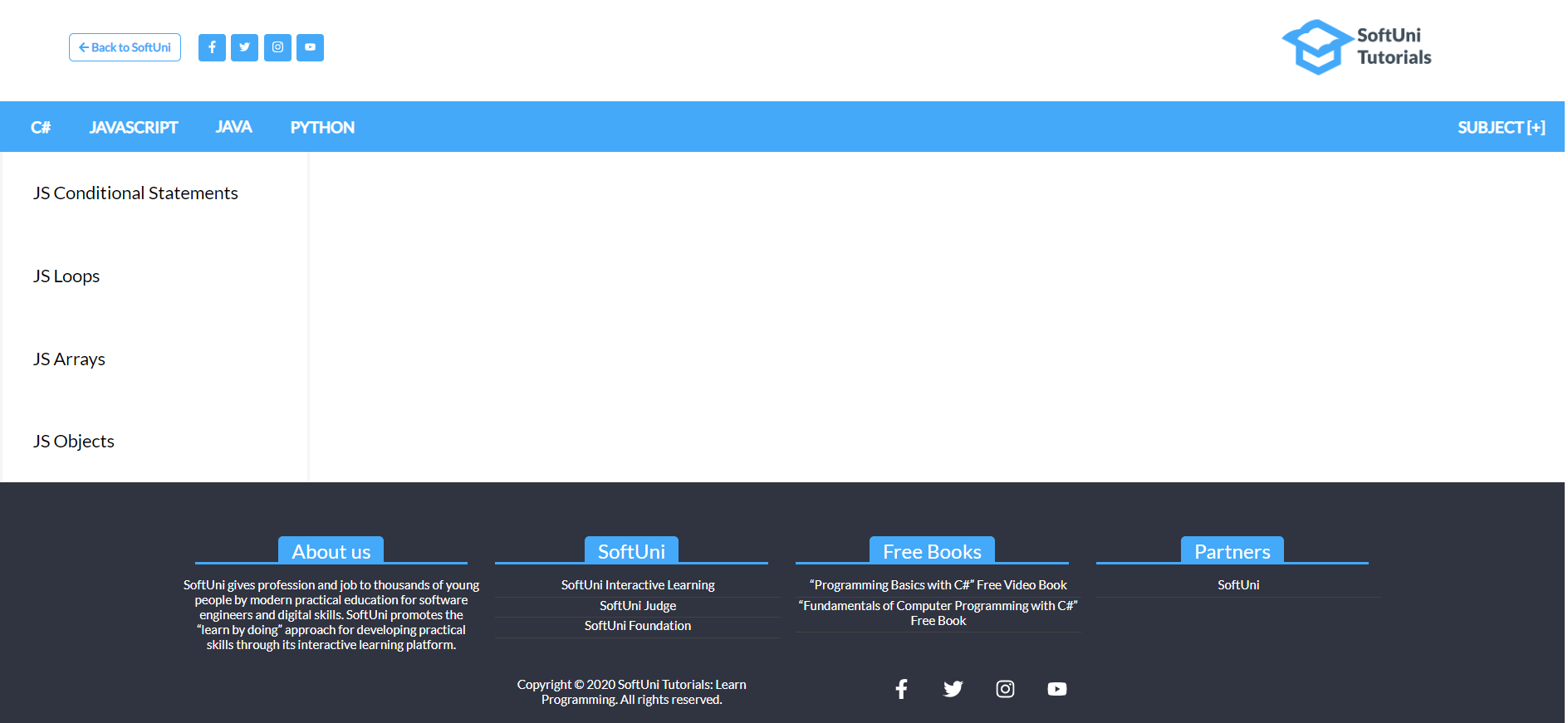
# SoftUni Tutorials - Workshop - Part 1

This application is created as a workshop for the [VueJS Course](https://softuni.bg/trainings/2820/vuejs-march-2020) @ [Software University](https://softuni.bg)

# Task Requirements

You will be provided with skeleton (**HTML**, **CSS** and **JSON** file)



The application is a platform for **programming tutorials (articles)** where each registered user can create article and any other user can read it. For now, your task is to check the given skeleton (HTML and CSS files) and **split** **them** **into** **components**, so each part can be **reusable**. Be careful when you structure the components.

Every technology should be **clickable**. When is clicked **all subjects** for the current technology should be listed below (on the left side) When some of the technology subject is being clicked, "**details view**" should be displayed so users can read the subject article (on the right side - **main section**)

If other technology or subject is clicked the current "**details view**" (article) should be replaced with the information on the last clicked cause.

