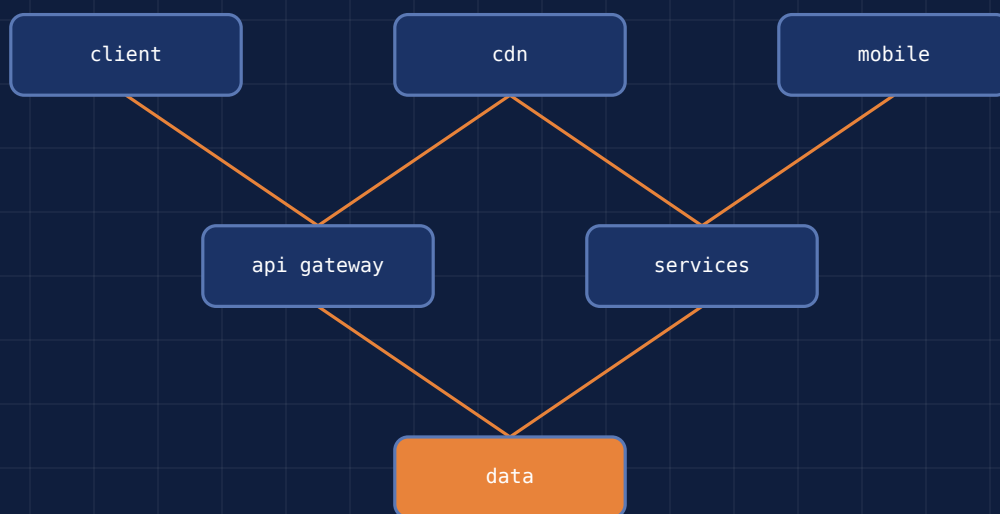




Full-Stack Web Architect

Master Modern Web Development, System Design & Scalable Applications

HTML · CSS · JavaScript · TypeScript · Node · React · Next.js · SQL · MongoDB
Docker · Cloud · Architecture · Design Patterns · System Design · Security

SUMAN BISWAS

Amazebazaar Publications · amazebazaar.com

Copyright

Full-Stack Web Architect *Master Modern Web Development, System Design & Scalable Applications*

First Edition, 2026

Copyright © 2026 Suman Biswas. All rights reserved.

Published by **Amazebazaar Publications** · amazebazaar.com

No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means — electronic, mechanical, photocopying, recording, scanning, or otherwise — without the prior written permission of the author, except for brief quotations used in reviews and articles.

The code examples in this book may be used freely in your own personal and commercial projects. You may not republish the examples as a collection, or redistribute this book, in whole or in part.

The official digital edition of this book is distributed through amazebazaar.com and authorised marketplaces such as Amazon Kindle Direct Publishing and Gumroad. If you received this file from anywhere else, please download the official copy from **amazebazaar.com** — it supports the author and guarantees you get the latest corrected edition.

All product names, logos, and brands mentioned in this book (including but not limited to React, Next.js, Node.js, MongoDB, PostgreSQL, MySQL, Docker, AWS, Azure, Google Cloud, GitHub, Stripe, Razorpay, Redis, Kafka, and RabbitMQ) are property of their respective owners. Their use here is for identification and educational purposes only and does not imply endorsement.

Author contact: developersuman.contact@gmail.com

Disclaimer

This book is written to teach. Every example has been written carefully and checked, but software moves fast. Library APIs change, cloud providers rename their services, and prices shift from month to month. Treat version numbers, prices, and salary ranges in this book as a snapshot of 2026, not as a promise.

The author and publisher make no warranties about the completeness, reliability, or suitability of the information for any particular purpose. Any action you take based on this book — deploying code to production, handling customer payments, storing personal data, or making career decisions — is strictly at your own risk.

Security chapters describe attacks so you can defend against them. Only ever test techniques like SQL injection or XSS on systems you own or have written permission to test. Testing someone else's system without permission is illegal in most countries.

Salary figures are indicative ranges gathered from public job boards and industry surveys. Your actual offer will depend on location, company, skills, and negotiation.

Preface

When I started learning web development, I kept running into the same problem. Every tutorial taught one thing in isolation. One course taught HTML. Another taught React. A third taught Docker. Nobody showed me how all the pieces fit together — how a click on a button in a browser in Kolkata becomes a row in a database in Mumbai, and how that same system survives when ten thousand people click at once.

