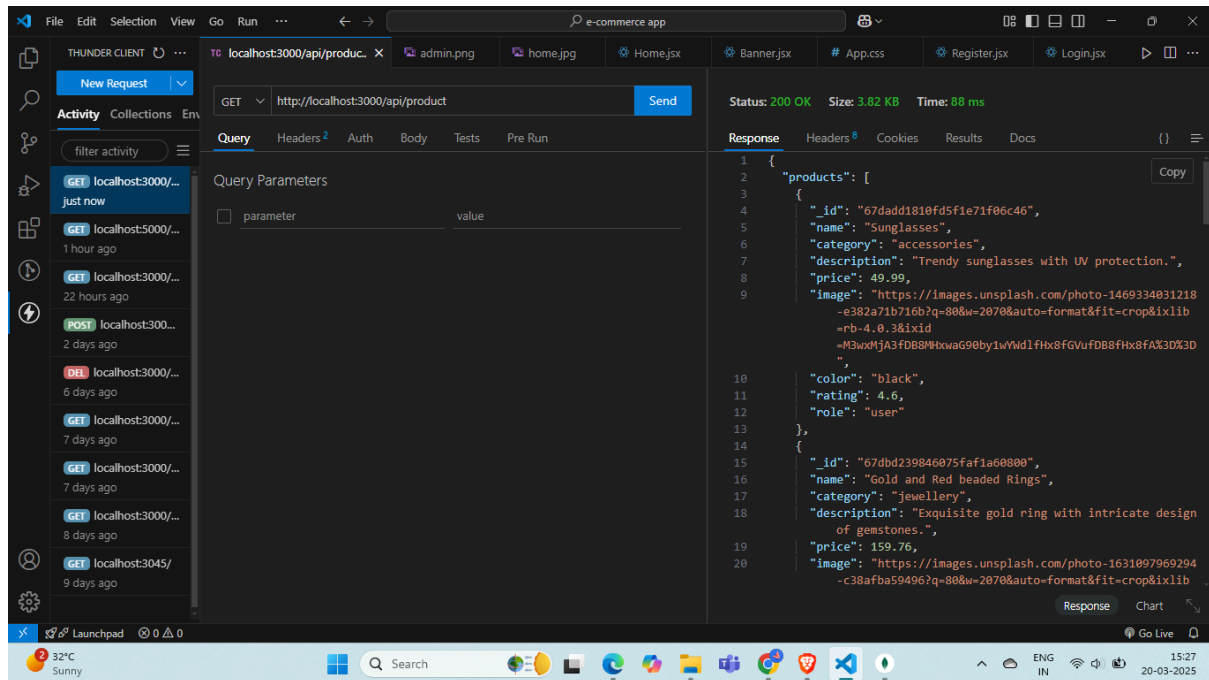


Products API Testing

Get All Products

Endpoint: GET localhost:3000/api/product

Expected Response: List of all products.

