

# CLA Web Development Bootcamp Assignment

**Assignment:** Building a product management system REST API with Express.js

## Objective:

The objective of this assignment is to create a robust REST API using Express.js for a product management system. The API will include CRUD operations for products, and order functionality.

## Tasks:

1. Setup Express.js Project:
  - Initialize a new Node.js project.
  - Install Express.js and other necessary packages (express, mongoose, dotenv, etc.).
  - Create the main file (app.js or server.js).
2. Create data file to simulate database interaction:
  - Set up a js file for storing data to interact with later on using the REST API.
  - The data should follow the structure mentioned below.
3. Implement CRUD Operations for Products:
  - Create routes for CRUD operations for products (GET, POST, PUT, DELETE).
  - Implement middleware for validating requests and handling errors.
  - Secure routes as needed.
4. Implement Order Functionality:
  - Create routes for placing orders (**POST /orders should be a protected route**) and retrieving orders (GET /orders) and getting an order by id (GET /orders/:id).
  - **Add a protected route to get all user orders (GET /orders/user)**
5. Implement Authentication System:
  - **Create routes for user authentication (POST /register, POST /login).**
  - **Use bcrypt to hash passwords before storing them in the database.**
  - **Use jsonwebtoken to generate and verify authentication tokens.**
  - **Add an admin route to get all users (GET /users)**
  - **Bonus: get current user (GET /users/profile), get a user by id (GET /users/:id), update a user and delete a user (All except the profile route should be admin routes, the profile one should be just protected).**
6. Testing:
  - Test your API using Postman.
  - Ensure that CRUD operations, and order functionality work as expected.

## **7. Documentation:**

- **Document your API endpoints, request/response formats, and authentication mechanism. You can use Swagger or Postman for that.**
- **Provide clear instructions on how to use your API.**

### **Product Model:**

Define a schema for products with the following properties:

- name: A string representing the name of the product. (Required)
- description: A string describing the product. (Required)
- price: A number representing the price of the product. (Required)
- quantity: A number representing the available quantity of the product. (Required)
- createdAt: A date representing the creation date of the product. (Default: Current Date)

### **Order Model:**

Define a schema for orders with the following properties:

- products: An array of references to the products included in the order. (Required)
- totalPrice: A number representing the total price of the order. (Required)
- createdAt: A date representing the creation date of the order. (Default: Current Date)

### **Submission Guidelines:**

- The work should be done in a GitHub repository to share later with the instructor
- After completing the tests and documentation with Postman, share the workspace with the instructor.

### **Note:**

- Follow standard conventions for defining schema properties.
- Include appropriate constraints such as required fields and unique values.
- Provide clear explanations for the different parts of code as comments for better understanding.