



Course Outline

Course Information

Course Title	Full Stack Web Development	
Type	Short Course	96 Days
Class Schedule	4 Days Per Week	
Duration	6 Months	

Eligibility Students & Beginners Interested in Web Development



Course Description

This practical-based Full Stack Web Development course is specially designed for students who want to build a successful career in the IT industry. The course focuses on modern website design and development techniques through 100% hands-on training and live project work. Students will learn professional frontend and backend development, responsive website creation, UI designing, database integration, and real-world deployment methods using industry-standard tools and technologies. By the end of the course, students will be able to develop complete dynamic websites and work confidently on freelance and professional projects.

Program Learning Outcome(PLO)

Program Learning Outcome

Description

PLO1	Advanced Full Stack Development Expertise	Develop professional, responsive, and high-performance web applications using modern frontend and backend technologies with industry-standard practices.
PLO2	Professional Project Development	Design, develop, and deploy real-world dynamic web applications through live project-based learning and practical implementation.
PLO3	Analytical & Problem-Solving Skills	Identify, analyze, and solve complex development challenges using logical thinking, debugging techniques, and modern development approaches.
PLO4	Modern Development Tools & Frameworks	Effectively utilize advanced development tools, frameworks, databases, APIs, and deployment platforms used in the software industry.
PLO5	Career & Freelancing Readiness	Build the professional skills required for freelancing, teamwork, client handling, and successful careers in the IT and software development industry.

Course Objectives (CO)

This course assumes that students design and develop fully functional website. When students complete this course, they will be able to;

(CO)	Objective	PLO
CO1	Understand the fundamentals of modern web development and the architecture of frontend and backend systems.	PLO1
CO2	Learn and implement the core concepts, syntax, and structure of modern web development technologies and programming languages.	PLO2
CO3	Design and develop responsive, interactive, and database-driven web applications using industry-standard development practices.	PLO2
CO4	Build real-world projects and practical web solutions to enhance problem-solving, teamwork, and professional development skills.	PLO5

Course Learning Outcomes (CLO)

At the end of this course students will be able to;

No.	Outcome	Relation with PLO	BT LEVEL	PLO LEVEL
CLO1	Understand the fundamental concepts, architecture, and workflow of modern web application development.	PLO1	C4	I
CLO2	Develop responsive and dynamic websites using frontend and backend development technologies.	PLO2	C3	I
CLO3	Apply modern development tools, frameworks, and databases in practical web development projects.	PLO3	C2	I
CLO4	Analyze real-world problems and create practical web-based solutions through live project implementation.	PLO4	C4	I
CLO5	Demonstrate professional, teamwork, and freelancing skills through project-based learning activities.	PLO5	C3	I

Lecture type Lectures, Lab sessions

WHAT STUDENTS WILL LEARN IN HTML5

1. Introduction to Web Development

- 1.1 How websites work
- 1.2 Frontend vs Backend Development
- 1.3 Client & Server Architecture
- 1.4 Domain & Hosting Basics
- 1.5 Static vs Dynamic Websites
- 1.6 Modern Website Technologies
- 1.7 Browser Rendering Process
- 1.8 Internet & WWW Fundamentals

2. HTML5 Fundamentals

INTRODUCTION

HTML BASIC

HTML4 DEFF HTML5

TAGS

PARAGRAPH

IMAGES

LISTS

DIV ELEMENTS

SPAN

TABLES

MAIN SEMANTIC ELEMENTS

HEADER

NAV

SECTION

FORM

ASIDE

ARTICLE

FOOTER



WHAT STUDENTS WILL LEARN IN CSS 3

1. CSS Introduction

INTRODUCTION CSS

COLOR

BACKGROUND

BORDER

PADDING

MARGIN

HEIGHT

WEIGHT

FONT

TEXT

HTML SELECTORS

BOX MODEL

OPACITY

POSITION

PSEUDO ELEMENT

Z-INDEX

SHADOW

BUTTON

FLEXBOX

ALIGN

FLOAT

OUTLINE

RESPONSIVENESS



Advanced JavaScript (ES6+)

1. JavaScript Fundamentals

- 1.1 Introduction to JavaScript
- 1.2 Role of JavaScript in Modern Web Development
- 1.3 JavaScript Execution Flow
- 1.4 Variables (var, let, const)
- 1.5 Data Types & Type Conversion
- 1.6 Operators & Expressions
- 1.7 User Input & Output Methods

2. Conditional Logic & Decision Making

- 2.1 If / Else Statements
- 2.2 Nested Conditions
- 2.3 Switch Statements
- 2.4 Ternary Operators
- 2.5 Real-World Logic Building

3. Loops & Iteration

- 3.1 For Loop
- 3.2 While Loop
- 3.3 Do While Loop
- 3.4 Nested Loops
- 3.5 Loop Optimization Techniques

4. Functions & Advanced Function Concepts

- 4.1 Function Declaration
- 4.2 Function Expressions
- 4.3 Arrow Functions
- 4.4 Parameters & Arguments
- 4.5 Return Statements
- 4.6 Callback Functions
- 4.7 Higher Order Functions

