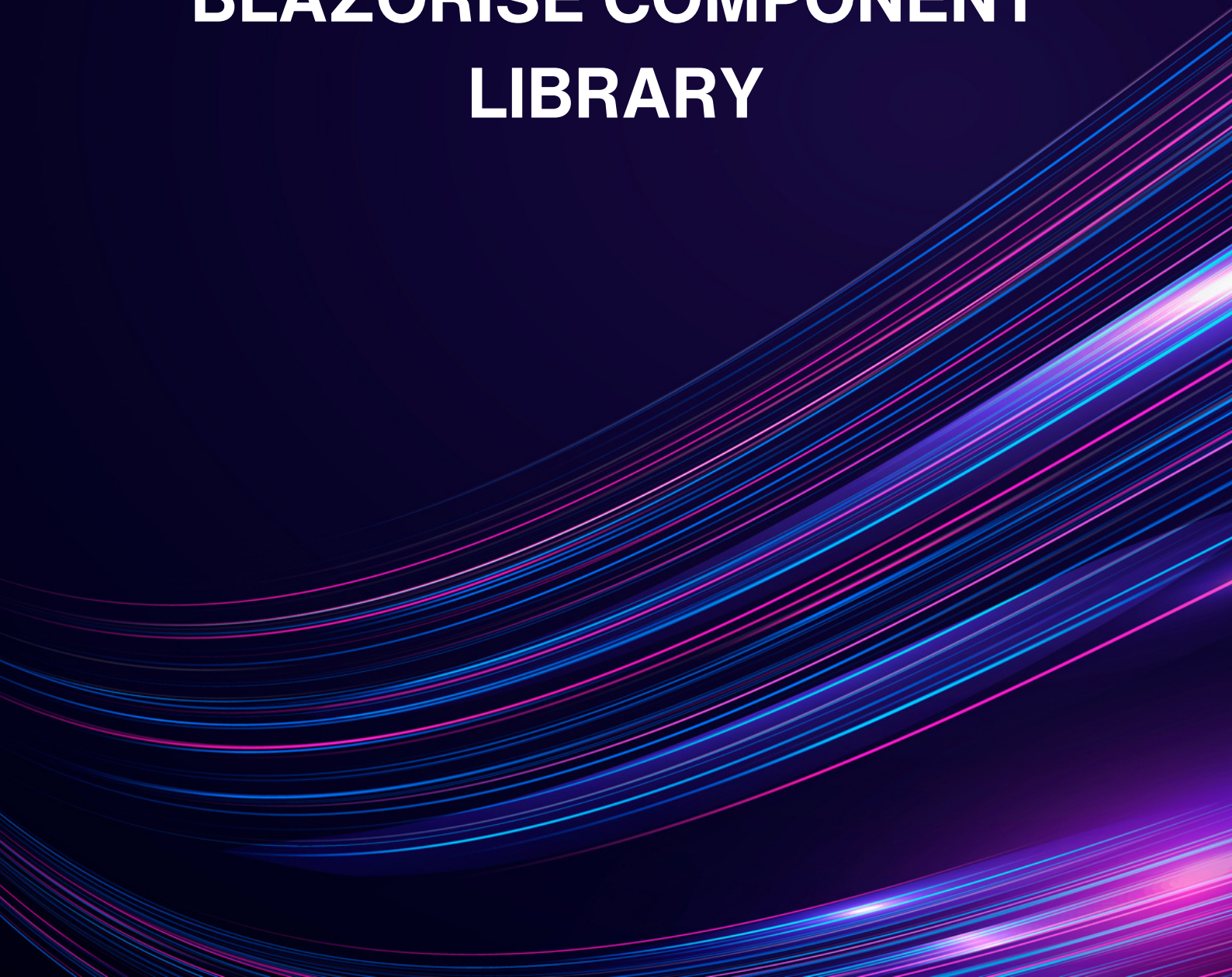




BLAZORISE COMPONENT LIBRARY



WHO ARE WE?

Blazorise, an *open-source* system, has evolved into a comprehensive **Blazor UI component library** offering *state-of-the-art* solutions for **startups and enterprises**.

The Blazorise team is **committed** to staying ahead of the curve by **keeping up** with the **latest technological developments** daily.

Our **main goal** is to raise the bar for **web development**, pushing the **industry forward** and encouraging others to **evolve** and **change** their thinking.

We aim to **identify** and promote **better** alternatives whenever they are **available** and help shape the future of **web development in an impactful way**.

WHAT DO WE DO?

Build components based on **blazor** technology that makes your web applications **easier** to build, **maintain** and cost effective in a long-run.

By utilizing pre-built components, you can **speed up** your coding process and spend more time developing the **functionality** of your application.

Instead of **spending hours** writing code for common elements, you can **focus** on the more critical aspects of your project, resulting in a **higher-quality** end product.

WHAT DO WE OFFER?

Multiple CSS framework

Blazorise is highly flexible and extensible. It works with different CSS frameworks, like ***Bootstrap, Tailwind, or Bulma***, making it an excellent choice for developers looking to build web applications quickly and efficiently.

Simple Syntax

With Blazorise, simple syntax without prefixes means that the framework's syntax does not require a prefix before each element, such as ***LibButton***, commonly used in other frameworks. The goal is to be: ***Easy*** to write, read and to be ***extenable***.

Wide range of components

Blazorise offers many ***premium*** web components that meet various web development needs. Every component is carefully crafted to provide the best possible user experience, with ***performance*** and responsiveness at the forefront.

Get components like...

DataGrid, DatePicker, Button, Modal, Autocomplete, Chart components and **many more...**

WHY WOULD YOU PICK US?

Blazorise's development process is:

- **efficient** and **effective**,
- saving you time and **money** in the long run
- no need to worry about **bloat**
- We create the code **without** unnecessary **extras**

This approach ensures your website or application runs **smoothly** and **quickly** without excess baggage

COMMUNITY DRIVEN

Community members are actively **contributing** to the development of **Blazorise** by reporting bugs, proposing **new features**, and even contributing code to the framework.

This community-driven approach has helped Blazorise overgrow in **popularity** and functionality. The team constantly explores new ways to improve the **framework** and make it more **versatile** and powerful.

OUR LATEST NEWS

Blazorise v1.5 development and FLUENT UI

- We brought Fluent UI 2 in v1.5
- New Toast component
- Progress bar
- Accordion components
- Font Awesome update
- AntDesign update
- And so much more...

HOW DID WE IMPLEMENT FLUENT UI 2?

PROBLEM

In the current version of Blazorise (v1.5), changing theme settings like those available in other Blazorise CSS providers is not possible.

This limitation was a deliberate choice while developing the new Fluent provider, primarily due to time constraints. The goal was to avoid delaying the release of Blazorise 1.5 and to allocate sufficient time to properly implement the Fluent theme generator in future versions of Blazorise.

SOLUTION(S)

We needed to override Fluent design tokens and it required a few steps.

- ➔ OPEN REACT FLUENT THEME DESIGNER**
- ➔ EXPORT THE THEME SETTINGS**
- ➔ COPY DESIGN TOKENS**
- ➔ APPLY RULES TO YOUR PROJECT CSS**
- ➔ INCLUDE THE CSS**

This way our users can still customize their Blazorise based projects with the Fluent design system, with a more coherent and visually appealing UI.

In future releases, we plan to fully implement Fluent UI without any constraints, as our user base is growing and our team is expanding. This will ensure that we can still provide our users with a smooth and consistent experience in their web app development.

