Full Stack Web Development with MERN

Duration: 100 hours
Objective: To acquire the knowledge of Full Stack Web Development using NodeJs, ReactJS and MySQL.
Prerequisites: Students are expected to know any OOP's Based Language. They should have undergone the Web Programming basics which includes HTML, CSS, JavaScript, Knowledge of any database is required.

Module 1: Introduction to Web

- Brief history of the Internet, How does the Internet work?
- Internet Protocol, Domain Name Service servers
- HTTP Protocol, Web Server vs Application Server
- Architecture of the Web

Module 2: HTML & HTML5

- Introduction to HTML, Basic HTML Tags
- HTML Form & Controls, HTML5: New features in HTML5

Module 3: Cascading Style Sheets (CSS)

- Introduction to CSS, Styling HTML with CSS, Structuring pages with CSS,
- Inline CSS, Internal CSS, External CSS
- CSS Selectors
  - Linking a style to an HTML document
- Responsive Web Design with Bootstrap

Module 4: JavaScript

- Introduction to JavaScript
- Variables in JavaScript, Statements, Operators, Comments, Expressions, and Control Structures
- JavaScript Scopes, Strings, Numbers, Date,Arrays, Array Methods

Module 5: JavaScript

- Objects, Object Definitions, Object Properties, Object Methods, Object Prototypes
- Functions, Function Definitions, Function Parameters, Function Invocation, Function Closures

Module 6: JavaScript

- Document Object Model (DOM)
  - o Object hierarchy in JavaScript
  - o HTML DOM, DOM Elements, DOM Events
  - o DOM Methods, DOM Manipulation, Forms & Forms Validation

Module 7: JSON & Ajax

- JSON: JavaScript Object Notation (JSON)
  - o Introduction and need of JSON o JSON Syntax Rules
  - o JSON Data - a Name and a Value,
  - o JSON Objects, JSON Arrays, JSON Files, JSON parsing
- Ajax
  - o Introduction to Ajax, Ajax Framework, Ajax Architecture
  - o Web services and Ajax, Ajax using JSON and jQuery

Module 8: Introduction to Node.js

- Introduction to Node.js, Browser JS vs. Node.js
- ECMAScript 2015 (ES6), Node.js REPL

Module 9: Node.js Asynchronous Programming

- Introduction to Asynchronous programming and callbacks
- Promises and async & await
- The Event Loop and Timers

Module 10: Node.js Modules

- Understanding Node modules, exports, and require
- Introduction to npm
  - package.json and package-lock.json files
  - Install, update, and manage package dependencies
  - Local and global packages

Module 11: Node.js Modules – *fs* and *http*

- File I/O – Sync & Async Methods
- HTTP Module – Building an HTTP server
- Developing a Node web application

Module 12: Introduction to Express

- Introduction to Express, Getting started with Express
- Application, Request and Response Objects
- Routes and Middlewares

Module 13: CURD using Express & MYSQL:

- Working with MYSQL
- Performing CURD operations with Express

Module 14: Introduction to React JS:

- Introduction to React, Getting started with React
- React Elements and React Components
- Function and Class Components
- Working with React Components and Props
  - Compose components
  - Render components
  - Declutter components

Module 15: React JS:

- Introduction to State and Lifecycle
- Stateful components and lifecycle methods
- Props vs. State vs. Context
- Handling events
- Conditional rendering

Module 16: React JS

- Lists and Keys
  - Rendering Multiple Components
  - Basic List Component
- Working with forms and inputs
- Composition vs. Inheritance
  - Containment, Specialization

Module 17: Express & React JS

- Build React App, Merging React with Express

Module 18: Capstone Project

Problem Solving using C & C++

Duration: 100 Hours

Course Objectives:

- To build logical thinking and problem-solving skills using programming.
- To develop a strong foundation in C and C++ Programming languages.
- To apply algorithmic thinking for solving real-world problems.

Prerequisite: Familiarity with computer fundamentals and Programming.

Module 1: Introduction to Programming & Problem Solving

Understanding problem-solving techniques, Algorithms, flowcharts, and pseudocode, Introduction to programming paradigms, Overview of C and C++ development environments, Writing and executing the first C program

Module 2: Programming Fundamentals Using C

Data types, variables, constants, and operators
Control structures: if, if-else, switch-case, loops (for, while, do-while)
Functions, recursion, and scope of variables
Arrays & String manipulation
Pointers and dynamic memory allocation

Module 3: Structured & Object-Oriented Programming in C++

Differences between C and C++, Classes and objects
Function and Operator overloading, Constructors, destructors, and member functions
Inheritance, Exception handling
File handling in C++, Templates in C++

Module 4: Data Structures & Algorithms

Linear data structures: arrays, stacks, queues, linked lists
Searching and sorting, Time and space complexity, Big-O notation

Module 5: Practical Problem Solving (20 Hours)

Hands-on coding exercises, Debugging strategies and best practices
Solving standard problems on strings, arrays, and recursion
Competitive programming basics

Module 6: Capstone Project