

C#,ASP.NET ,SQL , Azure , Microservices , Architecture , AI ML interview Questions

Compiled by

www.questpod.com

Contributed by India's Largest .NET whatsapp community

Do join and let's hunt jobs together.

<http://whatsapp.questpod.com>

Contents

What is this Document ?	2
Structure of Document	3
Some common problems	3
How to prepare ?	4
Most asked Questions Across Companies	4
OOP / SOLID	4
C#	5
JAVASCRIPT	7
ANGULAR	8
.NET CORE WEB API	9
ORM	10
Azure	11
SQL	12
Design Pattern and Architecture	13
SCENARIO BASED	14
API Testing	15
SCENARIO BASED	16
One of the Big multinational company	21
CODING QUESTIONS	22
Company specific Interview question and experiences.....	24



www.questpod.com

GlobalLogic Interview Questions :	24
LTI Mindtree.....	28
EPAM.....	31
Carelon.....	31
Accenture (1st Round)	32
Cognizant	33
Deloitte	34
GlobalData	35
Persistent - Architect interview questions asked to me	36
CapGemini.....	37
5 years CapGemini exp.....	37
Capgemini Azure Lead.....	38
Persistent Azure Dotnet Role.....	38
Mhpasis.....	39
1st round.....	39
2nd round.....	40
Virtusa.....	47
Virtusa Interview Process (Lead Software Engineer, YOE 10+)	47
Virtusa 11/June/2026 (Microservices and Docker Focussed).....	54
TCS Consultancy 20 +LPA Package (.NET Full Stack + AI)	55
TCS Experience 2 Special Notes from a developer.	58
Wolters interview	58
Infosys	62
WIPRO	62
Wipro 1st round held on 30th April 2026	62
Wipro 2nd round held on 8th May 2026	65
FN(LastLetterAlphabet) 4 years Questions	68

What is this Document ?

This document contains some of the most frequently asked interview questions for C#, ASP.NET, SQL Server, Angular, Azure, Microservices, React, Coding, and other technologies commonly expected from a Microsoft Full Stack Developer.



The questions in this document are not randomly collected from the internet. They are based on real interview experiences shared by members of the Questpond WhatsApp community. You can join our community using (<http://whatsapp.questpond.com>)

Over the years, hundreds of developers from different companies have shared the questions they faced during technical interviews. We have consolidated those experiences, identified the commonly repeated patterns, and compiled the most frequently asked questions into this document.

The objective of this document is to help C# and .NET developers understand current market expectations, prepare for interviews more effectively, and focus on topics that are repeatedly asked by recruiters and technical panels.

Whether you are preparing for interviews in C#, ASP.NET Core, SQL Server, Angular, Azure, Microservices, React, or Coding Rounds, this document serves as a practical reference based on real-world interview feedback from the Questpond community.

Structure of Document

This document is divided into two sections.

Section 1: Most Asked Questions

This section contains the most frequently asked interview questions across multiple companies. We have analyzed interview experiences shared by developers in the Questpond WhatsApp community and consolidated the common question patterns that repeatedly appear in technical interviews. If your goal is general interview preparation, this is the section you should focus on first.

Section 2: Company-Specific Questions

This section contains questions reported by candidates who attended interviews at specific companies. For example, if you have an interview scheduled with Infosys, TCS, Accenture, Cognizant, Capgemini, or any other company listed in this document, you can directly refer to that company's section and review the questions shared by candidates who have already appeared for interviews there.

In short, if you want to prepare broadly for the market, focus on the **Most Asked Questions** section. If you are targeting a specific company, review the corresponding **Company-Specific Questions** section for a more focused preparation.

Some common problems

[15 days' Notice period ISSUES.](#)

Interview Ghosting

How to prepare ?

If you are preparing for these questions by simply reading one-line answers, you may face difficulties during interviews. In most interviews, the interviewer will ask follow-up and cross-questions to test your understanding of the fundamentals.

The best way to build a strong foundation is by watching the interview question videos provided below. My recommendation is to first watch the videos and then go through the questions. Once you see the concepts explained with examples and demos, it becomes much easier to answer cross-questions confidently.

I want to emphasize this point: once you watch a **DEMO**, handling follow-up questions becomes significantly easier.

30 Important C# Interview Questions : <https://youtu.be/BKynEBPqiIM>

Software Architecture and Design Patterns Interview Questions : <https://youtu.be/AtTgcbLOqMM>

OOPS Interview Questions and Answers :- <https://youtu.be/u99wAoBjDvQ>

35 Important JavaScript Interview Questions : <https://youtu.be/Zb4dPi7CANU>

25 Important ASP.NET Interview Questions : <https://youtu.be/pXmMdmJUC0g>

25 Angular Interview Questions : <https://youtu.be/-jeoyDJDsSM>

20+ SQL Server Interview Questions : <https://youtu.be/SEdAF8mSKS4>

Most asked Questions Across Companies

OOP / SOLID

- 1) What is method signature?
- 2) What is the Order of constructor execution?
- 3) [What is destructor and how many can be there in a class\(only 1\)?](#)
- 4) [Why do you need a destructor when GC is there ?](#)
- 5) What happens when constructor is private?
- 6) If an Abstract class has concrete method so can its child class have same method name and can it modify it?
- 7) Abstract vs Static class
- 8) [Abstract vs Interface](#)
- 9) Why abstract class has constructors



www.questpod.com

- 10) When to use abstract and when to use interface give real time example in c#
- 11) Implicit and explicit interface
- 12) What is default interface and why do we need it ?
- 13) Polymorphism in c#
- 14) Method overriding with example
- 15) Difference b/w Virtual, Override and new keywords in C# ?
- 16) Method hiding with example
- 17) Is return type considered in method overloading?
- 18) Why and when to use method hiding? give real time scenario
- 19) How to extend a sealed class?
- 20) What is extension method? Write an example
- 21) What is solid principle and why to implement?
- 22) State all the principles with example
- 23) What is Liskovs Substitute Principle, explain with example
- 24) Difference in Open close and Liskovs principle
- 25) Diff between Liskov vs ISP ?
- 26) Calculate salary of employee based on grades (strategy pattern)

C#

THEORY QUESTIONS -

- 1) Value type vs Ref type.
- 2) Ref vs Out, give real time example and usage.
- 3) Can ref keyword be used in Async programming?
- 4) Const vs Readonly.
- 5) What is garbage collector?
- 6) How to remove unmanaged code?(How we remove garbage in unmanaged code?)
- 7) Throw vs Throw ex.
- 8) What are access modifiers?
- 9) What is Partial class give real time scenario.
- 10) What are generics? Have you implemented any in your project?
- 1) What are collections and its type.
- 2) What is Dictionary, List, HashTable.
- 3) What is Stack and Queue?



www.questpod.com

- 4) Difference between Generics and non generics?
- 5) What is Concurrent collection?
- 6) What are delegates and built in delegates.
- 7) What is LINQ?
- 8) Single vs SingleOrDefault - First vs FirstOrDefault - Select vs SelectMany
- 9) LINQ Queries practice , left join , right , inner . get 3rd highest
- 10) What is the effect of combining AsParallel with LINQ queries?
- 11) IEnumerable vs toList.
- 12) IQueryable vs toList.
- 13) IQueryable vs IEnumerable.
- 14) What is yeild keyword?
- 15) break vs continue vs return.
- 16) Reflections in c#.
- 17) What is TPL?
- 18) TPL vs Thread
- 19) Async vs Thread
- 20) How to start a Task?
- 21) Does Async create threads ?
- 22) what is async and await?
- 23) difference between Async/Await and TPL.
- 24) how async await work under the hood?
- 25) difference between ValueTask vs Task
- 26) what is ThreadPool Starvation?
- 27) ISyncStateMachine
- 28) What is structure of query pipeline in EF ?
- 29) what latest .net version you have used and whats is special feature you have seen?
- 30) When we create a new .NET 6 or .NET 7 Console Application, there is no explicit Main method in the Program.cs file. Why is that, and how does the application know where to start execution?"
- 31) What is a concurrency issue in an application, and how do you handle it? Is this related to performance?
- 32) How to do memory optimization ?
- 33) Suppose the database layer is already optimized, but performance issues still exist on the application side. What steps would you take to improve performance at the application level?
- 34) If an application has heavy database interactions, how can we optimize performance— whether it's a legacy application, a modern Web API, or a front-end app like React?
- 35) Imagine there's a production issue with very limited information available. How would you approach diagnosing and resolving the root cause of the issue?
- 36) Which code will perform better ?



www.questpod.com

```
string longString = "";
for (int i = 0; i < 1000; i++)
{
    longString += "one more ";
}

//-----

StringBuilder sBuilder = new StringBuilder();
for (int i = 0; i < 1000; i++)
{
    sBuilder.Append("Add This");
}
string str = sBuilder.ToString();
```