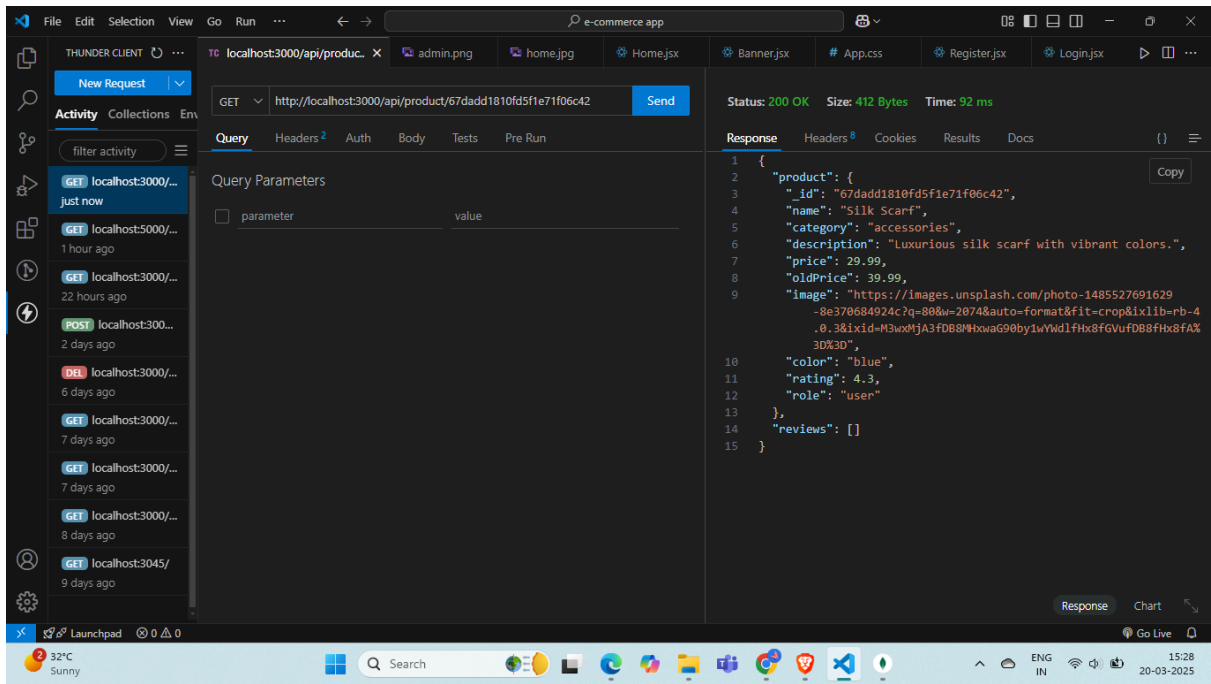


Get Product by ID

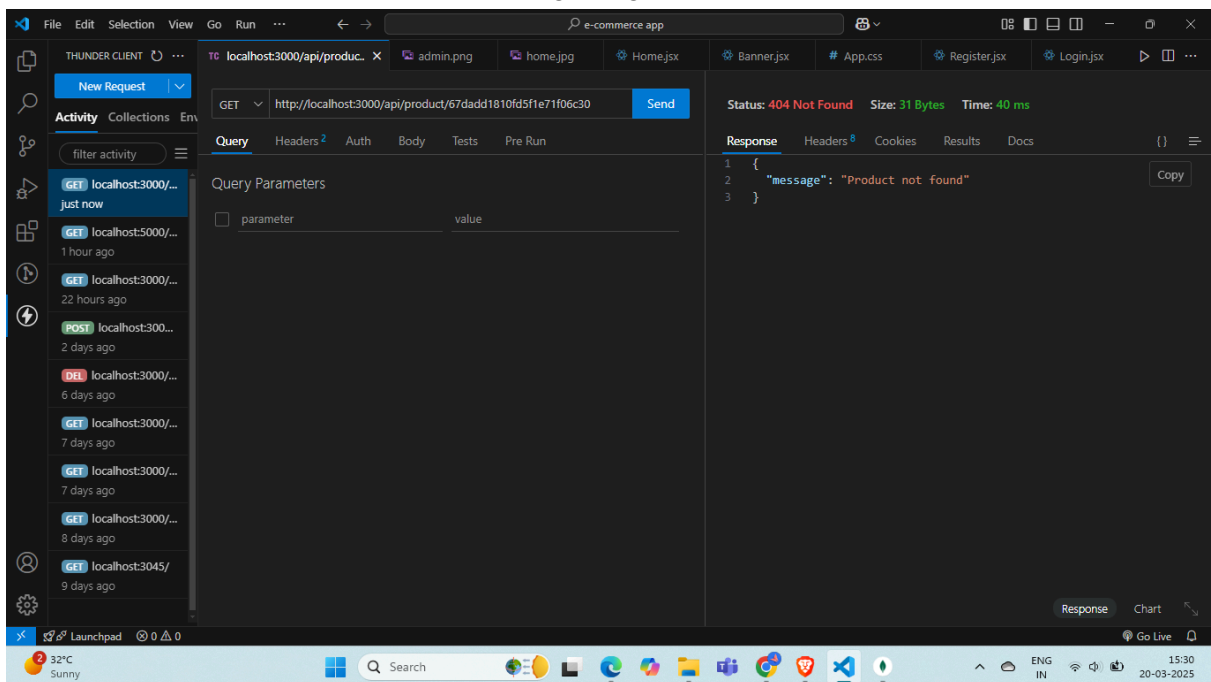
Endpoint: GET localhost: 3000/api/product/67dadd1810fd5f1e71f06c42

Expected Response: Product details based on ID.

Test Cases: Correct ID: Should return the product details



Incorrect ID: Should return an error message (e.g., "Product not found").



