

ASP.NET Interview Questions Day 1



Hi , this conversation continues for 10 days,so give me 1 day worth matter then I'll ask for day 2 ,I am a asp.net developer and i have around 2.6 year experience,with knowledge of c#,SQL, javascript,html and css .give me interview questions in a tabular format,ACT as a trainer and give me questions for day 1 in a tabular format



Here's a set of interview questions for Day 1, focusing on ASP.NET fundamentals, C#, and SQL. We'll start with foundational questions to assess your knowledge and build up gradually.

Topic	Question	Answer Key/Hint
ASP.NET	What is ASP.NET, and how does it differ from classic ASP?	ASP.NET is a web application framework developed by Microsoft. It differs from classic ASP by using .NET languages like C# and VB.NET, and offering features like web forms and MVC.
	Explain the ASP.NET page life cycle.	Page life cycle events include initialization, load, control events, rendering, and unloading.
	What are HTTP handlers, and how do you create a custom handler in ASP.NET?	HTTP handlers process requests and return responses. Custom handlers implement <code>IHttpHandler</code> .
C#	Describe the differences between <code>abstract</code> classes and <code>interfaces</code> in C#.	Abstract classes allow implementation of methods; interfaces only define methods with no implementation.
	What are <code>Nullable</code> types in C#?	Nullable types allow value types to hold a <code>null</code> value, using <code>Nullable<T></code> or <code>?</code> .
	Explain <code>boxing</code> and <code>unboxing</code> in C#.	Boxing converts a value type to <code>object</code> ; unboxing extracts the value type from <code>object</code> .
SQL	What is the purpose of indexes in SQL databases?	Indexes improve query speed by allowing faster data retrieval, though they can slow down inserts/updates.
	Differentiate between <code>INNER JOIN</code> and <code>OUTER JOIN</code> in SQL.	<code>INNER JOIN</code> returns matching rows; <code>OUTER JOIN</code> includes non-matching rows as well.
	What are primary keys and foreign keys? Explain their significance in SQL.	Primary keys uniquely identify a row; foreign keys create a relationship between tables by referencing a primary key.
JavaScript	Explain <code>hoisting</code> in JavaScript.	Hoisting moves declarations to the top of their scope during runtime, though only <code>var</code> declarations are hoisted (not initializations).
HTML/CSS	What is the difference between inline, internal, and external CSS?	Inline is within an element's style attribute; internal is within a <code><style></code> tag in HTML; external is in a separate <code>.css</code> file.
	What does the <code>DOCTYPE</code> declaration do in HTML?	It tells the browser which HTML version is being used, affecting rendering.

Let me know when you're ready for Day 2!



its day two



For Day 2, we'll dive deeper into ASP.NET concepts, touch on more advanced C# topics, and cover SQL optimization techniques.

