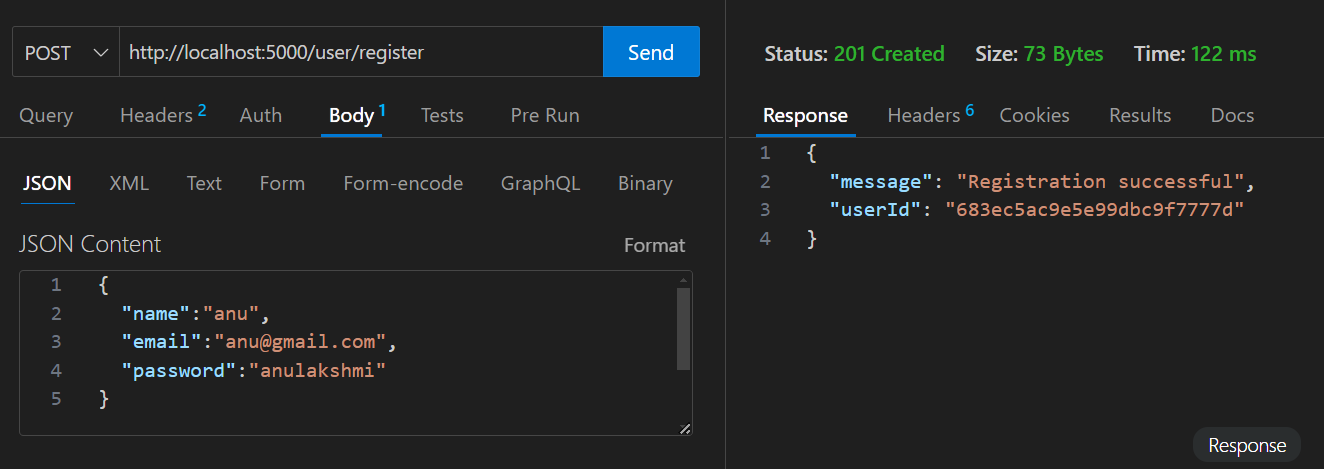
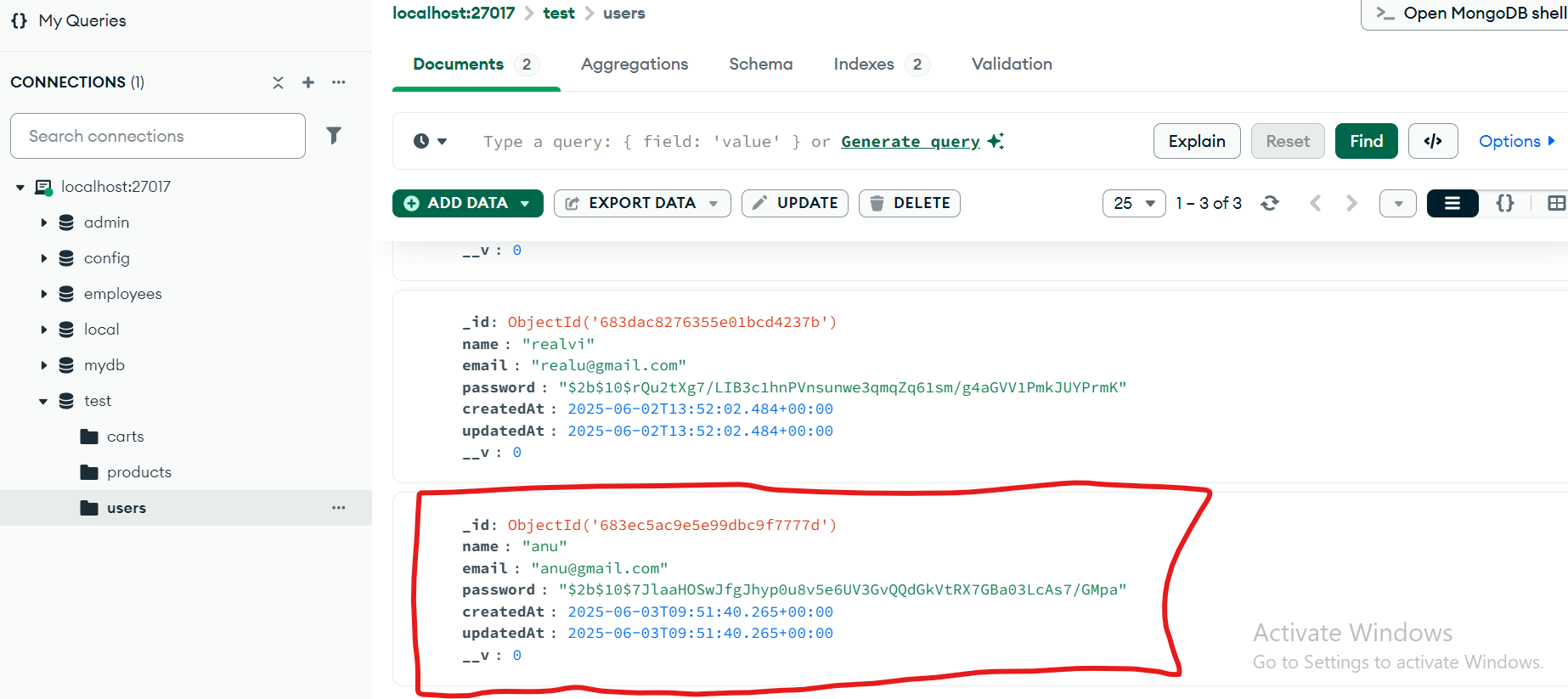
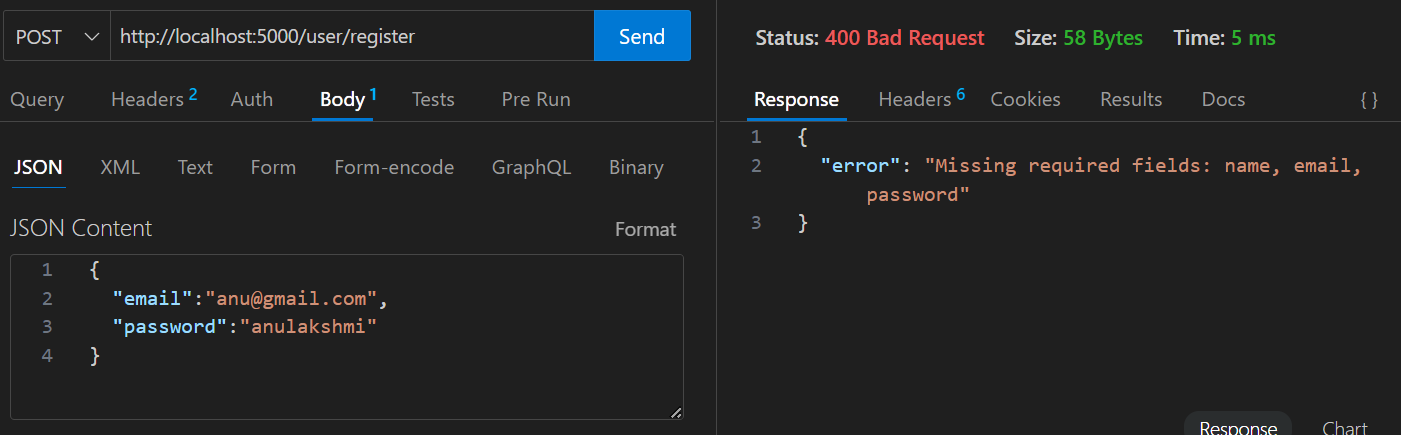
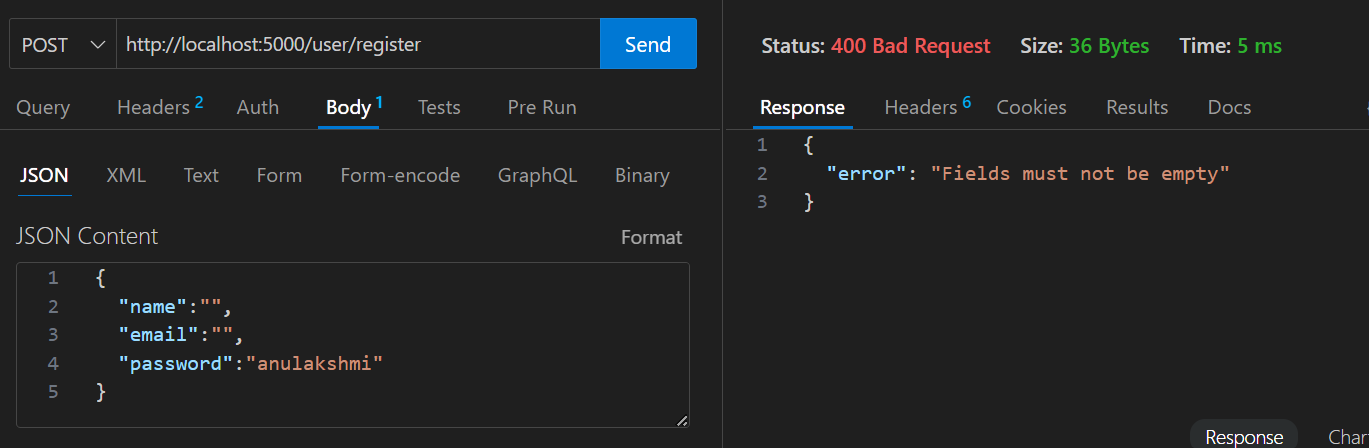
**API Testing Documentation using Thunder Client and Mongodb.  
  
  
1. User Authentication  
  
 i) POST /user/register   
 Description:** Registers a new user

**Scenario A**: All valid fields

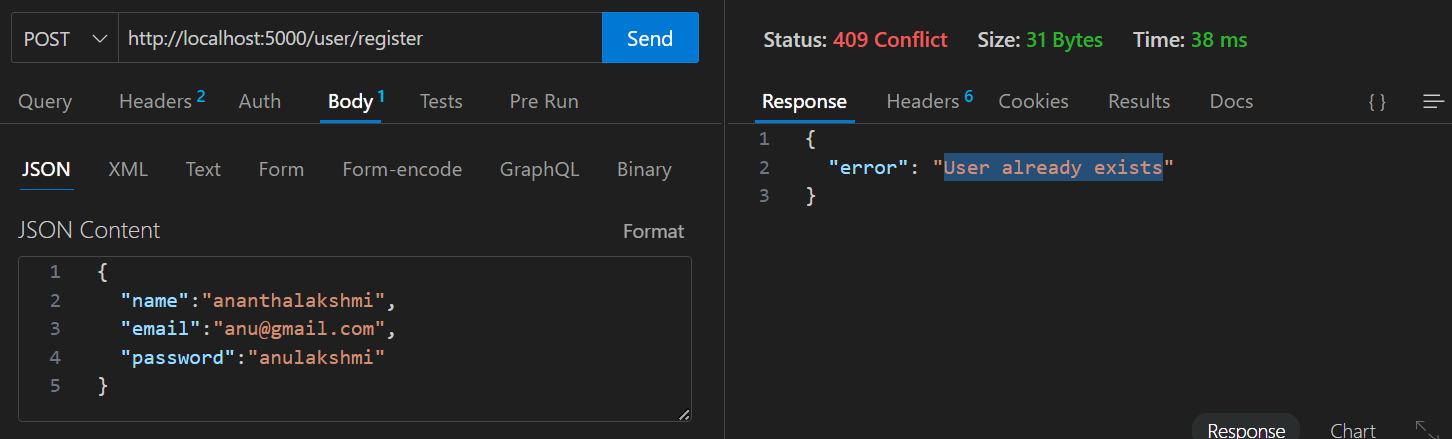
**Expected Result**: Status 201 Created with Newly created user id and message Registration successful, user saved in MongoDB.

  
  
  
  
**Scenario B:** Missing required fields (e.g. no name)

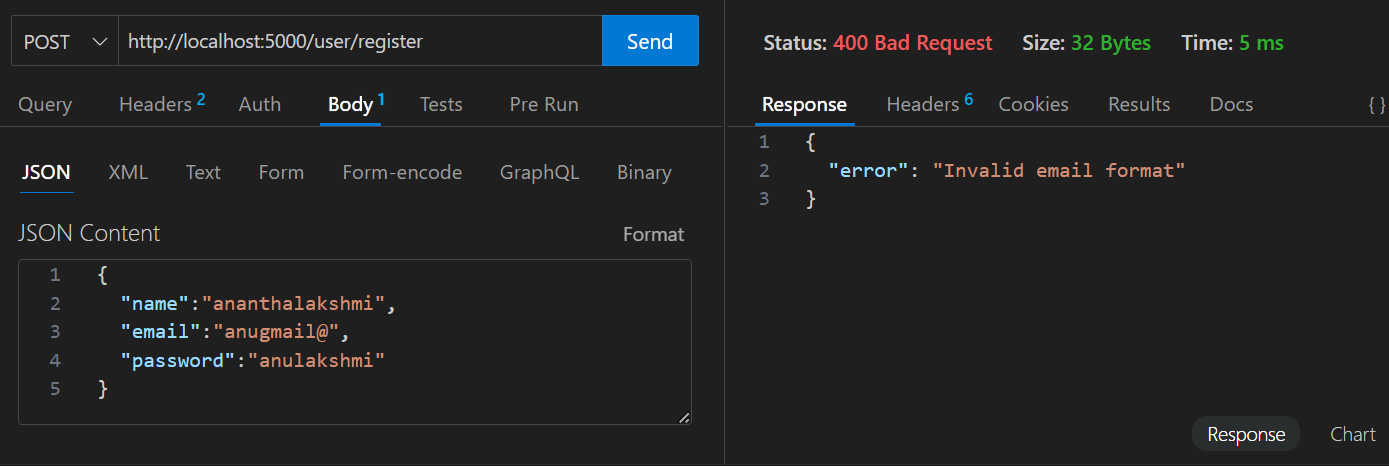
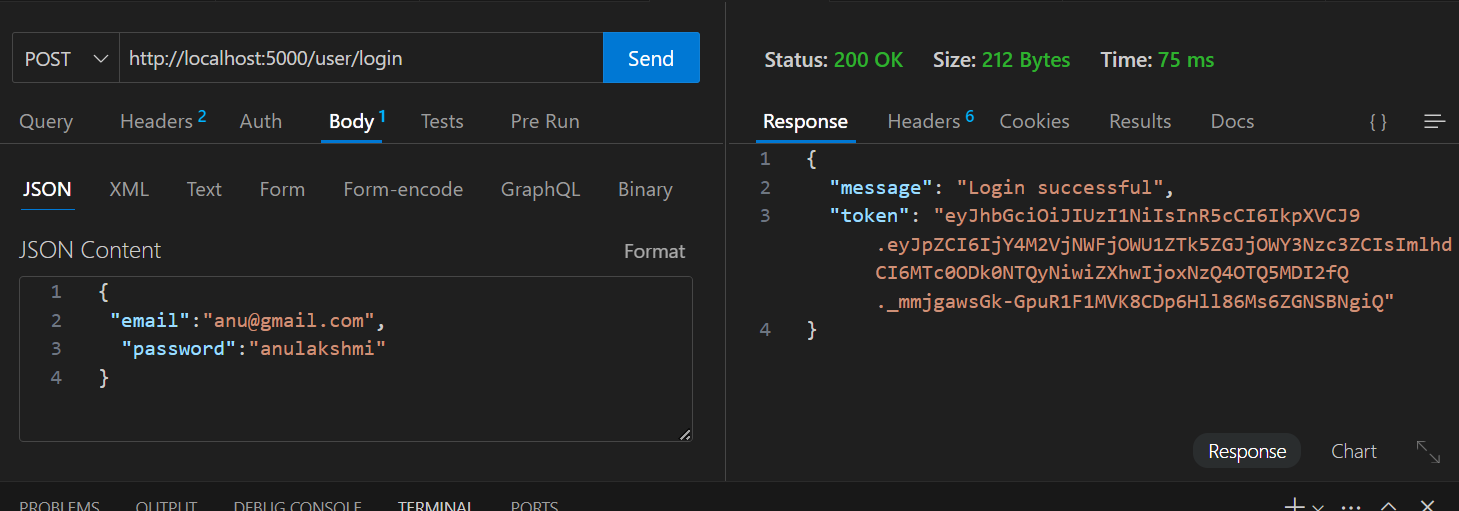
**Expected Result**: 400 Bad Request, error: Missing required fields: name, email, password  
  
  
  
  
  
  
**Scenario C:** Empty fields

**Expected Result**: 400 Bad Request, error: Fields must not be empty  
  
  


**Scenario D:** Duplicate Email Registration

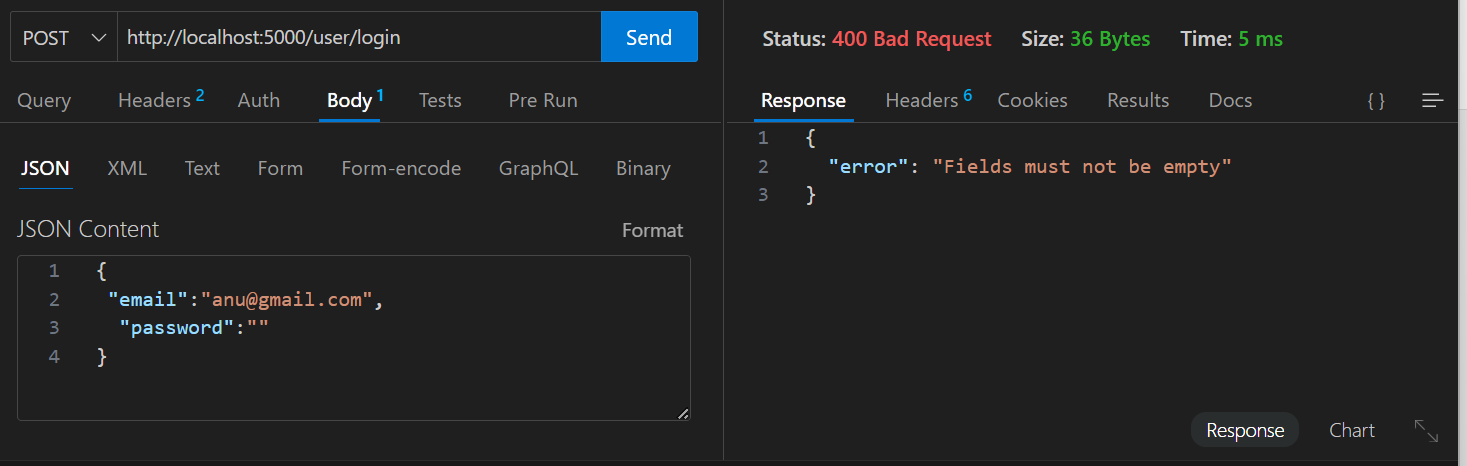
**Expected Result**: 409 Conflict, error: " User already exists  
  
  


**Scenario E**: Invalid Email Format  
**Expected Result**: 400 Bad Request, error: "Invalid email format"

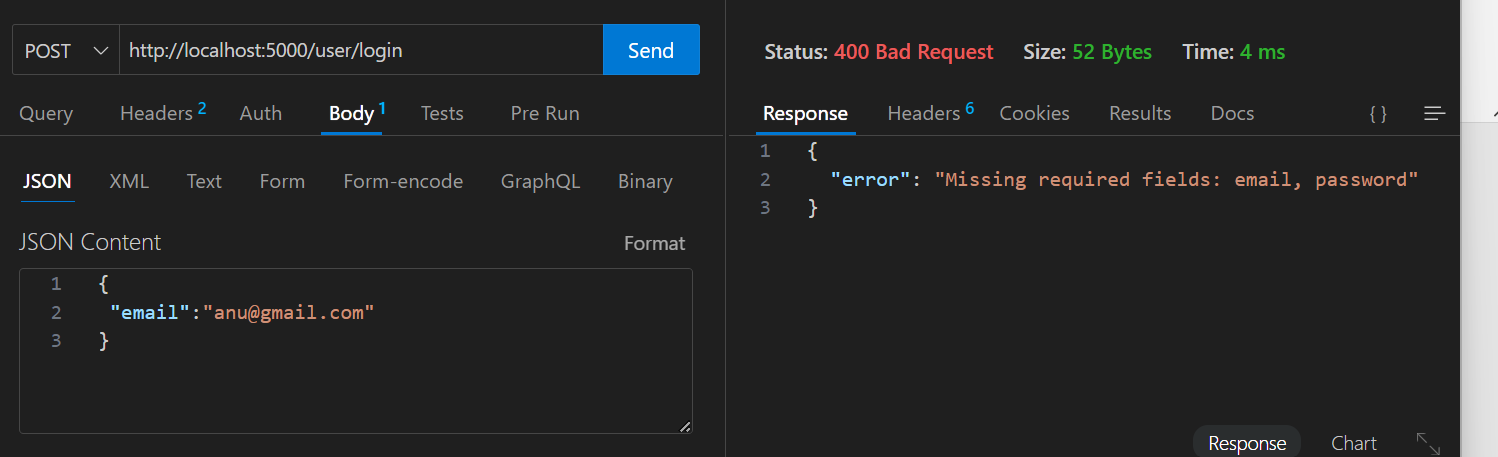
  
  
  
  
  
  
  
  
**II) POST /user/login  
  
Description**: Logs in the user and returns a JWT token.  
  
