

Project Title

E-Commerce Web Application

Objective

To design and develop a full-stack e-commerce website where:

- Users can sign up, login, view products, add items to cart, and checkout
- Admin can add, update, and delete products
- Backend APIs communicate with frontend using JSON and JWT authentication

Tech Stack

Area	Technology
Frontend	ReactJS
Backend	Node.js
Database	MongoDB
State/Requests	Axios
Authentication	JWT
Version Control	Git & GitHub

Database Schema

Users

Field	Type
name	String
email	String (unique)
password	Hashed
role	user/admin
cart	Array of product + qty

Products

Field	Type
name	String
price	Number
description	String
stock	Number

Orders

Field	Type
user	ObjectId
items	product + qty + price
total	Number
status	placed/shipped

Frontend UI

Page	Purpose
Home	Landing screen
Signup	Create user account
Login	Authentication
User Dashboard	Browse products
Cart Page	View and checkout
Admin Dashboard	CRUD Products

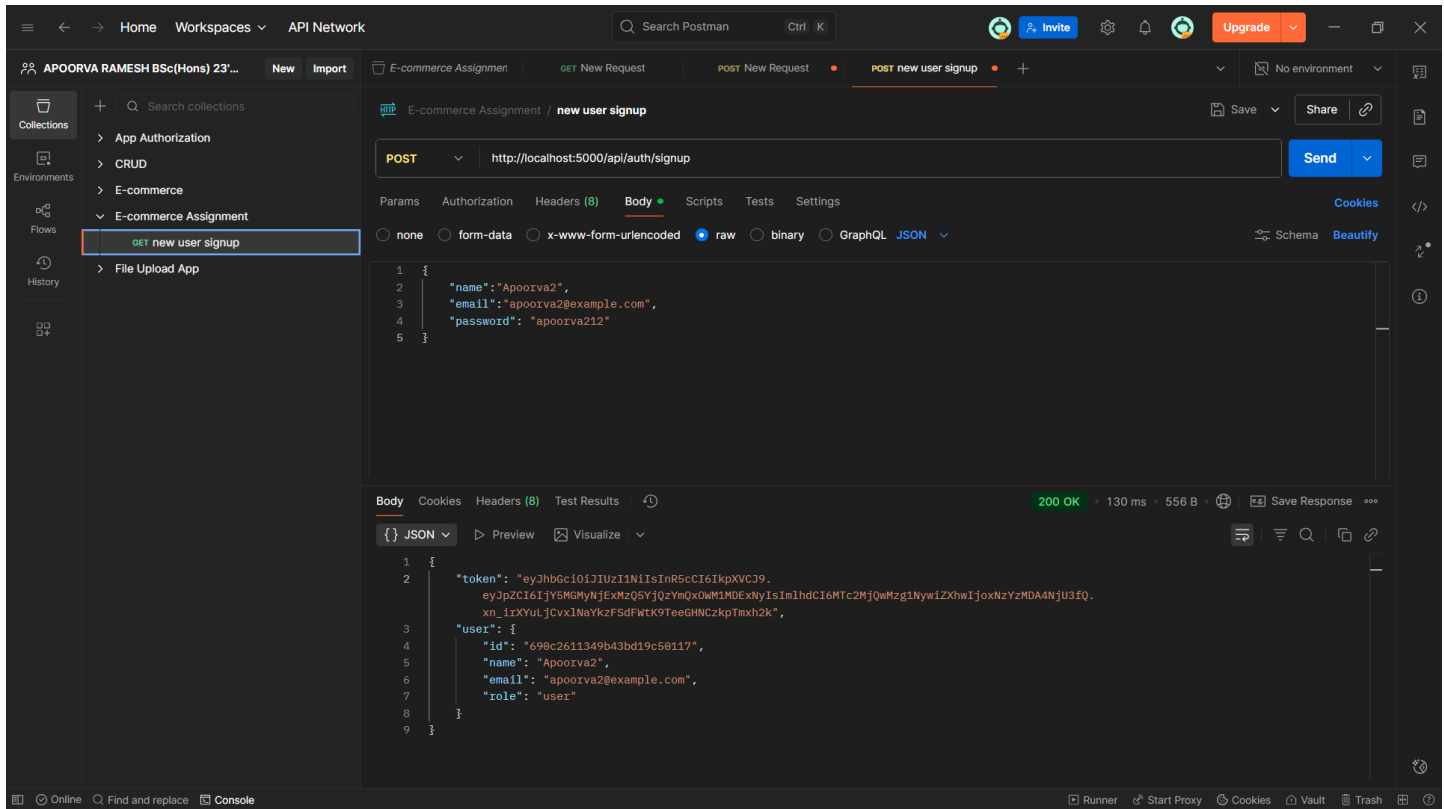
Backend & API Integration

Key Routes

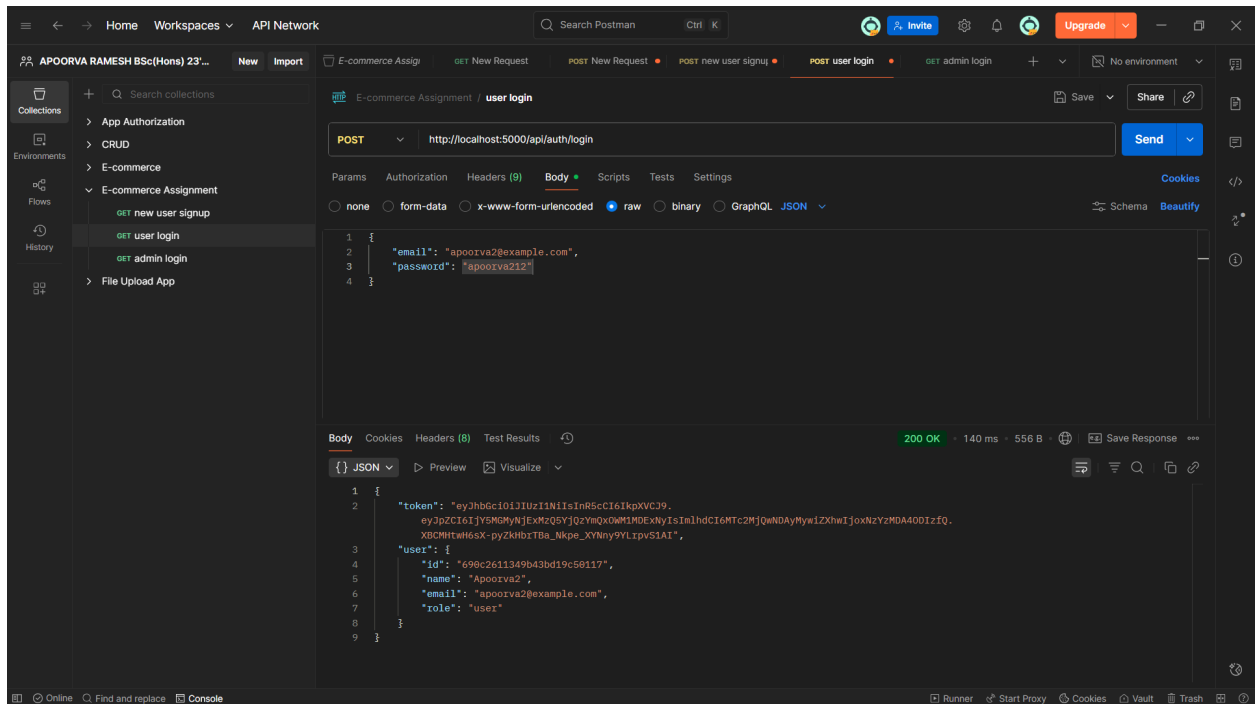
Endpoint	Function
POST /auth/signup	User signup
POST /auth/login	JWT login
GET /products	Fetch products
POST /products	Admin add product
PUT /products/:id	Update product
DELETE /products/:id	Delete product
POST /cart/add-to-cart	Add to cart
POST /orders	Checkout order

