Senior .NET Developer Interview Q&A Sheet

Candidate Role: Senior Full Stack .NET Developer **Projects:** Medicine/Pharmacy E-Commerce Platform, Delivery Tracking System

---

1️⃣HR / Behavioral Questions (20+ questions)

Q1: Tell me about yourself. A1: I am a Senior Full Stack .NET Developer with 5+ years of experience building scalable applications. I led the development of a Medicine/Pharmacy E-Commerce Platform and a Delivery Tracking System using ASP.NET Core, React.js, SQL Server, and Azure.

Q2: Why should we hire you? A2: I bring extensive experience in designing full-stack solutions, secure authentication, cloud deployment, and optimizing system performance to deliver business value.

Q3: Describe a challenging situation and how you resolved it. A3: Handling real-time inventory for thousands of SKUs was challenging. I optimized SQL queries, added indexing, and implemented caching, reducing errors and improving performance.

Q4: How do you prioritize tasks? A4: I break projects into modules, prioritize based on dependencies and business impact, and follow Agile methodology for timely delivery.

Q5: How do you handle conflicts in a team? A5: I encourage open communication, listen to all viewpoints, and focus on data-driven solutions to reach consensus.

Q6: Describe your biggest achievement in your previous projects. A6: Successfully built a Pharmacy E-Commerce Platform supporting thousands of SKUs with secure prescription management and real-time order tracking.

Q7: How do you stay updated with technology? A7: I follow Microsoft docs, attend webinars, complete online courses, and experiment with new .NET and cloud features.

Q8: What motivates you at work? A8: I am motivated by building scalable solutions that directly impact users and improve business efficiency.

Q9: Describe a situation where you took initiative. A9: Implemented centralized logging and error handling in the Delivery Tracking System, proactively identifying and resolving issues.

Q10: How do you manage stress during tight deadlines? A10: Prioritize critical tasks, communicate with stakeholders, and ensure team collaboration to meet deadlines without compromising quality.

(Additional HR questions: teamwork, leadership, learning, adaptability, career goals, communication, problem-solving, client interaction, decision-making, work ethics, handling failure, remote work, mentoring, project management.)

2️⃣Technical Questions (30+ questions)

Q1: Explain your role in the Medicine/Pharmacy E-Commerce Platform. A1: Designed and implemented RESTful APIs, JWT authentication, shopping cart, order workflows, prescription management, and deployed the platform on Azure SQL and App Service.

Q2: How did you implement JWT authentication and authorization? A2: JWT tokens carry role claims. APIs are secured using `[Authorize(Roles="RoleName")]` for Admin, Pharmacist, Customer, and Delivery Agent roles.

Q3: How did you design the database schema? A3: Created normalized tables for Users, Roles, Medicines, Categories, Orders, Prescriptions, Inventory, and DeliveryEvents, with indexing for performance.

Q4: How did you optimize performance for high-traffic endpoints? A4: Used indexing, query optimization, eager loading in EF Core, caching, and asynchronous programming.

Q5: How did you handle secure prescription uploads? A5: Validated file type and size, stored files in secure Azure Blob Storage, and encrypted sensitive information.

Q6: Explain your approach to RESTful API design. A6: Followed REST principles, used proper HTTP verbs, meaningful status codes, and documented APIs with Swagger.

Q7: How did you implement real-time delivery tracking? A7: Logged order status in DeliveryEvents table, exposed APIs for frontend polling/subscription, and indexed columns for fast queries.

Q8: How did you deploy your applications on Azure? A8: Deployed backend on Azure App Service, database on Azure SQL, configured CI/CD via Azure DevOps/GitHub Actions, monitored with Application Insights.

Q9: How did you handle exception handling and logging? A9: Centralized middleware for exception handling, structured logging with Serilog, monitored via Azure Application Insights.

Q10: How do you test your APIs? A10: Unit tests for business logic, integration tests for endpoints, Postman/Swagger manual testing, SQL Profiler for query validation.

Q11: How do you implement role-based access in .NET Core? A11: Used ASP.NET Core Identity, defined roles in database, assigned roles to users, and used `[Authorize(Roles="RoleName")]`.

Q12: Explain Entity Framework Core usage in your projects. A12: Used EF Core for ORM, wrote LINQ queries, used migrations for database updates, and optimized queries with eager/lazy loading.

Q13: How did you handle concurrency issues in order processing? A13: Implemented row versioning in SQL Server and handled DbUpdateConcurrencyException to prevent data conflicts.

Q14: How did you optimize SQL queries? A14: Used indexing, avoided N+1 queries, used joins instead of subqueries, and monitored execution plans.

Q15: How did you integrate React.js with ASP.NET Core Web API? A15: Built React frontend to consume RESTful APIs, used Axios for API calls, managed state with Redux, and ensured proper authentication headers.

Q16: How did you implement shopping cart functionality? A16: Created Cart table, managed items with Add/Update/Delete APIs, calculated totals server-side, and validated inventory availability.

Q17: How did you handle checkout and payment integration? A17: Designed order workflow APIs, validated cart and inventory, created orders, and integrated payment gateways using secure API calls.

Q18: How do you handle exception handling globally in .NET Core? A18: Implemented custom middleware to catch all exceptions, log details, and return standard error responses.

Q19: Explain how you implemented prescription validation. A19: Validated file type, size, and content pattern, checked against inventory rules, and linked prescriptions to orders.

Q20: How did you ensure API security? A20: JWT authentication, HTTPS enforcement, input validation, role-based access control, and logging of suspicious activity.

(Add more questions on async programming, caching, dependency injection, Azure services, scalability, performance tuning, security, Swagger documentation, database migrations, CI/CD, monitoring, logging, system design, and error handling.)

**✅ This document now contains 50+ HR and Technical questions with model answers tailored to Senior .NET projects (E-Commerce & Delivery Tracking System).