Responsive design frameworks or custom responsive design?

* Problem: Building a different design.
* Uikit: The styles and the framework can be modified and this gives different possibilities to achieve a customized look (such as border radius, vertical alignment, display utilities, responsive visibility and more) but it still looks like a frame. And we use the same core elements to build the page.
* Custom: It gives the opportunity to make a unique design which is suitable for the brand and will make the page more noticeable. The downside is it takes time, it is more expensive and it is necessary for the designer to have experience and knowledge.
* Which is better: When it comes to design the custom one is always better but our customers in real life rarely have extra time and money.
* Problem: Creating well developed functionality.
* Uikit: Offers different types of functionalities (most of them are in the JavaScript section) which are enough to build a well-functioning page. They are easy and fast to implement and function well with the other components of the page. The downside is that it is not always so easy to customize the built-in functionality in the way you want.
* Custom: It is necessary to have extra knowledge and sometimes the debugging process takes too much time. Sometimes it is possible that different components of the page can be in conflict with each other.
* Which is better: If we do not need to build some form of rare functionality it is better to use the framework solution, because it will work exactly as we expect.
* Problem: Do it faster.
* Uikit: The first thought is that is much faster to build the page with framework. But sometimes that is not true. If we use the particular framework for the first time or the version is brand new it takes time to read the documentation and to do exactly what we want using the libraries’ tools.
* Custom: When we work from scratch it takes time to test and order every element on the page to optimize the styles and manage all the files, but we know every part of the code and we can easily make changes if it is necessary. This last argument can also be weakness because if we work with other people or we change the front-end developer it might take time for new people to then understand the construction of the page.
* Which is better: The framework provides easy to implement solutions and if we work consistently with some frameworks and the web site has many pages it would be much faster to use frameworks than do everything from scratch.
* Problem: Browser compatibility.
* Uikit: Most of the frameworks are compatible with commonly used browsers and they offer components that look and work equally. Of course there are exceptions and we cannot be totally sure.
* Custom: It is necessary to test every part of the website, to see if it will work correctly with different versions of the chosen browsers. We can make a list of preferred browsers, because it will take some time to make our page cross browser compatible.
* Which is better: It is almost impossible to develop a website which is compatible with every browser, but when you do a custom web page you can do the compatibility you need. With frameworks it is necessary to test first and check what functions work, especially if there is no information in the documentation.
* Problem: File management or which loads faster?
* Uikit: Like most of the frameworks they come with a large number of files, and we do not necessarily need every one of them, but just deleting the extra files is not always the best solution. Therefore, we will lose some server space. However, that does not mean that we will definitely have a slower loading web page, because we can always link the minimum files to the html or do some other improvements to enhance performance.
* Custom: If we work carefully there will not be extra files that we do not use, and the images will be an appropriate size.
* Which is better: If we ignore the time lost it is much easier to build a fast loading web page by yourself.
* Problem: Which is easier to do?
* Uikit: The framework will take some time to study the documentation and sometimes it is necessary to think and read more if you want to implement correctly some of the functions, but it provides ready to use components and styles. However, if you know the documentation it can be really easy to order the elements for a brand new web page.
* Custom: Even for experienced developers it can take a lot of time and effort to build and test the web page.
* Which is better: If it is not necessary to customize more than styling and we do not need some kind of extra functionality, frameworks can make building webpages much easier.
* Problem: Accessibility.
* Uikit: Some frameworks provide really well developed accessibility, and UIkit also provides some functionality - like alt tags and well-structured and correctly used HTML.
* Custom: When it comes to accessibility we can always do more so we can write every part of the page and improve it, but that will take some extra effort as we implement new methods and techniques and then test them.
* Which is better: It depends. It is easier to choose the already working and tested option which has everything already done. But if you want perfection in the details it is necessary to write all of them yourself.