



MERN STACK DEVELOPMENT

8790732332, 9700610241

RD NO 1, KPHB, KUKATPALLY, HYD.

10000coders.in

Table of Contents

- 1. HTML Basics & Git**
- 2. CSS Fundamentals & Advanced Styling**
- 3. JavaScript Concepts**
- 4. React.js Concepts**
- 5. Node.js & Express.js for Backend**
- 6. MongoDB**
- 7. Full-Stack Application Development**



HTML AND GIT



1. Introduction to HTML & Basic Structure

- **What is HTML:** Understanding HTML's role in web development.
- **HTML Document Structure:** `<!DOCTYPE html>`, `<html>`, `<head>`, `<body>`, and the basic flow of an HTML document.
- **HTML Tags & Elements:** Understanding of HTML tags and elements, their structure (e.g., opening and closing tags), and how they form the building blocks of a webpage.

2. Text Formatting & Structuring Content

- **Headings (`<h1>` – `<h6>`):** Using headings for organizing content hierarchically; their significance for SEO and readability.
- **Paragraphs (`<p>`):** Creating and formatting paragraphs.
- **Inline vs Block Elements:** Explanation of inline elements (e.g., ``) and block elements (e.g., `<div>`).
- **Text Formatting Tags:** Basic formatting tags like ``, ``, ``, `<i>`, and `<u>` for emphasizing and styling text.

3. Links, Lists, and Media

- **Anchor Tags (`<a>`):** Creating links using the `<a>` tag, relative vs absolute URLs, and key attributes (`href`, `target`, `rel`).
- **Lists:** Creating ordered (``) and unordered lists (``) with list items (``), and nested lists.
- **Multimedia Tags:** Inserting images (``), audio (`<audio>`), video (`<video>`), and embedding content using `<iframe>`. Also, using the `<track>` tag for captions.



4. Forms, Tables & Semantic HTML

- **Forms:** Creating and handling forms with `<form>`, form controls like `<input>`, `<button>`, `<select>`, and `<textarea>`, and using attributes like `action`, `method`, and `name`.
- **Tables:** Structuring tables using `<table>`, `<tr>`, `<th>`, and `<td>`. Understanding how to format tables with attributes.
- **Semantic HTML:** Importance of using semantic tags like `<header>`, `<footer>`, `<section>`, `<article>`, and `<nav>` for better accessibility and SEO.

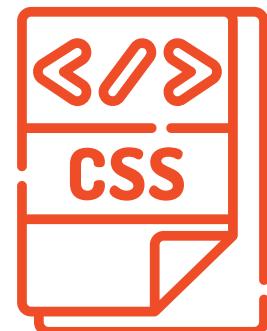
5. Additional HTML Concepts

- **Void Elements:** Understanding self-closing (void) tags like ``, `
`, `<input>`, `<hr>`, and their usage.
- **HTML Entities:** Representing special characters using HTML entities (e.g., `&amp`, `&lt`, `&gt`).
- **SEO and Meta Tags:** Importance of meta tags for SEO (`<meta name="description">`, `<meta charset>`).

6. Git As a Bonus (Version Control)

- **Introduction to Git:** What is Git? The importance of version control in web development.
- **Git Repositories:** Creating and cloning repositories, initializing a local repository with `git init`, and understanding the concept of remote repositories.
- **Git Workflow:** Basic Git commands:
 - `git add`: Staging changes.
 - `git commit`: Committing changes to the local repository.
 - `git push`: Pushing changes to the remote repository (e.g., GitHub).
 - `git pull`: Pulling updates from the remote repository.
- **Branching and Merging:** Introduction to branching in Git and merging branches to manage different versions of a project.
- **GitHub Basics:** Using GitHub for remote repository hosting and collaboration, understanding issues, pull requests, and collaboration workflows.

CSS Fundamentals & Advanced Styling



1. CSS Basics

- CSS Syntax & Structure
- Linking CSS to HTML (Inline, Internal, External)

2. CSS Selectors

- Simple Selectors (Element, Class, ID)
- Combinator Selectors (Descendant, Child, Sibling)
- Attribute Selectors
- Pseudo-Classes (e.g., :hover, :active)
- Pseudo-Elements (e.g., ::before, ::after)

3. CSS Box Model

- Box Model Overview (Margin, Border, Padding, Content)

4. Positioning & Display Properties

- Positioning (Static, Relative, Absolute, Fixed, Sticky)
- Display Property (Block, Inline, Inline-block, None)

5. CSS Layouts

- Flexbox Layout (Container, Items, Justify Content, Align Items)
- Grid Layout

6. CSS Advanced Techniques

- CSS Transitions
- CSS Animations (Keyframes)
- CSS Variables (Custom Properties)
- Media Queries
- Object-fit and Box-Shadow

JavaScript



1. JavaScript Basics

- Introduction to JavaScript
- Adding JavaScript to HTML
- JavaScript Syntax

2. Variables, Data Types, and Operators

- Variables (var, let, const)
- Data Types (String, Number, Boolean, etc.)
- Type Conversion
- Operators (Arithmetic, Comparison, Logical, etc.)