Topic	Question	Answer Key/Hint
ASP.NET	Explain the difference between <code>`Session`</code> and <code>`ViewState`</code> in ASP.NET.	<code>`Session`</code> is server-side, used to store data per user session. <code>`ViewState`</code> is client-side, persisting state within a page.
	What are <code>`Web Forms`</code> and <code>`MVC`</code> in ASP.NET? How do they differ?	Web Forms use event-driven model with controls, MVC follows a Model-View-Controller architecture for better separation of concerns.
	Describe how authentication and authorization work in ASP.NET.	Authentication identifies the user; authorization determines access levels.
	What is a <code>`Global.asax`</code> file, and what is its purpose?	<code>`Global.asax`</code> handles application-level events, like <code>`Application_Start`</code> , <code>`Session_Start`</code> , and error handling.
C#	What is <code>`Polymorphism`</code> , and how is it achieved in C#?	Polymorphism allows methods to perform differently based on input, achieved via method overriding and overloading.
	Explain <code>`Delegates`</code> and <code>`Events`</code> in C#.	Delegates reference methods with specific signatures; events use delegates to notify subscribers.
	What is <code>`LINQ`</code> , and why is it useful in C#?	LINQ (Language Integrated Query) simplifies queries across collections (objects, databases, XML).
	Describe the difference between <code>`throw`</code> and <code>`throw ex`</code> in exception handling.	<code>`throw`</code> preserves the original stack trace, <code>`throw ex`</code> resets it, making it harder to debug the error's origin.
SQL	How does a <code>`Clustered Index`</code> differ from a <code>`Non-Clustered Index`</code> in SQL?	Clustered indexes sort data physically, one per table; non-clustered indexes don't, multiple can exist.
	What is the purpose of <code>`JOINS`</code> in SQL? Explain <code>`LEFT JOIN`</code> , <code>`RIGHT JOIN`</code> , and <code>`FULL JOIN`</code> .	Joins combine rows from tables; <code>`LEFT`</code> shows all from left table, <code>`RIGHT`</code> all from right, <code>`FULL`</code> from both.
	What is a <code>`subquery`</code> , and when would you use it?	Subqueries are nested queries; useful for complex filtering or aggregations.
	Explain <code>`Normalization`</code> in SQL. Why is it important?	Normalization reduces redundancy, organizes data efficiently, follows rules like 1NF, 2NF, etc.
JavaScript	What is <code>`event bubbling`</code> in JavaScript, and how can you stop it?	Event bubbling propagates events from child to parent elements; <code>`stopPropagation()`</code> can prevent it.
HTML/CSS	Describe the box model in CSS and its components.	The box model includes <code>`content`</code> , <code>`padding`</code> , <code>`border`</code> , and <code>`margin`</code> layers around HTML elements.
	What are <code>`pseudo-classes`</code> in CSS? Give examples.	Pseudo-classes define special states of elements, like <code>`:hover`</code> , <code>`:active`</code> , <code>`:first-child`</code> .

Let me know when you're ready for Day 3!



Day 3



Here's the set of questions for Day 3, focusing on ASP.NET security, advanced C# topics, and database optimization techniques.

Topic	Question	Answer Key/Hint
ASP.NET	What is Cross-Site Scripting (XSS) and how can you prevent it in ASP.NET?	XSS allows attackers to inject scripts; prevent it by encoding output and using anti-XSS libraries like AntiXssEncoder .
	Explain Cross-Site Request Forgery (CSRF) and how to prevent it in ASP.NET.	CSRF tricks users into performing actions; prevent it using AntiForgeryToken() in forms and verifying on the server.
	What is the purpose of web.config and machine.config files in ASP.NET?	web.config stores application-level settings; machine.config is a system-wide configuration file.
	How does ASP.NET MVC handle routing?	MVC uses the RouteConfig class to define URL patterns and map them to controller actions.
C#	What are generics in C#, and why are they useful?	Generics allow type-safe data structures and methods, reducing code duplication and improving performance.
	Explain extension methods in C# with an example.	Extension methods add new functionality to existing types without modifying them; defined as static methods in static classes.
	What is the difference between ref and out parameters in C#?	ref requires initialization before passing; out does not require initialization, but must be set within the method.
	Describe Dependency Injection (DI) and its purpose in C#.	DI is a design pattern for managing dependencies; it improves flexibility, testability, and maintenance by injecting dependencies.
SQL	What is SQL Injection , and how can you prevent it?	SQL Injection is an attack to manipulate queries; prevent by using parameterized queries and ORM libraries like Entity Framework.
	What is a Stored Procedure ? List advantages of using stored procedures in SQL.	Stored procedures are precompiled SQL queries, offering performance benefits, reusability, and security.
	Explain the concept of transactions in SQL and their importance.	Transactions ensure multiple queries execute as a single unit, maintaining data integrity. Use BEGIN , COMMIT , and ROLLBACK .
	How do views work in SQL, and when should you use them?	Views are virtual tables based on queries, useful for simplifying complex queries and securing underlying tables.
JavaScript	What is closure in JavaScript? Give an example.	Closure allows inner functions to access outer function variables even after the outer function has executed.
HTML/CSS	What is CSS Flexbox , and how does it help in responsive design?	Flexbox is a CSS layout model for creating flexible and responsive layouts, controlling alignment, direction, and spacing.

