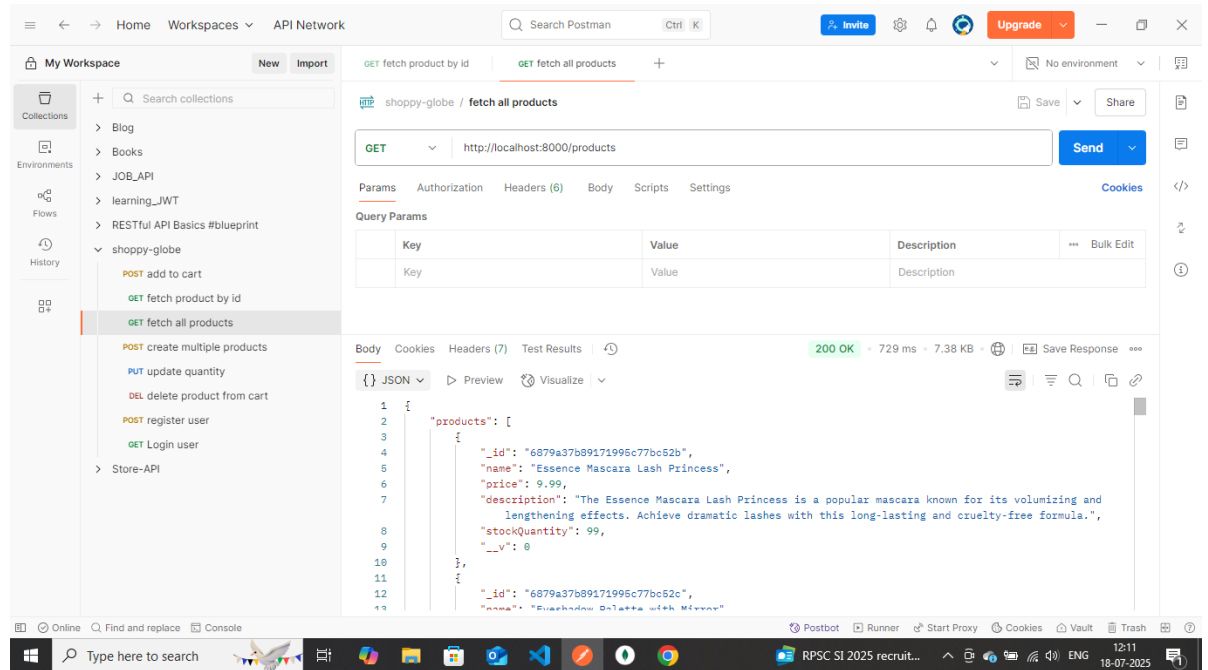


# API Testing with Postman

## 1. Get all Products :

Method- GET

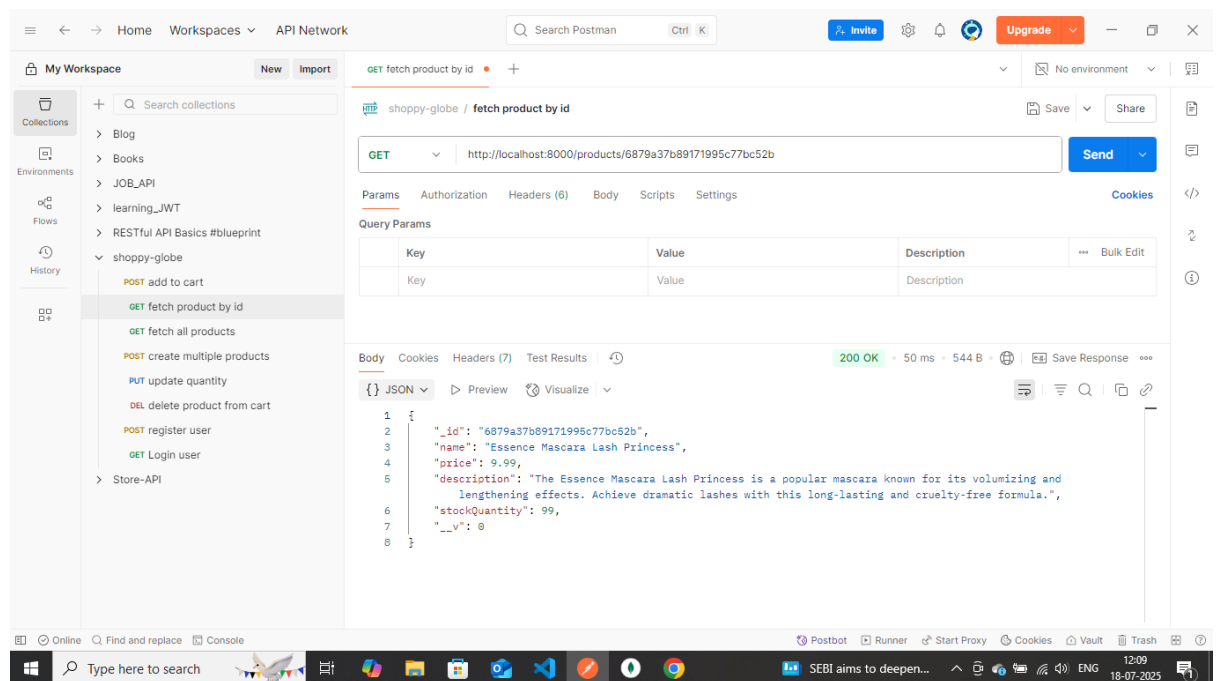
URL- <http://localhost:8000/products>



