



MERN STACK

DURATION
90 Days

Career Track : Full Stack Developer

Mode : Online / Classroom

Certification : Skill Ascent – Certificate of Completion

Become a Certified Full Stack Developer with MERN

The MERN Stack Development Program is a comprehensive, industry-oriented course designed to equip you with end-to-end web development skills using the MongoDB, Express.js, React.js, and Node.js stack. This program is perfect for aspiring full stack developers who want to build dynamic, responsive, and scalable web applications.

Through hands-on projects, real-world scenarios, and practical coding exercises, you will learn how to:

- ◆ Build modern web applications with a robust backend and dynamic frontend
- ◆ Work with MongoDB for database management
- ◆ Implement server-side logic using Node.js and Express.js
- ◆ Develop interactive user interfaces with React.js
- ◆ Integrate APIs and implement RESTful services
- ◆ Deploy full-stack applications for real-world usage

By the end of the course, you will have a portfolio of live projects demonstrating your expertise in full stack development, making you industry-ready for roles like Full Stack Developer, Web Developer, or Software Engineer.

Course Introduction

Understanding Full Stack Development

- ❖ What is Full Stack Web Development?
- ❖ What is a Web Application?
- ❖ Roles & Responsibilities of a Full Stack Developer
- ❖ Types of Applications
 - Standalone Applications
 - Web Applications
 - Mobile Applications
- ❖ High-Level Architecture of Web Applications
- ❖ Technologies Used in Web Development
- ❖ Introduction to MERN Stack (MongoDB, Express, React, Node)
- ❖ Developer Tools Setup
 - VS Code Installation
 - Required Extensions
 - Node.js Installation
 - Git & GitHub Setup

Module 1: HTML5 – Structure of the Web

HTML Fundamentals

- ❖ Introduction & Purpose of HTML
- ❖ HTML Elements & Tags
- ❖ Structure of an HTML Document
 - Document Declaration
 - Head & Body
 - Scope of HTML

Types of Tags

- ❖ Paired Tags
 - Opening Tag
 - Closing Tag
- ❖ Self-Closing Tags

Core HTML Elements

- ◆ Heading Elements
- ◆ Paragraph Element
- ◆ Button Element
- ◆ Anchor Element (Inter & Intra Navigation)
- ◆ Image Element
 - Attributes: src, alt, width, height, title, id, class
- ◆ Table Element
 - Table Head, Body, Footer
 - Rows, Cells, Caption
 - Attributes: colspan, rowspan, align

Lists in HTML

- ◆ Ordered List
- ◆ Unordered List
- ◆ Description List
- ◆ Attributes & Markers

Forms & Inputs

- ◆ Form Element
- ◆ Input Types
- ◆ Select, Option, Optgroup
- ◆ Textarea, Label
- ◆ Input Attributes
 - required, placeholder, pattern, minlength, maxlength, readonly, disabled

Media & Semantic Elements

- ◆ Audio & Video
- ◆ Header, Footer, Main, Section, Nav, Article, Aside

Module 2: CSS3 – Styling & Layout

CSS Basics

- ◆ What is CSS?
- ◆ Purpose & Advantages of CSS
- ◆ Separation of Content and Design
- ◆ CSS Syntax & Rules

CSS Selectors

- ◆ Universal Selector
- ◆ Tag Selector
- ◆ Class Selector
- ◆ ID Selector
- ◆ Group Selector
- ◆ Descendant & Child Selector

CSS Box Model

- ◆ Margin
- ◆ Padding
- ◆ Border
- ◆ Width & Height

Layout Techniques

- ◆ Positioning
 - Static, Relative, Absolute, Fixed, Sticky
- ◆ Flexbox
 - Flex Direction
 - Justify Content
 - Align Items
 - Order & Flex Basis
- ◆ Grid Layout
 - Grid Columns & Rows
 - Gap & Alignment

Responsive Design

- ◆ Media Queries
- ◆ Breakpoints
- ◆ Orientation Handling

CSS Effects

- ◆ Backgrounds & Colors
- ◆ Transforms (Translate, Rotate, Scale)
- ◆ Transitions
- ◆ Box Shadow

