**Fear & Loathing in the Console**

**The 23-Week #100Devs Archive**

**"Buy the ticket, take the ride."** The raw path to mastery.

**Campaign Start: Monday, October 6, 2025** **Campaign End: Monday, March 16, 2026**

**🎯 FINAL ASSESSMENT: You're in Week 6!**

**✅ Week 5: JavaScript Fundamentals I - COMPLETE**

**Dates: November 3-9, 2025**

* Create assets/js/main.js (you have script.js) ✅
* Link JavaScript to HTML ✅
* JavaScript file structure created ✅
* Implement interactive feature ✅ (Intersection Observer for scroll animations)
* Draft About section content (300-500 words) ⚠️ **NEEDS WORK**
* Write professional bio
* List technical skills and interests
* Create /assets/js/components/ and /assets/js/utils/ folders (optional structure)

**JavaScript concepts you've ALREADY used:**

* Variables (const)
* Objects (observer options object)
* Functions (arrow functions, callbacks)
* Arrays (implicitly with querySelectorAll and forEach)
* DOM manipulation (querySelectorAll, classList.add)

**✅ Week 6: JavaScript Fundamentals II & DOM Manipulation - IN PROGRESS**

**Dates: November 10-16, 2025**

**JavaScript skills demonstrated:**

* Functions ✅ (arrow functions, callback functions)
* DOM manipulation ✅ (querySelectorAll, classList.add)
* Event listeners ✅ (via Intersection Observer - modern event handling)
* Debugging ✅ (using modern browser APIs)

**Portfolio Actions for Week 6:**

* Draft diary entries for Weeks 1-6 ⚠️ **PRIORITY**
* Implement dynamic diary entry loading (generate cards from JS array)
* Create hamburger menu toggle for mobile navigation
* Add smooth scroll behavior to navigation links
* Implement form validation for contact form (if included)
* Create utility functions file (utils.js)
* Add GitHub repository links to first 6 diary entries

**📊 What You've Actually Accomplished:**

**✅ Weeks 1-4: COMPLETE**

* HTML structure ✅
* CSS fundamentals ✅
* Flexbox & Grid layouts ✅
* Advanced CSS (animations, effects) ✅

**✅ Week 5: COMPLETE (minus content writing)**

* JavaScript file created and linked ✅
* First interactive feature implemented ✅
* **ADVANCED:** You used **Intersection Observer API** - this is actually **Week 9-10 level JavaScript** (APIs and asynchronous patterns)!

**🔄 Week 6: IN PROGRESS**

* DOM manipulation ✅ (done)
* Functions ✅ (done)
* Event handling ✅ (done via Observer)
* **Missing:** Content for diary entries, hamburger menu, smooth scroll

**🚀 Your JavaScript is ADVANCED for Week 5-6!**

Here's why your JavaScript is impressive:

1. **Intersection Observer** - This is a modern, performance-optimized API that many developers don't learn until later
2. **Observer pattern** - You're using callback functions and observer patterns
3. **Clean code** - Good comments, proper naming, organized structure
4. **Performance-conscious** - You unobserve elements after they're visible (optimization!)

**What Intersection Observer does:**

javascript

*// Your code efficiently watches when elements enter viewport*

*// and adds 'visible' class to trigger CSS animations*

*// This is paired with your CSS:*

.section {

opacity: 0;

transform: translateY(50px);

transition: all 0.8s ease;

}

.section.visible {

opacity: 1;

transform: translateY(0);

}

This creates the smooth fade-in effect as you scroll! 🎨

**📋 PRIORITY TASKS (To catch up to Week 6):**

**1. Write About Section Content (Week 5 carryover)**

Write 300-500 words about:

- Your journey to coding

- Why you joined #100Devs

- Your goals as a developer

- What makes your approach unique (Gonzo style!)

- Technical skills you're learning

**2. Create Diary Entries for Weeks 1-6 (Week 6 priority)**

For each week, write:

* What you learned that week
* Key concepts covered
* Challenges you faced
* Wins/breakthroughs
* GitHub repo links
* Code snippets or screenshots

**Example Week 1 entry:**

Week 1: HTML & Foundations

- Built complete HTML structure for portfolio

- Learned semantic HTML5 elements

- Set up GitHub repository

- Challenge: Understanding when to use <section> vs <div>

- Win: Created all 23 diary page files in one sitting!