**Scenario A**: Valid email and password  
**Expected Result**: status 200 OK, JSON with token  
  
  
  
  
**Scenario B:** Invalid password or unregistered email

**Expected Result**: status 401 Unauthorized, error message “Invalid credentials”  
  
  


**Scenario C**: Empty email/password  
**Expected**: 400 Bad Request, validation error "Fields must not be empty"

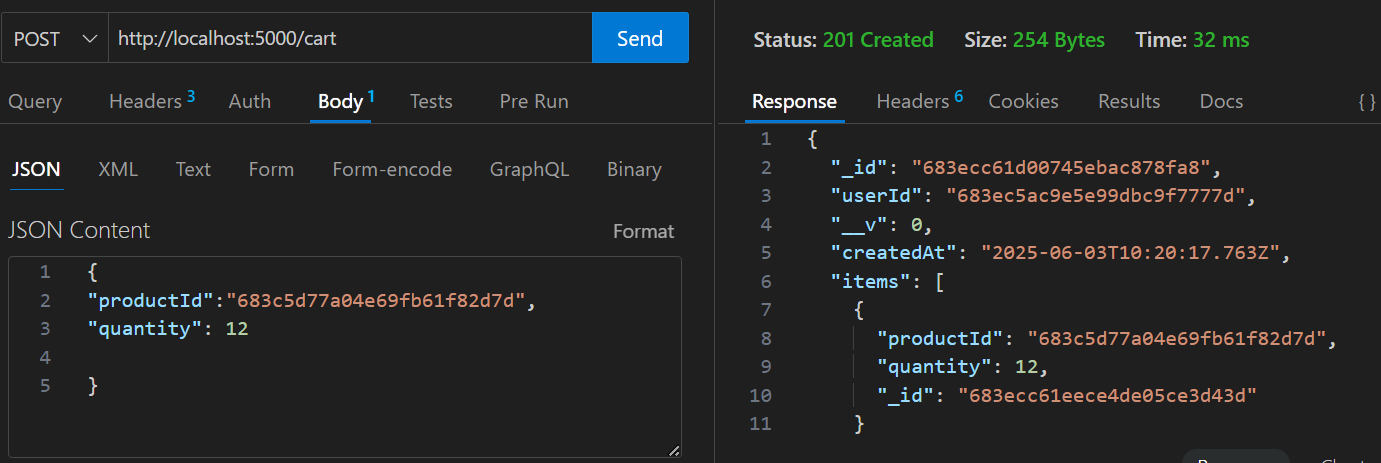
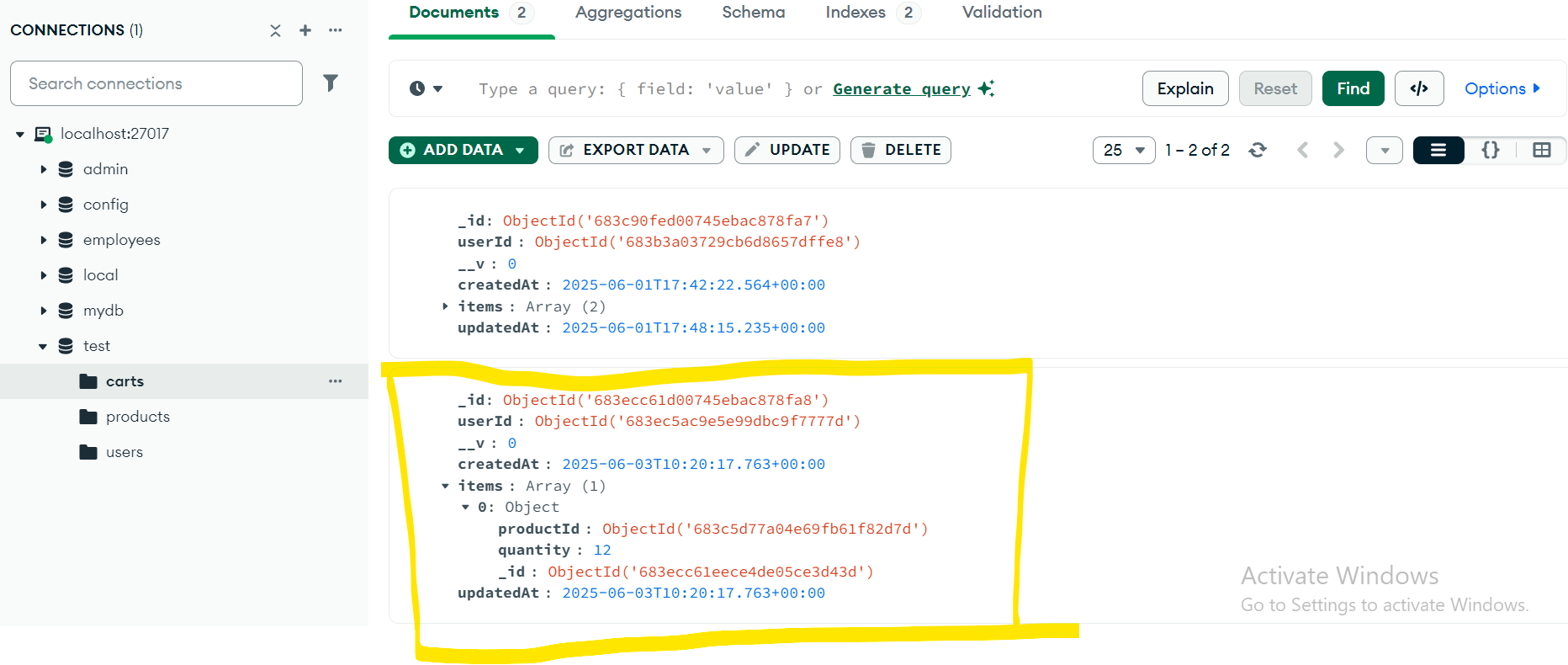


**Scenario D**: Missing required fields  
**Expected Result**: 400 Bad Request, error message "Missing required fields: email, password"

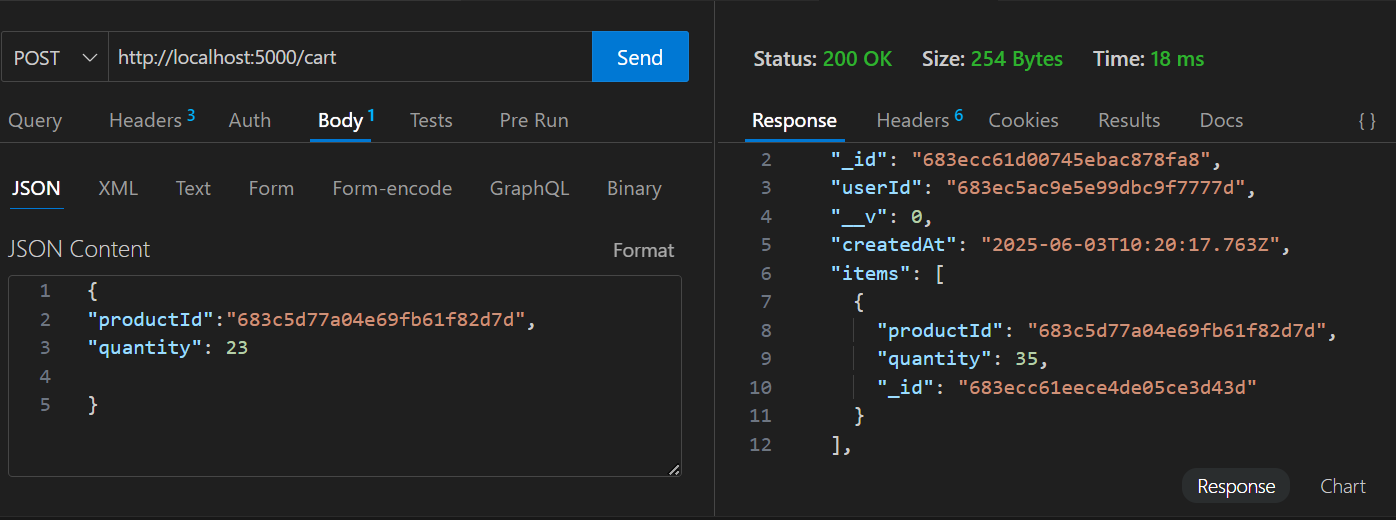


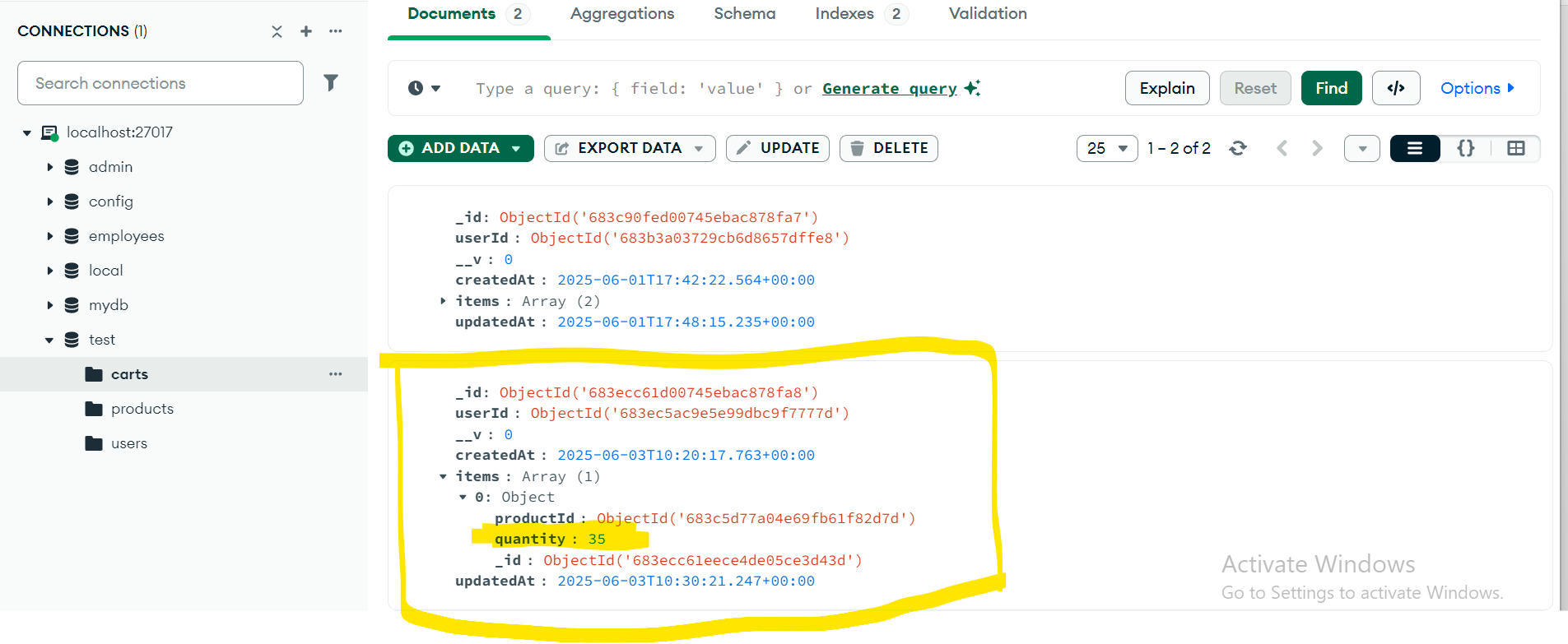
**2. Cart Management  
  
i)** **POST /cart  
Description:** Add item to user's cart.  
  
**Scenario A:** Valid productId and quantity if Product does not exist in the cart

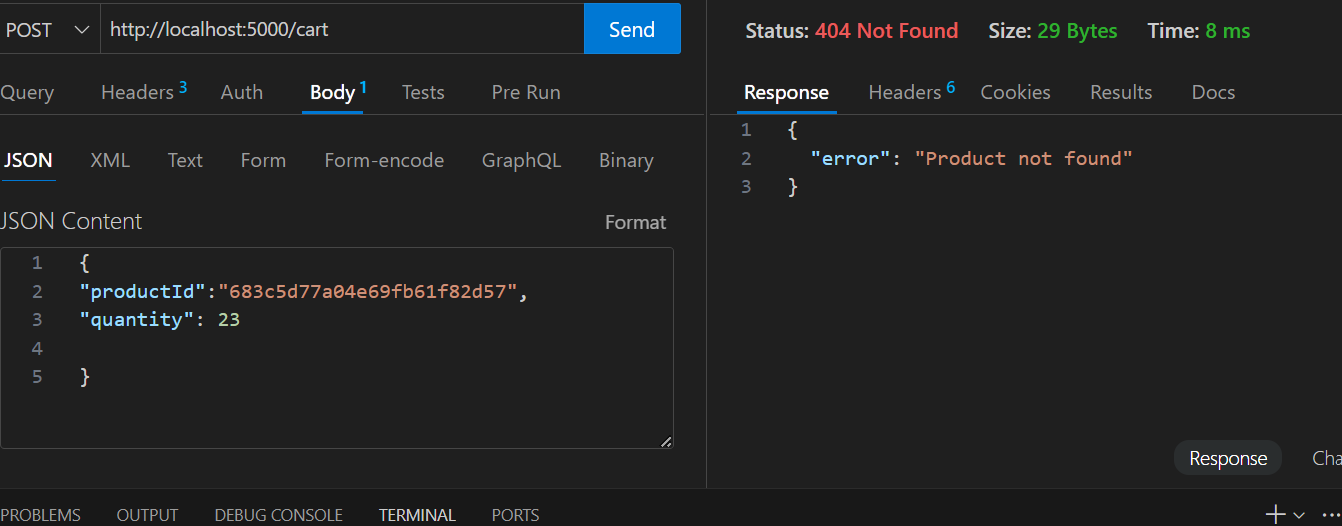
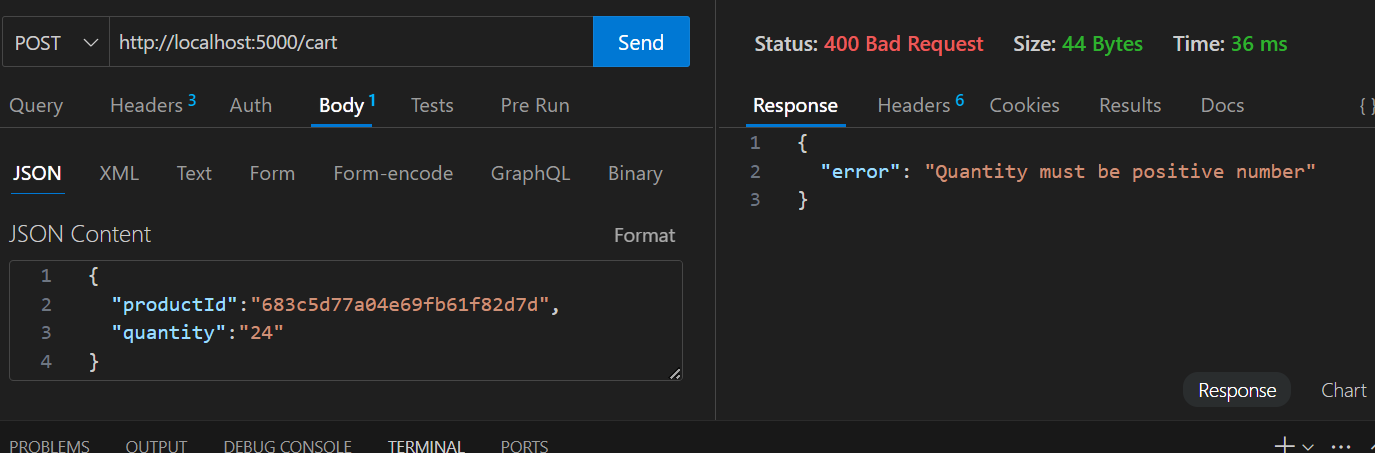
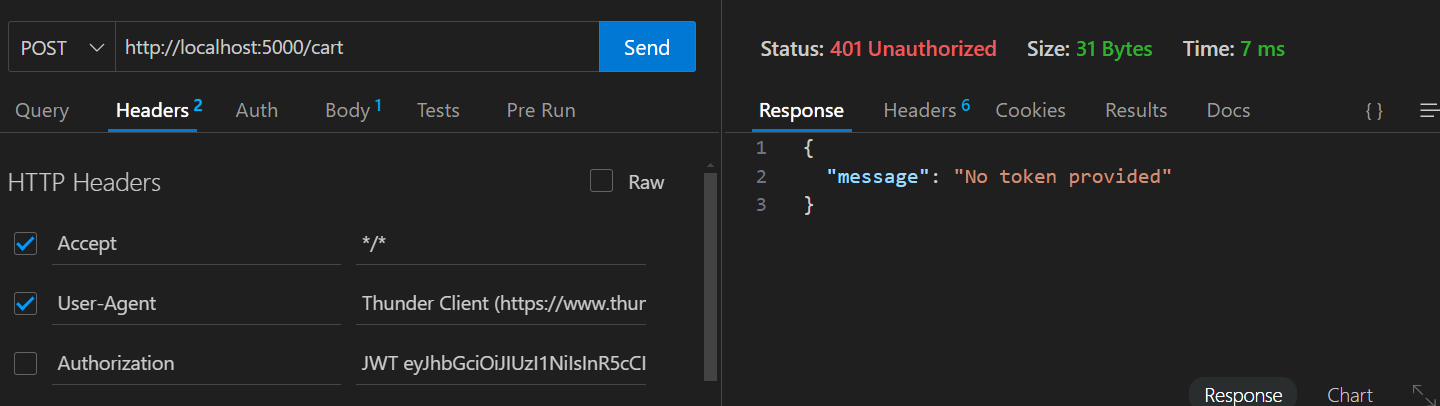
**Expected Result**: 201 Created, new cart item is created with the specified quantity.

