|  |  |  |  |
| --- | --- | --- | --- |
| **Problem** | **UIkit** | **Custom** | **Which is better?** |
| **Cross-browser compatibility:**  *deliver an acceptable  looking website that works on all browsers.* | According to the documentation:  “UIkit will work on pretty much any modern browser.” | Apply JavaScript solutions (polyfills) which are beneficial for serving responsive websites to the older browsers, lacking CSS3 support (media queries) and HTML5.  Explicitly define CSS to make the same behavior for different browsers with CSS Reset (or/and Normalize).  Incorporate CSS hacks or separate stylesheets for different browsers with conditional comments.  Incorporate fonts which are widely supported.  Incorporate JS library (for instance, jQuery) which will abstract away difference in the DOM, AJAX and JavaScript.  Validate and test website. | If majority of predictable site users will be using modern browsers – UIkit is a perfect choice, otherwise – custom website can be a right decision. |
| **Accessibility:**  *build a website with a consideration of how a disabled person accesses or benefits from a site.* | No accessibility features have been found. | Follow best practices. | Accessibility wise, custom website might be a better option. Alternatively, best practices can be implemented in a UIkit framework. |
| **Styling:**  *beautiful website with noticeably original style.* | Ability to modify the UIkit styling and create our own theme with CSS preprocessors (SASS, LESS). | Create custom CSS. | With no doubt, for the original design better to build a custom website, otherwise – UIkit is a good alternative. |
| **Speed:**  *ability to load a dynamic content of the site quickly.* | UIkit is specifically built to work with reactive JavaScript frameworks (React and Vue). | Integrate reactive JavaScript framework (React) into existing website. | Both options are quite sufficient for achieving integration with reactive frameworks |
| **Sort items:**  *ability to sort items in a list.* | UIkit has a component which creates sortable grids and lists to rearrange the order of its elements. | Incorporate JavaScript. | UIkit might be more convenient. |

**Cross-browser compatibility**

Responsive front-end framework is capable of making cross-browser compatibility easier, and UIkit framework might be an excellent candidate for taking all the hassle with browser support for us, as long as our concern is restricted to the modern browsers only. Unfortunately, for the older browsers (even for IE 8) we need to perform an extra work to be able to achieve an acceptable looking website.

**Accessibility**

From accessibility perspective is irrelevant whether we are utilizing UIkit framework or if we are building a custom responsive website. In both situations we anyway ought to incorporate accessibility features into our site. For instance, we can adhere to the next methodologies:

* structured content which will be in a logical sequence without CSS styles;
* semantic HTML;
* consider color not as single instance to differentiate items;
* alternative text for images;
* transcripts for podcasts, videos;
* skip navigation.

**Styling**

The prevailing problem of utilizing frameworks is that it is hard to achieve a clear and elegant presentation of a content that can distinguish one site from another. UIkit framework allows to modify the UIkit styling and incorporate our own theme with SASS or LESS.

In my opinion, if the main intention is to build a website with a distinct style, arguably UIkit would be the finest choice, comparing to a custom responsive website. Regardless of that, descent level of customization allows us to build a nice looking website.

**Speed**

The key difference of UIkit among other frameworks is an ability to integrate websites into React or Vue. In a nutshell, it is capable of forcing the websites to render data noticeably fast. On the other hand, we can easily integrate reactive framework (like React) into our custom site.

Once again, if to take into consideration a cross-browser compatibility, an ability to connect with React and Vue, neglecting unique style for the website, then ‒ UIkit might an excellent choice. Otherwise I would recommend to build a website, utilizing React on a first place, and then mind about styles and layouts.

**Sort**

UIkit has a component Sortable which allows to sort items in a list. We might accomplish that by incorporating JavaScript into our website. However, from my point of view, it is more convenient and easier to utilize aforementioned component of UIkit, because we might achieve the same result with a lesser custom coding.

**Conclusion**

UIkit is a light and modular front-end framework that helps build websites fast with significant customization of style. It’s ability to integrate site with reactive JavaScript frameworks (React and Vue) makes a future for the framework quite bright and level of popularity are expected to grow exponentially. Regardless of lack accessibility features and not well formed documentation, UIkit framework can be a good choice for front-end web developers with a bit experience in the field.