Topic	Question	Answer Key/Hint
	Explain the purpose of <code>`media queries`</code> in CSS.	Media queries apply styles based on device attributes like screen width, enabling responsive design for different screen sizes.

Let me know when you're ready for Day 4!



hey its day 4



Here's the set of questions for Day 4, focusing on ASP.NET advanced topics, C# asynchronous programming, and SQL indexing.

Topic	Question	Answer Key/Hint
ASP.NET	What is <code>`AJAX`</code> , and how is it used in ASP.NET?	AJAX enables asynchronous requests to the server, enhancing user experience by not refreshing the entire page.
	Explain <code>`ViewState`</code> vs. <code>`SessionState`</code> vs. <code>`ApplicationState`</code> in ASP.NET.	<code>`ViewState`</code> stores data at the page level, <code>`SessionState`</code> at the user level, and <code>`ApplicationState`</code> across all users.
	What is an <code>`HttpModule`</code> in ASP.NET, and when would you use one?	<code>`HttpModule`</code> handles application-level events in the request pipeline, useful for tasks like authentication and logging.
	Describe the purpose and usage of <code>`async`</code> and <code>`await`</code> keywords in ASP.NET and C#.	<code>`async`</code> / <code>`await`</code> enable asynchronous programming by marking tasks as non-blocking, improving application responsiveness.
C#	What is the <code>`Task`</code> class in C#, and how does it relate to asynchronous programming?	<code>`Task`</code> represents an asynchronous operation, used for multithreading and async/await patterns to improve performance.
	Explain <code>`Parallel`</code> vs. <code>`Task`</code> in C#. When would you use each?	<code>`Task`</code> is for async tasks; <code>`Parallel`</code> performs operations concurrently. Use <code>`Parallel`</code> for independent tasks, <code>`Task`</code> for async ops.
	What is <code>`garbage collection`</code> , and how does it work in C#?	Garbage collection manages memory automatically by reclaiming unused objects to free up resources.
	What is a <code>`Memory Leak`</code> , and how can it occur in .NET?	A memory leak occurs when objects aren't released from memory; occurs in .NET if event handlers or resources are not properly disposed.
SQL	Explain <code>`indexing`</code> and its impact on database performance.	Indexes improve query speed by providing quick lookups, though they can slow down inserts/updates.
	What is the difference between a <code>`unique`</code> index and a <code>`non-unique`</code> index?	A unique index does not allow duplicate values, while a non-unique index can have duplicates.
	How does <code>`normalization`</code> impact database indexing?	Normalization organizes data, reducing redundancy, which often complements effective indexing.
	What is a <code>`composite key`</code> , and why would you use it in SQL?	A composite key uses multiple columns to uniquely identify a row, useful for complex relationships.

Topic	Question	Answer Key/Hint
JavaScript	What is Debouncing , and how does it work in JavaScript?	Debouncing limits function calls, only triggering after a delay, often used to optimize repeated events like resize or scroll .
HTML/CSS	Explain the CSS Grid layout and its differences compared to Flexbox .	CSS Grid provides two-dimensional layouts (rows and columns), while Flexbox is one-dimensional (row or column).
	What is the purpose of the z-index property in CSS?	z-index controls the stack order of overlapping elements; higher values appear above lower ones.

Let me know when you're ready for Day 5!



day 5



Here's the set of questions for Day 5, focusing on ASP.NET Core basics, C# design patterns, SQL optimization, and modern web development concepts.