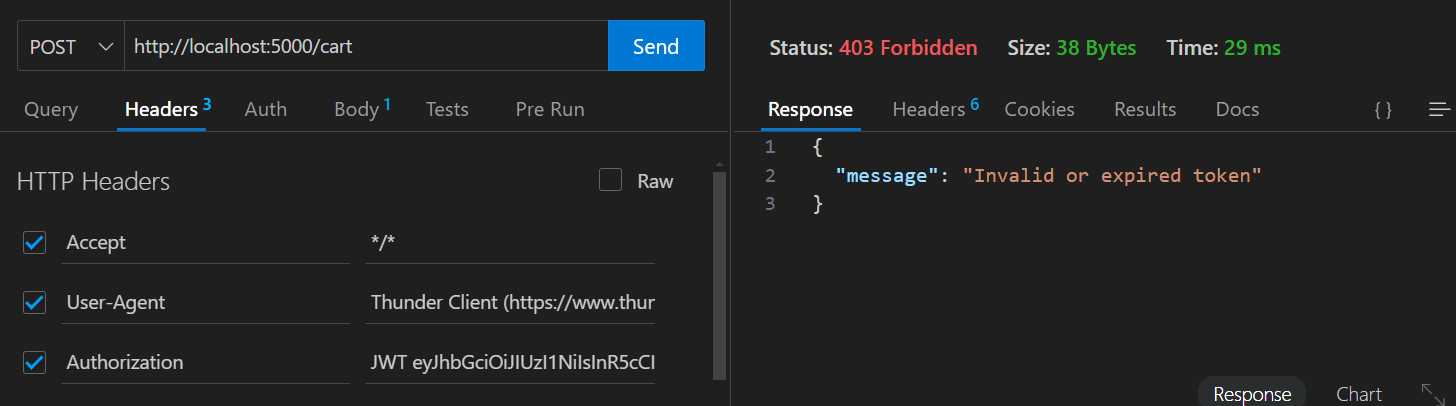
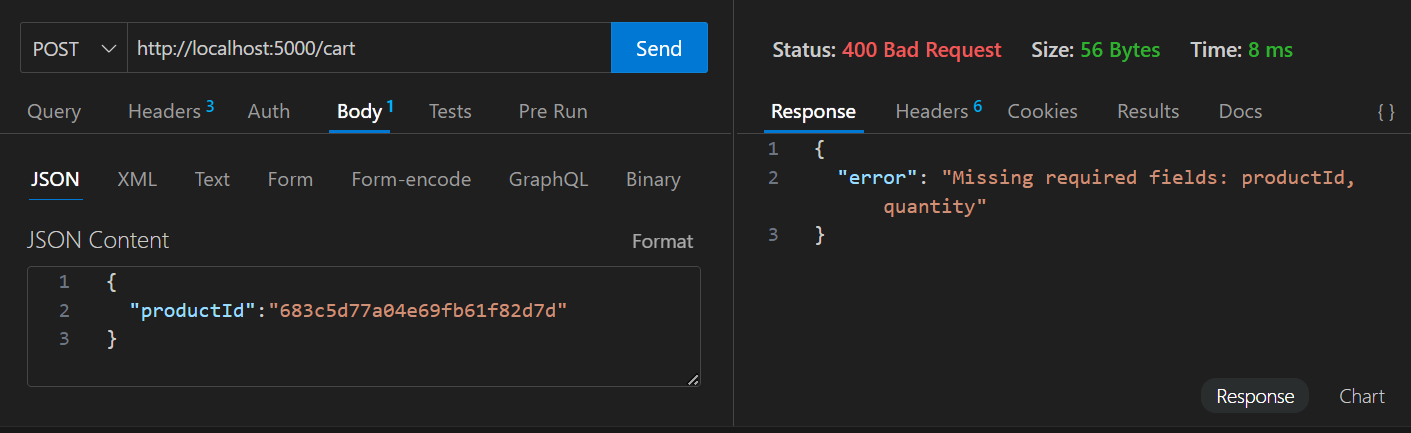
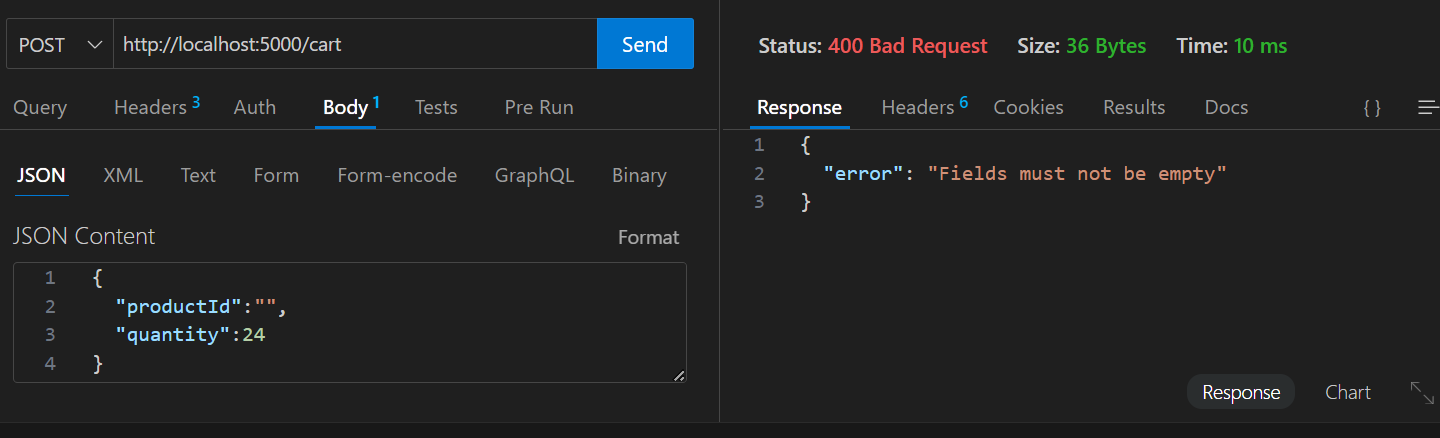
  
  
  
  
  
  
  
  
**Scenario B**: Valid productId and quantity and Product already exists in the cart

**Expected Result**: 200 OK, existing cart item's quantity is updated by adding the new quantity



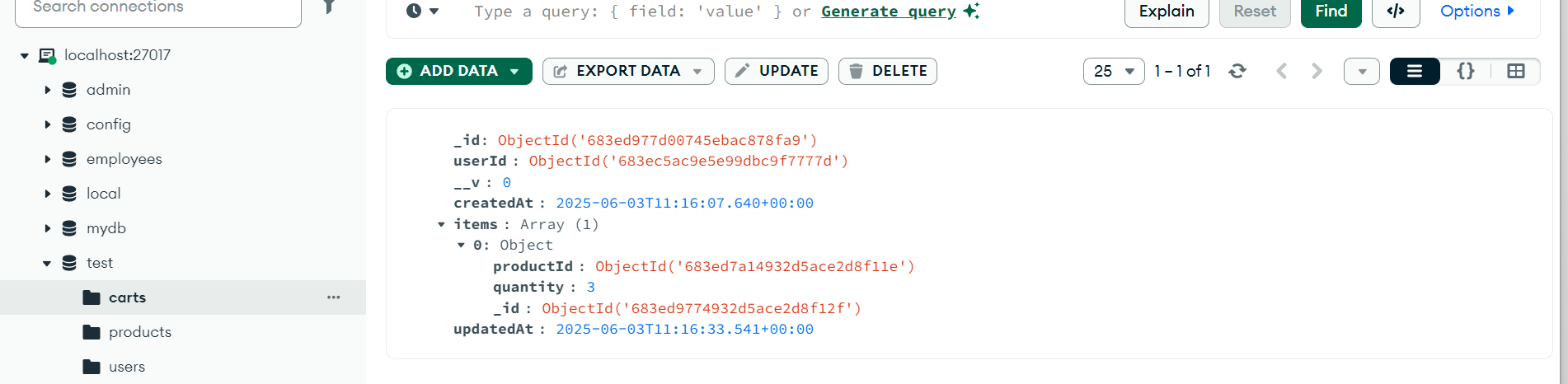
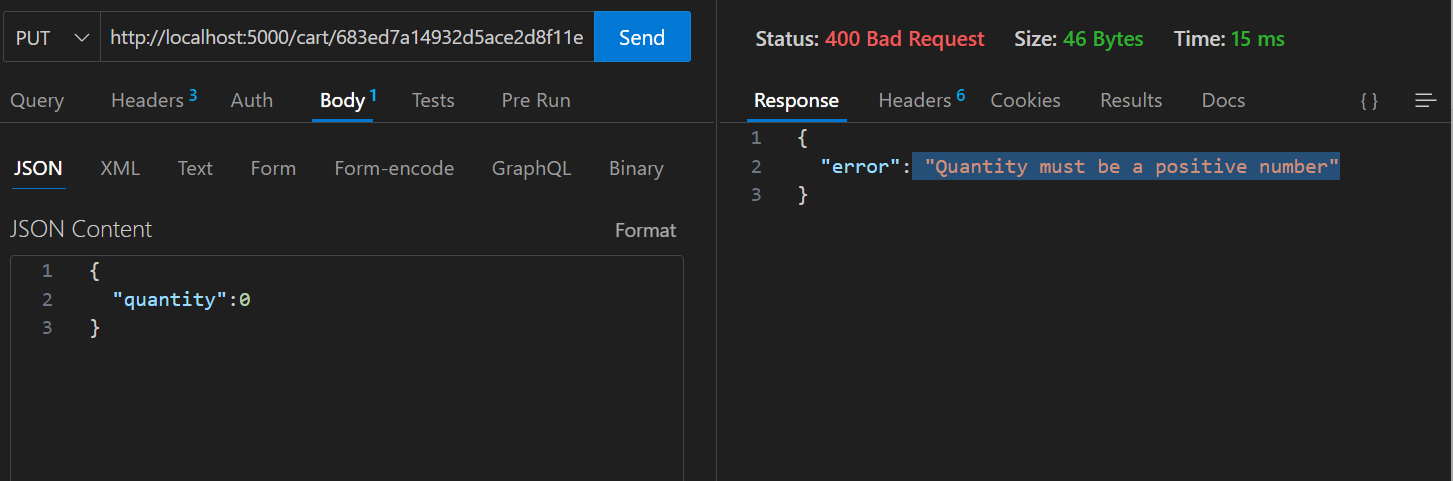
  
  
  
 **Scenario C**: Invalid productId (product does not exist in products collection)  
**Expected**: 404 Not Found and error "Product not found"