## 2. Get a product by Id

Method- GET

URL- <http://localhost:8000/products/:id>



### 3. Register new user

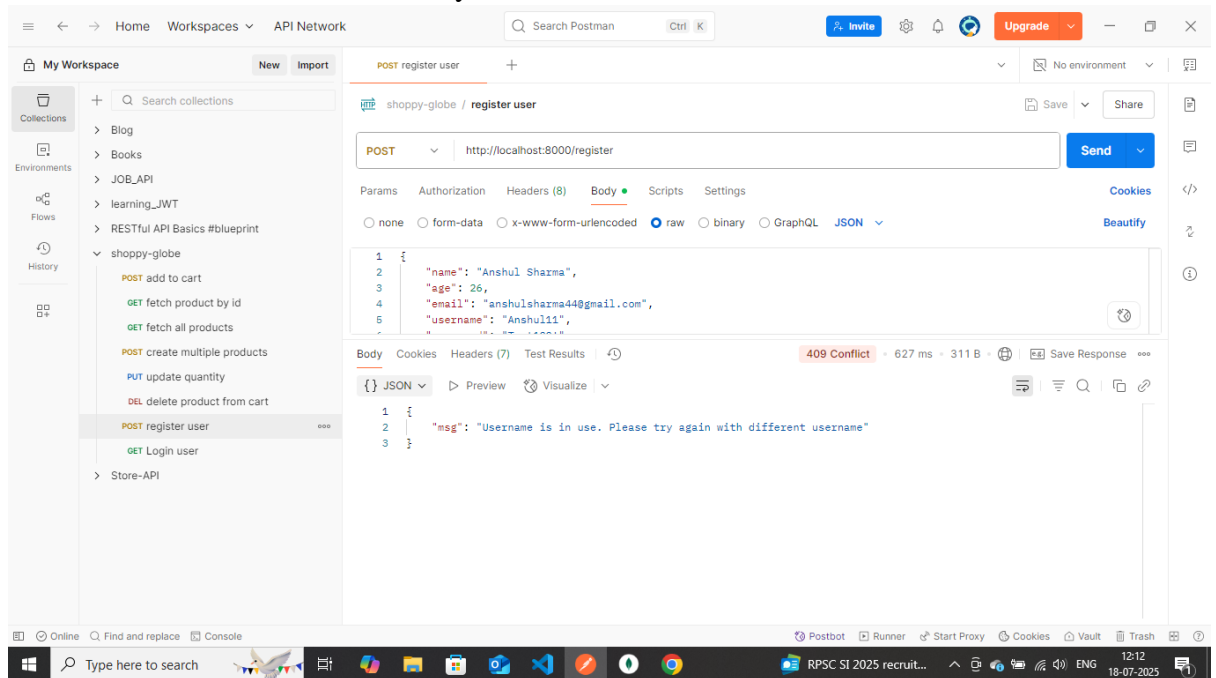
Method- POST

URL- <http://localhost:8000/register>