API Testing Using Postman

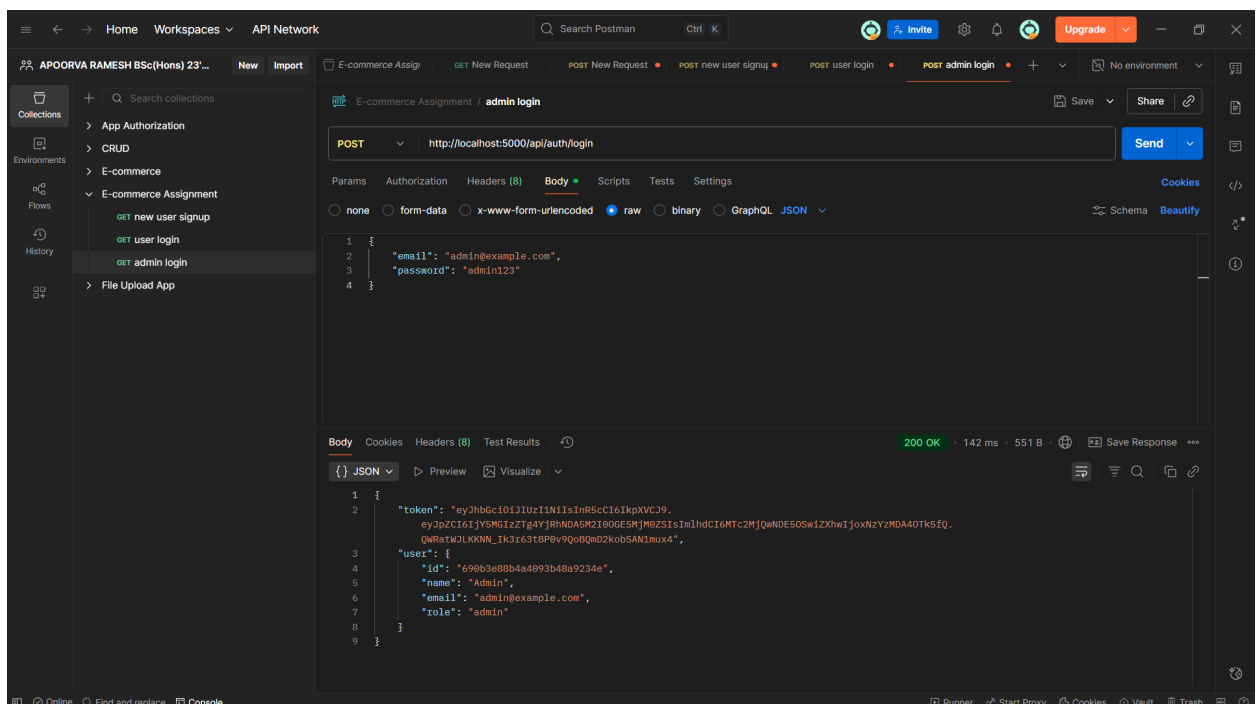
1. Signup request success JSON



User login

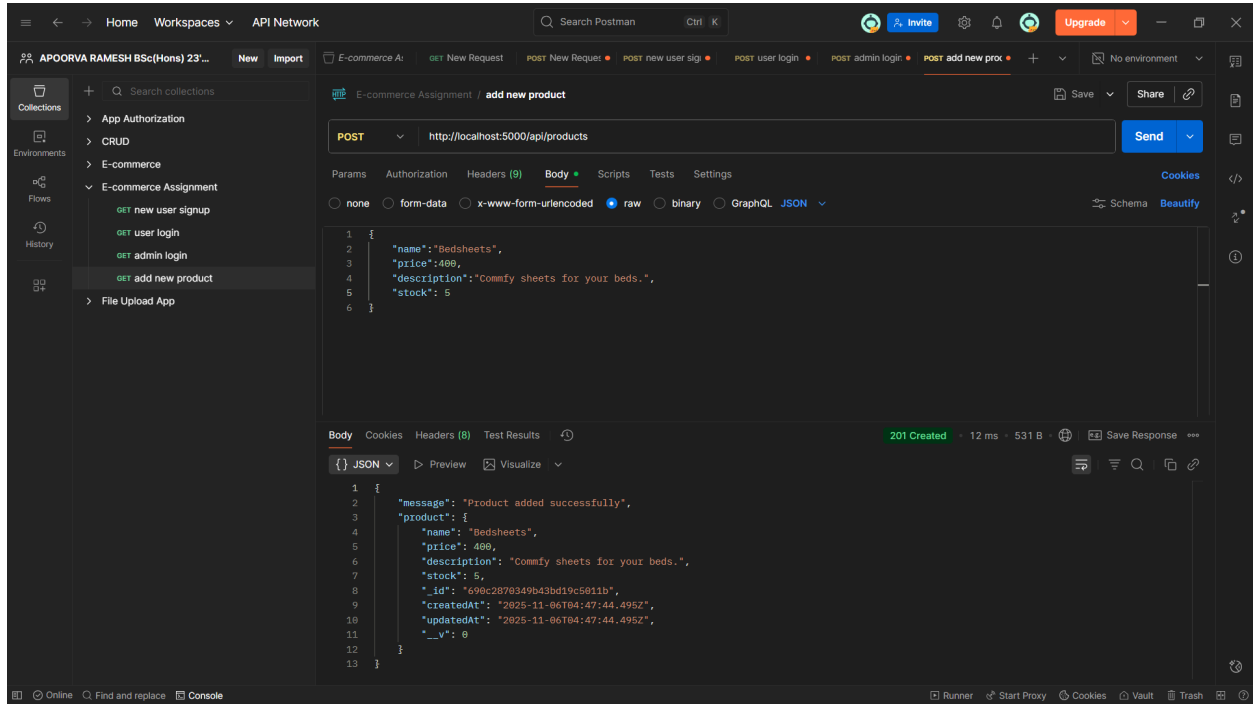


Admin Login

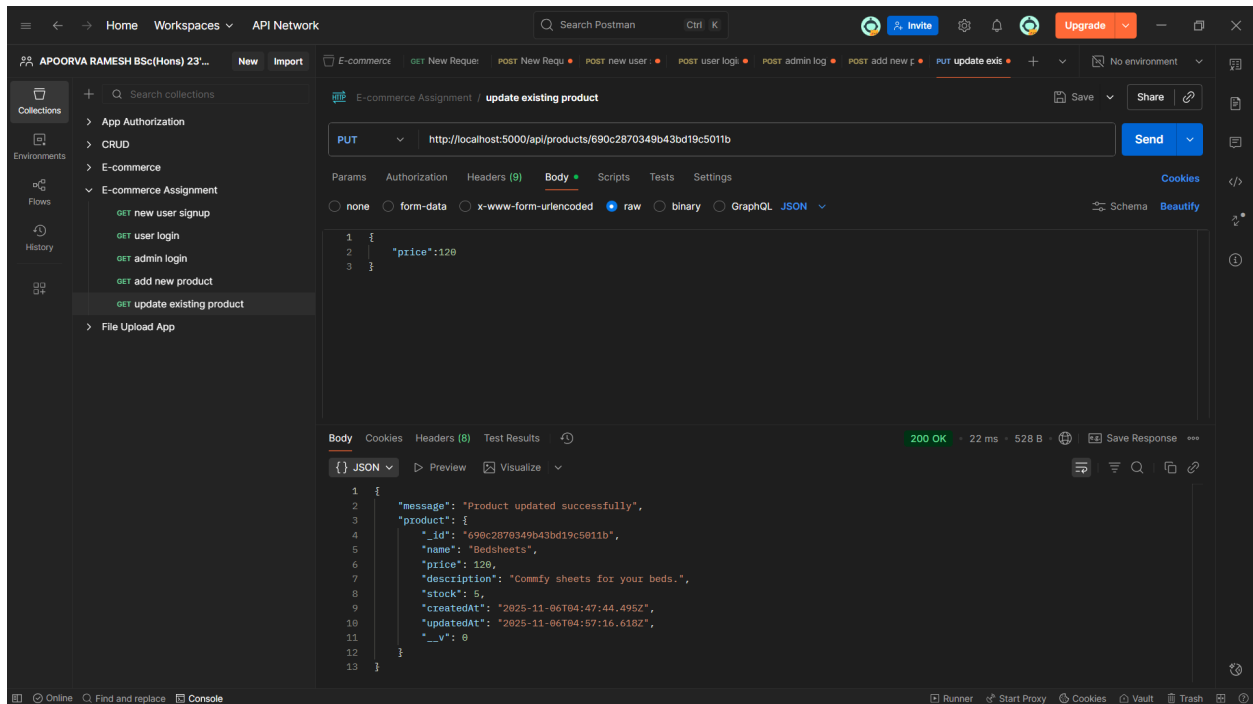


3. CRUD Operations on Products (Admin)

Create a new product



Update an existing product



Fetch products

The screenshot shows the Postman interface with a workspace named "API Network". The left sidebar displays a collection of requests under "E-commerce Assignment", with "GET fetch products" selected. The main panel shows the details of this request:

- Method:** GET
- URL:** http://localhost:5000/api/products
- Headers:** 7 headers are shown, with "Authorization" selected. The value is "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjE2IiwiaWF0Ijoi2025-11-06T04:47:44.495Z" (Note: the image shows a truncated value).
