an indexed view stores its result set on disk.
- **Performance Benefits:** Because the results are stored, queries that use the indexed view can be significantly faster, especially if the view involves complex aggregations or joins.
- **Maintenance Overhead:** The database engine must maintain the indexed view whenever the underlying data changes. This can add some overhead to data modification operations (INSERT, UPDATE, DELETE).
- **Restrictions:** To create an indexed view, several requirements must be met. For example, the view must be created with the SCHEMABINDING option, and the query defining the view must follow certain rules (e.g., deterministic functions, no outer joins, etc.).

****

Q-1:What are the types of UML?

1. Business Use Case diagram
2. Use Case diagram
3. Activity diagram
4. Sequence diagram
5. Collaboration diagram
6. Class diagram
7. Statechart diagram
8. Component diagram
9. Deployment diagram

Q-2:What is CSS?

Cascading Style Sheets is a style sheet language used to define the presentation of a web page. It provides complete control over the appearance of HTML elements

Q-3:What is Bootstrap?

Bootstrap is a CSS framework that provides prebuilt styles and components to quickly build responsive and modern designs.

Q-4:What is the difference between Bootstrap and CSS?

Feature	**CSS**	**Bootstrap**
Customization	Fully customizable, unique designs.	Limited by predefined styles unless overridden.
Learning Curve	Requires deeper CSS knowledge.	Easier to use with basic HTML knowledge.
Development Speed	Slower, styles created from scratch.	Faster, prebuilt components and utilities.
Responsive Design	Manual implementation required.	Built-in grid system and responsive utilities.
Design Consistency	Depends on the developer.	Consistent out-of-the-box styles.

Feature	**CSS**	**Bootstrap**
Code Complexity	Can be complex for large projects.	Simplifies repetitive tasks.
Use Case	Best for unique, tailored designs.	Best for quick prototyping and standard designs.

Q-5:What is **TCP/IP**

TCP/IP is the foundational suite of protocols that governs how the data is transmitted and routed across the internet and most networks.

- **TCP**: Ensures **reliable, ordered delivery** of data by breaking it into packets, checking for errors, and reassembling them at the destination.
- **IP**: Handles **addressing and routing** of packets to their destination using IP addresses (e.g., 192.168.1.1).

Q-6:What are **Firewalls**

A firewall is a network security device that monitors and controls incoming/outgoing traffic based on predefined rules.

- Acts as a barrier between trusted internal networks and untrusted external networks (e.g., the internet).
- Blocks malicious traffic (e.g., hacking attempts) and unauthorized access.

Q-7:What is VPN

A VPN creates a secure, encrypted tunnel between your device and a remote server, masking your IP address and protecting data from eavesdropping.

Q-8:What is Encryption

Encryption is the process of converting **plaintext** (readable data) into **ciphertext** (unreadable data) using an algorithm and a key. Only authorized parties with the correct key can decrypt and access the original data.

Q-8:What is Docker

Docker is a popular platform for developing, shipping, and running applications using containerization. It allows developers to package an application and its dependencies into a lightweight, portable container that can run consistently across different environments.

Q-9:What is DevExtreme?

DevExtreme is a comprehensive suite of high-performance HTML5 and JavaScript components created by DevExpress. It offers a wide range of UI controls—including data grids, charts, forms, and more—that help developers quickly build responsive and modern web applications.

Q-9:What is DevExpress?

DevExpress is a software company known for creating a wide range of developer tools and UI component libraries. These include frameworks for building web applications (like DevExtreme).