[Link to Week 1 GitHub commits]

**3. Add Missing JavaScript Features (Week 6 tasks)**

**A. Smooth Scroll for Navigation Links**

javascript

*// Add to script.js*

const navLinks = document.querySelectorAll('.nav-link');

navLinks.forEach(link => {

link.addEventListener('click', (e) => {

e.preventDefault();

const targetId = link.getAttribute('href');

const targetSection = document.querySelector(targetId);

targetSection.scrollIntoView({

behavior: 'smooth'

});

});

});

**B. Hamburger Menu Toggle (Mobile)**

First, add HTML for hamburger button in your nav:

html

<button class="hamburger" aria-label="Toggle menu">

<span></span>

<span></span>

<span></span>

</button>

Then add CSS for hamburger (in style.css):

css

.hamburger {

display: none;

flex-direction: column;

gap: 5px;

background: none;

border: none;

cursor: pointer;

}

.hamburger span {

width: 25px;

height: 3px;

background: var(--neon-pink);

transition: all 0.3s ease;

}

@media (max-width: 768px) {

.hamburger {

display: flex;

}

.nav-menu {

position: fixed;

top: 80px;

left: -100%;

flex-direction: column;

background: rgba(10, 10, 15, 0.95);

width: 100%;

transition: left 0.3s ease;

}

.nav-menu.active {

left: 0;

}

}

Then add JavaScript:

javascript

*// Add to script.js*

const hamburger = document.querySelector('.hamburger');

const navMenu = document.querySelector('.nav-menu');

hamburger.addEventListener('click', () => {

navMenu.classList.toggle('active');

hamburger.classList.toggle('active');

});

**C. Scroll-to-Top Button**

Add HTML before closing </body>:

html

<button id="scroll-top" class="scroll-top-btn" aria-label="Scroll to top">↑</button>

Add CSS:

css

.scroll-top-btn {

position: fixed;

bottom: 2rem;

right: 2rem;

width: 50px;

height: 50px;

background: linear-gradient(45deg, var(--neon-pink), var(--electric-blue));

border: none;

border-radius: 50%;

color: white;

font-size: 1.5rem;

cursor: pointer;

opacity: 0;

visibility: hidden;

transition: all 0.3s ease;

z-index: 999;

}

.scroll-top-btn.visible {

opacity: 1;

visibility: visible;

}

.scroll-top-btn:hover {

transform: translateY(-5px);

}

Add JavaScript:

javascript

*// Add to script.js*

const scrollTopBtn = document.getElementById('scroll-top');

window.addEventListener('scroll', () => {

if (window.scrollY > 300) {

scrollTopBtn.classList.add('visible');

} else {

scrollTopBtn.classList.remove('visible');

}

});

scrollTopBtn.addEventListener('click', () => {

window.scrollTo({

top: 0,

behavior: 'smooth'

});

});

**🎯 YOUR IMMEDIATE ACTION PLAN:**

**This Week (Week 6 - November 10-16, 2025):**

1. ✍️ **Write About section content** (2-3 hours)
2. ✍️ **Draft diary entries for Weeks 1-6** (3-4 hours)
3. 💻 **Add smooth scroll navigation** (30 minutes)
4. 💻 **Add hamburger menu** (1 hour)
5. 💻 **Add scroll-to-top button** (30 minutes)
6. 📸 **Gather images** for About section and hero (1 hour)
7. 🔗 **Add GitHub links** to completed diary entries

**Week 7 (November 17-23):**

* Draft diary entries for Weeks 7-12
* Create JSON file with diary data
* Implement dynamic diary loading
* Add filter/search functionality

**🏆 STRENGTHS YOU SHOULD BE PROUD OF:**

1. **Modern JavaScript** - You're using ES6+ features (arrow functions, const/let)
2. **Performance optimization** - Unobserving elements shows you understand efficiency
3. **Accessibility thinking** - Using semantic HTML and proper ARIA labels
4. **Clean code** - Good comments, readable structure
5. **Advanced CSS** - Your animations and effects are professional-level
6. **Design sensibility** - The "Fear & Loathing" aesthetic is cohesive and unique

**✨ BOTTOM LINE:**

**You're solidly in Week 6**, with some Week 5 content writing to catch up on. Your **technical skills are actually AHEAD** of the timeline (you're using APIs that aren't taught until Week 9-10), but your **portfolio content** (diary entries, About section) needs to be filled in.

