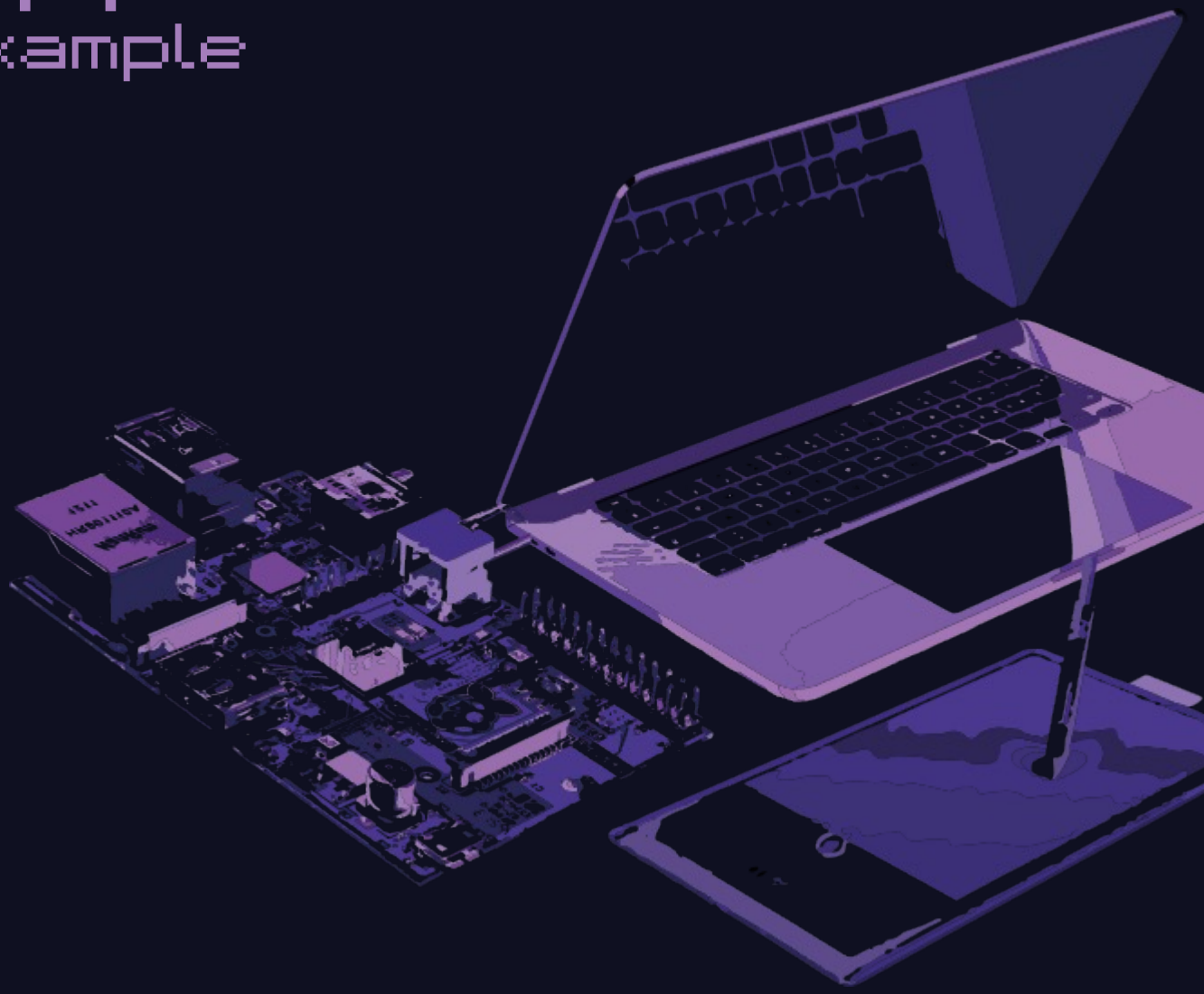
Modern Web Apps

Using Blazor as an Example

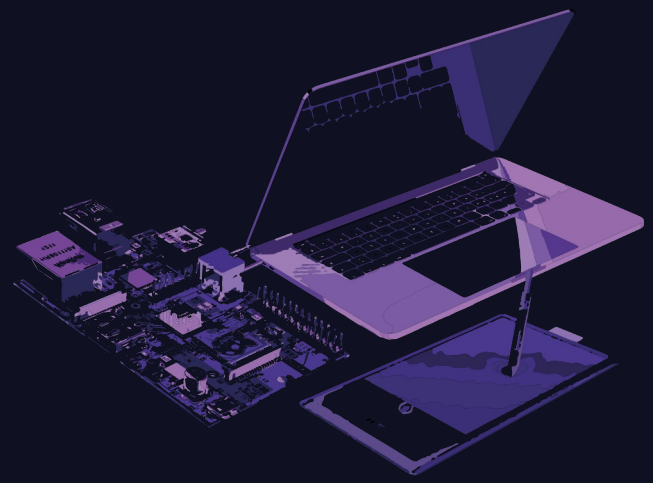


RGU
COMPUTING
<society>

What is Blazor?

- A web app framework in C#.
- Renders as HTML and CSS.
- Can be run fully client-side or completely pre-rendered on the server.
- Relatively young - released in 2018 and continually evolving.

.NET Blazor



RGU COMPUTING <society>

Why Blazor?

- You might already be familiar with C# or prefer strongly-typed languages.
- Write your UI in the same language as your backend logic.
- Can share common logic between front and back ends, encourages common behaviours and simplifies development.
- You can interface with the thriving .NET ecosystem of libraries available on NuGet.

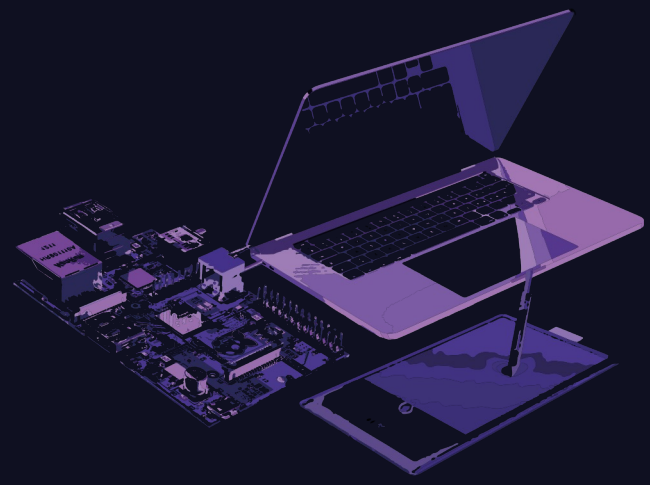
RGU COMPUTING <society>

Why Not Blazor?

- You might not be familiar with languages like C#.
- You don't want to worry about the complexities of a backend/server logic. (i.e. client-only application).
- Uncomfortable with larger boilerplate templates (there's a lot of setup)
- Struggle with the idea of dependency injection (will mention this shortly).

If you're not a fan of these options, I'll touch on some alternatives at the end!

RGU COMPUTING <society>



Why do I Use Blazor?

- I develop Blazor apps professionally, and such am very familiar with how it works.
- I enjoy developing in C#, and the way Blazor works makes sense to me personally.
- It's abundance of libraries to perform common tasks simplifies many angles of development.
- I do not enjoy writing JavaScript applications with non-JavaScript backends - unable to share logic/code easily.
- This presentation is not to convince you to use Blazor, but to offer it as an option and to show why I would use it.

RGU COMPUTING <society>

Components!

- Blazor pages are primarily composed of, well, components.
- These are bundles of HTML and behaviours, which can be reused to build up a webpage.
- You can also find libraries which have big bundles of components ready for reuse. More on this shortly.
- This idea is not unique to Blazor - most modern UI frameworks have some concept of components.

