



# Learn to become a modern Frontend Developer in 2019

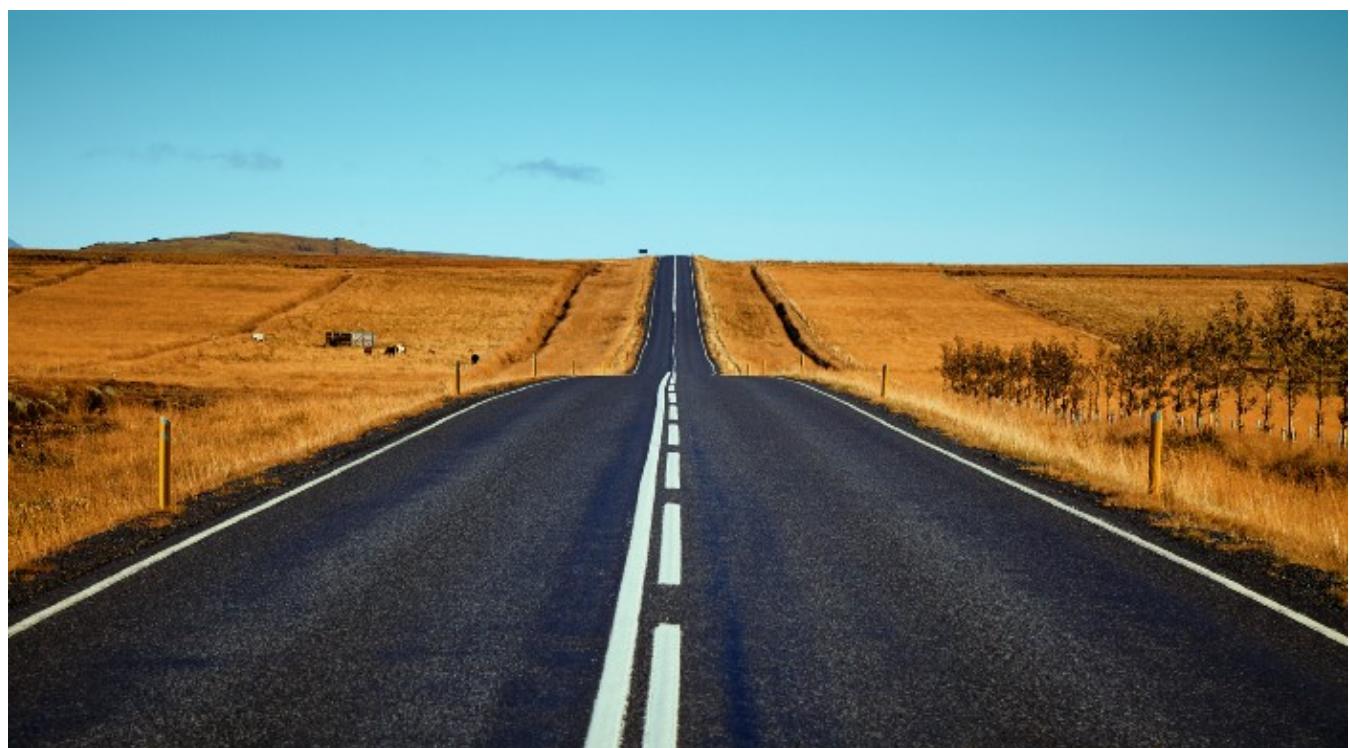


Kamran Ahmed

Mar 26, 2018 · 10 min read

*I have re-written/replaced this post for the new year, making it a bit more concise and easier to follow with some recommended exercises in between, so don't fret if you find it a bit different from the one that you have read before or if you find contradictory comments. The version for previous years can be found in the [GitHub archives](#)*

Web Development is an ever-changing field — the way we build websites today is completely different from how we used to do it a couple of years ago. With the plethora of tools available and the new ones popping up every day, most of the time developers find themselves confused about which way to go.



Just a random image because thumbnail needs to be a part of the article 😊



frontend, backend or the operations. I initially created these roadmaps in 2017, then updated in 2018 and most recently revised in 2019. At the time of this writing, frontend developer roadmap has been updated for the current year. I am still working upon the Backend and Operations/DevOps roadmaps and will hopefully release it in the next few days.



### kamranahmedse/developer-roadmap

Roadmap to frontend, backend or operations in 2019.

[github.com](https://github.com/kamranahmedse/developer-roadmap)

In this post, I am going to recap a bit and share with you the motivation behind making these roadmaps, a few disclaimers if you decide to follow these roadmaps and finally have a breakdown and tips on how to learn if you were to become a modern frontend developer in 2019.

## Motivation

Before we get started with the article, just to give you an idea about me, I have been doing the Fullstack Development for the past 6 years and currently working as a lead engineer at [tajawal](#) where I have to wear many different hats as a part of job. It is not only my hobby but also one of my job responsibilities to keep an eye on the trends, taking up the tech decisions and keeping the developers motivated and trained.

Web Development has evolved a lot and has evolved fast; the number of options available on the frontend are enough to baffle anyone. I am pretty active in the opensource and the community; if I had a penny for every time I was asked or saw this question “what should I learn next” pop up in the forums, I would have retired a couple of years ago. Circa 2017, an old university professor of mine was preparing a set of paths for her students to give them an idea about the market and reached out to me to give her a list of tools and tech recommendations for web development. I jotted down a rough sketch and forwarded to her but later on I decided to clean it up a little and put it on GitHub so that I could refer anyone to that whenever I am asked this question. And that is how these roadmaps were born.



