

I'm doing great—and yep, I'm **READY** for this heavy task. This is actually a *perfect* way to start. Respect. ☒
You asked for **ONLY TOPIC NAMES**, very detailed, nothing missed, job-ready, company-level, latest till 2026, separated by MongoDB / Express / React / Node, plus real-world working topics.
I got you. This is going to be long — don't panic, this is your full career map ☒☒

☒ BEFORE MERN (VERY IMPORTANT – FOUNDATION)

(Companies expect this even if you're beginner)

Computer & Web Basics

- What is a Computer
- Hardware vs Software
- Operating System Basics
- How Internet Works
- Client vs Server
- What is a Website
- What is a Web Application
- Static vs Dynamic Websites
- Browser Internals
- How Browsers Render Pages
- HTTP vs HTTPS
- Request & Response Cycle
- DNS
- IP Address
- Hosting Basics
- Domain Basics
- Web Servers
- CDN Basics

Programming Basics (Language Independent)

- What is Programming

- Algorithm Basics
 - Flowcharts
 - Pseudocode
 - Variables
 - Data Types
 - Operators
 - Conditional Statements
 - Loops
 - Functions
 - Scope
 - Error Handling Basics
 - Debugging Basics
 - Time & Space Complexity (Basic)
-

☒ WEB CORE (MANDATORY FOR MERN)

HTML (Complete)

- HTML Structure
 - Tags & Attributes
 - Head vs Body
 - Meta Tags
 - Text Formatting
 - Lists
 - Tables
 - Forms
 - Input Types
 - Semantic HTML
 - Accessibility (ARIA Basics)
 - SEO Friendly HTML
 - HTML5 APIs
 - Best Practices
-

CSS (Complete)

- CSS Syntax
 - Selectors
 - Box Model
 - Colors & Units
 - Display Properties
 - Positioning
 - Flexbox
 - Grid
 - Responsive Design
 - Media Queries
 - Animations
 - Transitions
 - Pseudo Classes & Elements
 - CSS Variables
 - BEM Methodology
 - Modern CSS Practices
 - Tailwind CSS (Company Level)
 - CSS Optimization
-

JavaScript (🔑 MOST IMPORTANT)

Core JavaScript

- JavaScript Basics
- Variables (var, let, const)
- Data Types
- Operators
- Conditionals
- Loops
- Functions
- Arrow Functions

- Scope & Hoisting
- Closures
- Callbacks
- Promises
- Async / Await
- Error Handling
- Event Loop
- Call Stack
- Microtasks & Macrotasks

Advanced JavaScript

- Objects
- Prototypes
- Classes
- Inheritance
- Modules
- Destructuring
- Spread & Rest
- Array Methods
- Functional Programming
- Immutability
- Deep vs Shallow Copy
- Memory Management
- Garbage Collection
- Design Patterns (Basics)

Browser JavaScript

- DOM Manipulation
- Events
- Event Bubbling & Capturing
- Local Storage
- Session Storage
- Cookies

- Fetch API
 - Web APIs
-

☒ REACT JS (FRONTEND – COMPANY LEVEL)

React Fundamentals

- Why React
- SPA Concept
- React Architecture
- JSX
- Components
- Functional Components
- Props
- State
- Event Handling
- Conditional Rendering
- Lists & Keys

React Hooks

- useState
- useEffect
- useContext
- useRef
- useMemo
- useCallback
- Custom Hooks

React Advanced

- Component Lifecycle
- Controlled vs Uncontrolled Components
- Lifting State Up
- Props Drilling

- Context API
- Error Boundaries
- React Fragments
- React.memo
- Lazy Loading
- Suspense
- Code Splitting

React Routing

- React Router Basics
- Dynamic Routing
- Nested Routes
- Protected Routes
- Route Guards
- URL Params
- Query Params

State Management

- Redux Basics
- Redux Toolkit
- Store
- Reducers
- Actions
- Middleware
- Async Thunks
- RTK Query
- Zustand (Modern)
- Context vs Redux

