

FrontEnd syllabus

Week 1: Introduction to HTML

- **Overview of HTML**
 - What is HTML?
 - Structure of an HTML document
 - Basic HTML tags (<html>, <head>, <title>, <body>)
- **Text Formatting**
 - Headings (<h1> to <h6>)
 - Paragraphs (<p>)
 - Emphasis and strong (,)
 - Lists (ordered and unordered , list items)
- **Attributes**
 - What are attributes?
 - Common attributes (id, class, style)
- **Creating Links and Images**
 - Hyperlinks ()
 - Images ()
- **Practical Exercises**
 - Create a simple webpage with text and images

Advanced HTML Elements

- **Tables**
 - Basic structure (<table>, <tr>, <td>, <th>)
 - Table attributes and styles
- **Forms**
 - Form elements (<form>, <input>, <textarea>, <button>, <select>)
 - Input types (text, password, submit, radio, checkbox, etc.)
 - Form validation basics
- **Semantic HTML5 Elements**
 - Introduction to semantic tags (<header>, <footer>, <section>, <article>, <aside>, <nav>)
- **Media Elements**
 - Embedding videos and audio (<video>, <audio>)
 - Embedding other media (iframes)
- **Practical Exercises**
 - Create a contact form
 - Create a webpage with a table and embedded video

Week 2: Introduction to CSS

- **Overview of CSS**
 - What is CSS?
 - How to link CSS to HTML (inline, internal, external)
- **Basic Selectors**
 - Type, class, and ID selectors
- **Color and Backgrounds**
 - Setting text color
 - Setting background color and images
- **Text Styling**
 - Font families, sizes, and styles
 - Text alignment, decoration, and transformation
- **Box Model**
 - Understanding the box model (margin, border, padding, content)
- **Practical Exercises**
 - Style a simple webpage with text and background styles

Intermediate CSS

- **Positioning**
 - Static, relative, absolute, fixed, and sticky positioning
- **Display and Visibility**
 - Block, inline, and inline-block
 - Visibility and display properties
- **Flexbox**
 - Introduction to Flexbox
 - Flex container properties
 - Flex item properties
- **Responsive Design Basics**
 - Media queries
 - Responsive units (em, rem, vh, vw)
- **Practical Exercises**
 - Create a responsive layout using Flexbox

Week 3: Advanced CSS

- **Grid Layout**
 - Introduction to CSS Grid
 - Grid container properties
 - Grid item properties

- **Transitions and Animations**
 - CSS transitions
 - Keyframe animations
- **Transformations**
 - 2D and 3D transforms (rotate, scale, translate, skew)
- **Pseudo-classes and Pseudo-elements**
 - Common pseudo-classes (
 - ,
 - ,
 -)
 - Common pseudo-elements (::before, ::after)
- **Practical Exercises**
 - Create an animated button
 - Design a complex layout using CSS Grid

Project and Best Practices

- **Project**
 - Develop a multi-page website combining all learned concepts
- **Best Practices**
 - Accessibility considerations
 - SEO basics for HTML
 - Performance optimization tips
- **Advanced Topics (Optional)**
 - CSS preprocessors (Sass, LESS)
 - CSS frameworks (Bootstrap, Tailwind CSS)
- **Review and Q&A**
 - Recap of key concepts
 - Addressing questions and challenges

JavaScript Basics

- **JavaScript Syntax:**
 - Variables, Data Types, Operators
- Control Flow (if-else, switch-case, loops)
- **Functions and Events:**
 - Function Declarations and Expressions

- Event Handling and DOM Manipulation
- **Asynchronous JavaScript:**
 - Callbacks and Promises
 - Introduction to Fetch API
- **Error Handling:**
 - Try-Catch in JavaScript
 - Handling Errors with Promises
- **Day5:**
 - **Practice:**
- Building an Interactive Web Page with JavaScript

Introduction to React.js

- **What is React.js?**
 - Overview of JavaScript libraries and frameworks
 - History and evolution of React
 - Advantages of using React

Setting Up the Development Environment

- **Installation**
 - Installing Node.js and npm
 - Setting up a React project using Create React App
 - Understanding the project structure
- **Tools and Editors**
 - Recommended code editors (VS Code, Sublime Text, etc.)
 - Browser extensions (React Developer Tools)

JSX and Rendering Elements

- **JSX Syntax**
 - Embedding expressions in JSX
 - JSX vs. HTML
 - JSX best practices
- **Rendering Elements**
 - Rendering elements to the DOM
 - Updating the rendered elements

Components

- **Function and Class Components**
 - Creating and using function components
 - Creating and using class components
- **Props and State**
 - Passing data with props
 - Handling state in class components
 - Using useState hook in function components

Component Lifecycle

- **Lifecycle Methods in Class Components**
 - Mounting, updating, and unmounting phases
 - Common lifecycle methods (componentDidMount, shouldComponentUpdate, etc.)
- **Hooks for Function Components**
 - useEffect for side effects
 - Custom hooks

Handling Events

- **Event Handling in React**
 - Handling events in function components
 - Binding event handlers in class components
- **Synthetic Events**
 - Understanding React's synthetic event system

Conditional Rendering

- **Rendering based on Conditions**
 - Using if-else statements
 - Conditional operators
 - Ternary operators
 - Switch-case statements

Lists and Keys

- **Rendering Lists**
 - Mapping over lists to render components
 - Using keys for list elements

Forms and Controlled Components

- **Working with Forms**
 - Controlled vs. uncontrolled components
 - Handling form submissions

- Validating form inputs

State Management

- **Lifting State Up**
 - Sharing state between components
- **Context API**
 - Creating and using context
 - Context consumer and provider
- **Redux (optional)**
 - Introduction to Redux
 - Setting up Redux in a React project
 - Actions, reducers, and store
 - Connecting React components to Redux

Routing

- **React Router**
 - Setting up React Router
 - Route, Link, and NavLink components
 - Nested routes
 - URL parameters and query strings

Advanced Topics

- **Higher-Order Components (HOCs)**
 - Understanding HOCs
 - Creating and using HOCs
- **Render Props**
 - Using render props for component logic
- **Portals**
 - Creating and using portals for rendering outside the DOM hierarchy
- **Error Boundaries**
 - Catching and handling errors in components

Performance Optimization

- **Optimizing React Apps**
 - React memoization
 - Code splitting with React.lazy and Suspense
 - Using React Profiler

Testing

- **Testing React Components**
 - Introduction to testing frameworks (Jest, React Testing Library)
 - Writing unit tests for components
 - Snapshot testing

Building and Deploying

- **Building React Apps**
 - Production build process
 - Analyzing the bundle size
- **Deploying React Apps**
 - Deployment options (Netlify, Vercel, GitHub Pages, etc.)
 - Setting up Continuous Integration (CI)

Practical Projects

- **Project Ideas**
 - Todo App
 - Weather App
 - E-commerce site
 - Blog platform

Resources and Further Learning

- **Official Documentation**
 - React.js official docs
- **Community and Support**
 - Stack Overflow, Reddit, GitHub issues
- **Tutorials and Courses**
 - Online platforms (Udemy, Coursera, freeCodeCamp)