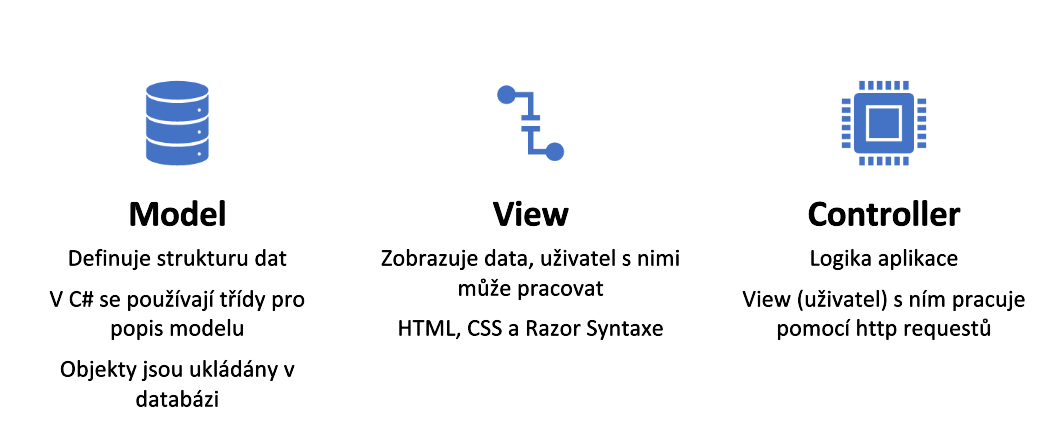
# 21 Návrh webové aplikace ASP.NET

## Razorpages

* Jednoduchý princip
* Kompaktní pro různé účely
* Jednoduchá logika směřována do jednoho místa
* Speciální syntax razor (html a v něm je c#)
* **Pagemodel** – připravuje data pro zobrazení stránky
* **Pagemodel** – třída s běžnými službami (databáze, session, cookies)
  + Tyto služby jsou předávány přes dependency injection (princip razor pages)
  + Tyto obecné třídy nadeklarovány ve startupu do dependency kontejneru

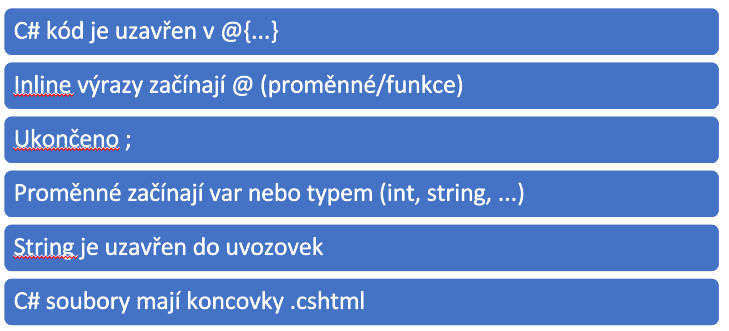
## MVC



## Page a PageModel

* **Pages** – to je ta složka Pages 😊
* **@page**
* Ve view je možno vložit inline PageModel
* Page o tomto modelu ví (@model)
* **PageModel** – víceúčelový
* Obsahuje data vykreslovaná do View, řeší zpracování požadavků od klienta a obvykle má vazbu na další služby

## Razor syntaxe



## Startup.cs

* Od .NET 6 spojen s Program.cs
* ConfigureServices a Configure se volají při startu aplikace
* Obsah obrázku text

  Popis byl vytvořen automaticky

### ConfigureServices

* Volitelná metoda, konfiguruje služby aplikace
* AddService, AppDbContext, AddRazorPages, AddDefaultIdentity
* Znovu použitelná komponenta, která zprostředkovává funkcionalitu aplikace
* Služby jsou registrovány v ConfigureServices a používány v aplikaci pomocí Dependency injection nebo Application Service

### Configure

Používá se k určení, jak aplikace reaguje na požadavky http

## Služba

* ASP.NET Core používá dependency
* Aby DI vědělo, jak rozpoznat závislosti, musíme vše nejprve nakonfigurovat
* Obsah obrázku text

  Popis byl vytvořen automaticky
* Je možné taky vytvořit novou službu, která může obsahovat logiku aplikace (metody, práce s databází)

## Databáze

* Permanentní ukládání dat pro všelijaké účely
* Mnoho příkladů implementace (connection string)
* **Model** – obsahuje data a chování modelu, co třída to entita
* **Relace** – V relační databázi je každá entita uložena ve své vlastní oddělené tabulce identifikována unikátním klíčem
* Relace identifikovány pomocí klíčů (FK)
* **Connection string** – obsahuje informace o zdroji dat
* 1:N, 1:1, M:N
* **DBContext** – reprezentuje relaci s DB a zprostředkovává API pro komunikaci s DB a připojení k DB
* **DbSet** – třída pro kolekci pro danou entitu v databázi v rámci modelu, brána k databázovým operacím proti entitě
  + Přidávají se jako vlastnosti do DbContext a jsou standartně mapovány na databázové tabulky, které přebírají název vlastnosti DbSet<TEntity>

## EntityFramework

* Obsah obrázku text

  Popis byl vytvořen automaticky

## Identity

* Rozhraní API, které podporuje přihlašování uživatelského rozhraní
* Sada balíků, hotové řešení umožňující přihlašování uživatel
* Staví na EF, rozšiřuje DbContext o další tabulky
* Umožňuje scaffolding pro identity
* Spravuje uživatele, hesla, data profilu, role, deklarace identity, tokeny, potvrzení emailu atd.
* Uživatel si může vytvořit účet nebo použít externího poskytovatele přihlášení