



# MERN Backend Developer - Pro Track



*Brought to you by*

**STEM Link**

**Dive in for a step-by-step guide to all of the concepts you need to know to become a Well-paid Backend Developer**

**We gathered this Roadmap from 2 Software Engineers building with MERN Stack, already earning above 400K monthly. So this should work for you too**

## Side notes

For some concepts, we have added more than 1 tool you can consider using. Ex: Global State Management we have suggested both Zustand and Redux, and you can feel free to use any of them based on the requirement

This is a cut-the-bullshit react developer tool & concept list and knowing every single tool or concept is going to come handy as a developer

# 1. Web Basics (Networking and Web Protocols)

## Web Fundamentals:

- HTTP/HTTPS Protocols: Requests, Responses, and Headers.
- Status Codes: Success (200s), Redirection (300s), Client Errors (400s), Server Errors (500s).
- RESTful Architecture: Principles and guidelines.
- Web Security Basics: SSL/TLS, CORS, CSRF, and XSS.
- DNS Basics: How domain names resolve to IP addresses.

## API Basics:

- Request Methods: [GET](#), [POST](#), [PUT](#), [DELETE](#).
- Request/Response Structure: Headers, Body, and Parameters.
- Query vs. Path Parameters.

---

# 2. Backend Development (Node.js & Express.js)

## Backend Basics

- **Node.js Core:**
  - Event Loop and Non-blocking I/O.
  - Core Modules: `fs`, `http`, `path`.
- **Express.js Fundamentals:**
  - Routing and Route Grouping.
  - Middleware: Built-in (`body-parser`, `cors`) and custom middleware.
  - Error Handling Basics.

## Advanced Backend

- **Global Error Handling:**
  - Centralized error middleware in Express.
  - Error logging using `winston` or `pino`.
- **API Design Patterns:**
  - Pagination, Filtering, and Sorting.
  - Versioning APIs.
- **Real-Time Communication:**
  - WebSockets with `socket.io`.
  - Server-Sent Events (SSE).
- **Background Jobs:**
  - Queue Management with `bull` or `Agenda.js`.
- **Advanced Middleware:**
  - Request validation (`express-validator` ).
  - Rate Limiting (`express-rate-limit`).
  - Custom Authentication Middleware.

---

## 3. Database Management

## Basic Database Management

- **MongoDB Fundamentals:**
  - CRUD Operations: Insert, Read, Update, Delete.
  - Indexing Basics for performance.
  - Schema Design - Flat vs. Nested Structures.
- **Mongoose ORM:**
  - Defining Schemas and Models.
  - Mongoose Middleware: Pre and Post Hooks.

## Advanced Database Management

- **Performance Optimization:**
    - Advanced Indexing: Compound Indexes, TTL Indexes.
    - Aggregation Framework: Pipelines and Stages.
  - **Database Security:**
    - Role-Based Access Control (RBAC).
    - Securing Connections: TLS and Authentication.
  - **Vector Databases:**
    - Use Cases: Smart search and embeddings.
    - Tools: Pinecone, Weaviate.
  - **Database Design for Scalability:**
    - Partitioning and Sharding.
    - Caching Strategies with Redis.
- 

## 4. Authentication and Authorization

- **Authentication Basics:**
  - JWT: Access and Refresh Tokens.

- Session-Based Authentication with Cookies.
  - **OAuth:**
    - Implementing Social Logins (Google, GitHub, etc.).
  - **Auth Providers:**
    - Tools like Clerk, Auth0, Firebase Authentication.
  - **Authorization:**
    - Role-Based Access Control (RBAC).
    - Permissions and Hierarchies.
- 

## 5. API Mocking and Testing

### Mocking APIs

#### Thunder Client

- Lightweight testing within VSCode
- Manage multiple environments to test APIs across different setups
- Organize and group API requests for efficient testing

#### Postman:

- Collections and Workspaces.
- Automated Testing with Newman.
- Environment Variables for Mock Data.
- Mock Service Worker (MSW): For intercepting network requests during development.

### API Testing:

- Supertest for Unit Testing APIs.
- Jest for Testing Logic.

---

## 6. Deployment and DevOps

### Deployment Basics

- **Frontend Hosting:** Vercel, Netlify.
- **Backend Hosting:** Railway, Render, Heroku.
- **Environment Variables:** Managing securely with `.env` and tools like Doppler.
- **MongoDB Atlas:** Cloud database management.

### Advanced Deployment (Containerization and Orchestration)

#### Docker:

- Docker Basics: Images, Containers, Volumes, and Networks.
- Dockerizing MERN Applications - Multi-stage builds for frontend and backend.

#### Kubernetes:

- Basics: Pods, Deployments, Services.
- Scaling Applications and Load Balancing.

#### CI/CD Pipelines:

- GitHub Actions for Build and Deploy Automation.
- Using Jenkins for CI/CD.

---

## 7. Advanced Tools and Utilities

### Code Quality and Linting:

- ESLint for JavaScript/TypeScript.

- Prettier for Code Formatting.

## Form Management and Validation:

- React Hook Form for forms.
- Zod/Yup for schema-based validation.

## Task Scheduling:

- `node-schedule`
- `Agenda.js`

## Monitoring and Debugging:

- Tools: PM2 for process management, LogRocket for performance monitoring.
  - Profiling with Node.js tools.
- 



**Register Free Webinar on the 15th of December to Know More**