5 Mistakes Every Frontend Developer Should Avoid

CODING WITH OUTDATED HTML STANDARDS

The inanest thing to do here is to not evolve into the newest technology like HTML5. You need to go with the trends and act progressively and code with new HTML elements to access and avail the prevailing standards. This way your product would not just put up with the best of browser support but you will be able to attain best of quality with your design interface. Besides, this will also allow you to make the most out of the latest features that the newest facilities have to offer.

NOT CREATING RESPONSIVE MOBILE DESIGN

Each day passing people are increasingly shifting towards mobile devices. Which spikes up the urge for developers to be mobile-responsive in their approach. If they miss out on creating a responsive mobile design they are going to lose a large chunk of users who mostly prefer to browse through their hand-held gadgets and this is only going to grow with time. Therefore, not creating an interface that does well with mobile devices is the worst thing you can do to your product.

USING OUTMODED JAVASCRIPT LIBRARIES

Using JavaScript libraries that will not be supported in the future will only make your website go outdated with time. Make sure you are functional and responsive enough at the frontend to meet advanced user needs in the coming time and avoid being landed up in an adverse situation that will follow if you continue to use the obsolete JavaScript libraries. AngularJS and KnockoutJS are good examples of the libraries that will ensure you all the competence and consistency with future-proof frontend design.

DUPLICATED CODE

There is one programming principle called DRY (Don't Repeat Yourself). When you are copy-pasting from one place to another, which means you are duplicating the code.
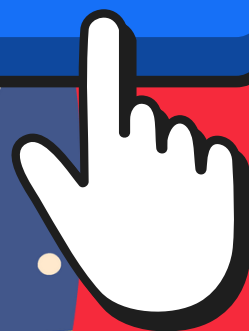
IMPORTING THE WHOLE LIBRARY

Avoid importing the whole library at all costs! This will increase the bundle size. I will take lodash as an example. Do not <u>import _ from "lodash"</u> . This will increase your bundle size by over 500kb.
You should import the only function you need. Eg <u>import isEmpty from "lodash/isEmpty"</u> ,