

Days	Category	Focus Task (Approx. 2 Hours)	Example/Goal
Days 1-20	Front-End Fundamentals (HTML, CSS, JS)		
1-2	HTML	Basics & Semantic Structure	Build a simple, unstyled "About Me" page.
3-4	CSS	Basic Selectors & Styling	Add background color, fonts, and borders to the "About Me" page.
5-6	CSS	Box Model & Positioning	Understand margins, padding, and use absolute/relative positioning for one element.
7-8	CSS	Flexbox Fundamentals	Use Flexbox to align content (e.g., center a section or create a navigation bar).
9-10	JavaScript	Variables, Data Types, & Operators	Complete a few code-challenges (e.g., variables, basic calculations).
11-12	JavaScript	Control Flow (If/Else, Loops)	Write a function to check if a number is even/odd and loop through an array.
13-14	JavaScript	Functions & Scope	Write 3 reusable functions (e.g., calculate area, format text).
15-16	JavaScript	DOM Manipulation (Read/Modify)	Use JS to change the text content and style of elements on your page.
17-18	JavaScript	Event Listeners & User Interaction	Add a button that changes an image or displays a secret message when clicked.
19-20	Review	Mini-Project: Interactive Page	Build a simple color-flipper or a basic counter app using HTML, CSS, and vanilla JS.
Days 21-40	Front-End Framework (React)		
21-23	React	Intro, JSX, & Components	Set up a new React project and create 3 simple functional components.
24-26	React	Props & Component Reusability	Create a generic <Card> component and pass different data to it via props.
27-29	React	State (useState)	Build a simple form where the input value is stored and displayed in real-time.
30-32	React	Lifecycle (useEffect)	Use useEffect to fetch static local data or run code when a component loads.
33-35	React	Conditional Rendering & Lists	Render a list of items (.map()) and conditionally display an "empty list" message.
36-37	React Router	Basic Routing	Set up react-router-dom with 2-3 pages (Home, Details).
38-40	Review	Mini-Project: API Display	Fetch and display data from a public, simple API (e.g., weather or jokes) in your React app.
Days 41-50	Back-End Fundamentals (Node, Express, MongoDB)		
41-42	Node.js	Basics & NPM	Install Node, set up a new project, and run a simple server-side JS file.
43-44	Express.js	Server Setup & Routing	Create a basic Express server and a single /api/test endpoint.
45-46	Express.js	Handling Requests (GET/POST)	Add routes to return a list of fictional items (GET) and accept a new item (POST - in-memory).
47-48	MongoDB	Database Connection & Mongoose	Set up a MongoDB Atlas account and connect your Express server using Mongoose.
49-50	Mongoose	Creating a Schema & Model	Define a Mongoose Schema for a simple resource (e.g., "Note") and create your first database entry.
Days 51-60	Integration & Final Project		
51-52	CRUD	Read Operations (Find)	Implement the code to fetch all "Notes" from the MongoDB and return them via an API GET route.
53-54	CRUD	Create Operations (Save)	Implement the code to receive data from a POST request and save a new "Note" to the database.
55-56	Integration	Connecting Front-End to Back-End (Read)	In your React app, use the fetch API or a library like Axios to call your back-end and display the list of "Notes."
57-58	Integration	Connecting Front-End (Create)	Build a form in React to send data and create a new "Note" via your API.
59	Deployment	Front-End Deployment	Deploy your current React front-end (e.g., using Netlify or Vercel).
60	Final Review	Clean Up & Next Steps	Refactor code, ensure basic responsiveness, and plan the next 60 days of learning (focusing on Update/Delete/Authentication).