



## HTML, CSS, JavaScript, and ReactJS Curriculum - Beginner to Intermediate

### ❖ 1. Beginner Curriculum

- 1) Introduction to Web Development
- 2) Overview of Web Development
- 3) How Websites Work: Frontend vs. Backend
- 4) Introduction to HTML, CSS, and JavaScript

### ❖ 2. HTML Basics

- 1) HTML Structure and Document Layout
- 2) Tags, Elements, and Attributes
- 3) Basic HTML Tags: Headings, Paragraphs, Links, Images
- 4) Lists and Tables
- 5) Forms and Input Elements
- 6) Best Practices in HTML

### ❖ 3. CSS Fundamentals

- 1) Introduction to CSS and How It Works with HTML
- 2) CSS Syntax and Selectors
- 3) Styling Text: Font, Color, Alignment, etc.
- 4) Layouts: Margins, Padding, Borders
- 5) Box Model Concept
- 6) Working with Backgrounds and Images
- 7) CSS Units (px, %, em, rem) and Positioning
- 8) CSS Flexbox and Grid Basics





## **4. JavaScript Basics**

- 1) What is JavaScript and How It Works in the Browser
- 2) Variables, Data Types, and Operators
- 3) Control Structures: Conditionals and Loops
- 4) Functions and Scope
- 5) Basic DOM Manipulation
- 6) Event Handling (Click, Hover, etc.)
- 7) Arrays, Objects, and Basic Array Methods
- 8) Debugging in JavaScript

## **5. Introduction to ReactJS**

- 1) Understanding Single Page Applications (SPA)
- 2) What is React and Why Use It?
- 3) Setting Up a React Environment (Using npx create-react-app)
- 4) React Component Basics: Functional Components
- 5) JSX: JavaScript Syntax Extension for React
- 6) Props and State
- 7) Handling Events in React
- 8) Basic Styling in React (Inline CSS and CSS Modules)

## **6. Building a Simple Project**

- 1) Integrating HTML, CSS, and JavaScript Skills
- 2) Creating a Static Web Page (Portfolio or Blog Page)
- 3) Adding Basic Interactivity with JavaScript

## **7. Building a React Project**

- 1) Project Overview: Building a Simple To-Do App or Counter
- 2) Structuring Components
- 3) Managing State and Props
- 4) Adding Dynamic Styling and Animations
- 5) Deploying the Project on GitHub Pages or a Similar Platform

## **8. Final Project and Review**

- ❖ Project: Personal Portfolio with React
- Review of Key Concepts





## Intermediate Curriculum

---

### **1. Advanced HTML and CSS**

- 1) HTML5 Semantic Elements
- 2) Accessibility Best Practices
- 3) Responsive Web Design with CSS Media Queries
- 4) Advanced CSS Selectors and Pseudo-Classes
- 5) CSS Animations and Transitions
- 6) CSS Preprocessors (e.g., SASS) Overview
- 7) Responsive Layouts with Flexbox and Grid in Depth
- 8) Working with CSS Variables

### **2. JavaScript Deep Dive**

- 1) ES6+ Syntax and Modern JavaScript Features
- 2) let, const, arrow functions, template literals
- 3) Destructuring, spread/rest operators
- 4) Working with Promises and Async/Await
- 5) Understanding Closures, Scope, and the "this" Keyword
- 6) JavaScript Modules and Imports/Exports
- 7) Advanced Array and Object Methods (map, filter, reduce)
- 8) Error Handling and Debugging Techniques
- 9) Introduction to JavaScript Fetch API and AJAX

### **3. Intermediate DOM Manipulation and Events**

- 1) Advanced DOM Selection and Manipulation
- 2) Dynamic Content Creation with JavaScript
- 3) Event Delegation and Bubbling/Capturing
- 4) Form Validation and Regular Expressions
- 5) Using Local Storage and Session Storage
- 6) Modularizing Code with JavaScript Classes and Prototypes

### **4. Advanced ReactJS Concepts**

- 1) React Functional vs. Class Components
- 2) Introduction to React Hooks
- 3) useState, useEffect, and Custom Hooks
- 4) Component Lifecycle and Side Effects
- 5) State Management with Context API
- 6) Working with Forms in React and Controlled Components
- 7) Prop Drilling and Composition Techniques
- 8) Prop Types and Type Checking
- 9) Basic Routing with React Router





## **5. API Integration in React**

- 1) Introduction to RESTful APIs and JSON
- 2) Using the Fetch API and Axios in React
- 3) Fetching Data from Public APIs
- 4) Error Handling with API Calls
- 5) Displaying Loading States and Error Messages
- 6) Building a CRUD App with API Integration

## **6. Advanced Project: Building a Responsive React App**

- 1) Project Planning and Component Structuring
- 2) State and Data Flow Management
- 3) Implementing Responsive Design with CSS and Flexbox/Grid
- 4) Adding Interactivity and Animations with CSS and JavaScript
- 5) API Integration with CRUD Operations (e.g., a To-Do App or Blog)
- 6) Using React Router for Multi-Page Functionality

## **7. Testing and Optimization in React**

- 1) Basics of Unit Testing in React
- 2) Testing Tools: Jest and React Testing Library
- 3) Writing Tests for Components and Hooks
- 4) Performance Optimization Techniques in React
- 5) Memoization with React.memo and useMemo
- 6) Code Splitting and Lazy Loading
- 7) Minimizing Re-Renders and Improving Load Time
- 8) Analyzing Performance with React DevTools

## **8. Deployment and Version Control**

- 1) Introduction to Git and GitHub (Branching, Merging, Pull Requests)
- 2) Preparing a React App for Production
- 3) Deployment on Platforms like Netlify, Vercel, or GitHub Pages
- 4) Setting up Continuous Integration/Continuous Deployment (CI/CD) Basics

## **9. Final Project and Portfolio Preparation**

***Project: Create a Fully-Functional Web App (e.g., Weather App, Blog Platform, or e-Commerce Interface)***