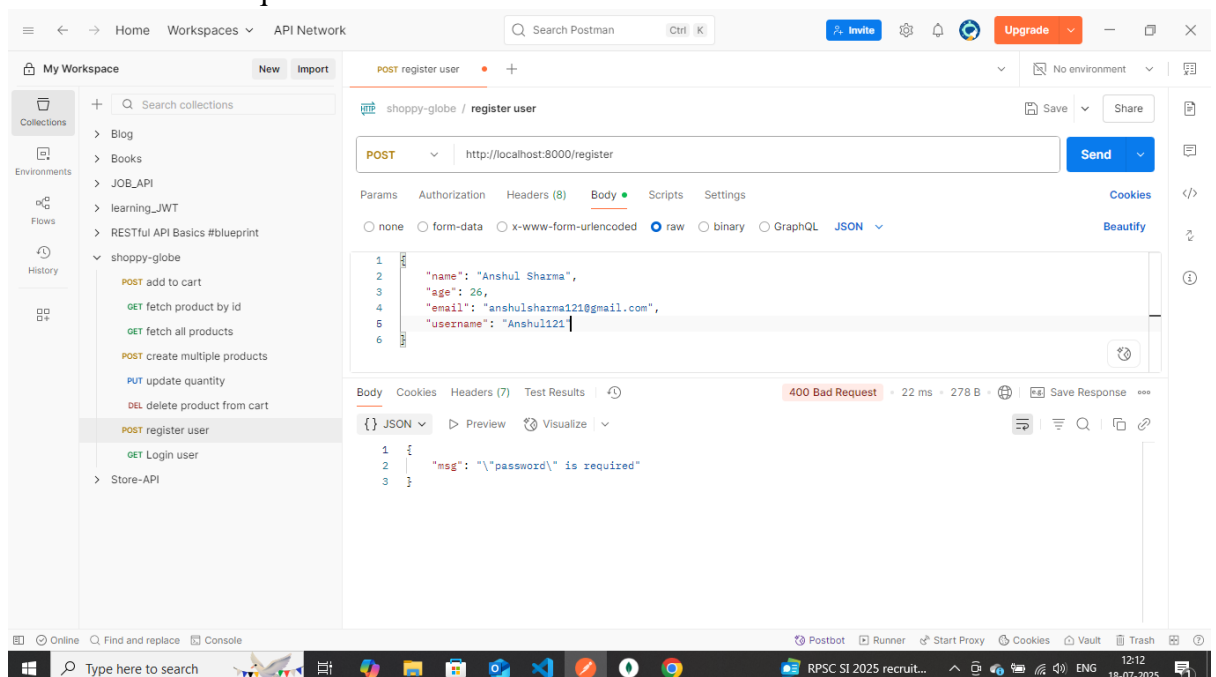
Request Body- {

```
"name": "Anshul Sharma",  
"age": 26,  
"email": "anshulsharma121@gmail.com",  
"username": "Anshul121",  
"password": "Test123!" }
```