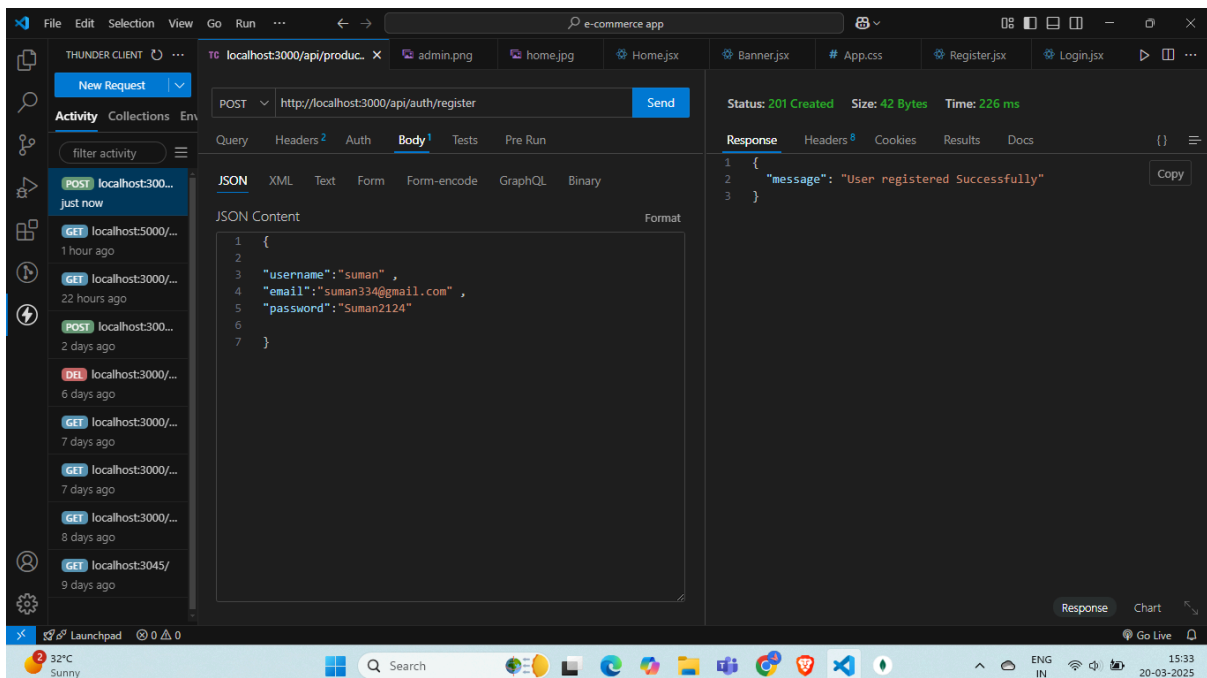
Authentication API Testing

User Registration

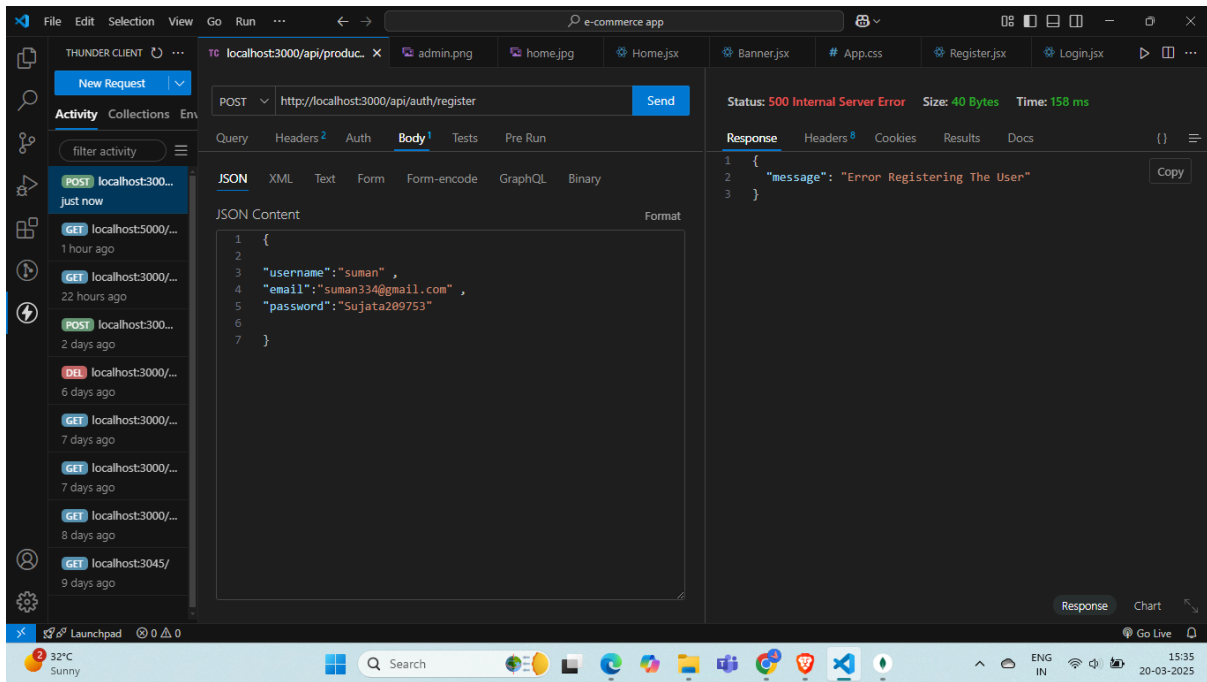
Endpoint: POST localhost:3000/api/auth/register

Body : { "username" : "suman" , "email" : "suman334@gmail.com" ,
"Password" : "Suman2124" }

Expected Response: User registered successfully



With Error (Duplicate username/email): Should return an error message (e.g., "Error Registering The User").

