

Using the current DbContext and the connection string examines the db it's pointing to as well as all the migrations its pointing to and apply any migrations to the db that weren't previously there.

Index.cshtml   
  
<!--

<a href="/home/details/@r.Id">Details</a>

@Html.ActionLink("Go", "Details", new {id = r.Id})

-->

\_\_\_\_\_\_\_\_\_\_\_\_\_

Mvc controllers are the best for developing web services, http based services rendering json etc.

Razor Pages for creating HTML

Partial Views Use cases - Modular code

@Html.PArtial("\_Summary", restaurant)

View Component - render it anywhere on the page. (Mini mvc request - not dependant on the parent)

Reusable bit of UI

Individual User Accounts in-app

Asp.net identity will store user names and passwords and other user information in a local sql server database, this is configurable to any database connection string

Connect to an existing store in the cloud. Create logins instead of storing user names and passwords in a sql server database it'll

use open ID connect, OIDC, uses Azure AD B2C application (Building an application for your company with logins but without the responsibility of managing accounts or storing)  
  
Managed with azure - also supports 3rd party logins Twitter, Facebook, Google etc   
  
purchase SSL certificate

ASP.NET Core Authentication and Authorization - An overview of ASP.NET Identity

UserManager / SignInManager are provided by ASP.NET Identity

OIDC - Open Identity DC - Register with the application provider -

Process for web application that renders HTML on the server - if you're building an API, service or native app for a device the OIDC flow changes

Url specified by our identity provider, in the example the login.microsost.com azure directory  
Find any other like github in their documentation

client id