**1. How do you implement Dependency Injection in ASP.NET Core?**

**Answer: Dependency Injection (DI) is built into ASP.NET Core. We register services in Startup.cs using services.AddScoped(), AddTransient(), or AddSingleton(), and inject them through constructors into controllers or other classes.**

**2. Explain the Repository Pattern. How have you used it?**

**Answer: The Repository Pattern abstracts data access logic. It decouples the business logic from EF Core queries. I created interfaces like IUserRepository and implemented them using EF Core. This allowed easier unit testing and better separation of concerns.**

**3. How do you handle configuration settings in ASP.NET Core?**

**Answer: We use appsettings.json for configuration and bind values to POCO classes using IConfiguration. For secrets, User Secrets and Azure Key Vault can be used in production.**

**4. What is the difference between IHostedService and BackgroundService in .NET Core?**

**Answer:**

* **IHostedService is an interface to run background tasks.**
* **BackgroundService is an abstract class that implements IHostedService with a long-running ExecuteAsync() method. We use it for scheduled jobs or message queue listeners.**

**5. How do you secure sensitive API data in Web API?**

**Answer:**

* **Use HTTPS (SSL)**
* **Authenticate using OAuth2/JWT**
* **Apply input validation and output encoding**
* **Use DataProtection APIs for encryption**
* **Avoid exposing internal structure via verbose errors**

**6. What are the different return types from a Web API Controller?**

**Answer:**

* **IActionResult**
* **ActionResult<T>**
* **Specific types like Ok(), BadRequest(), NotFound(), etc.**
* **Custom objects**

**7. What is a DTO and why do we use it?**

**Answer: A Data Transfer Object (DTO) is used to shape data sent across layers, especially from API to client. It avoids exposing domain models and can reduce over-fetching of data.**

**8. What is the difference between FirstOrDefault and SingleOrDefault in LINQ?**

**Answer:**

* **FirstOrDefault: Returns the first matching record or null.**
* **SingleOrDefault: Expects exactly one record. Throws exception if multiple records exist.**

**9. How do you implement pagination in Web API?**

**Answer: Use query parameters like pageNumber and pageSize. Use Skip((pageNumber - 1) \* pageSize).Take(pageSize) in LINQ. Return metadata like total pages and current page.**

**10. What is Azure Key Vault and how have you used it?**

**Answer: Azure Key Vault stores secrets, keys, and certificates securely. We can access secrets using Managed Identity and configure apps to fetch values securely at runtime.**

**11. What is the difference between Azure App Service and Azure Function?**

**Answer:**

* **App Service: For hosting web apps with full control**
* **Azure Function: Event-driven, serverless compute; ideal for small units of logic**

**12. What is Entity Framework Core Migration?**

**Answer: Migrations allow EF Core to track schema changes. We use commands like Add-Migration, Update-Database to generate and apply SQL scripts that sync model and database.**

**13. How do you handle exception handling globally in ASP.NET Core?**

**Answer: Use UseExceptionHandler() middleware. Create a custom error-handling middleware or use filters like ExceptionFilterAttribute for MVC.**

**14. Explain the use of AutoMapper in your projects.**

**Answer: AutoMapper is used to map between DTOs and domain models. We define mapping profiles and inject IMapper to convert data automatically, reducing manual mapping code.**

**15. Code Challenge: Reverse a string in C#.**

**Answer:**

**public string ReverseString(string input)**

**{**

**char[] chars = input.ToCharArray();**

**Array.Reverse(chars);**

**return new string(chars);  
}**