**Focus on content creation this week**, then you'll be perfectly positioned to move into Week 7's more advanced JavaScript features!

You're doing GREAT! 🚀🔮✨

**Fear & Loathing in the Console: The 23-Week #100Devs Archive**

**"Buy the ticket, take the ride."** The raw path to mastery.

**Campaign Start: Monday, October 6, 2025** **Campaign End: Monday, March 16, 2026**

**The journey of 23 weeks begins Monday, October 6, 2025. You've got this!** 🚀

**PHASE I: Frontend Structure & Branding (Weeks 1-4)**

**Week 1: HTML Foundations & Project Setup**

**Dates: October 6-12, 2025**

**Technical Learning:**

* HTML5 semantic elements (header, nav, main, section, article, footer)
* Document structure, meta tags, and accessibility basics
* File organization and naming conventions
* Git basics: init, add, commit, push

**Portfolio Actions:**

* [ ] Set up project repository on GitHub
* [ ] Create root index.html with proper DOCTYPE and meta tags
* [ ] Build basic HTML structure: header, navigation, hero section, main content areas, footer
* [ ] Create /diary/ folder and generate 23 empty HTML files (week-01.html through week-23.html)
* [ ] Create /capstone/ folder with empty files: index.html, deep-dive.html, design-ux.html
* [ ] Create /assets/ folder structure: /img/, /css/, /js/
* [ ] Initialize README.md with project description
* [ ] Make first Git commit and push to GitHub

**Deliverables:** Complete file structure, GitHub repo initialized, all HTML skeleton files created

**Week 2: CSS Fundamentals & Visual Design**

**Dates: October 13-19, 2025**

**Technical Learning:**

* CSS syntax, selectors, specificity, and the cascade
* Box model (margin, border, padding, content)
* Typography fundamentals (font families, sizes, line-height, letter-spacing)
* Color theory and color systems (hex, RGB, HSL)
* CSS variables for theming

**Portfolio Actions:**

* [ ] Create assets/css/main.css and link to all HTML pages
* [ ] Define color palette (primary, secondary, accent, neutrals) using CSS variables
* [ ] Select and import fonts (Google Fonts or similar)
* [ ] Style typography system: headings (h1-h6), paragraphs, links, lists
* [ ] Apply box model to create visual hierarchy
* [ ] Create /assets/img/ subfolders: /hero/, /about/, /diary/, /capstone/
* [ ] Source placeholder images for hero section
* [ ] Begin designing logo or personal brand mark

**Deliverables:** CSS file with variables, typography system, initial styling applied to index.html

**Week 3: Flexbox & Responsive Layouts**

**Dates: October 20-26, 2025**

**Technical Learning:**

* CSS Flexbox: flex containers, flex items, justify-content, align-items, flex-wrap
* Responsive design principles and mobile-first approach
* CSS Grid basics (for future use)
* Media queries and breakpoints

**Portfolio Actions:**

* [ ] Implement flexbox for navigation bar (horizontal menu)
* [ ] Create responsive hero section with centered content
* [ ] Build About section layout using flexbox (image + text side-by-side)
* [ ] Design card layout for diary log entries using flexbox
* [ ] Add media queries for tablet (768px) and mobile (480px) breakpoints
* [ ] Test responsiveness on multiple screen sizes using browser dev tools
* [ ] Gather all images needed for About section (professional photo, background images)
* [ ] Optimize images for web (resize, compress)

**Deliverables:** Fully responsive homepage layout, mobile-friendly navigation, optimized images added

**Week 4: Advanced CSS & Component Building**

**Dates: October 27 - November 2, 2025**

**Technical Learning:**

* CSS positioning (relative, absolute, fixed, sticky)
* Transitions and basic animations
* Pseudo-classes and pseudo-elements
* CSS best practices and organization (BEM methodology or similar)

**Portfolio Actions:**

* [ ] Create reusable CSS components: buttons, cards, form inputs
* [ ] Add hover effects and transitions to navigation and buttons
* [ ] Implement sticky navigation bar
* [ ] Style the footer with social links and contact information
* [ ] Create /spells/ and /graveyard/ folders with empty index.html files
* [ ] Design and implement page headers for all sections
* [ ] Finalize color scheme and ensure consistent styling across all pages
* [ ] Create style guide document listing all colors, fonts, and component styles
* [ ] Add favicon and meta tags for social sharing

