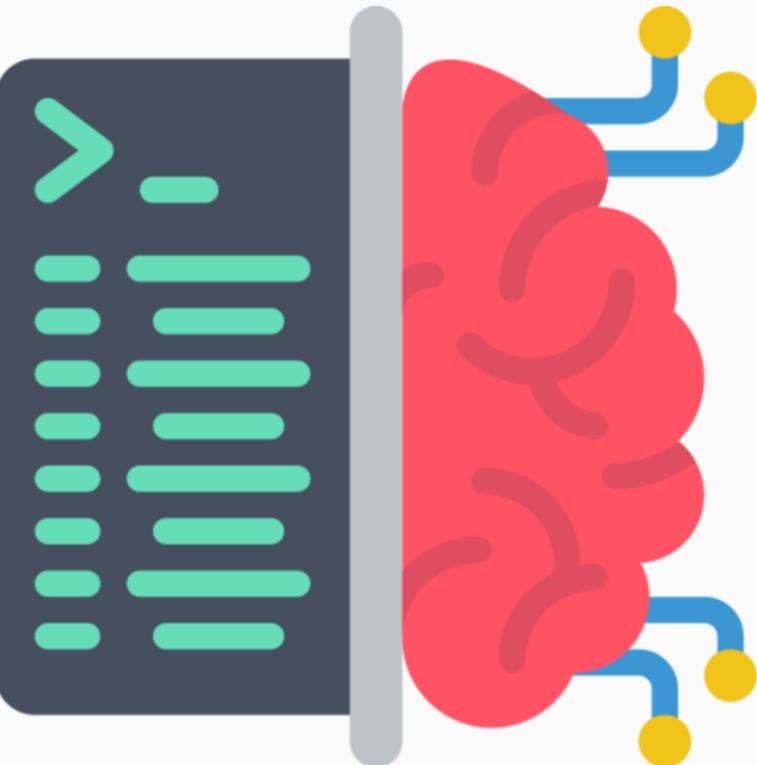


Phase One - Months 1-3

Foundations of Web Development

The initial phase of web development focuses on learning essential technologies that form the backbone of all websites: HTML, CSS, and JavaScript.



Learning Prep

Before learning a specific technology, figure out how you are going to learn as a self-taught developer.

There are many resources such as video courses, books, AI, YouTube, documentation, blogs, etc.

HTML

Structure Of The Web

HTML is the standard markup language for creating web pages. It defines the structure of a page using elements and attributes. Learn to build static websites that convey content effectively.

HTML



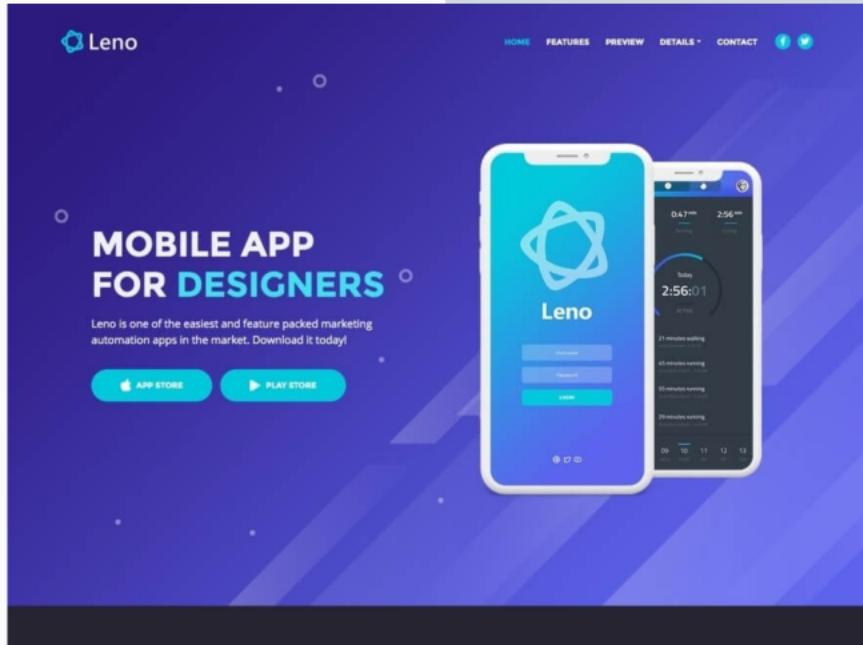
CSS

Styling The Web

CSS is the **stylesheet language** used to describe the presentation of HTML. Learn to apply styles, manage layouts, and create responsive designs, enhancing user experience across devices.

