

## React Developer Course Outline

Course Duration: 2.5 Months

Timing: Flexible - 1 Hour Daily

### Skills Covered in This Course

1. HTML
2. CSS
3. JAVASCRIPT
4. BOOTSTRAP
5. REACT JS
6. REDUX
7. CSS MODULES
8. SASS(SCSS)
9. ZUSTAND
10. TYPESCRIPT BASICS
11. NEXT JS BASICS
12. GIT BASICS
13. VERCEL HOSTING BASICS

### TOOLS USED

- VISUAL STUDIO CODE
- GITHUB DESKTOP

# HTML OUTLINE

## **1. Introduction to HTML**

- What is HTML
  - What is HTML5
  - W3C and Web Standards
  - What is a Markup Language
  - Accessibility in HTML
- 

## **2. Basic Structure of an HTML Document**

- HTML File Basic Structure
  - Doctype Declaration
  - <html>, <head>, and <body>
  - Self-closing Tags
- 

## **3. Text and Formatting Tags**

- Heading Tags
  - Paragraph Tag
  - Horizontal Line and Line Break
  - Bold and Strong Tags – Difference
  - Italic and Emphasis Tags – Difference
  - Preformatted Text
  - Quote Tag
  - Blockquote Tag
  - Abbreviation Tag
  - HTML Entities
- 

## **4. Hyperlinks and Navigation**

- Anchor Tag
  - Different Target Attributes in Anchor
  - Navigating to Another Section in a Webpage (Using IDs and Anchors)
  - Iframe Tag
    - Attributes of Iframe
    - \_self and \_top behaviour in Iframe
-

## 5. Multimedia in HTML

- Showing Images
  - Video in Webpage
  - Audio in Webpage
  - Downloading a Document in Webpage
  - Marquee Tag
- 

## 6. Forms and Input Elements

### 6.1 Form Basics

- Form Tag
- Fieldset and Legend
- Label Tag

### 6.2 Common Input Types

- Textbox
- Password Textbox
- Number Textbox
- Email Textbox
- Telephone Textbox
- Search Textbox
- Date Picker
- Time Picker
- Date and Time Picker Together
- Month Picker
- Week Picker

### 6.3 Selection Controls

- Dropdown List
- Grouping in Dropdown List
- Data list
- Checkbox
- Radio Button

### 6.4 Other Form Elements

- Text area
- File Upload

### 6.5 Buttons

- Normal Button
  - Reset Button
  - Submit Button
-

## 7. HTML Layout and Structure

- Block-level Elements
  - Inline Elements
  - div Tag
  - span Tag
- 

## 8. Semantic HTML (HTML5 Structural Tags)

- header
  - footer
  - article
  - section
  - main
  - aside
  - nav
  - Importance of Semantic HTML
- 

## 9. Additional Topics

- Best Practices
  - SEO Related Meta Tags
  - Difference between HTML and HTML5
- 

# CSS OUTLINE

## 1. Introduction to CSS

- What is CSS
  - Need for CSS
  - CSS Syntax and Structure
  - Types of CSS:
    - Inline Styling
    - Embedded / Internal Styling
    - External Styling
  - Adding a Favicon
  - Setting Page Title
- 

## 2. Colors and Backgrounds

- Color Property
  - Background Color
  - Background Image
  - Background Size
  - Background Repeat
  - Background Attachment
    - scroll vs local
  - Background Position
-

### 3. Box Model

- Understanding the Box Model
- Width and Height
- Max-width and Max-height
- Padding
  - Padding Top
  - Padding Right
  - Padding Bottom
  - Padding Left
- Margin
  - Margin Top
  - Margin Right
  - Margin Bottom
  - Margin Left
- Border
  - Border Width
  - Border Style
  - Border Color
  - Border Radius
- Outline
- Box Sizing
- Box Shadow
- Overflow
  - Overflow-X
  - Overflow-Y

