**به‌نام خدا**

**تمرین مفاهیم ASP.NET Core**

**1 - Middleware**

**Middleware بخش‌هایی از برنامه هستند که در زنجیره پردازش درخواست‌ها قرار می‌گیرند و می‌توانند درخواست‌ها را قبل از رسیدن به مقصد بررسی، تغییر یا متوقف کنند.  
مثال: در سیستم مدیریت دانشجویان، یک middleware می‌تواند بررسی کند آیا کاربر وارد سیستم شده یا نه، و اگر نه، او را به صفحه ورود هدایت کند.**

**2 - Routing**

**Routing فرآیند هدایت درخواست‌های ورودی به متدهای مناسب در کنترلرهاست.  
مثال: اگر کاربر به مسیر /students/details/3 برود، Routing درخواست را به متدی هدایت می‌کند که اطلاعات دانشجوی با شناسه 3 را نمایش می‌دهد.**

**3 - Db Context**

**Db Context کلاس مرکزی ارتباط با پایگاه داده در Entity Framework است که امکان افزودن، خواندن، ویرایش و حذف داده‌ها را فراهم می‌کند.  
مثال: در سیستم مدیریت دانشجویان، کلاس Student Db Context می‌تواند شامل جداولی مانند Students، Courses و Enrollments باشد.**

**4 - Entity Framework (EF Core)**

**EF Core یک ORM (Object-Relational Mapper) است که اجازه می‌دهد با پایگاه داده بدون نوشتن دستورات SQL به صورت مستقیم کار کنیم.  
مثال: برای افزودن دانشجو به جدول، به‌جای SQL از کدهای C# استفاده می‌شود مثل context. Students. Add(student).**

**5 - MVC**

**MVC یک الگوی طراحی است که برنامه را به سه بخش Model، View و Controller تقسیم می‌کند.  
مثال: Model کلاس دانشجو را تعریف می‌کند، View فرم ثبت‌نام را نمایش می‌دهد و Controller ورودی‌های کاربر را پردازش می‌کند.**

**6 - Razor**

**Razor زبان ترکیبی HTML و C# است که در View ها برای تولید صفحات داینامیک استفاده می‌شود.  
مثال: در فرم نمایش اطلاعات دانشجو، می‌توان با @student.Name نام دانشجو را نمایش داد.**

**7 - View Model**

**View Model کلاس‌هایی هستند که برای نمایش اطلاعات ترکیبی از چند Model درView استفاده می‌شوند.  
مثال: برای نمایش مشخصات دانشجو و کلاس‌های ثبت‌نام شده، یک View Model می‌سازیم که شامل اطلاعات هر دو باشد.**

**8 - CRUD**

**CRUD عملیات پایه‌ای روی داده‌هاست: Create ، Read، Update، Delete.  
مثال: در سیستم مدیریت دانشجو، ثبت‌نام دانشجو (Create)، مشاهده لیست دانشجویان (Read)، ویرایش اطلاعات (Update) ، حذف دانشجو (Delete).**

**9 - Authentication**

**Authentication فرآیند شناسایی هویت کاربران است.  
مثال: دانشجو با وارد کردن نام کاربری و رمز عبور وارد سیستم می‌شود، و سیستم بررسی می‌کند که اطلاعات او صحیح است یا نه.**

**10 - Validation**

**Validation برای بررسی صحت داده‌های ورودی کاربران استفاده می‌شود.  
مثال: هنگام ثبت‌نام دانشجو، بررسی می‌شود که فیلد "ایمیل" به‌درستی پر شده باشد و شماره دانشجویی تکراری نباشد.**

**11 - Dependency Injection (DI)**

**DI مکانیزمی است که وابستگی‌های کلاس‌ها را به‌جای ساختن مستقیم، از بیرون به آن‌ها تزریق می‌کند.  
مثال: کلاس کنترلر دانشجو، به جای ساخت Db Context خودش، آن را به‌صورت خودکار از طریق DI دریافت می‌کند.**

**12 - Design Patterns**

**Design Patterns الگوهای استاندارد حل مسائل رایج در طراحی نرم‌افزار هستند.  
مثال: استفاده از الگوی Repository برای جداسازی منطق دیتابیس از منطق برنامه در سیستم مدیریت دانشجویان.**

**13 - Signal R**

**Signal R برای ارتباط بلادرنگ (real-time) بین سرور و کلاینت استفاده می‌شود.  
مثال: وقتی یک دانشجو به‌صورت آنلاین ثبت‌نام می‌کند، مدیر سیستم بلافاصله نوتیفیکیشن دریافت می‌کند.**

**14 - Caching**

**Caching نگهداری موقت اطلاعات در حافظه برای افزایش سرعت دسترسی به آن‌هاست.  
مثال: لیست کلاس‌ها که کمتر تغییر می‌کند، در cache ذخیره می‌شود تا در هر بار مراجعه نیاز به کوئری جدید نباشد.**