CSS





Project

Landing Page

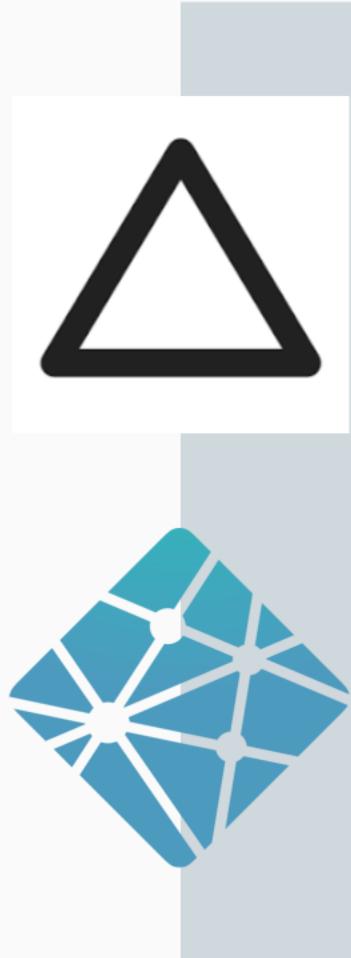
Create a landing page using the HTML elements & attributes along with the CSS selectors that you have learned.

This can be for any real or fictional product or service.

Version Control

Understanding Git and GitHub is essential for collaboration and code management. Gain hands-on experience with Git commands, branching, and merging to maintain project integrity.





Basic Deployment

Now that you can create a website, you need to be able to put it online.

There are really easy-to-use platforms like **Vercel**, **Netlify** and **Github Pages** that have generous free tiers for small projects and for testing.

A large, bold, black "JS" logo is centered on a yellow rectangular background. The yellow rectangle is positioned on the left side of the slide, overlapping a light gray vertical bar.

JS

Basic JavaScript

Adding Interactivity

Basic JavaScript encompasses fundamental programming concepts such as variables, loops, and functions. Learn to create interactive web pages by manipulating the Document Object Model (DOM).

Project

Simple Interactivity

Create some simple websites/apps
that have interactive elements and
work with the DOM.

Which language runs in a web
browser?

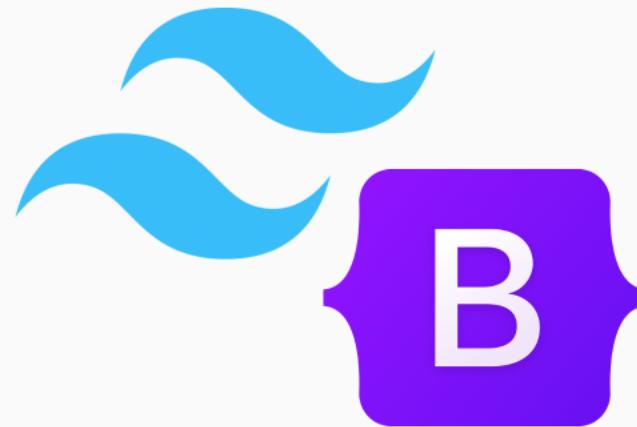
- Java
- C
- Python
- JavaScript

Submit

Optional

CSS Framework

CSS frameworks are used to streamline the process of styling websites/apps by providing a set of pre-defined styles



1

Phase One - Months 1-3

Foundations of Web Development

The initial phase of web development focuses on learning essential technologies that form the backbone of all websites: HTML, CSS, and JavaScript.

Learning Prep
Before learning a specific technology, figure out how you are going to learn as a self-taught developer.
There are many resources such as video courses, books, AI, YouTube, documentation, blogs, etc.

HTML Structure Of The Web
HTML is the standard markup language for creating web pages. It defines the structure of documents, including headings, paragraphs, and other content. Learn to build static websites that convey certain effects.

CSS Styling The Web
CSS is the stylesheet language used to describe the presentation of HTML. Learn how to use CSS to create responsive designs, enhancing user experience across devices.

Landing Page
Create a landing page using the HTML, CSS, and JavaScript concepts that you have learned.
This can be for a new or finished product or service.