---

### 4. Text and Font Styling

- Font Family
  - Font Size
  - Font Weight
  - Font Style
  - Font Variant
  - Line Height
  - Text Align
  - Text Decoration
    - Text Decoration Line
    - Text Decoration Style
  - Letter Spacing
  - Word Spacing
  - White-space
  - Text Overflow
  - Text Shadow
-

## 5. Display and Visibility

- Display Property
    - Display Inline
    - Display Block
    - Display Inline-block
  - Visibility Property
  - Opacity
  - Z-index
- 

## 6. Positioning and Layering

- Position Property
    - Static
    - Relative
    - Absolute
    - Fixed
    - Sticky
  - Object-fit
  - Object-position
- 

## 7. Lists and Tables

- List Style Type
  - List Style Image
  - List Style Position
  - Table:
    - Colspan and Rowspan
    - Border Spacing
    - Empty Cells
    - Caption Side
- 

## 8. Layout Techniques

- Flexbox
    - Flex Container and Flex Items
    - Justify Content, Align Items, Flex Wrap, etc.
  - CSS Grid
    - Grid Container and Grid Template
    - Rows, Columns, and Areas
- 

## 9. Responsive Design

- Media Queries
  - Bootstrap Basics (Introduction and Common Classes)
-

## 10. Transitions and Transformations

### 10.1 CSS Transitions

- Transition Property
- Transition Duration
- Transition Delay
- Transition Timing Function

### 10.2 CSS Transformations

- Transform Property
  - Transform Rotate
  - Transform Scale
  - Transform Translate
  - Transform Skew
- 

## 11. CSS Animations

- Animation Basics
  - Keyframes
  - Animation Properties (Duration, Delay, Iteration, etc.)
- 

## 12. CSS Selectors

### 12.1 Basic Selectors

- Universal Selector
- Type Selector
- Class Selector
- ID Selector
- Attribute Selectors
  - Exact Match
  - Starts With
  - Ends With
  - Contains

### 12.2 Combinators

- Nested Child Selectors
- Direct Child Selector
- Adjacent Sibling Selector
- General Sibling Selector

### 12.3 Pseudo Classes

- Structural Pseudo Classes
  - :first-child
  - :last-child
  - :nth-child()
  - :nth-last-child()
  - :only-child
  - :first-of-type
  - :last-of-type
  - :only-of-type
  - :nth-of-type()
  - :nth-last-of-type()
- UI State Pseudo Classes
  - :hover
  - :active
  - :focus
  - :focus-visible
  - :focus-within
  - :enabled
  - :disabled
  - :checked
  - :required
  - :optional
  - :valid
  - :invalid
  - :read-only
  - :in-range
- Document & Language Pseudo Classes
  - :root
  - :lang()
  - :target
- Advanced Pseudo Classes
  - :not()
  - :is()
  - :where()
  - :has()
  - :empty

### 12.4 Pseudo Elements

- ::before
- ::after
- ::first-letter
- ::first-line
- ::selection
- ::marker
- ::backdrop

## 12.5 Direction & Autocomplete

- :dir()
  - :autocomplete
- 

## 13. Advanced CSS Features

- CSS Variables (:root custom properties)
  - Layering and Stacking Contexts
  - Responsive Units (% em rem vw vh)
  - Best Practices and Performance Tips
- 
- 

# JAVASCRIPT OUTLINE

## . Introduction to JavaScript

- What is JavaScript
  - Features and Uses of JavaScript
  - Where JavaScript Runs (Browser + Node.js)
  - What is Node.js
  - What is V8 Engine
  - Installing Node.js
  - Running the First JavaScript Program
    - Using Node Terminal
    - Using Browser
  - JavaScript File Extension
  - Role of JavaScript in Web Development (HTML + CSS + JS)
- 

## 2. Basics of JavaScript Syntax

- Variables and Constants
    - var, let, and const
    - Difference between var, let, and const
    - Hoisting in Variables
  - Identifier Naming Rules
  - Comments in JavaScript
  - Printing Output
  - Debugging in VS Code and Browser Console
-

