

What next?

Hearty congratulations! You have taken the very first step towards Web Development. And if you were able to follow this course, you can be sure that your foundation is now pretty strong. Hence you can move on confidently in any direction of web development.

If you don't know how to proceed next, you are just like how I was 9 years ago. Which is why, this document is going to help you decide what you can get your hands dirty with next.

Strengthening HTML & CSS Skills

This course has given you a strong hold on these two languages but it hasn't covered everything required to become a professional. You have learnt the minimum basic stuff to put a simple website together. Here are some important concepts you could learn next. "Google" these concepts and you'll find enough tutorials to learn them from

HTML Forms

We all have filled up online forms – the smallest form being a login form where you enter your email and password and click "Login". These are also basic HTML elements. Follow the w3schools reference of [HTML forms](#) and you can quickly learn how to add these elements to your website. However, making the forms collect some data and put into database needs backend development – which I will cover further below.

CSS Transitions & Animations

You surely have seen buttons that fade on hover, images that slightly grow in size when you get your mouse on it and more such "animations". These are achieved using CSS transitions and animations. Check out this [great tutorial for beginners](#)

CSS Flexbox

The 'float' property is old CSS. It's important to learn this, but it's not used for building the page layouts anymore. CSS Flexbox makes it very easy to build the complex layouts of modern web design. [This free course on Udemy](#) gives you a fairly good understanding of CSS Flexbox in just 90 mins. Of course, feel free to explore even better courses or tutorials.

Media Queries

CSS media queries are a great way to give different styles to elements based on the device – hence a great way to achieve responsive web design. Watch this [YouTube video](#) to learn the basics in 5 minutes.

The language you cannot ignore – JavaScript (JS)

JavaScript is widely used for animations on web pages – one classic example is a slideshow (or a carousel). It's used in almost every single website alongside HTML & CSS to make it dynamic. When you scroll through Amazon website and more products keep loading on the same page without you having to refresh the page right? That's JavaScript in action.

These are simplest examples of client-side functionalities. JavaScript can also be used as a server-side languages lately – as if you needed more reasons to learn this language! Here is an article with [top 5 resources to learn JavaScript](#)

In the meantime, if you come across these terms – jQuery, React, Angular, VueJS, NodeJS – all these are JavaScript frameworks. These frameworks provide you with shortcuts of things that can be achieved using plain JavaScript. After learning JavaScript, you can choose to learn any of the above frameworks if you are interested.

Design websites faster with frameworks

While coding from scratch gives you control over every single pixel of your website, it's not always necessary. You can make use of design templates in frameworks to quickly develop responsive websites. Here is one very well known and widely used framework of HTML, CSS and JS

Bootstrap

Learning this framework is highly recommended if you want to feel the control of building websites from scratch and also save your time by using pre-built blocks of code. The latest version (at the time of writing this document) available is Bootstrap 4. Use this [w3schools tutorial](#) to get started with Bootstrap.

Tailwind CSS

Once you have designed quite a few websites using HTML and pure CSS, learn Tailwind CSS to make your development really fast. You can learn it from plenty of videos on my channel [playlist – Tailwind CSS](#).

Going Full Stack

If you want to build a dynamic website – one that can store data, do some operations on it and so on, you need to get into backend web development. There are front end web developers, backend web developers and those that can develop both front end and backend – called full stack web developers.

Some languages / terms you might have heard like PHP, Angular, MySQL, Ruby on Rails, NodeJS – are all used for backend web development. Here's [an article](#) that can give you a basic understanding of web development stacks and what to choose.

This document might have given you clarity or it might have left you thoroughly confused. Both are possible. If you are not sure how to proceed next, feel free to comment on the video. I will try and help you decide.

All the best for your future! Do share the course ahead with your friends / family.