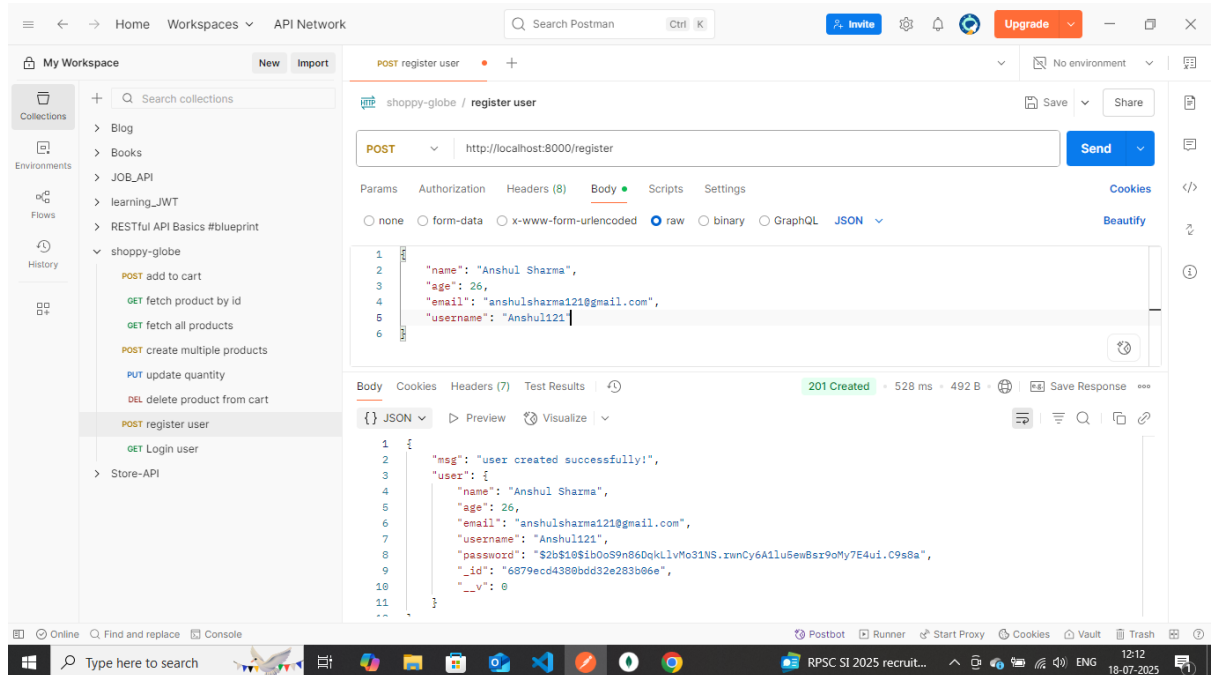
Failure Case- If username is already in use