5. Projects

- 5.1 Calculator Logic
- 5.2 Dynamic Greeting System



6. Arrays & Array Methods

6.1 Arrays Introduction

6.2 Array Manipulation

6.3 Push, Pop, Shift, Unshift

6.4 Map, Filter, Reduce

6.5 Find & Sort Methods

Real-World Practice

7.1 Product Filtering System

7.2 Student Result System

Objects & JSON

8.1 Object Creation

8.2 Nested Objects

8.3 Object Methods

8.4 JSON Introduction

8.5 JSON Parsing & Stringify

Mini Project

9.1 Student Management System

10. DOM Manipulation Mastery

11. DOM Fundamentals

11.1 Selecting Elements

11.2 Changing Content Dynamically

11.3 Modifying Styles

11.4 Creating & Removing Elements

12. Events Handling

12.1 Click Events

12.2 Keyboard Events

12.3 Mouse Events

12.4 Form Events

12.5 Event Bubbling



13. Advanced DOM Projects

- 13.1 Interactive Quiz App
- 13.2 Dynamic Todo Application
- 13.3 Notes Management System

14. Forms Validation & Authentication Logic

- 14.1 Form Validation Techniques
- 14.2 Email Validation
- 14.3 Password Validation
- 14.4 Error Handling
- 14.5 Dynamic Validation Messages

15. Projects

- 15.1 Advanced Login Form
- 15.2 Registration System

16. Modern JavaScript ES6+

- 16.1 Template Literals
- 16.2 Destructuring
- 16.3 Spread Operator
- 16.4 Rest Parameters
- 16.5 Default Parameters
- 16.6 Modules in JavaScript

17. Practical Work

- 17.1 Dynamic Data Rendering
- 17.2 Reusable Components Logic

18. Local Storage & Session Storage

- 18.1 Local Storage Basics
- 18.2 Session Storage
- 18.3 Saving User Data
- 18.4 Persistent Applications

19. Projects

- 19.1 Notes Saving App
- 19.2 Task Manager



20. Fetch API & API Integration

20.1 What is API?

20.2 Fetch API

20.3 GET & POST Requests

20.4 JSON Data Handling

20.5 Error Handling in APIs

21. API-Based Projects

21.1 Weather Application

21.2 Currency Converter

21.3 Live News App

21.4 Movie Search App

22. Asynchronous JavaScript

22.1 Synchronous vs Asynchronous

22.2 Callbacks

22.3 Promises

22.4 Async / Await

22.5 Error Handling

23. Advanced Projects

23.1 Real-Time Data Fetching

23.2 API Dashboard

24. Advanced JavaScript Concepts

24.1 Closures

24.2 Scope & Hoisting

24.3 Event Loop

24.4 Debouncing & Throttling

24.5 JavaScript Memory Basics

25. Industry-Level Concepts

25.1 Clean Code Principles

25.2 Scalable JavaScript Architecture

25.3 Reusable Logic Systems

26. Advanced UI Interactions

26.1 Dark Mode System

26.2 Theme Switcher

26.3 Dynamic Sliders

26.4 Accordions



26.5 Tabs System
26.6 Search Filters

27. UI Projects

27.1 E-Commerce Filter System
27.2 Interactive Portfolio

28. JavaScript Animation Projects

28.1 Typing Effect
28.2 Scroll Animations
28.3 Animated Counters
28.4 Image Sliders
28.5 Loading Screen

29. API Integration Concepts

29.1 Introduction to APIs
29.2 REST API Fundamentals
29.3 Fetch API
29.4 Axios Basics
29.5 GET, POST, PUT & DELETE Requests
29.6 JSON Data Handling
29.7 API Authentication Basics
29.8 Error Handling & Loading States

Master Level JS Projects

1. Advanced E-Commerce Website

1.1 Product Filtering
1.2 Add to Cart
1.3 Dynamic Pricing
1.4 Search System

2. AI Todo Productivity App

2.1 Local Storage
2.2 Dynamic Tasks
2.3 Smart Filters

3. Online Quiz & Exam Portal

3.1 Timer System
3.2 Dynamic Questions
3.3 Result Calculation

4. Weather & Travel Guide App

4.1 Weather API
4.2 Location Search
4.3 Dynamic Backgrounds

5. Finance Tracker & Currency Converter

5.1 Real-Time Currency API
5.2 Expense Tracking
5.3 Dynamic Charts Concepts

Advanced Class Activities

6.1 Logic Building Challenges
6.2 Live Coding Competitions
6.3 API Integration Workshop
6.4 JavaScript Debugging Sessions
6.5 Dynamic UI Recreation Tasks

Advance Level Bootstrap

1. Bootstrap Introduction

- 1.1 What is Bootstrap?
- 1.2 Importance of Bootstrap in Modern Web Design
- 1.3 Bootstrap Versions
- 1.4 Installing Bootstrap
- 1.5 CDN vs Local Files
- 1.6 Bootstrap File Structure

2. Bootstrap Grid System

- 2.1 Containers
- 2.2 Rows & Columns
- 2.3 Grid Breakpoints
- 2.4 Responsive Columns
- 2.5 Nested Grids
- 2.6 Alignment & Ordering

3. Bootstrap Typography

- 3.1 Headings
- 3.2 Display Headings
- 3.3 Text Colors
- 3.4 Text Alignment
- 3.5 Font Weight & Styling
- 3.6 Blockquotes

