

■ 100 Days Web Development Challenge

PHASE 1: HTML & CSS (Day 1–15)

Goal: Learn HTML & CSS basics and build your first project.

Day 1: Install VS Code, set up browser (Chrome/Edge), learn basic structure.

Day 2: Headings, paragraphs, links, images.

Day 3: Lists (ordered, unordered, nested).

Day 4: Tables + attributes.

Day 5: Forms (input, textarea, select, button).

Day 6: Semantic tags (header, footer, section, article, nav).

Day 7: CSS basics – inline, internal, external CSS.

Day 8: Colors, backgrounds, fonts, text styling.

Day 9: Box model – margin, border, padding.

Day 10: CSS selectors (class, id, universal, pseudo-classes).

Day 11: Positioning (static, relative, absolute, fixed, sticky).

Day 12: Flexbox basics.

Day 13: CSS Grid basics.

Day 14: Responsive design (media queries).

Day 15 Project: Build a Personal Portfolio (HTML + CSS only).

PHASE 2: JavaScript Basics (Day 16–35)

Goal: Learn JavaScript fundamentals and build interactive projects.

Day 16: JS setup (linking script), variables (var, let, const).

Day 17: Data types (number, string, boolean, null, undefined).

Day 18: Operators (+, -, *, /, %, ++, --).

Day 19: Conditionals (if-else, switch).

Day 20: Loops (for, while, do-while).

Day 21: Functions (declaration, expression, arrow).

Day 22: Arrays (create, access, modify, methods like push, pop).

Day 23: Objects (key-value pairs, methods).

Day 24: DOM basics – getElementById, querySelector.

Day 25: DOM – changing text, HTML, CSS via JS.

Day 26: DOM – create, append, remove elements.

Day 27: Events – click, input, mouseover.

Day 28: Event listeners, preventDefault.

Day 29: ES6 – template literals, default parameters.

Day 30: ES6 – spread/rest, destructuring.

Day 31: JSON & localStorage.

Day 32: Error handling (try, catch).

Day 33: Timing functions (setTimeout, setInterval).

Day 34 Project: To-Do List App (Save tasks in localStorage).

Day 35 Project: Calculator (Basic Arithmetic).

PHASE 3: Advanced CSS + UI (Day 36–50)

Goal: Master modern CSS & build beautiful responsive designs.

Day 36: Pseudo-classes (hover, focus, nth-child).

Day 37: Transitions (hover effects).

Day 38: Animations (@keyframes).

Day 39: Transformations (scale, rotate).

Day 40: Responsive images & units (em, rem, vh, vw, %).

Day 41: CSS Variables.

Day 42: CSS Flexbox practice (navbar, footer).

Day 43: CSS Grid practice (photo gallery).

Day 44: CSS Frameworks – Bootstrap basics.

Day 45: Bootstrap components (navbar, cards, forms).

Day 46: Bootstrap grid system.

Day 47: Tailwind CSS basics.

Day 48: Tailwind CSS responsive design.

Day 49 Project: Build a Responsive Blog Website (HTML, CSS/Bootstrap).

Day 50 Project: Landing Page for a Startup (Responsive + Styled).

PHASE 4: JavaScript Advanced + APIs (Day 51–70)

Goal: Work with APIs, async functions, and build real-world JS projects.

Day 51: Fetch API basics.

Day 52: Fetch API with JSON.

Day 53: Promises (then, catch).

Day 54: Async/await.

Day 55: API error handling.

Day 56: Forms + JS validation.

Day 57: Modular JS (import/export).

Day 58: Practice small APIs (jokes API).

Day 59: Small project – Random Joke Generator.

Day 60: Small project – Quote Generator with API.

Day 61: Small project – Quiz App (MCQs).

Day 62: Small project – Weather App (OpenWeather API).

Day 63: Small project – Currency Converter (Exchange Rate API).

Day 64–66 Project: Weather Dashboard with Search & History.

Day 67–70 Project: Recipe Finder App (fetch meals API).

PHASE 5: Backend & Databases (Day 71–85)

Goal: Learn backend development and connect to a database.

Day 71: Introduction to backend (how frontend & backend connect).

Day 72: Node.js basics (npm, modules).

Day 73: Creating an HTTP server in Node.js.

Day 74: Express.js basics (routes, middleware).

Day 75: Express.js advanced (params, queries).

Day 76: REST API concepts (GET, POST, PUT, DELETE).

Day 77: MongoDB introduction (NoSQL database).

Day 78: MongoDB CRUD operations.

Day 79: Connect MongoDB with Node.js.

Day 80: Mongoose basics (schemas, models).

Day 81: Postman testing for APIs.

Day 82: Authentication basics (hashing passwords).

Day 83: JWT authentication.

Day 84 Project: Simple Blog Backend (CRUD posts, MongoDB).

Day 85 Project: User Authentication System (Register + Login).

PHASE 6: Full-Stack Projects (Day 86–100)

Goal: Combine frontend + backend + database into full-stack apps.

Day 86–88 Project: Notes App with User Login (Full-Stack).

Day 89–91 Project: Task Manager (CRUD + Authentication).

Day 92–95 Project: E-commerce Product Catalog (Frontend + Backend + Database).

Day 96–98 Project: Chat App (WebSockets + Database).

Day 99–100 Final Project: Full-Stack Portfolio Website (Showcase all projects).

■ After 100 days, you'll have:

- Strong HTML, CSS, JS, and responsive design skills.
- Backend + database experience (Node.js, Express, MongoDB).
- 7–10 projects (frontend + full-stack) for your portfolio.