

Course Contents

MERN Stack

Frontend Development

A. Introduction to HTML

- What is HTML, and why is it essential for web development?
- Differences between HTML4 and HTML5
- New web standards and evolution of HTML

B. Tags, Elements, and Attributes

- Inline vs. Block-level elements
- Custom data attributes (data-*)
- Global attributes in HTML5

C. Basic Syntax

- HTML document structure: <html>, <head>, <body>, and the importance of the DOCTYPE declaration
- Valid and invalid HTML syntax examples

D. Tables

- Advanced table styling using CSS
- Table layout techniques
- Responsive tables

E. Lists

- Customizing lists with CSS
- Nesting lists within other lists

F. Forms

- Form validation techniques in HTML5
- New input elements in HTML5: date, number, email, tel, etc.

G. Semantic and Non-Semantic Tags

- The transition from HTML4 to HTML5
- HTML5's new structural elements: <header>, <footer>, <article>, <section>, <nav>

- Deprecation and removal of tags in HTML5
- List of non-semantic tags and their usage (<div>,)

I. HTML5 Features

- Web Storage (localStorage, sessionStorage)
- Geolocation API
- HTML5 Forms and Validation

K. Forms Attributes

- Formaction , formmethod(get,post)
- HTML5 form validation attributes: pattern, required, minlength

L. Audio and Video Tags

- Implementing HTML5 audio and video
- Handling media events and controls with JavaScript

CSS

A. CSS Selectors and Attributes (Style, Title)

- Overview of common CSS attributes
- Types of Selectors
- Differences between id and class selectors

B. CSS Types (Inline, Internal, External)

- Advantages and disadvantages of each CSS method
- Linking external stylesheets and using inline styles

C. Box-Model

- Understanding the CSS box model (Content, Padding, Border, Margin)
- Modifying box model properties to affect layout

D. Display Property (Block, Inline, None)

- The difference between block, inline, and inline-block elements
- display: none vs visibility: hidden

E. Visibility-Hidden

- Practical usage of visibility property
- Difference between visibility: hidden and display: none

F. Position Property (Static, Relative, Absolute, Fixed)

- Positioning elements: static, relative, absolute, fixed, and sticky
- Practical examples of element positioning

G. Z-Index Property

- Layering elements using z-index
- Importance of stacking context

H. Combinators

- Descendant Selector (space), Child Selector (>), Adjacent Sibling Selector (+), General Sibling Selector (~)
- Practical examples for combining selectors

I. CSS Pseudo-Classes

- Pseudo-classes like :hover, :focus, :nth-child, and :not
- Styling interactive elements with pseudo-classes

J. CSS Pseudo-Elements

- Styling the first letter, first line, and content insertion using ::before, ::after
- Selection and ::selection pseudo-element

K. Static Web Page

- Overview of static web pages and their applications
- Building a simple static web page

L. Viewport Meta Tag

- Making websites mobile-friendly with the viewport meta tag
- How the viewport affects layout on different screen sizes

Advanced CSS

A. Background and Multiple Backgrounds

- Applying background colors, images, and gradients
- Layering multiple backgrounds

B. Font-Related Features

- Using custom web fonts via @font-face and Google Fonts
- Font properties: font-family, font-size, font-weight, font-style, etc.

C. Text-Effect and Box-Effect

- Text shadow, box shadow, text transform
- Adding 3D effects using CSS properties

D. Gradients (Linear and Radial)

- Creating linear and radial gradients with CSS
- Practical uses for gradient backgrounds

E. Transition

- Introduction to CSS transitions for smooth state changes
- Common transition properties: transition-property, transition-duration, transition-timing-function

F. Transformation

- Using transform to rotate, scale, skew, and translate elements
- Combining transformations for advanced effects

G. Media Queries

- Creating responsive layouts using @media queries
- Customizing designs for various screen sizes and devices

Bootstrap

A. Introduction to Bootstrap (Responsive)

- The power of Bootstrap's grid system
- Importance of mobile-first design

B. Typography

- Font styles, sizes, and alignments in Bootstrap
- Bootstrap typography classes

C. Tables

- Advanced table designs using Bootstrap
- Responsive tables with .table-responsive

D. Images, Buttons

- Using Bootstrap classes for styling images and buttons
- Responsive image classes: .img-fluid

E. Grid Structure and Types of Columns

- Understanding the Bootstrap grid system: Rows, Columns, and Breakpoints
- Creating flexible layouts with col-xs, col-sm, col-md, and col-lg

F. Forms

- Styling forms with Bootstrap's built-in form controls
- Using custom input groups and validation styles

G. Jumbotron

- Using .jumbotron for prominent page sections
- Customizing jumbotrons for different screen sizes