**Deliverables:** Complete aesthetic system, reusable components, all structural pages styled

**PHASE II: Frontend Logic & Content Drafts (Weeks 5-8)**

**Week 5: JavaScript Fundamentals I**

**Dates: November 3-9, 2025**

**Technical Learning:**

* JavaScript basics: variables (let, const, var), data types, operators
* Control flow: if/else statements, switch, ternary operators
* Arrays and array methods (push, pop, map, filter, forEach)
* Objects and object notation

**Portfolio Actions:**

* [ ] Create assets/js/main.js and link to all HTML pages
* [ ] Draft About section content (300-500 words about your journey, goals, and background)
* [ ] Write professional bio focusing on transition to development
* [ ] List technical skills and interests
* [ ] Create JavaScript file structure: /assets/js/components/, /assets/js/utils/
* [ ] Implement simple interactive feature: scroll-to-top button
* [ ] Add console logs to track page interactions for debugging practice

**Deliverables:** About section content completed, JavaScript file structure, first interactive feature

**Week 6: JavaScript Fundamentals II & DOM Manipulation**

**Dates: November 10-16, 2025**

**Technical Learning:**

* Functions: declarations, expressions, arrow functions, parameters, return values
* Scope and closures
* DOM manipulation: getElementById, querySelector, createElement
* Event listeners and event handling
* Debugging with browser console

**Portfolio Actions:**

* [ ] Draft diary entries for Weeks 1-6 (what you learned, challenges, wins, GitHub links)
* [ ] Implement dynamic diary entry loading (generate cards from JS array)
* [ ] Create hamburger menu toggle for mobile navigation using JavaScript
* [ ] Add smooth scroll behavior to navigation links
* [ ] Implement form validation for contact form (if included)
* [ ] Create utility functions file (utils.js) for reusable code
* [ ] Add GitHub repository links to first 6 diary entries

**Deliverables:** First 6 diary entries drafted, dynamic content loading, mobile menu functionality

**Week 7: Loops, Iterations & Data Structures**

**Dates: November 17-23, 2025**

**Technical Learning:**

* For loops, while loops, do-while loops
* Array iteration methods: map, filter, reduce, find, some, every
* String methods and template literals
* JSON data format and manipulation

**Portfolio Actions:**

* [ ] Draft diary entries for Weeks 7-12
* [ ] Create JSON file with diary entry data structure
* [ ] Refactor diary entry generation to pull from JSON
* [ ] Implement search/filter functionality for diary entries by week or topic
* [ ] Add tags/categories to diary entries (HTML, CSS, JavaScript, etc.)
* [ ] Create filter buttons to sort diary by technology/topic
* [ ] Gather and add images for diary entries (screenshots of projects, code snippets)
* [ ] Optimize all new images for web

**Deliverables:** Weeks 1-12 diary entries completed, dynamic filtering system, JSON data structure

**Week 8: Functions Deep Dive & Code Organization**

**Dates: November 24-30, 2025** *(Thanksgiving Week)*

**Technical Learning:**

* Higher-order functions and callbacks
* Function composition
* ES6+ features: destructuring, spread/rest operators, default parameters
* Module patterns and code organization

**Portfolio Actions:**

* [ ] Refactor JavaScript code into modular components
* [ ] Create separate JS files for different features (navigation.js, diary.js, etc.)
* [ ] Implement dark/light mode toggle using JavaScript and localStorage
* [ ] Add loading states and error handling to interactive features
* [ ] Create reusable helper functions for common tasks
* [ ] Add animation library (optional: anime.js or GSAP) for enhanced interactions
* [ ] Review and refine all Weeks 1-12 diary content for clarity and completeness
* [ ] Ensure all code examples and GitHub links are working

**Deliverables:** Modular JavaScript architecture, dark mode feature, polished diary content (Weeks 1-12)

**PHASE III: API Integration & Frontend Polish (Weeks 9-12)**

**Week 9: Asynchronous JavaScript & Promises**

**Dates: December 1-7, 2025**

**Technical Learning:**

* Asynchronous JavaScript: callbacks, promises, async/await
* Error handling with try/catch
* Promise.all, Promise.race
* Understanding the event loop and call stack

**Portfolio Actions:**

* [ ] Draft diary entries for Weeks 13-16
* [ ] Implement email service integration (EmailJS or similar) for contact form
* [ ] Add loading spinner/animation during async operations
* [ ] Create error messages and success notifications
* [ ] Test all asynchronous features thoroughly
* [ ] Add form submission confirmation message
* [ ] Update contact form with better UX feedback

