Full Stack Web Development Roadmap

Guidelines for this roadmap:

Active Learning: Approach each topic with curiosity and engagement. Don't just watch tutorials passively; actively code along and experiment with the concepts you're learning.

Practice Regularly: Consistent practice is key to mastering web development. Set aside dedicated time each day to work on coding exercises, projects, or reviewing concepts.

Hands-On Projects: Apply what you learn by working on real-world projects. Start with small projects and gradually increase complexity as you gain confidence.

Seek Understanding: Don't just memorize syntax or code snippets. Strive to understand the underlying concepts and principles behind each topic. This deeper understanding will serve you well in problem-solving and troubleshooting.

Documentation and Resources: Make use of official documentation, tutorials, articles, and forums to supplement your learning. Don't hesitate to explore different resources to find what works best for you.
Collaboration and Community: Engage with the developer community through forums, social media, and coding meetups. Collaboration and sharing experiences with others can provide valuable insights and support.

Reflect and Iterate: Regularly reflect on your progress, strengths, and areas for improvement. Iterate on your learning approach based on feedback and experiences.

Stay Updated: Web development is a rapidly evolving field. Stay updated with the latest trends, technologies, and best practices by following industry blogs, attending webinars, and participating in online courses.

Debugging Skills: Debugging is an essential skill for developers. Learn to effectively use debugging tools and techniques to identify and resolve issues in your code.
Patience and Persistence: Learning web development can be challenging at times. Be patient with yourself and persist through difficulties. Remember that every obstacle you overcome brings you closer to your goals.

Week 1-2 : Let's explore the world of **web**

The focus is on understanding the core concepts of web development, starting with the fundamentals of internet protocols and communication. You'll dive into HTML for structuring web pages and CSS for styling them, alongside learning version control using Git and basic debugging techniques with Chrome Dev Tools.

Day 1	Day 2	Day 3	Day 4	Day 5 (Git day)
How does internet work? (1h) Things you should know about web dev (15m)	Basics of HTML (2h)	Basics of CS (4h)	Intro to chrome dev tools (4h)	What is Git? (2h)
What is HTTP? (30m)	Basis of HTML (3h)	Basics of CS (4h)	Dev tools Tips and Tricks (30m)	Git and Github Course (4h)
What is web Hosting? (10m)	Basics of CS (3h)		Flexbox (2h)	Git cheat sheet (2h)
DNS (1h)			Grids (1h30m)	
How browsers work? (1h)				
Front-end Introduction (10m)				
Back-end Introduction (10m)				
APIs Explained (5m)				
Basics of HTML (4h)				

Home Assignment: Convert this Figma design to responsive HTML, CSS website.

Week 3-4: Scripting Javascript Fundamentals

Week 3-4 introduces JavaScript, where you'll build a strong foundation covering data types, variables, and control structures, enabling you to manipulate the Document Object Model (DOM) and handle asynchronous operations effectively.

Day 1	Day 2	Day 3	Day 4	Day 5 (Git day)
Javascript crash course for beginners (4h)	Loops (Video) Loops and Iteration (1h)	this (video) This keyword (30m)	DOM Manipulation(7h30m)	Git branches (2h)
Variables (Video) Variable Scopes Variables (1h)	Conditions (Video) Conditions (30m)	Async Javascript (4h)	Concepts of javascript you should know (15m)	Merge and Rebase (2h)
Objects intro (1h) OOP in Javascript (2h)	Arrays (Video) Arrays (Theoretical) (2h)	Modules Import and Export (1h)	JS Pro tips (15m)	Git tips & tricks Git with Fireship (1h)
Primitive types (20m)	Functions Intro Functions in detail (1h30m)	JSON Intro (1h)		Resolving merge conflicts (3h)
	Arrow Functions (30m) Function borrowing (30m)	Javascript Array challenges (1h30m)		
	Javascript Objects Challenge (2h)			

Home Assignment: Build **Mastermind** game using Javascript, HTML and CSS. Share Github repo link.

Week 5-6 : Time to **React**

In Week 5-6, the focus shifts to React, a popular JavaScript library for building dynamic user interfaces. You'll learn about React components and their lifecycles, state management using hooks, and how to handle form submissions and API calls efficiently. Additionally, you'll explore client-side routing and create a complete React application integrating frontend components with backend APIs.

Day 1	Day 2	Day 3	Day 4	Day 5
Vite (30m)	React Tutorial (8h)	Component Lifecycle (1h)	Formik (4h)	API calls in react (2h)
NPM (1h)		Lists and Keys (30m)	How to style JSX? (30m)	Axios (2h)
Javascript for React (1h30m)		React hooks (4h) Only learn these hooks: useState, useEffect, useMemo, useCallback, useContext, useReducer, useRef	Zustand (3h)	SWR for data fetching SWR docs (4h)
Intro to react (5h) Please create a complete react application on your system.		Custom Hooks (1h30m)		All React concepts explained (15m)
		React Router (1h)		

Home Assignment: Please complete this project.

Week 7-8 : Navigating the **Backend** Realm

Week 7-8 delves into backend development with Node.js and Express.js, where you'll understand RESTful API design principles, perform CRUD operations with databases like MongoDB, and implement authentication and authorization mechanisms. This week also covers common pitfalls and best practices in backend development, including error handling and performance optimization, providing a well-rounded understanding of full-stack web development.

Day 1	Day 2	Day 3	Day 4	Day 5
REST API (30m)	Nodejs Complete Course (8h) Watch first 35 videos	Cron Jobs (30m)	Web Storages (30m)	Final Assessment (4 days)
Intro to Node Intro to Node - 2 (4h)		Express JS (4h)	JWT Tokens Session vs JWT tokens (1h)	
Relational Databases vs Non-Relational Databases (30m)		Express JS Mistakes to Avoid (30m)	Session Authentication in react (2h)	
Mongodb Crash Course (3h)		React Mistakes to avoid React mistakes to avoid - 2 React mistakes to avoid - 3 (2h)	Why not to use Local storage (30m) Redis Crash Course (4h)	