Topic	Question	Answer Key/Hint
ASP.NET	What is ASP.NET Core, and how does it differ from the traditional ASP.NET Framework?	ASP.NET Core is cross-platform, open-source, and modular, offering better performance and flexibility compared to ASP.NET Framework.
	What are Middleware components in ASP.NET Core, and how are they used?	Middleware are request pipeline components to handle requests/responses, configured in Startup.cs .
	Explain the role of Dependency Injection (DI) in ASP.NET Core.	DI is built into ASP.NET Core for managing object dependencies, improving modularity and testability.
	What is Kestrel , and how does it work in ASP.NET Core?	Kestrel is a cross-platform web server used as the default in ASP.NET Core.
C#	What is the Singleton design pattern, and how is it implemented in C#?	Singleton ensures a class has only one instance, implemented using a private constructor and a static property.
	Describe the Factory design pattern and provide a use case in C#.	The Factory pattern creates objects without specifying their concrete classes, useful for abstracting object creation logic.
	What is Reflection in C#, and how can it be used?	Reflection allows inspection of metadata (types, properties, methods) at runtime, useful for dynamic programming scenarios.
	Explain lazy loading in C#.	Lazy loading delays the initialization of objects until needed, improving performance.
SQL	What is the difference between HAVING and WHERE clauses in SQL?	WHERE filters rows before grouping; HAVING filters groups after aggregation.
	Explain execution plans in SQL and how to analyze them.	Execution plans show query execution steps; use them to identify bottlenecks and optimize performance.
	What are triggers in SQL, and when should they be used?	Triggers are special procedures executed automatically on DML events, used for enforcing business rules.
	What are CTEs (Common Table Expressions), and how are they useful in	CTEs simplify complex queries by creating temporary named result sets, improving readability.

Topic	Question	Answer Key/Hint
	SQL?	
JavaScript	What is <code>Promise.all()</code> , and how is it used in JavaScript?	<code>Promise.all()</code> resolves multiple promises concurrently, succeeding only if all promises succeed.
HTML/CSS	What is the difference between <code>em</code> , <code>rem</code> , and <code>px</code> units in CSS?	<code>em</code> is relative to the parent, <code>rem</code> is relative to the root, and <code>px</code> is absolute.
	How does <code>responsive design</code> differ from <code>adaptive design</code> in web development?	Responsive adjusts layout fluidly; adaptive uses specific layouts for different screen sizes.

Let me know when you're ready for Day 6!



Day 6



Here's the set of questions for Day 6, focusing on advanced ASP.NET Core concepts, multithreading in C#, SQL data integrity, and modern front-end practices.

Topic	Question	Answer Key/Hint
ASP.NET	What is <code>Razor Pages</code> in ASP.NET Core, and how does it differ from MVC?	Razor Pages are a simplified framework for building page-focused web apps, unlike MVC, which uses controllers and views.
	What are <code>Tag Helpers</code> in ASP.NET Core, and how are they used?	Tag Helpers are server-side components that render HTML elements dynamically, simplifying Razor syntax.
	How does <code>Model Binding</code> work in ASP.NET Core?	Model Binding maps HTTP request data (e.g., query strings, form data) to action method parameters or properties.
	Explain the concept of <code>Routing</code> in ASP.NET Core. How is it different from traditional ASP.NET?	Routing in Core uses middleware and is more flexible, supporting attribute-based and conventional routing together.
C#	What is <code>multithreading</code> , and how is it implemented in C#?	Multithreading runs multiple threads concurrently using classes like <code>Thread</code> , <code>Task</code> , or <code>ThreadPool</code> .
	Explain the difference between <code>lock</code> and <code>Monitor</code> in C#.	Both synchronize access to shared resources, but <code>Monitor</code> provides more control, like <code>TryEnter</code> and signaling.
	What are <code>async</code> and <code>await</code> , and how do they relate to multithreading?	<code>async</code> and <code>await</code> simplify asynchronous programming, but they don't create threads; they utilize the thread pool for efficiency.
	What is the difference between <code>IEnumerable</code> and <code>IQueryable</code> in C#?	<code>IEnumerable</code> executes queries in memory, while <code>IQueryable</code> performs queries on the data source, enabling server-side execution.
SQL	What is <code>data integrity</code> , and how is it enforced in SQL?	Data integrity ensures accuracy and consistency of data, enforced using primary keys, foreign keys, unique constraints, and triggers.

