FRONTEND DEVELOPMENT ROADMAP

swipe

1- GIT :

Git is a version control system that enables developers to track changes in their project. Git also helps developers collaborate as a team.
Here are some resources to learn Git:

- Git Handbook - Github guides
- Git - Git
- Git book - Git
- An introduction to Git - Freecodecamp
- Version Control with Git - Udacity
- Git & GitHub Crash Course For Beginners - Traversy media

2- BASIC TERMINAL USAGE (COMMAND LINE) :

The terminal is an interface used to execute text commands, and it gives you access to the underlying operating system.
Command line is very important so I strongly recommend you study more on how to use it. The better you are with the command line, the more efficient you will be as a front-end developer.
Here are some resources to learn command line:

- Command-line crash course - MDN web docs
- Getting Started with the Linux Command Line - Pluralsight
- Linux Command Line Basics - Udacity
- Learn the Command Line in Terminal - Openclassrooms
- Learn the Command Line - Codecademy

3- TEXT EDITOR / IDE (INTEGRATED DEVELOPMENT ENVIRONMENT) :

The text editor is where you will write your code. Having the right text editor can improve your productivity. There are a lot of IDE to pick from, but let's look at a few:

Visual studio code

- visual studio code is the best IDE for front-end development. It works across all operating systems such as macOS, Windows, and Linux. Visual studio code comes with a great deal of extension which helps in improving productivity as a front-end developer.

Sublime text

- sublime text is also available on macOS, Windows, and Linux. It is fast, easy, and flexible to use.

Atom

- Atom is an open-source code editor for macOS, Linux, and Windows developed by Github with supports for plugins.

4- WEB FUNDAMENTALS :

It is important to understand how the web works before you start learning any technologies. You should learn about things like:

- What is the internet and how it works
- HTTP / HTTPS
- Browsers and how they work

5- GITHUB :

GitHub is a code hosting platform for software development. GitHub lets teams work together on projects and it is also used for version control. It can be used among teams to collaborate on a project.
To get started, sign up for GitHub.

Here are some resources to learn Github:

- Github Guides - Github guides
- What is GitHub? - Github
- Git and GitHub for Beginners - Crash Course - Freecodecamp

6- LEARN HTML:

HTML stands for Hypertext Markup Language. It is the markup language for building web pages, it is also the building block of the web. HTML is easy to learn and comprehend. With just HTML, you can build a basic website.

Here are some resources to learn HTML:

- HTML tutorial - w3schools
- HTML tutorial - javaTpoint
- HTML Full Course - Freecodecamp

7- LEARN CSS :

CSS stands for Cascading Style Sheets. It is the technology to learn after HTML. It is used for styling our HTML. For example, we can use CSS to space our content, colours, fonts, etc.

The basics of CSS you need to learn are:

CSS box model:
- The CSS box model consists of the margin, border, padding, and content.

CSS positioning:
- CSS positioning helps you manipulate an element to different locations such as fixed, relative, absolute, static, sticky, etc.

CSS grid:
- CSS grid is a two-dimensional system with rows and columns. CSS grid makes it easier to structure a web page without having to use floats.

CSS Flexbox:
- Flexbox is a one-dimensional system that allows us to choose between a row or a column as the main layout or structure of a web page.

Responsive design and media queries:
- Responsive design is the approach that an application should be built or designed with the user in mind irrespective of their environment such as screen size/devices. Media queries are useful when you want to modify your application to fit a device such as Desktops, tablets, and phones.

7- LEARN CSS :

CSS Frameworks:

Tailwind CSS:

- According to the official documentation, Tailwind CSS is a utility-first CSS framework for rapidly building custom user interfaces. Tailwind allows us to use inline styling and achieve incredible results without using a single line of CSS.

Bootstrap:

- Bootstrap helps us build fast and responsive websites.

Foundation:

- Foundation is a responsive front-end framework that makes it easier to design responsive websites, apps on any device.

Here are some resources to learn CSS:

- CSS tutorials - w3schools
- CSS Crash Course For Absolute Beginners - Traversy Media
- HTML5 & CSS Development - Udemy
- CSS Tutorial - Zero to Hero - Freecodecamp
- SASS Tutorial - w3schools
- Flexbox CSS in 20 minutes - Traversy media
- CSS Grid Layout Crash Course - Traversy media
- Less CSS PreProcessor Tutorial
- Tailwind CSS crash course - Traversy media
- Bootstrap 4 - w3school

8- LEARN JAVASCRIPT :

JavaScript is one of the most popular programming languages in the world. It is the language of the web. As a front-end developer, it is required you learn JavaScript.

Some of the basics of JavaScript you need to learn are:

JavaScript Syntax:

- Every programming language has its own rules on how a program is written.The syntax of JavaScript is the set of rules that determine how a program is written by a programmer and interpreted by a browser.

DOM Manipulation:

- DOM stands for document objects model. According to W3C (World Wide Web Consortium) standard, the DOM is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document. The DOM is a representation of how the content of a web page is structured. JavaScript manipulates the DOM by updating the content, the style, removing elements, adding new elements.

Learn Fetch API:

- API stands for Application Programming Interface. An API is an intermediary that allows two applications to communicate with each other. As a front-end developer, when building a web application, chances are that you will have to work with external data such as Third-party APIs, fetch API allows browsers to make HTTP requests to a web server.

8- LEARN JAVASCRIPT :

JavaScript Frameworks/Libraries

After learning the basics of JavaScript, you can pick any JavaScript framework of your choice.

React

According to the documentation, a JavaScript library for building user interfaces React lets you create reusable components.

Vue

Vuejs is an approachable, versatile, performant javascript framework that helps you create a maintainable and testable codebase.