H. Navbar and Nav Tabs

- Building responsive navigation bars using Bootstrap's Navbar component
- Styling navigation tabs and pills

I. Carousel

- Creating Bootstrap Carousels for image sliders

J. Responsive Web Page

- Best practices for responsive design with Bootstrap
- How to make websites mobile-friendly using the grid system

JavaScript

A. Introduction to JavaScript

- JavaScript in web development
- History and evolution of JavaScript

B. Use of JavaScript

- JavaScript in interaction and behavior of websites
- Integrating JavaScript with HTML and CSS

C. Variables

- Declaring variables using var, let, and const

D. Keywords

- JavaScript reserved keywords and naming conventions

E. Data Types

- Primitive types (String, Number, Boolean, Null, Undefined)
- Non-primitive types (Objects, Arrays, Functions)

F. JS Conditions

- Using if, else, switch, and conditional operators

G. Loops

- Using for, while, and do-while loops
- Loop control with break and continue

H. Functions

- Defining and calling functions
- Function parameters and return values

I. Arrays:

- Creating and accessing arrays.
- Array methods (push(), pop(), shift(), unshift(), map(), filter(), reduce(), etc.).
- Iterating through arrays using for, forEach(), map().
- Destructuring of array

J. String and their method

- Using backticks for multi-line strings and variable interpolation.

K. Spread and Rest Operators:

- Spread (...) for arrays and objects.
- Rest (...) for function parameters.

L. Objects:

- Creating objects: object literals, new Object().
- Accessing object properties: dot notation, bracket notation.
- Methods inside objects.
- Destructuring of object

M. SetTimeout and SetInterval

- Using setTimeout() and setInterval() for time-based actions

N. HTML DOM

- Introduction to the Document Object Model (DOM)
- DOM methods for interacting with HTML elements

- Selecting DOM Elements:
 - getElementById (), getElementsByClassName()
 - Changing text content (innerText).
 - Changing HTML content (innerHTML).
 - Changing attributes (setAttribute(), getAttribute()).
 - Adding/removing elements (createElement(),removeChild(),appendChild())

O. Event Handling:

- Adding event listeners (addEventListener()).

P. Asynchronous JavaScript:

- Understanding synchronous vs asynchronous code.
- Callback functions.
- Promises
- Async/Await.
- Error Handling: try, catch, finally.

Q. JavaScript Classes and Prototypes:

- Introduction to ES6 Classes.
- Constructor function, this in classes.
- Inheritance and Polymorphism

R. JQuery and Its Methods

JQuery introduction ,syntax and its method

React

1. Introduction to React

- **What is React?**
 - Introduction to React: A component-based library for building user interfaces.
 - Understanding the component-based architecture and the importance of reusable components.
- **Installing React using Create React App**
 - Setup of a development environment with create-react-app.
 - Folder structure and basic setup.

2. JSX Syntax and Databinding

- **What is JSX?**
 - Writing HTML inside JavaScript (React's syntax extension).
- **Embedding expressions, attributes, and class names in JSX**
 - Expressions in JSX: {} syntax for embedding JavaScript.
 - Using className for class attributes in JSX.

3. Components

- **Functional Components vs. Class Components**
 - Key differences and when to use each type.
- **Props and State in React**
 - What are props? Passing data to components.
 - State in React: How to use state within components (useState hook).
- **Component Life Cycle**
 - Understanding lifecycle methods in class components (e.g., componentDidMount).
 - Using useEffect hook in functional components for side effects.
- **Component API and Props Validation**
 - Defining custom props and validation with PropTypes.

4. Event Handling in React

- **Handling user interactions:**
 - Handling events like onClick, onChange, and onSubmit.
 - Using functions to handle event-driven logic.

5. State Management

- **State and Lifecycle Methods**
 - Deep dive into state management and how to update state in functional components using useState.
- **Context API**
 - Managing global state without prop drilling using React Context API.
 - Passing data to components without manually passing props.

6. React Router

- **Setting up React Router for navigation**
 - Installing and configuring react-router-dom for routing.

- **Creating Routes and Navigation Links**
 - Defining routes and linking components via the <Route> and <Link> components.

7. React Forms and User Input

- **Controlled Components and UnControlled Components**
 - Managing form state using React's state management.
- **Form Submission, Validation, and Reset**
 - Handling form submission, validation, and resetting form fields.

8. React Hooks

- **Using Built-in Hooks**
 - useState, useEffect, useContext, useMemo, useReducer, useRef.
- **Custom Hooks**
 - Creating reusable logic using custom hooks.
- **Understanding useReducer and useContext**
 - Handling more complex state management with useReducer and global state using useContext.