4. Bootstrap Colors & Backgrounds

- 4.1 Theme Colors
- 4.2 Background Utilities
- 4.3 Gradient Backgrounds
- 4.4 Text Utilities
- 4.5 Opacity Utilities

5. Bootstrap Spacing & Sizing

- 5.1 Margin Classes
- 5.2 Padding Classes
- 5.3 Width & Height Utilities
- 5.4 Spacing Best Practices
- 5.5 Responsive Spacing



6. Bootstrap Buttons

- 6.1 Button Classes
- 6.2 Outline Buttons
- 6.3 Button Sizes
- 6.4 Button Groups
- 6.5 Icon Buttons

7. Bootstrap Forms

- 7.1 Form Controls
- 7.2 Input Types
- 7.3 Select Menus
- 7.4 Checkboxes & Radios
- 7.5 Floating Labels
- 7.6 Form Validation

8. Bootstrap Tables

- 8.1 Basic Tables
- 8.2 Striped Tables
- 8.3 Hoverable Tables
- 8.4 Responsive Tables
- 8.5 Table Utilities

9. Bootstrap Images & Media

- 9.1 Responsive Images
- 9.2 Image Thumbnails
- 9.3 Rounded Images
- 9.4 Image Alignment
- 9.5 Responsive Embeds

10. Bootstrap Cards

- 10.1 Basic Cards
- 10.2 Card Headers & Footers
- 10.3 Card Groups
- 10.4 Profile Cards
- 10.5 Pricing Cards
- 10.6 Service Cards

11. Bootstrap Navbar & Navigation

- 11.1 Navbar Basics
- 11.2 Responsive Navbar
- 11.3 Navbar Brand



- 11.4 Dropdown Menus
- 11.5 Sticky Navbar
- 11.6 Offcanvas Navigation

12. Bootstrap Components

- 12.1 Alerts
- 12.2 Badges
- 12.3 Breadcrumbs
- 12.4 Progress Bars
- 12.5 Spinners
- 12.6 Pagination

13. Bootstrap Interactive Components

- 13.1 Collapse
- 13.2 Accordion
- 13.3 Carousel
- 13.4 Modal
- 13.5 Tooltips
- 13.6 Popovers

14. Bootstrap Flex Utilities

- 14.1 Flex Container
- 14.2 Justify Content
- 14.3 Align Items
- 14.4 Flex Direction
- 14.5 Gap Utilities
- 14.6 Responsive Flexbox



15. Bootstrap Responsive Design

- 15.1 Responsive Utilities
- 15.2 Mobile First Layouts
- 15.3 Responsive Visibility
- 15.4 Responsive Typography
- 15.5 Responsive Sections

16. Bootstrap Icons

- 16.1 Bootstrap Icons Setup
- 16.2 Using Icons
- 16.3 Icon Styling
- 16.4 Social Media Icons
- 16.5 Interactive Icons

17. Bootstrap Utility Classes

- 17.1 Display Utilities
- 17.2 Border Utilities
- 17.3 Shadow Utilities
- 17.4 Position Utilities
- 17.5 Overflow Utilities

18. Bootstrap Advanced Layouts

- 18.1 Dashboard Layouts
- 18.2 Admin Panels
- 18.3 Portfolio Layouts
- 18.4 Landing Page Layouts
- 18.5 Business Website Layouts

19. Bootstrap UI/UX Design

- 19.1 Modern Hero Sections
- 19.2 CTA Sections
- 19.3 Testimonial Sections
- 19.4 Team Sections
- 19.5 Feature Sections

20. Bootstrap Projects

- 20.1 Portfolio Website
- 20.2 Business Website
- 20.3 Startup Landing Page
- 20.4 Admin Dashboard
- 20.5 E-Commerce Homepage

21. Bootstrap Optimization & Best Practices

- 21.1 Clean Bootstrap Structure
- 21.2 Reusable Components
- 21.3 Performance Optimization
- 21.4 Responsive Workflow
- 21.5 Industry-Level UI Practices



Advance Level React

1. Introduction to React.js

- 1.1 What is React.js?
- 1.2 Why React is Used in Modern Web Development
- 1.3 React vs Traditional JavaScript
- 1.4 SPA (Single Page Application) Concepts
- 1.5 React Ecosystem Overview
- 1.6 Virtual DOM Concepts
- 1.7 React Development Environment Setup
- 1.8 Vite & Create React App

2. React Fundamentals

3. JSX Concepts

- 3.1 JSX Syntax
- 3.2 Embedding JavaScript in JSX
- 3.3 JSX Rules
- 3.4 Rendering Dynamic Content

4. React Components

- 4.1 Functional Components
- 4.2 Reusable Components
- 4.3 Component Architecture
- 4.4 Clean Folder Structure