*The purpose of this roadmap is to give you an idea about the landscape and to guide you if you are confused about what to learn next and not to encourage you to learn what is hip and trendy. You should grow some understanding of why one tool would be better suited for some cases than the other and remember hip and trendy never means best suited for the job*

And secondly do some research your job market

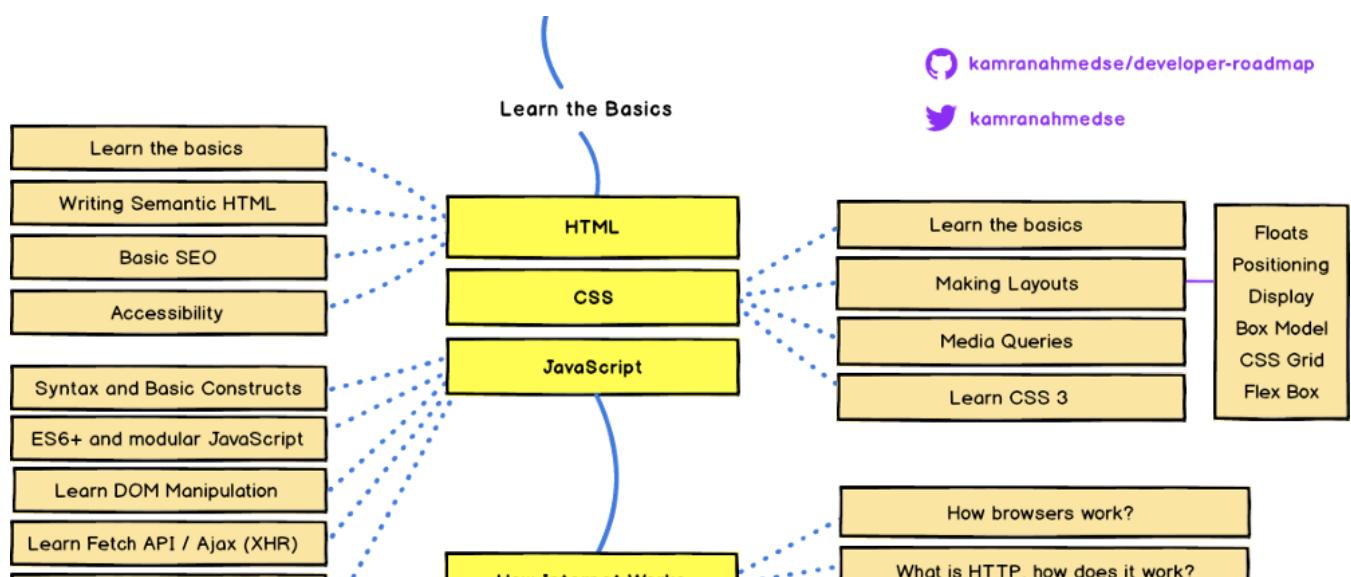
*Do not ignore the fact that languages and tools can be very market dependent so do some research on the market that you are targeting.*

Thirdly, you don't need to know everything listed here

*You don't need to learn all of the things listed here to land your first job. The roadmaps might seem humongous but don't let them scare you if you are just stepping into the web development. I have just tried to cover everything that you will eventually be learning or using. You can learn the bare minimum and keep learning the rest while you start building things.*

## Phase 1 — Job Ready

If you are a beginner and just stepping into the web development, there is a different roadmap for you. Have a look at the roadmap below and stop reading any further; complete the items listed in this roadmap and come back once you have made a few projects using them





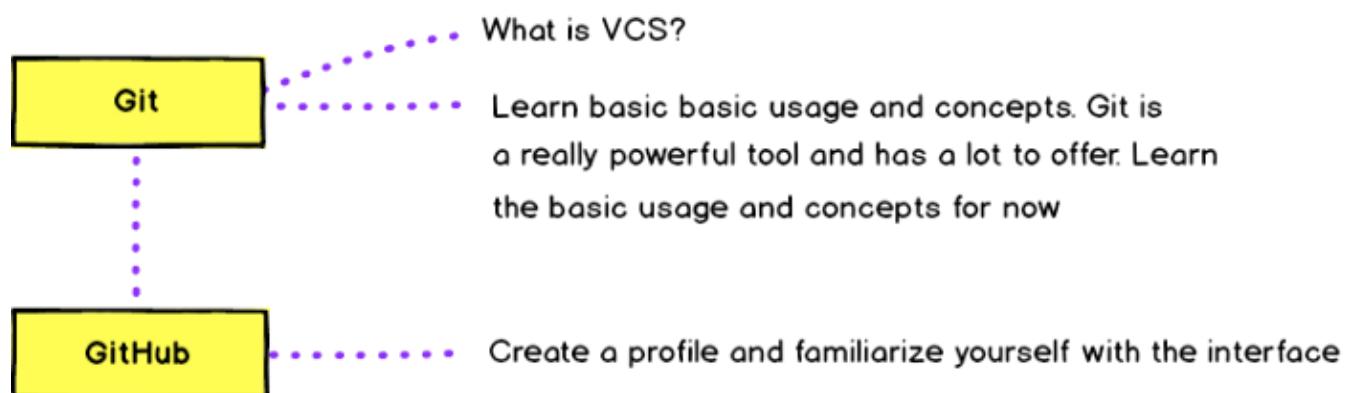
WHAT IS DNG, HOW DOES IT WORK?