3. Control Flow

- Conditional Statements (if, else if, else, switch)
- Ternary Operator
- Loops (for, while, do...while, for-in and for-of)
- Break and Continue
- Nested Loops

4. Functions

- Function Declaration and Expression
- Parameters and Arguments
- Return Statement
- Anonymous Functions
- Arrow Functions
- Callback Functions
- Higher-Order Functions



5. Strings, Arrays and Objects, Number, Math and Dates

- Array Methods
- Iterating through Arrays
- String Methods
- Iterating through Strings
- Introduction to Objects
- Creating and Accessing Objects
- Number methods, Date Methods and Math methods

6. DOM Manipulation

- Introduction to DOM
- Selecting Elements (`getElementById`, `querySelector`, etc.)
- Modifying Elements (`innerHTML`, `textContent`, `style`)
- Adding and Removing Elements (`appendChild`, `removeChild`)
- Working with Attributes

7. Events

- Introduction to Events
- Event Listeners (`addEventListener`)
- Common Events (form events, mouse events, keyboard events)
- Event Propagation (Bubbling and Capturing)
- Event Delegation

8. Asynchronous JavaScript

- Introduction to Asynchronous JavaScript
- Timing Functions (`setTimeout`, `setInterval`, `clearTimeout`, `clearInterval`..)
- Promises
- Error Handling with Promises (`try`, `catch`, `finally`)
- Async and Await
- Fetch API with all CRUD operations



9. ES6+ Features

- Template Literals
- Destructuring Assignment
- Spread and Rest Operators
- Default Parameters
- Classes and Object-Oriented Programming
- Inheritance and Prototypes

10. Advanced JavaScript Concepts

- The this Keyword
- Constructor Functions
- Call, Apply, and Bind Methods
- Closures and Lexical Scope
- Immediately Invoked Function Expressions (IIFE)

11. Browser Object Model (BOM)

- Introduction to BOM
- Browser History
- Navigator Object
- Location Object
- Screen Object
- Window Object

12. Local Storage & Session Storage

- Introduction to Local Storage
- Introduction to Session Storage
- Differences Between Local and Session Storage
- Storing and Retrieving Data



13. Form Validation and Regular Expressions

- Form Validation in JavaScript
- Introduction to Regular Expressions
- Using Regular Expressions for Form Validation

React.js



1. Introduction to React & Setup

- What is React.js?
- Why Use React?
- Setting Up a React Project (Vite)
- React Folder Structure

2. Components & Styling

- Creating Components (Functional & Class)
- Nesting Components
- Implementing Styles in React
- Conditional Styling

3. Props & Data Rendering

- Rendering Array of Objects as Cards & Tables (map method)
- Introduction to Props
- Accessing & Passing Props Between Components
- React Bootstrap Setup & Usage
- Bootstrap Components: Cards, Accordions

4. State Management & Events

- Understanding state in Class Components (this.setState)
- Prop Drilling & Conditional Rendering
- State Updating in Class Components (Examples)
- Handling Form Events in Class Components
- Mini Project: Todo List (Adding Items to a List)



5. Lifecycle Methods & Hooks

- JSX Rules & Best Practices
- React Lifecycle Methods:
 - Mounting Phase (componentDidMount, HTTP Requests with Fetch vs Axios)
 - Updating Phase (componentDidUpdate)
 - Unmounting Phase (componentWillUnmount)
- getDerivedStateFromProps Method
- Lifecycle in Functional Components with Hooks (useState)
- Mini Project: Counter App using a Button Component or any other)

6. Form Handling & Validations

- Handling Form Events in React
- Submitting Form Data
- Rendering Data through Bootstrap Modals
- Stateless vs Stateful Components

7. Advanced Hooks & Optimization

- Introduction to Hooks & Their Rules
- useState Hook Deep Dive
- useEffect Hook (Multiple Phases)
- API Calls with useEffect
- Virtual DOM, Diffing & Reconciliation
- Pure Components
- Mini Project: Login API Request & User Details Modal

8. React Router

- React Router DOM Setup
- Dynamic Routing with Array of Objects & API Responses



9. Context API & State Management

- Understanding Context API
- Using Context API for Global State Management
- useMemo & useCallback Hooks
- Handling Component Unmounting

