## **🔥 Full Backend Roadmap: Express + MongoDB + APIs**

### **Backend Development**

#### **Unit 1: Introduction to Backend**

* What is backend development?
* Frontend vs Backend
* Power of JavaScript in backend
* Client-server architecture
* What is an API?
* REST API fundamentals

🧠 **Mini Reflection Task:** Draw and explain client-server flow for a simple login form.

**Outcomes:**

* Understand the role of backend in web applications
* Explain client-server communication flow
* Define REST and explain how APIs work

#### **Unit 2: Intro to Express.js**

* What is Express.js?
* Setting up an Express server
* Understanding Middleware (use, next)
* Request and response objects
* Handling routes (GET, POST, PUT, DELETE)
* Serving static files

💡 **Checkpoint Activity:** Serve a static HTML + CSS login form using Express.

**Outcomes:**

* Set up a basic Express.js server
* Handle different routes and requests
* Serve static frontend content from the backend

#### **Unit 3: Build Your First API**

* What is JSON data?
* Using body-parser or express.json()
* CRUD operations (without DB)
* Error handling and HTTP status codes
* Modularizing routes and controllers
* Using Postman to test APIs

🧪 **Task:** Use Postman to test all CRUD endpoints for a basic Task Manager API.

🏅 **Milestone:** ✅ *Complete Postman Student Expert Certification*

***Outcomes:***

* *Build a functional REST API in Express*
* *Modularize routes and controllers*
* *Test endpoints using Postman*

#### **🔨 Mini Project Break 1: Task Manager API**

* In-memory CRUD
* Modular controllers
* Postman documentation

### **Working with Databases**

#### **Unit 4: Introduction to MongoDB**

* What is a database?
* SQL vs NoSQL
* What is MongoDB?
* Installing MongoDB locally or using Atlas
* Collections vs Documents
* Basic queries: find, insert, update, delete
* Using MongoDB Compass

📚 **Practice Task:** Insert, update, and delete a Books collection using Compass.

🏅 **Milestone:** ✅ *Earn a badge from* [*MongoDB University*](https://learn.mongodb.com/) (e.g., "Introduction to MongoDB")

**Outcomes:**

* Understand MongoDB structure and concepts
* Perform database operations via CLI and Compass
* Know how and when to use NoSQL databases

#### **Unit 5: Connecting Express to MongoDB**

* What is Mongoose?
* Connecting to MongoDB Atlas
* Defining schemas and models
* CRUD with Mongoose
* Schema validation and error handling

🛠️ **Refactor Task Manager API:**

* Replace in-memory storage with MongoDB
* Add proper schema validation

**Outcomes:**

* Connect an Express app to a real MongoDB database
* Create and validate schemas using Mongoose
* Refactor API to store and retrieve data from MongoDB

#### **🔒 Unit 6: Authentication & Authorization**

**Concepts Covered:**

* What is authentication vs authorization?
* Hashing passwords using bcrypt
* Generating and verifying JWT tokens
* Creating login and signup routes
* Protecting routes with middleware
* OAuth 2.0 login using Google (via Passport.js or Firebase Auth)

**Outcomes:**

* Implement secure login/signup using hashed passwords and JWT
* Create protected routes that only authenticated users can access
* Understand and integrate OAuth-based login (e.g., Google Sign-In)
* Differentiate between manual and third-party authentication flows

🧪 **Practice Task:** Add auth to Book or Task Manager API. Allow each user to manage their own data only.

#### **🔨 Mini Project Break 2: Book Review API**

* Users can register, login
* Authenticated users can CRUD books
* Users can leave reviews (nested schema)
* Include pagination, filtering, and validation

### **🌟 Capstone Project**

**🧱 Project Title:** "EduConnect API" (or any creative theme)

**Features:**

* Auth system (signup, login, JWT)
* Users can CRUD content (books/tasks/posts)
* Nested resources (e.g., reviews or comments)
* Pagination, search, filter
* Postman collection with tests