# 3) Inventory Management System

REST API | Validation | Authentication | Mongo Db

you'll create an API for managing products in an inventory system. Each product will have a name, category, quantity, price, and the date it was added.

1. **Setting up the Project Environment**:

- Create a new Node.js project by running npm init -y to generate the package.json file.
- Install necessary dependencies for the project:
  - **Express.js**: npm install express
  - **Mongoose (MongoDB ORM)**: npm install mongoose
  - **dotenv (for environment variables)**: npm install dotenv
  - **Other utility libraries** (optional, for validation or logging): npm install body-parser (for handling JSON bodies).

2. **Project Structure**: Ensure the following directory structure for your project:

```
root/
├── .env              # Environment variable file (to store sensitive data like DB URI)
├── index.js          # Main entry point to configure and run the Express server
├── models/           # Directory for MongoDB schemas
│   ├── book.js       # Example schema for the book model
│   ├── user.js       # Example schema for the user model
│   └── product.js    # Example schema for the product model
├── controllers/      # Directory for controller logic
│   ├── bookController.js
│   ├── userController.js
│   └── productController.js
├── middlewares/      # Directory for any custom middlewares
│   └── validateInput.js
└── package.json      # Contains project dependencies and scripts
```

1. Create a MongoDB schema for the Product model with the following fields:
    - name: String (required)
    - category: String (required)
    - quantity: Number (required)
    - price: Number (required)
    - dateAdded: Date (required)
2. Implement the following APIs:
    1. **GET /products**: Fetch all products from the database.
    2. **GET /products/category/:category**: Fetch products by their category.
    3. **POST /products**: Create a new product.
    4. **GET /products/:quantity/:price/:date**: Fetch products by quantity, price, and date.
    5. **DELETE /products/:id**: Delete a product by its ID.

Sample Data: orders

```
[
{
  "name": "Laptop",
  "category": "Electronics",
  "quantity": 10,
  "price": 899.99,
  "dateAdded": "2023-08-01T00:00:00Z"
},
{
  "name": "Smartphone",
  "category": "Electronics",
  "quantity": 50,
  "price": 599.99,
  "dateAdded": "2023-09-15T00:00:00Z"
},
{
  "name": "Desk Chair",
  "category": "Furniture",
  "quantity": 15,
  "price": 129.99,
  "dateAdded": "2023-07-20T00:00:00Z"
}]
```