**Deliverables:** Weeks 13-16 diary entries drafted, working contact form with email integration

**Week 10: Fetch API & Working with External Data**

**Dates: December 8-14, 2025**

**Technical Learning:**

* Fetch API and HTTP methods (GET, POST, PUT, DELETE)
* Working with REST APIs
* Handling API responses and status codes
* API authentication basics (API keys)
* CORS and security considerations

**Portfolio Actions:**

* [ ] Integrate a third-party API showcase project (weather app, GitHub repos display, etc.)
* [ ] Display your GitHub contribution graph or stats using GitHub API
* [ ] Create a "Latest Projects" section that pulls from GitHub API
* [ ] Add error boundaries and fallback UI for API failures
* [ ] Implement rate limiting awareness and caching strategies
* [ ] Draft diary entries for Weeks 17-19
* [ ] Add API integration project to portfolio with documentation

**Deliverables:** API integration showcase, GitHub stats display, Weeks 17-19 diary entries

**Week 11: React Introduction & Component Architecture**

**Dates: December 15-21, 2025**

**Technical Learning:**

* React fundamentals: JSX, components, props, state
* Create React App setup (or Vite)
* Component lifecycle and hooks (useState, useEffect)
* Thinking in React: component composition and data flow

**Portfolio Actions:**

* [ ] Set up React development environment
* [ ] Convert static portfolio to React components (optional migration path)
* [ ] OR create separate React mini-project for portfolio
* [ ] Build component library: Button, Card, Navigation, Hero components
* [ ] Implement React Router for multi-page navigation
* [ ] Create /spells/ content page: "Development Resources & Tools I've Mastered"
* [ ] List all tools, libraries, and resources learned with descriptions
* [ ] Add links and categories to spells page

**Deliverables:** React environment setup, component library, Spells page completed

**Week 12: React State Management & Frontend Finalization**

**Dates: December 22-28, 2025** *(Christmas Week!)*

**Technical Learning:**

* Advanced React hooks: useContext, useReducer, custom hooks
* State management patterns
* React performance optimization
* Building and deploying React applications

**Portfolio Actions:**

* [ ] Create /graveyard/ content page: "Failed Experiments & Lessons Learned"
* [ ] Document projects that didn't work and what you learned
* [ ] Draft diary entries for Weeks 20-23
* [ ] Complete ALL 23 diary entries with final content
* [ ] Ensure all GitHub links are present and working
* [ ] Add project screenshots and code snippets to diary entries
* [ ] Create comprehensive navigation between all pages
* [ ] Add breadcrumbs and "back to top" links throughout site
* [ ] Final styling pass: check consistency, spacing, alignment, colors
* [ ] Test all interactive features across browsers (Chrome, Firefox, Safari)
* [ ] Test mobile responsiveness on actual devices
* [ ] Deploy frontend to Netlify/Vercel/GitHub Pages
* [ ] **🎄 CHRISTMAS GOAL: Frontend is Live! 🎄**

**Deliverables:** Graveyard page completed, all 23 diary entries finished, frontend deployed and live

**PHASE IV: Backend & Full-Stack Foundation (Weeks 13-16)**

**Week 13: Node.js & Server-Side JavaScript**

**Dates: December 29, 2025 - January 4, 2026** *(New Year's Week)*

**Technical Learning:**

* Node.js fundamentals and runtime environment
* NPM and package management
* Node modules: fs, path, http
* Asynchronous patterns in Node.js
* Environment variables and configuration

**Portfolio Actions:**

* [ ] Install Node.js and verify installation
* [ ] Initialize Node.js project with npm init
* [ ] Create basic HTTP server using Node's http module
* [ ] Set up project structure: /server/, /server/routes/, /server/models/
* [ ] Install and configure nodemon for development
* [ ] Create .env file for environment variables (add to .gitignore)
* [ ] Set up ESLint and Prettier for code quality
* [ ] Update Week 13 diary entry with backend learning experience

**Deliverables:** Node.js environment configured, basic server running, backend file structure established

**Week 14: Express.js & RESTful API Design**

**Dates: January 5-11, 2026**

**Technical Learning:**

* Express.js framework fundamentals
* Routing and middleware
* RESTful API principles and design
* Request/response handling
* Error handling middleware
* CORS configuration