Topic	Question	Answer Key/Hint
	What are <code>`ACID`</code> properties in SQL transactions?	ACID stands for Atomicity, Consistency, Isolation, and Durability, ensuring reliable transactions.
	How does a <code>`full-text index`</code> differ from a regular index in SQL?	A full-text index is used for efficient text searches, while regular indexes improve query performance based on column values.
	What is the purpose of the <code>`NOLOCK`</code> hint in SQL queries?	<code>`NOLOCK`</code> allows reading data without placing shared locks, improving performance but risking dirty reads.
JavaScript	What is the difference between <code>`var`</code> , <code>`let`</code> , and <code>`const`</code> in JavaScript?	<code>`var`</code> has function scope, <code>`let`</code> and <code>`const`</code> have block scope. <code>`const`</code> is for constants.
	Explain how the <code>`Fetch API`</code> works in JavaScript for making HTTP requests.	The <code>`Fetch API`</code> provides a modern way to make HTTP requests, returning promises for better asynchronous handling.
HTML/CSS	What are <code>`CSS animations`</code> , and how are they different from <code>`transitions`</code> ?	Animations allow keyframe-based effects, while transitions handle state changes (e.g., hover).
	What is <code>`SVG`</code> , and why is it used in web development?	SVG (Scalable Vector Graphics) is an XML-based format for vector graphics, offering resolution-independent images.

Let me know when you're ready for Day 7!



Day 7



Here's the set of questions for Day 7, focusing on ASP.NET API development, C# LINQ, SQL optimization techniques, and advanced JavaScript and CSS.

Topic	Question	Answer Key/Hint
ASP.NET	What is <code>`Web API`</code> in ASP.NET Core, and how is it different from MVC?	Web API is designed for building RESTful services, while MVC is for web applications with views.
	Explain how <code>`attribute routing`</code> works in Web API.	Attribute routing uses annotations (e.g., <code>`[Route("api/products/{id}")]</code>) for defining routes directly on action methods.
	How do you secure a Web API in ASP.NET Core?	Use authentication (JWT, OAuth), HTTPS, CORS, and input validation.
	What is <code>`CORS`</code> , and why is it important in Web API development?	CORS (Cross-Origin Resource Sharing) allows restricted resources to be accessed from different domains, improving security.
C#	What is LINQ, and what are its advantages in C#?	LINQ (Language Integrated Query) simplifies data manipulation using queries integrated into C#.
	Explain the difference between <code>`LINQ to Objects`</code> , <code>`LINQ to SQL`</code> , and <code>`LINQ to Entities`</code> .	<code>`LINQ to Objects`</code> works with in-memory collections, <code>`LINQ to SQL`</code> interacts with databases, and <code>`LINQ to Entities`</code> uses Entity Framework.
	What is the difference between <code>`First()`</code> , <code>`FirstOrDefault()`</code> , <code>`Single()`</code> , and <code>`SingleOrDefault()`</code> ?	<code>`First()`</code> / <code>`FirstOrDefault()`</code> return the first element, while <code>`Single()`</code> / <code>`SingleOrDefault()`</code> expect exactly one element.