This book is my answer to that problem.

Full-Stack Web Architect walks you through the entire journey. We start with how the internet actually moves bytes from one machine to another. We build web pages with HTML and CSS, make them interactive with JavaScript, make that JavaScript safer with TypeScript, and then cross over to the server with Node.js, Express, and real databases. From there, we build modern front ends with React and Next.js, secure them, test them, ship them with Docker and CI/CD, and run them in the cloud. Finally, we zoom out and think like architects: patterns, system design, scaling, security, SaaS, and career.

Who this book is for

- **Beginners** who want one clear path instead of a hundred scattered tutorials.
- **Intermediate developers** who can build features but want to understand the *why* behind architecture decisions.
- **Senior developers and engineers** who want a structured refresher on system design, patterns, and security before an interview or a promotion.
- **Computer science students** who want the practical side their syllabus skips.
- **Freelancers and startup founders** who need to build, ship, and scale real products without a big team.

How each chapter works

Every chapter follows the same rhythm so you always know where you are:

1. **Introduction** — why this topic matters, in plain language.
2. **Learning Objectives** — what you will be able to do by the end.
3. **Explanation** — the concepts, with real examples and text diagrams.

4. **Code Examples** — commented, production-minded code.
5. **Best Practices** and **Common Mistakes** — the lessons people usually learn the hard way.
6. **Interview Questions** — the questions that actually get asked, with short answers.
7. **Mini Project** — a small build to lock in the skills.
8. **Summary** and **Exercises** — a quick recap and practice.

You will also see coloured callout boxes. Green boxes are **Pro Tips**. Red boxes are **Common Pitfalls**. Blue boxes are **Notes**. Orange boxes are **Real-World** stories.

How to read this book

If you are a beginner, read it front to back. The parts build on each other. If you already have experience, jump to the part you need — each chapter is designed to stand on its own, with pointers back to earlier material when needed.

Most importantly: type the code. Don't copy-paste it. The muscle memory of typing, breaking, and fixing code is how you actually learn. Reading about swimming has never taught anyone to swim.

Tools you will need

- A computer with Windows, macOS, or Linux.
- **Node.js** (the current LTS version) and **npm**.
- **Visual Studio Code** or any editor you like.
- **Git** and a free **GitHub** account.
- **Docker Desktop** (for Part 14 onwards).
- A modern browser — Chrome, Edge, or Firefox — with its developer tools.

That's it. Everything else we install along the way.

Let's build something.

— **Suman Biswas**, Kolkata, 2026

Acknowledgements

No book is written alone.

Thank you to my family, who put up with long nights and endless talk about servers and databases at the dinner table.

Thank you to my teachers at the University of Kalyani, who taught me how to think, argue, and write clearly — skills that turned out to matter in software as much as in political science.

Thank you to the trainers and classmates at Webskitters Academy, where my journey into UI/UX design and front-end development truly began.

Thank you to the open-source community — the thousands of developers who write, document, and maintain the tools in this book for free. Every chapter here stands on their work.

Thank you to every reader of AmazeBazaar Publications who sent feedback, corrections, and questions. Your messages shaped this book.

And finally, thank you — for choosing to learn. The web is built by people who decided to learn one more thing. Welcome to that group.

Table of Contents

Copyright	2
Disclaimer	3
Preface	4
Acknowledgements	6
Part 1 — Web Fundamentals	11
1. How the Internet Works: Clients, Servers & HTTP	12
2. DNS, Domains, Hosting, SSL & CDNs	22
3. How Browsers Render Pages & Web Security Basics	29
Part 2 — HTML5	35
4. HTML5 Foundations, Semantic HTML & SEO Markup	36
5. Forms, Tables, Multimedia, Canvas, SVG & Accessibility	43
Part 3 — CSS3	51
6. CSS Foundations: Selectors, Cascade, Box Model, Positioning & Variables	52
7. Flexbox, Grid, Responsive Design & Modern Layouts	60
8. Animations, Transitions & the Landing Page Project	66
Part 4 — JavaScript	73
9. JavaScript Fundamentals: Variables, Types, Operators & Functions	74
10. Objects, Arrays & Functional Programming	81
11. The DOM, Events & Interactive Projects	87
12. Asynchronous JavaScript, Fetch, Modules & OOP	95
Part 5 — TypeScript	104
13. TypeScript: Types, Interfaces, Generics & Advanced Types	105

