**<https://100daysofwebdevelopment.com/next-steps>**

**How should you proceed?**

With the completion of our "Web Development - The Complete Guide" course, you learned about all the key concepts that make up web development.

You also learned way more than just the basics - with the course you also got a deep dive into advanced CSS, JavaScript, backend development, databases and much more!

But as a web developer, learning never stops. Your journey as a web developer actually just began! The web and web technologies evolve and there are many concepts, technologies and (niche) areas that you can now explore with all your newly gained knowledge.

It's time for your next 100 days (or more)!

Below, you find a thorough list of topics, ideas, concepts, courses and other resources that you might want to use to continue on your path as a web developer!

**Example Topics To Explore**

As a web developer, there are many areas and concepts that you can explore - here's a list of key topics you might want to search for!

* **Advanced HTML & CSS**

Learn more about HTML & CSS and dive into advanced elements & styles!

* **Advanced JavaScript**

Explore advanced JavaScript concepts and features to build even more exciting websites!

* **Advanced NodeJS**

Dive deeper into NodeJS (and frameworks like Express) to build more advanced websites.

* **Other Backend Languages**

NodeJS is a very popular backend programming language but it's not the only one. Also explore PHP, C#, Java or other languages!

* **SQL Databases**

Build up on your SQL knowledge and learn about more complex database structures and queries.

* **MongoDB / NoSQL Databases**

Dive deeper into MongoDB or alternative NoSQL database engines.

* **React, Angular & Vue.js**

Build more powerful frontend user interfaces by diving deeper into these modern JS frameworks.

* **Websockets**

Learn how to build real-time websites (e.g. livechat) with help of websockets - an alternative to the traditional request-response flow.

* **GraphQL & More on APIs**

Build more advanced APIs by diving deeper into (REST) APIs or explore alternatives like GraphQL (for building APIs).

* **Frontend Build Tools**

Optimize your frontend (JavaScript) code with build tools like Webpack & Babel.

* **Static Site Generators**

Build static websites (HTML, CSS & JS) with advanced tools and workflows like Hugo, Next.js or the "Jamstack" in general.

* **Explore Content Management Systems (CMS)**

When building websites, it is quite popular to use content management systems like WordPress. Learning how to configure and use these is a great idea therefore.

* **Search Engine Optimization (SEO)**

Become a SEO pro and optimize your content, descriptions and titles for maximum "findability"!

* **Optimize Performance**

Learn all about website performance optimization - both on the backend and frontend (e.g. code optimization, image optimization etc.).

* **Accessibility**

Grow your knowledge on accessibility and how to improve it (e.g. by picking proper HTML elements, adding attributes & more).

* **Automated Testing**

Dive into unit tests, integration tests and end-to-end tests to - well - test your website automatically (via code).

* **DevOps (CI / CD)**

Explore "Continuous Integration" and "Continuous Development" to automate deployments & more.

* **Build Mobile Apps**

It might not be obvious but you can actually build native mobile apps (Android + iOS) with web dev technologies and tools like Ionic.

**How To Learn & Dive Deeper**

Above, you find a long (non-exhaustive) list of interesting topics that you could explore to become a more advanced web developer.

But how **do** you dive deeper into these topics? Where can you learn more about the mentioned concepts? Which resources should you consult?

The answer is: The internet is full of helpful resources - courses, articles, blog posts, tutorials, YouTube videos, Tweets and much, much more!

Google (or any other search engine) is your friend!

You can always search for terms like "devops tutorial" or "what is devops" to start your journey into a brand-new topic. You'll find blog articles which then give you more ideas for topics and concepts you could explore.

That is how we typically learn new things: Step by step by picking up more and more knowledge via helpful articles and tutorials.

Taking a dedicated online course is of course another great way of learning a new concept or topic. The advantage of this approach is, that you don't have to manually dig deeper and deeper and search for more advanced concepts step by step.

Instead, with online courses, knowledge has already been organized for you and you can consume it in bite-sized chunks.

**Next Courses**

A great way of learning more about web development and specific topics of course is to dive into dedicated courses on a given topic.

(Video) courses allow you to learn all the key aspects of a topic in a relative short amount of time. And thankfully, the internet is full of free and premium courses!

If you liked our [**"Web Development - The Complete Guide" course**](https://acad.link/web-dev), you might also want to explore our other, topic-specific courses. :)

[**Explore All Our Courses**](https://academind.com/courses)

**Project ideas**

Only practice makes perfect! To grow as a web developer, you definitely should practice, dive into new problems, build (dummy) websites and grow your portfolio with more demos, examples and projects!

We often get asked which kind of projects you should build - hence we compiled a list of project ideas for you to explore.

* **Build this website on your own**

Try buildling a clone of this website on your own - without inspecting this website's code too much upfront.

* **Build another online shop**

Try building a highly simplified Amazon clone from the ground up - with user authentication, products, maybe also product management, a cart, orders & payments.

* **Build a (simple) social network**

Build a social networks site with signup & login, multiple users, friendships (i.e. user-to-user relations), a messaging system and likes.

* **Build your own personal portfolio website**

Build a website where you present yourself, your skills and projects. You can then use that site for job applications or to find clients!

* **Build a fake product landing page**

Build a landing (and explanation) page / website for a imaginary product. Take inspiration from existing pages and use what you learned to rebuild those sites.

* **Build a real blog website**

Build a real blog website - your own one or one that is open to the public (via user authentication). Store and show blog posts and optimize for search engines!