Module 3: Bootstrap – Responsive UI

- ◆ Introduction to Bootstrap
- ◆ Grid System
- ◆ Typography
- ◆ Buttons
- ◆ Cards
- ◆ Forms
- ◆ Tables
- ◆ Pagination
- ◆ Modal
- ◆ Carousel
- ◆ Utility Classes
- ◆ Responsive Layout Design

Module 4: JavaScript – Programming Fundamentals

JavaScript Basics

- ◆ What is JavaScript?
- ◆ Variables (var, let, const)
- ◆ Data Types
- ◆ Operators
- ◆ Control Statements
- ◆ Loops

Functions

- ◆ Function Declaration
- ◆ Parameters & Arguments
- ◆ Arrow Functions
- ◆ Callback Functions
- ◆ Higher Order Functions
- ◆ Async Functions

Arrays & Objects

- ◆ CRUD Operations
- ◆ Array Methods
 - map, filter, forEach, slice, splice
- ◆ Object Methods
- ◆ Spread & Rest Operators
- ◆ Destructuring

DOM Manipulation

- ◆ DOM Structure
- ◆ Selecting Elements
- ◆ Modifying Content
- ◆ Creating & Removing Elements

Advanced JavaScript

- ◆ Scope & Closures
- ◆ this Keyword
- ◆ Call, Apply, Bind
- ◆ Promises
- ◆ Async / Await
- ◆ Fetch API
- ◆ Modules (Import & Export)

Module 5: React JS – Frontend Development

React Fundamentals

- ◆ Introduction to React
- ◆ Virtual DOM
- ◆ JSX
- ◆ Components (Functional & Class)
- ◆ Props & State

Hooks

- ◆ useState
- ◆ useEffect
- ◆ useContext
- ◆ useRef
- ◆ useReducer

Routing

- ◆ React Router
- ◆ BrowserRouter
- ◆ Routes & Route
- ◆ Nested Routing
- ◆ useParams & useNavigate

Forms & Data Handling

- ◆ Controlled Components
- ◆ Axios Integration
- ◆ API Consumption
- ◆ Context API
- ◆ Redux Basics

Performance

- ◆ Lazy Loading
- ◆ Optimization Techniques

Module 6: Backend – Node.js & Express.js

Node.js Fundamentals

- ◆ JavaScript Runtime
- ◆ Event Loop
- ◆ Asynchronous Programming
- ◆ Modules
- ◆ File System
- ◆ HTTP Module

Express.js

- ◆ Creating Servers
- ◆ Routing
- ◆ Middleware
- ◆ REST API Development
- ◆ Error Handling
- ◆ Authentication & Authorization
- ◆ JWT Implementation

Module 7: MongoDB & Mongoose

MongoDB Basics

- ◆ NoSQL Concepts
- ◆ CRUD Operations
- ◆ Data Types
- ◆ Relationships
- ◆ Embedded Documents

Advanced MongoDB

- ◆ Operators
- ◆ Indexing
- ◆ Aggregation Pipeline
- ◆ Text Search

Mongoose

- ◆ Schemas & Models
- ◆ Validation
- ◆ CRUD with Mongoose
- ◆ Relationships

Module 8: MERN Integration & Deployment

- ◆ MERN Architecture
- ◆ Connecting React with Backend
- ◆ JWT-based Authentication
- ◆ Role-Based Access
- ◆ Environment Variables
- ◆ Deployment
- ◆ Frontend: Vercel
- ◆ Backend: Render / AWS
- ◆ Git & GitHub Workflow

Final Capstone Project

End-to-End MERN Stack Application

- ◆ Authentication
- ◆ CRUD Operations
- ◆ API Integration
- ◆ Deployment
- ◆ Industry-Level Architecture

Career Outcomes

- ◆ Full Stack Developer
- ◆ MERN Stack Developer
- ◆ React Developer
- ◆ Backend Developer
- ◆ Web Application Developer