

E-Commerce Backend – Student Project Brief

Goal:

Build a basic e-commerce backend using **Node.js**, **Express.js**, and **MongoDB (Mongoose)**.

Focus on:

- CRUD operations
- Data modeling
- API design (RESTful)
- MongoDB relationships using Mongoose

No authentication or frontend required.

Project Structure

You must implement and connect these 3 main entities:

- **Users**
- **Products**
- **Orders**

Entity Descriptions

User

Field	Type	Notes
name	String	Required
email	String	Required, must be unique
address	String	Optional
createdAt	Date	Auto-generated

Product

Field	Type	Notes
name	String	Required
description	String	Optional
price	Number	Required
stock	Number	Defaults to 0
sellerId	ObjectId (reference to User)	Required – user who owns the product
createdAt	Date	Auto-generated

Order

Field	Type	Notes
buyerId	ObjectId (reference to User)	Required – user who made the purchase
sellerId	ObjectId (reference to User)	Required – seller of the products
products	Array of { productId, quantity }	At least one required
total	Number	Should be calculated manually
createdAt	Date	Auto-generated

Tasks

Users

- Create a user
- Read all users
- Read a user by ID
- Update a user
- Delete a user

Products

- Add a new product (must include sellerId)
- Read all products
- Read a product by ID
- Update product details
- Delete a product

Orders

- Create a new order (must include buyerId, sellerId, and products)
- Read all orders
- Read one order by ID
- Get all orders for a specific buyer
- Delete an order

Relationships Summary

Relationship	Description
User → Products	One user can have many products (seller)
User → Orders (as buyer)	One user can make many orders
User → Orders (as seller)	One user can receive many orders as seller
Order → Products	An order includes multiple products

Guidelines

- Use Mongoose schemas and references (ObjectId)
- Use RESTful routes
- Handle errors gracefully
- Use Postman to test endpoints

Bonus (Optional)

- Add product filtering or sorting
- Use `.populate()` to show full user and product data in orders