React Js

HTML (HyperText Markup Language)

Basics

Introduction to HTML

Anatomy of an HTML document:

- <!DOCTYPE>
- <html>
- <head>
- <body>

Tags, elements, and attributes

Writing semantic HTML :

- <header>
- <footer>
- <article>

Core Elements

Headings : <h1> to <h6>

Paragraphs <p>

line breaks
 and horizontal rules <hr>

Lists:

- Ordered
- unordered
- description lists <dl>

Links <a>:

- Internal
- external
- anchor links

Images <img>: Attributes like src, alt, and title

Advanced Structure

Tables :

- <table>
- <tr>
- <td>
- <th>

Forms:

- <form>
- <input>
- <label>
- <textarea>
- <button>
- <select>
- <option>

Form attributes: action, method, type, name, id, validation (required, pattern)

Divs and spans (,) for layout

Semantic HTML5 tags: <nav>, <aside>, <section>, <main>

Additional Topics

HTML entities (e.g., &, <)

Embedding media: <audio>, <video>, <iframe>

Meta tags in head: SEO and viewport settings

Accessibility (ARIA roles, alt text, tabindex)

Progressive Enhancement and Graceful Degradation

HTML Project

This project introduces basic HTML concepts by building a simple static website. It includes essential elements like headings, paragraphs, images, links, lists, and tables. The goal is to understand webpage structure and how to organize content effectively.

CSS (Cascading Style Sheets)

Basics

Introduction to CSS: Purpose and syntax

Adding CSS: Inline, internal, external.

Selectors: Element, class, ID, universal, attribute selectors

Properties: color, fontsize, background, margin, padding

Layout and Box Model

Box model: content, padding, border, margin

Display properties: block, inline, inline block, none

Positioning: static, relative, absolute, fixed, sticky

Float and clear (legacy methods)

Styling

Typography: fontfamily, fontweight, textalign, lineheight

Colors: Hex, RGB, RGBA, HSL

Backgrounds: backgroundcolor, backgroundimage, gradients

Borders: borderwidth, borderstyle, borderradius

Advanced CSS

Flexbox: display: flex, justify-content, alignitems

CSS Grid: display: grid, grid template columns, gap

Pseudoclasses: hover, active, focus, nthchild

Pseudoelements: before, after

Transitions and Animations

Smooth transitions (transition, timingfunction, duration)

Keyframe animations (@keyframes, animationname, animationduration)

2D & 3D Transforms (transform: translate, rotate, scale, skew)

Responsive Design

Media queries: @media for screen sizes

Relative units: rem, em, %, vw, vh

Mobile first design principles

CSS Frameworks (Tailwind CSS, Bootstrap introduction)

CSS project

This project focuses on styling a basic web page using CSS. It covers topics like colors, fonts, box model, positioning, Flexbox, and Grid. The objective is to enhance the appearance of a webpage while making it responsive and visually appealing.

JavaScript

Basics

Introduction to JavaScript: Role in web development

Variables: var, let, const

Data types: Numbers, strings, booleans, arrays, objects

Operators: Arithmetic, comparison, logical

Control Flow

Conditionals: if, else if, else, switch

Loops: for, while, dowhile

Break and continue statements

Functions

Function declarations, expressions, and arrow functions

Parameters, return statements, default values

Scope: Global vs. local

DOM Manipulation

Selecting elements: getElementById, querySelector, querySelectorAll

Modifying elements: innerHTML, textContent, style

Event handling: addEventListener (click, input, submit)

ES6+ Features

Let & Const

Template literals

Destructuring

Spread and Rest operators

Modules: import and export

Promises and async/await

Advanced Topics

Asynchronous JavaScript: Callbacks, Promises, Fetch API

Error handling: try, catch, throw

Browser storage: localStorage, sessionStorage, Cookies

Security: XSS prevention, CORS, Content Security Policy

5. Frontend Frameworks & Libraries

Bootstrap (or Tailwind CSS)

Grid system

Typography and buttons

Components (Cards, Modals, Forms)

Javascript Project

This project introduces JavaScript to add interactivity to a webpage. It includes basic concepts like variables, functions, loops, and event listeners. The goal is to create simple features such as form validation, interactive buttons, and dynamic content updates.

React.js

JSX & Components

props in React Js

State Management (useState, useEffect)

React Router

API calls (fetch, Axios)

Performance optimization (useMemo, useCallback)

Custom Hooks

Material UI

Final Project

Features should include:

This project explores React.js by building a web application with reusable components. It covers concepts like JSX, props, state management, and event handling. The objective is to understand how React works and how to create a dynamic, user-friendly UI.

We are providing Training in Below Courses

Project Traning -:- **Internship**

C/C++	Python / Django	Data Analysis
DSA	React Js / Angular Js	Power BI
DBMS	MERN / MEAN	AI / ML
Mongo DB	Full Stack Development	Data Science
Java/Ad. Java		Generative AI

- ***Certified Courses.***
- ***100% job assistance*.***
- ***Courses with Hands-On Projects.***