Topic	Question	Answer Key/Hint
	<code>`Single()`</code> , and <code>`SingleOrDefault()`</code> in LINQ?	
	What is <code>`Deferred Execution`</code> in LINQ?	LINQ queries are executed only when enumerated, improving performance for large datasets.
SQL	What are the advantages of using <code>`Indexed Views`</code> in SQL Server?	Indexed views improve performance by storing query results, but they require stricter constraints.
	Explain the difference between <code>`Clustered`</code> and <code>`Non-Clustered Index`</code> .	Clustered indexes sort data physically; non-clustered indexes create a separate structure for lookup.
	How do you optimize a slow SQL query?	Analyze execution plans, use proper indexing, avoid <code>SELECT *</code> , reduce joins/subqueries, and consider indexing columns used in filters.
	What are <code>`temporary tables`</code> , and when should they be used?	Temporary tables store data for intermediate processing, useful in complex queries or stored procedures.
JavaScript	What is <code>`event bubbling`</code> and <code>`event capturing`</code> in JavaScript?	Bubbling propagates events from child to parent; capturing starts from parent to child.
	Explain the concept of <code>`Promises`</code> and <code>`async/await`</code> in JavaScript.	Promises handle asynchronous operations; <code>`async/await`</code> simplifies promise handling for better readability.
HTML/CSS	What is the <code>`CSS Box Model`</code> , and how does it affect layout?	The box model consists of content, padding, border, and margin, determining element size and spacing.
	Explain the difference between <code>`absolute`</code> , <code>`relative`</code> , <code>`fixed`</code> , and <code>`sticky`</code> positioning in CSS.	<code>`Absolute`</code> positions relative to the nearest positioned ancestor; <code>`relative`</code> to itself; <code>`fixed`</code> to the viewport; <code>`sticky`</code> to scroll.

Let me know when you're ready for Day 8!



Day8



Here's the set of questions for Day 8, focusing on **ASP.NET Core hosting, advanced C# concepts, SQL transactions, and modern web development tools.**

Topic	Question	Answer Key/Hint
ASP.NET Core	What are the different hosting models in ASP.NET Core?	In-Process Hosting (IIS), Out-of-Process Hosting (Kestrel), and Self-hosted applications using Kestrel.
	Explain the role of <code>`appsettings.json`</code> in ASP.NET Core applications.	<code>`appsettings.json`</code> stores configuration settings such as connection strings, app settings, and logging settings.
	What is <code>`IConfiguration`</code> in ASP.NET Core? How do you access configuration values?	<code>`IConfiguration`</code> provides access to configuration values using dependency injection (<code>`Configuration.GetValue<T>("key")`</code>).
	How do you enable logging in ASP.NET Core?	Use built-in <code>`ILogger`</code> or third-party providers (e.g., Serilog, NLog) configured via <code>`Startup.cs`</code> .
C#	What is a <code>`delegate`</code> in C#, and how is it different from an <code>`event`</code> ?	Delegates are function pointers, while events are specific to signaling actions using delegates.

Topic	Question	Answer Key/Hint
	Explain the concept of <code>Generics</code> in C#. Why are they important?	Generics allow type safety without specifying the exact type, reducing code duplication and improving performance.
	What is the <code>Dispose</code> method and <code>IDisposable</code> interface in C#? Why are they used?	<code>IDisposable</code> ensures proper resource cleanup, often used with <code>using</code> blocks for unmanaged resources like file streams or database connections.
	What are <code>extension methods</code> in C#? Provide a use case.	Extension methods add functionality to existing types without modifying them, e.g., adding utility functions for <code>string</code> or <code>List<T></code> .
SQL	What are transactions in SQL? Why are they important?	Transactions ensure operations are executed reliably using <code>BEGIN TRANSACTION</code> , <code>COMMIT</code> , or <code>ROLLBACK</code> for atomicity.
	Explain the difference between <code>COMMIT</code> and <code>ROLLBACK</code> in SQL transactions.	<code>COMMIT</code> saves changes permanently, while <code>ROLLBACK</code> undoes changes made within the transaction.
	What is <code>deadlock</code> in SQL? How do you resolve it?	Deadlocks occur when two transactions block each other. Resolve it using query optimization, indexes, or setting lock timeouts.
	What is the purpose of the <code>SAVEPOINT</code> command in SQL?	<code>SAVEPOINT</code> allows partial rollback of transactions by marking specific points within the transaction.
JavaScript	What is the difference between <code>localStorage</code> and <code>sessionStorage</code> in JavaScript?	<code>localStorage</code> persists data indefinitely, while <code>sessionStorage</code> persists until the browser/tab is closed.
	Explain the concept of <code>closures</code> in JavaScript.	Closures occur when a function retains access to its lexical scope, even when invoked outside that scope.
HTML/CSS	What are media queries , and how are they used in responsive web design?	Media queries apply styles conditionally based on screen size, resolution, or device characteristics (<code>@media</code>).
	Explain the difference between <code>flexbox</code> and <code>CSS grid</code> layout systems.	Flexbox is for one-dimensional layouts (row/column), while CSS Grid supports two-dimensional layouts (rows and columns).
Tools	What is Webpack , and why is it used in modern web development?	Webpack is a bundler that bundles JavaScript, CSS, and other assets, optimizing performance for production.
	What is the purpose of <code>npm</code> or <code>Yarn</code> in a web project?	<code>npm</code> and <code>Yarn</code> are package managers for JavaScript that help install, update, and manage project dependencies.