Failure Case- If Request is Invalid



## Success Case-



## 4. Login

Method- POST

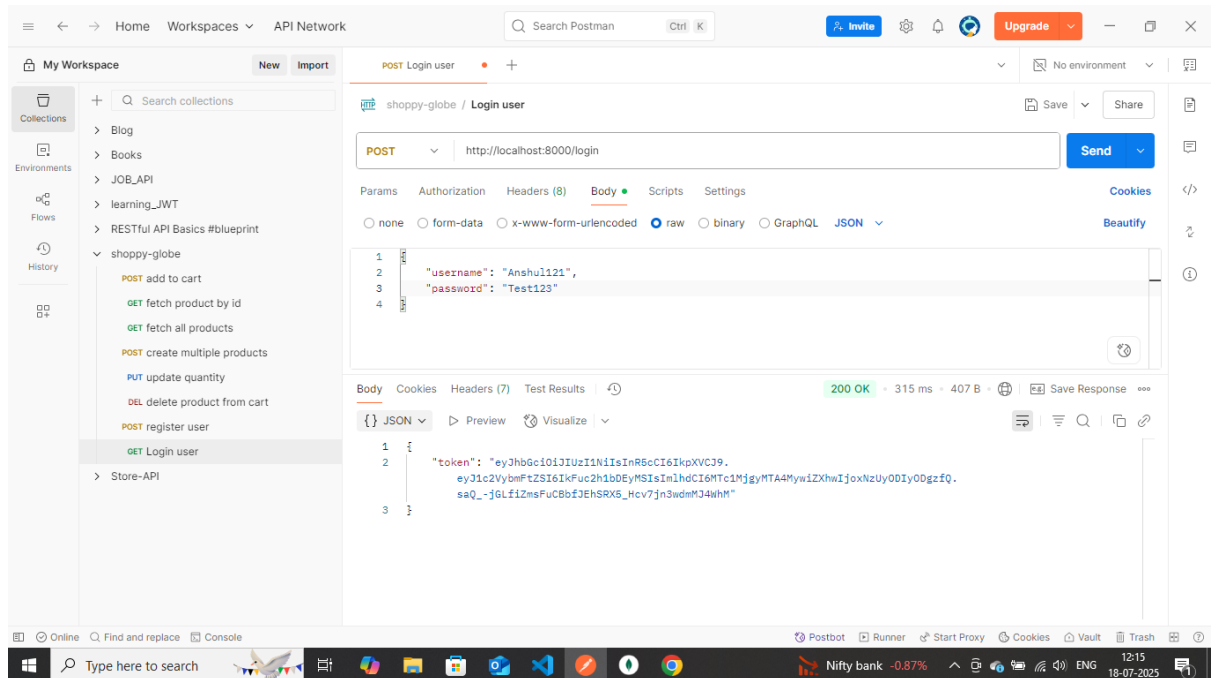
URL- <http://localhost:8000/login>

Request Body- {

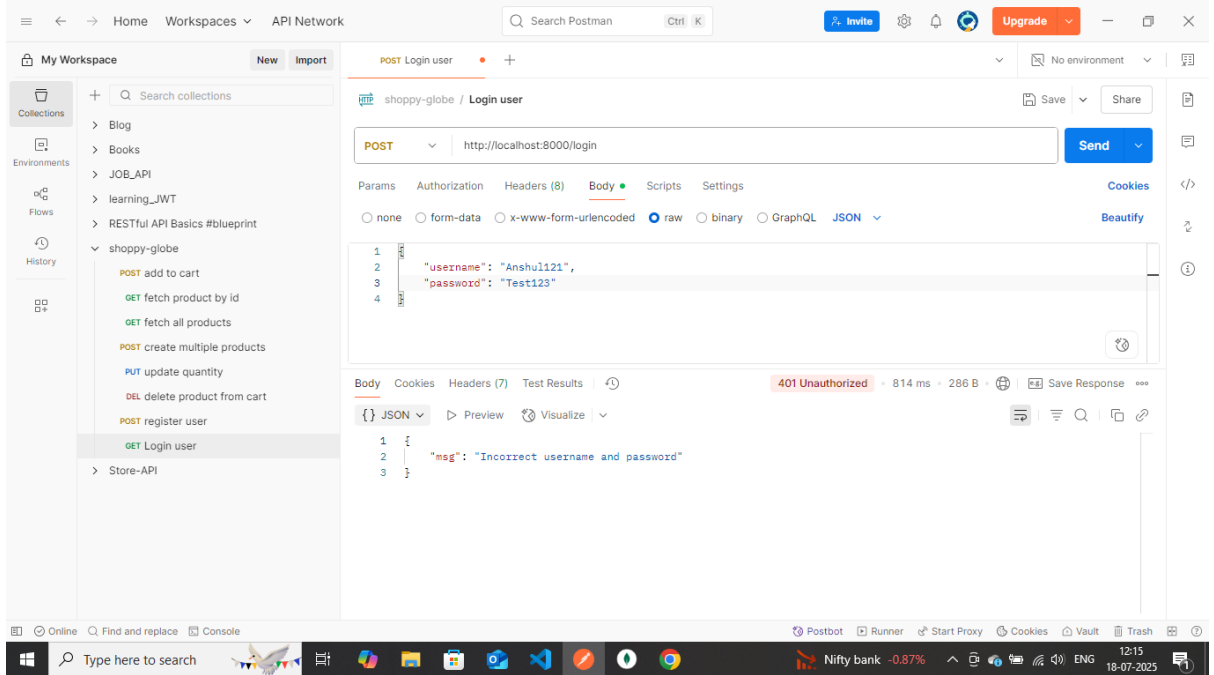
```
"username": "Anshul121",
"password": "Test123!"
```

