By Claude Nikula
info@mesquiteit.com
Company Phone: +1 (573) 234-6081
Direct line: +1 (858) 395-0571

https://mesquiteit.com

Choosing the Right Microsoft Technology for Your Application

When deciding between **Blazor Server**, **Blazor WebAssembly**, or **ASP.NET Core MVC/Razor Pages**, you should evaluate them based on performance, scalability, complexity, interactivity, hosting requirements, and user experience.

Technology Breakdown

Technology	Pros	Cons	Best Suited For
Blazor Server	- Faster initial load- Smaller client-side footprint- Access to .NET backend directly- Centralized security model	- Requires constant server connection (SignalR)- Higher latency for UI interactions- Increased server load	- Internal business apps- Dashboard/admin panels- Apps with complex backend interactions
Blazor WebAssembly	- Runs fully in the browser (no server round trips for UI updates)- Can work offline (PWA capabilities)- Lower server load	- Larger initial download size (can be several MBs)- Limited by browser sandbox (e.g., no direct database access)- Requires API for backend interactions	- Apps with rich UI that need to run offline- Single Page Applications (SPA)- Apps with minimal server interaction
ASP.NET Core MVC/Razor Pages	- Well-established architecture- Server-rendered pages (good for SEO)- Lower client-side resource usage	- Requires full page reloads (unless using AJAX or HTMX)- Less interactive compared to Blazor	- Traditional websites- Content-heavy applications- Public-facing sites (SEO-critical)

Key Questions to Ask

1. Application Type

- Is it an **internal** or **external** application?

By Claude Nikula
info@mesquiteit.com
Company Phone: +1 (573) 234-6081
Direct line: +1 (858) 395-0571

https://mesquiteit.com

  - **Internal** → Blazor Server or MVC/Razor Pages
  - **External** → Blazor WebAssembly or ASP.NET Core MVC
- Is **SEO important**?
  - Yes → ASP.NET Core MVC/Razor Pages
  - No → Blazor WebAssembly or Blazor Server

2. Performance & Scalability

- Do I need **real-time updates** (e.g., dashboards, notifications)?
  - Yes → Blazor Server (uses SignalR)
  - No → Blazor WebAssembly (if offline is needed) or ASP.NET Core MVC
- Do I expect **high concurrent users**?
  - Yes → Blazor WebAssembly or ASP.NET Core MVC (less server load)
  - No → Blazor Server (if real-time interaction is needed)

3. Deployment & Hosting

- Where will the application be hosted?
  - **Limited backend hosting?** → Blazor WebAssembly (served as static files)
  - **Full backend available?** → Blazor Server or ASP.NET Core
- Will the app need **offline support**?
  - Yes → Blazor WebAssembly (supports Progressive Web Apps)
  - No → Any option works

4. Security & Data Handling

- Does the app process **sensitive data**?
  - Yes → Blazor Server (more secure backend processing)
  - No → Blazor WebAssembly (ensure secure API calls)

By Claude Nikula
info@mesquiteit.com
Company Phone: +1 (573) 234-6081
Direct line: +1 (858) 395-0571

https://mesquiteit.com

- Will users need **authentication & authorization**?
    - Blazor Server: Easier to implement as everything runs on the server
    - Blazor WebAssembly: Requires API authentication

5. Development & Maintenance

- Do I have **C# developers**?
    - Yes → Any option works
    - No → Consider ASP.NET Core MVC (easier to onboard frontend developers)
- How **complex is the UI**?
    - Simple UI with forms → Blazor Server or ASP.NET Core MVC
    - Complex UI with heavy interactivity → Blazor WebAssembly

---

Summary: When to Choose What?

Scenario	**Recommended Technology**
Internal business dashboard	**Blazor Server**
Public-facing website (SEO matters)	**ASP.NET Core MVC/Razor Pages**
Highly interactive SPA	**Blazor WebAssembly**
High number of concurrent users	**Blazor WebAssembly / ASP.NET Core MVC**
Low latency, fast interactions	**Blazor WebAssembly**
Limited backend hosting	**Blazor WebAssembly**
Real-time updates	**Blazor Server**

By Claude Nikula
info@mesquiteit.com
Company Phone: +1 (573) 234-6081
Direct line: +1 (858) 395-0571

https://mesquiteit.com

Here's a **checklist** to help you systematically determine the best Microsoft technology for your application.