Let me know when you're ready for **Day 9!**



Day 9



Here's the set of questions for **Day 9**, focusing on **ASP.NET Core advanced concepts, threading in C#, SQL performance tuning, and front-end tools**.

Topic	Question	Answer Key/Hint
ASP.NET Core	What is Dependency Injection (DI) , and what are the three lifetimes in ASP.NET Core DI?	Singleton, Scoped, and Transient lifetimes.
	How do you implement global exception handling in ASP.NET Core?	Use a custom middleware or <code>UseExceptionHandler()</code> in <code>Startup.cs</code> .
	What is the difference between <code>IActionResult</code> and <code>ActionResult<T></code> in ASP.NET Core?	<code>IActionResult</code> is a base interface; <code>ActionResult<T></code> is a strongly-typed result.
	How can you enable Cross-Origin Resource Sharing (CORS) in an ASP.NET Core API?	Configure CORS in <code>Startup.cs</code> using <code>services.AddCors()</code> and <code>app.UseCors()</code> .
C#	What is the difference between thread and task in C#?	Threads are lower-level constructs; tasks are higher-level abstractions managed by the ThreadPool.
	Explain parallel programming and how it is implemented in C#.	Use <code>Parallel.For</code> , <code>Parallel.ForEach</code> , and <code>Task.WhenAll</code> for concurrent execution.
	What are <code>Semaphore</code> and <code>Mutex</code> in C#?	Both manage concurrency. <code>Semaphore</code> limits resource access for multiple threads; <code>Mutex</code> is used for exclusive access.
	What is a thread-safe collection ? Provide an example in C#.	Collections like <code>ConcurrentDictionary</code> and <code>BlockingCollection</code> ensure safe access in multithreaded scenarios.
SQL	What is the execution plan in SQL Server? How can it be viewed?	Execution plans show how queries are executed. Use <code>EXPLAIN</code> or the SQL Server Management Studio "Execution Plan" feature.
	Explain index fragmentation . How do you resolve it?	Fragmentation occurs when indexes become inefficient. Use <code>ALTER INDEX REBUILD</code> or <code>REORGANIZE</code> .
	What are covered indexes in SQL?	A covered index includes all columns needed for a query, reducing I/O operations.
	What are stored procedures , and how are they different from functions in SQL?	Stored procedures can perform DML operations and return multiple result sets, while functions must return a single value or table.
JavaScript	What is debouncing in JavaScript, and when is it used?	Debouncing delays function execution until after a specified time has passed since the last invocation. Used for search inputs.
	Explain the concept of hoisting in JavaScript.	Hoisting moves declarations (not initializations) to the top of their scope during compilation.
HTML/CSS	What is z-index in CSS? How does it work?	<code>z-index</code> controls the stacking order of elements with <code>position</code> values like <code>relative</code> or <code>absolute</code> .
	What is the difference between transitions and animations in CSS?	Transitions handle simple state changes, while animations use keyframes for complex, multi-step effects.
Tools	What is Docker , and how is it useful in ASP.NET Core development?	Docker creates containers to package and deploy applications consistently across different environments.
	Explain the purpose of GIT branching strategies like <code>GitFlow</code> .	GitFlow provides a structured workflow with branches like <code>master</code> , <code>develop</code> , <code>feature</code> , <code>hotfix</code> , and <code>release</code> .