5. Class Activity

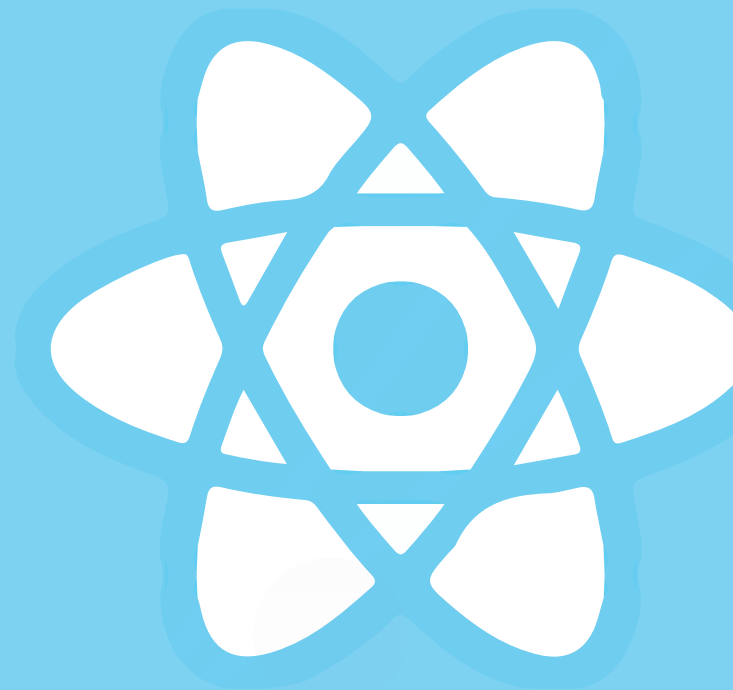
- 5.1 Create reusable UI sections

6. Props & Dynamic Data Handling

- 6.1 Passing Data through Props
- 6.2 Dynamic Components
- 6.3 Props Drilling Concepts
- 6.4 Reusable Cards & Layouts

7. Practical Tasks

- 7.1 Product Card System
- 7.2 Team Member Components



8. State Management in React

8.1 What is State?

8.2 useState Hook

8.3 Dynamic State Updates

8.4 Multiple State Management

8.5 Interactive UI Logic

9. Mini Projects

9.1 Counter App

9.2 Dynamic Theme Switcher

9.3 Interactive FAQ System

10. React Event Handling

10.1 Click Events

10.2 Input Events

10.3 Form Events

10.4 Keyboard Events

10.5 Dynamic Event Management

11. Practical Work

11.1 Interactive Forms

11.2 Dynamic Search Systems

12. React Forms & Validation

12.1 Controlled Components

12.2 Form Validation

12.3 Dynamic Form Handling

12.4 Error Messages

12.5 Authentication UI Logic

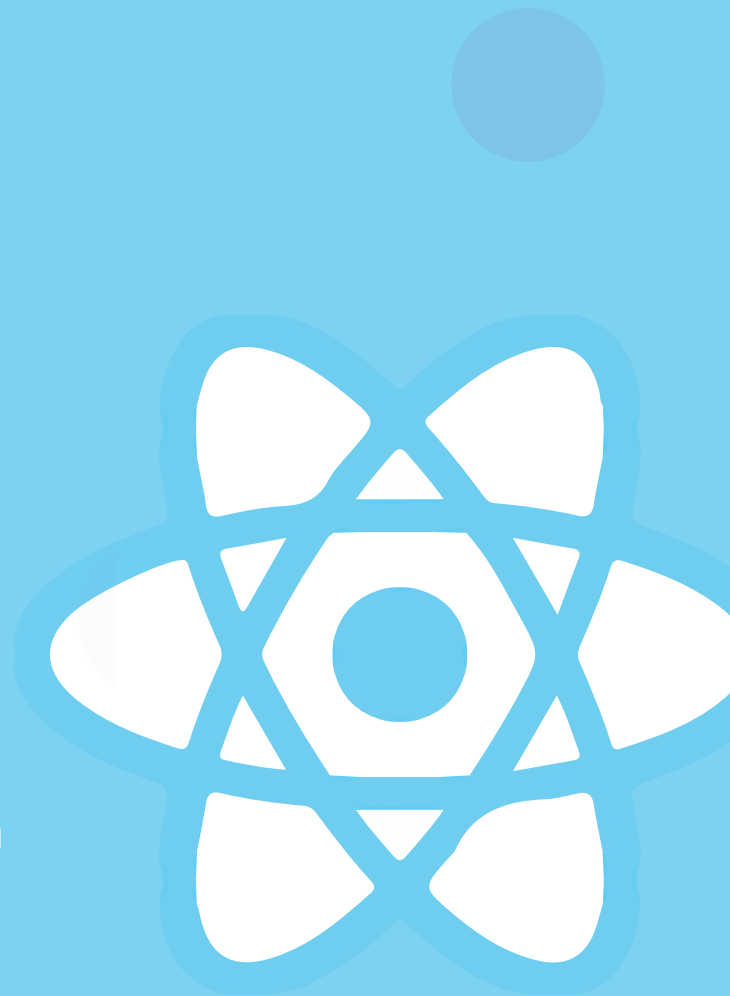
13. Projects

13.1 Login System

13.2 Registration Form

13.3 Contact Form

14. React Hooks Mastery



15. Core Hooks

- 15.1 useState
- 15.2 useEffect
- 15.3 useRef
- 15.4 useContext

16. Advanced Hooks Concepts

- 16.1 Custom Hooks
- 16.2 Hook Reusability
- 16.3 Performance Optimization

17. Practical Activities

- 17.1 Dynamic Data Fetching
- 17.2 Real-Time Updates

18. API Integration in React

19. API Handling

- 19.1 Fetch API in React
- 19.2 Axios Integration
- 19.3 API Error Handling
- 19.4 Loading States
- 19.5 Dynamic Rendering