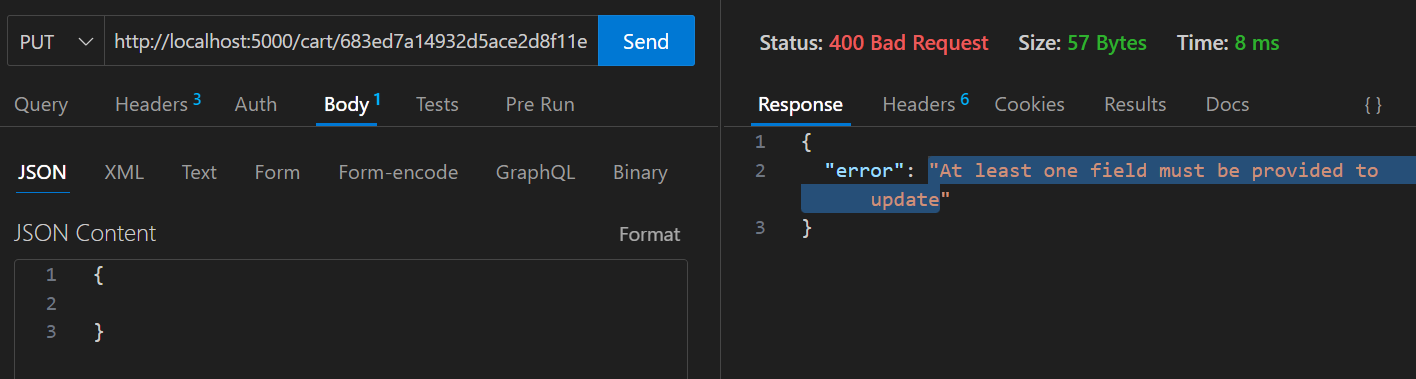
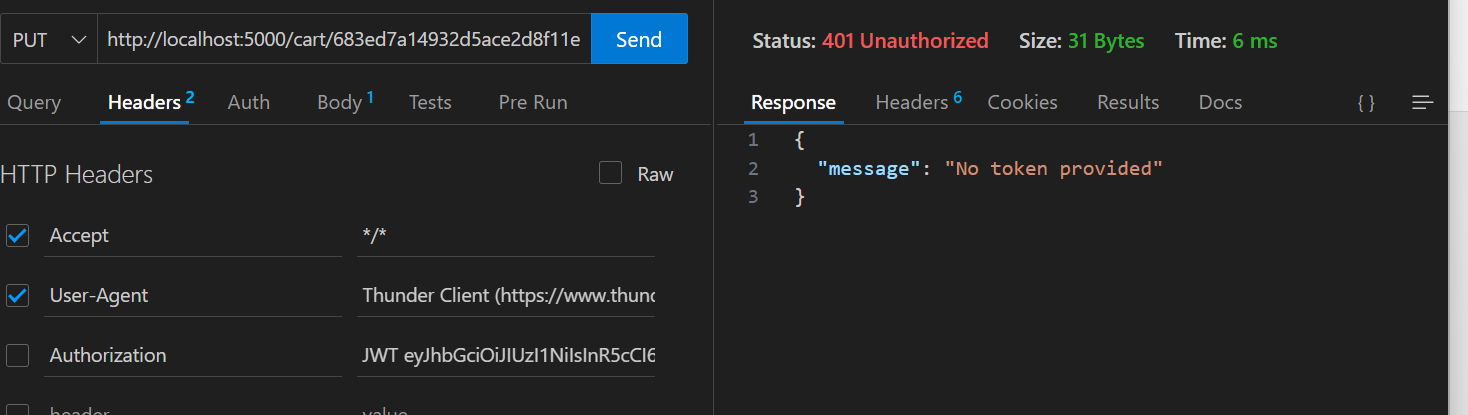
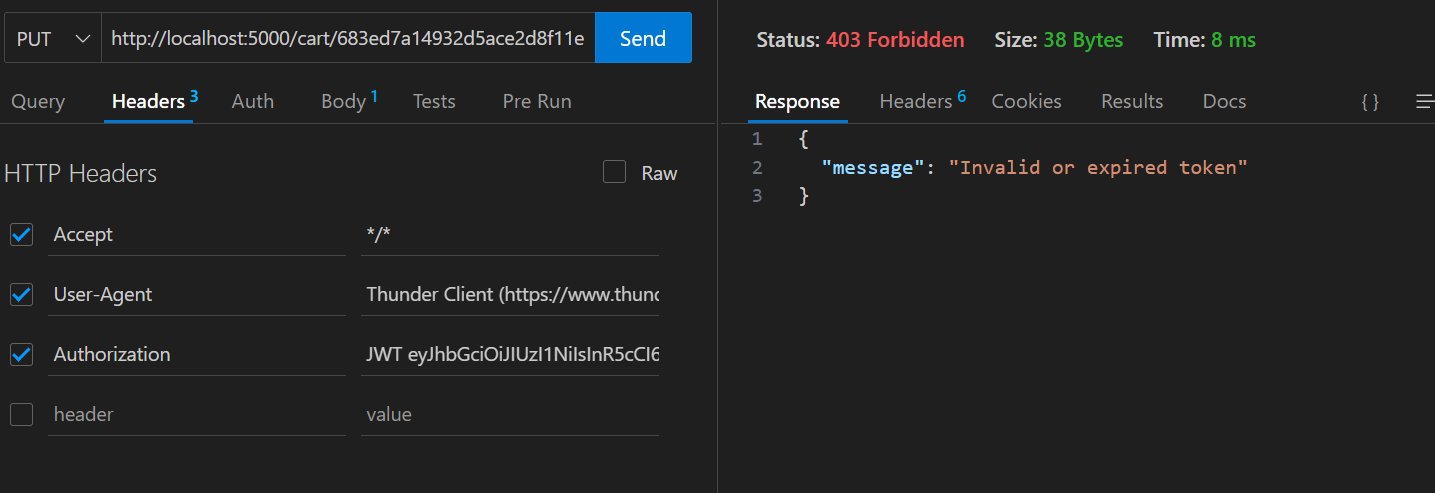
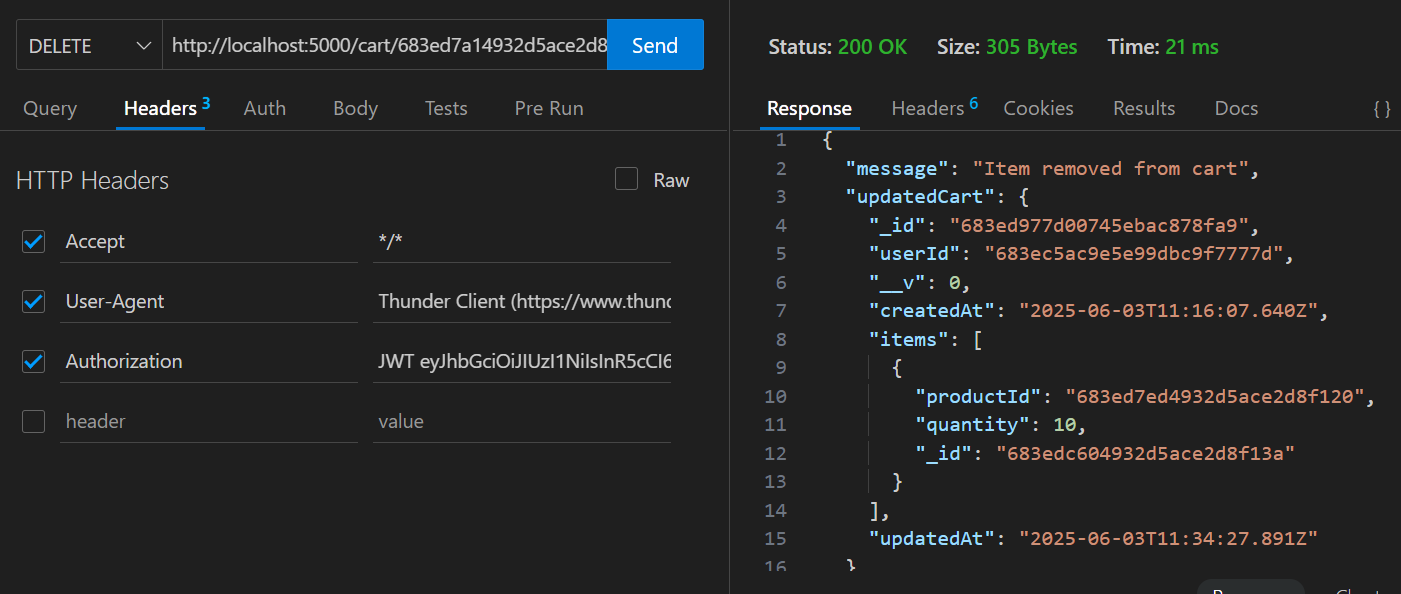
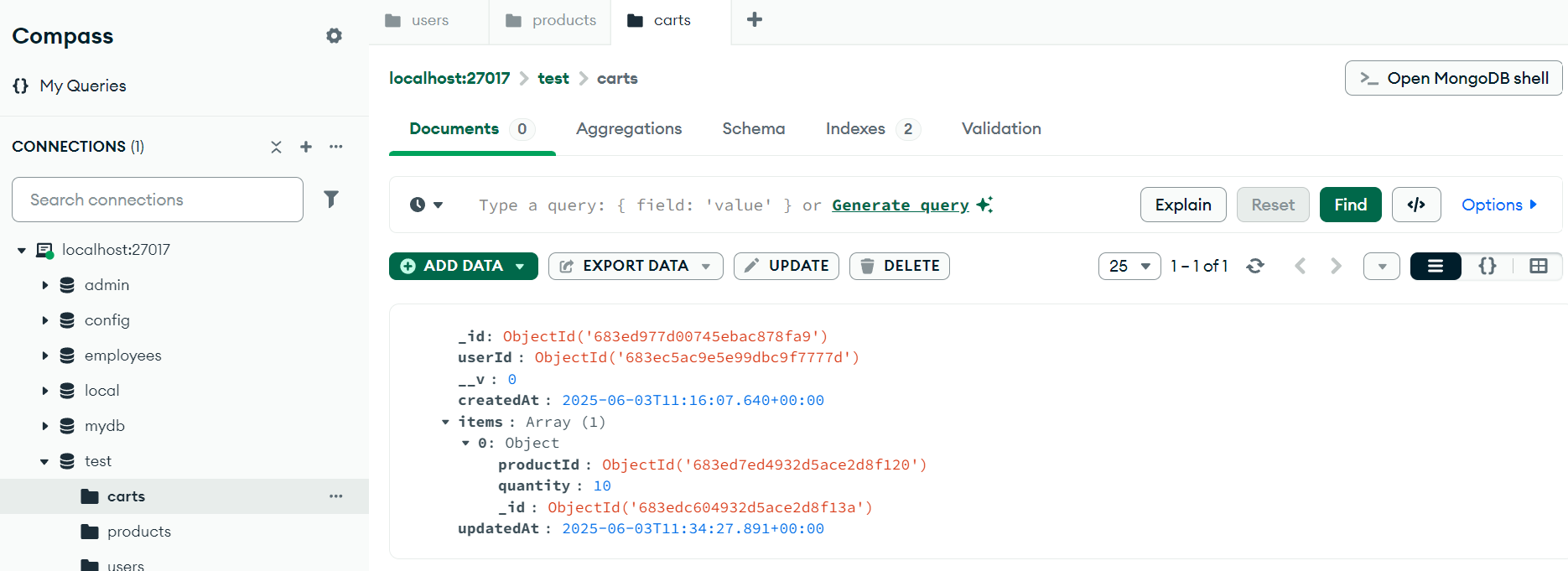
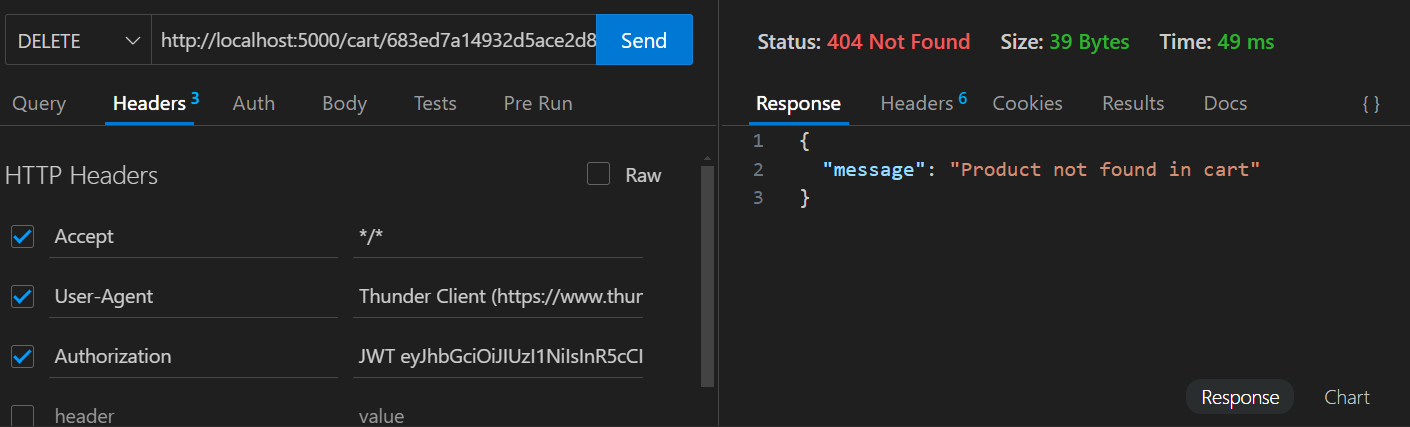
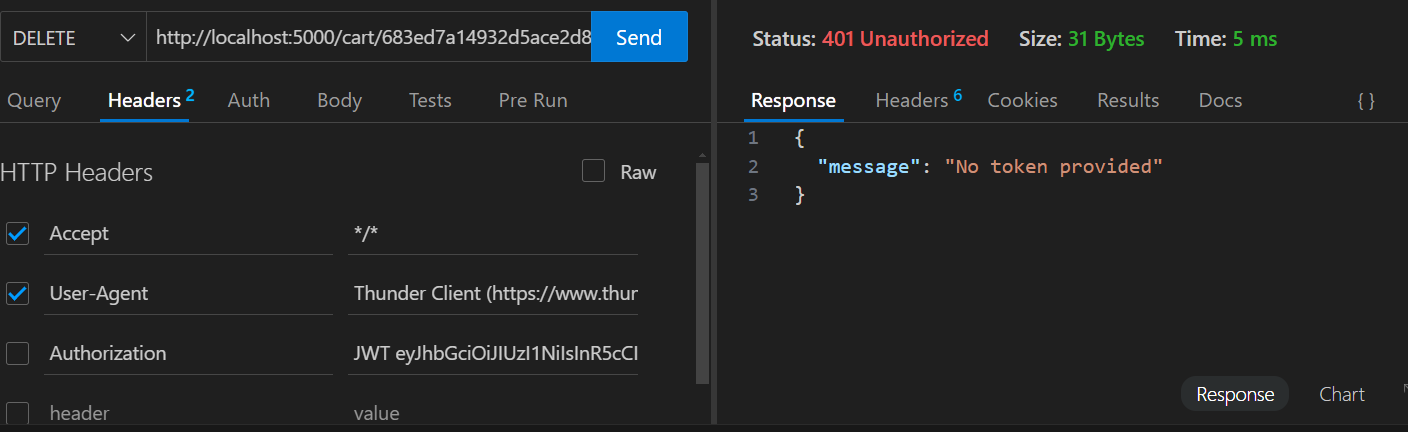
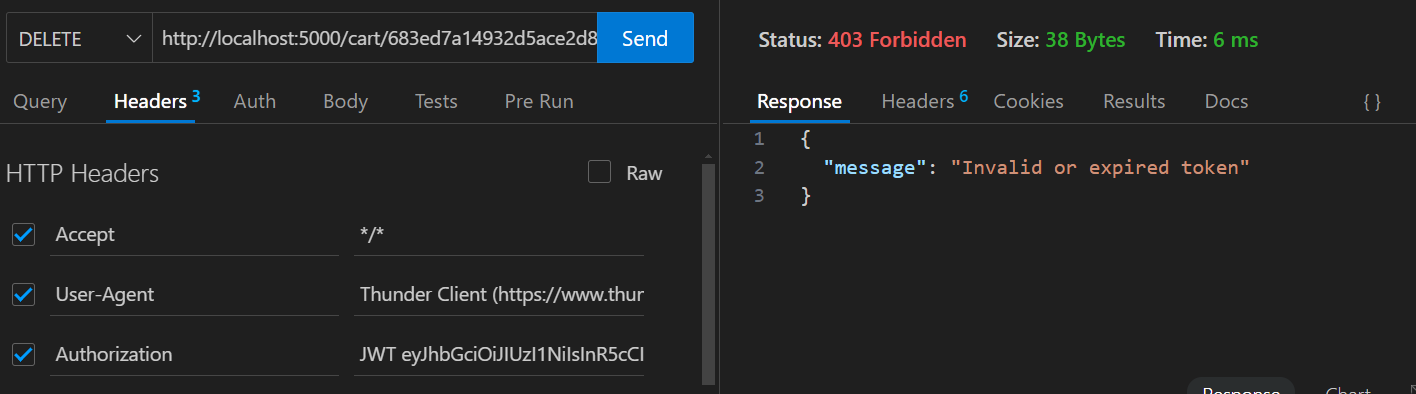
  
  
**Scenario D**: Invalid quantity (quantity <=0)  
**Expected Result**: 400 Bad Request and error " Quantity must be positive number"  
  
  
**Scenario E**: Missing JWT Token (The client does not send an Authorization header)  
**Expected Result:** status 401 Unauthorized and Error message: "No token provided"  


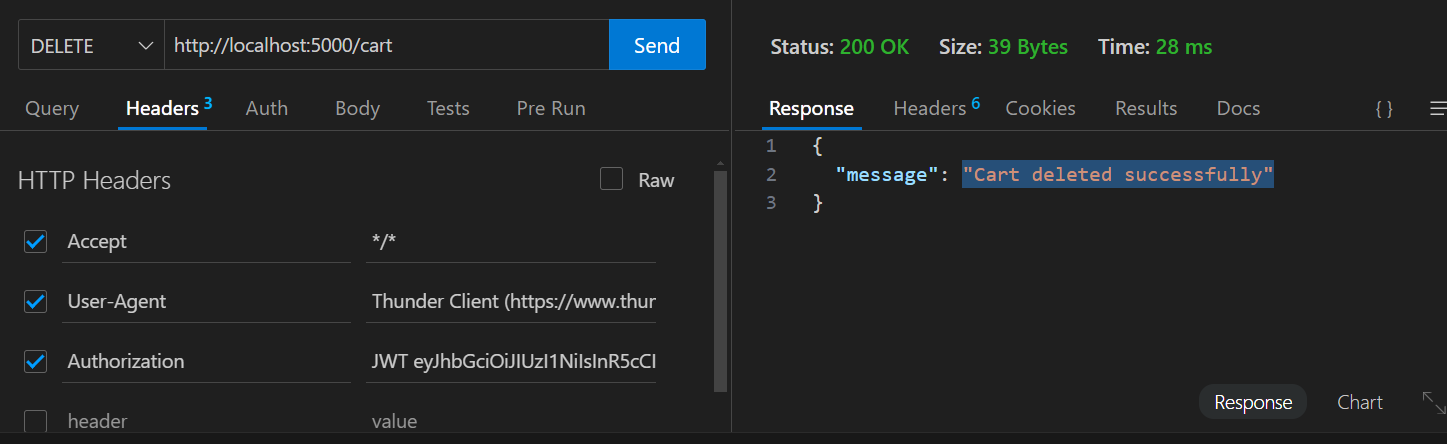
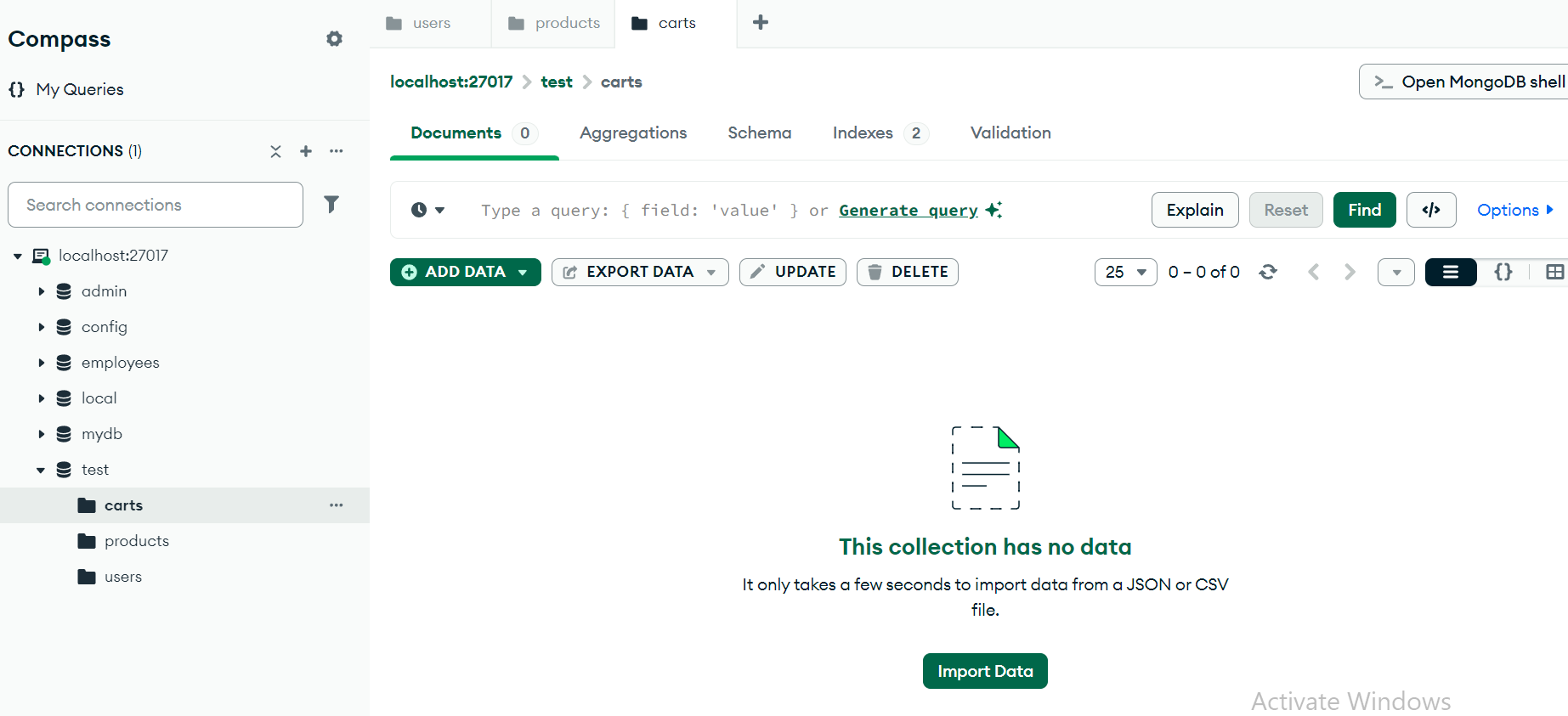
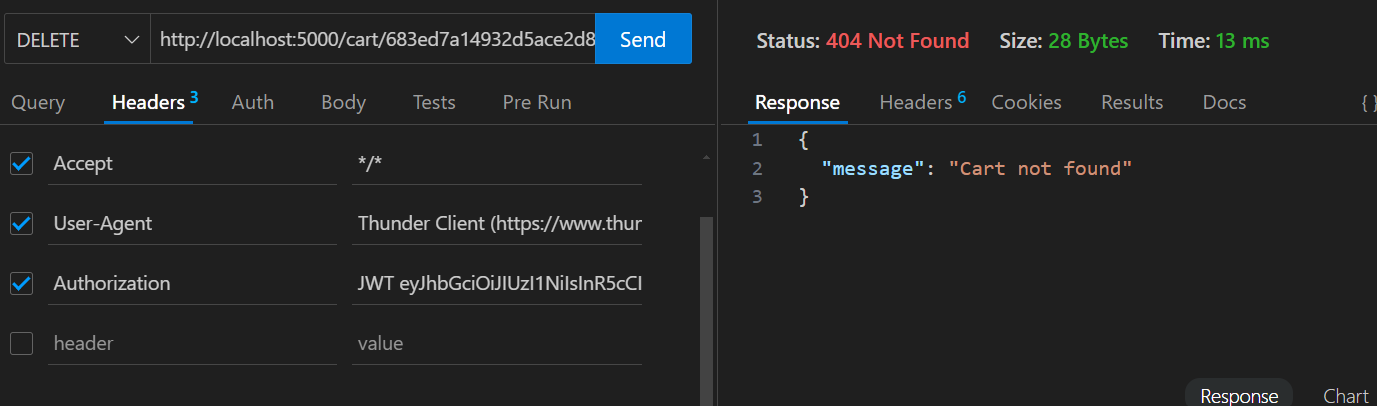
**Scenario F**: Invalid or Expired JWT Token  
**Expected Result:** status 403 Forbidden and Error message: " Invalid or expired token"  
  