- Body:** The response is shown in JSON format, indicating a 200 OK status with a response time of 17 ms and a size of 907 B. The JSON data is as follows:

```
1 [
2   {
3     "_id": "698c2878349b43bd19c5e11b",
4     "name": "Bedsheets",
5     "price": 120,
6     "description": "Comfy sheets for your beds.",
7     "stock": 5,
8     "createdAt": "2025-11-06T04:47:44.495Z",
9     "updatedAt": "2025-11-06T04:57:16.618Z",
10    "__v": 0
11  },
12  {
13    "_id": "698b4fff2cf95cbe3297dda7",
14    "name": "Headphones"
```

4. Add to cart (User)

The screenshot shows the Postman interface with a workspace named "API Network". The left sidebar displays a collection of requests under "E-commerce Assignment", with "POST add product to cart" selected. The main panel shows the details of this request:

- Method:** POST
- URL:** http://localhost:5000/api/cart/add-to-cart
- Body:** The request body is shown in JSON format, indicating a 200 OK status with a response time of 16 ms and a size of 508 B. The JSON data is as follows:

```
1 {
2   "product": {
3     "product_id": "698c2878349b43bd19c5e11b"
4   }
5 }
```

The response body is shown in JSON format, indicating a 200 OK status with a response time of 16 ms and a size of 508 B. The JSON data is as follows:

```
1 {
2   "msg": "Added to cart",
3   "product": {
4     "_id": "698c2878349b43bd19c5e11b",
5     "name": "Bedsheets",
6     "price": 120,
7     "description": "Comfy sheets for your beds.",
8     "stock": 4,
9     "createdAt": "2025-11-06T04:47:44.495Z",
10    "updatedAt": "2025-11-06T09:31:26.477Z",
11    "__v": 0
12  }
13 }
```


Github Repo: [E-Commerce Website](#)