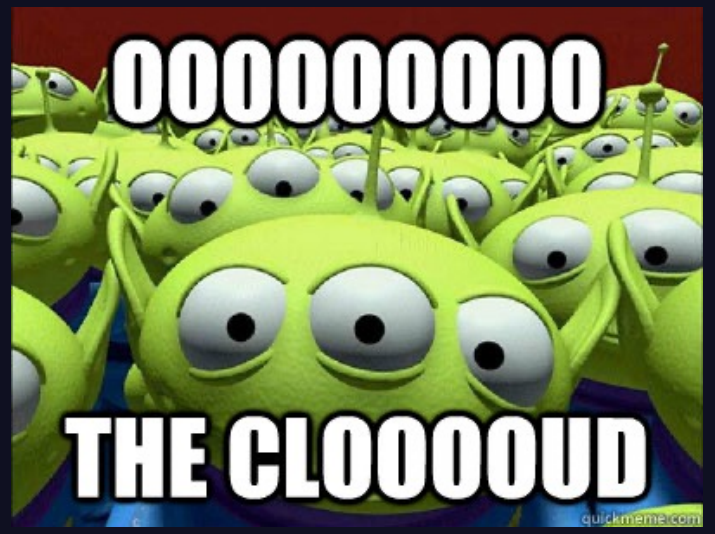
RGU COMPUTING <society>

Interactivity Types

- Static Server - you likely don't want this. It generates once and cannot be dynamic.
- Interactive Server - Server-side rendered, but interactions are sent to the server to update the contents dynamically.
- Interactive WebAssembly - All code in the client project is run in the browser, you will want to use APIs to access your back-end system instead of calling it in your components.

My recommendation for the hack would be to use Interactive Server, it's likely the easiest way to begin. If you need help setting this up, come and find me!

RGU COMPUTING <society>



Name	Description	Render location	Interactive
Static Server	Static server-side rendering (static SSR)	Server	✗
Interactive Server	Interactive server-side rendering (interactive SSR) using Blazor Server.	Server	✓
Interactive WebAssembly	Client-side rendering (CSR) using Blazor WebAssembly†.	Client	✓
Interactive Auto	Interactive SSR using Blazor Server initially and then CSR on subsequent visits after the Blazor bundle is downloaded.	Server, then client	✓

Dependency Injection

- This is a tricky topic when working with ASP.NET/Blazor.
- Essentially, this is a programming pattern that allows you to build small units of code (services) and inject them into other areas of your program without having to manually pass them around.
- I will provide a quick demo of this shortly and am happy to go into more depth if there is interest.

I highly recommend that you check out Microsofts guide to DI:

http://tinyurl.com/2p9z53b3

RGU COMPUTING <society>

Getting Started

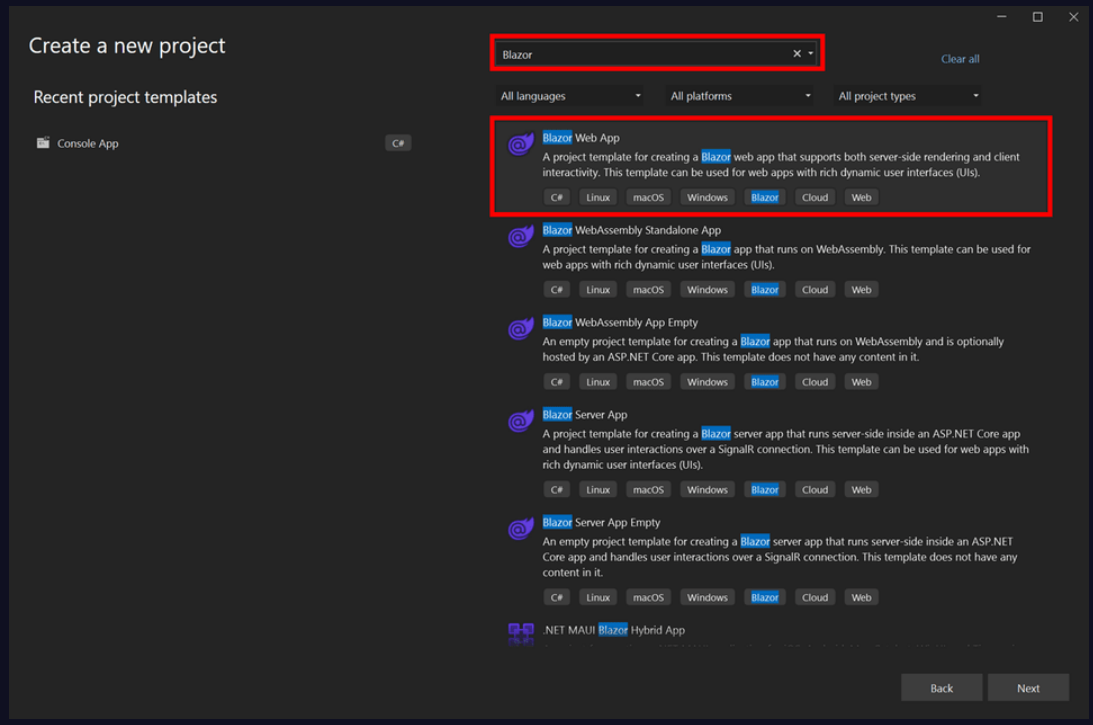
You'll need to install the following:

- A C# IDE: Usually Visual Studio or Jetbrains Rider (recommended). VS Code works too with the plugin.
- .NET 8 SDK (if not installed by your IDE)

RGU COMPUTING <society>

Project Setup

- There are many ways to launch a Blazor project, dotnet CLI (check MS documentation), Visual Studio templates, GitHub template repositories, and many other ways - have a dig around!
- For the following examples, I used a template for the component library, MudBlazor.




```
dotnet new install MudBlazor.Templates
dotnet new mudblazor --interactivity Server --all-interactive --name <Your Project Name Here>
```

RGU COMPUTING <society>

In Practice

- We're going to be using an anonymised version of the Eventbrite registration information.
- I'm going to demonstrate how to parse this data and display it in an application with a data grid and some charts.

The Data:

```
17/01/2024 14:39,Undergraduate,N/A,TRUE,4th Year
17/01/2024 14:39,Undergraduate,N/A,TRUE,4th Year
17/01/2024 14:39,Undergraduate,N/A,TRUE,4th Year
18/01/2024 09:59,Undergraduate,N/A,TRUE,2nd Year
18/01/2024 10:01,Undergraduate,N/A,TRUE,1st Year
18/01/2024 10:01,Undergraduate,N/A,TRUE,1st Year
18/01/2024 12:13,Other,N/A,TRUE,Unspecified
19/01/2024 12:25,Graduate,Vegetarian,TRUE,Unspecified
22/01/2024 09:42,Other,Gluten-free,TRUE,Unspecified
22/01/2024 17:14,Graduate,Vegetarian,TRUE,Unspecified
23/01/2024 11:08,Undergraduate,N/A,TRUE,2nd Year
23/01/2024 19:45,Undergraduate,N/A,TRUE,Unspecified
24/01/2024 11:52,Undergraduate,N/A,TRUE,2nd Year
24/01/2024 14:14,Undergraduate,N/A,TRUE,3rd Year
24/01/2024 15:07,Undergraduate,N/A,FALSE,2nd Year
25/01/2024 12:24,Undergraduate,N/A,TRUE,3rd Year
25/01/2024 12:27,Graduate,Vegetarian,TRUE,Unspecified
```

RGU COMPUTING <society>

Live Demo Time!

Download My Example

If you wish to view this example yourself, you can check out the following GitHub repo:

https://github.com/RGUCompSoc/RGUHack2024-BlazorExample/

RGU COMPUTING <society>

Other Frameworks

Not into C#/Blazor - Try one of these!

- React (Pairs well with something like Next.JS)
- Vue (Jordan has experience with this!)
- E.JS with Express in NodeJS
- Good old-fashioned HTML, CSS & JS!
- Flutter - niche, but can be used for web, desktop and mobile!

RGU COMPUTING <society>

Useful Resources

I've uploaded these to the GitHub Repository as well:

Useful Resources

These are some useful resources if you intend on working with Blazor during the hack. Feel free to find Reece at the hack if you have more questions as you go!

- https://dotnet.microsoft.com/en-us/apps/aspnet/web-apps/blazor - Microsoft's introduction to Blazor.
- https://dotnet.microsoft.com/en-us/learn/aspnet/blazor-tutorial/intro - Blazor introductory tutorial.
- https://learn.microsoft.com/en-us/dotnet/core/extensions/dependency-injection - Documentation for Dependency Injection
- https://learn.microsoft.com/en-us/aspnet/entity-framework - Entity Framework; you can use this to write database queries in Linq, consider using Sqlite or something instead of SQL Server for ease of setup during the hack.
- https://www.mudblazor.com/ - The component library used in this example
- https://antblazor.com/ - Another component library
- https://www.fluentui-blazor.net/ - Another component library

Alternatives

If you don't fancy using Blazor for you app, you can always consider these alternatives:

- https://ejs.co/ - Taught during RGU classes
- https://vuejs.org/ - Jordan will be speaking about this one!
- https://react.dev/ - consider mixing with something like https://nextjs.org/
- https://flutter.dev/ - Modern Web, Mobile and Desktop Applications!
- HTML/CSS/JS - classic style!
- Anything else you may already know!

RGU COMPUTING <society>

Feel free to ask me any questions now, or around and about during the hack!