Dear Sirs and Madames,

the application was created using the Flask framework because, unlike similar technologies, it is rather lean and clear.

In the architecture, the necessary libraries were first imported to be able to initialize the functions.

First, the file index.html is rendered in the browser.

As a result, two paragraphs of text appear, each of which gives a brief explanation of the application in German and English (translated).

Centered and not overlooked, there are 2 input fields in which an integer can and should be entered.

If both input fields contain a valid value, the "ergebnis" function can be used for the calculation.

If one of the two input fields does not contain a number that is greater than 0 when the mouse cursor “hovers” over the “submit” button, an information window will inform the user to adjust the entries.

The "ergebnis" function accepts the values of the user input and saves them in variables, these variables are converted into the "intenger" type and processed by the "lcm" function from "numpy".

The output of the "lcm" function (smallest common multiple) is stored as a string in the "kgV" variable.

Finally, the value of the variable is re-rendered in the browser together with the values of the previous user input on the page index.html in order to visualize the user a confirmation of his input.

The design is kept minimalist. The font has been enlarged and does not correspond to the browser standard. The input fields and the "submit" button have been centered and have defined distances.

The further development of the application would only make it possible to press the "submit" button if both input fields were checked for their validity.

A side-by-side history with the latest results could be a useful support for the user.

The information window could be replaced by a visual information box directly on the input field.

In addition, there would be optical highlights that give the "button" an effect when "hovering" and activating, conceivable versions would be rotating frames or animations of the background.

The background would also be upgraded with color gradients or transparent images in an update.

Basically, I would prefer to have such applications run on the client side using JavaScript and only use a backend if necessary to counteract performance and reduce data traffic, which would make such an application executable offline.

The code was recorded in the "Visual Studio Code" code editor and tested with the "Chrome" browser.

Sincerely yours

Christian Plöger