Version Control
Understanding Git and GitHub is essential for collaboration and code management. Learn how to use Git commands, branching, and merging to maintain project integrity.

Simple Interactivity
Create some simple websites/recipes that have interactive elements and work with the DOM.

Basic JavaScript Adding Interactivity
Basic JavaScript encompasses fundamental programming concepts such as variables, loops, and functions. Learn to create interactive websites by manipulating the Document Object Model (DOM).

CSS Framework
CSS frameworks are used to streamline the process of styling websites by providing a set of pre-defined styles.

Basic Deployment
Once you can create a website, you need to be able to put it online.
There are many ways to host websites, but the most common are free services like GitHub Pages for small projects and Heroku for testing.

Phase Two - Months 4-8

Frontend Essentials

Enhance your web development capabilities by getting into more advanced JavaScript, APIs, and frontend frameworks

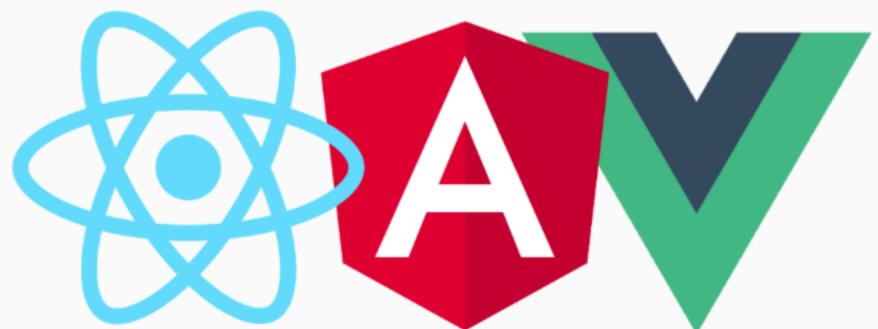
JavaScript Deep Dive

Master ES6+ features such as arrow functions, destructuring, and modules.

Working with APIs using fetch and async/await will enhance data handling capabilities and improve asynchronous operations.

You will also be learning about NPM (Node Package Manager)





Frontend Framework

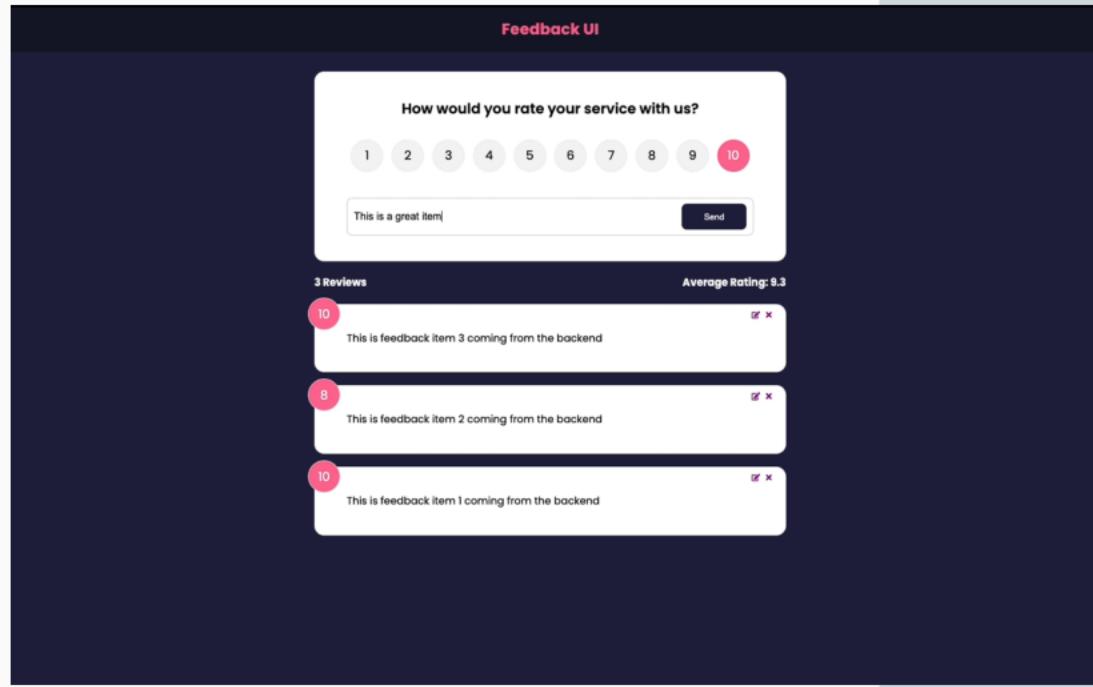
Choose a frontend framework like React/Vue/Svelte/Angular. This allows for efficient component-based development. Understanding props, state management, and the lifecycle of components is crucial for building dynamic, interactive user interfaces.

