

## **HTML**

### **Introduction to HTML**

- What is HTML?
- Basic structure of an HTML document
- Tags, elements, and attributes
- Headings, paragraphs, and line breaks
- Lists (ordered and unordered)
- Links and anchors
- Images and attributes

### **Intermediate HTML**

- Forms and input fields
- Buttons and labels
- Tables and table elements
- Semantic HTML (divs, spans, headers, footers)
- HTML entities and special characters
- Embedding multimedia (audio, video, iframes)
- HTML5 features (local storage, geolocation)

## **CSS**

### **Introduction to CSS**

- What is CSS?
- Inline, internal, and external stylesheets
- CSS syntax and selectors
- Basic styling (colors, fonts, text properties)
- Box model (margin, padding, border)
- Backgrounds and gradients

### **Intermediate CSS**

- Positioning (static, relative, absolute, fixed)
- Display property (block, inline, inline-block)
- Flexbox layout
- Grid layout
- Responsive design and media queries
- CSS preprocessors (Sass, Less)
- CSS frameworks (Bootstrap, Bulma)

## JavaScript

### Introduction to JavaScript:

- What is JavaScript?
- Overview of JavaScript: history, purpose, and usage.
- Setting up the development environment: text editors, browsers, and developer tools.
- Basic syntax: variables, data types, operators, and comments.
- Control Flow and Functions
- Conditional statements: if, else if, else.
- Loops: for, while, do-while.
- Functions: defining functions, parameters, return statements, and function expressions.
- Arrays and Objects
- Arrays: creating arrays, accessing elements, array methods (push, pop, shift, unshift, etc.).
- Objects: creating objects, accessing properties, object methods, and object-oriented programming concepts.
- DOM Manipulation
- Introduction to the Document Object Model (DOM).
- Selecting DOM elements: getElementById, querySelector, etc.
- Manipulating DOM elements: changing content, styling, adding/removing elements.

### Intermediate JavaScript:

- Events and Event Handling
- Understanding events in JavaScript.
- Adding event listeners: click, hover, submit, etc.
- Event handling techniques: event delegation, bubbling, and capturing.
- Error Handling
- Introduction to error handling in JavaScript.
- Using try-catch blocks.
- Throwing and catching custom errors.
- Asynchronous JavaScript
- Introduction to asynchronous programming.
- Callback functions.
- Promises: creating, chaining, and error handling.

### Advanced JavaScript:

- ES6 and Beyond
- Introduction to ES6 features.
- Arrow functions, template literals, destructuring, let and const, etc.
- Advanced array methods: map, filter, reduce.
- Modules and Module Bundlers

- Using ES6 modules.
- Introduction to module bundlers like Webpack or Rollup.
- AJAX and Fetch API
- Introduction to AJAX (Asynchronous JavaScript and XML).
- Using the Fetch API for making HTTP requests.
- Handling responses and error cases.
- Introduction to Node.js
- Overview of Node.js and its ecosystem.
- Setting up a Node.js environment.
- Basics of server-side JavaScript.
- Express.js and RESTful APIs
- Introduction to Express.js.
- Building RESTful APIs with Express.
- Handling HTTP methods (GET, POST, PUT, DELETE).
- Project-Based Learning
- Encourage learners to apply their knowledge through projects at each level.
- Examples could include building a simple to-do list app, a weather app using APIs, or a portfolio website with dynamic content.

## **ReactJS**

### **Introduction to ReactJS**

- Understanding the basics of ReactJS
- Setting up the development environment
- Creating your first React app

### **Components and Props**

- Exploring React components
- Passing data with props
- Building reusable components
- Conditional rendering and ternary operators

### **State and Lifecycle**

- Managing state in React components
- Understanding component lifecycle methods
- Handling user input and events

### **Handling Data**

- Fetching data from APIs
- Managing data with state and props

- Working with forms and form validation

## **Routing and Navigation**

- Implementing routing with React Router
- Creating nested routes and dynamic routing
- Building a multi-page React application

## **Building a Full-Stack Application**

- Creating a simple blog and to-do list app
- Integrating React with a backend Api
- Handling user authentication
- Deployment options