Just learn the listed items and you should be able to call yourself a web developer and find a job in the market. I know a lot of people who have been doing just these and making decent money from freelance or a day job. Spend some time on all the items listed in the roadmap, get a solid grip on all of these and practice a lot. Make lots and lots of projects; here is the list of some of ideas that you can make

## Tasks

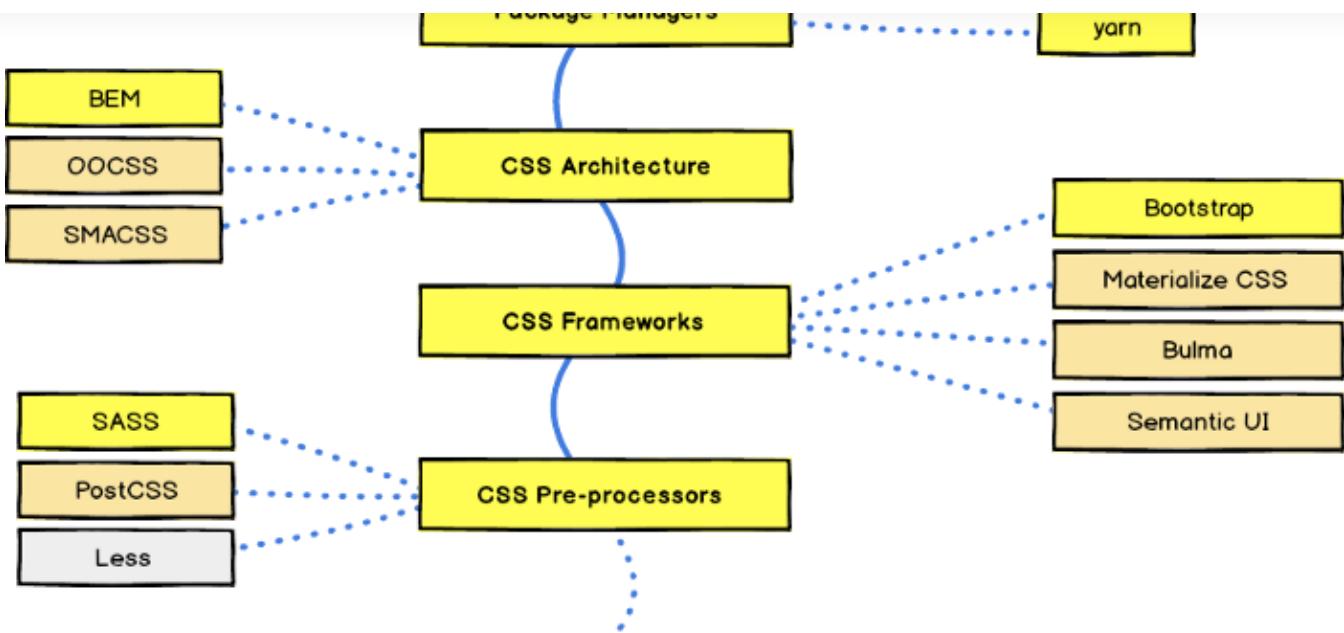
- Create a Pomodoro Application. You may clone and make the web version of this application
- Create a webpage that uses GitHub Repositories API to fetch and show the top 10 repositories of the current week in a nice responsive format.
- Create a simple todo list application that allows you to add tasks, mark them complete, edit them and delete them.
- Create a simple stop watch where user can start, stop, pause and reset.

Once you are done with this, learn about version control systems, learn the basic usage of Git and create your profile on GitHub



## Phase 2 — Write Better CSS

Once you have covered the basics, move on to next step and learn how to write maintainable CSS and use CSS frameworks. Here is how the roadmap should look like



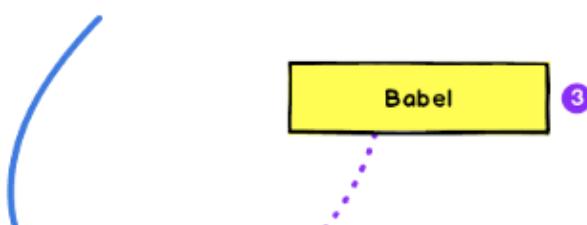
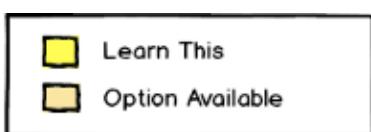
Once you are done with this step, go ahead and do the following tasks on the projects that you made above in phase 1