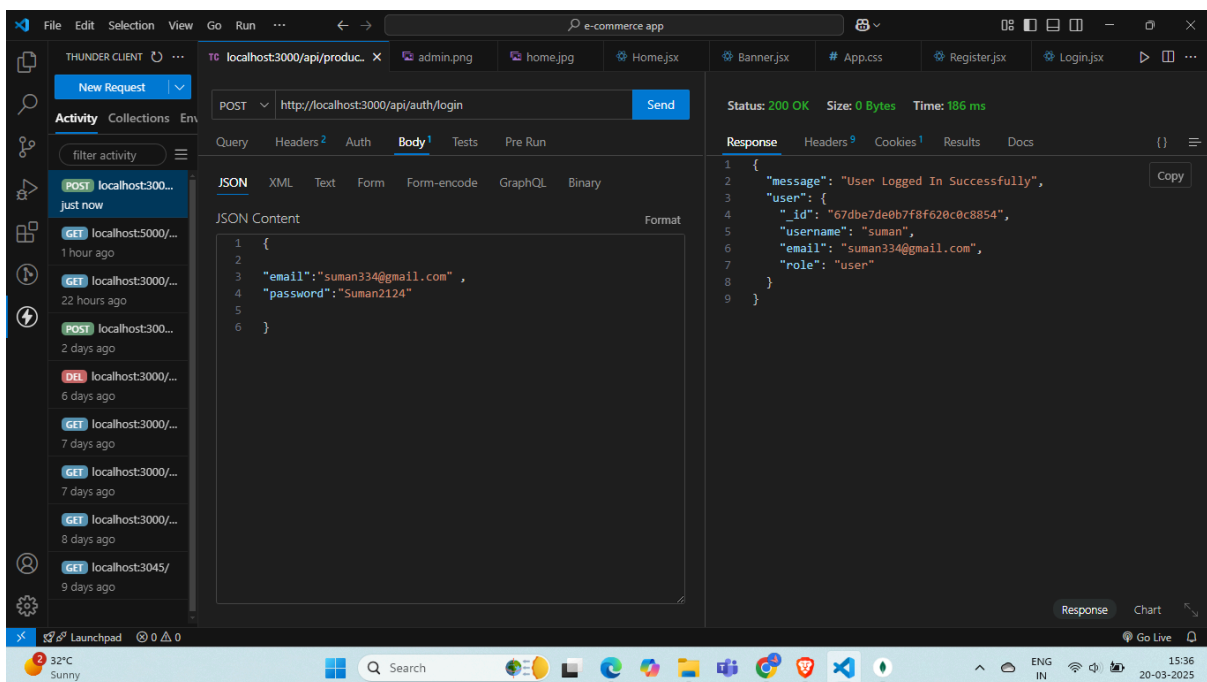


User Login

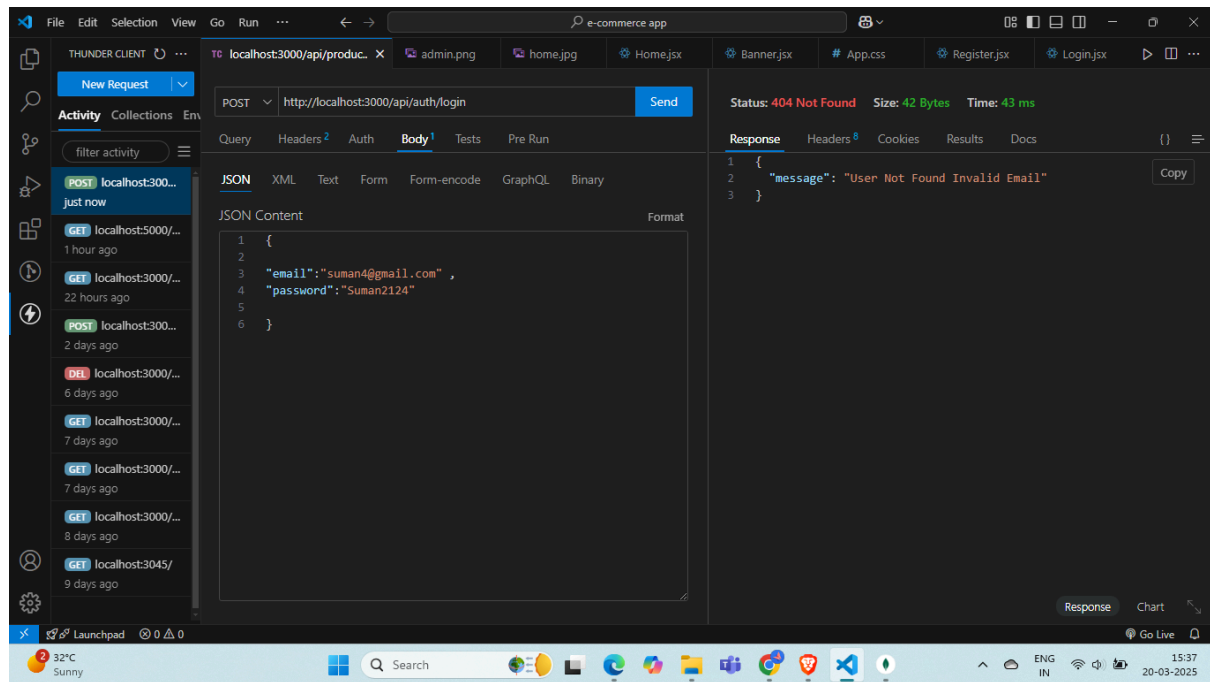
Endpoint: POST localhost:3000/api/auth/login

Expected Response: User logged in successfully, and token stored in cookies.