✅ Technology Selection Checklist

1. Application Type

- Is it an internal business app? → **Blazor Server or ASP.NET Core MVC**
- Is it a public-facing website? → **ASP.NET Core MVC/Razor Pages**
- Is it a Single Page Application (SPA) with rich UI? → **Blazor WebAssembly**
- Is the app data-intensive (e.g., reports, dashboards)? → **Blazor Server or WebAssembly**

---

2. Performance & Scalability

- Does the app need to support a high number of concurrent users?
  - Yes → **Blazor WebAssembly or ASP.NET Core MVC**
  - No → **Blazor Server is acceptable**
- Do you require **real-time updates** (e.g., live dashboards, chat, notifications)?
  - Yes → **Blazor Server**
  - No → **Blazor WebAssembly or ASP.NET Core MVC**
- Does the app have minimal UI interactions?
  - Yes → **ASP.NET Core MVC/Razor Pages**
  - No → **Blazor Server or Blazor WebAssembly**

---

3. Deployment & Hosting

By Claude Nikula
info@mesquiteit.com
Company Phone: +1 (573) 234-6081
Direct line: +1 (858) 395-0571

https://mesquiteit.com

- Will the application run **on a low-resource server** (e.g., cloud hosting with limited backend processing)?
    - Yes → **Blazor WebAssembly (Static hosting)**
    - No → **Blazor Server or ASP.NET Core MVC**
- Does the application need **offline functionality**?
    - Yes → **Blazor WebAssembly**
    - No → Any option works

---

4. Security & Data Handling

- Will the application process **sensitive or confidential data**?
    - Yes → **Blazor Server or ASP.NET Core MVC (Keeps data on the server)**
    - No → **Blazor WebAssembly (Ensure secure API calls)**
- Will the app interact **directly with a database**?
    - Yes → **Blazor Server or ASP.NET Core MVC**
    - No → **Blazor WebAssembly (Needs an API for data access)**
- Does the app require **complex authentication/authorization**?
    - Yes → **Blazor Server (easier to manage auth centrally)**
    - No → **Blazor WebAssembly or ASP.NET Core MVC**

---

5. User Experience & Interactivity

- Does the app require **instant page loads with minimal wait time**?
    - Yes → **ASP.NET Core MVC or Blazor Server**
    - No → **Blazor WebAssembly (Initial load is heavier)**
- Does the app require **smooth client-side interactivity without delays**?

By Claude Nikula
info@mesquiteit.com
Company Phone: +1 (573) 234-6081
Direct line: +1 (858) 395-0571

https://mesquiteit.com

  - Yes → **Blazor WebAssembly**
  - No → **Blazor Server**
- Is **SEO (Search Engine Optimization) important**?
  - Yes → **ASP.NET Core MVC/Razor Pages**
  - No → **Blazor WebAssembly or Blazor Server**

---

6. Development & Maintenance

- Do you want to **keep everything in C# (minimal JavaScript)?**
  - Yes → **Blazor (Server or WebAssembly)**
  - No → **ASP.NET Core MVC (More frontend flexibility)**
- Will frontend developers work on the project?
  - Yes → **ASP.NET Core MVC (Easier for frontend devs)**
  - No → **Blazor (More C#-centric)**
- Do you need an **established and well-documented architecture**?
  - Yes → **ASP.NET Core MVC**
  - No → **Blazor (Newer, evolving technology)**

Decision Guide

- ✅ **If you checked mostly ASP.NET Core MVC boxes → Use ASP.NET Core MVC/Razor Pages**
- ✅ **If you checked mostly Blazor Server boxes → Use Blazor Server**
- ✅ **If you checked mostly Blazor WebAssembly boxes → Use Blazor WebAssembly**