■ Backend Development 20-Week Schedule (Node.js + Express + MongoDB)

Week 1

Day 1: Backend vs Frontend, Client-Server model
Day 2: Internet basics (DNS, IP, TCP/IP)
Day 3: HTTP vs HTTPS, Request/Response cycle
Day 4: Status codes (200, 301, 404, 500)
Day 5: REST APIs overview
Day 6: Practice: Use Postman with JSONPlaceholder API
Day 7: Rest

Week 2

Day 1: Variables, datatypes, operators
Day 2: Functions & arrow functions
Day 3: Arrays & objects, destructuring
Day 4: Spread/rest operator, template literals
Day 5: Modules (CommonJS vs ES Modules)
Day 6: Practice: Student Grade Calculator
Day 7: Rest

Week 3

Day 1: Event loop & concurrency
Day 2: Callbacks & callback hell
Day 3: Promises (.then/.catch)
Day 4: async/await with try/catch
Day 5: Promise.all, timers (setTimeout, setInterval)
Day 6: Practice: Fake API Fetcher
Day 7: Rest

Week 4

Day 1: What is Node.js (V8, libuv)
Day 2: Running scripts, using process, REPL
Day 3: Core modules: fs, path
Day 4: Core modules: http (basic server)
Day 5: NPM basics, using nodemon
Day 6: Practice: Basic HTTP Server
Day 7: Rest

Week 5

Day 1: Installing & setting up Express
Day 2: Routes (GET, POST, PUT, DELETE)
Day 3: Request & response objects
Day 4: Middleware (built-in, custom)
Day 5: Serving static files

Day 6: Practice: Simple Express API
Day 7: Rest

Week 6

Day 1: Organizing routes with Router
Day 2: Error-handling middleware
Day 3: Request lifecycle
Day 4: API versioning strategies
Day 5: Async/await with Express routes
Day 6: Practice: Notes API with Express
Day 7: Rest

Week 7

Day 1: Intro to NoSQL vs SQL
Day 2: Installing MongoDB locally
Day 3: MongoDB Atlas setup
Day 4: CRUD operations in Mongo shell
Day 5: Indexes & query optimization
Day 6: Practice: Contact Book (Mongo Shell)
Day 7: Rest

Week 8

Day 1: Connecting Node.js to MongoDB
Day 2: Defining schemas & models
Day 3: CRUD with Mongoose
Day 4: Schema types, validation
Day 5: Pre/post middleware (hooks)
Day 6: Practice: Task Manager API (Mongoose)
Day 7: Rest

Week 9

Day 1: One-to-one relationships
Day 2: One-to-many relationships
Day 3: Many-to-many with refs
Day 4: Populating documents
Day 5: Embedding vs referencing data
Day 6: Practice: Blog API with Users & Posts
Day 7: Rest

Week 10

Day 1: REST principles
Day 2: Designing endpoints
Day 3: Status codes & error handling
Day 4: Sending JSON responses
Day 5: Structuring controllers & services
Day 6: Practice: Books API with Error Handling
Day 7: Rest

Week 11

Day 1: Filtering & searching
Day 2: Sorting
Day 3: Pagination
Day 4: Nested routes
Day 5: API documentation (Swagger)
Day 6: Practice: Product API with Filters & Docs
Day 7: Rest

Week 12

Day 1: Authentication vs authorization
Day 2: Password hashing with bcrypt
Day 3: Sessions vs JWT
Day 4: Implement JWT in Express
Day 5: Protecting routes with middleware
Day 6: Practice: Auth System with JWT
Day 7: Rest

Week 13

Day 1: Input validation & sanitization
Day 2: Rate limiting & brute-force prevention
Day 3: Helmet.js & securing headers
Day 4: HTTPS overview
Day 5: CORS in Express
Day 6: Practice: Secure Auth System
Day 7: Rest

Week 14

Day 1: Uploading files with Multer
Day 2: Serving uploaded files
Day 3: File validation (type, size)
Day 4: Storing images in MongoDB (GridFS)
Day 5: Cloud storage options (AWS S3/Cloudinary)
Day 6: Practice: Image Upload API
Day 7: Rest

Week 15

Day 1: Sending emails with Nodemailer
Day 2: Logging with Winston & Morgan
Day 3: Caching with Redis (intro)
Day 4: Scheduled tasks with Node Cron
Day 5: Environment variables with dotenv
Day 6: Practice: Email + Logging Extension
Day 7: Rest

Week 16

Day 1: Capstone: Setup & env config
Day 2: User model & auth routes
Day 3: JWT auth, role-based access
Day 4: Products model & CRUD
Day 5: Product filtering, pagination
Day 6: Practice: Test APIs with Postman
Day 7: Rest

Week 17

Day 1: Cart & order models
Day 2: Relationships (user ↔ orders ↔ products)
Day 3: File uploads for product images
Day 4: Email confirmation for orders
Day 5: Secure routes with middleware
Day 6: Practice: API documentation with Swagger
Day 7: Rest

Week 18

Day 1: Add logging & error handling
Day 2: Add rate limiting & security headers
Day 3: Add cloud DB (MongoDB Atlas)
Day 4: Deploy backend to Heroku/Render
Day 5: Final testing with Postman
Day 6: Practice: Submit E-commerce API
Day 7: Rest

Week 19

Day 1: Preparing for production (compression, dotenv)
Day 2: MongoDB Atlas setup for production
Day 3: Deploying to Heroku
Day 4: Deploying to Render/Railway
Day 5: Intro to Vercel/AWS deployment
Day 6: Practice: Deploy smaller project
Day 7: Rest

Week 20

Day 1: Load balancing basics
Day 2: Horizontal vs vertical scaling
Day 3: Docker basics for Node.js apps
Day 4: CI/CD with GitHub Actions
Day 5: WebSockets with Socket.io (chat app)
Day 6: Practice: Chat API with Socket.io
Day 7: Rest