

# MERN STACK DEVELOPMENT

Duration : 360hours

Course Fee:-16500

## HTML (HyperText Markup Language)

**Goal:** Understand how to build structure of webpages.

### 1. Introduction to HTML

- What is HTML, HTML syntax
- HTML Document Structure
- HTML tags and attributes
- HTML comments

### 2. Basic HTML Elements

- Headings <h1>—<h6>
- Paragraphs <p>
- Line breaks and horizontal rules <br> <hr>
- Text formatting tags (<b>, <i>, <u>, <em>, <strong>, <mark>)

### 3. Links and Media

- Anchor tags <a> (relative & absolute URLs, target)
- Images <img>
- Audio <audio>, Video <video>
- Iframes <iframe>

### 4. Lists and Tables

- Ordered <ol>, Unordered <ul>, Description lists <dl>
- Tables: <table>, <tr>, <td>, <th>, table attributes

### 5. Forms and Input

- <form>, <input> types
- <textarea>, <select>, <option>
- Labels, fieldsets, legends
- Form validation attributes (required, min, max, pattern)

### 6. Semantic HTML

- Semantic tags: <header>, <nav>, <section>, <article>, <aside>, <footer>
- Accessibility & SEO basics

## CSS (Cascading Style Sheets)

**Goal:** Style and design HTML pages.

## 1. Introduction to CSS

- Types of CSS: Inline, Internal, External
- CSS syntax and selectors
- Colors (named, HEX, RGB, HSL)

## 2. Selectors and Combinators

- Class, ID, element selectors
- Grouping, Universal, Attribute selectors
- Child, Descendant, Adjacent, Sibling combinators
- Pseudo-classes and Pseudo-elements

## 3. Box Model and Layout

- Margin, Border, Padding, Content
- Display property (block, inline, inline-block, none)
- Width, height, overflow

## 4. Positioning

- position (static, relative, absolute, fixed, sticky)
- z-index
- float and clear

## 5. Typography and Background

- Fonts, font families, web fonts
- text-align, text-transform, letter-spacing
- background-color, background-image, background-size

## 6. Advanced CSS

- Flexbox
- CSS Grid
- Media Queries (Responsive design)
- Transitions, Transforms, Animations
- CSS Variables

## JavaScript (JS)

**Goal:** Add interactivity and logic to web pages.

### 1. Introduction to JavaScript

- What is JS, adding JS to HTML
- console.log, comments, syntax rules

### 2. Variables and Data Types

- var, let, const
- String, Number, Boolean, Null, Undefined, Object

### 3. Operators & Control Structures

- Arithmetic, Comparison, Logical operators
- if-else, switch
- for, while, do-while loops

### 4. Functions

- Function declaration & expression
- Arrow functions
- Parameters, return values
- Scope and hoisting

### 5. Arrays and Objects

- Array methods (push, pop, shift, map, filter, reduce)
- Object creation, properties, methods
- JSON basics

### 6. DOM Manipulation

- Selecting elements (getElementById, querySelector)
- Changing content and styles
- Event handling (click, submit, keypress)
- Creating and removing elements dynamically

### 7. Advanced JS Concepts

- Callbacks, Promises, async/await
- ES6 features (destructuring, spread/rest, template literals, modules)
- LocalStorage / SessionStorage
- Error handling (try/catch)

## Bootstrap

**Goal:** Use prebuilt responsive CSS/JS components quickly.

### 1. Introduction

- What is Bootstrap
- Installing Bootstrap (CDN, npm)

### 2. Layout System

- Containers, Rows, Columns
- Grid system (12-column)
- Breakpoints & responsive design

### 3. Bootstrap Components

- Navbar, Buttons, Forms, Tables
- Cards, Badges, Alerts, Modals, Dropdowns

- List groups, Pagination, Breadcrumbs

#### 4. Utilities

- Colors, Spacing, Borders, Shadows
- Flex utilities, Alignment classes
- Display and visibility utilities