### 3. Data Types and Operators

- **Primitive Data Types:**
    - String, Number, Boolean, Null, Undefined, Symbol, BigInt
  - **typeof Operator**
  - Difference between null and undefined
  - Type Coercion and Type Conversion
  - **String Concatenation and Template Literals**
  - **Arithmetic Operators**
  - **Assignment Operators**
  - **Increment and Decrement Operators**
  - **Comparison Operators**
  - **Logical Operators**
  - **Ternary Operator**
- 

### 4. Control Flow Statements

- if, else if, else
  - Nested If Conditions
  - switch Statement
  - break and continue Keywords
  - Conditional vs Control Statements Overview
- 

### 5. Loops in JavaScript

- for Loop
  - while Loop
  - do...while Loop
  - Nested Loops
  - Difference between while and do...while
  - Modern Loops:
    - for...of
    - for...in
- 

### 6. Functions

- What is a Function
  - Function Declaration and Calling
  - Function Parameters and Return Values
  - Default Parameters
  - Function Expressions
  - Arrow Functions
  - Anonymous Functions
  - Callback Functions
  - Higher-Order Functions
  - Hoisting in Functions
  - Function Scope vs Block Scope
  - Call, Apply, Bind Methods
-

---

## 7. Arrays

- What is an Array
  - Array Creation and Accessing Elements
  - Looping through Arrays
    - Traditional Loops
    - forEach()
  - **Common Array Methods:**
    - push(), pop(), shift(), unshift()
    - slice(), splice()
    - indexOf(), includes()
  - **Advanced Array Methods:**
    - map(), filter(), reduce()
    - find(), some(), every()
  - Array Destructuring
  - Array Spread Operator (...)
  - Multi-dimensional Arrays
- 

## 8. Objects

- What is an Object
  - Object Properties and Values
  - Adding, Updating, and Deleting Keys
  - Accessing Values (Dot & Bracket Notation)
  - Nested Objects
  - Object Methods
  - Shorthand Property Names
  - Object Destructuring
  - Object.keys(), Object.values(), Object.entries()
  - Spread and Rest Operators with Objects
- 

## 9. Strings, Numbers, Dates, and Math

### Strings

- String Creation and Access
- Common String Methods:
  - length, toUpperCase(), toLowerCase(), slice(), substring(), substr(), trim(), split(), includes(), replace(), etc.

### Numbers

- Number Methods:
  - toFixed(), parseInt(), parseFloat(), toPrecision()

### Dates

- Creating Date Objects
- Date Methods:getFullYear(), getMonth(), getDate(), etc.

## Math

- Common Math Methods:
    - Math.round(), Math.ceil(), Math.floor(), Math.random(), Math.pow(), Math.max(), Math.min()
- 

## 10. Understanding the DOM (Document Object Model)

- What is the DOM
  - Selecting Elements
    - getElementById()
    - getElementsByClassName()
    - querySelector() / querySelectorAll()
  - Changing Text, HTML, and Styles Dynamically
  - Creating and Removing Elements
  - DOM Tree Traversal
- 

## 11. Events in JavaScript

- What are Events
  - Adding Event Listeners
  - Common Events:
    - click, mouseover, mouseout, input, change, keydown, keyup, submit
  - Accessing Form Fields
  - Form Validation Using JavaScript
  - Displaying Error Messages Dynamically
  - Event Bubbling and Capturing
  - this keyword in Event Handlers
- 

## 12. Scope and Hoisting

- Global Scope
  - Local / Function Scope
  - Block Scope (let, const)
  - Hoisting of Variables and Functions
  - Temporal Dead Zone (TDZ)
- 

## 13. Error Handling

- try...catch...finally
  - throw Keyword
  - Error Object (name, message)
  - Custom Errors
-

## 14. JSON (JavaScript Object Notation)

- What is JSON
  - JSON Syntax and Rules
  - Converting JSON to JS Object
  - Converting JS Object to JSON
  - Practical Examples
- 