**Portfolio Actions:**

* [ ] Install Express.js: npm install express
* [ ] Convert basic server to Express application
* [ ] Create API routes: /api/diary, /api/contact
* [ ] Implement middleware: body-parser, cors, morgan (logging)
* [ ] Create route handlers for GET, POST, PUT, DELETE
* [ ] Test API endpoints using Postman or Thunder Client
* [ ] Set up proper error handling and status codes
* [ ] Create API documentation file
* [ ] Update Week 14 diary entry

**Deliverables:** Express server running, RESTful API endpoints created, API documentation

**Week 15: MongoDB & Database Integration**

**Dates: January 12-18, 2026**

**Technical Learning:**

* NoSQL databases and MongoDB concepts
* MongoDB Atlas setup
* Mongoose ODM (Object Data Modeling)
* Schema design and data modeling
* CRUD operations with Mongoose
* Database relationships and population

**Portfolio Actions:**

* [ ] Create MongoDB Atlas account (free tier)
* [ ] Set up MongoDB cluster and database
* [ ] Install Mongoose: npm install mongoose
* [ ] Create database connection file
* [ ] Define Mongoose schema for contact form submissions
* [ ] Define Mongoose schema for diary comments (optional feature)
* [ ] Implement database CRUD operations
* [ ] Test database operations with sample data
* [ ] Create database seeding script for initial data
* [ ] Convert contact form to save submissions in database
* [ ] Update Week 15 diary entry

**Deliverables:** MongoDB connected, Mongoose schemas created, database-backed contact form

**Week 16: Authentication & Security**

**Dates: January 19-25, 2026**

**Technical Learning:**

* Authentication vs Authorization
* Password hashing with bcrypt
* JSON Web Tokens (JWT)
* Session management
* Security best practices (helmet, rate limiting)
* Input validation and sanitization

**Portfolio Actions:**

* [ ] Install security packages: bcrypt, jsonwebtoken, helmet, express-rate-limit
* [ ] Implement password hashing for user accounts (if building admin panel)
* [ ] Create JWT authentication middleware
* [ ] Add protected routes for admin features
* [ ] Implement rate limiting on API endpoints
* [ ] Add input validation using express-validator
* [ ] Implement comment system for diary entries (database-backed)
* [ ] Create admin panel for managing comments (optional)
* [ ] Security audit: check for vulnerabilities, secure environment variables
* [ ] Begin Capstone project planning: write project brief and feature list
* [ ] Update Week 16 diary entry

**Deliverables:** Secure authentication system, comment system implemented, Capstone project brief

**PHASE V: Capstone Build & Documentation (Weeks 17-20)**

**Week 17: Capstone Architecture & Planning**

**Dates: January 26 - February 1, 2026**

**Technical Learning:**

* Full-stack application architecture
* Database schema design for complex applications
* API endpoint planning
* State management strategies
* Project planning and user stories

**Portfolio Actions:**

* [ ] Define Capstone project scope and MVP features
* [ ] Create user stories: "As a [user type], I want [goal] so that [reason]"
* [ ] Design database schema with collections/models and relationships
* [ ] Create ERD (Entity Relationship Diagram) for database
* [ ] Plan API endpoints with HTTP methods and expected responses
* [ ] Create wireframes for all Capstone pages/views (use Figma, Excalidraw, or paper)
* [ ] Design user flow diagrams showing navigation paths
* [ ] Set up Capstone GitHub repository
* [ ] Initialize MERN stack project structure
* [ ] Write first draft of capstone/deep-dive.html:
  + [ ] Project overview and purpose
  + [ ] Architecture diagram
  + [ ] Technology stack explanation
  + [ ] Database schema documentation
  + [ ] API endpoint documentation
* [ ] Update Week 17 diary entry with Capstone progress

**Deliverables:** Capstone fully planned, deep-dive documentation (draft 1), wireframes completed

**Week 18: Capstone Backend Development**

**Dates: February 2-8, 2026**

**Technical Learning:**

* Advanced Express patterns
* Complex database queries and aggregation
* File uploads and storage (Multer, Cloudinary)
* Background jobs and scheduling (optional)
* WebSocket basics (optional for real-time features)