State Management

State management is essential in handling data and UI changes.

This will allow you to maintain consistent application states across components, crucial for complex applications.





Project **Interactive UI**

Create applications with an interactive UI, preferably with a FE framework

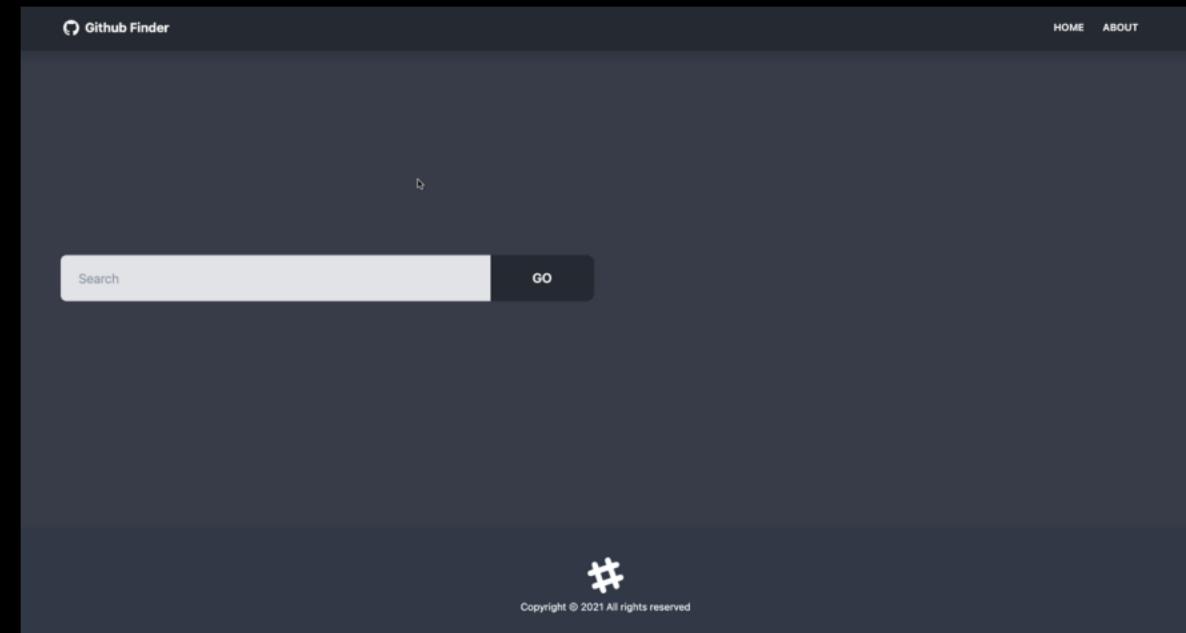
An example would be a calorie tracker, event scheduler, budgeting app, etc

Project

API Project

Create a project that works with a 3rd-party API. It could be something like the GitHub API or a Weather API.

Understand Fetch as well as HTTP methods and status codes



Phase Three - Months 8-12

Backend & Full Stack

Transitioning from frontend to backend skills is crucial for becoming a proficient full-stack developer, allowing the integration of user interfaces with server-side logic.

Backend Fundamentals

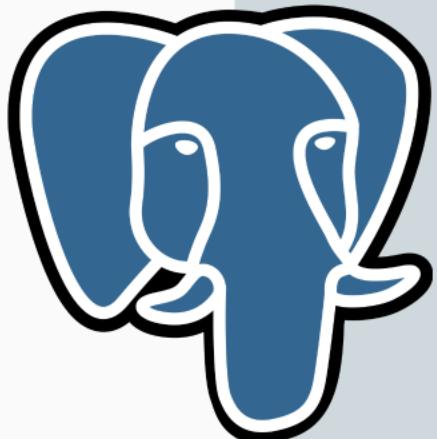
Learn the basics of Node.js/Express and learn about simple HTTP servers. Understand RESTful APIs and communication between frontend and backend services.