10. Firebase Integration

- Setting Up Firebase in a React Project
- Authentication with Firebase
- Firebase Firestore for Data Storage

11. Higher-Order Components (HOC) & Advanced React hooks

- Introduction to HOCs
- Implementing HOCs in React

12. Redux Fundamentals

- What is Redux?
- Redux Store, Actions & Reducers
- Connecting Redux to React (useSelector, useDispatch)

13. Redux Advanced Concepts

- Middleware in Redux
- Redux Thunk for Asynchronous Actions
- Managing Complex State with Redux Toolkit

Node.js Brochure



Introduction to Node.js

- What is Node.js?
- Why Use Node.js?
- Where is Node.js Used?
- How Node.js Works

2. Understanding APIs & HTTP Protocols

- What is an API?
- Types of APIs (REST vs SOAP)
- HTTP Methods (GET, POST, PUT, DELETE)
- HTTP Status Codes
- HTTP Headers

3. How the Internet Works & 3-Tier Architecture

- How Data Travels Over the Internet
- Understanding Client-Server Communication
- What is 3-Tier Architecture?

4. Node.js Fundamentals

- How Node.js Works
- Event-Driven Architecture in Node.js

5. Node.js Global Objects & REPL

- What is the Read-Eval-Print Loop (REPL)?
- Global Objects in Node.js

6. Modules in Node.js

- CommonJS Modules (require, module.exports)
- ES6 Modules (import, export)
- Creating Local Modules



7. Core Modules in Node.js

- File System (fs)
- Operating System (os)
- Path (path)
- URL (url)

8. Asynchronous Nature of Node.js

- How Node.js Handles Asynchronous Code
- Event Loop & Non-Blocking I/O

9. HTTP Module & Event-Driven Architecture

- What is the HTTP Module?
- Creating a Basic Web Server in Node.js
- Understanding Streams & Chunks
- How Web Servers Work

10. API Development in Node.js

- Getting Data from Clients
- Routing in Node.js
- Handling Parameters in Node.js

11. Express.js Fundamentals

- What are Frameworks & Libraries?
- Introduction to Express.js
- Creating REST APIs in Express
- Implementing the Same Concepts as in Core Node.js

12. Working with Packages & Middleware

- Multer: Handling File Uploads (Single & Multiple)
- Bcrypt: Password Hashing & Security
- Nodemailer: Sending Emails with Node.js
- MySQL2: Database Connectivity
- User Authentication: Login & Registration Implementation



- **13. Caching & Cookies in Node.js**
 - What is Caching?
 - Node-Cache Implementation
 - What are Cookies?
 - Managing Cookies in Node.js
- **14. Introduction to MongoDB**
 - What is MongoDB?
 - Why Use MongoDB with Node.js?
 - Basics of NoSQL Databases(covers all CRUD Operations)



Transforming how stuff moves



1600+

HIRING PARTNERS





NEXT
YOU



10000
CcodERS™
Right path for a Bright Career.

NEXT
YOU



NEXT
YOU



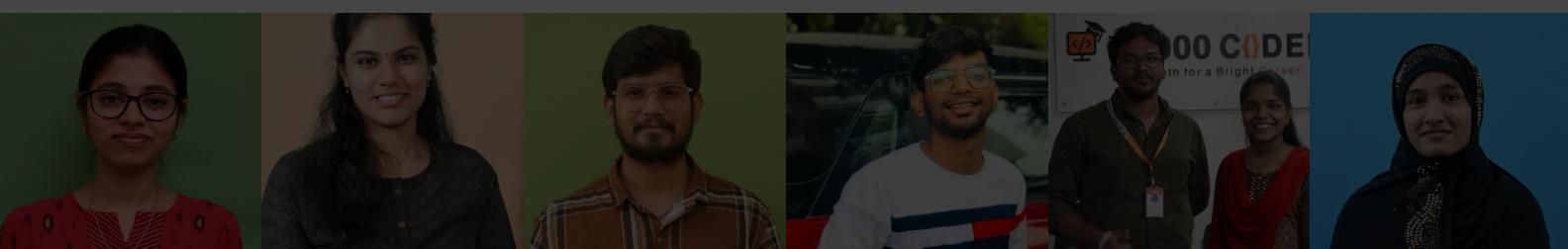
NEXT
YOU



2500+ Students Placed
1600+ Hiring Partners

NEXT
YOU

NEXT
YOU



Address

Rd no 1, KPHB, Kukatpally,
Hyderabad.

NEXT
YOU

NEXT
YOU



Contact

8790732332

9700610241

