Web Development Roadmap Workshop: One-Page Overview

Workshop Title:
Modern Web Development: Frontend-First Roadmap for 2026

Duration: 2-3 hours (including break)
Audience: Aspiring and junior web developers
Focus: Practical, career-oriented roadmap from fundamentals to production

---

Workshop Structure & Timing

Part	Topic	Duration
1	The Frontend Developer's Role & Current Landscape	10 min
2	Level 1: Core Foundations (HTML, CSS, JavaScript)	25 min
3	Level 2: React Ecosystem & Modern Frameworks	30 min
4	State Management & Styling Solutions	20 min
BREAK		10 min
5	Level 3: Performance, Testing & Advanced Patterns	25 min
6	Backend Essentials for Frontend Developers	20 min
7	Deployment, Tools & Real Interview Expectations	15 min
8	Learning Path Strategy & Q&A	15 min

Key Learning Objectives

- Master the three-tier progression: Beginner -> Intermediate -> Advanced
- Understand JavaScript fundamentals deeply (event loop, closures, async patterns)
- Learn React and Next.js as the modern frontend standard
- Navigate the ecosystem of state management, styling, and tooling choices
- Prepare for real 2026 frontend interviews (not 2022 interviews)

- Build a personalized learning roadmap based on current skill level

---

Core Concepts Covered

Foundation Layer:

- Semantic HTML, CSS Grid/Flexbox, JavaScript fundamentals
- Event loop, microtasks vs macrotasks, scope & closures
- Browser rendering pipeline and Web APIs
- Responsive design and accessibility basics

Framework & Tools:

- React (Hooks, re-renders, reconciliation, memoization)
- Next.js (SSR/SSG/ISR, Server vs Client Components)
- State management (Context API, Redux, Zustand, React Query)
- TypeScript for type safety
- Tailwind CSS and component libraries

Advanced Topics:

- Performance optimization (code splitting, lazy loading, Core Web Vitals)
- Testing strategies (Jest, React Testing Library, Vitest)
- Build tools (Vite, bundlers, tree-shaking)
- Security basics (XSS, CSRF, CSP)
- Service Workers and PWAs

Backend Essentials:

- Node.js, Express, Nest.js fundamentals
- REST vs GraphQL APIs
- Authentication (JWT, OAuth)
- Databases (PostgreSQL, MongoDB) and ORMs (Prisma, Drizzle)

Technologies Covered by Category

Must-Know Stack:

- React.js + Next.js
- TypeScript
- Tailwind CSS
- Git & GitHub

State & Data:

- Redux Toolkit / Zustand
- React Query
- Axios
- Zod / Yup

Backend & APIs:

- Node.js + Express
- PostgreSQL + Prisma
- REST & GraphQL
- WebSocket / Socket.io

Tools & Deployment:

- Vite
- NPM / PNPM
- Vercel
- Jest + React Testing Library

Interactive Elements

- Live coding: Build a React component with proper patterns
- Discussion: When to use which state management solution
- Activity: Identify performance bottlenecks in sample code
- Q&A on career paths and interview preparation

Current Interview Reality (2025-2026)

What's Expected Now:

- Deep JavaScript fundamentals (not just syntax)
- Build functional components from scratch (no libraries)
- React with engineering depth (why re-renders happen, how reconciler works)
- Next.js as default (even for "React" roles)
- Frontend system design (LLD + HLD)

Common Interview Formats:

- Machine coding (modals, infinite scroll, debounce/throttle)
- System design (component architecture, performance tradeoffs)

- React deep-dives (optimization, rendering patterns)
- Live debugging and refactoring

Deliverables for Participants

- Three-tier learning roadmap (Beginner -> Intermediate -> Advanced)
- Technology decision matrix (when to use what)
- Interview preparation checklist
- Resource list (documentation, courses, communities)
- Project ideas for portfolio building

Learning Path Recommendations

Beginners (0-6 months):
HTML/CSS/JavaScript fundamentals -> Build 5-10 static projects -> Learn React basics

Intermediate (6-18 months):
Master React -> Add Next.js -> State management -> Build full-stack projects

Advanced (18+ months):
System design -> Performance optimization -> Multiple frameworks -> Mentoring

---

Contact & Preparation

Presenter: Mehdi Maleki
Email: mosioc79@gmail.com
Pre-workshop: Participants should have basic HTML/CSS knowledge and a code editor installed (VS Code recommended)