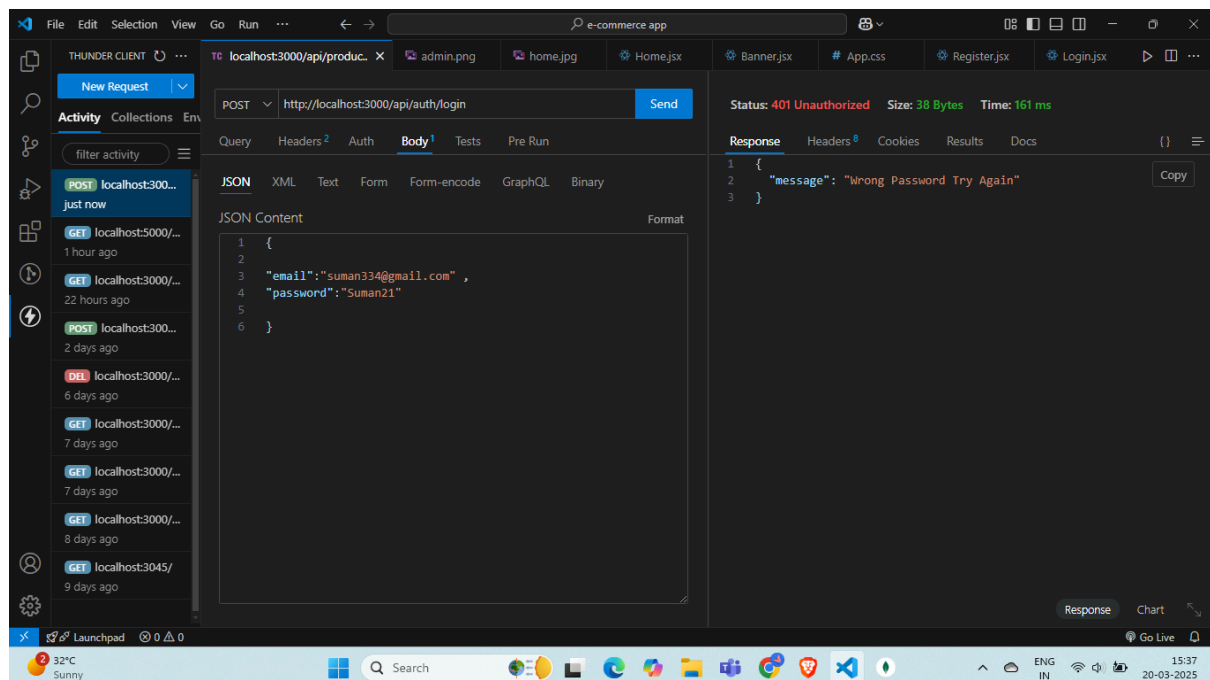
Test Cases: Valid Email & Password: Should return success response with authentication token.



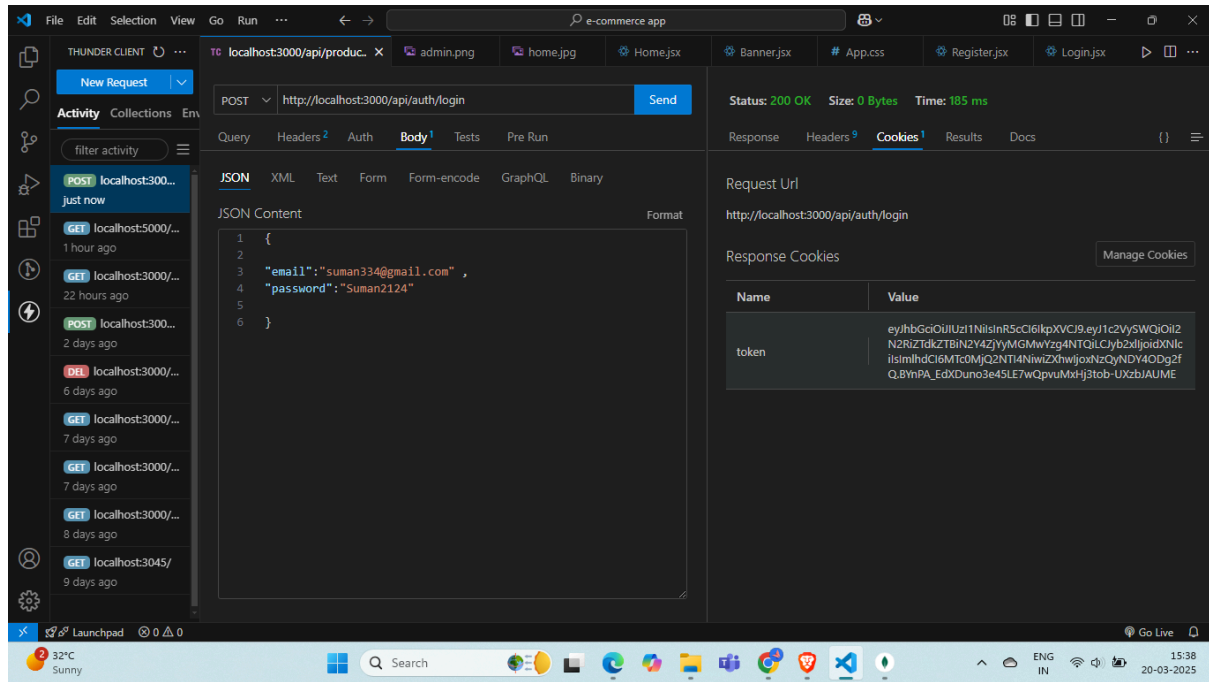
Invalid Email : Should return an error message (e.g., "User Not Found Invalid Email")



Wrong Password: Should return an error message (e.g., "Wrong Password Try Again")



After Successful Login: The token should be present in cookies.