}

## Success Case-



## Failure Case- If credentials are incorrect



## 5. Add a product to cart

Method- POST

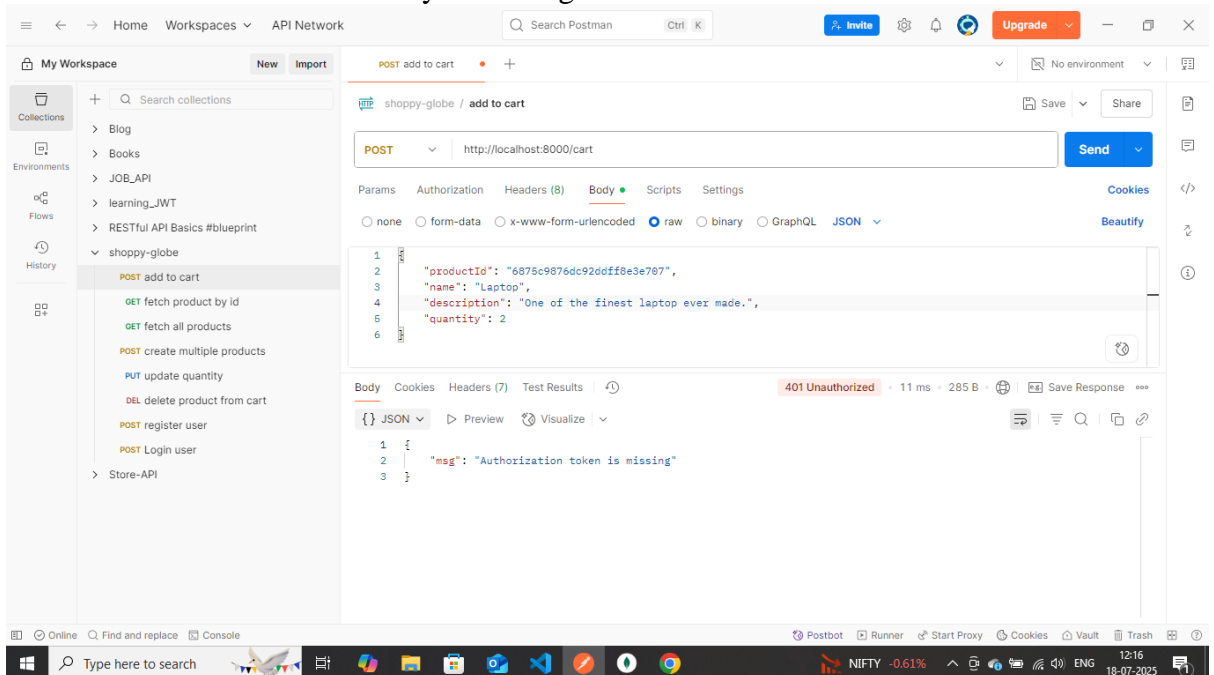
URL- <http://localhost:8000/cart>

Request Body- {

```
"productId": "6879a37b89171995c77bc52b",
"name": "Essence Mascara Lash Princess",
"description": "The Essence Mascara.",
"quantity": 2
}
```

Should have Authorization key in headers that will contain the JWT.

## Failure Case- If authorization key is missing



## Failure Case- If there is no record in product collection for the provided productId

The screenshot shows the Postman interface with a workspace named 'shoppay-globe'. A POST request is configured to 'http://localhost:8000/cart'. The request body is raw JSON, containing the following data:

```
1 {
2   "productId": "6875c9876dc92ddff8e3e787",
3   "name": "Laptop",
4   "description": "One of the finest laptop ever made.",
5   "quantity": 2
6 }
```

The response is a 404 Not Found status, indicating that no product was found for the provided productId. The response body is JSON, containing the following data:

```
1 {
2   "msg": "No product found for provided productId"
3 }
```