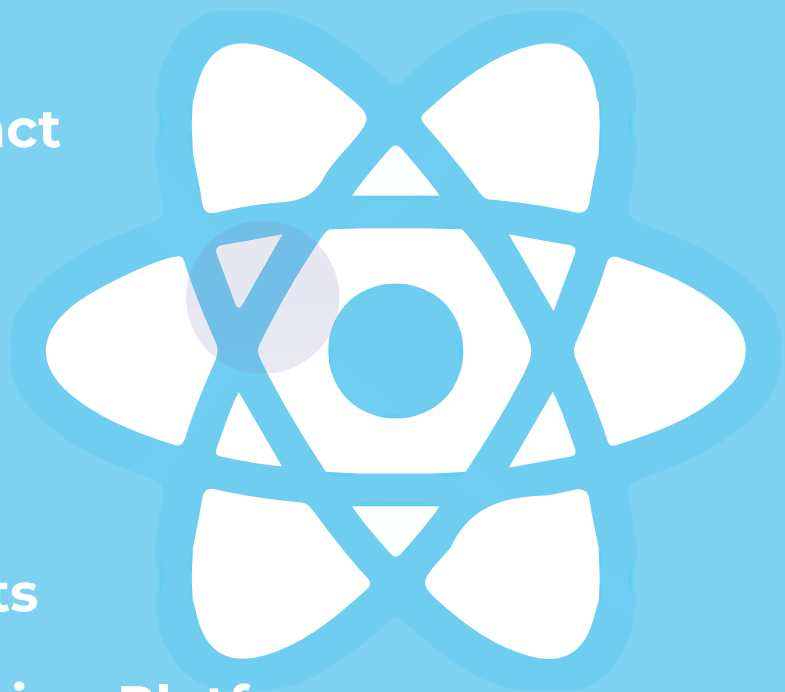
20. Real-Time API Projects

Project 1 — Movie Streaming Platform

- 21.1 Movie Database API
- 21.2 Search Movies
- 21.3 Trending Movies Section
- 21.4 Dynamic Movie Details

Project 2 — Crypto Dashboard

- 22.1 Live Crypto API
- 22.2 Real-Time Price Updates
- 22.3 Dynamic Charts Concepts
- 22.4 Market Tracking UI



Project 3 — Weather Forecast System

23.1 Weather API Integration

23.2 Dynamic Search

23.3 Location-Based Weather

24. React Routing

24.1 React Router DOM

24.2 Dynamic Routing

24.3 Nested Routes

24.4 Route Parameters

24.5 Protected Routes

25. Practical Project

25.1 Multi-Page React Website

26. Context API & Global State Management

26.1 Context API Fundamentals

26.2 Global Data Handling

26.3 Theme Management

26.4 Authentication Context

27. Real-World Practice

27.1 Dark / Light Theme System

27.2 User Authentication Flow

28. Redux Toolkit

28.1 Redux Fundamentals

28.2 Store Management

28.3 Slices

28.4 Actions & Reducers

28.5 Async Thunks

29. Advanced Practice

29.1 E-Commerce Cart System

29.2 Global Product Management

30. Authentication Systems

30.1 JWT Authentication Concepts

30.2 Login & Signup Flow

30.3 Protected Routes



- 30.4 Token Storage
- 30.5 User Session Handling

31. Projects

- 31.1 Authentication Dashboard
- 31.2 Role-Based Access UI

32. Advanced UI Development

- 32.1 Dynamic Dashboards
- 32.2 Admin Panels
- 32.3 Interactive Charts Concepts
- 32.4 Search & Filter Systems
- 32.5 Responsive React Layouts

33. UI Projects

- 33.1 Admin Dashboard
- 33.2 SaaS Dashboard
- 33.3 CRM Interface

34. React Performance Optimization

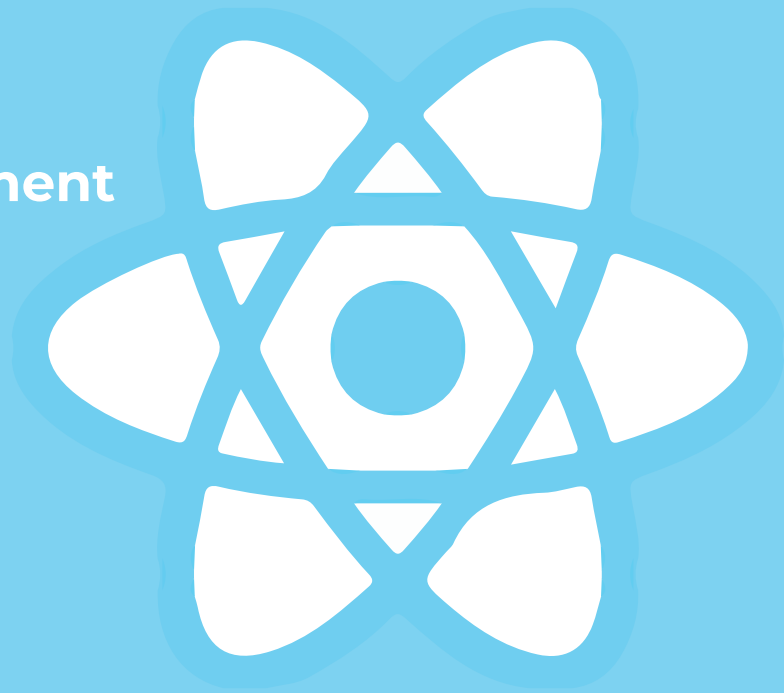
- 34.1 Lazy Loading
- 34.2 Memoization
- 34.3 Code Splitting
- 34.4 Component Optimization
- 34.5 Performance Best Practices