9. Advanced React Concepts

- **Error Boundaries**
 - Handling errors in React applications
- **Higher-Order Components (HOCs)**
 - Understanding HOCs for reusing component logic.
- **Virtual DOM**
 - Explanation of how React's virtual DOM works and improves performance.

10. State Management Libraries

- **Redux**
 - Introduction to Redux (Actions, Reducers, and Store).
 - Integrating Redux with React to manage global state efficiently.
- **State Management with Context API**
 - Understanding when to use Context API vs. Redux for state management.

11. Integrating GraphQL with React

- **What is GraphQL?**

- Introduction to GraphQL and how it compares to REST APIs.

12. Working with HTTP Requests in React

- **Making HTTP Requests**
 - Sending GET and POST requests to external APIs.
 - Adjusting request headers, handling responses, and using Axios or Fetch API.

13. Bootstrap for Styling

- **Basic Setup of Bootstrap**
 - Integrating Bootstrap with React for responsive UI design.

14. Authentication in React Apps

- **Introduction to JWT**
 - How JWT (JSON Web Tokens) works for user authentication.

Backend Development

Node JS (24 hours)

- ❖ Introduction
 - What is Node JS?
 - Features of Node JS
 - Environment Setup
 -
- ❖ Module
 - Module Introduction
 - Exporting module
 - Importing module
- ❖ Http Module
 - Creating Server using HTTP module
 - Http module API
 - Request handler using HTTP module
 - Launching Http Node Server
 - Accessing using web browser
 - URL parameters
- ❖ Synchronous vs Asynchronous Programming

- What is synchronous programming?
- What is Asynchronous programming?

- ❖ File System Module
 - Importing fs module in Node JS App
 - Reading data from file
 - Writing data to file
 - Appending data to file

- ❖ Buffer
 - What is Buffer?
 - Creating and allocating Buffer
 - Buffer and it's methods

- ❖ Stream
 - What is Stream?
 - Types of Stream
 - Stream and it's methods

- ❖ Event Loop
 - Event Driven Programming
 - Event loop
 - Feature of Event Loop
 - How event loop works?

- ❖ Global Objects
 - Console
 - Process
 - setTimeout
 - setInterval
 - OS Module

- ❖ REPL Terminal
 - What is REPL?
 - Launching REPL
 - REPL Expression

Express JS

- ❖ Introduction
 - What is Express JS?
 - Features of Express JS
 - Environment Setup

- Installing express
- Creating first express app
- Handling request using get(), post(), put(), delete() methods

- ❖ Middlewares
 - What is Middleware?
 - How middleware works
 - Types of Middleware
 - App level middleware
 - Router level middleware
 - Built-in middlewares
 - Third party middleware

- ❖ Body Parser Middleware
 - JSON body parser
 - Raw body parser
 - URL encoded body parser
 - Text body parser

- ❖ Handling Cookies
 - What is Cookie?
 - Using Cookie Parser middleware
 - Creating and sending cookie as part of http response
 - Receiving Cookies from client

- ❖ Session Tracking
 - What is Session?
 - What is Session Tracking?
 - Session Management
 - Session Id
 - Using Express Session middleware

- ❖ Connecting to MongoDB
 - Using mongoose API
 - Mongoose Schema
 - Mongoose Model
 - Performing CRUD Operations

- ❖ Creating RESTful Web Service

MongoDB

- ❖ Introduction to NoSQL Database
 - What is NoSQL?
 - SQL vs NoSQL Database

- Features of NoSQL
- ❖ MongoDB Introduction
 - What is MongoDB?
 - Features of MongoDB
 - Environment Setup
- ❖ Working with Cloud Based MongoDB
 - Connecting to Cloud based Database using Mongo Shell
 - Listing all the databases in cloud based database
 - Creating Database
 - Creating Collection in Database
 - Creating Documents in Collection
 - Performing basic CRUD operations
 -
- ❖ Working with Local MongoDB
 - Connecting to Local Database using Mongo Shell
 - Listing all the databases in cloud based database
 - Creating Database
 - Creating Collection in Database
 - Creating Documents in Collection
 - Performing basic CRUD operations
- ❖ Query Documents
 - Comparison Operators
 - Relational Operators
 - Logical Operators
 - Array Operators
 - Regular Expression
 - Project certain fields
- ❖ Updating Documents
 - Adding new field in the document
 - Removing existing field in the document
 - Renaming the field
 - Incrementing numerical value
 - Using min, max, mul operators
 - Array update operators
- ❖ Aggregation Pipeline
 - What is Aggregation Pipeline?
 - Aggregation Pipeline Operators
 - Matching documents as per criteria
 - Limiting Documents
 - Sorting Documents
 - Grouping Documents
- ❖ Joining and Merging Documents

- Joining two different document using lookup operator
- Merging document using merge operator