API Handling

- REST APIs
- Axios
- Fetch

- Error Handling
- Loading States
- Pagination
- Infinite Scroll
- Debouncing
- Throttling

Forms & Validation

- Controlled Forms
- Form Libraries (Formik / React Hook Form)
- Validation
- File Uploads

UI & Styling

- CSS Modules
- Tailwind CSS
- Material UI
- Shadcn UI
- Responsive UI
- Dark Mode

Performance & Optimization

- Reconciliation
- Virtual DOM
- Memoization
- Performance Profiling

Testing

- Unit Testing
 - Jest
 - React Testing Library
 - Component Testing
-

☒ **NODE JS (BACKEND CORE)**

Node Fundamentals

- What is Node.js
- V8 Engine
- Event Driven Architecture
- Non-Blocking I/O
- Node Runtime
- npm & yarn
- package.json

Core Modules

- fs
- path
- http
- os
- events
- streams
- buffers

Backend Concepts

- REST API Design
 - MVC Architecture
 - Environment Variables
 - Config Management
 - Logging
 - Error Handling
 - Validation
-

☒ **EXPRESS JS (BACKEND FRAMEWORK)**

Express Basics

- Express Setup
- Routing
- Request & Response
- Middleware
- Custom Middleware
- Error Middleware

Advanced Express

- RESTful APIs
 - CRUD Operations
 - Pagination
 - Filtering
 - Sorting
 - File Uploads (Multer)
 - Authentication Flow
 - Authorization
 - JWT
 - Cookies & Sessions
 - Rate Limiting
 - Security Headers
 - CORS
-

☒ MONGODB (DATABASE)

MongoDB Basics

- What is NoSQL
- MongoDB Architecture
- BSON
- Collections & Documents

CRUD Operations

- Insert
- Find
- Update
- Delete

Advanced MongoDB

- Indexes
- Aggregation Pipeline
- Lookup
- Transactions
- Schema Design
- Data Modeling
- Performance Optimization

Mongoose

- Schema
- Models
- Validation
- Middleware (Hooks)
- Population
- Virtuals

☒ AUTHENTICATION & SECURITY (VERY IMPORTANT)

- Authentication vs Authorization
- JWT
- Refresh Tokens
- OAuth
- Google Login

- Role Based Access
 - Password Hashing
 - CSRF
 - XSS
 - SQL / NoSQL Injection
 - Security Best Practices
-

☒ DEVOPS & DEPLOYMENT

- Git & GitHub
 - Branching Strategy
 - CI/CD Basics
 - Environment Setup
 - Build Process
 - Vercel
 - Netlify
 - Render
 - AWS Basics
 - Docker Basics
 - Nginx
 - Reverse Proxy
 - HTTPS & SSL
-

☒ SYSTEM DESIGN (ENTRY LEVEL)

- Monolith vs Microservices
 - API Design
 - Database Scaling Basics
 - Caching (Redis)
 - Load Balancing
 - Basic Architecture Diagrams
-

☒ TESTING & QUALITY

- Unit Testing
 - Integration Testing
 - API Testing (Postman)
 - Debugging Production Issues
 - Logging & Monitoring
-

☒ COMPANY & JOB READY SKILLS

- Real-World Project Structure
 - Clean Code
 - Code Reviews
 - Agile & Scrum
 - Jira Basics
 - Documentation
 - API Documentation (Swagger)
 - Resume Projects
 - Interview Questions
 - Coding Rounds Preparation
 - System Design Interviews
 - Open Source Contribution
-

☒ LATEST & FUTURE-READY (2025–2026)

- Server Components
- Next.js App Router
- Edge Functions
- AI APIs Integration
- WebSockets
- Real-Time Apps
- Micro-Frontends

- GraphQL
- tRPC
- Modern Auth Systems
- Performance-First Web Apps