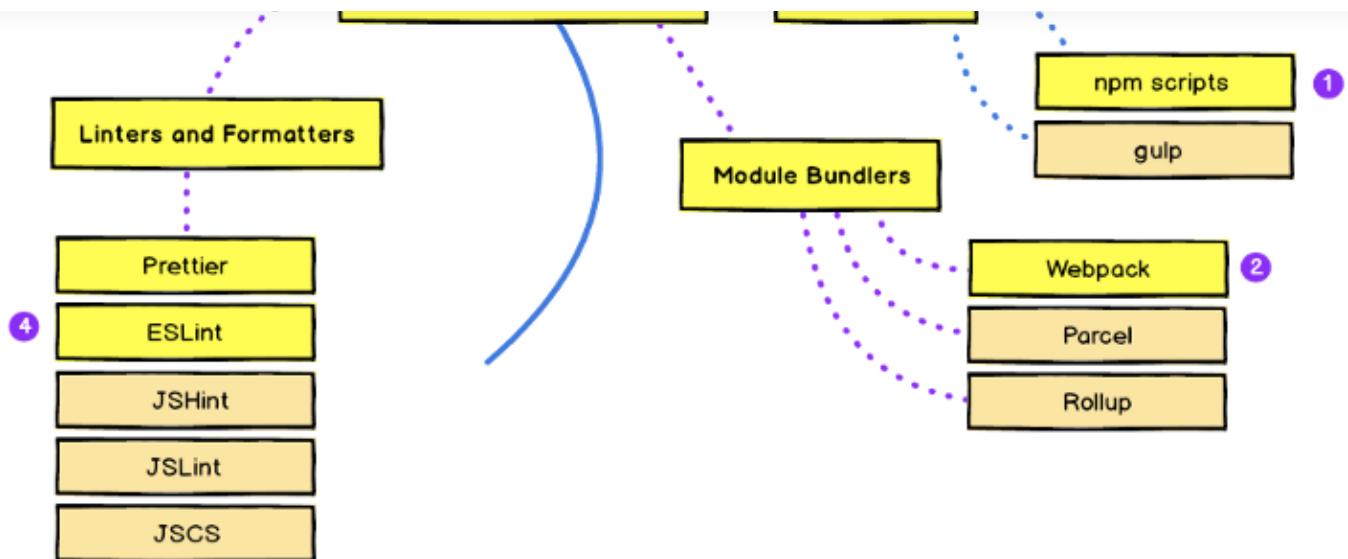
## Tasks

- Add bootstrap to the above projects using *npm* or *yarn*.
- Convert the projects that you made in Phase 1 to use BEM
- Write the CSS in SASS
- Automate the SASS to CSS conversion using NPM Script

## Phase 3 — Ramping Up

This phase is going to be some time and is your step into the modern frontend development. Go ahead and learn more about the JavaScript. Learn what is Webpack, understand the different concepts and why was it ever needed. Understand what is babel, why do we use it and learn how to integrate with webpack and lastly learn how to lint your code using ESLint. All of the items listed in this phase revolve around webpack.





You will start getting the taste of the modern frontend development by the end of this week. Once you are done with this phase, go ahead and do the below steps to get a good grip on the build tools

## Tasks

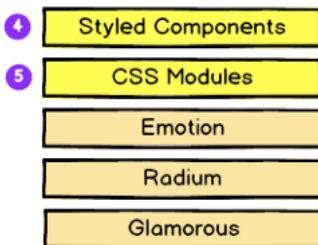
- Create an npm package that takes a username and gives back the list of links found on the social networks (if they exist). It should be usable in the browser, bundle it with webpack, use babel to transpile it and use ESLint for linting.
- Create a simple todo list application, use SASS for the CSS, add bootstrap for styles, use BEM, transpile JavaScript using babel, bundle it with webpack, create optimized production build and deploy it on Github Pages.

## Phase 4 — Modern Frontend Applications

Next step is learning about some frontend framework. There are multiple options but the ones most commonly being used these days are React, Angular and Vue. I would recommend you to go with React.

First of all learn React, then have a look at redux and after that learn about CSS in JS; which isn't required but have a look at Styled Components and CSS modules if you would like.





3. CSS in JS

Vue.js

Vuex

Once you have learnt react, go ahead and read about Progressive web apps. Now that you know frontend frameworks, it shouldn't be that difficult for you. Have a look at the [PWA checklist](#), read about service workers, measuring performance, using lighthouse and look at the different browser APIs that you can use to your advantage e.g. Storage, Location, Notifications, Device Orientation and Payments. Also read about [RAIL model](#) and [PRPL pattern](#).

Once you are done with this, you should be able to call yourself a modern frontend developer. Make sure to practice what you learn. Here is the list of tasks that you may pick from, if you are looking for ideas.

## Tasks

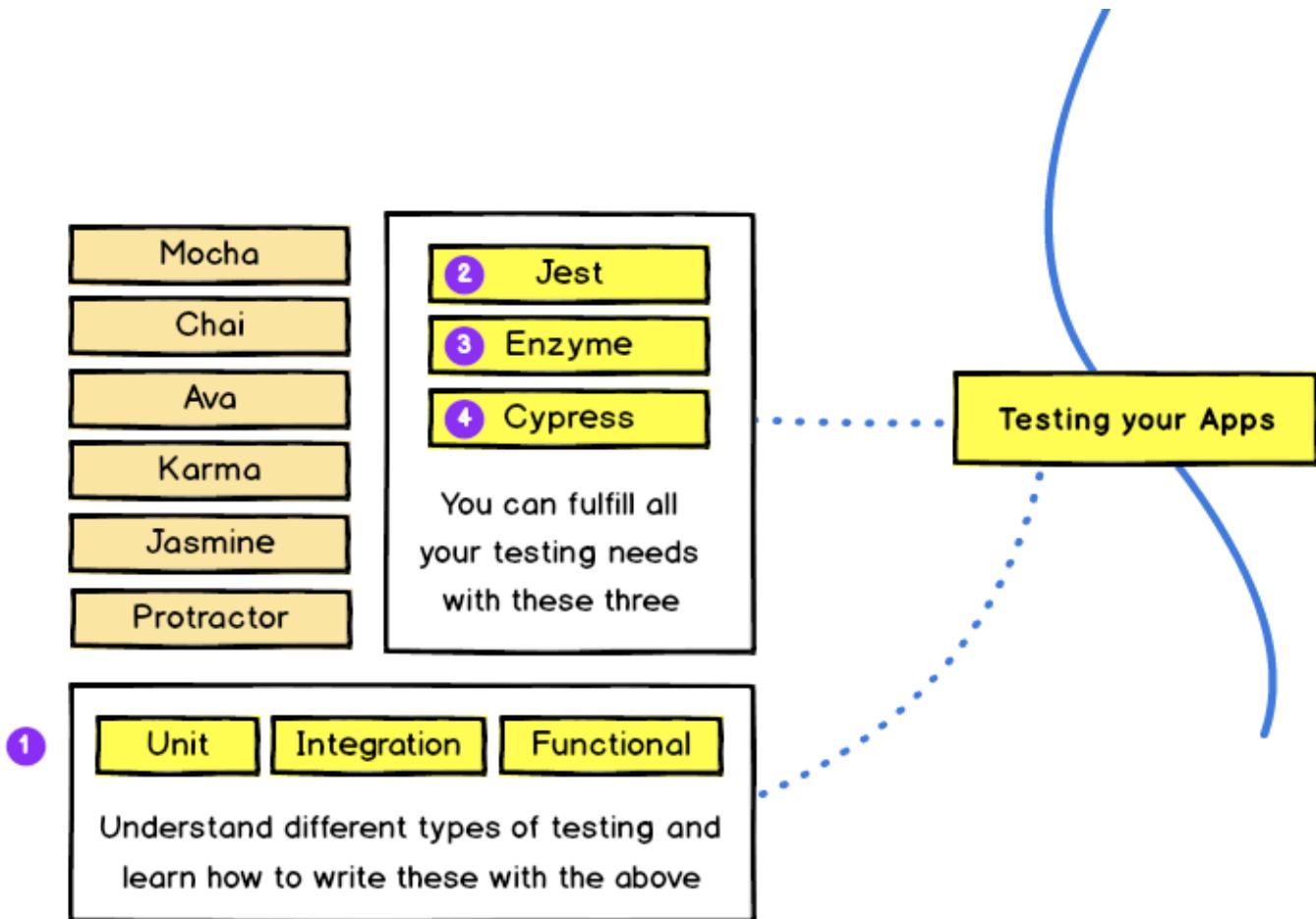
- Create a simple application that lets you pick a few hash tags and uses [twitter's search API](#) to fetch and show you the most recent tweets for those hashtags in a trello like layout grid. Try to pin the hashtags so that when the user refreshes the page, it remembers the hashtags that you picked. Use react router and add about pages.
- Create a [pomodoro application similar to this one](#) that lets the users configure the duration for work and breaks, shows notifications and plays a sound whenever work or break has ended/started.
- Re-create the [github trending page](#) using React and allow filtering using language and dates just like github. You may add any libraries for dates.