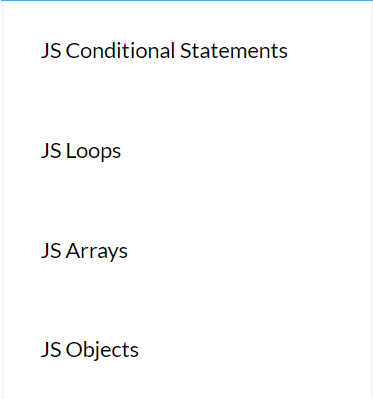
### Technologies



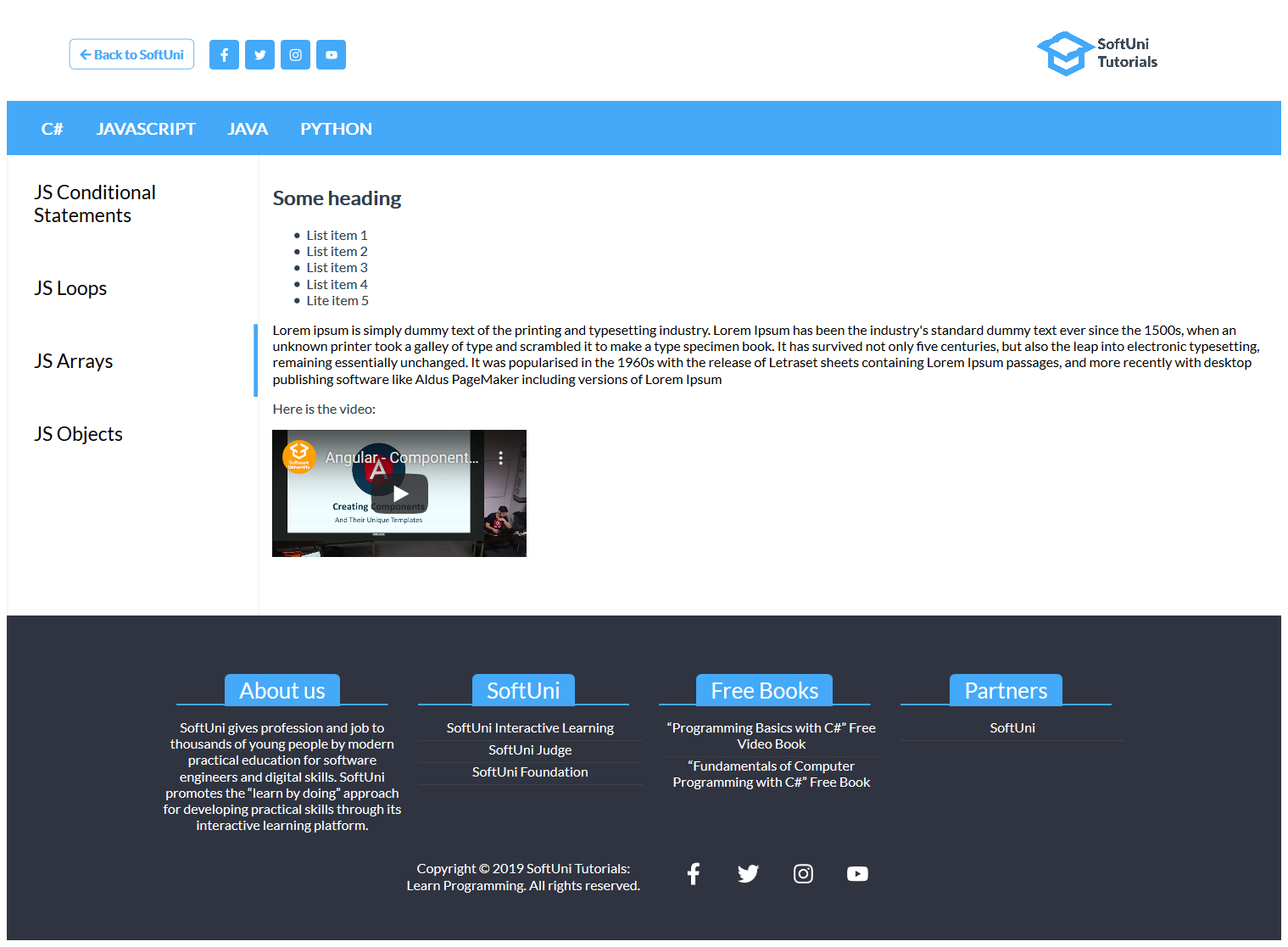
And [**SUBJECT]** on the right side



### Technology's subjects



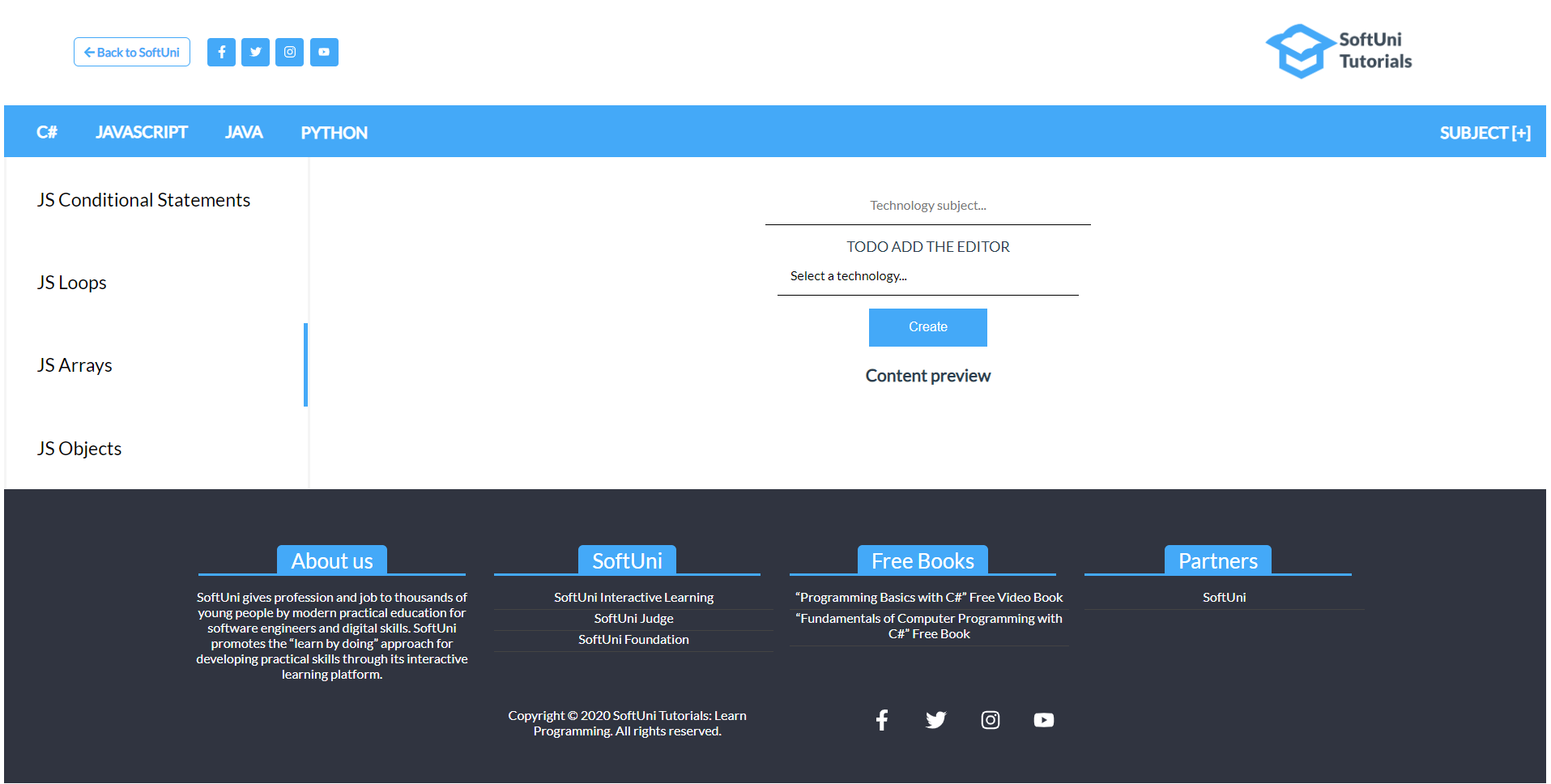
### Subject details view (article)



So far, we don’t saw how to route between pages, so you need to do this with some help with **passing data between components** to manipulate the selected technology, technology' subject

## Subject

One of the links you should implement is the view where the technology's subject is created.