## 15. Web Storage

- Local Storage
  - Session Storage
  - Setting, Getting, and Removing Data
  - Example: Simple To-Do List using Local Storage
- 

## 16. Modules

- Introduction to JavaScript Modules
  - Export and Import Syntax
  - Default vs Named Exports
- 

## 17. Asynchronous JavaScript

- Synchronous vs Asynchronous Code
  - Event Loop and Call Stack Overview
  - Callbacks
  - Promises
    - .then() and .catch()
  - Async/Await Syntax
  - Chaining Promises
  - Error Handling in Async Code
  - Callbacks
  - Callback Hell
-

## 18. Additional Core Topics

- This keyword
  - Strict Mode ("use strict")
  - Template Literals and Tagged Templates
  - Destructuring in Arrays and Objects
  - Spread and Rest Operators
  - Short-circuit Evaluation (||, &&)
  - Default Function Parameters
  - Immediately Invoked Function Expressions (IIFE)
  - Prototypal Inheritance
  - Event Loop
  - Garbage Collection and Memory Management
  - Map, Set, Weak Map
  - Yield
  - Generators
  - Debouncing
  - Throttling
  - Currying
  - Event Delegation and Event Propagation
  - Polyfills
  - Event Loop, Call Stack, and Web APIs
  - Closures and Lexical Scope
  - Procedural Programming and Functional Programming Basics
  - Higher-Order Functions
  - Deep vs Shallow Copy
  - Optional Chaining (?.)
  - Nullish Coalescing Operator (??)
-

# REACT JS OUTLINE

## **1. Introduction to React**

- What is React and Why Use It
  - Features of React (Declarative, Component-Based, Virtual DOM)
  - Single Page Applications (SPA) Concept
  - **Project Setup**
    - Using Create React App (CRA)
    - Using Vite for Fast Builds
    - Folder Structure and File Organization
  - **JSX Basics**
    - What is JSX
    - JSX Rules & Syntax
    - Embedding Expressions in JSX
    - JSX vs HTML Differences
- 

## **2. React Components and Props**

- What are Components
  - **Functional Components**
    - Creating and Exporting Components
  - **Props (Properties)**
    - Passing Data Between Components
    - Props Destructuring
    - Default Props and PropTypes (Type Checking)
  - Component Composition and Reusability
- 

## **3. State and Events in React**

- Understanding Component State
  - **useState Hook**
    - Initializing and Updating State
    - Multiple State Variables
  - Event Handling in React
    - Handling Clicks, Inputs, and Form Events
    - Synthetic Events
- 

## **4. Conditional Rendering and Lists**

- Conditional Rendering with if, &&, and Ternary Operators
  - Rendering Arrays in JSX
  - Using map() for Dynamic Lists
  - Importance of Keys in Lists
-

## 5. React Forms and Controlled Components

- Controlled vs Uncontrolled Components
  - Handling Input Changes
  - Handling Form Submission
  - Validation in React Forms
  - Using Multiple Input Fields
- 

## 6. useEffect Hook and Side Effects

- Introduction to useEffect Hook
  - Performing Side Effects
  - Data Fetching using fetch() and axios
  - Cleanup Functions
  - Dependency Array and Common Mistakes
- 

## 7. React Router Basics

- What is React Router
  - Setting Up react-router-dom
  - Navigation using Link
  - Navigation using NavLink
  - Route Configuration with
  - Dynamic Routing with Parameters
  - Nested Routes
  - Protected Routes (Basic Concept)
- 

## 8. React Context API

- What is Context and Why Use It
  - Creating Context with
  - Providing Context
  - Consuming Context
  - Avoiding Prop Drilling
  - Example: Global Theme or Language Context
- 

## 9. useReducer Hook (State Management in React)

- Introduction to useReducer
  - When to Use useReducer vs useState
  - Understanding Reducer Functions and Action Objects
  - Dispatching Actions
  - Example: Counter App using useReducer
  - Combining useReducer with useContext for Global State
  - Comparison with Redux
-

