Dotnet Core MVC + Microservices Interview Cheat Sheet

1. .NET Core MVC - Key Concepts

- MVC = Model, View, Controller
- Model: Holds data, View: UI, Controller: Handles requests and returns view or JSON
- ViewBag/ViewData/TempData:
  - ViewBag: dynamic, per-request only
  - ViewData: dictionary, per-request
  - TempData: persists across redirects
- Filters: Authorization, Action, Result, Exception
- Routing: Conventional vs Attribute-based routing

2. Web API Essentials

- Attribute Routing: [HttpGet("route/{id}")]
- JWT Authentication: Add Authentication Middleware, validate token in controller
- Middleware: For logging, error handling, etc.
- Exception Handling: Use try-catch or custom middleware
- Dependency Injection: Register services in Startup.cs

3. Microservices

- Microservices: Small, independent services doing one job
- Communication: REST, gRPC, Message Queues (e.g., RabbitMQ)
- API Gateway: Centralized entry (e.g., Ocelot)
- DB per service: Each service manages its own data
- Authentication: Shared identity service or token-based
- Resilience: Polly (retry, circuit breaker)

4. EF Core & SQL

- EF Core: ORM for .NET Core
- Code-First: Define models -> Migrations -> Update DB
- Migrations: Add-Migration, Update-Database
- LINQ: .Where(), .Select(), .GroupBy()
- SQL: INNER JOIN, LEFT JOIN, Stored Procedure, Index

Dotnet Core MVC + Microservices Interview Cheat Sheet

5. C# Concepts

- Interface vs Abstract Class
- Virtual vs Override vs New keyword
- async/await for asynchronous programming
- SOLID Principles: Single Responsibility, Open-Closed, etc.

6. React + Redux Basics

- Redux Flow: Action -> Reducer -> Store -> Component
- Hooks: useState, useEffect
- Controlled Components: Handle form state via component
- Props drilling vs Context API