## Phase 5 — Automated Testing

Learning to write automated tests for your applications is going to save you a lot of headache in future and is going to put you in a better position when looking for a job.



Cypress in the respective manner. Also learn to calculate the test coverage.



## Tasks

For the tasks, go ahead and write unit, integration and functional tests for the application(s) that you created in Phase 4 above.

## Phase 6 — Static Type Checkers

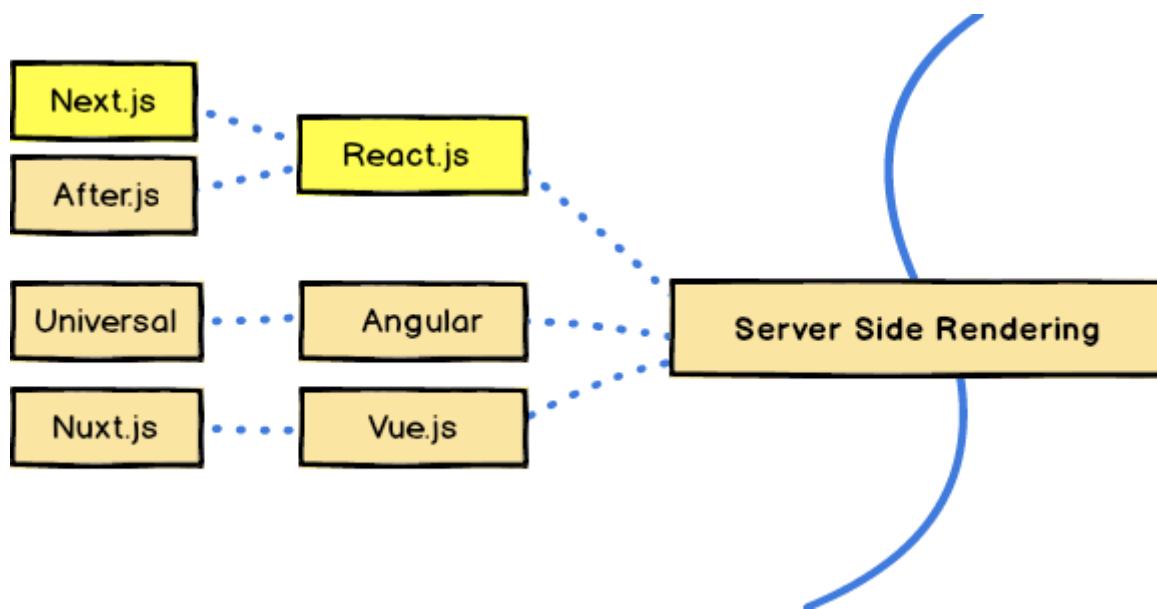
Type checkers allow you to make your code more maintainable as it grows, increases your agility when doing refactoring, provide better support in the IDEs and are the best form of documentation that you can have. There are mainly Flow and TypeScript in this domain. However, there is more push towards TypeScript and I would recommend you to go with that.



Once you are done learning TypeScript, go ahead and convert any of your existing JavaScript applications to use TypeScript.