Angular

According to the documentation, Angular is an application design framework and development platform for creating efficient and sophisticated single-page apps.

Ember

Ember allows developers to create a scalable single page application. It includes everything you need to build a rich user interface that works on any device.

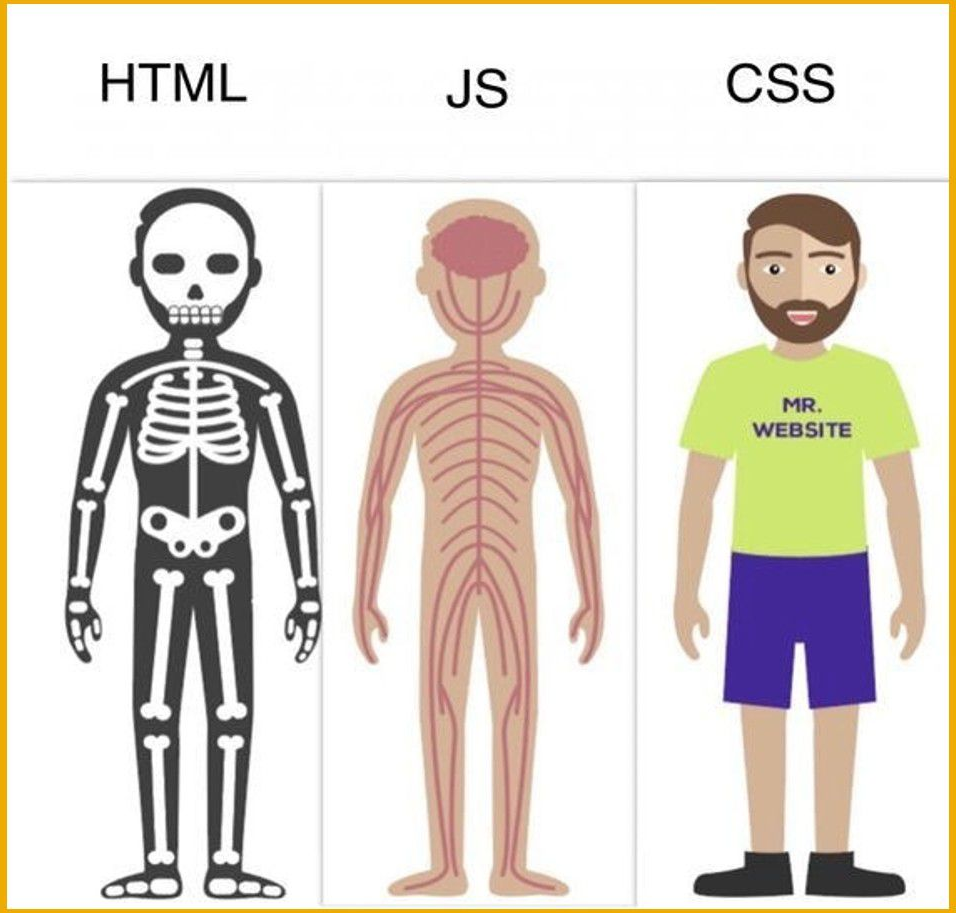
Svelte

Svelte provides a different method for building web applications. It helps developers build fast web pages and a great user interface.

8- LEARN JAVASCRIPT :

Here are some resources to learn JavaScript:

- JavaScript Tutorial - W3schools
- Learn JavaScript - Freecodecamp
- JavaScript Tutorial for Beginners - Programming with Mosh
- JavaScript Tutorial for Beginners - Clever programmers
- JavaScript Crash Course For Beginners - Traversy media
- React Tutorial - React
- Getting started with Vue - Vue Guide
- Build An Ember.js App - Program with Erik
- Getting started with PREACT - PREACT guide



9- PACKAGE MANAGERS :

A package manager is a tool that allows users to install, update, configure, and manage software packages and product dependencies. Examples of package managers are NPM and Yarn.

10- ACCESSIBILITY :

Web accessibility is the process of making your website usable by all people. Web accessibility is an essential part of front-end development. It is important that as a front-end developer, you need to build web pages with accessibility in mind. When you build with accessibility in mind, there is no restriction to anyone. People with disabilities, slow networks, visual impairments, hearing impairments can have access to all the information on the website without obstructions or difficulties.

Here are some resources to learn more about Accessibility:

- Web accessibility - Udacity
- Introduction to Web Accessibility - Class Central

11- PERFORMANCE :

Web performance refers to how long it takes for an application to be rendered in the browser as well as how responsive it is to user interaction.
For a better user experience, it is recommended that developers adopt different web optimization techniques. This includes using a Content Delivery Network (CDN) which is a strategically distributed web server that delivers contents to users based on location.

To get the exact performance data of your web application, you can make use of any of the below-listed tools:

- Loadview
- LoadNinja
- NeoLoad

12- TESTING YOUR APPS :

Testing involves scanning a piece of software for a potential bug during development. During testing, the tester performs an action on a piece of software while expecting a specific result. Possible things to test in an application includes security, functionality and ease of use amongst others.
Testing does not guarantee that a piece of software will function properly under any circumstance but it, however, provides us with information on how it could behave under certain conditions.

Testing tools and frameworks to try out include:

- Mocha
- Puppeteer
- Jasmine

12- WEBSITE DEPLOYMENT :

With the knowledge gotten from HTML, CSS, and JavaScript you can build simple web pages using those technologies. Most time after building, it remains in our computer or Github repo and there is no way people can see what you built. But with web deployments, you can host your simple application on the internet. It is super easy to do.
You can use tools like:

- GitHub pages,
- Netlify