## Success Case-

The screenshot shows the Postman interface with a workspace named 'shoppay-globe'. A POST request is configured to 'http://localhost:8000/cart'. The request body is raw JSON, containing the following data:

```
1 {
2   "productId": "6879a37b89171995c77bc52b",
3   "name": "Essence Mascara Lash Princess",
4   "description": "The Essence Mascara Lash Princess is a popular mascara known for its volumizing and lengthening effects. Achieve dramatic lashes with this long-lasting and cruelty-free formula.",
5   "quantity": 2
6 }
```

The response is a 201 Created status, indicating that the item was successfully added to the cart. The response body is JSON, containing the following data:

```
1 {
2   "cartData": {
3     "productId": "6879a37b89171995c77bc52b",
4     "quantity": 2,
5     "name": "Essence Mascara Lash Princess",
6     "description": "The Essence Mascara Lash Princess is a popular mascara known for its volumizing and lengthening effects. Achieve dramatic lashes with this long-lasting and cruelty-free formula.",
7     "_id": "6879eef388bdd32e283b876",
8     "__v": 0
9   },
10  "msg": "Item added to cart successfully!"
11 }
```

6. Update quantity of item added in cart

### Method- PUT

URL- <http://localhost:8000/cart/:id>

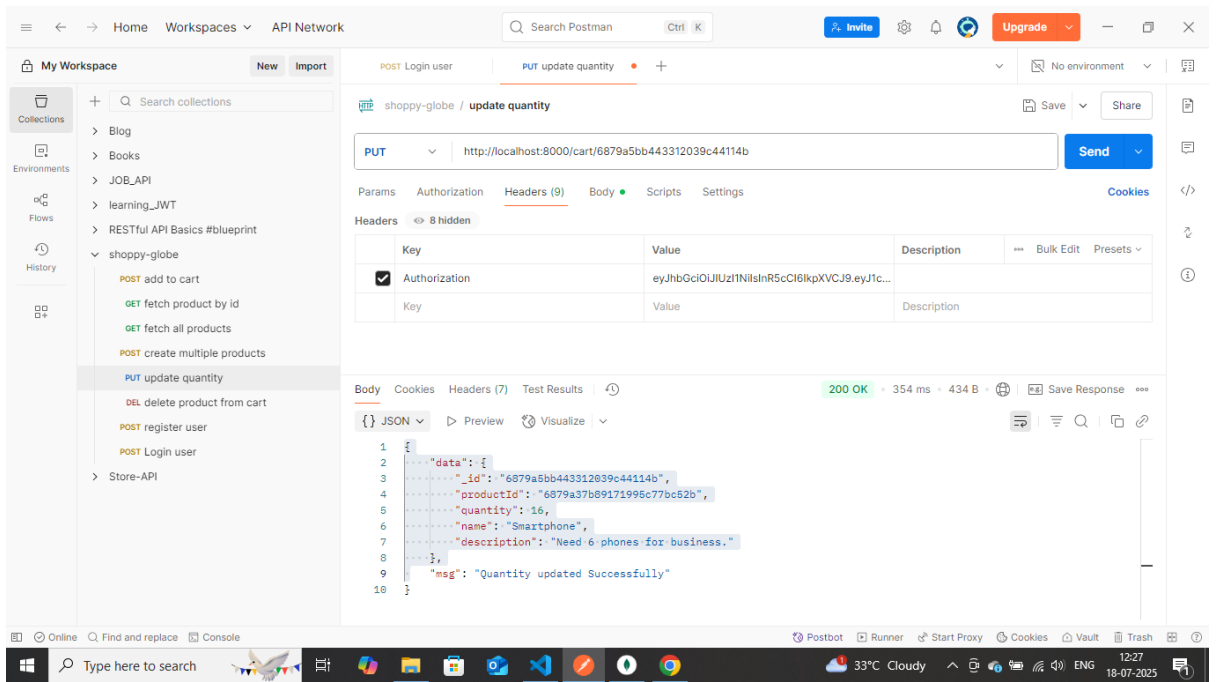
Request Body- {

```
"quantity": 2