## Phase 7 — Server Side Rendering

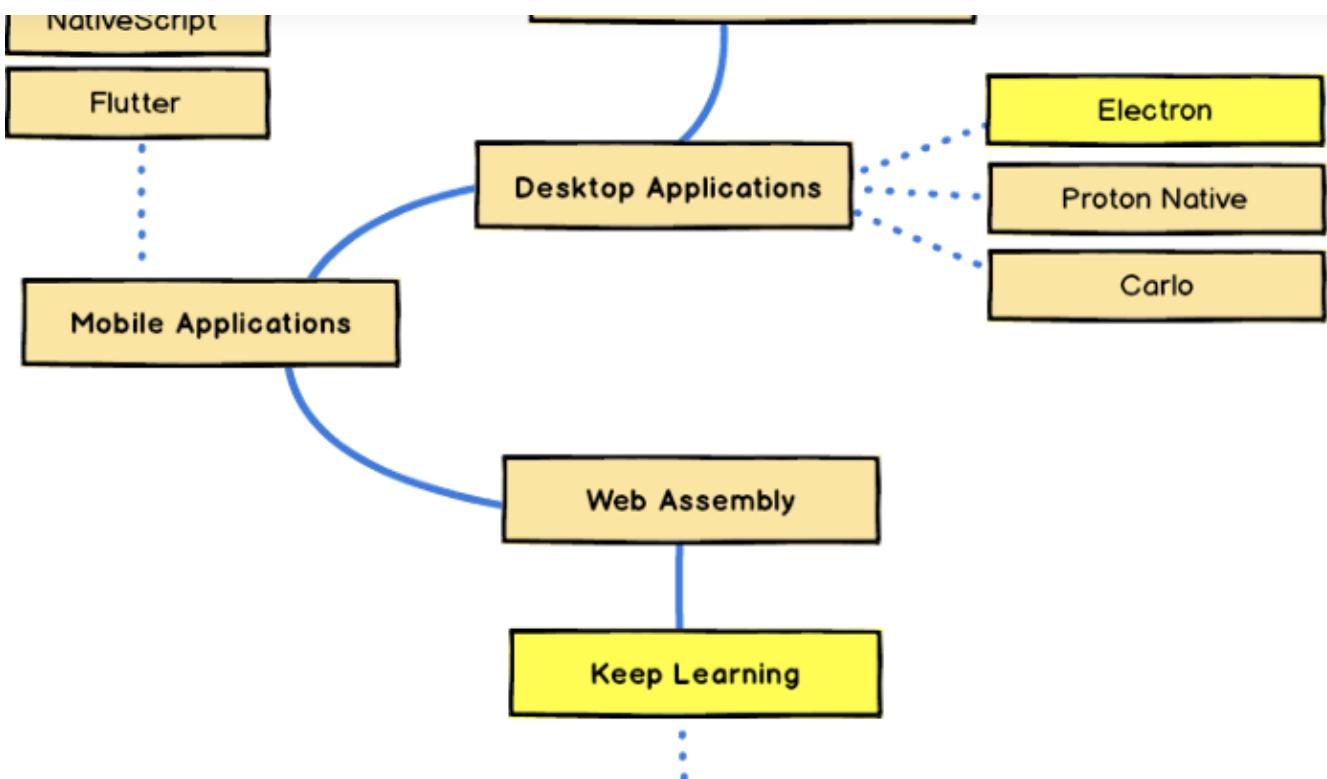
Server rendered applications allow achieving better performance and improved SEO as compared to the client rendered applications. Although not a requirement but it would definitely help you in crafting better frontend applications. There are different options available, based on the frontend framework of your choice; but if you picked React.js then you should go with Next.js which makes SSR a breeze.



For the Tasks, convert any applications that you made above to be rendered on the server side using Next.js

## Phase 8 — Go Beyond

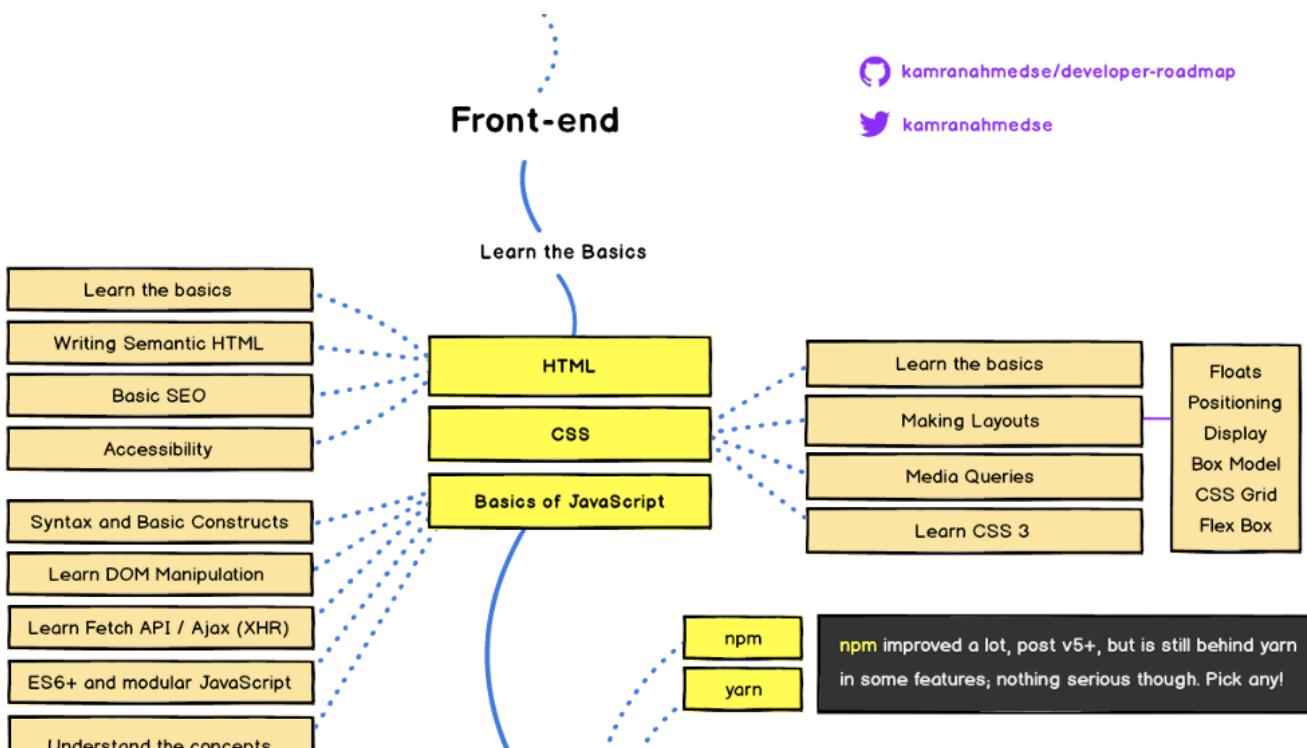
Everything shown in this phase is optional and is not really required for you but if you would like to try them out, go ahead and have a look.