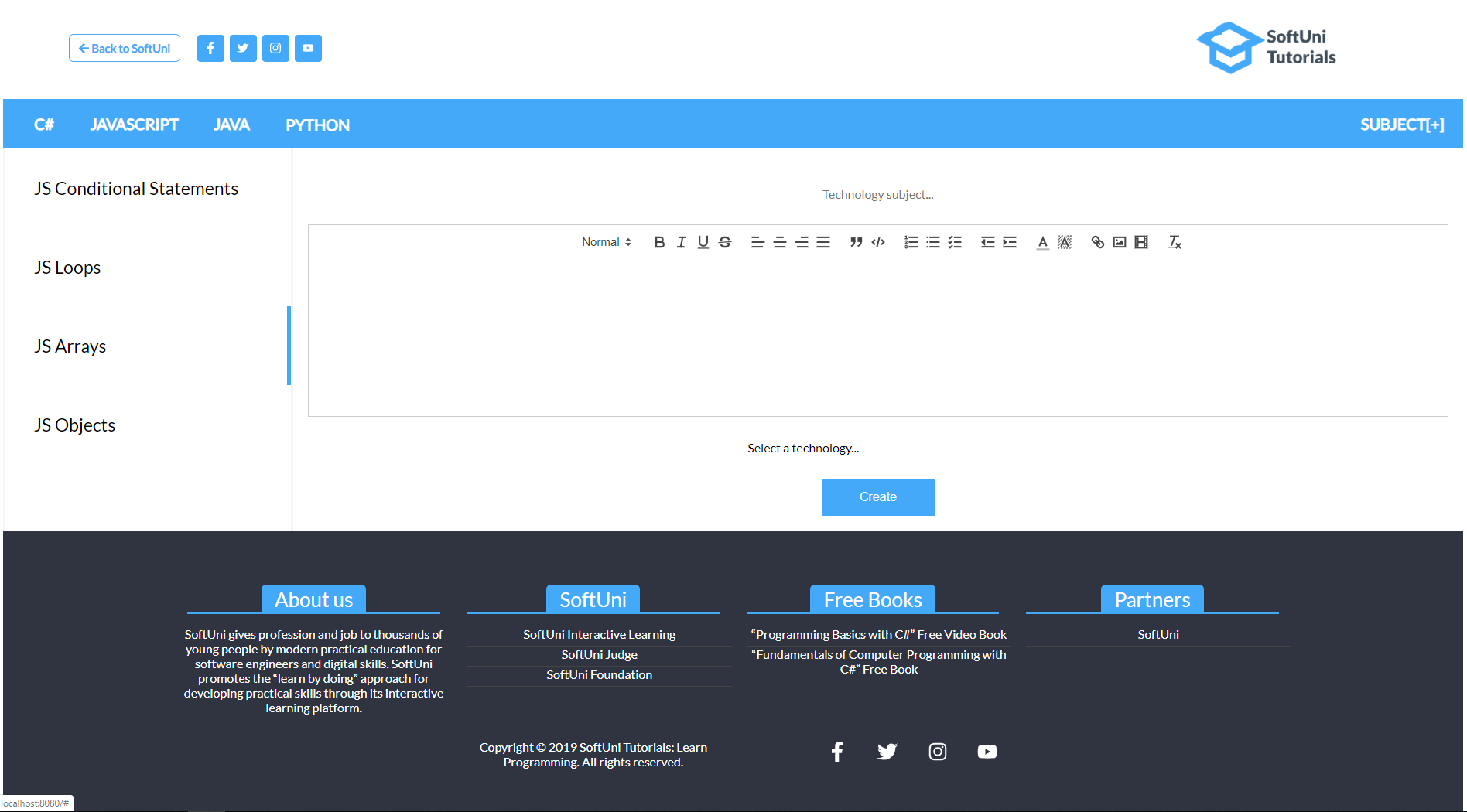


**TODO Section** is the interesting part. You should download a tool/library for **rich text editing**. We recommend some of these:

* <https://editorjs.io/>
  + <https://www.npmjs.com/package/@editorjs/editorjs>
* <https://www.vue2editor.com/>
  + <https://www.npmjs.com/package/vue2-editor>

These tools help controlling the appearance of your text using, sometimes called a **WYSIWYG** (What You See Is What You Get). Check the links above to see how to use them properly.

If you import correctly the component right, the output should be something like that (**using Vue2Editor**)



The cool part in this case is now if you two-way-data-bind the article content coming from the editor and represent it in other element. The result will be exactly as you wrote it in the editor.

## 

### \*\*Note

Make sure when some of the links from navigations (technologies or subjects) is **clicked**, **styles for active link** is applied. (Check the **CSS file** for more clarity)

***GOOD LUCK* 😊**