# View Engine Razor:

Razor is a popular view engine used in ASP.NET for rendering HTML views with embedded C# code. It is commonly used with ASP.NET MVC and ASP.NET Core MVC applications.

Razor is a simple, templating language that allows you to write dynamic content in your views. It uses the .cshtml file extension (for C# HTML files) and combines HTML and C# code to render dynamic content on the web page.

How Razor Works:

Simplicity: Razor syntax is lightweight and avoids the need for traditional XML-style markup like other templating engines (e.g., JSP or ASPX). Razor’s syntax is designed to be simple, readable, and intuitive.

Embedding C# Code in HTML: Razor allows you to embed C# code directly into HTML using the @ symbol.

For example: `<h1>@Model.Title</h1>`

This would output the value of the Title property from the Model object.

Control Flow: Razor supports C# control structures like if, for, and foreach to dynamically generate content. Example:

```
@foreach(var item in Model.Items)

{

   <p>@item.Name</p>

}
```

Code Blocks: Razor provides several ways to work with code:

@{ ... } for code blocks:

```
@{
    var greeting = "Hello";
}
<h1>@greeting, World!</h1>
```

@ for inline expressions:

```
<p>The current time is @DateTime.Now</p>
```

Layout and Partial Views: Razor makes it easy to implement layouts and partial views. A layout is a template for your entire site, while partial views are reusable chunks of HTML.

Example:

```
@RenderBody()  <!-- This placeholder gets replaced with content from views -->
```

Automatic HTML Encoding: Razor automatically encodes content to prevent XSS attacks. For example:

```
<p>@Model.UserInput</p>
```

If UserInput contains HTML or JavaScript, Razor ensures it is encoded, making it safe to display.