  
**Scenario G:** Missing fields  
**Expected Result**: 400 Bad Request, error: Missing required fields  
  
  
**Scenario H:** Empty string or null values **Expected Result**: 400 Bad Request, error: Fields must not be empty  
  
  


**ii)** **PUT /cart/:id**

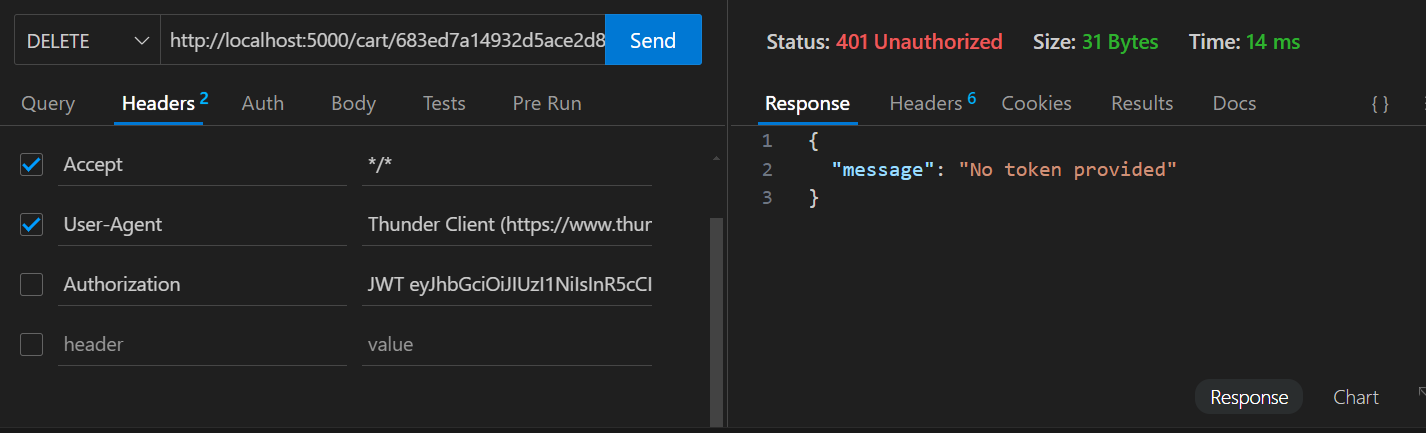
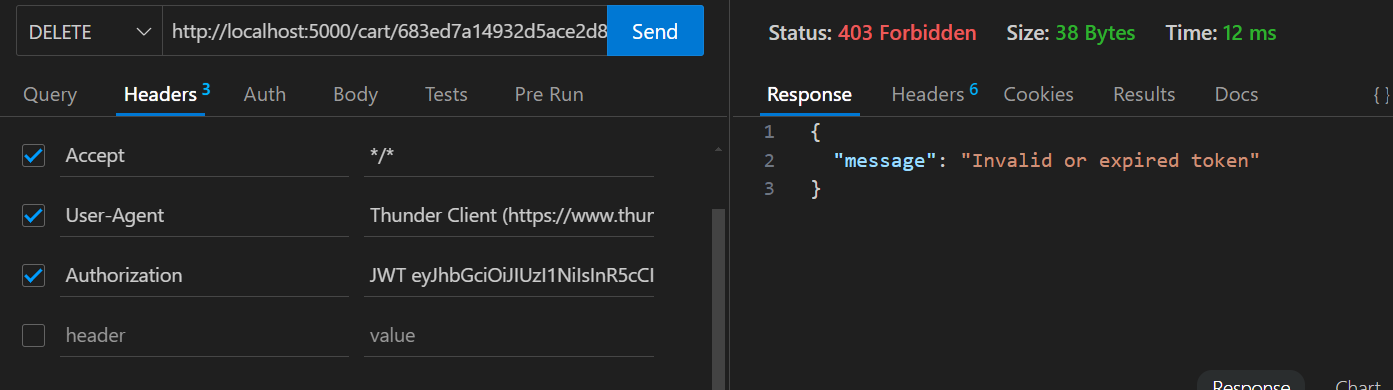
**Description**: Update quantity of a cart item.  
  
**Scenario A**: Valid productid and quantity update

**Expected:** status 200 OK, cart item updated and return updated cart.  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
**Scenario B:** Product (cart item) does not exist  
**Expected:** 404 Not Found, error: "Product not found  
  
  
  
  
  
**Scenario C:** Quantity is 0  
**Expected Result**: status 400 Bad Request and error "Quantity must be a positive number"  
  
  
**Scenario D:** No fields sent in request body

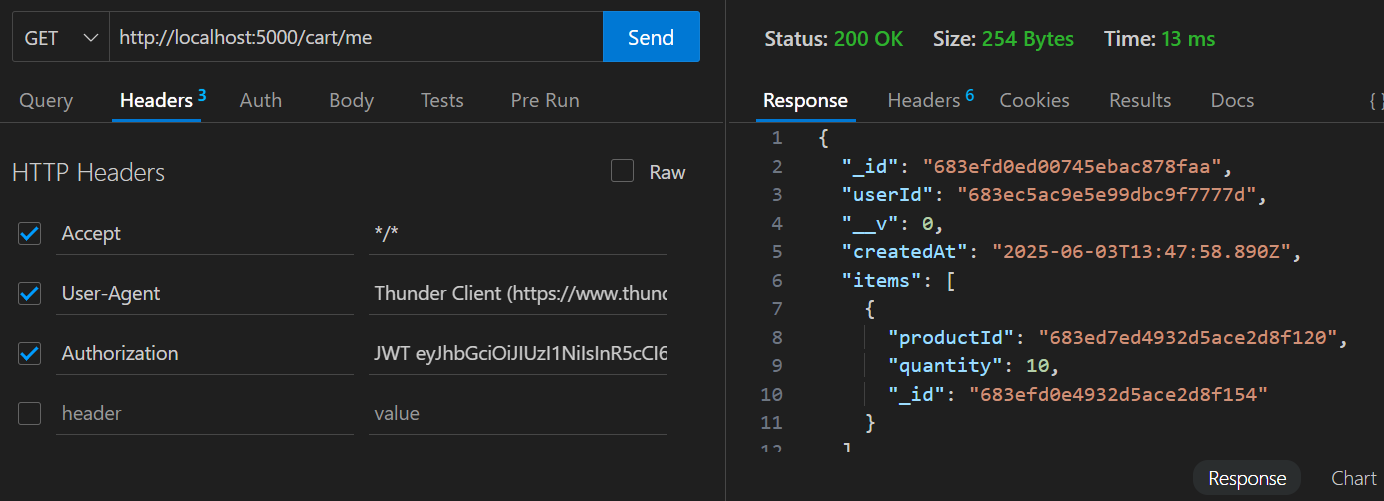
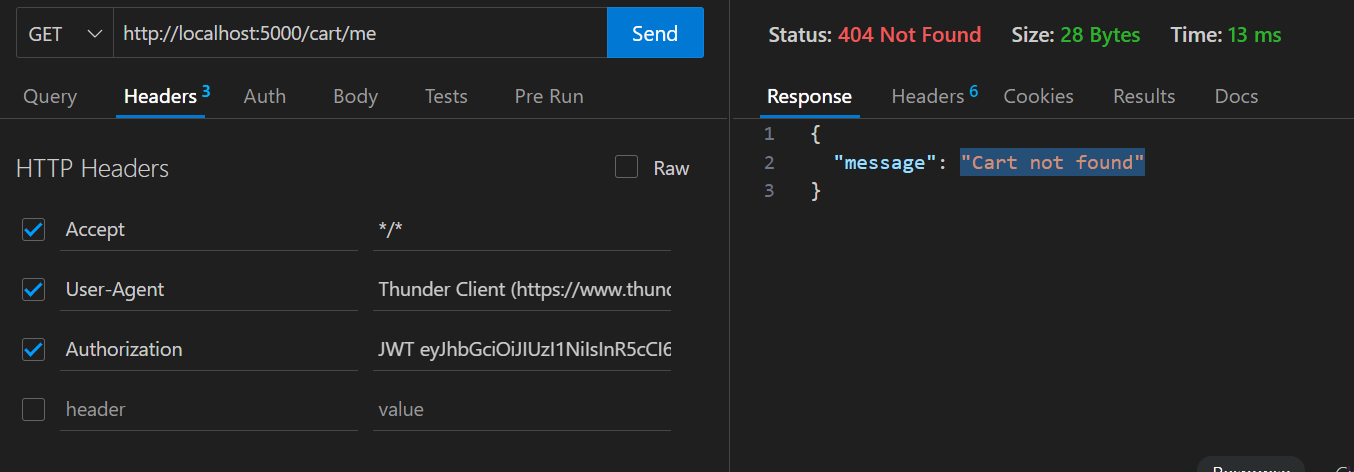
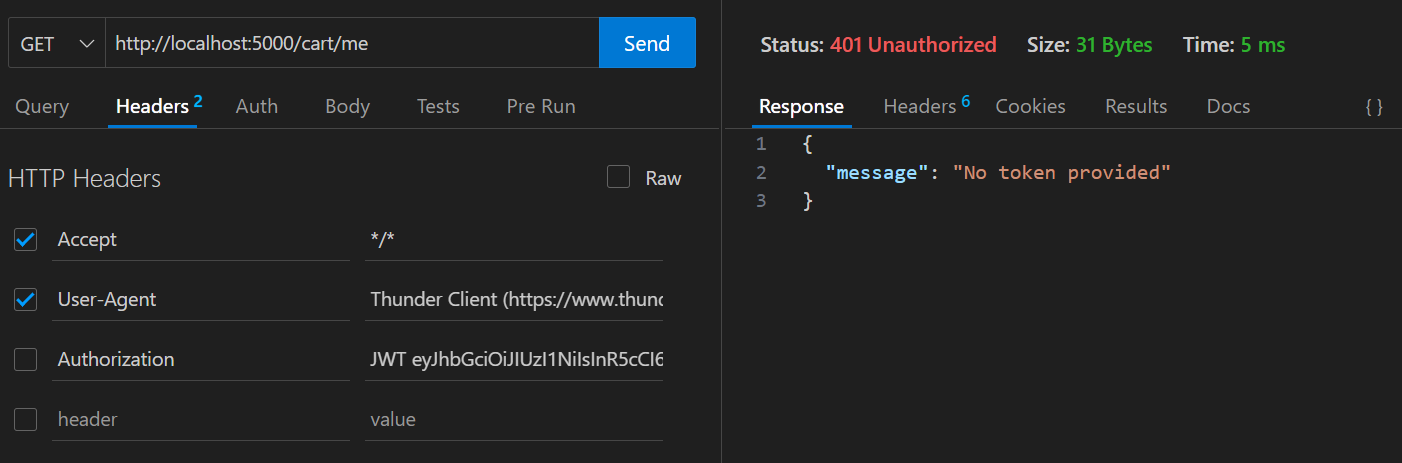
**Expected Result**: status 400 Bad Request and error "At least one field must be provided to update”  
  
  
**Scenario E:** Missing JWT token  
**Expected Result:** 401 Unauthorized, error: "No token provided"  
  
  
  
  
 **Scenario F:** Invalid or expired JWT token  
**Expected Result:** status 403 Forbidden, error: "Invalid or expired token"  
  
  
  
  
  
  
  
  
  
  
  
  
**iii) DELETE /cart/:id  
Description:** Deletes a single cart item based on the given product ID and userid which is extracted from req.user.  
  
**Scenario A**: Valid ProductId and userid  
**Expected Result :**200 OK, Item removed from cart  
  
  
  
  
  
  
  
  
  
  
  
  
**Scenario B**: Given ProductId not present in Cart  
**Expected Result**: status 404 Not Found and error Product not found in cart  
  
  
  
**Scenario C**: Authenticated User Does Not have a Cart   
**Expected Result**: status 404 Not Found and error cart not found  
  
  
 **Scenario D:** JWT Token Missing  
**Expected Result**: 401 Unauthorized, error: "No token provided"  
  
  
  
**Scenario D**: JWT Token Expired or Invalid   
**Expected Result**: 403 Forbidden, error: "Invalid or expired token".  
  
  
  
**iv) DELETE /cart**

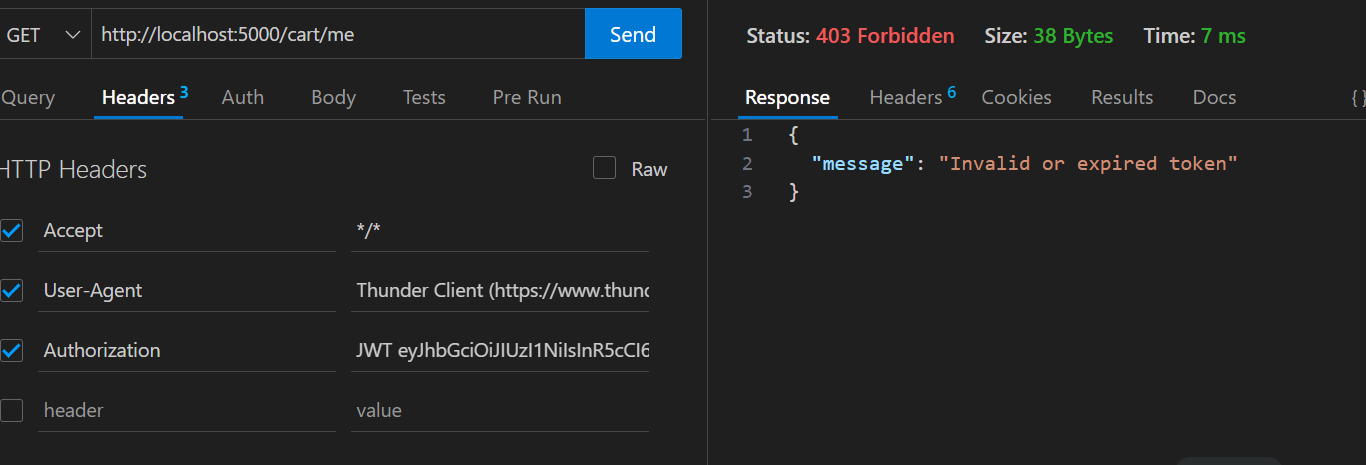
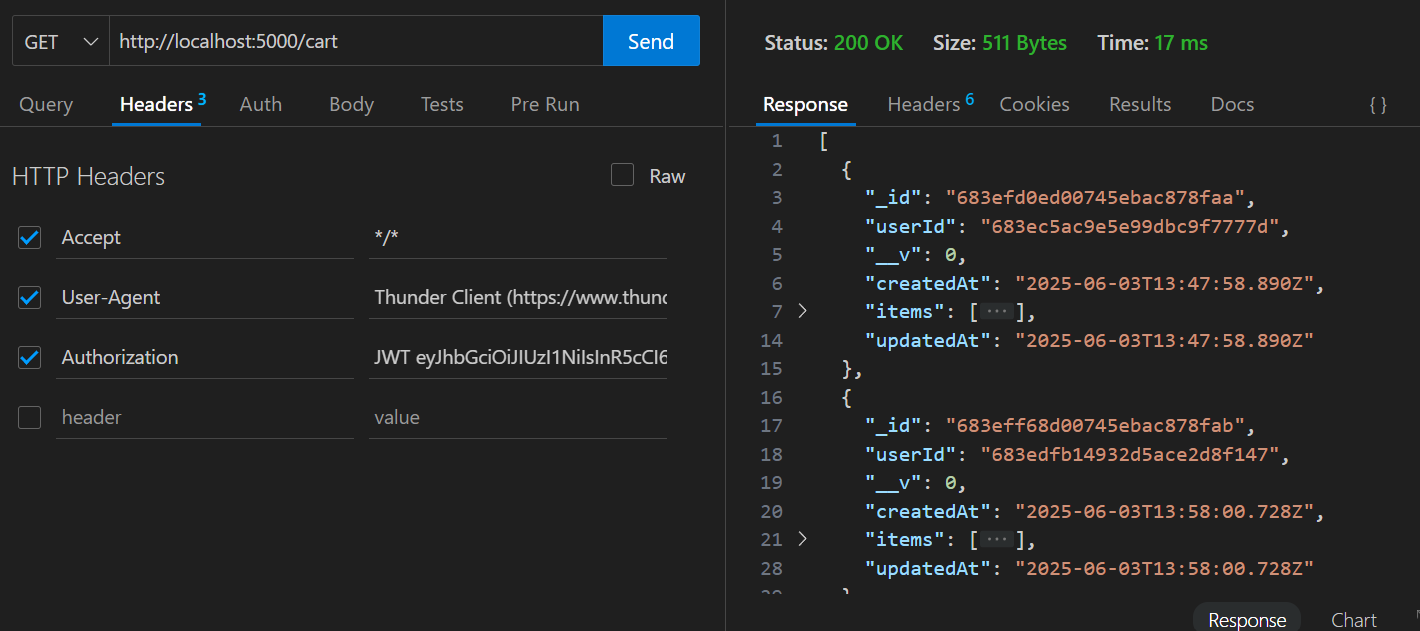
**Description:** Clears the entire cart for a user (removes all items associated with req.user.id).  
  