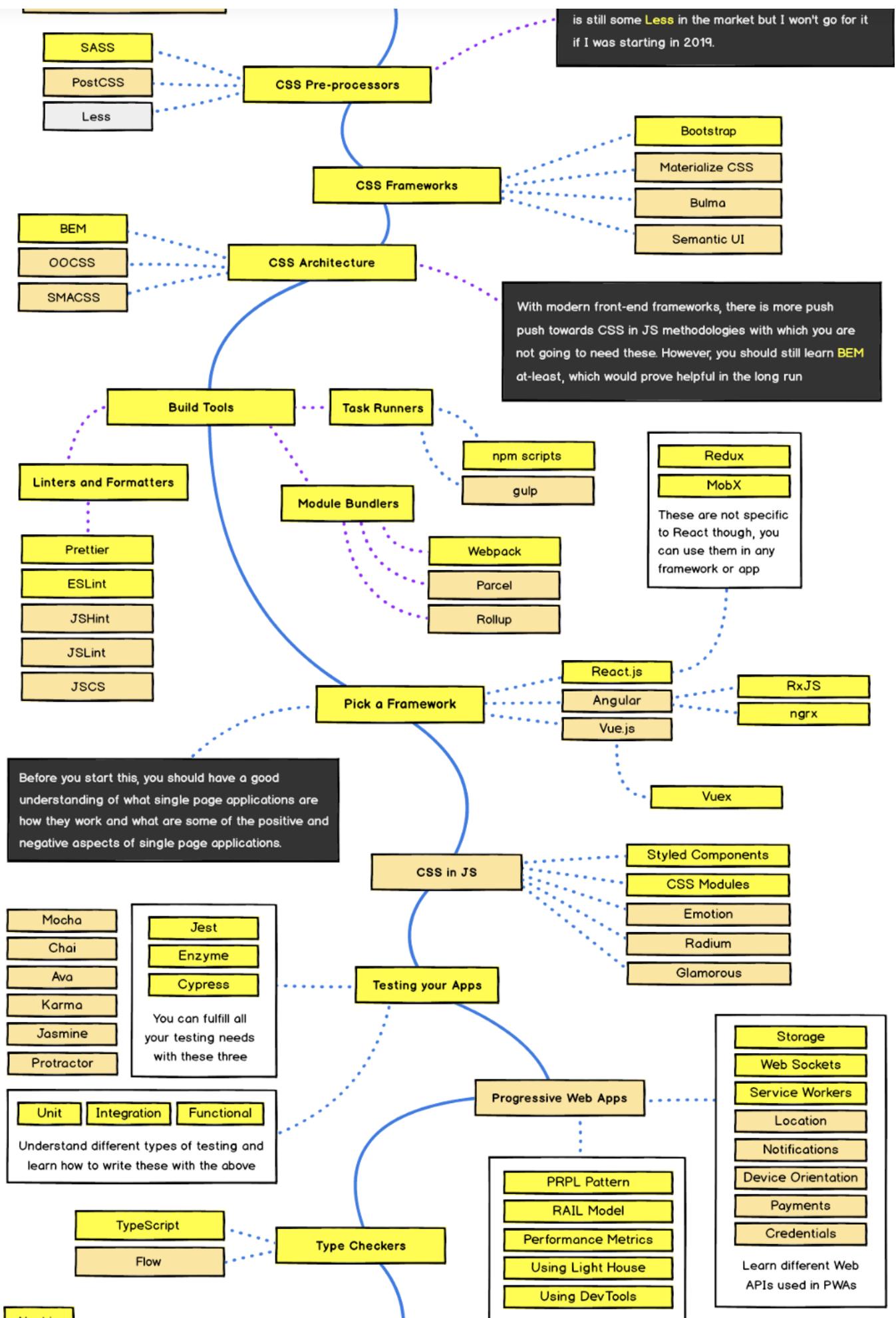


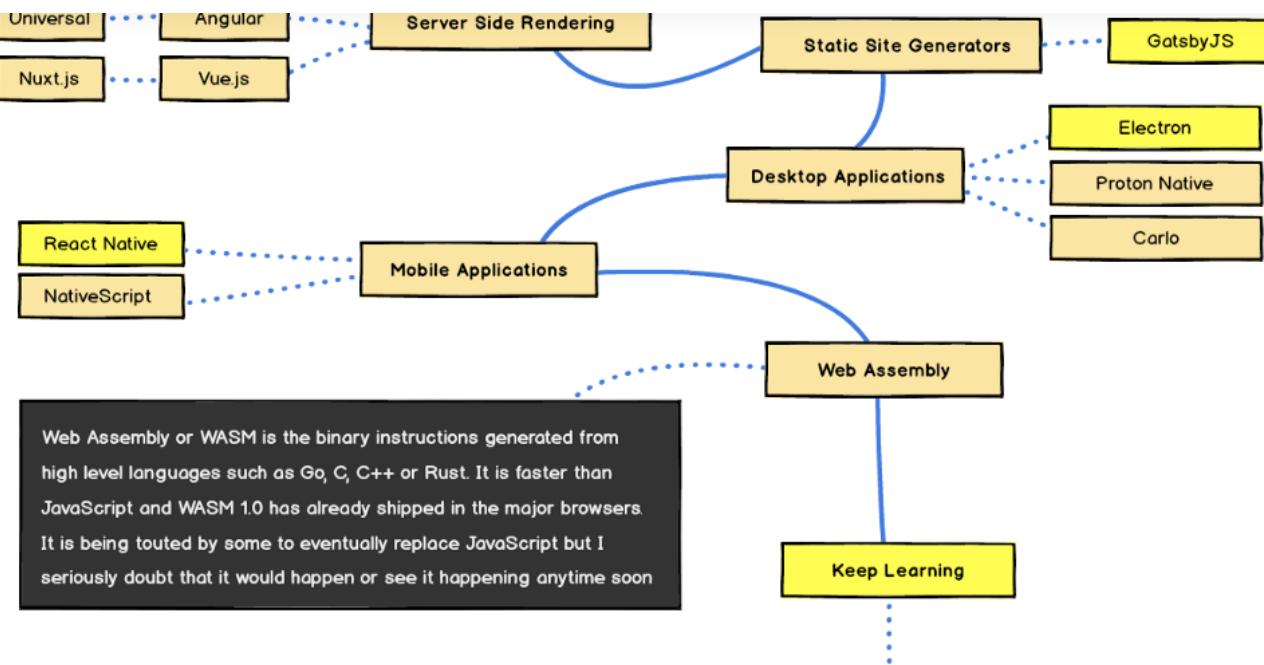
Please note that for the sake of brevity, I haven't gone into the nitty gritty and have tried to give you a bigger picture; go and figure out while you learn.

## The Complete Roadmap

The complete chart for the whole roadmap is given below with some additional details written inside it.







There could be things still missing in the roadmap but this is all of what you need for any “Frontend Engineering” role. And remember the key is to practicing as much as you can. It might look scarier in the beginning and you will feel like you are not grasping much but that is normal and over time you will feel that you are getting better and better. And don’t forget to ask for help if you are stuck, you will be amazed by how many people are willing to help.

This roadmap along with the backend and operations roadmaps can be found through my GitHub profile. Please note that I am still working on upgrading the Backend and DevOps and plan to release them in the next few days.

