



★ MERN Stack – Complete Syllabus

■ **Module 1: Introduction to MERN Stack**

- What is MERN Stack?
- Evolution from MERN (2018) → Modern MERN
- Why MERN is in high demand
- Full-Stack development architecture
- Client–Server model
- Understanding REST API & JSON
- Setting up VS Code, Node.js, MongoDB, Git
- Installing pnpm / npm / yarn

■ **Module 2: HTML, CSS, JavaScript (Fast-Track Bootcamp)**

- HTML5 Semantic Elements
- CSS3, Flexbox, Grid
- Media Queries & Responsive Design
- JavaScript ES6+
 - let, const, arrow functions, promises, async/await
 - DOM Manipulation
 - Fetch API
- Modern JS Features
- Hands-on mini projects

■ **Module 3: TypeScript for MERN Developers**

- Why TypeScript in MERN
- Types, Interfaces, Enums
- Generics & Utility Types
- TypeScript with React
- TypeScript with Node.js & Express
- TypeScript with MongoDB (Mongoose)
- Building type-safe APIs
- TypeScript + Axios

■ **Module 4: React.js****React Fundamentals**

- CRA vs Vite vs Next.js
- JSX
- Components (Functional)
- Props & State
- Event handling

React Hooks (Complete)

- useState
- useEffect
- useRef

- useMemo / useCallback
- useReducer
- useContext

Forms

- Controlled & uncontrolled forms
- Formik & React Hook Form
- Yup Validation

UI & Styling

- CSS Modules
- Styled Components
- Tailwind CSS
- Material UI / ShadCN UI

Performance Optimization

- Memoization
- Code splitting & Lazy loading
- React Suspense

React 19 New Updates

- React Compiler
- Server Components
- Actions

■ **Module 5: Next.js 16****App Router**

- Layouts
- Server Components & Client Components
- Partial Prerendering (PPR – 2025 Feature)

Routing

- Dynamic routes
- Parallel routes
- Intercepting routes
- Route Handlers (API routes)

Data Fetching

- fetch() in Server Components
- Revalidation
- Caching
- Streaming

Next.js Server Actions

- Server-based forms
- File uploads
- Secure mutations

Authentication

- NextAuth.js v5
- JWT & OAuth

SEO

- Metadata API
- Sitemap / robots

Deployment

- Deploy on Vercel

■ **Module 6: State Management (2025 Standards)**

- Context API + useReducer
- Redux Toolkit (RTK)
- RTK Query (API caching)
- Zustand (Modern, lightweight store)
- React Query (TanStack Query)
 - Infinite queries
 - Optimistic updates
 - Background refetching

■ **Module 7: Node.js – Backend Development**

- Node.js architecture
- Event Loop
- Async/Await deeply
- Modules (ESM)
- Core modules: fs, path, http, crypto

■ **Module 8: Express.js – Building REST APIs**

- Express routing
- Middleware (custom + third-party)
- Error-handling middleware
- File uploads (Multer)
- Pagination, sorting, filtering
- Validation with Zod / Joi
- Role-based routes
- Production folder structure

■ **Module 9: MongoDB + Mongoose (Database Layer)**

MongoDB Basics

- CRUD operations
- Aggregation Pipeline
- Indexing for performance

Mongoose (with TypeScript)

- Schemas, Models, Validation
- Relationships (ref, populate)
- Transactions
- Aggregations
- Mongo Atlas Cloud

■ **Module 10: Connecting React/Next.js with Node.js**

- CORS
- Proxying requests
- API versioning

- Secure API consumption

■ **Module 11: Authentication & Security (Enterprise Level)**

- JWT Auth
- OAuth2 (Google, Facebook, GitHub)
- Refresh Token Rotation
- HttpOnly Cookies vs localStorage
- Protecting routes
- Rate limiting
- Helmet
- CSRF & XSS prevention
- Password Hashing with bcrypt
- Encryption with crypto

■ **Module 12: Redis – Caching & Performance**

- Installing Redis (Optional)
- Caching database queries
- Redis for session storage
- Rate limiting using Redis
- Pub/Sub for real-time updates

■ **Module 13: Real-Time Apps – WebSockets**

- WebSockets vs SSE vs Long Polling
- Socket.io
- Real-time chat app
- Live notifications
- Real-time dashboard

■ **Module 14: Microservices (Advanced MERN)**

- Monolith vs Microservices
- Domain-driven design
- Message queues:
 - RabbitMQ (Optional)
 - Kafka (Optional)
- API Gateway
- Authentication for microservices
- Event-driven architecture

■ **Module 15: Background Jobs & Queues**

- BullMQ with Redis (Optional)
- Worker threads
- Cron jobs
- Email sending & scheduling

■ **Module 16: File Uploads & Media Management**

- Cloudinary (Optional)

- AWS S3 (Optional)
- UploadThing
- Image optimization

■ **Module 17: Testing (Full Stack Testing)**

- Jest (Optional)
- React Testing Library (Optional)
- Supertest for APIs
- Playwright/Cypress for E2E
- Mocking database calls

■ **Module 18: DevOps & Deployment**

- Docker & Docker Compose (Frontend + Backend + DB) (Optional)
- CI/CD using GitHub Actions (Optional)
- Deploying Frontend on: (Optional)
 - Vercel
 - Netlify
- Deploying Backend on: (Optional)
 - Render
 - Fly.io
 - AWS EC2
- Deploying MongoDB on: (Optional)
 - Mongo Atlas
 - Docker Mongo instance
- PM2 & Load Balancing (Optional)
- CDN basics (Optional)

■ **Module 19: Performance & Optimization**

- DB indexing
- Caching layers
- API optimization
- Redis performance
- React memoization
- Bundle analysis
- Next.js performance checklist

■ **Module 20: Capstone Project (Enterprise MERN Project)**

A full-scale production-ready MERN application, including:

Features:

- Authentication + Authorization
- Role-based dashboard
- REST + WebSockets
- Payment Integration (Stripe/Razorpay)
- File uploads

- Cloud deployment
- Unit + Integration Tests
- CI/CD pipeline

Example Projects:

- E-Commerce Platform
 - Social Media App
 - Real-Time Chat + Notification App
 - Job Portal
 - LMS (Learning Management System)
 - SaaS Dashboard
-
-