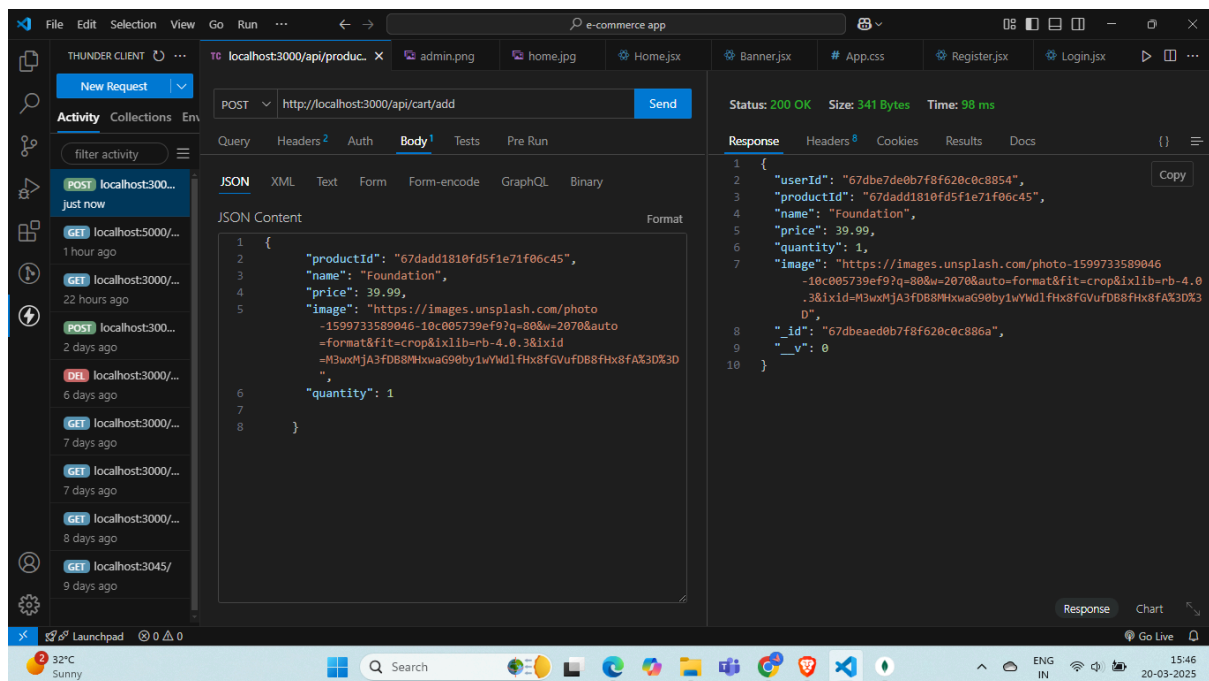
Cart API Testing

Add Product to Cart

Endpoint: POST localhost:3000/api/cart/add

```
Body: { "productId": "67dadd1810fd5f1e71f06c45",  
      "name": "Foundation",  
      "price": 39.99,  
  
      "image": "https://images.unsplash.com/photo-1599733589046-10c005739ef9?q=80&w=20...",  
      "quantity": 1  
}
```

Expected Response: Product added to cart successfully.

