Guide2Code - Express.js Roadmap

🟢 Phase 1: Beginner Level

📚 Topics to Learn:

1. Introduction to Express.js (What is Express? Why use it?)
2. Setting Up Express.js (Installing Node.js & npm)
3. Creating a Basic Server with Express.js
4. Understanding Middleware (Built-in & Custom Middleware)
5. Handling Requests & Responses (GET, POST, PUT, DELETE)
6. Working with Static Files (Serving HTML, CSS, JS)
7. Template Engines (EJS, Pug, Handlebars)
8. Basic Routing & URL Parameters
9. Error Handling in Express (try-catch, error middleware)
10. Working with Postman for API Testing

🛠 Beginner Project Ideas:

- **Basic Web Server** – Serve static web pages using Express.js
- **Simple REST API** – CRUD operations for a resource (e.g., users, products)
- **To-Do List API** – Create, update, and delete tasks
- **Weather App Backend** – Fetch weather data from an external API
- **Contact Form Handler** – Store form submissions in a database

🟡 Phase 2: Intermediate Level

📚 Topics to Learn:

1. Express Router & Modularization (Splitting Routes & Controllers)
2. Working with Databases (MongoDB with Mongoose, PostgreSQL with Sequelize)
3. Authentication & Authorization (JWT, bcrypt, OAuth)
4. Express with Middleware (Morgan, Helmet, CORS, Compression)
5. File Uploading (Multer for handling file uploads)
6. Implementing WebSockets (Real-time communication with Socket.io)
7. Logging & Monitoring (Winston, PM2)

8. Deploying Express Apps (Heroku, Vercel, AWS, DigitalOcean)
9. API Rate Limiting & Security Best Practices
10. Integrating Third-Party APIs (Payment Gateways, Social Logins)

🛠️ Intermediate Project Ideas:

- **User Authentication System** – Signup, login, and JWT-based authentication
- **Blog API** – CRUD API for managing blog posts and comments
- **E-commerce Backend** – Cart, orders, and payment gateway integration
- **Chat App Backend** – Real-time messaging with WebSockets
- **Expense Tracker API** – Manage income and expenses

🔴 Phase 3: Advanced Level

📚 Topics to Learn:

1. Microservices Architecture with Express.js
2. Advanced Database Optimization (Indexing, Transactions, Caching)
3. CI/CD Pipelines for Express Apps (GitHub Actions, Jenkins)
4. Containerization with Docker & Kubernetes
5. GraphQL with Express (Building APIs using Apollo Server)
6. Event-Driven Architecture (RabbitMQ, Kafka)
7. Serverless Express Applications (AWS Lambda, Firebase Functions)
8. Load Balancing & Performance Optimization (NGINX, Redis)
9. Building Scalable APIs with Rate Limiting & Caching
10. Full-Stack Development with Express & React/Next.js

🛠️ Advanced Project Ideas:

- **Microservices-Based Application** – Scalable Express.js services
- **Real-Time Stock Market Dashboard** – WebSockets-powered live stock updates
- **AI-Powered Chatbot Backend** – Integrate OpenAI API for chatbot responses
- **SaaS Application Backend** – Multi-tenant architecture with Express
- **Streaming Service Backend** – Video processing and real-time streaming