JAVASCRIPT

THEORY QUESTIONS

- 1) difference in reassign and redeclaration
- 2) var vs let vs const
- 3) types of object creation in js
- 4) what is hoisting concept in js?
- 5) what do you mean by temporal dead zone?
- 6) what is closure in js?
- 7) what is a prototype?
- 8) what is spread operator?
- 9) what is rest operator?
- 10) Difference between spread and rest operator
- 11) splice vs slice
- 12) What is arrow function?
- 13) what is anonymous function?
- 14) What is array and its types?
- 15) How to flatten multi-dimensional array?
- 16) = vs == vs === state the name of operators and explain the use



www.questpod.com

ANGULAR

THEORY QUESTIONS -

- 1) Why Angular?
- 2) Which is the latest version going on?
- 3) Difference JS and Angular?
- 4) How to install angular?
- 5) What is AOT?
- 6) What are decorators?
- 7) How do you create a component?
- 8) How does angular app starts?
- 9) Package.json vs package.lock.json file?
- 10) How to download or install other packages?
- 11) What is angular.json file?
- 12) What are modules?
- 13) What is ngModel?
- 14) What is routing and how it is implemented?
- 15) Lazy loading in routing?
- 16) What are directives and its type. Explain with real examples
- 17) How to create custom Directives?
- 18) Types of data binding in angular with examples
- 19) What is string interpolation?
- 20) Difference between string interpolation and Property binding
- 21) What is Http Interceptor? How to implement it?
- 22) What are services in Angular?
- 23) Dependency injection in Angular
- 24) Ways to pass data from one component to another?
- 25) List down all Life cycle hooks in sequence with examples
- 26) Constructor vs ngOnInit
- 27) ngOnChanges vs ngDoCheck
- 28) What is ViewChild?
- 29) How to create api calls in Angular?
- 30) What are HTTP filters in Angular?
- 31) How to add http headers or authorization in post call?
- 32) What are Pipes and its type?
- 33) How to create Pipes?
- 34) Promise vs Observables
- 35) Observables vs Subject
- 36) Different Types of Subject
- 37) Can we do asynchronous programing in Angular? How?
- 38) How to implement asynchronous programing in Angular?
- 39) Template driven forms and Reactive forms with their difference
- 40) What is auth guard in angular?
- 41) Types of auth guards in angular.



www.questpod.com

- 42) What is local storage?
- 43) Have you used RXJS? why it is used?
- 44) Give some operators of RXJS
- 45) What is map operator?
- 46) What is Template Reference Variable?
- 47) How to do testing in Angular?
- 48) State management in Angular
- 49) Angular new features

.NET CORE WEB API

- 1) Framework vs Core
- 2) Advantages/features of .net core
- 3) Disadvantages of web api
- 4) How does .net core starts?
- 5) What is middleware in dot net core
- 6) Minimal API
- 7) How did you create a customer middle ware ?
- 8) what is request delegate?
- 9) app.Use vs app.Run
- 10) What are filters in dot net core?
- 11) What is Authorization filter?
- 12) What is Exception filter?
- 13) What is Action filter?
- 14) Middleware vs filter?
- 15) Configure vs configure Services?
- 16) Dot net core 5.0 vs 6.0
- 17) Can we implement startup.cs file in 6.0?
- 18) What is dependency injection and types
- 19) What is service life cycle? (addTransient, addSingleton, addScoped), state thier difference
- 20) What is Routing and its types
- 21) What is an Attribute?
- 22) Serialization and Deserialization
- 23) How to implement Global exception?
- 24) What is session management?
- 25) What is caching and type of caching and how to implement it in dot net core?
- 26) What is MVC pattern?
- 27) What is MVVP pattern?
- 28) What is CORS policy?
- 29) Why CORS no issue in postman but in browser.



www.questpod.com

- 30) What is Restful API?
- 31) SOAP vs REST
- 32) What is JSON?
- 33) JSON vs XML
- 34) POST vs PUT (Do not use records , idempotent)
- 35) If We can modify resource via Post call then why we need PUT call?
- 36) Can GET call have request body?
- 37) Can we add resource using GET call?
- 38) What is Content negotiations in Web API ?
- 39) If REST API is slow what will you do , where will you hunt , how to optimize ?
- 40) What is Api Versioning and its type? How to implement it
- 41) Return types in .net core?
- 42) Can we have multiple return types in net core?
- 43) How to implement sorting and paging in .net core?
- 44) How to implement Authentication/Authorization in dot net core?
- 45) What is JWT? State its parts. How it is implemented?
- 46) JWT logout.
- 47) When should we use Openid , OAuth as compared to Custom JWT ?
- 48) How to save sensitive data in database?
- 49) What are some common encryption techniques?
- 50) What data encryption techniques you have used in project?
- 51) What is API gateway?
- 52) What is Rate limiter?
- 53) Suppose There is a requirement to upload a file with lakhs of records in database from mvc.
- 54) How to do it so transaction will be faster
- 55) Can you explain the MVC lifecycle and describe the flow of a request from the browser to the server and back?
- 56) What is cache management? How can it be implemented in a React front-end application?
How do we manage caching in an ASP.NET MVC web application? What about in an ASP.NET Core Web API?

ORM

THEORY QUESTIONS

- 1) What is ORM?
- 2) Have you used any ORM in your project ? Which one?
- 3) What is Dapper?
- 4) What is EF core?
- 5) EF core vs framework?
- 6) What are navigation property in EF core?
- 7) How did you implement EF core with relational database?
- 8) Which approach did you follow in your project?
- 9) Code first vs Database first?
- 10) Commands used for code first



www.questpod.com

- 11) What are fluent APIs?
- 12) How do you call a Stored Procedure from EF core?
- 13) What is lazy loading
- 14) Eager loading in EF core
- 15) 3.Tracking and Non Tracking in EF core
- 16) 4.Can we handle transactions in EF core?
- 17) 5.What is explicit transaction?

Azure

1. What is Azure App Service and how do you deploy a .NET application to it?
2. What are options to deploy a Web API on Azure? (App Service / AKS / Azure Container Service)
3. Why did you choose App Service to deploy WebAPI?
4. What is the difference between Azure App Service and Azure VM?
5. What is Azure Functions? How is it used with .NET?
6. How will you secure Azure Function ?
7. I want to create storage account and function app. I don't want to enable storage account but Azure Function should read data otherwise hacker can tamper storage account.
8. What are durable function apps ?
9. Check function app is healthy ?
10. 14. I want to receive an email if the function app is failing how do you configure this?
11. What is Azure Key Vault and why is it important?
12. What is Managed Identity in Azure?
13. what is key vault and how to connect kayvault
14. How do you implement CI/CD for a .NET project in Azure?
15. What is Azure Service Bus and how does it work with .NET?
16. What is azure Service bus and how we decide we required queue or topic inside.
17. what is event grid in azure, when we use and why?
18. what is Eventhub xplain with example?
19. What is Data mart ?
20. How do you implement Logging and Monitoring in Azure for .NET applications?



www.questpod.com

21. Azure APIM Questions
22. How do you secure an ASP.NET Core API in Azure?
23. Explain some AI services in Azure ?
24. Have you used AKS ?
25. How is app insights configured in your ecosystem? Can you see the app insight logs of all of your applications in single place or do I have to go to different URL/Tabs for each of the applications? How do you configure the same?
26. How do you configure the deployment parameters (number of pods/max pods/min pods) for your Web App/API to be deployed to App Service? Do I need to have separate configuration for each of the WebApp/API or can I configure all the apps from single configuration?

SQL

THEORY QUESTIONS

- 1) What is Normalization and state its form?
- 2) What is BOYCE CODD normal form?
- 3) Indexes in SQL
- 4) Cluster vs Non cluster
- 5) What is execution plan?
- 6) Primary vs Unique
- 7) Union vs Union ALL
- 8) ISNULL vs Coalesce
- 9) Truncate vs Delete
- 10) Table vs Views
- 11) Stored Procedure vs Function
- 12) What is CTE?
- 13) Temporary table vs CTE
- 14) Temporary table vs Temporary Table variable
- 15) Sub Queries vs CTE
- 16) SQL Injection and how will you prevent it?
- 17) Triggers in sql, purpose?
- 18) What is Rownumber?
- 19) Rank vs DenseRank
- 20) What is partition
- 21) Ties
- 22) What are Magic tables?
- 23) What is Deadlock?
- 24) What is WITH (NOLOCK)?
- 25) What is isolation level?
- 26) What is transaction and How to implement it?



www.questpod.com

- 27) Merged Statement
- 28) Send unlimited parameters through stored procedures
- 29) in sql How we optimize queries returning 100 millions of records?