Backend Framework

A backend framework is a must for building complex APIs, services and backend applications

Express 



Databases

Databases store, retrieve, and manage data efficiently. SQL databases (like MySQL or PostgreSQL) support structured data with relationships, while NoSQL (like MongoDB) offers flexibility for unstructured data. Mastering CRUD operations, joins, and indexing is essential for effective data manipulation.



Database ORM/ODM

Writing raw SQL queries in your code is a thing of the past. We use **Object Relational Mappers** to interact with the database from within our project's code.





DigitalOcean



Fullstack Deployment

Deploying a fullstack or backend project is a bit more complex because you need a server and database.

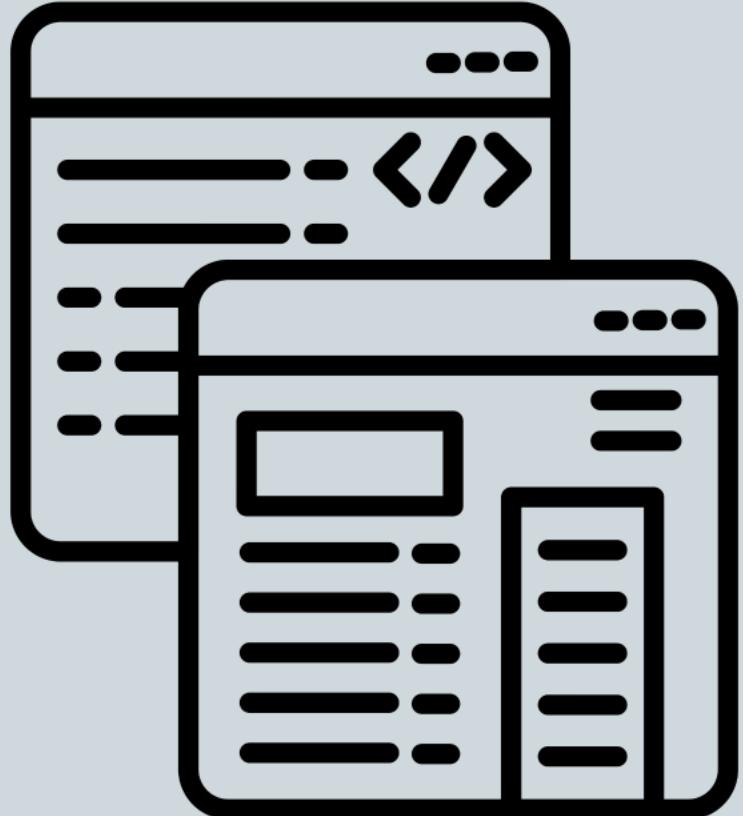
You have cloud databases like **Digital Ocean** and Linode as well as PaaS like **Heroku** and **Render**.

Project **RESTful** **API**

Create a REST API that does CRUD. This will teach all about routing, database integration, project structure and more.

Suggested Course:
Node.js API Masterclass





Frontend/Backend Integration

Now that you know how to build a frontend app and a backend API, you can learn how to put it all together.



Authentication

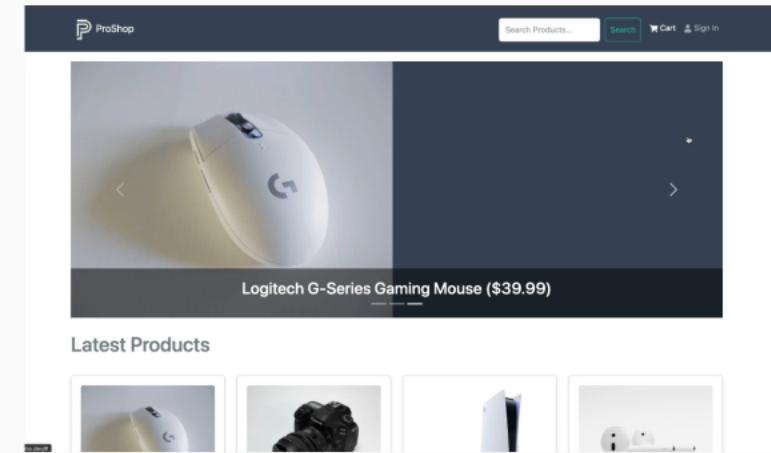
Authentication is vital for securing applications, ensuring only authorized users access resources. Implementing JWT or OAuth allows for secure token-based authentication, while understanding security practices such as password hashing is crucial in protecting user data.