**Scenario A**: Valid request with existing user id(req.user.id)  
  
**Expected Result:** status 200 OK and message "Cart deleted successfully"  
  
  
  
  
  
 **Scenario B**: Cart already empty or user that does not have a cart  
  
**Expected Result**: 404 Not Found and "message": "Cart not found"  
  


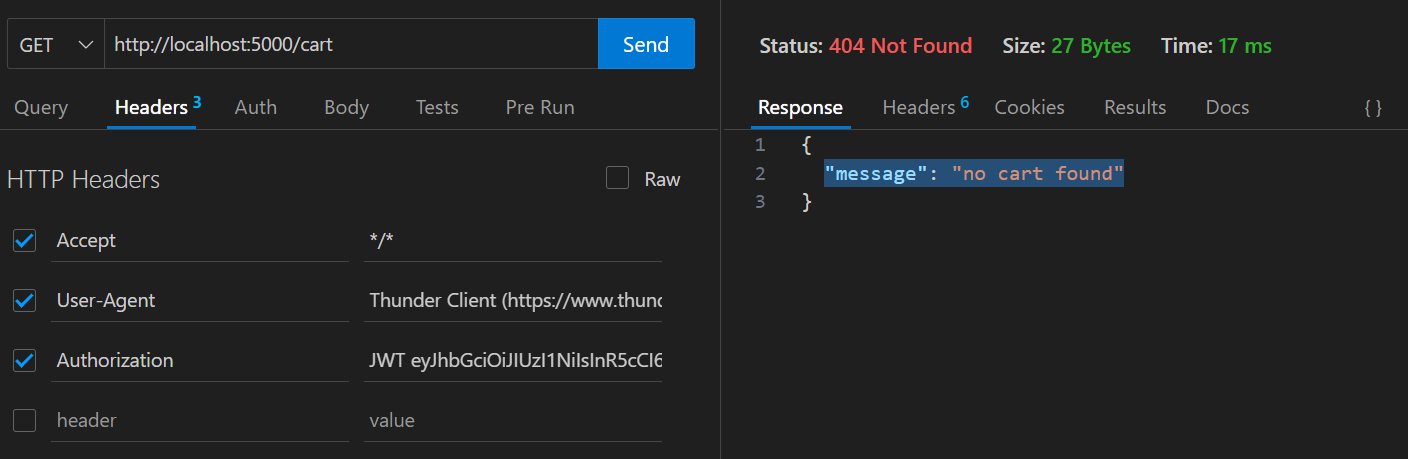
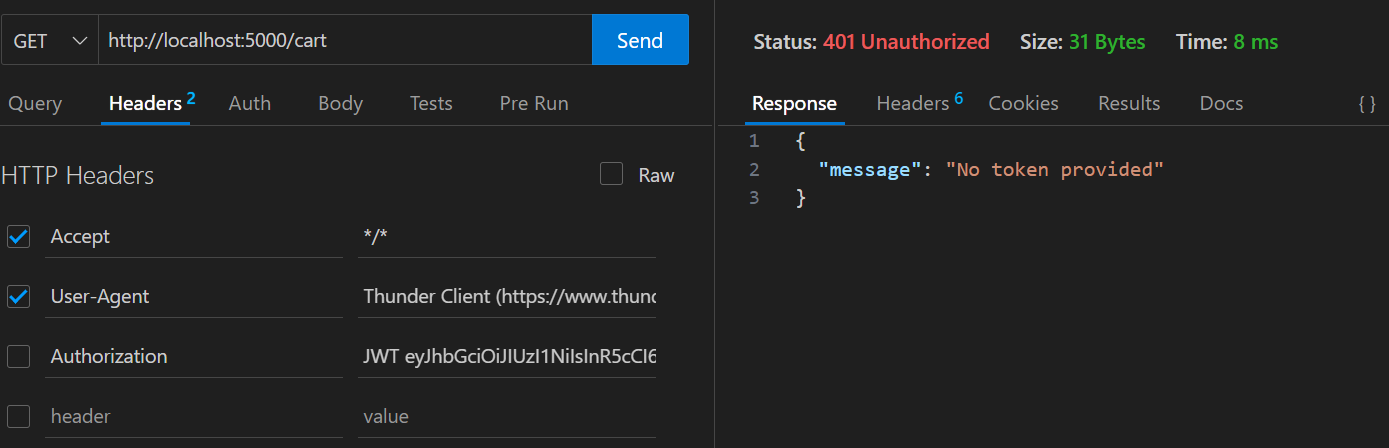
**Scenario C**: JWT Token Missing  
  
**Expected Result**: 401 Unauthorized, error: "No token provided"

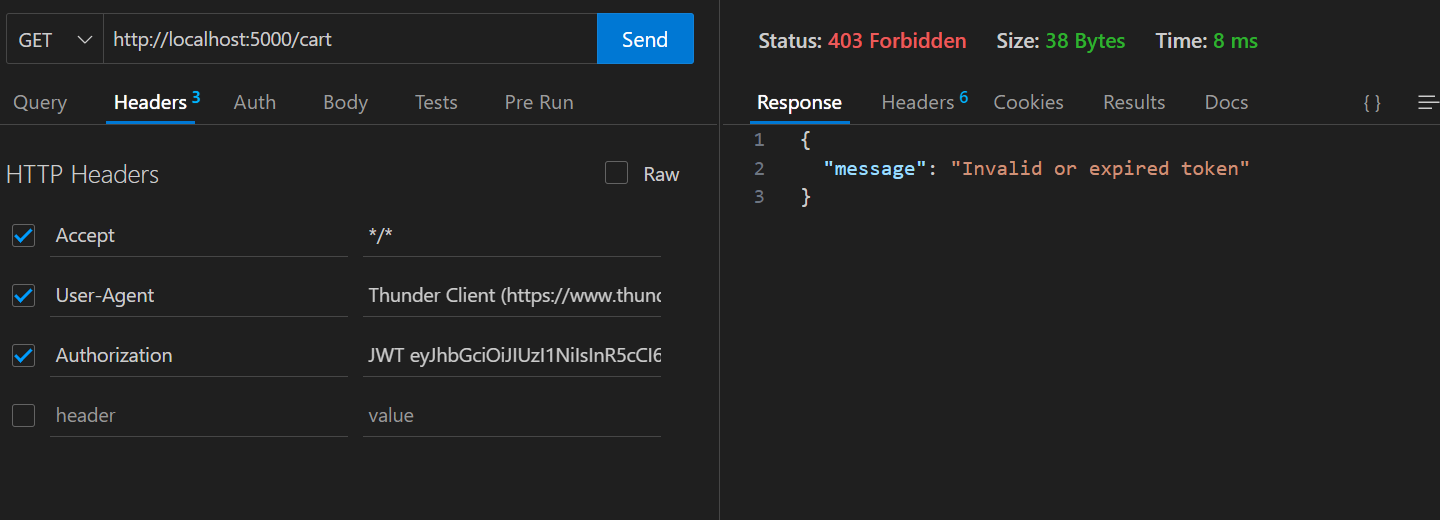
  
  
  
 **Scenario D**: JWT Token Expired or Invalid   
  
**Expected Result**: 403 Forbidden, error: "Invalid or expired token"  
  
  
  
  
  
  
  
  
  
  
**v) GET /cart/me  
  
Description**: Get cart for logged-in user.

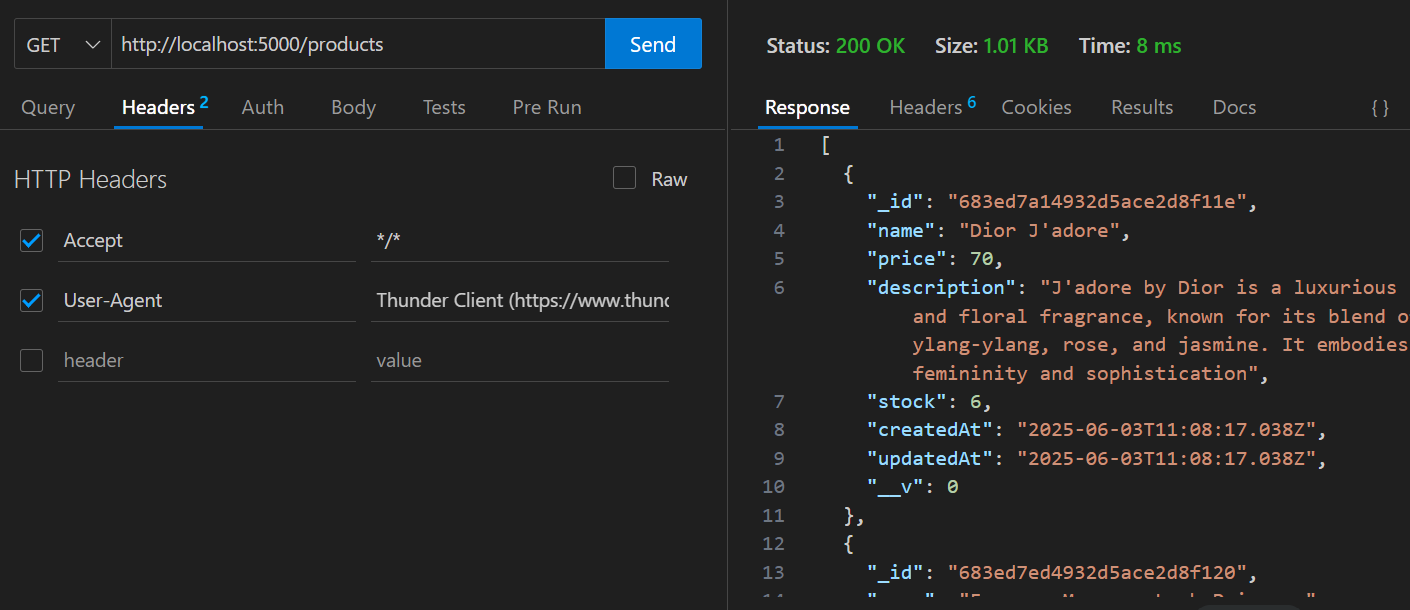
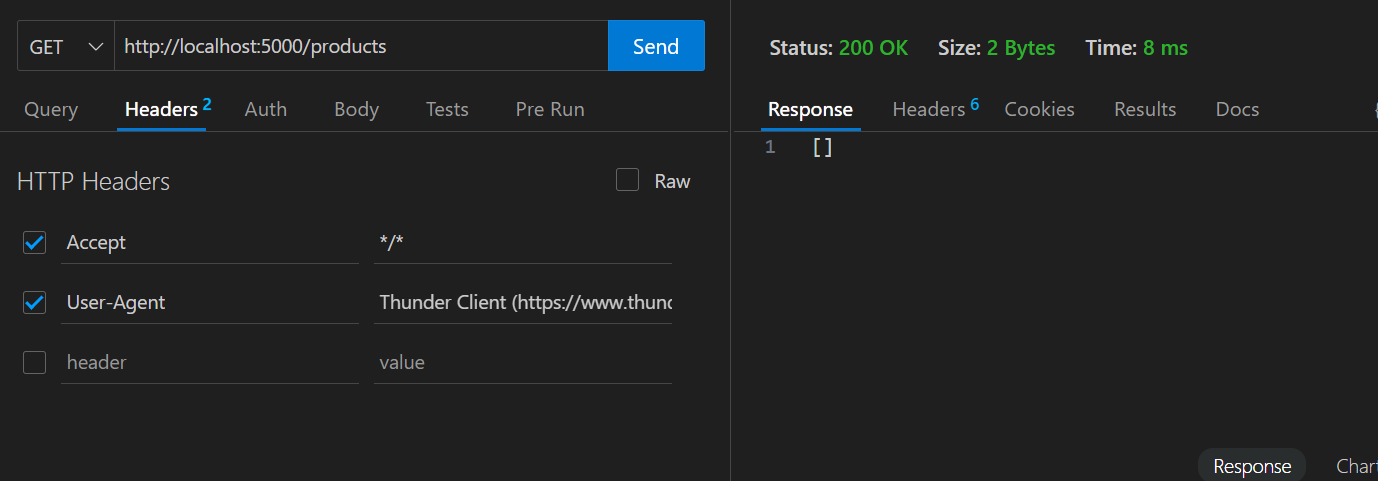
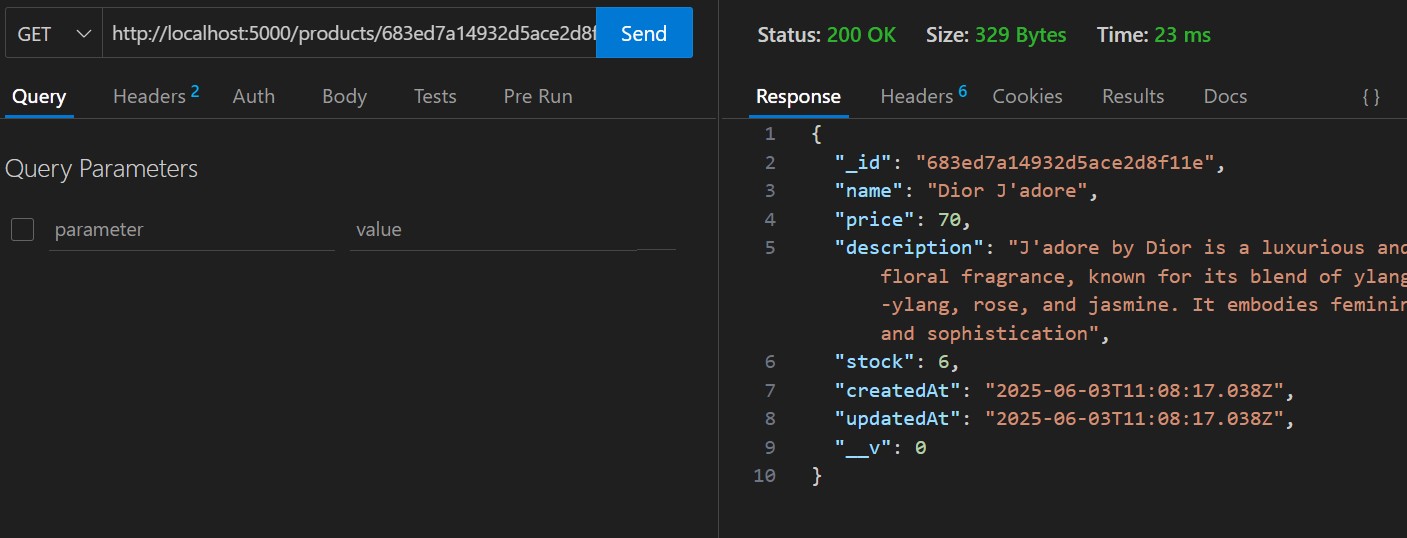
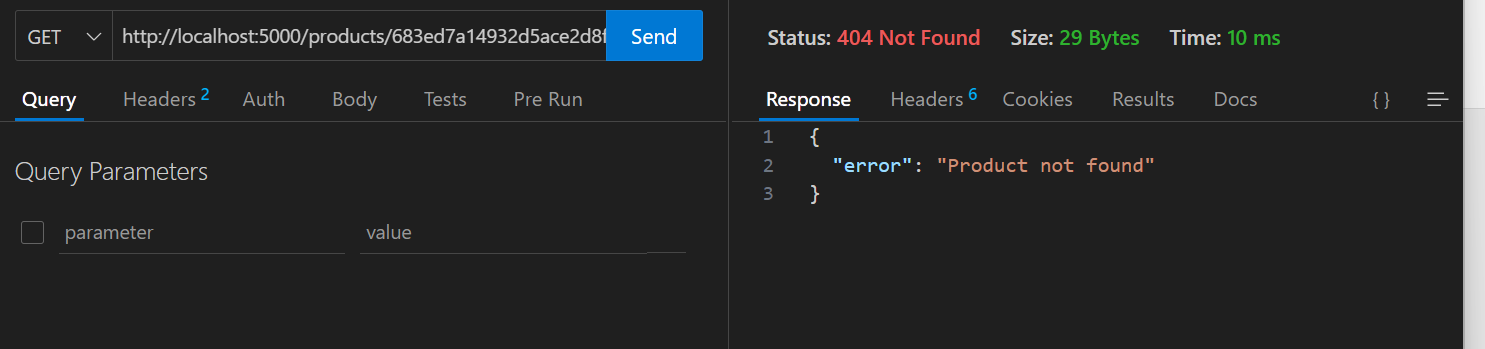
**Scenario A:** User has items

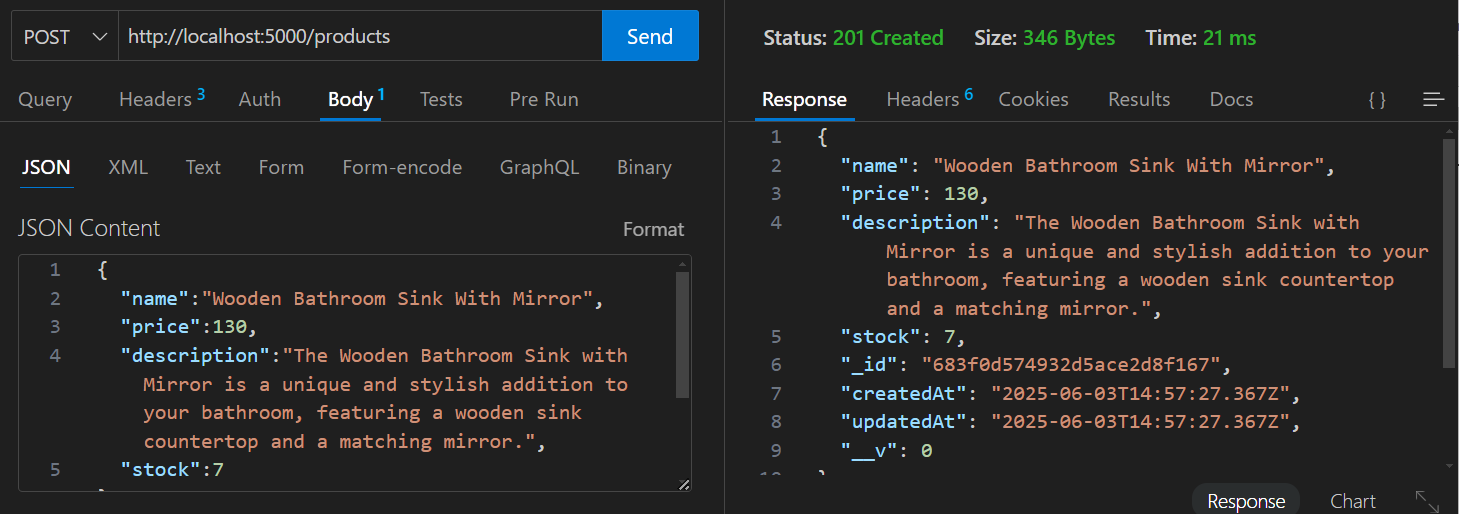
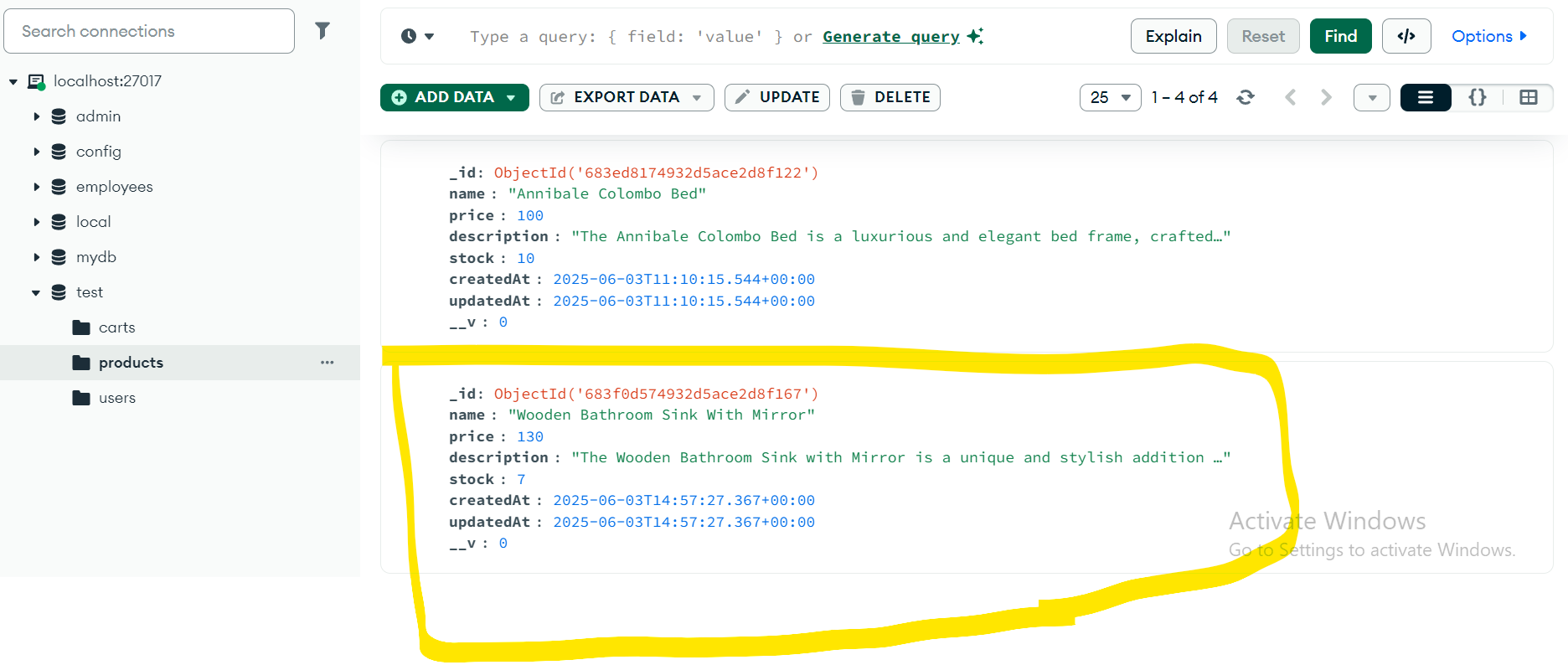
**Expected Result**: status 200 OK and cart Details  
  