Design Pattern and Architecture

- 1) Sometimes they tell to write code also , Singleton pattern code so try to attempt
- 2) What is Design Pattern and why do we need?
- 3) Types of Design Pattern - Creational, Structural, Behavioral
- 4) Which Design patterns you have used in in your project?
- 5) Which Design patterns you know?
- 6) What is Singleton DP? Type down the singleton pattern
- 7) Connection pooling many answer in singleton but that can be problematic. Especially from DB perspective.
- 8) Singleton pattern disadvantage
- 9) Breaking the Singleton using direct references. Iterator , IEnumerable
- 10) What is Strategy DP?
- 11) What is Adapter DP?
- 12) What is Abstract Factory DP?
- 13) What is Repository DP? list its advantages
- 14) What is CQRS DP?
- 15) What are 3 important elements
- 16) Why is mediator in CQRS
- 17) Why CQRS is important in MS.
- 18) What is event sourcing in CQRS
- 19) Is SQL server Event source ?
- 20) Have you worked on Clean Architecture?
 - 1) What are Microservices?
 - 2) Monolithic vs Microservices
 - 3) Layer vs Tiered
 - 4) Tier and N tier difference
 - 5) How does one microservice call other microservice internally (Queues , WebAPI)
 - 6) Scenario in we have microservice in 2 services, we want to use Service 1 data in service 2 how we can use in between? if we use web Api then how we authenticate in microservices?
 - 7) Strategic DDD vs Tactical DDD.
 - 8) Whats a domain and sub domain ?
 - 9) Types of sub domain
 - 10) Bounded context , context map and UL.
 - 11) What is a SAGA pattern?
 - 12) What is DDD?
 - 13) What are different classes , Entity , Value objects and Services
 - 14) Aggregate root.



www.questpod.com

- 15) CQRS Pattern
- 16) Repository
- 17) Queues.
- 18) Draw infrastructure diagram
- 19) How do you decide between a GraphQL and the Backend for Frontend (BFF) pattern?
- 20) How do you optimize costs in an Azure Cloud microservices architecture?
- 21) How do you implement the Rate Limiting pattern to isolate resource failures?
- 22) What is the role of Consumer-Driven Contracts (CDC) in preventing breaking changes?
- 23) How do you handle "Reference Data" (e.g., countries, currencies): replicate or query?
- 24) When is the Anti-Corruption Layer (ACL) pattern essential?
- 25) When do you need a Distributed Lock vs. Optimistic Concurrency?
- 26) What specific challenges does Event Sourcing introduce in a microservices environment?
- 27) How do you manage schema versioning for events in an Event-Driven Architecture?
- 28) When is CQRS (Command Query Responsibility Segregation) necessary versus over-engineering?
- 29) How do you securely inject secrets into services at runtime?
- 30) What strategies do you use for Distributed Rate Limiting?
- 31) How do you approach Multi-Tenancy: Silo model vs. Pool model?
- 32) What are the pros and cons of Client-side vs. Server-side Service Discovery?
- 33) How do you handle time synchronization and clock skew across distributed nodes?
- 34) How do you manage long-running processes without holding open HTTP connections?
- 35) What is your strategy for Canary Deployments in a dependent chain of services?
- 36) How do you design a robust strategy for distributed cache invalidation?
- 37) How do you implement Leader Election for singleton background tasks?
- 38) How does the CAP theorem practically influence your choice of data stores per service?
- 39) How does Conway's Law influence your decisions on team structure and service boundaries?
- 40) How do you manage a "Shared Kernel" without creating tight coupling?
- 41) How do you apply Chaos Engineering to validate system recovery?
- 42) How do you detect and solve the "N+1 Problem" in distributed service calls?

SCENARIO BASED

- 1) How will you increase performance in angular application?
- 2) If there is case of bulk upload photos/videos, what will be your approach?
- 3) If you find slowness in angular application how will you optimize it?
- 4) **If an abstract Employee class has abstract method GetManager() and one concrete method GetSalary().**

And Developer class and QA class inherits it.

Now your Client said there is a requirement to add a CEO class which does not have a manager but has salary.

How you will implement this scenario?

Option - You can extend it as it is or you can change the existing code. But which one would be optimal?

- 5) There are 2 different Interfaces A, B and both are having same method name.
Can a Class e.g. class C, implement both of them?
If yes, then how will it use the method from Class A.
- 6) There is a sealed class and I want to add new methods to it without changing the existing logic.
- 7) Can a class have multiple constructor with different access modifiers also with static constructor, if yes then in what order it will execute?
- 8) If you need to fetch data but parameter/query string is complex then you will use GET or POST call and why?
- 9) How to increase performance in dot net core?
- 10) How to do bulk insert, update in dot net core?
- 11) How to limit the API hits in dot net core?
- 12) How to handle huge payloads that expires in 30 secs?
- 13) In case of complex logic, will you use linq (dot net core) or SQL (database)?
- 14) How to optimize a Stored procedure?
- 15) If we told you to create a new project then by which approach you will choose Code first or Database first and why?
- 16) How to do bulk insert, update in EF core?

API Testing

- 1) Why we need testing?
- 2) What is Unit testing?
- 3) What is Integration testing?
- 4) Difference Unit and Integration?
- 5) Which unit testing framework you used in your project?
- 6) What is the code coverage percentage?
- 7) What is Moq. Have you used it in project?
- 8) How to handle exception in unit testing?

SCENARIO BASED

`int[] arr = {2,2,5,6,1,3,3,6,6,5,4,5,5}`

Sort the array in descending order of frequency of occurrence of each number. One with highest frequency should come first. And if frequency is same highest number should come first.

For example here 5 is occurring 4 times then 5 should come first

Then 6 is 3 times 6 will be next

2 and 3 both are 2 times. Same fre

quency so 3 should come first then 2

Similar is 4 and 1

So out put will be

`{5,5,5,5,6,6,6,3,3,2,2,4,1}`

Given a array `{3,7,2,4,5,1}`

Sort the array in ascending order

Output `{1,2,3,4,5,7}`

`{3,7,3,2,1,4,5,5,2,2}`

Eliminate duplicate character from array

Output `{3,7,2,1,4,5}`

Count the occurrence of each character in string

"hello world"



www.questpod.com

Output

h 1

e 1

l 3

o 2

w 1

r 1

d 1

Print the maximum occurring character in string

for example here it will be

l 3

Encode a String

Input AACBAEFEEH

Output should be

A2C1B1A1E1F1E2H1

Decode a string

Input A2C3B1A1

Output should be

AACCCBA

These encode and Decode string where asked in capgemini

Write a program to print following pattern

2



www.questpod.com

1 4

3 6 5

8 7 10 9

12 11 14 13 16

*

* *

* * *

* *

*

*

* *

* * *

* *

*

In Hpcl

These were sql queries asked

EmpName	MobileNo
---------	----------

A	8888888888
---	------------

B	7777777777
---	------------

C	8999999999
---	------------

D	8888888888
---	------------

E	7777777777
---	------------

F	7777777777
---	------------

G	9888888888
---	------------

H	8888888888
---	------------

Fetch the names of Employee who have same mobile Number



www.questpod.com

Here Output will be

A

B

C

D

E

F

H

As they have same mobile number

tblConsumer

Id Name

1 A

2 B

3 C

4 D

5 E

6 F

tblCustomer

OrderId ConsumerId DeliverDate

1 1 2023-04-01

2 1 2023-04-30

3 1 2023-06-30

4 1 2023-08-31

5 2 2023-01-04

6 2 2023-04-30

7 2 2023-08-31

8 2 2023-08-31



www.questpod.com

9	4	2023-08-01
10	4	2023-08-31
11	3	2023-06-30
12	3	2023-01-04
13	3	2023-01-04
14	3	2023-04-01
15	3	2023-04-01
16	6	2023-04-01
17	6	2023-04-30
18	6	2023-08-31
19	6	2023-08-31

Output

RefillCount ConsumerCount

0	2
2	2
3	1
5	1

Write a sql query to generate above output that includes count of consumers based on number of refills(orders) they have received between 1 april 2023 and 30 June 2023. Include those with zero refills.

Example here consumer ID 4 and 5 has no orders place between 1 april and 30 June 2023. So for 0 refills there is 2 consumer count.

Consumer ID 2 and 6 have two orders each between 1 april and 30 June 2023. One is on 1 april and other on 30 april for both. So for 2 refills consumer count is 2

Consumer ID 1 has 3 orders between 1 april and 30 June 2023. So for 3 refills there is only 1 consumer count.

There are no consumers who have place 4 orders. So there is no refill count 4 here

Consumer ID 3 has 5 orders between 1 april 2023 and 30 June 2023. So for 5 refills there is 1 consumer count.

This sql query was also asked in hpcl written test.

Hint they give as use temp table to store consumerId and refillcount(order count) per consumer between 1 april 2023 and 30 June 2023

Then join consumer table with temp table to get refillcount and consumercount

Suppose there is WebForm on that there is textbox to enter pan numver , in textbox while entering PAN Number.

Example XYGPN1122P. How will you do masking and umasking?

After XYG it should be *** and in Database it should be saved as XYGPN1122P but in text box

When user enters pan number it should be displayed as

XYG***

In one company this was program was to check if two strings are anagrams. Strings are anagram if one string is obtained by rearranging the letters of another

abac , baac are anagrams

rat , tar are anagrams

India , diani are anagrams

But abc and cbcaa are not anagrams

One of the Big multinational company

- 1) How do you convert your current project to Microservices ?
- 2) If you are asked to design a heavy traffic service with concurrent user requests and data should not be corrupt, how would you do that?
- 3) How do you write middleware components in .Net Core?
- 4) If you have to migrate a project from .Net 4.5 to .Net 8.0 then how would you do it?
- 5) Where do you write AddScoped and AddTransient
- 6) What design patterns have you used?
- 7) What is SSL certificate?
- 8) How is deployment carried out?



www.questpod.com

9) In entity framework how do you connect to database?

10) If you are architect, before project getting into development phase, what all questions you will ask customer?

11) What logging mechanism have you used?

12) Have you worked with queues or events before?

