Practical Full Stack Development with MERN Stack and BEAST AI

Duration: 5 days

Day 1: MERN Stack Foundation and React Front-End Development

Module 1: MERN Stack Architecture and Project Setup

Topics

- Understanding MERN Stack architecture
- Role of MongoDB, Express.js, React.js, and Node.js
- Front-end, back-end, database, and API communication flow
- Setting up development environment
- Node.js, npm, VS Code, Git basics
- Creating a full-stack project structure
- Basic system design view of a MERN application
- Overview of BEAST AI for AI-assisted full-stack development
- Using BEAST AI to generate project ideas, user stories, and starter prompts

Lab

Lab 1: MERN Project Setup

- Create project folders for client and server
- Initialize React application
- Initialize Node.js backend
- Configure basic project structure
- Use BEAST AI to create a MERN project plan and prompt checklist
- Review AI-generated structure and adjust it for the training project

---

Module 2: React.js Practical Development

Topics

- React components and JSX
- Functional components
- Props and state
- Event handling

- Conditional rendering
- Lists and keys
- Forms and controlled components
- Basic styling and responsive UI design

Lab

Lab 2: Build Front-End UI

- Create pages for Home, Login, Register, Dashboard
- Create reusable components such as Navbar, Button, Card, Form
- Implement form handling using React state

Module 3: React Routing and API Integration Basics

Topics

- React Router setup
- Page navigation
- Creating layout-based routing
- Fetching data using fetch or Axios
- Handling loading and error states
- Basic front-end validation

Lab

Lab 3: Build React Front-End Flow

- Configure routing between pages
- Create registration and login forms
- Prepare front-end structure for backend API integration

Day 2: Node.js, Express.js, REST APIs, and MongoDB

Module 1: Node.js and Express.js Backend Development

Topics

- Introduction to Node.js runtime
- Express.js application setup
- Middleware concept
- Request and response handling

- Creating REST API routes
- HTTP methods: GET, POST, PUT, DELETE
- API testing using Postman
- Using BEAST AI prompts to design REST endpoints and sample JSON payloads
- Validating AI-generated backend code before implementation

Lab

Lab 1: Create Express REST API

- Create Express server
- Build routes for users and products/tasks
- Test APIs using Postman
- Use BEAST AI to draft Express route logic and compare it with manual implementation

---

Module 2: MongoDB and Mongoose Integration

Topics

- MongoDB database concepts
- Collections and documents
- Connecting MongoDB with Node.js
- Mongoose schema and models
- CRUD operations using Mongoose
- Basic data validation in schema

Lab

Lab 2: MongoDB CRUD Implementation

- Connect backend with MongoDB
- Create User and Task/Product models
- Implement create, read, update, delete APIs

Module 3: Full-Stack Front-End and Back-End Connection

Topics

- Calling backend APIs from React
- Handling API responses
- Displaying MongoDB data in React

- Creating and updating records from React UI
- Error handling in full-stack applications

Lab

Lab 3: Connect React with Express APIs

- Submit form data from React to backend
- Store data in MongoDB
- Display records from database on React dashboard

Day 3: Authentication, TypeScript Basics, SQL, and DSA Practical

Module 1: Authentication and Authorization

Topics

- User registration and login flow
- Password hashing using bcrypt
- JSON Web Token concept
- Token-based authentication
- Protected routes
- Role-based access basics
- Secure API design practices

Lab

Lab 1: Implement Login and Authentication

- Create register API
- Create login API
- Hash user password
- Generate JWT token
- Protect dashboard API using middleware

Module 2: JavaScript, TypeScript, and Code Quality

Topics

- JavaScript ES6+ practical concepts
- Array methods: map, filter, reduce, find
- Async and await

- Error handling
- Introduction to TypeScript
- Type annotations
- Interfaces and types
- Using TypeScript in React or Node.js basics
- BEAST AI-assisted refactoring, debugging, and TypeScript migration prompts
- Writing safe prompts for code explanation, error fixing, and code quality review