  
  
**Scenario B:** if user doesn’t have a cart  
**Expected Result**: status 404 Not Found and message "Cart not found"  
  
  
  
  
  
  
  
  
**Scenario C**: JWT Token Missing  
  
**Expected Result**: 401 Unauthorized, error: "No token provided"  
  
  
  
  
**Scenario D**: JWT Token Expired or Invalid

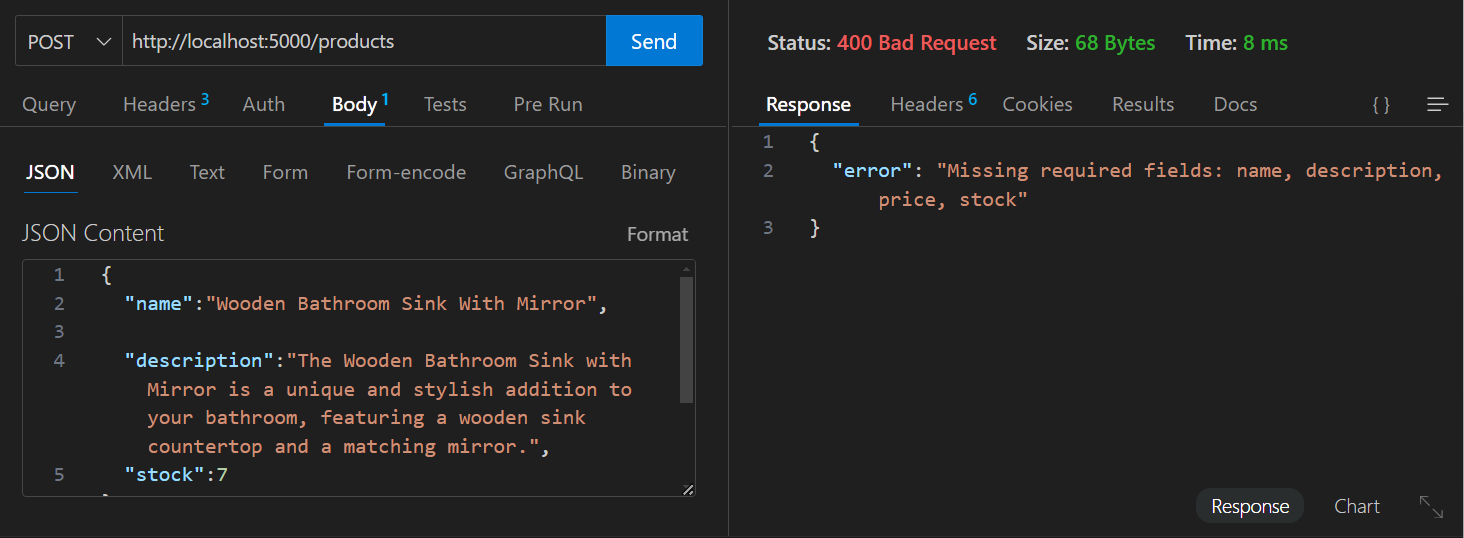
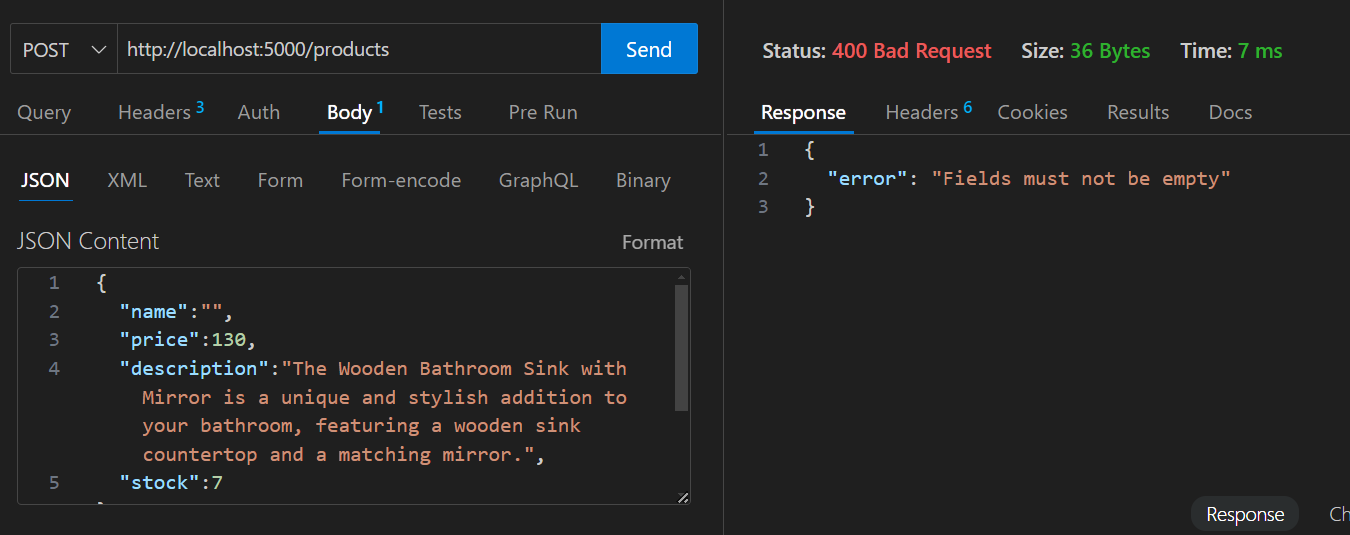
**Expected Result**: 403 Forbidden, error: "Invalid or expired token"  
  
  
  
  
  
  
  
  
  
  
  
**vi) GET /cart**  
**Description**: fetches all users' carts.  
  
**Scenario A:** Multiple user carts exist **Expected Result**: 200 OK, array of carts  
  
****  
  
**Scenario A:** when the carts are empty

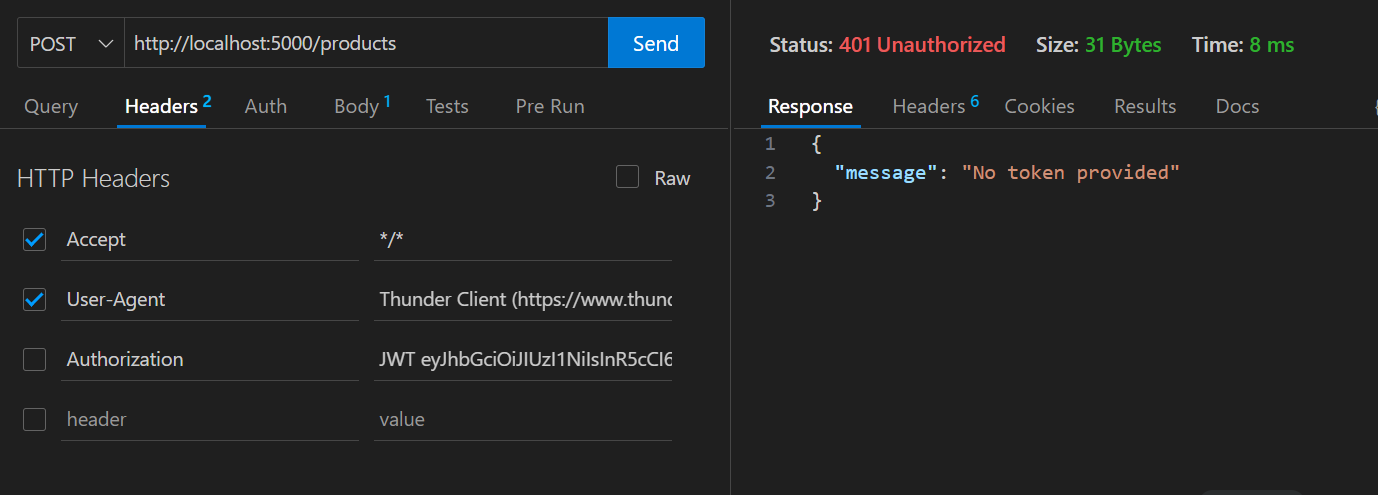
**Expected Result**: 404 Not Found and "message": "no cart found"   
  
  
  
  
  
  
  
**Scenario C**: JWT Token Missing  
  
**Expected Result**: 401 Unauthorized, error: "No token provided"  
  
  
  
**Scenario D**: JWT Token Expired or Invalid

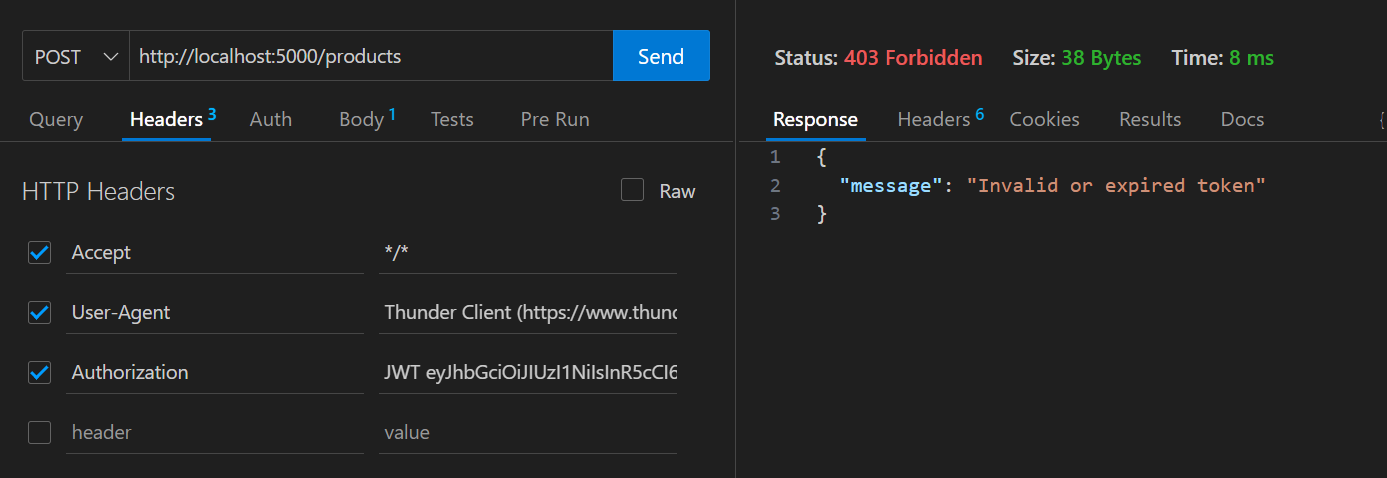
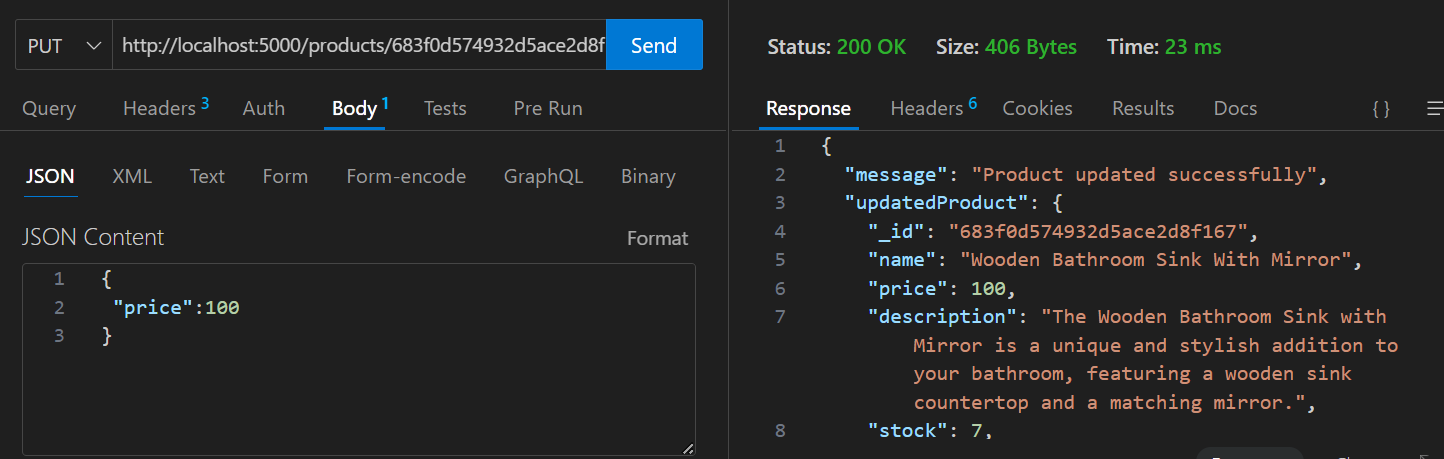
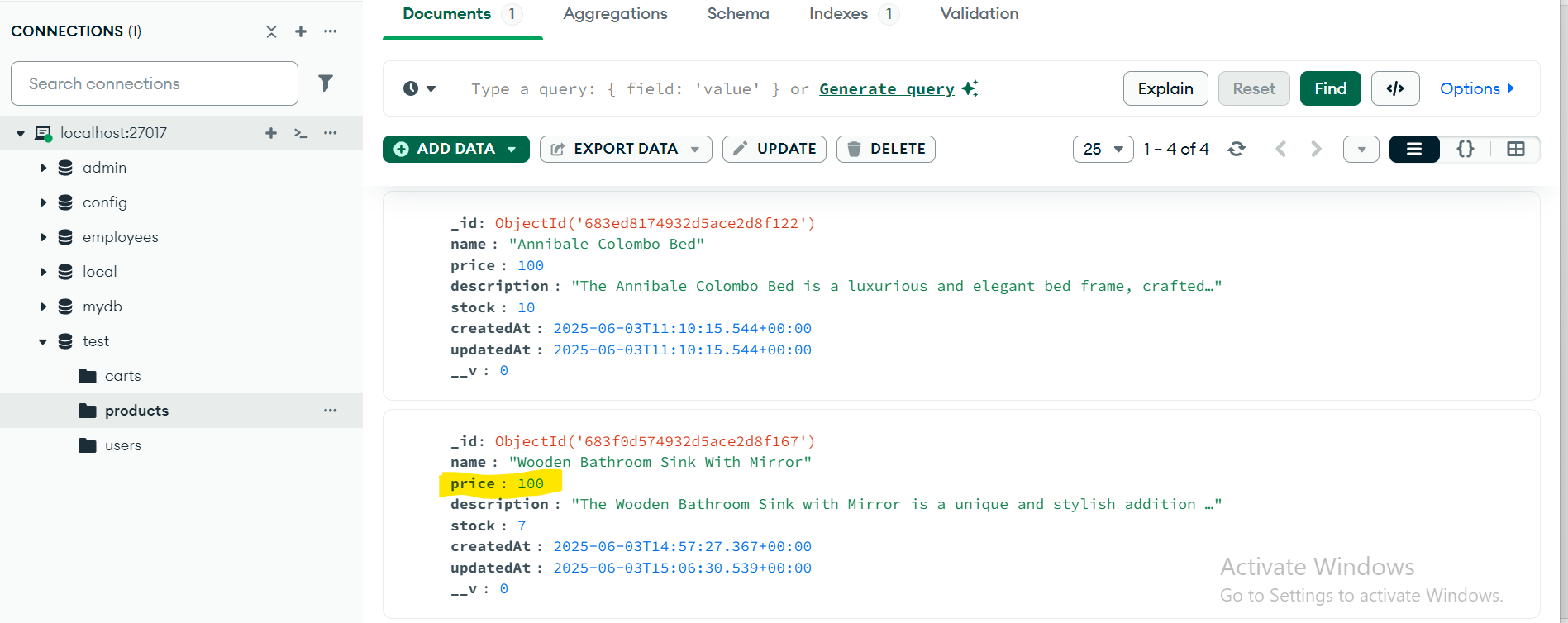
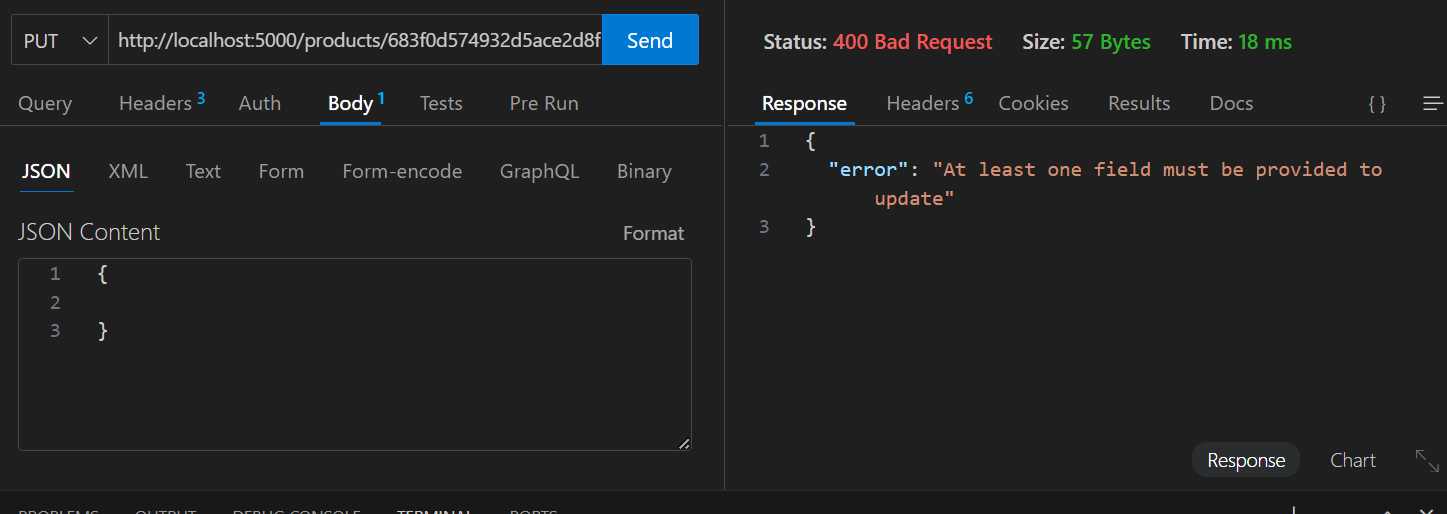
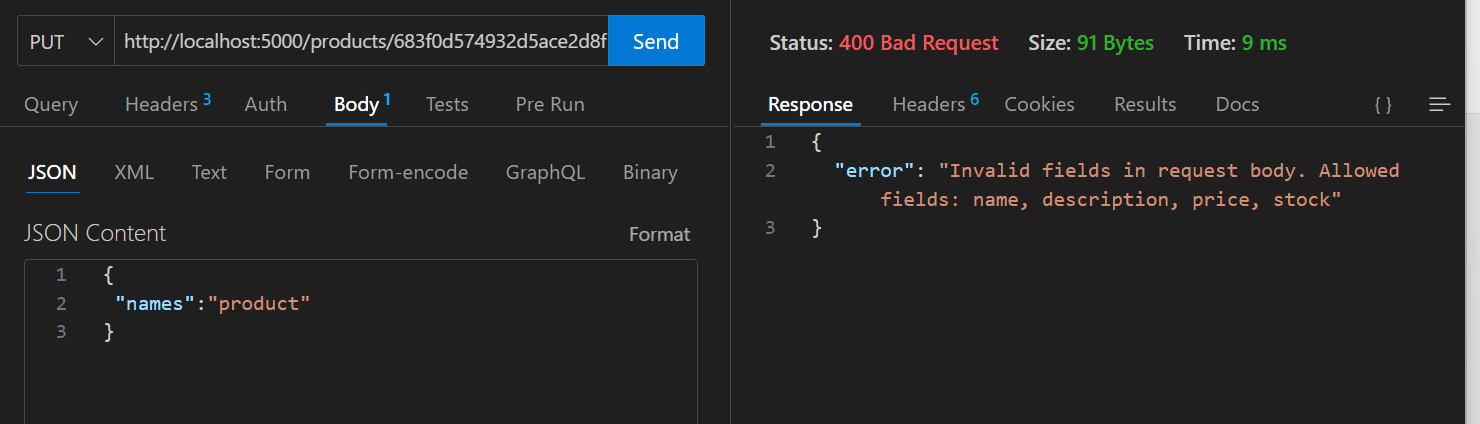
**Expected Result**: 403 Forbidden, error: "Invalid or expired token"  
  