Part 6 — Git & GitHub	113
14. Git & GitHub: Version Control, Branching, Pull Requests & Open Source	114
Part 7 — Node.js	120
15. Node.js: Runtime, Modules, File System, Streams, Events, Process & Buffers	121
Part 8 — Express.js	129
16. Express.js: Routing, Middleware, MVC & REST APIs	130
17. Authentication & Authorization with JWT: Building a Complete API	137
Part 9 — Databases	144
18. SQL Databases: MySQL & PostgreSQL — CRUD, Relations, Indexing & Transactions	145
19. MongoDB & NoSQL: Documents, Aggregation, Indexes & Transactions	153
Part 10 — React	160
20. React Fundamentals: Components, JSX, Props, State & Hooks	161
21. Advanced React: Routing, Context, State Management, Data Fetching, Performance & Admin Dashboard	169
Part 11 — Next.js	178
22. Next.js: App Router, Rendering Strategies, Server Actions, Middleware, Auth & Deployment	179
Part 12 — Authentication	188
23. Authentication Deep Dive: Sessions, Cookies, JWT, OAuth 2.0, Google & GitHub Login	189
Part 13 — APIs	197
24. REST, GraphQL & WebSockets: Building and Consuming APIs and Real-Time Apps	198
Part 14 — DevOps	206
25. Linux Essentials & Nginx for Web Developers	207
26. Docker, Docker Compose, CI/CD & GitHub Actions	213

Part 15 — Cloud	220
27. Cloud Platforms: AWS, Azure & Google Cloud — Services and Deployment	221
Part 16 — Software Architecture	228
28. Software Architecture: MVC, Layered, Clean, Hexagonal, Monolith, Microservices, Event-Driven & Serverless	229
Part 17 — Design Patterns	238
29. Design Patterns in JavaScript & TypeScript: Singleton, Factory, Builder, Strategy, Observer, Repository & Dependency Injection	239
Part 18 — System Design	247
30. System Design Fundamentals: Scalability, Load Balancing, Caching, Redis, Queues, Kafka, RabbitMQ, Sharding, Replication & CDN	248
31. System Design Case Studies: Facebook, Netflix, YouTube, WhatsApp & Amazon	258
Part 19 — Security	265
32. Web Application Security: OWASP Top 10, SQL Injection, XSS, CSRF, CORS, CSP & API Security	266
Part 20 — Testing	275
33. Testing: Unit, Integration & End-to-End with Jest and Cypress	276
Part 21 — Performance Optimization	284
34. Web Performance: Lazy Loading, Code Splitting, Tree Shaking, Images, Caching & Lighthouse	285
Part 22 — SEO	292
35. Technical SEO: Core Web Vitals, Structured Data, Open Graph, Robots & Sitemaps	293
Part 23 — AI in Web Development	301
36. AI in Web Development: AI APIs, Chatbots, LLM Integration, Image Generation, Voice & AI Coding Tools	302
Part 24 — Building SaaS	312

37. Building SaaS: Architecture, Multi-Tenancy, Authentication, Billing & Dashboards	313
Part 25 — Capstone Projects	324
38. Capstone Projects I: Social Media, E-Commerce, LMS & Hospital Management	325
39. Capstone Projects II: Hotel Booking, CRM, News Portal & AI Chat App . . .	333
Part 26 — Career Guide	339
40. Your Career: Frontend, Backend, Full-Stack & Software Architect — Portfolio, Resume, Interviews, Salaries, Freelancing & Remote Work	340
Part 27 — Bonus Chapters	349
41. Web Accessibility (WCAG 2.2) & Progressive Web Apps	350
42. Payment Gateway Integration (Stripe & Razorpay) & Internationalization (i18n)	358
43. Logging, Monitoring, Feature Flags & Versioning Strategies	365
44. Scaling to Millions of Users: A Step-by-Step Journey	373
45. AI-Assisted Development Workflows	380
46. Real-Time Collaborative Apps & the API Integration Playbook	386
Afterword	392
Glossary	393
Acronyms Index	406
Resource List	411
Recommended Books	413
Learning Roadmap	414
Interview Cheat Sheet	416
Appendix	421
Index	426