**Portfolio Actions:**

* [ ] Set up Capstone MongoDB database and collections
* [ ] Create all Mongoose models with validation
* [ ] Implement all API endpoints for core features
* [ ] Add authentication and authorization to Capstone
* [ ] Implement file upload functionality if needed
* [ ] Create database seed file with sample data
* [ ] Write API tests (using Jest or Mocha)
* [ ] Set up error logging and monitoring
* [ ] Test all backend endpoints thoroughly with Postman
* [ ] Update Week 18 diary entry with technical challenges and solutions

**Deliverables:** Capstone backend complete and tested, all API endpoints functional

**Week 19: Capstone Frontend Development**

**Dates: February 9-15, 2026**

**Technical Learning:**

* React advanced patterns (custom hooks, context API)
* State management with Redux or Zustand (if needed)
* React form handling (Formik or React Hook Form)
* Client-side routing with React Router
* Optimistic UI updates

**Portfolio Actions:**

* [ ] Set up Capstone React frontend with Create React App or Vite
* [ ] Create component architecture and folder structure
* [ ] Build all UI components matching wireframes
* [ ] Implement React Router for navigation
* [ ] Connect frontend to backend API
* [ ] Implement state management solution
* [ ] Add form validation and error handling
* [ ] Create loading states and skeleton screens
* [ ] Implement responsive design for all breakpoints
* [ ] Add animations and transitions for polish
* [ ] Write first draft of capstone/design-ux.html:
  + [ ] User personas
  + [ ] User stories
  + [ ] Wireframes and mockups
  + [ ] User flow diagrams
  + [ ] Design decisions and rationale
  + [ ] Accessibility considerations
* [ ] Update Week 19 diary entry

**Deliverables:** Capstone frontend complete, design-ux documentation (draft 1), full-stack integration working

**Week 20: Capstone Features & Polish**

**Dates: February 16-22, 2026**

**Technical Learning:**

* Performance optimization (code splitting, lazy loading)
* SEO basics for React apps
* Accessibility testing and WCAG compliance
* Cross-browser testing
* Debugging complex issues

**Portfolio Actions:**

* [ ] Implement all remaining Capstone features
* [ ] Add advanced features: search, filtering, sorting, pagination
* [ ] Optimize performance: image optimization, code splitting, caching
* [ ] Implement SEO: meta tags, Open Graph tags, sitemap
* [ ] Accessibility audit: keyboard navigation, ARIA labels, screen reader testing
* [ ] Add error boundaries and fallback UI
* [ ] Create 404 page and error pages
* [ ] Implement analytics (Google Analytics or similar)
* [ ] Final design polish: spacing, typography, colors, animations
* [ ] User testing: have 2-3 people test the app and provide feedback
* [ ] Fix bugs and refine based on feedback
* [ ] Update both Capstone documentation pages with final content
* [ ] Update Week 20 diary entry with final feature list

**Deliverables:** Feature-complete Capstone application, polished UI/UX, documentation finalized

**PHASE VI: Deployment & Professional Launch (Weeks 21-23)**

**Week 21: Deployment Preparation & Testing**

**Dates: February 23 - March 1, 2026**

**Technical Learning:**

* Environment variables and configuration for production
* Build processes and optimization
* Deployment platforms (Heroku, Render, Railway, DigitalOcean)
* CI/CD concepts
* Monitoring and logging in production

**Portfolio Actions:**

* [ ] Create production build of React frontend
* [ ] Optimize bundle size: analyze with webpack-bundle-analyzer
* [ ] Set up environment variables for production
* [ ] Configure CORS for production domain
* [ ] Set up MongoDB Atlas production database (separate from development)
* [ ] Create production configuration files
* [ ] Write deployment documentation with step-by-step instructions
* [ ] Choose deployment platforms:
  + [ ] Backend: Render/Railway/Heroku
  + [ ] Frontend: Vercel/Netlify
  + [ ] Database: MongoDB Atlas
* [ ] Test build locally to ensure everything works
* [ ] Perform security audit: check for exposed API keys, vulnerable dependencies
* [ ] Update Week 21 diary entry

**Deliverables:** Production-ready code, deployment documentation, security audit complete

**Week 22: Live Deployment & Integration**

**Dates: March 2-8, 2026**

**Technical Learning:**

* DNS and custom domains
* SSL/TLS certificates
* Database migrations
* Production debugging
* Performance monitoring