Azure interview questions

Azure Bus

CODING QUESTIONS

14 Points to remember:-

1. To not use in-built functions and should have good performance.
2. Write on Notepad → Browser → VSCode / Visual Studio.
3. Use Fiddlenet to practice.
4. Camel Case and Pascal Case
5. Basic NameSpaces
 - a. System.Text
 - b. System.Collections.Generic (Collections Plural , Singular).
6. AI Software used for interviewing
 - a. Berris MasterMind
 - b. VeriKlick
 - c. TestDom
 - d. Mercer Mettl test
 - e. Hacker Rank.
 - f. AI TalentQuest (WebMobi 360)
 - g. Zinterview.ai
7. Performance is always measured by BIG O notation.
8. Remember only 1 way which is the performance way and not using inbuilt function.
9. String is an Array in C# with 0 Index.
10. Strings are immutable.
11. ASCII understanding.
12. L shape LESS THAN. Keep one in Mind.
13. Writing the Forloop back and forth.
14. Understanding of collections very important.

Questions asked.



www.questpod.com

- 1) Reverse character in a string:-
 - a. Prepend Approach.
 - b. String Buffer.
 - c. **Swap:- 2 pointer technique.**
 - i. Many times interviewer say to Reverse Array Same technique.
 - d. Reverse words in a string (Home Work)
- 2) Find occurrence of each character :- Dictionary methodology
- 3) Find the first non-repeating character :- Dictionary Methodology.
- 4) Find 2 numbers in an array that add up to a target :- Complement
- 5) Is String is a Palindrome: - Two pointer technique
- 6) Anagram : - ASCII + Tug of War technique.
- 7) Max Sum of a Contiguous Subarray of Size k :- Fixed Window Sliding.
- 8) Sort array which has 3 unique values :- Dutch Nation Flag
 - a. Bubble Sort
 - b. Inserted Sort.
- 9) Valid Parentheses: - LIFO
- 10) Basic Algo : Prime , Factorial and Fibonnaci.
 - a. Should know how to check for remainders. %
 - b. Should know ODD and Even (Divide by 2).
 - c. Whole numbers :- All positive integers including zero, with no fractions or decimals.
 - d. Prime numbers , greater than 1 , Divisible by 1 and himself , 2,3,5,7,11
 - e. Fibonacci :- Next number by adding up the two numbers 0, 1, 1, 2, 3, 5, 8, 13
 - f. 4 factorial is $(4 \times 3 \times 2 \times 1 = 24)$
- 11) What is the next number approach ---- (Diff, Diff of Diff, Alternates , Cubes / Squares , Gap, ,VIP)
 - a. 4, 7, 10, 13, 16, ? :- Increasing
 - b. 1, 4, 9, 16, ? :- Big difference Multiplication.
 - c. 1,8,27,64,125 (think about 124) :- More Big difference.
 - d. 64, 58, 52, 46, ? :- Decreasing (-6)
 - e. 2, 4, 8, 10, 14, ? :- Alternating
 - f. 1,4,10,19,31 ? (Difference of Difference)
 - g. 1, 3, 7, 13, 21, 31, ? (GAPS)
 - h. 10, 21, 35, 56, 90, 145, 231, 360, ? :- Difference and GAPS
 - i. Special series like Prime , Fibonacci.
- 12) Move 0's to end of the array :- Single-Pass swap reader/placer
- 13) Left/Right rotate an array by K places :- Reversal technique(Mirro,Front(K),Back(K-N)/Back , Front and Mirror).
- 14) Longest Substring Without Repeating Characters : Sliding window technique.
- 15) Reverse a number without converting to string :- Modulus division technique
- 16) Remove Duplicates from a String :- Dictionary method
- 17) Basic Star pattern examples
- 18) Find Nth highest Salary
- 19) Swapping two number
- 20) Extract the surname



www.questpod.com

- 21) Remove All White Spaces from a String
- 22) Find the Longest Palindromic Substring
- 23) . int[] GivenArray = { 1, 2, 3, 4, 5, 6 }; //Print Output-> //1 2 3 4//5 6 1 2
- 24) Find max number in Array.
- 25) Find Duplicate records in table
- 26) Given an array of positive numbers, calculate the number of possible contiguous subarrays having product lesser than a given number K.
/// _____
///Example Input/Output:
///Input : arr[] = [1, 2, 3, 4]
/// K = 10///Output : 7
///The subarrays are {1}, {2}, { 3}, { 4}, { 1, 2}, { 1, 2, 3} and {2, 3}
27. given a target number of ingredients you must buy (for example, **3 items**: Item 0, Item 1, and Item 2).You are handed a coupon sheet with mixed-up items and prices.Go through the coupon sheet, find the **lowest price** for each required item, and add them together.
28. Min and Max numbers from Array.

Company specific Interview question and experiences.

GlobalLogic Interview Questions :

1. What are .Net components - CLR, CTS

CLR - converts IL to Machine Code, Garbage collection, Exception Handling, Thread Mgmt, Security

BCL(Base Class Library) - collection of reusable classes - File Handling, DB Access([ADO.Net](#)), Collections and LINQ, Networking, IO

CTS(Common Type System) - how datatypes are declared and used, cross language compatibility(int in c# to int32 in Int32)

CLS(Common Language Specifications) - Rules that all .Net languages must follow

JIT(Just in time compiler) -

Garbage Collector -

2. What is Type safety?

Type safety means:A variable can only hold values of its declared type, and operations are allowed only on compatible types. In .NET, it is enforced by the CTS and CLR, preventing invalid casts, memory corruption, and runtime errors

3. Features of .Net 6.0 used

C# 10 version - global using directive file scoped namespace,

Hot Reload in .Net 6 - modify source code while it is running

AZ functions supports running serverless functions in .Net 6.0

4. What is Garbage Collection and How does it work?

Garbage Collection is an **automatic memory management system** in .NET.

It, Allocates memory for objects, Tracks object usage, Frees memory when objects are no longer needed, All of this is handled by the CLR

5. Difference between abstract class and Interface?

An abstract class is used for sharing common implementation among related classes, while an interface defines a contract that multiple unrelated classes can implement. Interfaces support multiple inheritance, whereas abstract classes do not.

Abstract classes can have **constructors**, interfaces cannot

A class can implement **multiple interfaces** but inherit **only one abstract class**

6. Why NuGet packages used in .net application?

NuGet packages are **pre-built libraries** that you can add to your .NET projects.

Instead of writing everything from scratch, you use existing libraries

Saves time by using tested solutions

NuGet packages are used in .NET for managing reusable libraries and dependencies. They help speed up development, ensure consistency, and simplify version management by automatically resolving dependencies.

7. What is difference between .Net core and .Net framework?

.NET Framework is a Windows-only, legacy platform, while modern .NET (formerly .NET Core) is a cross-platform, high-performance, and open-source framework designed for cloud and modern application development."

8. Explain SOLID principles

9. What is Tuple in c#

A tuple in C# is a lightweight data structure used to store multiple values without defining a class. Modern C# uses ValueTuple with named fields, making it useful for returning multiple values from methods.

10. What are Generic Delegates and its types

Generic delegates are predefined delegates in C# that use generic type parameters, so you don't need to define custom delegate types.

1. Func<T> - Used when: Method returns a value
2. Action<T> - Used when: Method does NOT return anything (void)
3. Predicate<T> - Used when: Method returns bool (true/false)

All are widely used in **LINQ and lambda expressions**

11. Why we use Async and Await with Task-

They are used to **write asynchronous (non-blocking) code** in a clean and readable way.

Instead of blocking a thread while waiting (e.g., API call, DB call), your program can **do other work**.

It improves application responsiveness and scalability by freeing threads while waiting for I/O operations, especially in web and UI applications.

12. How you create custom middleware?

In ASP.NET Core, **middleware** is a component that handles HTTP requests and responses in a pipeline.

In ASP.NET Core, **middleware** is a component that handles HTTP requests and responses in a pipeline.

13. How microservices and different from Monolithic applications

Monolithic architecture is a single unified application, whereas microservices architecture consists of independently deployable services. Microservices provide better scalability and fault isolation but introduce complexity compared to monolithic systems.

14. What are the different microservice architecture design patterns

Microservices architectures rely on several **design patterns** to handle communication, scalability, resilience, and data management.

1. API Gateway Pattern - A single entry point for all clients which Handles: Authentication, Routing, Rate limiting
2. Service Discovery Pattern - Services dynamically find each other, Types: Client-side discovery, Server-side discovery
3. Circuit Breaker Pattern - Prevents cascading failures, Stops calls to failing service, Returns fallback response
4. Saga Pattern - Manages distributed transactions, Two types: Choreography (event-based), Choreography (event-based)
5. Database per Service Pattern - Each service has its own database



www.questpod.com

6. Each service has its own database - Event-Driven Architecture, RabbitMQ, Kafka
7. CQRS (Command Query Responsibility Segregation) - Separate: Read operations, Write operations
8. Strangler Fig Pattern - Gradually replace monolith with microservices
9. Retry Pattern - Automatically retry failed requests
10. Sidecar Pattern - Attach helper service to main service - Logging, Monitoring, Security

15. How do you implement fault tolerance in microservice based system

Fault tolerance means a system continues to function **even when some services fail**.

In microservices, failures are **inevitable** (network issues, service downtime), so we design systems to **handle and recover from them gracefully**.

Use - Circuit Breaker Pattern, Retry Pattern,

16. What is circuit breaker pattern

The **Circuit Breaker pattern** is used to **prevent repeated calls to a failing service**, avoiding cascading failures in a distributed system.

