MudBlazor

* Material UI for Blazor WASM
* Problems:
  + Configuration problems 🡪 getting started of the official MudBlazor site did not work:
    - “MudBlazor could not be found”
* Problem solutions:
  + By downgrading MudBlazor to version 5 the “MudBlazor could not be found” error could be solved.
    - As I am using .net 5 and the latest MudBlazor is developed with .net 6, it was not compatible. The official documentation only described that MudBlazor uses the latest Version but didn’t have anything written about the compatibility.

BlazoredVideo

* VideoPlayer for Blazor WASM
* Problem:
  + After dynamically changing the source of the Player the stream does not get refreshed
* Problem solution
  + The JavascriptEngine needs to be used to invoke a JS method which locates the video player by id and calls load(). There are no other possible solutions, as C# cannot interact with the video tag, therefore a JS call is needed.

What was positive:

* Build-in OpenFileDialog
  + There is a build-in function to open the file dialog so that the user can select a or multiple files. With this given event it is possible to access the files easily and afterwards to convert it to a data so that I can be played with the Video Player in the MP4 Content section
* Reusability / Plugins
  + Use of Plugins such as Material Design and other stuff is easy. Blazor also tries to implement key plugins as a build-in function. Like for example the InputFile Component was not a function of Blazor but since the .net 5 core version it is a built-in function and very easy to use.
    - <https://github.com/SteveSandersonMS/BlazorInputFile>
* Official Documentation
  + Microsoft has a huge documentation describing the parts of Blazor as well as C#. Therefore, there is no need for additional help.
* Used Structure
  + It is like programming html with C#. As I am working with C# and Java, there was no need for an introduction phase, I immediately started programming.
  + So, if someone is familiar with C# or Java, the structure will not be a problem at all.
* Combination of razor and C#
  + It is possible to add if statements within the razor part. So, it is possible to create if within the html section and also with the help of two-way binding.
* Data Binding
  + With Data Binding it was very easy to pass the source of the selected stream to the view without any additional calls or something else. It is also possible to call a method and pass a value to it, but not necessary.

What was negative:

* Accessing DOM elements
  + If you try to access a DOM Element, in this case the <video> tag, there is no possibility to do that with C#, you will need JavaScript for that. So, there will not be much to do with JavaScript, but some simple calls could be necessary.
    - Important: Blazor allows to call JavaScript functions with C# and the JavaScriptEngine. So, the code structure does not get destroyed. We still do not have any JavaScript fragments within the major razor file.