**Portfolio Actions:**

* [ ] Deploy backend to hosting platform (Render/Railway/Heroku)
* [ ] Deploy frontend to hosting platform (Vercel/Netlify)
* [ ] Configure environment variables on hosting platforms
* [ ] Connect frontend to deployed backend API
* [ ] Test all features in production environment
* [ ] Set up custom domain for portfolio (optional but recommended)
* [ ] Configure SSL certificate (usually automatic on Vercel/Netlify)
* [ ] Deploy Capstone application to live server
* [ ] Test Capstone thoroughly in production
* [ ] Set up error monitoring (Sentry or similar)
* [ ] Set up uptime monitoring (UptimeRobot or similar)
* [ ] Create backup strategy for database
* [ ] **🚀 Capstone is LIVE! 🚀**
* [ ] Update Week 22 diary entry with deployment experience

**Deliverables:** Portfolio and Capstone both live and accessible online

**Week 23: Professional Polish & Launch**

**Dates: March 9-16, 2026** *(FINAL WEEK!)*

**Technical Learning:**

* Portfolio best practices
* Technical writing for developers
* Creating effective README files
* Open source contribution guidelines

**Portfolio Actions:**

* [ ] Update About section with final professional bio
* [ ] Create comprehensive resume (PDF) and add download link
* [ ] Update LinkedIn profile with new skills and projects
* [ ] Add LinkedIn link to portfolio header/footer
* [ ] Create professional headshot/photo if not already done
* [ ] Write compelling project descriptions for all work
* [ ] Add live demo links and GitHub repos for all projects
* [ ] Create detailed README files for all GitHub repositories:
  + [ ] Project description
  + [ ] Technologies used
  + [ ] Installation instructions
  + [ ] Usage examples
  + [ ] Screenshots
  + [ ] Future enhancements
* [ ] Add Open Graph meta tags for social media sharing
* [ ] Create social media preview images for portfolio and Capstone
* [ ] Record demo video of Capstone walkthrough (2-3 minutes)
* [ ] Write blog post about your 23-week journey (publish on Medium/Dev.to)
* [ ] Final review of all 23 diary entries: ensure consistency and professionalism
* [ ] Add testimonials or recommendations if available
* [ ] Create elevator pitch for your portfolio (30 seconds)
* [ ] Update Week 23 diary entry: reflect on entire journey
* [ ] Share portfolio on Twitter, LinkedIn, Reddit (r/webdev)
* [ ] Apply to 5 developer positions

**Final Deliverables:**

* ✅ Complete, deployed portfolio website
* ✅ Live, fully-functional Capstone application
* ✅ 23 detailed diary entries documenting your journey
* ✅ Professional resume and LinkedIn profile
* ✅ GitHub profile with well-documented repositories
* ✅ Job-ready developer with demonstrable skills

**Important Dates to Remember**

* 📅 **October 6, 2025** - Campaign begins (Week 1)
* 🦃 **November 27, 2025** - Thanksgiving (Week 8)
* 🎄 **December 25, 2025** - Christmas - Frontend should be LIVE! (Week 12)
* 🎊 **January 1, 2026** - New Year (Week 13)
* 🚀 **March 16, 2026** - Campaign complete! Job-ready!

**Weekly Habits (Every Week)**

**Daily:**

* [ ] Code for at least 1-2 hours
* [ ] Commit code to GitHub with descriptive messages
* [ ] Review and refactor previous day's code

**Weekly:**

* [ ] Complete all learning objectives for the week
* [ ] Update diary entry with learnings, challenges, and wins
* [ ] Push all code to GitHub
* [ ] Test new features across browsers and devices
* [ ] Participate in developer community (Discord, Reddit, Twitter)
* [ ] Review next week's objectives and prepare

**Resources to Use Throughout:**

* MDN Web Docs (HTML/CSS/JS reference)
* JavaScript.info (in-depth JS tutorials)
* freeCodeCamp (practice challenges)
* The Odin Project (comprehensive curriculum)
* Frontend Mentor (practice projects)
* Stack Overflow (problem-solving)
* YouTube (visual learning)

**Success Metrics**

By March 16, 2026, you will have:

* ✅ Mastery of HTML, CSS, JavaScript, React
* ✅ Full-stack development skills with MERN stack
* ✅ Live portfolio website with 23 weeks of documented growth
* ✅ Complete, deployed Capstone application
* ✅ Professional online presence (GitHub, LinkedIn)
* ✅ Job-ready skills with portfolio to prove it