Think of it like an **electrical circuit breaker**: If something goes wrong → it “trips”, Stops further damage

The Circuit Breaker pattern prevents repeated calls to a failing service by opening the circuit after a failure threshold. It has three states—closed, open, and half-open—and helps avoid cascading failures in microservices.

17. What is CQRS

CQRS (Command Query Responsibility Segregation) is an architectural pattern that **separates read and write operations** into different models.

In simple terms: **Command** → modifies data (Create, Update, Delete)

Query → reads data

CQRS separates read and write operations into different models to improve performance, scalability, and flexibility. The command side handles updates, while the query side is optimized for reads, often using separate data stores.

18. What is service discovery in microservice architecture

CQRS separates read and write operations into different models to improve performance, scalability, and flexibility. The command side handles updates, while the query side is optimized for reads, often using separate data stores.



www.questpod.com

19. What are the different ways of communication between microservices

Microservices communicate in **two main ways**:

1. Synchronous Communication: Request → Response (real-time) - HTTP / REST APIs , gRPC (High Performance)
2. Asynchronous Communication - Message/Event-based (decoupled) - Message Queue , Event-Driven (Pub/Sub)

20. What is a role of API gateway in microservice

An API Gateway acts as a single entry point in microservices, handling routing, authentication, rate limiting, and response aggregation. It simplifies client interaction and centralizes cross-cutting concerns.

21. What is content negotiation

Content Negotiation is the process where the **server selects the best response format** (like JSON or XML) based on the **client's request**.

In ASP.NET Core, this is handled by formatters like JSON and XML automatically.

22. ACID properties in RDBMS

23. How we measure database servability

24. What is replication

25. How you identify slow running queries and optimize them

26. What is indexing and how does it improve performance

27. What is triggers and types of triggers

28. What is difference between commit, rollback and savepoints

29. What is difference between @@Identity and @ScopeIdentity

30. What are materialized view

31. What is difference between RANK and Dense Rank and Entile

32. What are different types of NOSQL

33. Why do we choose NOSQL over SQL



www.questpod.com

.NET / C#

- Stack vs Heap
- What is Garbage Collector? How does it work?
- If I don't want to create instance of a class, how to do it (without static/abstract)?
- How to create Dependency Injection?
- Where does injected dependency reside?
- Constructor chaining
- Partial class – what & use
- Can partial class exist in different projects?
- Types of NullReferenceException
- Task vs Thread / Multithreading
- IEnumerable vs IQueryable (inheritance difference)
- Hashtable vs Dictionary / HashSet vs Dictionary
- Ref vs Out
- IDisposable & where to use
- .NET Framework vs .NET Core (developer perspective)
- Async & Await

Entity Framework / ADO.NET

- What is ADO.NET?
- Can we use Stored Procedure with ADO.NET?
- Difference between SQL query & Stored Procedure in ADO.NET
- EF Migrations commands
- AsNoTracking
- EF Code First vs Database First
- Eager loading

SQL

- Views vs Indexes
- Indexes (types)
- Triggers



www.questpod.com

- SP vs Function
- What is SP?
- Where vs Having
- CTE
- Functions – can functions have multiple return types?
- Types of joins
- Query: If commission is null return 0
- Get 5th highest salary
- Department with 0 employees
- How to identify missing index?
- SQL optimization techniques
- Keyword to get last inserted record

ASP.NET Core / MVC / API

- Authentication & Authorization
- JWT (working, expiry, content)
- Logging
- Exception handling
- Middlewares
- REST principles
- Routing
- Model validation
- HTTP methods
- 200 vs 201 status code
- Can we use [Post] on Get action?
- Clean code principles (DRY)
- Return error message to client
- Multiple views / Same view for different controllers
- DropDown performance handling

Azure



www.questpod.com

- Azure Functions (data extraction from request)
- Azure Key Vault integration with .NET API
- Handling API issue from Cloud (analysis steps)

Architecture / Design

- Design patterns (Factory)
 - SOLID principles
 - Open/Closed principle violation example
 - Caching (why, memory vs Redis)
 - Microservices architecture
-

EPAM

Coding

- Count repeated characters → Alphabet + Number format
- LINQ group & sum
- Letter count in string
- LINQ join 2 tables
- Parallel programming

Concepts

- DI lifetimes (Scoped explanation specifically)
 - Authentication & Authorization
 - Logging
 - Middlewares
 - REST
 - Routing
 - Async / Await
-

Carelon

.NET / Project

- Current .NET version used



www.questpod.com

- Latest .NET features
- Performance difference (.NET vs Framework)
- Ref vs Value types
- JWT working & expiry

MVC / API

- Multiple views same data
- AJAX / jQuery
- Error handling
- Display dropdown data efficiently

Coding

- Letter count program
 - LINQ join
-

Accenture (1st Round)

OOPS

- OOPS concepts
- Inheritance & Multiple inheritance
- Types of Polymorphism
- Calling same method from multiple interfaces

C#

- Swap without 3rd variable
- Delegates
- Async programming
- Multithreading

MVC / API

- DTO
- ViewBag
- MVC lifecycle



www.questpod.com

- Action return types
- Security in API

SQL

- Design DB
- Constraints
- SP
- Index & Clustered index

Other

- Microservices
- Hosting
- Attributes

Cognizant

Project & Process

- Roles & responsibilities
- Task planning
- Agile
- Estimation
- Client interaction
- Helping juniors

.NET / API

- DI & lifetimes
- Async/Await
- Authentication & Authorization
- API versioning
- Rate limiting
- Validations before controller
- Exception handling



www.questpod.com

SQL / EF

- Prevent SQL injection
- Stored Procedures
- CTE
- Functions
- Types of joins
- EF parent-child loading
- Handling scoped DB connection expiry

Azure / DevOps

- Azure services used
 - CI/CD
 - OAuth
-

Deloitte

Concepts

- SOLID principles
- Dependency Injection
- JWT mechanism
- Authentication & Authorization
- Async programming
- Exception handling
- CQRS

Coding

- Singleton implementation
- Delegates (Func vs Action)
- Anagram program
- 3rd largest number

Testing



www.questpod.com

- Unit testing framework
 - NUnit
 - Mocking vs Stubbing
-

GlobalData

Project

- Module worked on
- ORM used
- Querying DB

ASP.NET

- Constructor in API
- app.Use vs app.Run
- Fix slow MVC page
- Validation before controller

SQL

- 5th highest salary
- Department with 0 employees
- Joins
- Optimization

Caching

- Types of caching
 - Redis storage location
 - In-memory cache configuration
-

Verisk Interview

Project / Basics

- Tech stack
- SQL version
- SQL connection with application

SQL

- Optimize query
- Group patients by doctor (today)

Coding

- 3rd largest element using 1 loop

API

- REST API skeleton
- What is REST API
- Parallel request execution

Persistent - Architect interview questions asked to me

- 1) If a client comes to you for their application modernization, what would be your approach?
- 2) Explain flow of authentication in your Web application / Web API
- 3) Explain flow of authentication in general (JWT / OAuth2)
- 4) What are options to deploy a Web API on Azure? (App Service / AKS / Azure Container Service)
- 5) Why did you choose App Service to deploy WebAPI?
- 6) How do you do logging/tracing in your Web Application / WebAPI? How did you design observability/traceability in your Application ecosystem?
- 7) How do you access data / database in your Web API? Dapper / EF Core / ADO.Net? Which one would you choose under what circumstances?
- 8) How is app insights configured in your ecosystem? Can you see the app insight logs of all of your applications in single place or do I have to go to different URL/Tabs for each of the applications? How do you configure the same?
- 9) How do you configure the deployment parameters (number of pods/max pods/min pods) for your Web App/API to be deployed to App Service? Do I need to have separate configuration for each of the WebApp/API or can I configure all the apps from single configuration?
- 10) Your application is built in a legacy version of Angular. How would you estimate/approach for upgrading it to the latest framework version?
- 11) How do you conduct TDD of DB objects (Stored procedures)?
- 12) Design patterns used in Event Driven Architecture



www.questpod.com

13) What's use of Key Vault in your project?

CapGemini

5 years CapGemini exp

Was asked Mostly Fundamentals was asked

1. Explain D from SOLID with written example
2. Any design pattern which you have implemented. I explained Singleton.
3. How to pass data between controllers
4. Automapper
5. Bundles.config
6. How to implement caching
7. Difference between list and dictionary
8. How to implement security JWT token
10. Return type in API
11. How to handle exception on application level
12. SQL query for 2nd highest salary
13. SQL query for deletion of one of the duplicate records
14. How can we optimize query performance
15. How to create directory in C#
16. Angular interpolation binding
17. Pipes, impure and pure pipes
18. Give duplicates from a list using LINQ
19. SQL Joins
20. select * from Employees where 1=0. What is the output?
21. Abstract class vs Interface

Capgemini Azure Lead

How do you handle State Management in Azure Functions?

How do you deploy APIM in Multi-Region with Budget Constraints?

How do you Handle Deduplication in Service Bus?

How do you Ensure Message Reliability in Service Bus?

My backend service is deployed inside a VNet, private endpoint but my APIs in APIM are public, how do I ensure connectivity between them? What should I do?

How do I ensure that while using App Service Plan for Azure Functions, I always get pre-warmed instances? Is that even possible?

How do we use partition keys in Cosmos DB?

What is meant by replay tools?