35. React Project Architecture

- 35.1 Professional Folder Structure
- 35.2 Component Reusability
- 35.3 Clean Code Practices
- 35.4 Scalable React Architecture
- 35.5 Production-Level Workflow

36. React + Tailwind CSS Integration

- 36.1 Tailwind Setup with React
- 36.2 Responsive UI Development
- 36.3 Modern Component Design
- 36.4 Animated Interfaces



37. Practical Projects

37.1 Startup Landing Page

37.2 SaaS Website UI

38. Advanced React Projects

Project 1 — AI-Powered Movie Streaming App

39.1 Movie API Integration

39.2 Search System

39.3 Authentication

39.4 Favorites System

39.5 Dynamic Categories

Project 2 — E-Commerce Frontend System

40.1 Product Listings

40.2 Cart System

40.3 Product Filters

40.4 Dynamic Routing

40.5 API Integration

Project 3 — Advanced Admin Dashboard

41.1 Charts Concepts

41.2 Analytics UI

41.3 Authentication System

41.4 Responsive Panels



Advanced Node.js Master Course Outline

1. Introduction to Node.js

- 1.1 What is Node.js?
- 1.2 Why Node.js is Used in Modern Development
- 1.3 JavaScript Runtime Environment
- 1.4 Node.js Architecture
- 1.5 Event-Driven Programming
- 1.6 Non-Blocking I/O Concepts
- 1.7 Installing Node.js & npm
- 1.8 Understanding REPL

2. Node.js Fundamentals

3. Core Concepts

- 3.1 Modules in Node.js
- 3.2 CommonJS Modules
- 3.3 Import & Export System
- 3.4 Global Objects
- 3.5 Process Object
- 3.6 Environment Variables

4. Practical Activities

- 4.1 Create custom modules
- 4.2 Build reusable utility functions

5. File System (FS Module)

- 5.1 Reading Files
- 5.2 Writing Files
- 5.3 Updating Files
- 5.4 Deleting Files
- 5.5 Working with Folders
- 5.6 Asynchronous File Handling

6. Projects

- 6.1 Notes Management System
- 6.2 File Organizer Tool

7. HTTP Module & Server Creation

- 7.1 Creating HTTP Server
- 7.2 Handling Requests & Responses



- 7.3 Serving HTML Pages
- 7.4 Serving JSON Data
- 7.5 Routing Concepts

8. Practical Work

- 8.1 Build custom Node.js server
- 8.2 Dynamic server responses

9. npm & Package Management

- 9.1 Understanding npm
- 9.2 Installing Packages
- 9.3 package.json
- 9.4 Dependencies & Scripts
- 9.5 Nodemon Usage

10. Practical Tasks

- 10.1 Setup project environment
- 10.2 Configure development workflow

11. Event Loop & Asynchronous Programming

- 11.1 Event Loop Concepts
- 11.2 Callbacks
- 11.3 Promises
- 11.4 Async / Await
- 11.5 Error Handling

12. Advanced Practice

- 12.1 Async file operations
- 12.2 API handling simulations

13. Working with APIs in Node.js

- 13.1 Fetching External APIs
- 13.2 REST API Concepts
- 13.3 JSON Data Handling
- 13.4 Real-Time API Requests
- 13.5 API Error Handling