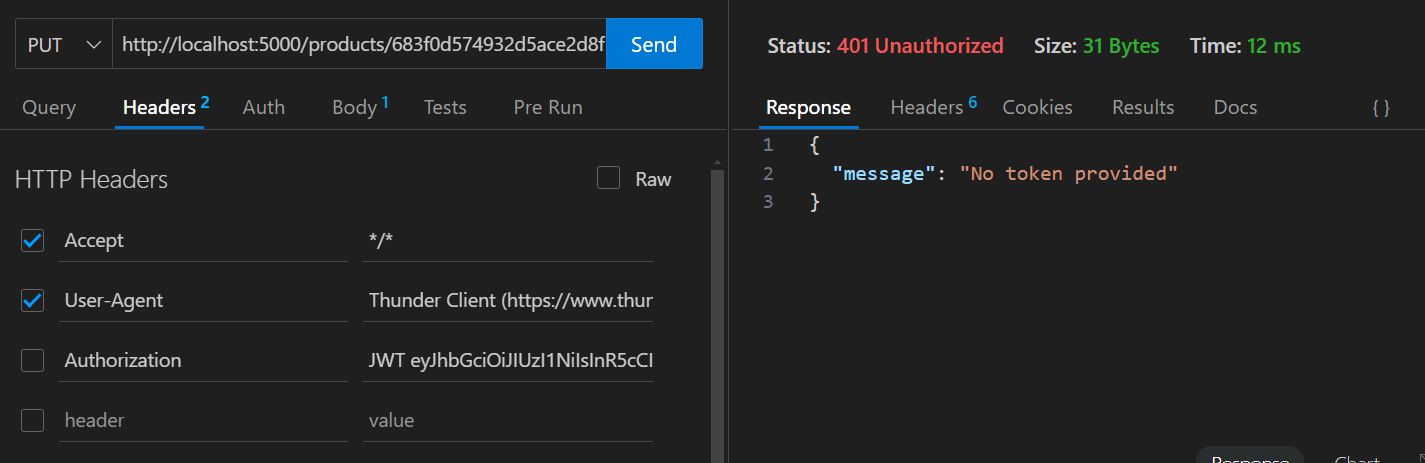
**Product Managmenet  
  
i) GET /products  
  
Description:** Fetch all products from the database  
  
**Scenario A:** Products exist  
**Expected Result:** 200 OK, returns array of product objects  
  
****  
  
**Scenario B**: No products in the database **Expected Result**: 200 ok, empty array  
  
  
  
  
  
  
**ii). GET /products/:id  
Description**: Get a single product by its MongoDB ID.  
  
**Scenario A:** Valid product ID  
 **Expected Result** :200 OK, returns matching product object  
  
****  
  
**Scenario B**: when productid doesn’t exist  
  
**Expected Result:** 404 Not Found, message like "Product not found**"  
  
**

**iii). POST /products  
Description:** Add a new product. **Scenario A:** All fields provided with valid values  
 **Expected Result:** 201 Created, returns saved product  
  
  
  
  
  


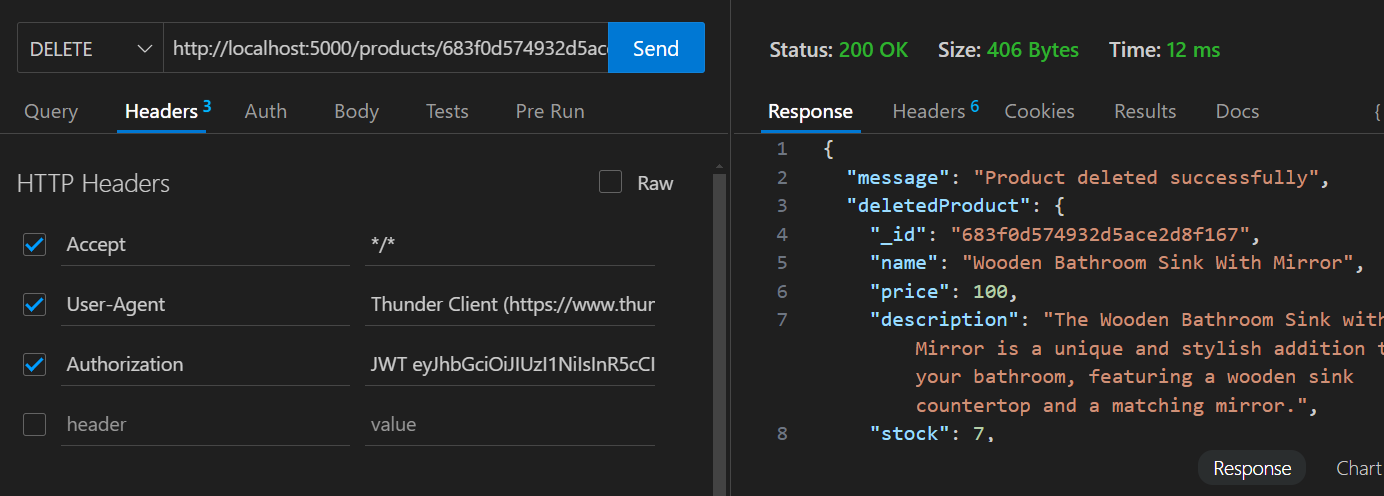
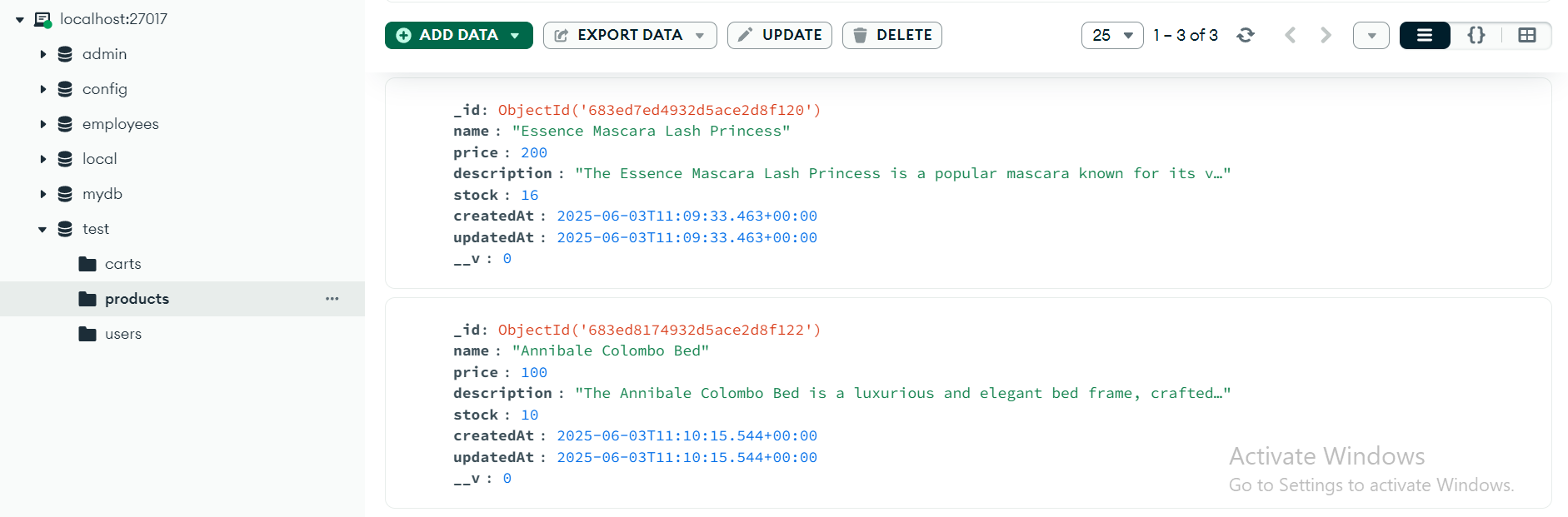
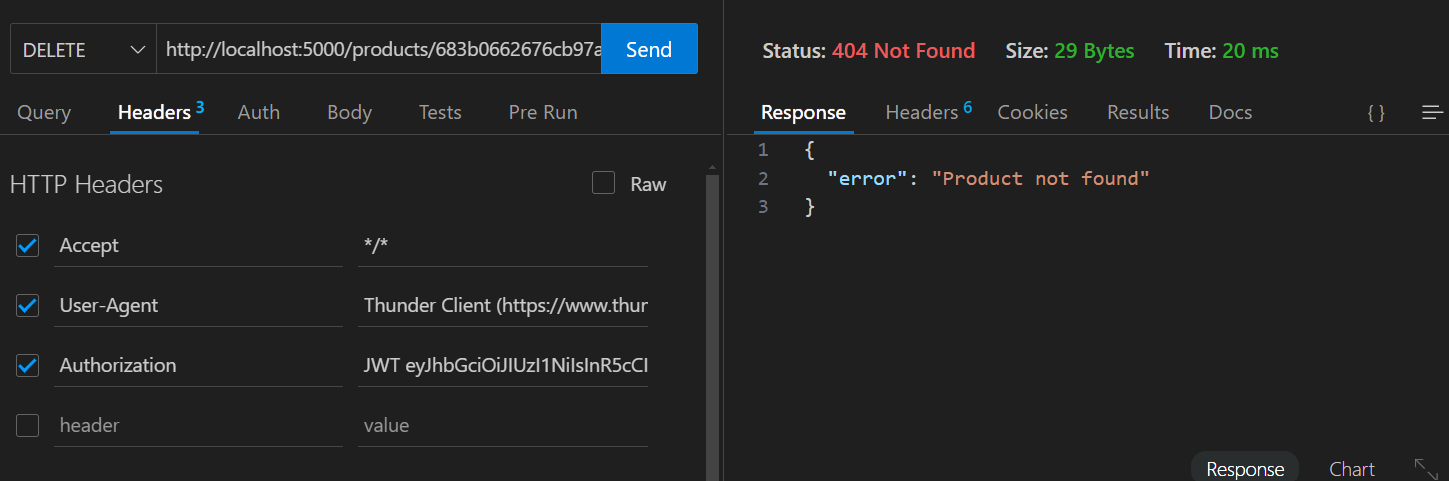
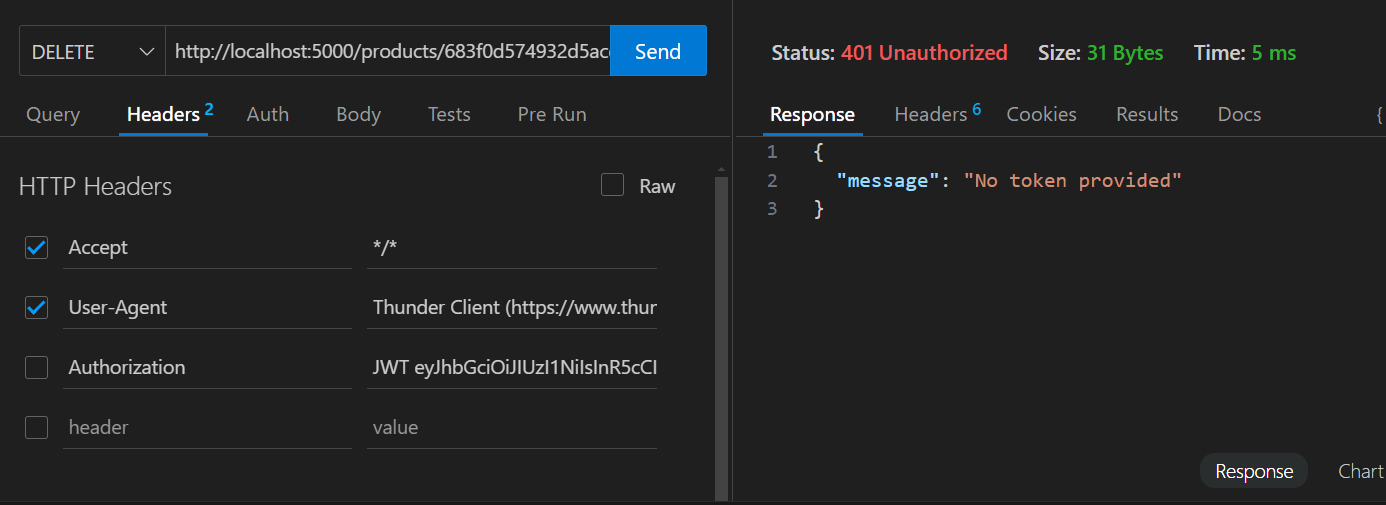
**Scenario B:** Missing required fields (e.g., no price)  
 **Expected Result:** 400 Bad Request, error:"Missing required fields: name, description, price, stock"  
****  
  
**Scenario C**: Empty values (e.g., name: "")  
  
**Expected Result:** 400 Bad Request, error: "Fields must not be empty"  
  
****  
  
  
  
  
  
  
  
 **Scenario D**: JWT Token Missing  
  
**Expected Result**: 401 Unauthorized, error: "No token provided"

**  
  
  
Scenario E**: JWT Token Expired or Invalid

**Expected Result**: 403 Forbidden, error: "Invalid or expired token"  
  
  
  
  
  
  
  
  
  
**iV). PUT /products/:id  
Description**: Update one or more product fields by ID.  
  
**Scenario A**: Update valid field  
  
**Expected Result**: 200 OK, returns updated product  
  
  
  
  
  
 **Scenario B:** No fields sent in body  
**Expected Result:** 400 Bad Request, error: "At least one field must be provided to update"  
****  
**Scenario C**: Empty value  
**Expected:** 400 Bad Request, error: "Fields must not be empty" ****  
 **Scenario D: Invalid fields  
Expected**: 400 Bad Request, error:  
"Invalid fields in request body. Allowed fields: name, description, price, stock" ****

**Scenario E**: JWT Token Missing  
  
**Expected Result**: 401 Unauthorized, error: "No token provided"  
  
  
  
  
**Scenario E**: JWT Token Expired or Invalid

**Expected Result**: 403 Forbidden, error: "Invalid or expired token"  
  
  
****  
  
  
  
  
  
  
  
**V). DELETE /products/:id**

**Description:** Delete a product by ID.  
  
**Scenario A: Valid existing product ID  
Expected Result:** 200 OK, message product deleted successfully and deleted product details  
  
  
  
  
  
 **Scenario B:** Non existing productid **Expected Result**: 404 Not Found, error: "Product not found"  
  
  
**Scenario C**: JWT Token Missing  
**Expected Result**: 401 Unauthorized, error: "No token provided"  
  
  
**Scenario D**: JWT Token Expired or Invalid   
**Expected Result**: 403 Forbidden, error: "Invalid or expired token"  
