به نام خدا

**ASP**  
یه تکنولوژی قدیمیه از مایکروسافت که می‌ذاشت بتونی صفحات وب بسازی که محتواشون بسته به شرایط مختلف عوض می‌شه. مثلا کاربر وارد بشه یه چیزی ببینه، مهمون یه چیز دیگه. اون موقع‌ها با زبان‌هایی مثل VBScript یا JavaScript توی سرور کار می‌کردن.

**.NET**  
یه پلتفرم بزرگه از مایکروسافت که باهاش می‌تونی انواع برنامه‌ها رو بسازی؛ مثلا برنامه‌های ویندوزی، وب‌سایت، اپلیکیشن موبایل، یا برنامه‌هایی که روی فضای ابری (Cloud) اجرا می‌شن. خلاصه‌اش اینه که دات‌نت یه بستر همه‌کاره برای برنامه‌نویسیه.

**ASP.NET Core**  
نسل جدید و خیلی بهینه‌تر از ASP.NET هست. سبک‌تره، سریع‌تره و مهم‌تر از همه اینکه روی ویندوز، لینوکس و مک اجرا می‌شه. برای ساخت سایت‌های حرفه‌ای، API و حتی اپلیکیشن‌های واقعی که نیاز به سرعت و امنیت دارن، خیلی خوبه.

**Middleware**  
وقتی کاربر یه درخواستی به سایت می‌فرسته، مثلا می‌خواد وارد حسابش بشه، این درخواست باید از یه سری مراحل رد بشه. این مراحل همون Middlewareها هستن، که هرکدوم یه کاری می‌کنن. مثلا بررسی می‌کنن که کاربر لاگین کرده یا نه، یا اگه خطایی پیش بیاد، ثبتش می‌کنن.

**Web Routing**  
یه قابلیته که کمک می‌کنه وقتی کاربر یه آدرس خاص رو تو مرورگر می‌زنه (مثلاً /products/details/3)، برنامه بفهمه باید کدوم بخشش اجرا بشه. یعنی URL به صورت هوشمند به یه متد مشخص از کنترلر وصل می‌شه.

**Data Annotation**  
یه جور تگ یا مشخصه‌ست که روی فیلدهای مدل‌هامون می‌ذاریم تا یه سری قانون براش تعریف کنیم. مثلا بگیم این فیلد خالی نباشه، فقط ایمیل معتبر قبول کنه، یا مثلا طول متن بیشتر از ۲۰ کاراکتر نباشه. بیشتر برای اعتبارسنجی کاربرد داره.

**DbContext**  
قلب تپنده‌ی ارتباط با دیتابیسه! یه کلاسه که از طریقش می‌تونیم به جدول‌هامون دسترسی داشته باشیم، اطلاعات رو بخونیم، اضافه کنیم، یا حذف و ویرایش کنیم. توی EF Core با DbContext سروکار زیادی داریم.

**Connection String**  
یه رشته متنیه (مثل یه آدرس همراه با رمز و اطلاعات) که برنامه از طریق اون می‌فهمه باید به کدوم دیتابیس وصل بشه، روی چه سروریه، با چه یوزرنیمی، با چه رمزی. خلاصه اطلاعات لازم برای برقراری ارتباط با دیتابیس توشه.

**Dependency Injection**  
یه روش باحاله تو برنامه‌نویسی که باعث می‌شه اجزای مختلف برنامه به جای اینکه خودشون چیزایی مثل سرویس‌ها یا دیتابیس‌ها رو بسازن، اون‌ها رو از بیرون بگیرن. اینطوری همه‌چی منظم‌تر، قابل کنترل‌تر و راحت‌تر برای تست‌کردن می‌شه.

**Controller**  
توی معماری MVC، کنترلر مغز متفکره! یعنی وقتی کاربر یه کاری می‌کنه (مثلاً روی دکمه ثبت‌نام کلیک می‌کنه)، این کنترلره که می‌فهمه باید چیکار کنه و بعد نتیجه رو به صفحه نشون می‌ده.

**ORM**  
یه راهکار برای اینه که راحت‌تر با دیتابیس کار کنیم. به جای اینکه بخوایم کلی کوئری SQL بنویسیم، با کلاس‌ها و شی‌ءها کار می‌کنیم و خودش پشت صحنه تبدیلشون می‌کنه به دستوراتی که دیتابیس بفهمه.

**Entity Framework**  
یکی از معروف‌ترین ORMهای دنیای دات‌نت هست. باهاش می‌تونی خیلی راحت بدون اینکه بری دنبال نوشتن SQL، با داده‌ها کار کنی؛ بخونی، بنویسی، ویرایش و حذف کنی. همه‌چی شی‌گرا و راحت‌تره.

**Razor**  
یه زبونه که ترکیبی از HTML و C# هست. باهاش می‌تونی صفحه‌های وبی درست کنی که هم ظاهر داشته باشن (مثل HTML) هم بتونن کار انجام بدن (با کدهای C#). این باعث می‌شه بتونی محتوای داینامیک تولید کنی.

**Validation**  
یعنی اینکه مطمئن بشیم اطلاعاتی که کاربر وارد می‌کنه درست هستن. مثلا اینکه شماره تلفن واقعا عدد باشه، ایمیل فرمتش درست باشه یا فیلدی که الزامی هست خالی نباشه. این کار باعث می‌شه داده‌های اشتباه وارد سیستم نشن.

نویسنده اثر :

یاشارمحمدنژاد دانشجوی دانشگاه فنی میناب   
درس مهندسی نرم افزارِ استاد احمد زاده