Persistent Azure Dotnet Role

☐ Middleware

☐ Exception handling

☐ Performance optimization technique in .NET API

☐ Explain your project structure

☐ How to make your application more scalable and more available

☐ Did you take downtime while deployment? How much? If yes then why?

☐ How to achieve 0 downtime (not even 0.00001%)

☐ What is Azure Service Bus

☐ Azure Service Bus vs Event Hub

☐ What is Azure Function App

☐ How do you decide to use Function App

☐ How to implement Azure Entra ID, what configuration do you have to do?

☐ How authentication works with Entra ID?

Mhpasis

1st round

Intro, roles & responsibilities

Project & tech stack

Polymorphism

How to call base class constructor

DI & life cycles of DI

Throw vs throw ex

Garbage collector

Cache & types of cache

Middle ware

How to create a custom middleware

App.use() vs App.Run()

Query parameters

How do you extract data from http request

How do you do Model validation (DataAnnotation , Regex)

What is Model binding

Asynchronous programming

Task vs thread

Multi threading

What is the use of IDisposable

Microservices architecture

Design patterns used in project

CQRS

API versioning

If a service is failing continuously, how will you handle this situation

How do you make communication between services

How do you call one service from another

Code :

1. given sentence and asked to reverse letters in a word without changing the order of words in the sentence
2. Count the occurrence of letters in a string

SQL

Indexes & types

How many cluster indexes will be there for table

CTE vs temp table

Window functions

Rank, dense rank

2nd round

Intro, roles responsibilities

Tell about current project, tech stack, what modules you worked on

Open note pad & Write a controller & show how you implemented repository pattern using DI and entity framework

What DI life cycles used in your project

How do you write unit testing for this ?

What frameworks do you use for unit testing

Middle wares & custom middle wares, how you used middle wares in your project?



Exception handling and how you return user friendly messages to front end

How you implemented Security in your project

Tell how jwt tokens work

Have you used cors and how does it work ?

What kind of requests it'll block?

What deployments used in your project

EPAM Interview

Epam interview questions

Demonstrates good and well-rounded understanding of .NET Core and .NET Framework differences, including runtime, hosting model, and performance implications.

Proficient in Dependency Injection concepts and practical implementation patterns.

Experienced with application logging using the ILogger interface and follows recommended logging practices.

ASP.NET Core

Shows good understanding of ASP.NET middleware: especially pipeline behavior.

One gap noted: did not correctly articulate the short-circuiting behavior in middleware and instead referenced the circuit breaker pattern.

Good understanding of JWT authentication, token validation, and middleware configuration.

Proficient in model binding and model validation, with clear real-world application examples.

C#

Good understanding of the IDisposable pattern and resource management.

proficiency with async/await and comprehension of Task vs Thread concepts.



Experienced in using Parallel.ForEach and demonstrates very good foundational knowledge of multithreading scenarios.

Clear understanding of ref vs out semantics and use cases.

Competent in exception handling, including structured use of try-catch-finally.

Successfully implemented a LINQ-based grouping and projection solution to aggregate sales by ProductId, showcasing confidence with collections and object transformations.

Attempted a balanced brackets function with good structure and correct syntax, but was unable to achieve the expected output

LINQ

Demonstrates good proficiency with LINQ.

Skilled in filtering, grouping, aggregation, and projection.

Shows good ability to apply LINQ in real-world business scenarios.

Software Design

Good understanding of DRY, clean coding guidelines, and maintainable design practices.

Capable of creating technical documentation, including coding standards and diagrams.

Design Patterns

Knowledgeable about the Decorator pattern with practical examples.

Understands Singleton and Factory patterns at a conceptual level, though examples were more high-level than deeply implementation-focused.

Entity Framework

Good working knowledge of Code First approach and Code First Migrations.

Understands EF's tracking vs no-tracking queries and how they impact performance and memory usage.

Comfortable discussing EF architecture and real-time usage scenarios.

SQL

Strong understanding of WHERE vs HAVING and SQL query execution order.



Knows how SQL handles NULL values, including comparisons and function behavior.

Demonstrated logical reasoning by structuring an UPDATE query using a CASE expression, even without recalling exact syntax showing strong conceptual SQL thinking.

Unit & Integration Testing

Demonstrates clear understanding of the AAA (Arrange, Act, Assert) pattern.

Good knowledge of mocking frameworks and how to isolate dependencies.

Can clearly differentiate between unit tests and integration tests, with examples of when each is appropriate.

Microsoft Azure

Shows good knowledge of Azure Functions, serverless execution, triggers, and bindings.

Familiar with Azure Key Vault, including secrets management.

Possesses working knowledge of Application Insights for observability.

Understands fundamentals of Azure Application Gateway

Function apps vs App Services

Authentication Approach - AD

What is JWT

OAuth 2.0 vs Open ID

While doing the API Authentication, Client Credential Flow

Type of Identities, System based Identify, Managed Identify and User Identify

I have a .NET Web API I want to consume SQL Server and I want to Consume Blob Storage, In Azure how the integration looks like

I have a background job which is recurring, what are the different ways to Do, for daily notifications

Web Jobs



Can I do 2 webjgbs in same app Service Plan?

How will you decide I need a separate App Service Plan, looking at scalability or availability

How will you track the issues when it raised in Functions, app Services or etc.. - About Application Insights

Have you worked on Azure Monitor - to track the issues which are attached to Application Insights

Publisher/Subscriber Pattern, How will you identify or track of where is my application, what is my end point and how will you approach.

Did you work on Queues like Service Bus or Storage account queues - Explain me the scenario

I am using the single Topic in SB and multiple subscriber, I want to trigger based on a Field as the parameters in Topics are same how would I achieve that

If you read the queue, how should we process at code level, how ServiceBusQueue class works, DeadLetter Queue or some other concepts in SB works at Code level, explain?

On how many ways you can trigger the Service Bus explain

From .NET I wanted to access the KeyVault - How do I implement it. Explain Class Level

AKS: Have you deploy yourself or DevOps Team, How did you do, Helm Files, Ingress Configuration, Configuration Service Files, CI/CD, etc..

ACR - what is use of it and explain - Mostly used for storing the doctor image

Container and Pod - Difference



Helm Charts - What do Helm Chart do. Why do we need

What CLI you use to test HelmChart in Local machine - Is it with Doctor or something else

Any Experience on Creating VNET or any Networking Concepts

Dependency Injection - Types.. Scoped, Single and Transient

How will you handle the CORS Issues

API End point is consumed by Multiple Applications, How do you manage (Question is related to APIM)

I don't want to write code but I want to validate each request - how can I do in APIM

Did you work on Middleware - Explain me the scenario

How Middleware Executes in a scenario of API Execution to check the Authentication and Authorisation before hitting the Web API

Difference between Sync and Async - give example scenario

Can we use Async without await ?



I have my Web API I wanted to call an external API - wait once the things done and I wanted to put in a queue base don the response for the further process, what should I Do?

What is the concept of concurrent list and normal list

Different between .Result vs ConfigureAwait

If want to connect to SQL what are all the possible approaches are there. ADO.NET, ORM Frameworks etc..

How do we scale the database when the application is scaling

How will you ensure Connection Pooling

While doing the load testing we simulate the Pooling what type of issues we will face mostly

Have you use LINQ

IEnumerable or IQueryable - what is the difference between and when to use

SQL Server - what did you work and explain.

Scaler Functions - explain the scenario



How will you check the query taking time - Execution Plan or SQL Profiler

What is Common Table Expression - explain the scenario

For Unit Testing what Tools you used for API's - NUnit, XUnit, and MSTest

GitHubCopilot - any Specific things to define

CQRS and Mediator Design patterns - Explain

Web Page is Unavailable

Virtusa

Virtusa Interview Process (Lead Software Engineer, YOE 10+)

in each L1, L2 and client round They gave multiple problems to solve (provided IDE for both SQL and C#)

In L1, they expect the exact output, so you need to write a complete and correct program.

In L2 and the client round, pseudocode and clear logic explanation are sufficient

1. AI Interview (MCQ Round)

- * Online assessment consisting of multiple-choice questions
- * Covers programming basics, and core concepts

2. L1 – Technical Interview

- * Focus on coding, problem-solving, and fundamental concepts
- * Includes SQL queries, C# programming, and output-based questions

3. L2 – Techno-Managerial Interview



www.questpod.com

- * Combination of technical knowledge and practical experience
- * Questions on system design, real-time scenarios, and project discussions
- * May include decision-making and problem-handling approaches

4. Client Round (Technical Test – HackerRank)

- * Hands-on coding test conducted on HackerRank
- * Focus on Communication, problem-solving, coding efficiency, and logic building

5. HR Discussion

- * Final round covering salary, availability, and cultural fit
- * General discussion about background, career goals, and expectations

SQL QUESTIONS

L1

1. Given the tables:

- * Employees
- * Projects
- * EmployeeProjects

Write a query to:



www.questpod.com

- * Find employees who have worked on at least 2 projects
- * Identify the longest duration project
- * Return employee names along with the longest project they worked on
- * If multiple longest projects exist, return any one

L2

2. Tables:

- * tblCustomer (CustId, CustName)
- * tbleOrderdtls (OrderId, CustId, OrderNumber)

Write a query to find:

- * CustId
- * CustName
- * OrderCount

For customers having more than 3 orders

3. Laptop Sales:

Tables:

- * Laptop (L_Id, Brand)
- * Sales (Sale_Id, L_Id, Quantity)

Write a query to display:

- * Brand

* TotalQuantity sold

Include brands with zero sales

C# QUESTIONS

L1

1. Given a string "ABC", generate all combinations and permutations

Example: ABC, BCA, CAB, AB, BC, AC

2. What will be the output?

```
class A
{
    public void Display ()
    {
        Console.WriteLine("A");
    }
}

class B:A
{
    public void Display ()
    {
        Console.WriteLine("B");
    }
}
```

```
B b = new A();
```

```
b.Display();
```

```
A a = new B();
```

```
a.Display();
```

Also consider variations using virtual and override

3. Sort the array

Input: {4,0,7,5,2,8,0,0,0}

Output: {0,0,0,0,4,7,5,2,8}

4. Write a program to find duplicate characters in a string

5. What is Concurrent Programming?

6. What is GraphQL and why is it used?

7. Explain CI/CD

8. Differences:

- * .NET 6 vs .NET 8

- * var vs dynamic

- * Task vs Thread

- * async vs await

9. Explain:

- * static class
- * middleware (ASP.NET)
- * temp table

Can temp tables be used in:

- * functions
- * triggers

10. How to handle exceptions in:

- * C#
- * React
- * SQL

11. REST API vs traditional API

12. How to implement API versioning

13. ASP.NET Filters

14. Can we create custom middleware? If yes, how?

15. Alternative for useEffect in React

JavaScript / React Output

16. What will be the output?

```
console.log(true + 1)
```

```
console.log(1 + '2')
```

```
console.log('2' + 1)
```

17. What will be the output?

```
let c;
```

```
console.log(a);
```

```
console.log(b);
```

```
console.log(c);
```

```
var a;
```

```
let b;
```

```
---
```

C# L2

18. Write a method:

```
public static string MergeStrings(string s1, string s2)
```

Requirements:

* If same length → merge alternately

Example: ABC + 123 → A1B2C3

* If S1 longer → append remaining at end

Example: ABCDEFG + 123 → A1B2C3DEFG

* If S2 longer → append remaining at end

Example: ABC + 123456 → A1B2C3456

Virtusa 11/June/2026 (Microservices and Docker Focussed)

General & Background

* Can you introduce yourself?

Microservices Architecture

* Have you done work with microservices?

* Have you done any monolith to microservices migration?

* Do you know how to identify what microservices to create from a monolith application?

* What is the advantage of using microservices over a monolithic application?

* What was your biggest challenge when developing the microservices project, and how did you solve it?

C# & .NET Core

* Write a C# code snippet to remove duplicate characters from an input string (e.g., if the input is "Anand", the output should be "AND").

* Why does .NET Core run faster than the older .NET Framework?

Cloud & DevOps (Azure & Docker)

- * What are the various Azure services you have used?
- * Can you give an example of a task you have done using Azure Functions?
- * How did you deploy the Azure function?
- * Have you worked with Docker files?
- * How did you deploy the microservice?

TCS Consultancy 20 +LPA Package (.NET Full Stack + AI)

Today, I attended an interview with TCS for a .NET Full Stack role, and I wanted to share some of the key 🔑 questions that were asked. It was a great experience with a strong focus 🧐 on real-time scenarios, problem-solving, and practical knowledge.

📄 .NET Core Questions

- 1) Explain Singleton lifetime in Dependency Injection with a real-time example.
- 2) Explain SOLID principles.
- 3) Difference between Hashtable and Dictionary.
- 4) Primary vs Secondary Constructor in C#.
- 5) Convert.ToString() vs ToString()
- 6) First(), FirstOrDefault(), Single(), SingleOrDefault() differences
- 7) Middleware + write custom middleware
- 8) GET vs POST - which is better and why
- 9) API performance optimization

10) Pagination in .NET Core

11) Filters in MVC

12) IActionResult vs ActionResult

13) Why API is slow and how to fix it

AI-Related Questions

1) Explain AI integration in your project

2) How to call an AI API in .NET

3) What is Prompt Engineering

4) Design AI-powered chatbot in .NET

5) Chatbot giving wrong answers - how to fix

Scenario-Based Questions

1) You already have authorization applied to an API, but inside the API method, you need to bypass authorization for a specific condition. How will you implement it in real-time?

2) You have `int a = 1`. At runtime, you want to assign dynamically like `var e = a`. Which data type would you use and why?

3) Your MVC API returns JSON like `{ "id": 1, "name": null, "age": 25 }`. How will you ignore null values in the response?

SQL Scenarios

Query Behavior:

1) `SELECT * INTO newtable FROM oldtable WHERE 1 = 0;`

2) `SELECT * INTO newtable FROM oldtable WHERE 1 = 1;`

What will happen and where are these used in real-time?

3) Can you write pagination in SQL?

4) Insert values like Male = 'M', Female = 'F', Other = 'O'. Write SQL query.

5) While creating a UserDetails table, set default age = 18. Write SQL query.

4) Find the 3rd highest tanker level of water and also find duplicate tanker names.

 Overall, the interview focused heavily on:

- ◆ Real-time scenarios
- ◆ API performance
- ◆ SQL problem-solving
- ◆ Basic AI understanding

It was a great learning experience!

 Tip:

Focus on real-time scenarios, API performance, SQL optimization, and system design thinking.

TCS Experience 2 Special Notes from a developer.

1st technical round is very important. Most of the time they start with very basic but practical questions.

For me, they asked:

- One simple coding program related to string splitting and finding the 3rd highest integer
- .NET Core basics
- Authentication vs OAuth
- How to expire a JWT token
- Custom Middleware
- Uses of async / await
- DB query optimization
- CTE
- Demerits of Cursor in SQL
- LINQ — when to use AsNoTracking in Entity Framework
- SQL query to get department-wise max salary

The 2nd round was managerial and was more straightforward around project responsibilities and tech roles.

This is why fundamentals + practical understanding matter a lot in interviews 😊

Wolters interview

What's your approach towards designing a microservice pattern basically? Like, how do you approach

to break down your project and the designing part of it? Like, how you design the solutions?

So, how you define the bounded context, like, if there are overlaps, then how will you take care of this? Okay.

So, if there are overlaps, so, we will have a database, okay, a common database, okay, to take care of that, right, like,

that we call the event sourcing, correct, eventual consistency. So, we will capture the sequence of events, like, user has done



button click, user has done this, this event is created, order is created. .

Can you explain, like, from practical standpoint, how you implement saga pattern.

So, basically, saga pattern is to take care of distributed transaction, right. We have one transaction, like, payment, invoice,

whatever it is. And then, it says, basically, commit all or commit nothing, right.

So, if something is broken, we will reverse everything, okay. Yeah, I mean, that's the theory part of it.

From practical standpoint,

can you explain how many services you will create? Will saga be a different service, or it will be embedded into your, any of the microservice, then how will you compensate it from practical standpoint? Okay.

So, where saga sits, basically, first of all?

Yeah. So, where that logic will be implemented?

Let's say, one service is inter-service communication. We are calling one service to another service, then another service to, from the same service, we are calling multiple service. So, there, we will implement saga pattern.

So, at every places where the communication happens, you will implement saga? So, who is responsible for

implementing the saga, the consumer or the producer,

or who is responsible? Okay. And can you explain event sourcing? You just momentarily talked about event sourcing. Yeah.

Okay. So, how do you connect your .NET Core Web API to any Azure resource which is protected? Azure resource



which is protected? Protected in the sense? It's private, basically. It's private, not available over the internet.

let's say, I want to manage my SQL connections efficiently, I mean, your .NET Core Web API app is creating SQL connections whenever there is a request

and there are high volume of traffic that is coming in and due to that, it is instantiating so many SQL connections. Okay.

So, but you know, SQL has its own limitation to open number of, on the number of connections to be open at a time, right? So, how you will optimize

or design this situation? Okay. You understood the problem statement, right? Yeah, I understood. Basically,

we are getting so many traffics on SQL connection, right, SQL server side.

Right. So, basically we have to manage that. Connection pool, we can manage, right? We can set a limit in the connection pool, like this 20, 30, 40,

whatever it is and then whatever the extra request will come, we will reject it.

No, we can't reject, we have to serve every request. Okay. So, there is a, okay, there is a pattern, CQRS pattern, okay.

what's your design strategy to design K8 which is deployed into your VNet, like, how many node pools you have and how many microservices you have and how you design that to be deployed into the K8.?

One pod can have multiple containers. Can a pod have different containers of different types?

whenever the request comes from outside the world, right, from HTTP, from mobile, whatever it is, it comes to ingress and ingress routes the traffic to the appropriate service.

Do you know how a pod connects to a storage account, Azure storage account, or maybe a key vault?



How it connects?

for microservices, do you think solid principles are applicable?

Do you know sidecar design pattern? Sidecar? Yeah.

Do you know Docker containers ? how to create a container and all?

So, how do you select an image? We, we pull the images, no? Docker pull images, right? No, no, I mean, I mean, what's,

what's the criteria to select an image? What is the criteria to select the image? I mean, how do you decide for this microservice

this should be the image, for that microservice this should be the image, how do you decide? So, that we will, anyway we will pull,

we will, you know, define in our YAML file, right? We have a YAML file, correct? So, we know this... Yeah, but the base, base image

you need to select anyway, right? Yeah, yeah, yeah. So, the base image, how do you decide this base image should be of this type,

this base image should be of that type for this microservice? Because base image is not common for every microservice, right?