Project

Full Stack App

Build an app that has a frontend SPA and a backend with Node.js. The frontend should make requests to your backend API and update the state.

Suggested Course:
MERN Ecommerce





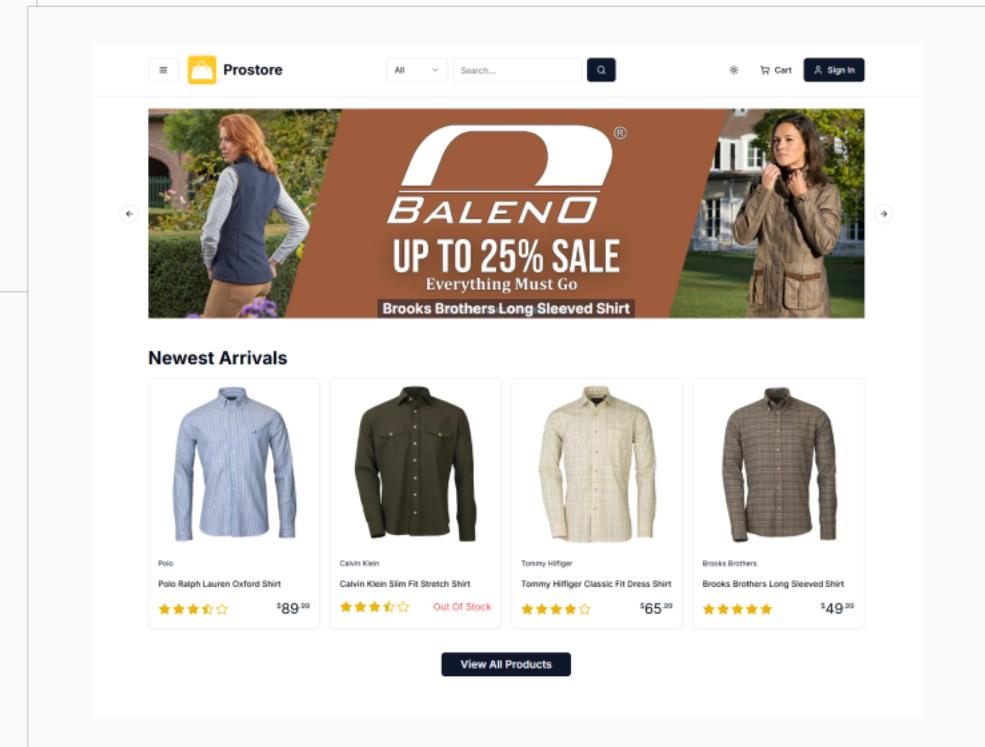
SSR (Server-Side Rendered) Websites

SSR is becoming more and more popular with platforms like Next.js. Components are rendered on the server, rather than on the client.

This offers advantages like file-based routing, SEO, performance, easier data fetching, etc.

Project SSR Application

Build a full stack application using SSR. This could be Next.js, Nuxt, Remix, SvelteKit, etc



Phase Four - Months 12-14

Career Development

This phase focuses on securing a job in web development by building a strong portfolio and mastering professional branding.



Build a Strong Portfolio

A strong portfolio showcases 3–5 well-rounded projects that demonstrate technical skills and creativity. Ensure projects highlight responsive design, full-stack capabilities, API integration, and authentication features to impress potential employers.



Social Media & Web Presence

Make yourself visible and known online. It will increase the chances of you being more discoverable and reputable.

In addition to your website/portfolio, an optimized LinkedIn profile displays projects through GitHub, engages connections, and highlights relevant industry experience.

Twitter will also allow you to share work and connect with other people in the industry.