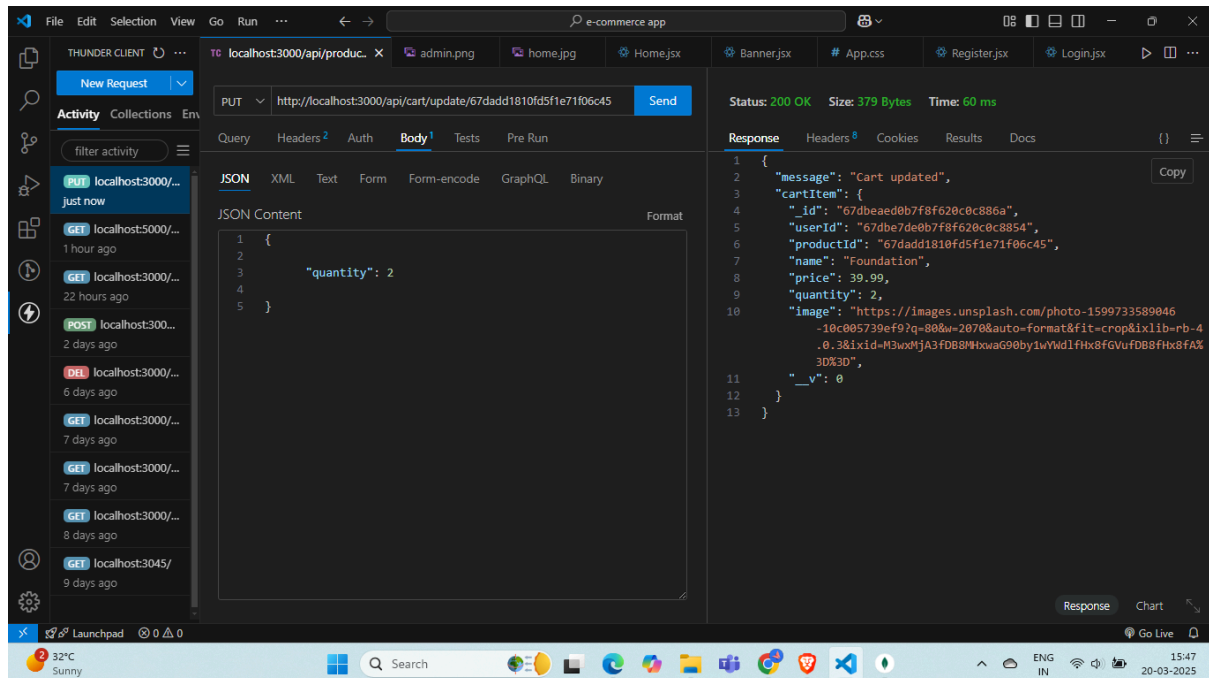


Update Product Quantity in Cart

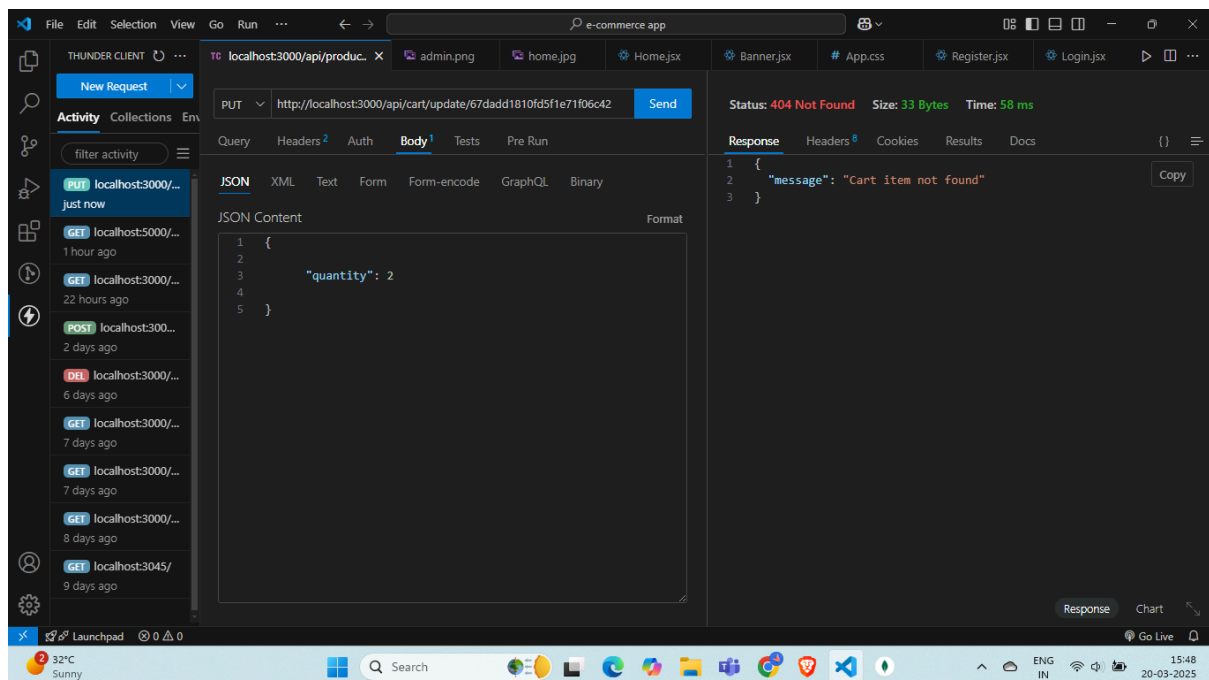
Endpoint: PUT localhost:3000/api/cart/update/67dadd1810fd5f1e71f06c45