## 10. Custom Hooks

- What are Custom Hooks
  - Why and When to Create Custom Hooks
  - Rules of Hooks
  - Examples:
    - useFetch()
    - useToggle()
    - useForm()
  - Best Practices for Reusability
- 

## 11. React Performance Optimization

- Why Optimization Matters
  - React.memo for Component Memoization
  - useCallback Hook
  - useMemo Hook
  - Avoiding Unnecessary Re-renders
  - Lazy Loading and Code Splitting with React.lazy and Suspense
- 

## 12. Error Handling in React

- Error Boundaries
  - Handling Errors in UI Gracefully
  - Try-Catch within Components
  - Fallback UI Components
- 

## 13. React Suspense and Lazy Loading

- What is React Suspense
  - Code Splitting using Dynamic Imports
  - Lazy Loading Components
  - Suspense Fallback UI
  - Performance Benefits
- 

## 14. API Integration in React

- Fetching Data using fetch() and axios
  - GET, POST, PUT, DELETE Requests
  - Handling Loading and Error States
  - Async/Await with Hooks
  - Displaying API Data in UI
-

## 15. Authentication and Authorization

- Login and Signup Flows
  - JWT (JSON Web Token) Basics
  - Storing Tokens (Local Storage / Cookies)
  - Protected Routes using React Router
  - Logout and Session Handling
- 

## 16. Deployment and Best Practices

- Preparing React App for Production
  - Deployment Platforms:
    - Vercel
    - External Server
  - Environment Variables in React
  - Folder Organization and Best Coding Practices
  - Using Prettier
  - Version Control with Git & GitHub
- 
- 

# REDUX

## 1. Introduction to Redux

- What is Redux and Why Use It
  - Core Principles of Redux (Single Source of Truth, State is Read-only, Pure Reducers)
  - Redux Architecture Overview
    - Store
    - Actions
    - Reducers
- 

## 2. Redux with React

- Installing and Setting Up Redux
  - Creating Store and Reducer
  - Connecting Redux to React
    - Using Provider from react-redux
    - useSelector() and useDispatch() Hooks
  - Dispatching Actions
  - Updating and Reading State
  - Example: Counter App using Redux
-

### 3. Redux Middleware (Thunk)

- What is Middleware
  - Introduction to Redux Thunk
  - Handling Asynchronous Actions
  - Fetching API Data using Thunk
  - Practical Example: Async API Call with Redux
- 
- 

## ZUSTAND

### 1. Introduction to Zustand

- What is Zustand and Why Use It
- Setting Up Zustand in a React App
- Creating a Simple Store
- Reading and Updating State
- Writing Actions in Zustand
- Comparison with Redux (Boilerplate vs Simplicity)

## CSS MODULES

### 1. CSS Modules

- What are CSS Modules
  - Advantages of CSS Modules
  - Creating and Using CSS Modules in React
  - Scoped Styling (Avoiding Global Conflicts)
  - Conditional Class Names in JSX
-

# SASS MODULES

## 1. Sass (Syntactically Awesome Style Sheets)

- What is Sass and Why Use It
  - Installing Sass in React
  - Sass Features:
    - Variables
    - Nesting
    - Mixins
    - Partials and Imports
  - Structuring SCSS Files
  - Using Sass with CSS Modules
  - Example Project Styling with Sass
- 

### Course Highlights

- 90 Hours of Training (1 Hour Daily, 2.5 Months)
- Complete Online Training
- Flexible Timing
- Complimentary Course - Github Essentials
- JavaScript & CSS Refreshers Included
- Complete React Basics → Advanced
- Redux & Zustand Covered
- Interview Training After Every 4 Topics
- Catch-up Sessions (Missing Topics Discussion)
- Hands-on Project & Deployment
- Interview Training till get hired
- Essential help after getting hired