Roadmap Updated

1. Fundamentals (Git, GitHub, VS Code, JS vs Node)

! Estimated Time: 4 hrs

What to Learn

- Git (init, add, commit, branch, merge, rebase, log, diff)
- GitHub (push/pull, fork, pull requests, resolving conflicts)
- JS vs Node.js differences

Q Under the Hood

- Git: .git folder stores snapshots in a DAG
- GitHub: adds collaboration, code reviews, and CI/CD
- VS Code: runs on Electron, talks to language servers for IntelliSense
- Node.js: V8 engine + libuv enables async, file access, and modules

Task

Install VS Code, set up Git, and make your first commit in a local repo.

2. JavaScript Essentials

Estimated Time: 12 hrs

What to Learn

- JavaScript data types and data structures
- stack and heap
- object and prototype
- factory function & constructor & class
- js grammar Scopes, closures, Hoisting
- expression vs statement
- event loop
- this
- Functions, async/await, promises
- ES Modules vs CommonJS

Q Under the Hood

- JS is parsed and executed by V8 (JIT compiled)
- Event loop handles async tasks (promises, timers)

Closures retain access to outer scope memory



Build a small CLI calculator app using functions and modules.

3. TypeScript

Estimated Time: 10 hrs

What to Learn

- Type annotations, interfaces, enums
- Unions, intersections, narrowing
- Generics and utility types
- tsconfig.json setup

Under the Hood

- TypeScript is compiled using tsc
- All types disappear at runtime purely static safety

Task

Convert your CLI calculator to TypeScript with typed inputs/outputs.

🗱 4. Node.js Core

Estimated Time: 12 hrs

What to Learn

- Core modules: fs , path , http , crypto , os , stream , events
- CommonJS: require, module.exports
- Buffers, file I/O, EventEmitter

Under the Hood

- Node.js uses V8 for JS execution
- libuv provides non-blocking I/O
- Each file is wrapped in a function for isolation

Task

Create a small file server using <a href="http://ht

5. Express.js (REST APIs)

Estimated Time: 14 hrs

What to Learn

- Request/response cycle
- Routers, middleware

Validation using Zod

- JWT-based auth and sessions
- Error handling
- . logs with winston

Under the Hood

- Express wraps Node's native http server
- Middleware chain uses next() to pass control
- Minimal layer over core network logic
- Task

Build a simple Todo API with CRUD endpoints and JWT authentication.

🔡 6. Databases & Prisma ORM

Estimated Time: 14 hrs

What to Learn

- DB types
- db terminologies
- SQL basics: tables, joins, constraints
- MySQL setup and schema
- Prisma modeling, migrations
- CRUD, pagination, filtering

Under the Hood

- Prisma generates a typed query client
- Runs optimized SQL under the hood using a Rust engine
- Supports batching and transactional queries
- Task

Connect your Todo API to MySQL using Prisma and persist data.

📆 7. NestJS (Scalable Server Framework)

**** Estimated Time: 20 hrs**

What to Learn

- Modules, controllers, services
- Dependency injection
- Guards, pipes, interceptors
- testing
- Prisma integration

Q Under the Hood

- Uses Reflect Metadata and decorators
- IOC container manages class dependencies
- Pluggable HTTP adapter: Express or Fastify



Build E Commerce App With Logs

8. Firebase & Google Cloud Functions

**** Estimated Time: 14 hrs**

What to Learn

- What is serverless?
- Firebase Auth, Firestore, and Functions v2
- Firestore rules and indexes
- FCM (notifications), Admin SDK

Q Under the Hood

- Functions run as containers on Cloud Run
- Triggers use Google Pub/Sub
- Firestore is a multi-region distributed NoSQL DB



Deploy a function that sends a notification when a new Firestore doc is added.

🧾 9. API Documentation with Swagger & Postman

Estimated Time: 4 hrs

What to Learn

- Swagger/OpenAPI basics
- Postman collections and environments
- Writing contract tests
- Mocking APIs

Under the Hood

- Swagger UI reads OpenAPI JSON and renders a docs interface
- Postman uses environment variables and scripts to automate testing

Task

Document your Todo API in Postman and export the collection.

1

10. Hosting & Deployment (Beginner Edition)

**** Estimated Time: 8 hrs**

What to Learn

- What's a VPS? SSH, ports, firewall
- Docker, PM2, Caddy 2 basics
- Static vs dynamic deployment

Under the Hood

- Docker runs apps in isolated containers
- PM2 manages Node processes and keeps them alive
- Caddy auto-generates HTTPS certs and proxies to your app



Deploy your app to a VPS using PM2 and expose it via Caddy 2.

11. Key Backend Terminology (Quick-Ref)

! Estimated Time: 4 hrs

Term	Definition	When Used
WebSockets	Two-way connection that stays open	Chat apps, games
SSE	Server sends text stream over HTTP	Live scores, dashboards
Long Polling	Client waits for server updates	Legacy real- time systems
WebRTC	Browser-to- browser media	Video calls, P2P files
Event-Driven	Architecture using event emitters	Node.js, pub/sub
Microservices	Small, independent services	Scalable systems
Monitoring Tools	Track uptime, logs, errors	
Caching	Redis, In- memory, CDN	



Write a short glossary (2 lines each) in your own words and match each term to a use case.