Lab

Lab 2: Refactor Code with TypeScript Concepts

- Add types/interfaces for user and task/product data
- Apply async/await in API calls
- Improve error handling in backend APIs
- Use BEAST AI to review async/await APIs, suggest TypeScript interfaces, and document changes

---

Module 3: SQL and DSA Practical Exposure

Topics

- SQL database concepts
- Tables, rows, columns
- SELECT, INSERT, UPDATE, DELETE
- Joins overview
- When to use SQL vs NoSQL
- Practical DSA for development
- Arrays, objects, searching, sorting basics
- Problem-solving in API and UI logic

Lab

Lab 3: SQL and DSA Practice

- Write basic SQL queries
- Compare MongoDB document design with SQL table design
- Solve small JavaScript problems using arrays and objects
- Apply search/filter logic in React UI

Day 4: System Design Basics, Docker, CI/CD Overview, and Cloud Deployment

Module 1: System Design Fundamentals for MERN Applications

Topics

- Client-server architecture
- API design principles
- Database design basics
- Authentication flow design
- Folder structure best practices
- Environment variables
- Logging and error handling
- Scalability basics
- Code review best practices
- Using BEAST AI for architecture review, folder-structure recommendations, and security checklist creation

Lab

Lab 1: Improve Application Architecture

- Add environment configuration
- Organize routes, controllers, models, and middleware
- Add centralized error handling
- Review and improve existing code structure
- Generate an AI-assisted architecture review checklist using BEAST AI

Module 2: Docker for MERN Applications

Topics

- Introduction to containers
- Docker images and containers
- Dockerfile basics
- Docker Compose overview
- Containerizing React application
- Containerizing Node.js backend

- MongoDB container overview

Lab

Lab 2: Containerize MERN Application

- Create Dockerfile for backend
- Create Dockerfile for frontend
- Use Docker Compose for frontend, backend, and MongoDB
- Run full-stack app using Docker

Module 3: CI/CD and Cloud Deployment Overview

Topics

- CI/CD pipeline concept
- Git and GitHub workflow
- Build and deployment process
- Environment-based deployment
- Overview of AWS, Azure, and GCP deployment options
- Basic cloud deployment architecture
- Deployment best practices

Lab

Lab 3: Deploy Full-Stack Application

- Push project to GitHub
- Prepare production build
- Deploy frontend on a hosting platform
- Deploy backend on a cloud/server platform
- Configure environment variables

Day 5: Capstone Project, Kubernetes Overview, Testing, and Final Review

Module 1: Capstone Project Development

Topics

- Requirement understanding
- Project planning
- UI design

- API design
- Database design
- Authentication planning
- Practical implementation approach
- BEAST AI-assisted feature planning, acceptance criteria, and backlog creation

Capstone Project Options

Students can build one of the following:

1. **Task Management Application**
2. **Student Management System**
3. **Job Portal Mini Application**
4. **Inventory Management System**
5. **Blog Management Platform**

Lab

Lab 1: Build Capstone Application

- Create React UI
- Create backend APIs
- Connect MongoDB
- Implement authentication
- Add CRUD functionality
- Use BEAST AI to generate test cases and improvement ideas for the capstone project

---

Module 2: Testing, Debugging, and Code Review

Topics

- Front-end debugging
- Backend debugging
- API testing with Postman
- Basic unit testing overview
- Testing REST APIs
- Common MERN project errors
- Code review checklist
- Security and validation checklist

Lab

Lab 2: Test and Debug Capstone Project

- Test all APIs using Postman
- Validate forms
- Fix common bugs
- Review code quality
- Perform peer code review

Module 3: Kubernetes Overview and Final Deployment Review

Topics

- Why Kubernetes is used
- Pods, services, and deployments overview
- Kubernetes vs Docker
- Basic deployment flow
- Cloud deployment recap
- Project presentation guidelines
- Interview preparation points based on MERN Stack

Lab

Lab 3: Final Project Presentation

- Demonstrate working application
- Explain architecture
- Show database design
- Explain API flow
- Explain deployment approach
- Submit GitHub repository