14. API Projects

- 14.1 Weather Data Fetcher
- 14.2 Crypto Price Fetcher



14.3 News API Integration

15. Express.js Fundamentals

16. Introduction to Express.js

16.1 What is Express.js?

16.2 Why Express is Used

17. Express.js Architecture

17.1 Express Server Setup

17.2 Express Routing

17.3 Middleware Concepts

17.4 Creating First Express Application

18. Practical Activities

18.1 Build Express server

18.2 Create routes system

19. Express Routing System

19.1 Basic Routing

19.2 Route Parameters

19.3 Query Parameters

19.4 Dynamic Routes

19.5 Route Organization

19.6 Express Router

20. Practical Activities

20.1 Create Multi-Page API System

20.2 Dynamic Route Handling

21. REST API Development

21.1 REST API Concepts

21.2 GET Requests

21.3 POST Requests

21.4 PUT Requests

21.5 DELETE Requests

21.6 CRUD Operations

21.7 API Structure

21.8 API Testing



22. Practical Projects

22.1 Student Management API

22.2 Product Management API

23. Middleware in Express.js

23.1 Built-in Middleware

23.2 Custom Middleware

23.3 Request Logging

23.4 Authentication Middleware

23.5 Error Middleware

24. Real-World Practice

24.1 Protected route systems

24.2 Request validation

25. Request & Response Handling

25.1 Request Object

25.2 Response Object

25.3 Sending JSON Responses

25.4 Status Codes

25.5 API Response Structure

26. Authentication & Security

26.1 JWT Authentication

26.2 Login Authentication

26.3 Protected APIs

26.4 Authorization Concepts

27. Password Security

27.1 bcrypt Password Hashing

27.2 Secure Authentication Flow

28. Projects

28.1 Login API System

28.2 Secure User Authentication

29. Validation & Security

29.1 Express Validator

29.2 Joi Validation

29.3 CORS Configuration

29.4 Helmet.js Security

v



29.5 Environment Variables
29.6 Secure API Development

30. Security Practice

30.1 API Protection Techniques
30.2 Secure Form Handling

31. Advanced Express.js Concepts

31.1 MVC Architecture
31.2 Controllers & Routes
31.3 Scalable Backend Structure
31.4 Clean Code Practices
31.5 Modular Backend Development

32. Industry-Level Workflow

32.1 Production-Level API Structure
32.2 Professional Folder Organization

33. File Upload Systems

33.1 Multer Middleware
33.2 Single File Upload
33.3 Multiple File Upload
33.4 Image Upload APIs
33.5 Cloudinary Integration Concepts

34. Projects

34.1 Profile Upload API
34.2 Media Management System

35. Real-Time Backend Features

35.1 Introduction to Socket.io
35.2 Live Notifications
35.3 Real-Time Messaging
35.4 Chat Backend Concepts

36. Real-Time Projects

36.1 Chat Application Backend
36.2 Notification System



37. Payment Gateway Integration

37.1 Stripe API Basics

37.2 Checkout APIs

37.3 Payment Workflow Concepts

38. Practice

38.1 Payment Integration Simulation

39. Error Handling & Debugging

39.1 Global Error Middleware

39.2 Try Catch Handling

39.3 API Debugging

39.4 Error Response Management

40. Practical Activities

40.1 API Testing with Postman

40.2 Backend Debugging Workshop

41. API Optimization Techniques

41.1 Pagination

41.2 Filtering

41.3 Searching APIs

41.4 Sorting Systems

41.5 Performance Optimization

42. Advanced Express.js Projects

Project 1 — Authentication Backend API

43.1 User Registration

43.2 Login System

43.3 JWT Authentication

43.4 Protected Routes

43.5 Password Encryption



Advanced MongoDB Master Course Outline

1. Introduction to MongoDB

- 1.1 What is MongoDB?
- 1.2 NoSQL Database Concepts
- 1.3 MongoDB Architecture
- 1.4 Collections & Documents
- 1.5 MongoDB Installation
- 1.6 MongoDB Compass Setup
- 1.7 MongoDB Atlas Cloud Setup

2. MongoDB Fundamentals

- 2.1 Databases & Collections
- 2.2 BSON Data Structure
- 2.3 CRUD Operations
- 2.4 Insert Documents
- 2.5 Update Documents
- 2.6 Delete Documents
- 2.7 Find Queries

3. Practical Activities

- 3.1 Create Database Collections
- 3.2 Manage User Data

4. Advanced Query Operations

- 4.1 Filtering Data
- 4.2 Sorting Data
- 4.3 Limiting Results
- 4.4 Search Queries
- 4.5 Conditional Queries

5. Practice

- 5.1 Product Filtering System
- 5.2 Search Functionality

6. Mongoose Integration

- 6.1 Introduction to Mongoose
- 6.2 Connecting MongoDB with Node.js
- 6.3 Schema Creation
- 6.4 Models in Mongoose
- 6.5 Validation Rules
- 6.6 CRUD with Mongoose