#### 5. Bootstrap JS Plugins

- Carousel
- Collapse / Accordion
- Tooltips, Popovers

## MySQL

**Goal:** Learn SQL for relational databases.

### 1. Introduction

- What is MySQL
- Installing MySQL, using MySQL Workbench

### 2. Database Basics

- CREATE DATABASE, USE
- CREATE TABLE, data types
- INSERT, UPDATE, DELETE

### 3. SELECT Queries

- SELECT columns
- WHERE, ORDER BY, LIMIT
- LIKE, IN, BETWEEN, IS NULL

### 4. Joins

- INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN
- UNION

### 5. Keys and Constraints

- PRIMARY KEY, FOREIGN KEY
- UNIQUE, NOT NULL, DEFAULT

### 6. Functions and Advanced

- Aggregate functions (COUNT, SUM, AVG, MIN, MAX)
- GROUP BY, HAVING
- Subqueries
- Views, Indexes, Transactions

# MongoDB

**Goal:** Learn NoSQL document database concepts.

## 1. Introduction

- What is MongoDB
- Installing MongoDB, Mongo Shell / Compass
- Document vs Table

## 2. Basic CRUD Operations

- insertOne(), insertMany()
- find(), findOne()
- updateOne(), updateMany()
- deleteOne(), deleteMany()

## 3. Query Operators

- \$gt, \$lt, \$in, \$or, \$and
- Sorting and projection

## 4. Collections and Indexes

- Creating and dropping collections
- Indexes

# React.js

**Goal:** Build dynamic front-end SPAs using components.

## 1. Introduction to React

- What is React and why use it
- Setting up React (Vite or Create React App)
- Folder structure

## 2. JSX and Components

- JSX syntax
- Functional components
- Props and PropTypes
- Conditional rendering

## 3. State and Events

- useState hook
- Handling events (onClick, onChange)
- Forms and controlled components

## 4. Lists and Keys

- Rendering lists with map()
- Keys concept

## 5. useEffect and Lifecycle

- useEffect hook
- Cleanup functions
- Fetching data from API

## 6. Routing

- React Router (BrowserRouter, Routes, Route, Link)
- Nested routes
- 404 Page

## 7. Advanced React

- Context API
- useReducer hook
- Custom hooks
- useRef, useMemo, useCallback

## 8. Project Structure and Best Practices

- Folder structure
- Component reusability
- Performance optimization
- Deployment (Vercel, Netlify)

# Node.js

## 1. Introduction to Node.js

- What is Node.js? (History, Features, Use Cases)
- Installing Node.js & npm
- Node.js Architecture (Single-threaded, Event-driven, Non-blocking I/O)
- REPL (Read-Eval-Print Loop)
- Running your first Node.js program

## 2. Core JavaScript Refresher (For Node.js)

- Variables (`let`, `const`, `var`)
- Functions & Arrow Functions
- Callbacks, Promises, `async/await`
- Modules (`require`, `import/export`)
- ES6+ features (Destructuring, Spread/Rest, Template literals)

## 3. Node.js Fundamentals

- Global Objects (`__dirname`, `__filename`, `process`)
- File System (`fs` module – read, write, append, delete files)
- Path module
- OS module
- Events & EventEmitter
- Buffers & Streams (Readable, Writable, Duplex, Transform)
- Timers (`setTimeout`, `setInterval`, `setImmediate`)

## 4. NPM (Node Package Manager)

- Installing & using packages
- Semantic versioning
- Local vs Global packages
- `package.json` & `package-lock.json`
- Custom scripts (`npm run`)
- Managing dependencies (`devDependencies`, `peerDependencies`)

## 5. Building Servers with Node.js

- HTTP module (Creating a basic server)
- Handling requests & responses
- Serving static files
- Routing (Manual implementation)

## 6. Express.js Framework

- Installing & setting up Express
- Middleware (Application, Router, Error-handling)
- Routing in Express
- Handling forms & JSON data
- Static files in Express
- Template engines (EJS, Handlebars, Pug)
- Express Router
- Error handling & Logging (`morgan`, `winston`)

## 7. Databases with Node.js

### (a) Relational Databases (SQL)

- MySQL/PostgreSQL integration
- CRUD operations using `mysql2` or `pg` package
- Query builder (`Knex.js`, `Sequelize ORM`)

### (b) NoSQL Databases

- MongoDB with Node.js (`mongodb` driver, `Mongoose`)
- Schema design, CRUD, population
- Indexes & Aggregations

## 8. Asynchronous Programmi

## ng in Node.js

- Callback functions
- Promises
- async/await
- Event Loop & Concurrency model
- Error handling in async code

---

## 9. RESTful API Development

- REST API basics v f      bm j
- CRUD operations with Express + DB
- Route parameters & query strings
- Postman / Thunder Client testing
- Status codes & response formats (JSON)

Achiever's Institute Of Computer Studies