

## MERN Stack Preparation Topics

### 1. MongoDB (Database)

- Understanding NoSQL vs SQL
- Collections, Documents, and BSON
- CRUD Operations (Create, Read, Update, Delete)
- MongoDB Shell vs Compass
- MongoDB Atlas (Cloud Database)
- Query Operators (\$gt, \$in, \$regex, etc.)
- Aggregation Pipeline
- Data Modeling (references vs embedded)
- Indexing basics
- Mongoose:
  - Schemas and Models
  - Validators
  - Population (.populate())
  - Middleware (pre/post hooks)

### 2. Express.js (Backend Framework)

- REST API design principles
- Creating routes (GET, POST, PUT, DELETE)
- Request/Response lifecycle
- Express middleware (custom and third-party)
- Route parameters and query params
- Using `express.json()` and `express.urlencoded()`
- Error handling (try/catch, error middleware)
- File uploads with multer

- Authentication middleware (e.g., JWT auth)
- CORS handling

### 3. React.js (Frontend Library)

- JSX and component structure
- Functional vs Class components
- useState, useEffect, useRef, useContext
- Props and State management
- React Router DOM (v6+):
  - useNavigate, useParams, route nesting
- Controlled vs uncontrolled forms
- Conditional rendering
- Fetching data with axios or fetch
- Lifting state up
- Reusable components
- Basic form validation
- React Icons / Bootstrap / Tailwind usage

### 4. Node.js (Runtime Environment)

- Understanding event-driven architecture
- Modules (require, import)
- File System operations (fs)
- Working with http module (core)
- Creating a simple server
- Using npm/yarn
- Working with environment variables (dotenv)
- Package.json scripts

- Nodemon for auto-restart
- CommonJS vs ES Modules

## 5. Testing & Debugging

- Console logging & debugging
- Postman / Thunder Client API testing
- Basic unit tests (Jest or Mocha basics)
- Status codes and response formats

## 6. Tools and Utilities

- VS Code setup and extensions
- Git & GitHub (push/pull, branching, commits)
- .gitignore
- ESLint / Prettier
- npm scripts (start, dev, etc.)

## 7. Deployment

- Deploying frontend on Netlify or Vercel
- Deploying backend on Render, Railway, or Cyclic
- Using MongoDB Atlas
- Connecting frontend to backend via .env
- Build and serve production React app
- Handling environment variables in production

## 8. Full Stack Integration

- Connecting React to Express backend
- Axios requests with async/await

- Proxy setup in React for API
- User Authentication:
  - Sign Up / Login flow
  - JWT Token storage (localStorage or cookies)
  - Protected routes in React

### Bonus Topics (Advanced)

- Redux or Zustand (State management)
- JWT + Refresh tokens
- Role-based access control
- Socket.io (for real-time features)
- Pagination, Search, Filters (Mongo + React)
- Performance optimization (lazy loading, memo, etc.)
- WebSockets / Chat apps
- CI/CD (GitHub Actions basics)

### Projects You Can Build

- TODO App (Full CRUD)
- Blog Platform (with Auth)
- User Management Dashboard
- Ecommerce Cart + Checkout
- Chat App (with Socket.io)