7. Practical Projects

7.1 User Management Database

7.2 Product Database System

8. Database Relationships

8.1 One-to-One Relationships

8.2 One-to-Many Relationships

8.3 Many-to-Many Relationships

8.4 Populate Method

8.5 Referenced Data

9. Real-World Practice

9.1 Orders & Users System

9.2 Blog & Comments System

10. Advanced Schema Design

10.1 Schema Validation

10.2 Default Values

10.3 Enums

10.4 Virtual Fields

10.5 Timestamps

10.6 Middleware in Mongoose

11. Aggregation Pipeline

11.1 Aggregation Basics

11.2 Match & Group

11.3 Project & Sort

11.4 Data Analytics Concepts

12. Practice

12.1 Sales Analytics

12.2 Dashboard Data Systems

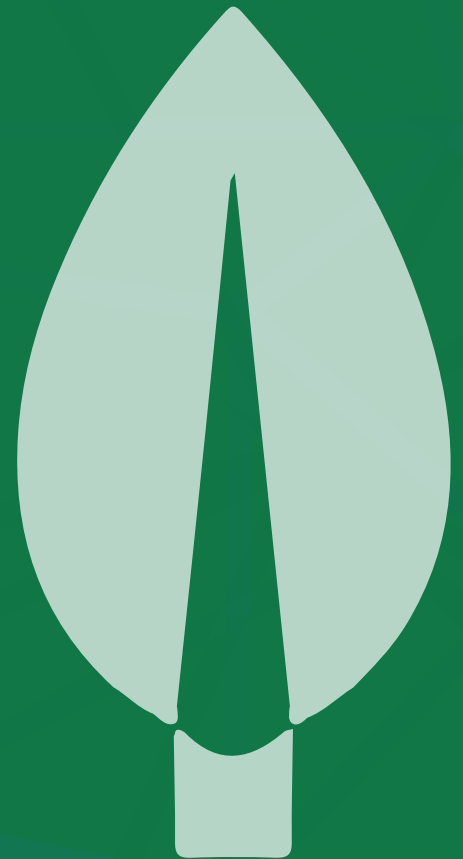
13. MongoDB Security

13.1 Database Authentication

13.2 Secure Database Access

13.3 Environment Variables

13.4 Production Database Setup



14. Cloud Database Management

- 14.1 MongoDB Atlas Deployment
- 14.2 Cloud Database Workflow
- 14.3 Database Monitoring
- 14.4 Backup Concepts

15. Real-World Database Systems

- 15.1 E-Commerce Database Design
- 15.2 LMS Database Structure
- 15.3 Chat Application Database
- 15.4 Admin Dashboard Data System

16. Database Optimization

- 16.1 Indexing Concepts
- 16.2 Query Optimization
- 16.3 Performance Best Practices
- 16.4 Scalable Database Design

17. MongoDB Integration with Node.js

- 17.1 Connecting MongoDB
- 17.2 Mongoose Introduction
- 17.3 Schema Design
- 17.4 CRUD Operations
- 17.5 Relationships Concepts

18. Database Projects

- 18.1 User Management System
- 18.2 Product Database System

19. Advanced Backend Architecture

- 19.1 MVC Architecture
- 19.2 Controllers & Routes
- 19.3 Service Layer Concepts
- 19.4 Clean Backend Structure
- 19.5 Scalable Backend Systems

20. Industry Practice

- 20.1 Production-Level API Structure
- 20.2 Professional Folder Organization



21. Real-Time Backend Concepts

- 21.1 Introduction to Socket.io
- 21.2 Real-Time Communication
- 21.3 Live Notifications
- 21.4 Chat Application Concepts

22. Projects

- 22.1 Real-Time Chat Backend
- 22.2 Notification System

23. File Upload & Cloud Integration

- 23.1 Uploading Files
- 23.2 Multer Middleware
- 23.3 Image Upload Systems
- 23.4 Cloudinary Integration Concepts

24. Practical Projects

- 24.1 Profile Upload System
- 24.2 Media Management Backend

25. Payment Gateway Integration Concepts

- 25.1 Stripe Payment Basics
- 25.2 Secure Transactions
- 25.3 Checkout APIs

26. Practice

- 26.1 Payment API Simulation

27. Error Handling & Debugging

- 27.1 Backend Error Handling
- 27.2 Try Catch
- 27.3 Global Error Middleware
- 27.4 API Debugging Techniques

28. Practical Activities

- 28.1 API Testing & Debugging

29. Backend Deployment

- 29.1 Environment Variables
- 29.2 Deployment Basics

29.3 Render Deployment
29.4 Railway Deployment
29.5 Production Environment Setup

30. Practical Work

30.1 Deploy Backend APIs Online

31. Advanced Node.js Projects

Project 1 — Authentication System API

32.1 User Registration
32.2 Login System
32.3 JWT Authentication
32.4 Protected Routes
32.5 Password Encryption

Project 2 — E-Commerce Backend API

33.1 Product Management
33.2 Cart APIs
33.3 Orders System
33.4 Payment Integration Concepts
33.5 Authentication

Sr Project Category High-Class Projects

1	HTML & CSS Premium Websites	3 Complete Websites: Modern Portfolio Website, Luxury Restaurant Website, Real Estate Business Website
2	Bootstrap Professional Website	1 Complete Website: Corporate Business Website with Responsive Dashboard UI
3	JavaScript Master-Level Applications	5 Complete Projects: Advanced E-Commerce Store, Online Quiz & Exam Portal, Weather & Travel Guide Website, AI-Based Todo Productivity App, Real-Time Currency Converter & Finance Tracker
4	React.js API Integration Projects	2 Advanced Projects: Movie Streaming Web App with API Integration, Live Crypto / Stock Market Dashboard with Real-Time API Data
5	Next.js Professional Projects	SEO-Friendly Business Website, Advanced Portfolio Platform, Modern Startup Landing Page
6	Node.js Backend Applications	File Management System, Backend Server Application, Authentication-Based Backend Project
7	Express.js API Development	REST API Development, Authentication System, Backend CRUD Application
8	MongoDB Database Integration	Database Architecture, CRUD Operations, User & Product Management Systems
9	Final Advanced MERN Stack Projects	2 Production-Level MERN Stack Applications with Authentication, APIs, Payment Integration, Cloud Storage, Admin Dashboard & Live Deployment

Final Advanced MERN Stack Project Options

AI-Powered Learning Management System (LMS)
Advanced E-Commerce Marketplace
Real-Time Chat & Video Calling Application
Freelancing & Service Booking Platform
Hospital / Clinic Management System
AI Content Generator Platform
Real Estate Property Management Platform

THANK YOU!