Let me know when you're ready for **Day 10**, the final day!



Day 10



Here's the set of questions for **Day 10**, focusing on **ASP.NET Core Deployment, advanced C# concepts, database optimization**, and modern development practices.

Topic	Question	Answer Key/Hint
ASP.NET Core	What are the different ways to deploy an ASP.NET Core application?	IIS, Kestrel (self-hosted), Azure App Services, Docker containers, and Linux environments.
	How do you configure environment-specific settings in ASP.NET Core?	Use <code>`appsettings.{Environment}.json`</code> files and the <code>`IHostingEnvironment`</code> or <code>`IWebHostEnvironment`</code> interface.
	What is Health Check Middleware in ASP.NET Core?	It provides endpoints to check application health using <code>`services.AddHealthChecks()`</code> and <code>`app.UseHealthChecks()`</code> .
	Explain the difference between HTTP 1.1, HTTP/2, and HTTP/3 .	HTTP/2 supports multiplexing and binary framing, and HTTP/3 runs over QUIC for better speed and reliability.
C#	What are design patterns , and name a few commonly used patterns in C#.	Examples include Singleton, Factory, Repository, Observer, and Dependency Injection patterns.
	Explain the SOLID principles in C#.	SOLID stands for Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, and Dependency Inversion principles.
	What is the difference between abstract classes and interfaces in C#?	Abstract classes can include implementation; interfaces only define contracts (methods/properties) to be implemented.
	What are tuples in C#? Provide an example of their usage.	Tuples group multiple values into a single object, e.g., <code>`(int, string) result = (1, "Hello");`</code> .
SQL	What are CTEs (Common Table Expressions) ?	CTEs are temporary result sets used within a SQL query, defined using <code>`WITH`</code> . Useful for recursive queries.
	Explain normalization and its types in SQL.	Normalization organizes data into tables to reduce redundancy; includes 1NF, 2NF, 3NF, BCNF, etc.
	What are triggers in SQL? Provide an example use case.	Triggers execute automatically in response to events like <code>`INSERT`</code> , <code>`UPDATE`</code> , or <code>`DELETE`</code> . Useful for logging changes.
	What is query optimization , and how does the SQL execution plan help?	Query optimization improves query performance. Execution plans analyze how SQL queries are executed.
JavaScript	What are Web Workers in JavaScript, and how do they improve performance?	Web Workers run scripts in background threads, offloading work from the main thread for better performance.
	Explain the async/await flow with an example of error	<code>`async/await`</code> waits for promises; use <code>`try/catch`</code> to handle errors in asynchronous code.

Topic	Question	Answer Key/Hint
	handling.	
HTML/CSS	What are pseudo-classes and pseudo-elements in CSS? Provide examples.	Pseudo-classes style specific states (<code>:hover</code>), while pseudo-elements target parts of elements (<code>::before</code> , <code>::after</code>).
	What is the difference between inline-block and block elements?	<code>block</code> elements take the full width; <code>inline-block</code> respects content size but allows block-level styling.
Tools/Concepts	What is CI/CD , and how is it implemented in a .NET project?	CI/CD automates build, test, and deployment pipelines using tools like Azure DevOps, Jenkins, or GitHub Actions.
	How does Docker Compose help when working with microservices?	Docker Compose allows defining and running multi-container Docker applications using a single YAML file.
	What is Kubernetes , and why is it used in modern deployments?	Kubernetes automates container deployment, scaling, and management for microservices architectures.

Wrap-Up:

- This concludes the **10-day interview preparation series!**
- **Review:** Go through these questions and answers repeatedly. Practice explaining your answers confidently.
- **Next Steps:** If you want help with coding examples, real-world scenarios, or mock interviews, let me know!

Good luck with your interview preparation! Let me know how else I can assist.