```

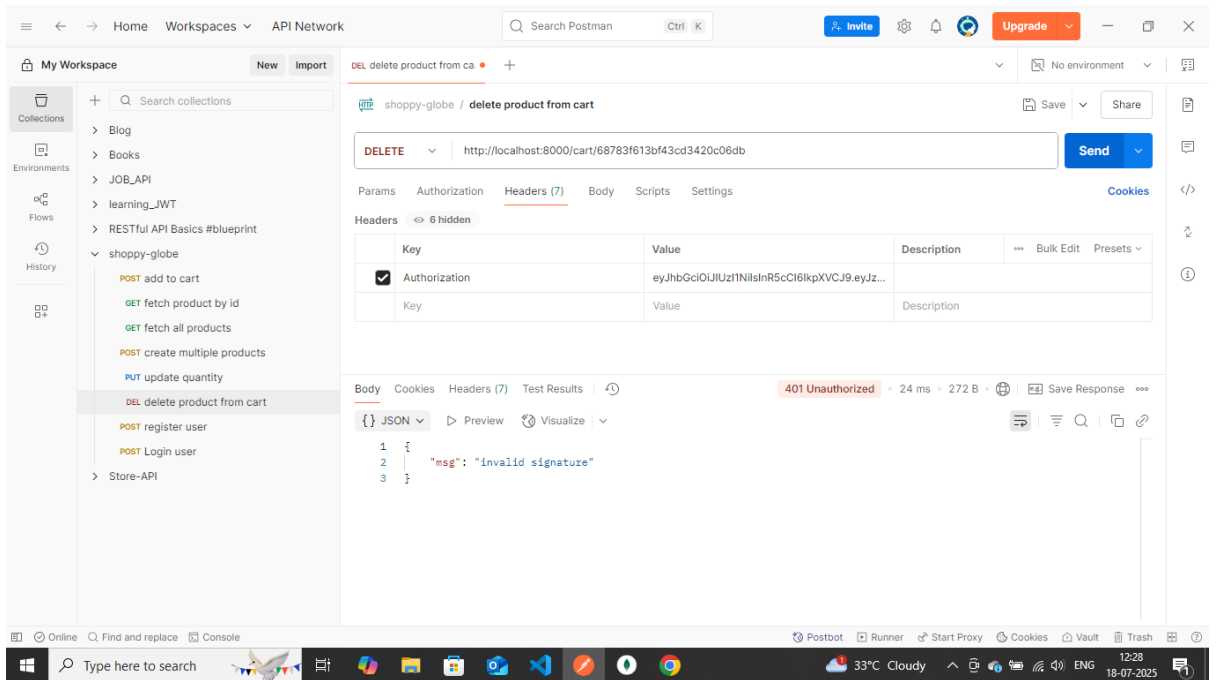
}

Should have Authorization key in headers that will contain the JWT.

### Success Case-



### Failure Case- If JWT is invalid



## 7. Remove product from the cart

Method- Delete

URL- <http://localhost:8000/cart/:id>

Should have Authorization key in headers that will contain the JWT.

Success Case-

The screenshot shows the Postman interface with a workspace named 'shoppy-globe'. The selected collection is 'RESTful API Basics #blueprint', and the specific request is 'DEL delete product from cart'. The URL is set to 'http://localhost:8000/cart/6879a5bb443312039c44114b'. The 'Headers' tab is active, showing an 'Authorization' header with a JWT token. The 'Body' tab shows a JSON response: 

```
{ 1: { 2: { "msg": "Product removed from the cart!" } } }
```

. The status is '200 OK' with a response time of 65 ms and a body size of 275 B.

Failure Case- If Id provided in request params is not a valid mongo ID

The screenshot shows the same Postman interface, but the URL is changed to 'http://localhost:8000/cart/12'. The 'Headers' tab still shows the 'Authorization' header. The 'Body' tab shows a JSON response: 

```
{ 1: { 2: { "msg": "ID should be a valid ObjectId" } } }
```

. The status is '400 Bad Request' with a response time of 25 ms and a body size of 283 B.