

# HackForge

## MERN Stack Certification Training

Mongodb Expressjs React Nodejs

60 Hours

Live Online in English | Andhra Pradesh | Telangana | Tamil Nadu | Karnataka

Phone :9884789292

### Module 1: Web Development Fundamentals

Foundation before React & Node

- How the Web Works
  - Client vs Server Architecture
  - HTTP / HTTPS Basics
  - REST API Concepts
  - JSON Structure
  - Developer Tools (Chrome DevTools)
  - VS Code Setup & Extensions
  - Git & GitHub Basics
- 

### Module 2: **HTML5 & CSS3**

Frontend layout fundamentals

#### HTML5

- Semantic Tags
- Forms & Input Types
- Tables
- Media Elements

#### CSS3

- Selectors

- Box Model
  - Flexbox
  - Grid Layout
  - Responsive Design
  - Media Queries
- 

## Module 3: JavaScript (Core + ES6+)

Most important MERN foundation

### JavaScript Basics

- Variables
- Data Types
- Operators
- Functions
- Arrays & Objects
- Loops & Conditions

### Advanced JavaScript

- Closures
- Hoisting
- Callbacks
- Promises
- Async / Await
- Event Loop

### ES6+ Features

- Arrow Functions
  - Destructuring
  - Spread & Rest Operators
  - Modules (import/export)
  - Template Literals
- 

## React.js

### Introduction to React

- What is React?
- Why React is used in MERN stack

- SPA vs Traditional websites
  - Virtual DOM concept
  - Setting up React project (Vite / Create React App)
- 

## JSX Fundamentals

- What is JSX?
  - Embedding expressions in JSX
  - Attributes & styling in JSX
  - Rendering elements dynamically
- 

## Components in React

- Functional components
  - Reusable components
  - Component structure best practices
  - Splitting UI into components
- 

## Props in React

- Passing data using props
  - Props vs variables
  - Default props
  - Props drilling basics
- 

## State Management (Core Concept)

- What is state?
  - `useState` hook
  - Updating UI dynamically
  - Handling user interactions with state
- 

## Event Handling

- Click events
- Form events
- Keyboard events
- Passing arguments to handlers

---

## Conditional Rendering

- if / else rendering
  - Ternary operators
  - Logical AND rendering
  - Rendering based on authentication state
- 

## Lists & Keys

- Rendering arrays
  - Using `.map()` in React
  - Importance of keys
  - Dynamic UI rendering
- 

## Forms Handling

- Controlled components
  - Handling multiple inputs
  - Form validation basics
  - Submit handlers
- 

## React Hooks (Essential)

- `useState`
  - `useEffect`
  - Lifecycle using hooks
  - API calls with `useEffect`
- 

## Routing in React

- Installing React Router
  - Creating routes
  - Route parameters
  - Nested routes
  - Protected routes (login-based access)
-

## API Integration

- Connecting React with Express backend
  - Fetch vs Axios
  - GET / POST requests
  - Loading states
  - Error handling
- 

## Global State Management (Intermediate)

- Context API introduction
  - Creating global store
  - Sharing data across components
- 

## Authentication Flow (MERN Integration)

- Login UI
  - Signup UI
  - Storing JWT token
  - Protected dashboard
  - Logout functionality
- 

## Project Folder Structure (Industry Pattern)

- Components folder
  - Pages folder
  - Services (API layer)
  - Utils/helpers
  - Environment configs
- 

## Performance Optimization Basics

- Avoid unnecessary re-renders
  - `useMemo` (intro)
  - `useCallback` (intro)
  - Lazy loading components
- 

## Deployment Basics

- Build React app for production
  - Environment variables
  - Deploy on Netlify / Vercel
  - Connect with backend server
- 

## React Mini Project

- Login / Signup screens
  - Dashboard page
  - Add / Edit / Delete tasks
  - API integration with Express
  - Protected routes
  - Full-stack Task Manager UI
- 

## Module 7: Node.js Fundamentals

Backend runtime environment

- What is Node.js
- Event-driven architecture
- Node Modules
- npm / package.json
- File System Module
- Environment Variables (.env)