Body: { "quantity": 3 }

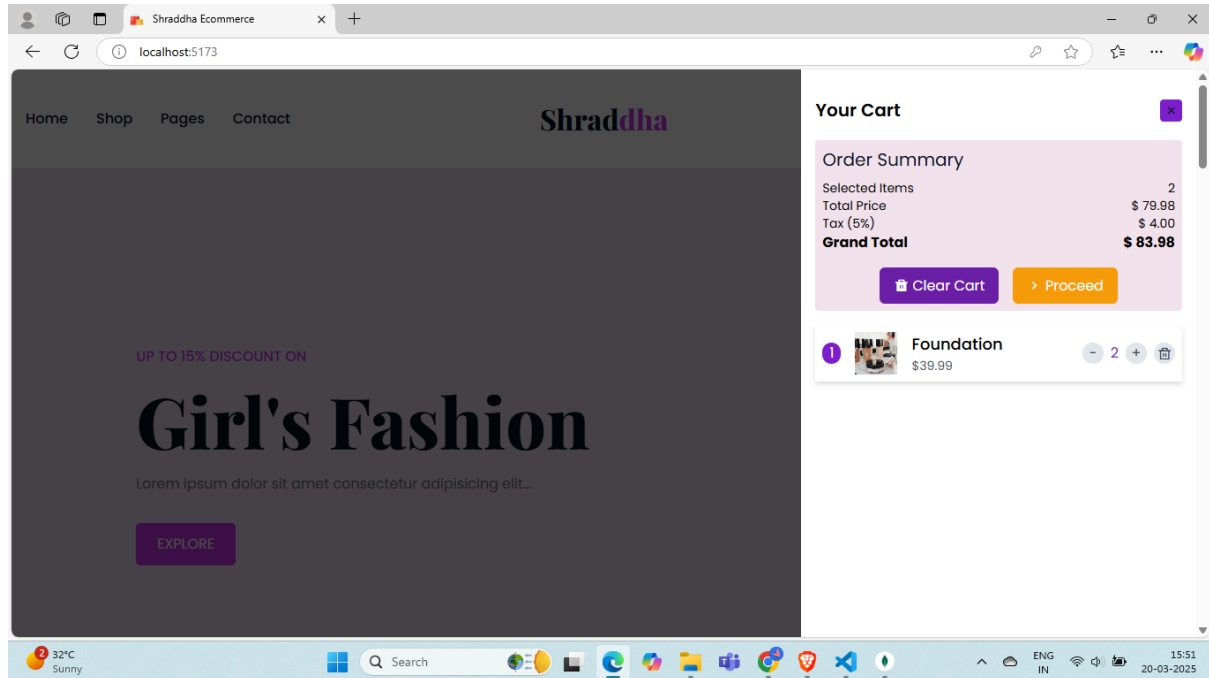
Test Cases: Product exists in cart: Should update the quantity



Product does not exist: Should return an error message (e.g., "Cart Item Not Found").



UI should reflect the changes after update.

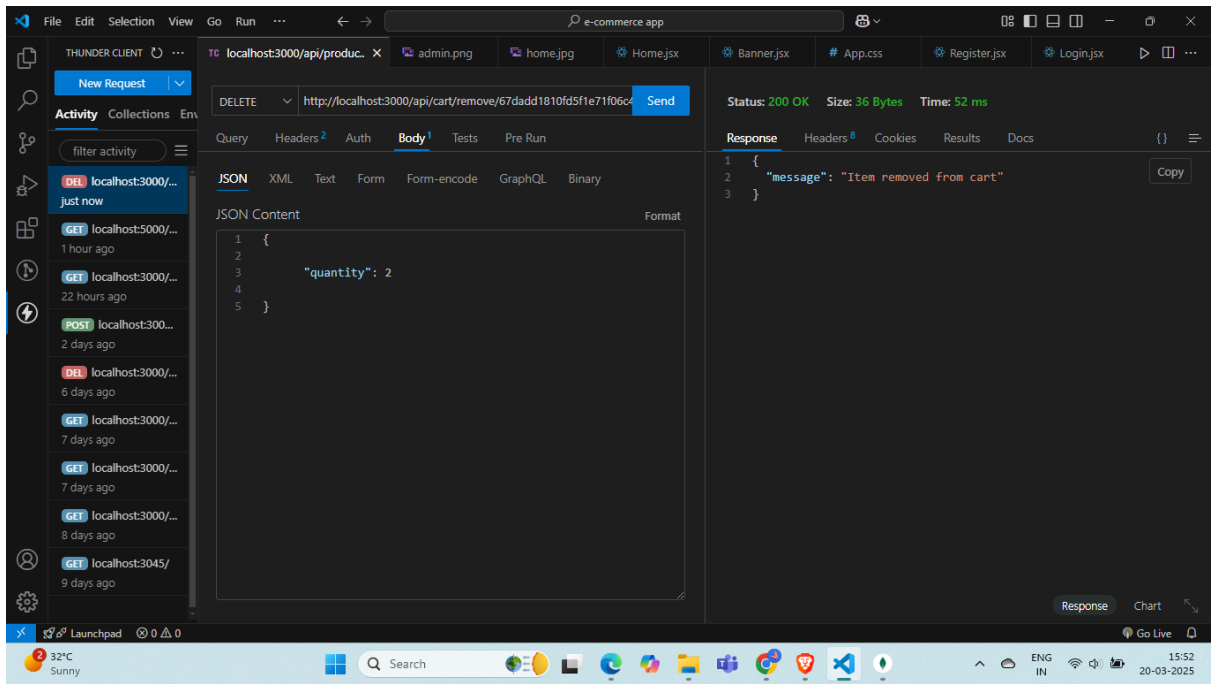


Remove Product from Cart

Endpoint: DELETE localhost: 3000/api/cart/remove/67dadd1810fd5f1e71f06c45

Expected Response: Product removed from cart successfully.

Test Cases: Product exists in cart: Should remove successfully.



UI should reflect the changes after update.

