# MERN FULL STACK DEVELOPMENT - SYLLABUS

### **Module 1: HTML**

**Objective**: Build strong foundational knowledge of web structure and semantics.

* **Introduction to HTML**
  + What is HTML?
  + Tools and Editors
* **Core Concepts**
  + HTML Basics: Elements, Attributes, Headings, Paragraphs, Styles
  + Semantic HTML: Layout (header, main, footer), Accessibility
* **Interactivity**
  + Forms: Input Types, Form Validation, Attributes
  + Multimedia: Images, Videos, Embeds
* **Responsive Design**
  + Meta Tags
  + Flexbox/Grid for layout structuring
  + HTML5 Semantic Elements

### **Module 2: CSS**

**Objective**: Style and layout web pages with an emphasis on responsiveness and modern practices.

* **CSS Foundations**
  + Syntax and Selectors
  + The Box Model
  + Colors (HEX, RGB, HSL)
* **Intermediate Styling**
  + Flexbox
  + Grid Layouts (including advanced topics like gap, auto-fit, and auto-fill)
  + Media Queries for Responsive Design
* **Modern Styling Techniques**
  + CSS Variables
  + Animations and Transitions
  + Pseudo-classes and Elements
* **Components and Utilities**
  + Styling Forms and Buttons
  + Utility Classes (e.g., Tailwind-like systems)

### **Module 3: JavaScript**

**Objective**: Introduce dynamic functionality and modern JavaScript concepts.

* **Foundations**
  + Introduction to JavaScript: Variables, Data Types, Loops, Conditionals
  + DOM Manipulation: Selecting and Styling Elements, Event Handling
* **Modern JavaScript**
  + OOPS - Objects & Classes
  + ES6+ Features: Let/Const, Arrow Functions, Template Literals
  + Spread/Rest Operators
  + Destructuring
* **Asynchronous JavaScript**
  + Promises and Fetch API
  + Async/Await
* **Interactive Web**
  + Working with Forms and Validation
  + Basic APIs and JSON

### **Module 4: MongoDB, Node.js, and Backend Development**

#### **Introduction to Backend Development**

* Overview of Backend and Full Stack Development
* Introduction to Node.js:
  + Event Loop and Non-blocking I/O
  + Asynchronous Execution and Callbacks

#### **Setting Up the Environment**

* Installing Node.js and NPM
* Setting up MongoDB (Local and Cloud with MongoDB Atlas)
* Using package.json and NPM for Dependency Management
* Nodemon for Development

#### **Node.js Fundamentals**

* Creating an HTTP Server with Node.js
* Understanding Streams and Buffers
* Working with Modules: Built-in, Third-party, and Custom

#### **Express.js Basics**

* Setting up Express.js and Middleware
* Creating and Handling Routes
* Understanding and Creating Custom Middleware

#### **Building RESTful APIs**

* What is REST?
* Designing Routes and Controllers
* CRUD Operations: GET, POST, PUT, DELETE
* Pagination, Sorting, and Searching Data
* Testing APIs with Postman

#### **Authentication and Security**

* Introduction to Authentication and Authorization
* Implementing JWT (JSON Web Token) for Authentication
* Building Signup and Login Features
* Handling Sessions and Cookies

#### **Database Integration with MongoDB**

* CRUD Operations with MongoDB:
  + Read, Insert, Update, Delete
  + Sorting, Counting, and Limiting
* Using MongoDB Operators: $eq, $ne, $lt, $gt, $in, etc.
* Designing and Optimizing MongoDB Schemas

#### **Advanced MongoDB Features**

* Indexing: Creating and Dropping Indexes
* Aggregations and Projections for Data Analysis
* Backup, Restore, Import, and Export Operations
* User Management and Role Assignment

#### **File Handling and Utilities**

* Uploading Files with Multer
* Logging with Tools like Winston or Morgan

#### **Deployment and CI/CD**

* Deploying Databases to MongoDB Atlas
* Setting Up Continuous Integration and Continuous Deployment (CI/CD)
* Basic Deployment Strategies for Backend Applications

### **Module 5: React JS**

#### **Week 1: Introduction to React**

* **Core Concepts**
  + What is React and Why Use It?
  + JSX and Virtual DOM
  + Setting Up a Project
  + Building and Running a React App

#### **Week 2: Components and State Management**

* **Components**
  + Component-Driven Development
  + State and Props
  + Functional vs. Class Components
  + Synthetic Events and Conditional Rendering
  + Rendering Lists with map() and Using Keys
* **Redux Basics**
  + Introduction to Redux
  + Actions, Reducers, and Store
  + Connecting Redux with React Components (React-Redux)
  + Redux DevTools

#### **Week 3: SPAs, HTTP, and Forms**

* **Single Page Applications (SPA)**
  + Introduction to SPAs
  + Container vs. Presentational Components
  + React Router Basics
* **HTTP and REST API Integration**
  + Fetching Data with Axios and Fetch API
  + Handling Promises and Errors
  + Understanding REST Concepts (Status Codes, Headers, Payloads)
* **Forms**
  + Building Forms with Formik
  + Validation with Yup
  + Posting Form Data

#### **Week 4: React Hooks and Advanced Concepts**

* **React Hooks**
  + Core Hooks: useState, useEffect, useReducer
  + Advanced Hooks: useMemo, useRef, useContext
  + Context API for State Management
  + Creating Custom Hooks
* **Miscellaneous Concepts**
  + Higher-Order Components (HOCs)
  + Styling Options in React (Inline Styles, CSS Modules, Tailwind CSS)

### **Module 6: Projects and Capstone**

#### **Project 1: Portfolio Website**

* **Description:** Build a personal portfolio using HTML and CSS.
* **Features:**
  + About Me section, skills, and projects list.
  + Responsive design with semantic HTML and CSS styles.
  + Image gallery and contact form.

#### **Project 2: E-commerce Landing Page**

* **Description:** Create a visually appealing e-commerce landing page.
* **Features:**
  + Responsive layout using CSS Grid and Flexbox.
  + Product cards with hover effects and animations.
  + Navigation bar, styled footer, and promotional banners.

#### **Project 3: Interactive Quiz App**

* **Description:** Develop a dynamic quiz app using JavaScript.
* **Features:**
  + Multiple-choice questions with a timer.
  + Real-time feedback and scoring system.
  + DOM manipulation for smooth navigation between questions.

#### **Project 4: Task Management API**

* **Description:** Build a RESTful API for task management.
* **Features:**
  + User authentication using JWT (Signup/Login).
  + CRUD operations for tasks (title, description, due date).
  + Pagination, filtering, and sorting.
  + MongoDB integration for data storage.

#### **Project 5: Weather Dashboard**

* **Description:** Create a weather dashboard using React and Redux.
* **Features:**
  + Fetch real-time weather data from a public API.
  + Navigation between city weather details using React Router.
  + State management with Redux.
  + Styled components for a professional UI.

#### **Capstone Project: TBD based on requirement!**