Having said that, this post comes to an end, feel free to befriend me on [twitter](#) or say hi by [email](#). Until next time, stay tuned!

**Kamran Ahmed (@kamranahmedse) | Twitter**

The latest Tweets from Kamran Ahmed (@kamranahmedse). Problem Solver, Engineer, Explorer, Opensourcerer, Lifelong...

[twitter.com](https://twitter.com/kamranahmedse)





WRITTEN BY

## Kamran Ahmed

Lead engineer at tajawal. Tech guy with an entrepreneurial spirit and knack for getting things done  
<http://github.com/kamranahmedse> <http://kamranahmed.info>

[Follow](#)[See responses \(93\)](#)

## More From Medium

Related reads





## Web Developer Roadmaps: All In One Place



Bradley Nice in Level Up!

Jul 29, 2018 · 2 min read

7.5K



Related reads



## How I Started in Web Development



Karel Moryoussef in codeburst

May 5 · 5 min read ★

1.1K

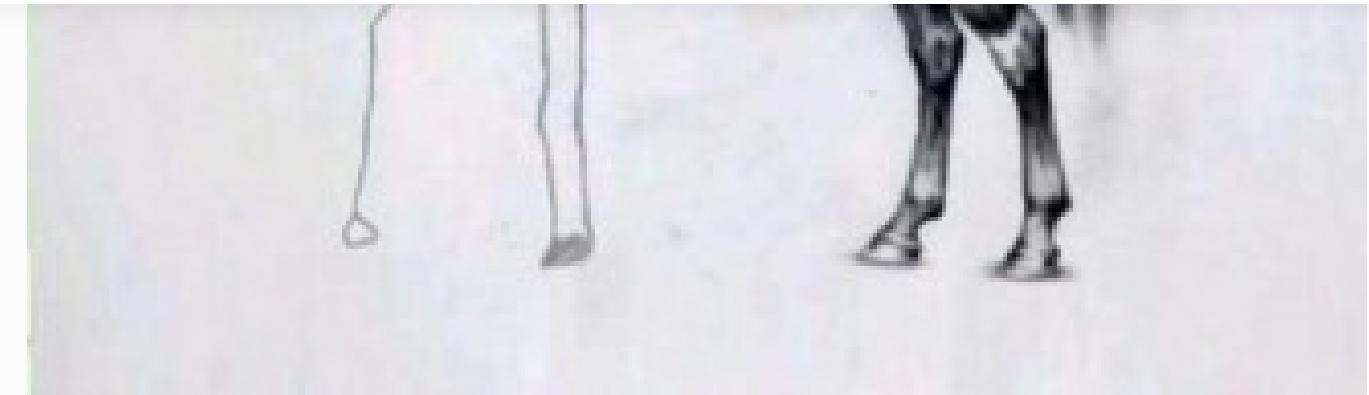


Related reads





Javascript



## A developer's guide to web design for non-designers



Patrik Krupar in freeCodeCamp.org

Aug 11, 2018 · 10 min read



19.7K

