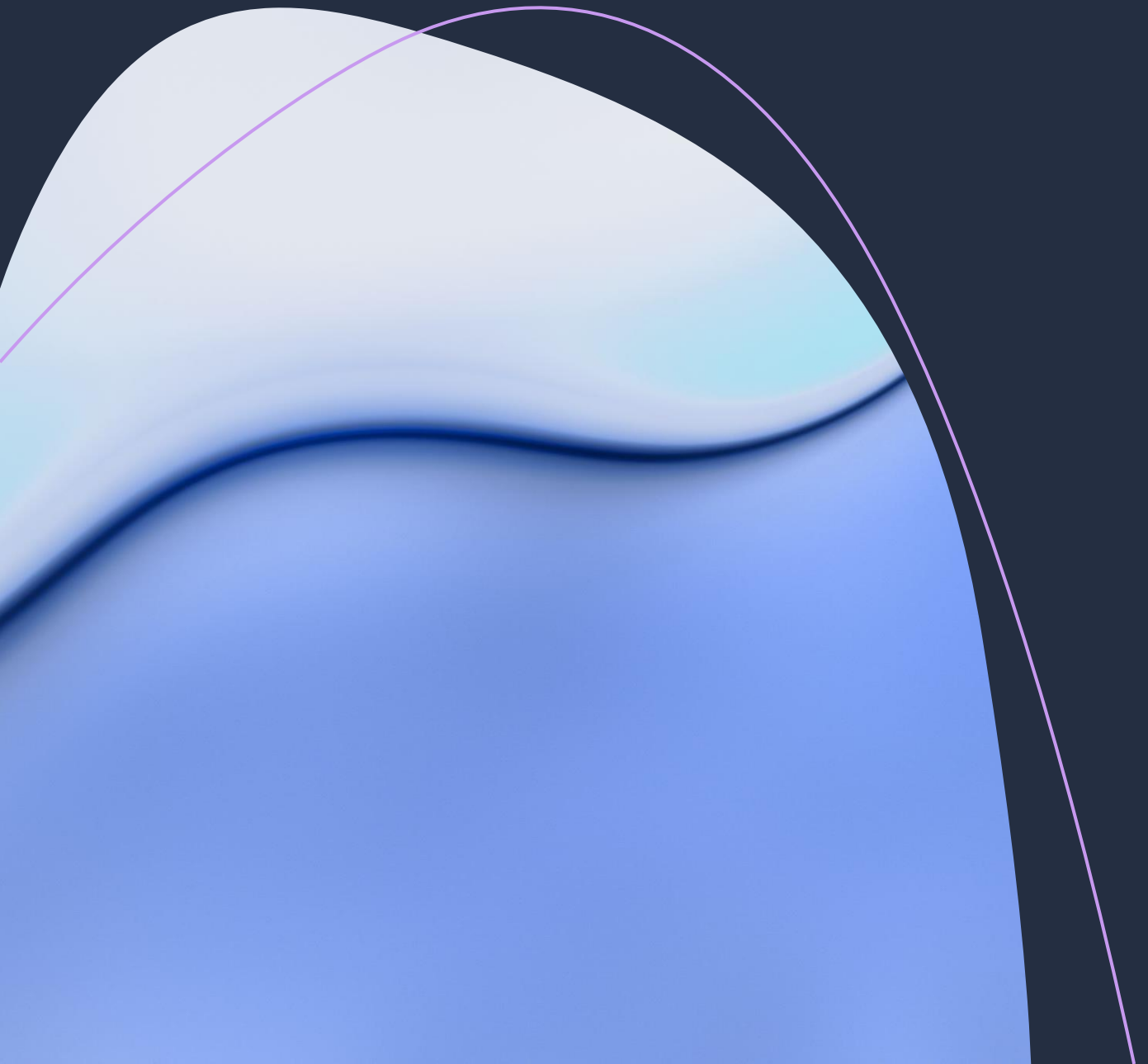

Session 1 - Introduction to Razor and Blazor

Enhancing Web Development with C# and Razor Syntax

Razor Overview

- ASP.NET MVC View Component
- Introduced in March 2009
- Combines Razor markup, C#, and HTML
- Used for dynamic web content
- Commonly used in ASP.NET MVC Views

How Razor Works

- Razor page processed on the server
- Sent to the client as a single HTML page
- Refreshing the page re-processes on the server
- Sent back to the client

Blazor

- Free open-source web framework
- Uses C#, HTML, and Razor syntax
- Replaces JavaScript for web UIs

Key Features of Blazor

- Create rich interactive UIs with C#
- Uses Razor Components
- Share logic between server-side and client-side apps
- Wide browser support, including mobile
- Utilizes WebAssembly and SignalR

Razor Components

- Components are the building blocks
- Define flexible UI rendering logic
- Handle user events
- Can be nested and reused
- Shareable as Razor class libraries or NuGet packages

Razor vs. Blazor

- Why Razor and Blazor Matter:
  - Razor and Blazor represent powerful tools in modern web development.
  - Razor, as a templating engine, simplifies the creation of dynamic web content by combining Razor markup, C#, and HTML.
  - Blazor takes it a step further by allowing developers to create interactive web UIs with C# and Razor syntax, reducing the reliance on JavaScript.