

Here's My 100-Day Learning Plan

Phase 1: Web Development Fundamentals (Days 1–20)

HTML & CSS Basics (Days 1–10):

- Day 1: Learn HTML structure, tags (<html>, <head>, <body>), and basic tags (<p>, <a>, , <div>,).
- Day 2: Work with headings, lists, tables, and form elements (<input>, <button>, <select>).
- Day 3: Introduction to CSS; styling basics (colors, fonts, margins, padding).
- Day 4: Learn about the CSS box model, display properties, and positioning (relative, absolute, fixed).
- Day 5: Introduction to flexbox for responsive layouts; create a simple navbar and card layout.
- Day 6: Practice with CSS Grid layouts for responsive designs.
- Day 7: Learn CSS selectors and pseudo-classes (:hover, :nth-child, etc.).
- Day 8: Responsive design basics with media queries.
- Days 9–10: **Mini Project:** Build a personal portfolio page using HTML and CSS.

JavaScript Basics (Days 11–20):

- Day 11: Learn JavaScript basics (variables, data types, and operators).
- Day 12: Explore conditional statements (if, else, switch) and loops (for, while).
- Day 13: Functions, scope, and closures.
- Day 14: Work with JavaScript arrays, objects, and basic array methods (map, filter, reduce).
- Day 15: Understand DOM manipulation basics (selectors, modifying elements).
- Day 16: Event handling basics in JavaScript (click events, input events).
- Day 17: Practice with basic animations in JavaScript.
- Days 18–20: **Mini Project:** Build an interactive To-Do List with HTML, CSS, and JavaScript.

Phase 2: Frontend Development with React (Days 21–40)

React Basics (Days 21–30):

- Day 21: Set up a React environment, understand JSX, and create your first component.
- Day 22: Learn about props and passing data to components.
- Day 23: State management with useState.
- Day 24: Component lifecycle with useEffect.
- Day 25: Build small components like counters and buttons to reinforce props and state.
- Day 26: Explore React Router for navigation.

- Day 27: Set up multiple routes and nested routes in React Router.
- Days 28–30: **Mini Project:** Create a simple blog with multiple pages using React Router.

Advanced React Features (Days 31–40):

- Day 31: Learn Context API for global state management.
 - Day 32: Refactor components using the Context API.
 - Day 33: Introduction to custom hooks in React.
 - Days 34–35: **Mini Project:** Build features using custom hooks and context.
 - Days 36–40: **Project:** Build a React app (e.g., blog or e-commerce frontend) using state, context, and routing.
-

Phase 3: Backend Development with Node.js & Express (Days 41–60)

Node.js Basics (Days 41–45):

- Day 41: Introduction to Node.js; create a basic server.
- Day 42: Understand modules, NPM, and packages.
- Day 43: Work with file system and path modules.
- Day 44: Learn asynchronous programming and callbacks.
- Day 45: Introduction to promises and async/await.

Express.js Basics (Days 46–50):

- Day 46: Set up an Express server, create routes.
- Day 47: Use middleware for logging and error handling.
- Day 48: Handle requests and responses.
- Day 49: Practice error handling and debugging.
- Day 50: **Mini Project:** Create basic CRUD routes.

REST API Development (Days 51–60):

- Day 51: Introduction to REST API design principles.
- Day 52: Create routes for a simple API (GET, POST, PUT, DELETE).
- Day 53: Practice CRUD operations for a resource (e.g., books or users).
- Day 54: Request validation with middleware.
- Day 55: Implement authentication (e.g., JWT).
- Day 56: Secure endpoints with authentication middleware.
- Days 57–60: **Project:** Build a “To-Do” REST API with authentication.

Phase 4: Database with MongoDB & Mongoose (Days 61–80)

MongoDB Basics (Days 61–70):

- Day 61: Set up MongoDB; create your first database and collection.
- Day 62: Perform CRUD operations in MongoDB.
- Day 63: Learn MongoDB queries and filters.
- Day 64: Understand indexing.
- Day 65: Set up Mongoose, create schemas and models.
- Day 66: Add Mongoose validation.
- Day 67: Understand relationships (embedded vs. referenced documents).
- Day 68: Customize queries (sorting, filtering, pagination).
- Days 69–70: **Mini Project:** Build a blog API with CRUD operations.

Advanced MongoDB (Days 71–80):

- Day 71: Work with aggregations for complex queries.
- Day 72: Practice aggregations.
- Day 73: Handle transactions in MongoDB.
- Day 74: Optimize with indexing.
- Day 75: Use MongoDB Atlas for cloud-hosted databases.
- Days 76–80: **Project:** Enhance the blog API with transactions and advanced queries.

Phase 5: Full-Stack Development (Days 81–100)

Connecting Frontend & Backend (Days 81–85):

- Day 81: Use Fetch or Axios to send HTTP requests from React to Express.
- Day 82: Set up a proxy for development.
- Day 83: Fetch data into React components.
- Day 84: Work with POST requests from the frontend.
- Day 85: Handle JWT-based authentication on the frontend.

Full-Stack Project (Days 86–100):

- Days 86–90: Design database schema, backend API, and frontend components for a full-stack app.
- Days 91–95: Implement user authentication with role-based access control.

- Day 96: Manage state across the app with Redux (or alternatives).
- Day 97: Integrate advanced UI elements.
- Day 98: Add pagination, sorting, and filtering on the frontend.
- Day 99: Test the app end-to-end and debug.
- Day 100: Finalize and deploy the app (e.g., on Heroku and MongoDB Atlas).