So, we have something called tagging, with the help of tagging we can define, we can identify,

I'm just guessing, I'm not sure. We have tagging in the image, okay

We give a name, with the help of tagging we can give a name, okay, in the Docker. Whenever we create a Docker image, Docker image

tagging hyphen T and then we give a name. With the help, obviously we will give a distinct name for all the services, right? There we can, yeah, tagging is there.

you have integrated Azure OpenAI into your project, so, what was the use case?

Do you know spec-driven development?

Infosys

I have attended the AI interview but not in Infosys so mostly it starts with the basic question with .Net.

And you will have two options you need to tell the answer oral like an interview or you can type it and enter, it will go through the answer and will give you a response. You can expect. Net core and net framework and async await kind of questions it will also ask questions mostly from your resume and the answer you are giving from previous questions mostly.

Better try to prepare string problems and small programs using C#.

I attended 2 AI interviews this was the pattern.

WIPRO

This is for 9 years experience.

Wipro 1st round held on 30th April 2026

1. General & Introduction

- * Can you please tell me about yourself?
- * How many years of experience do you have?
- * What technologies do you have experience in? (e.g., .NET Core, Angular, Entity Framework, SQL Server)
- * Tell me about your last or current project. What is your role and responsibility in that project?
- * Which version of .NET and Angular have you worked with?
- * How many projects have you completed using .NET Core and Angular?
- * On a scale of 1 to 5, how would you rate yourself in:
 - * .NET Core
 - * Angular
 - * Web API
 - * Entity Framework

* What was the architecture and data flow followed in your last project?

2. .NET Core & C# (Server-Side)

* What is the difference between an abstract class and an interface? Give a real-time scenario where you used an abstract class in your project.

* Have you used delegates? Can you provide a real-time scenario where you used them?

* What is the difference between `First`, `FirstOrDefault`, `Single`, and `SingleOrDefault` in LINQ?

* What is the difference between `readonly`, `const`, and `static`?

* What is the difference between String and StringBuilder?

* What is middleware? How do you create custom middleware in .NET Core?

* What is the Kestrel server in .NET Core?

* How do you handle sessions in .NET Core?

* How do you implement API versioning?

* What is content negotiation in .NET Core Web API?

SOLID Principles & Architecture

* What is the Single Responsibility Principle (SRP)?

* What is the Liskov Substitution Principle (LSP)?

* What is Dependency Injection (DI) and why do we use it?

* What is the difference between `Transient`, `Scoped`, and `Singleton` lifetimes? Where and why did you use them in your project?

* Can we inject a `Scoped` service into a `Singleton` service?

* **Scenario-based design question:** Suppose you go to a mall, purchase items, and go to the billing counter. There are multiple payment options available (Credit Card, Debit Card, Voucher). In the future, new options like Cash or Coupons might be introduced. How would you design a loosely



www.questpod.com

coupled system for this? How many classes and interfaces would you create so that adding a new system requires minimal changes?

Security & Authentication

- * What mechanism have you used for authentication and authorization in your applications?
- * How do you generate a JWT (JSON Web Token) in Web API?

3. SQL Server & Databases

- * How do you improve query performance on the SQL Server side?
- * What is the difference between a clustered index and a non-clustered index?
- * What is a covering index in SQL Server?
- * What is a Common Table Expression (CTE)? Have you used a CTE inside a stored procedure, and why?
- * What is the difference between a Temporary Table (`#temp`) and a Table Variable (`@table`)?
- * What is the difference between `ROW\_NUMBER()` and `DENSE\_RANK()` window functions?
- * **Coding Query Challenge:** > Write a query to find the **second highest salary department-wise**.

> * Table 1 (`Employee`): Columns are `EmployeeID`, `Salary`, and `DepartmentID`.

> * Table 2 (`Department`): Columns are `DepartmentID` and `DepartmentName`.

> * The final result set should display: `Employee Name`, `Salary`, and `Department Name`.

>

>

4. Entity Framework (EF)

- * What is the difference between Eager Loading, Explicit Loading, and Lazy Loading in Entity Framework?
- * What is the \$N+1\$ query problem? Does it happen while using eager loading or lazy loading?

5. Angular (Client-Side)

- * If there are multiple components with no direct parent-child relation, how do you transfer data between them?
- * What is the difference between a `Subject`, `BehaviorSubject`, and `ReplaySubject`?
- * What is an Interceptor, and why do we use them in Angular?
- * How do you maintain/keep tokens (like JWT) in Angular?
- * What is the `standalone` property in Angular components, and why do we use it?
- * How do you improve the performance of an Angular application?
- * What is the difference between an `Observable` and a `Promise`?

6. Cloud Services (Azure / AWS)

- * Have you used Azure or AWS in your project? Which specific services did you use?
- * *(Clarification from interviewer)* Have you worked with Azure Functions or AWS Lambda functions?

Wipro 2nd round held on 8th May 2026

1. Architecture & Design Principles

- * Can you explain the application architecture of your recent project? (Beyond just listing the tech stack).
- * What is the difference between a Clean Architecture and a Traditional Three-Layer Architecture?
- * Are you familiar with Layered Architecture? Can you explain how it works?

2. Angular (Client-Side)

- * How do you refresh an expired authentication token on the front end using an HTTP Interceptor?
- * *(Follow-up context)**: On receiving a `401 Unauthorized` error from the API while the user is actively using the app, what RxJS operators (e.g., `switchMap`, `catchError`) and logic do you use to seamlessly request a token refresh?
- * How do you make an Angular service (like a `UserService`) a global **Singleton Service** so that it maintains user entities/state consistently across multiple screens during CRUD operations?
- * What is the purpose of the `@Injectable()` decorator, and what does the `providedIn: 'root'` configuration indicate?

3. .NET Core & C# (Server-Side)

- * What is Middleware in .NET Core?
- * **Module Architecture Scenario**: If you are developing a brand new "Customer Module" within an existing application, what specific files/layers will you create on the .NET Core API side (e.g., Controllers, Services, Repositories, DTOs)?
- * Where and how do you configure connection strings and application settings in a .NET Core web API? (e.g., `appsettings.json`).

4. LINQ & Entity Framework Core (EF Core)

* **LINQ Coding Challenge:** How do you write a LINQ query to fetch the **third highest salaried employee** from a list of customers/employees?

* How do you handle cases where multiple employees have the exact same salary when trying to find the N^{th} highest salary using methods like `Skip()` and `Take()`? Why might simply using `FirstOrDefault()` fail in this scenario?

5. SQL Server & Databases

* **Index Design Strategy:** If you have a massive table with around 30 columns and need to create indexes on it, what specific factors or query patterns (e.g., `WHERE` clauses, selectivity, complexity) do you analyze to choose the correct columns?

* **Performance Tuning Use-Case:** You have a transactional database table containing over **1 million records**. A stored procedure needs to fetch records from this table, perform complex calculations, and return the output. Would you use a **Common Table Expression (CTE)** or a **Global Temporary Table (`##temp`)**, and why?

* **SQL Coding Challenge:** Write a query to **remove duplicate records** from a `User` table that has columns `Email`, `UserID`, and `UserName` but **no primary key**, utilizing a CTE and `PARTITION BY`.

* How many types of indexes are available in SQL Server?

6. Azure Cloud Services

* Which specific Azure cloud services have you utilized for application deployment and hosting?

* What is the difference between an **HTTP Trigger** and a **Timer Trigger** in Azure Functions?



www.questpod.com

**** Azure Coding Challenge:**** Write the syntax/definition for a basic **** Azure Function app**** using an HTTP trigger that handles adding an entity (like a customer).

FN(LastLetterAlphabet) 4 years Questions

Explain SOLID principles with real example.

Explain Open/Closed Principle using Strategy pattern.

Does Factory pattern violate OCP?

What is Dependency Inversion Principle and how DI implements it?

Why use constructor dependency injection?

How to design interest calculator using Strategy pattern?

When to use Factory vs Dependency Injection?

How to create concrete implementation of a strategy?

Design a simple banking system (Deposit & Withdraw).

Where should business logic be placed (Controller vs Service vs Domain)?

Why should controller not contain business logic?

How do you handle different account types without if-else?

How do you enforce rules like minimum balance?

What is resource leak?

Why should we use using keyword in C#?

What is SQL injection and how to prevent it?

What is race condition in banking transactions?

Why should we use database transactions?

Show total balance across accounts for each customer.

How to optimize query for latest transaction per account?

...

diff transient and scope lifetime method on .net core

what are performance level use on EF core

diff eager loading and lazy loading



www.questpod.com

how use or implment the oath 2.o security implment

Aync and await

angular - auth guard,interceptor

group by

having clause

id dept country

1 IT USA

2 HR USA

3 Account USA

select country from tblcountry group by country

```
string str="sarbeswarsahoo";
```

```
int count=0;
```

```
for(int i=0; i < str.length)
```

```
{
```

```
foreach(Listitem li in str[1]-1)
```

```
{
```

```
if(li )
```



www.questpod.com

```
{
```

```
}
```

```
char ch = li
```

```
count = li.count;
```

```
console.Write(count + )
```

```
}
```

```
}
```