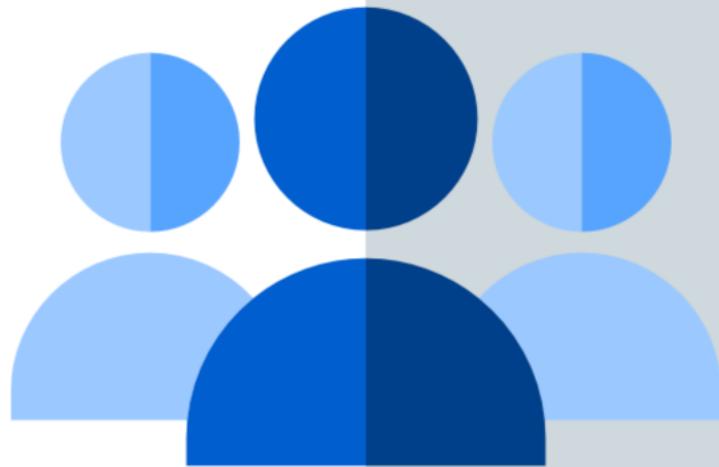
Apply for Jobs

Target roles that match your skillset, whether they focus on frontend, backend, or full-stack development; tailor applications and prepare for interviews.



Freelancing (Optional)

Freelancing allows developing practical experience while building a client base. Use platforms like Upwork or Fiverr to find small projects, enabling further development of skills in real-world scenarios while earning income.



Soft Skills & Collaboration

Once you get a job or start freelancing, working with other developers and/or clients is a skill in itself. Learn to work with others and become a productive team member.

Phase Five - Months 14-17

Continuous Learning

Working as a developer, you will constantly be learning. Master advanced concepts in frontend, backend, and deployment that are crucial for developing scalable applications.

Advanced Frontend

Continue your learning with the frontend and master your framework of choice as well as the ecosystem including SSR, state management libraries, high-order components, build tools, etc



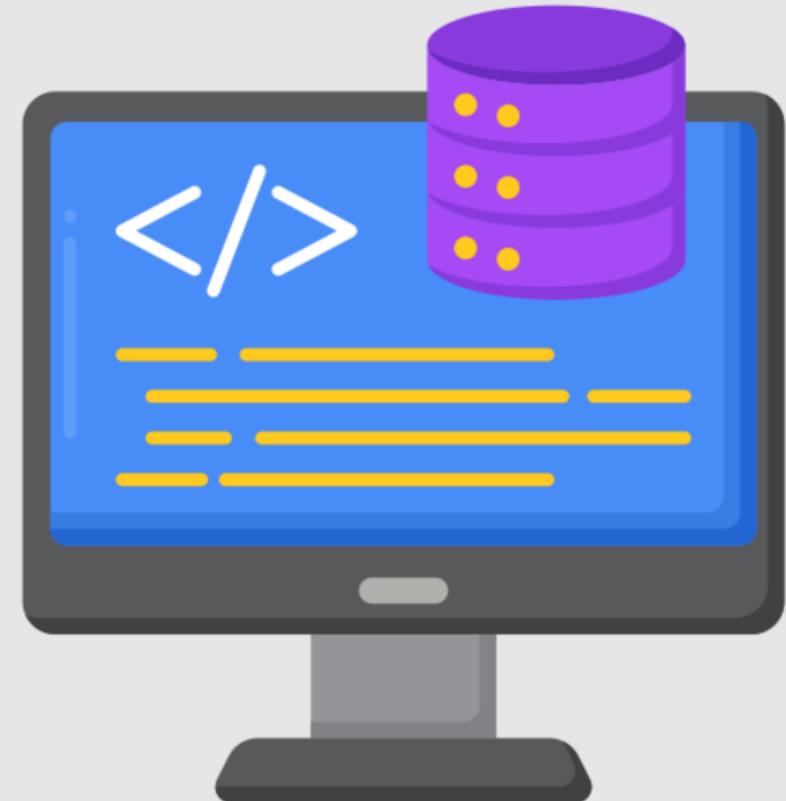


Testing & Debugging

Learn robust testing practices including writing unit tests with Jest and using tools like Sentry for debugging and error tracking.

Advanced Backend

Continue your learning with the backend and master Express and learn about topics like microservices.



DevOps and Deployment

Learn Docker for containerization, ensuring environment consistency and easier application deployment across various systems. CI/CD pipelines automate testing and deployment processes, enhancing reliability and reducing time to market for software releases.





Project **SaaS Application**

Developing a SaaS application involves creating multi-tenant architectures, implementing secure authentication. Focus on building scalable solutions that can handle varying user loads effectively while maintaining performance and security.