## Express.js

### Introduction to Express.js

- What is Express.js?
  - Role of Express in MERN architecture
  - Installing Node.js & Express
  - Creating first Express server
  - Understanding request–response lifecycle
- 

### Express Project Setup

- Folder structure best practices
  - Creating `package.json`
  - Installing dependencies (`express`, `nodemon`)
  - Using environment variables (`dotenv`)
  - Running development server
- 

## Routing in Express

- Basic routing (GET, POST, PUT, DELETE)
  - Route parameters
  - Query parameters
  - Organizing routes using Router
  - Modular route structure
- 

## Middleware in Express

- What is middleware?
  - Built-in middleware (`express.json()`)
  - Custom middleware creation
  - Third-party middleware usage
  - Request logging middleware
  - Error-handling middleware
- 

## REST API Development

- REST architecture basics
  - Creating CRUD APIs
  - Sending JSON responses
  - Handling request body data
  - Status codes best practices
- 

## MongoDB Integration (with Mongoose)

- Connecting Express to MongoDB
  - Installing & configuring Mongoose
  - Creating schemas & models
  - CRUD operations with MongoDB
  - Data validation using schema
-

## MVC Architecture

- Controller layer
  - Service layer (optional advanced pattern)
  - Model layer
  - Route separation
  - Clean folder structure implementation
- 

## Authentication & Authorization

- Password hashing using bcrypt
  - Login & signup APIs
  - JWT authentication
  - Protected routes
  - Role-based access control (optional advanced)
- 

## File Upload Handling

- Uploading files using multer
  - Storing files locally
  - Image upload API creation
- 

## Error Handling & Validation

- Centralized error handler
  - Try-catch best practices
  - Request validation using middleware
  - Handling invalid routes
- 

## Connecting Express with React (Frontend Integration)

- CORS setup
  - API communication workflow
  - Fetch / Axios integration
  - Structuring full-stack apps
- 

## Deployment Basics

- Preparing Express app for production

- Environment configs
  - Hosting backend (Render / Railway / VPS)
  - Connecting frontend + backend in production
- 

## Mini Project

Build a **User Management REST API**

- Signup/Login
  - JWT authentication
  - CRUD operations
  - Protected dashboard route
  - MongoDB integration
- 

## Module 9: MongoDB Database Integration

NoSQL database layer

### MongoDB Basics

- Documents & Collections
- CRUD Operations
- MongoDB Compass

### Mongoose ORM

- Schema Design
  - Models
  - Validation
  - Relationships (refs & population)
- 

## Module 10: Authentication & Authorization (JWT)

Secure application login system

- Password Hashing (bcrypt)
- JWT Authentication
- Token Generation
- Token Verification

- Middleware Protection
  - Role-Based Access Control
- 

## Module 11: Full Stack Integration Project

Frontend + Backend connection

- Connecting React with Express APIs
  - Axios Integration
  - Protected Routes
  - Authentication Flow
  - Error Handling Strategy
- 

## Module 12: File Upload & Advanced Features

Production-ready functionality

- File Upload (Multer)
  - Image Storage
  - Pagination
  - Search & Filtering
  - API Optimization
- 

## Module 13: Testing Basics (Optional but Valuable)

Improving application reliability

- API Testing using Postman
  - Debugging Techniques
  - Error Logging Strategy
- 

## Module 14: Deployment & DevOps Basics

Making apps live

### Frontend Deployment

- Vercel / Netlify deployment

## Backend Deployment

- Render / Railway deployment

## Database Deployment

- MongoDB Atlas setup

## Environment Configuration

- Production variables setup
- 

## Module 15: Docker (Optional Advanced Module)

Containerized deployment basics

- Docker Introduction
- Dockerfile Creation
- Image vs Container
- Dockerizing MERN App
- Docker Compose (overview)

## **Social Media Links:**

<https://www.linkedin.com/company/hackforge-academy>

[https://www.instagram.com/hackforge\\_english/](https://www.instagram.com/hackforge_english/)

[https://www.youtube.com/@hackforge\\_code](https://www.youtube.com/@hackforge_code)

**Daily coding whatsapp community in java,React,JavaScript,Python  
+919884789292**