**NAME: CHUKWU DANIEL NONSO**

**MATRIC NO.: BHU/20/04/05/0010**

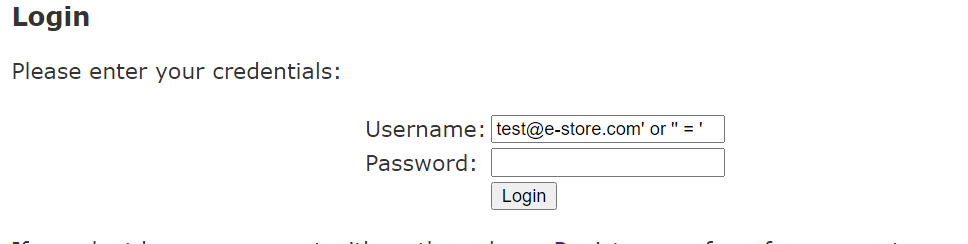
**CMP 408 ASSIGNMENT 1**

# 

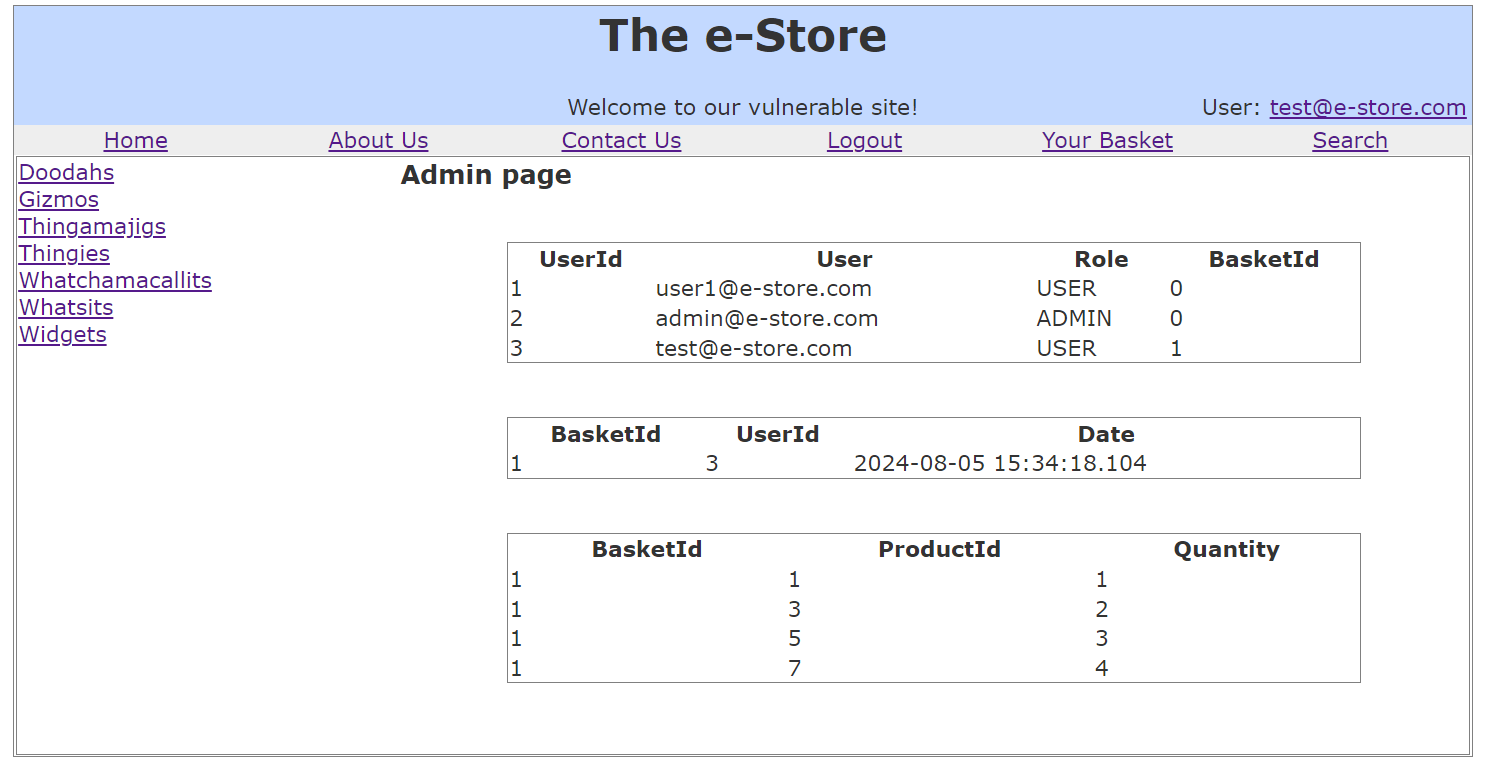
# 1: Login as test@e-store.com

# 3: Login as User1@e-store.com

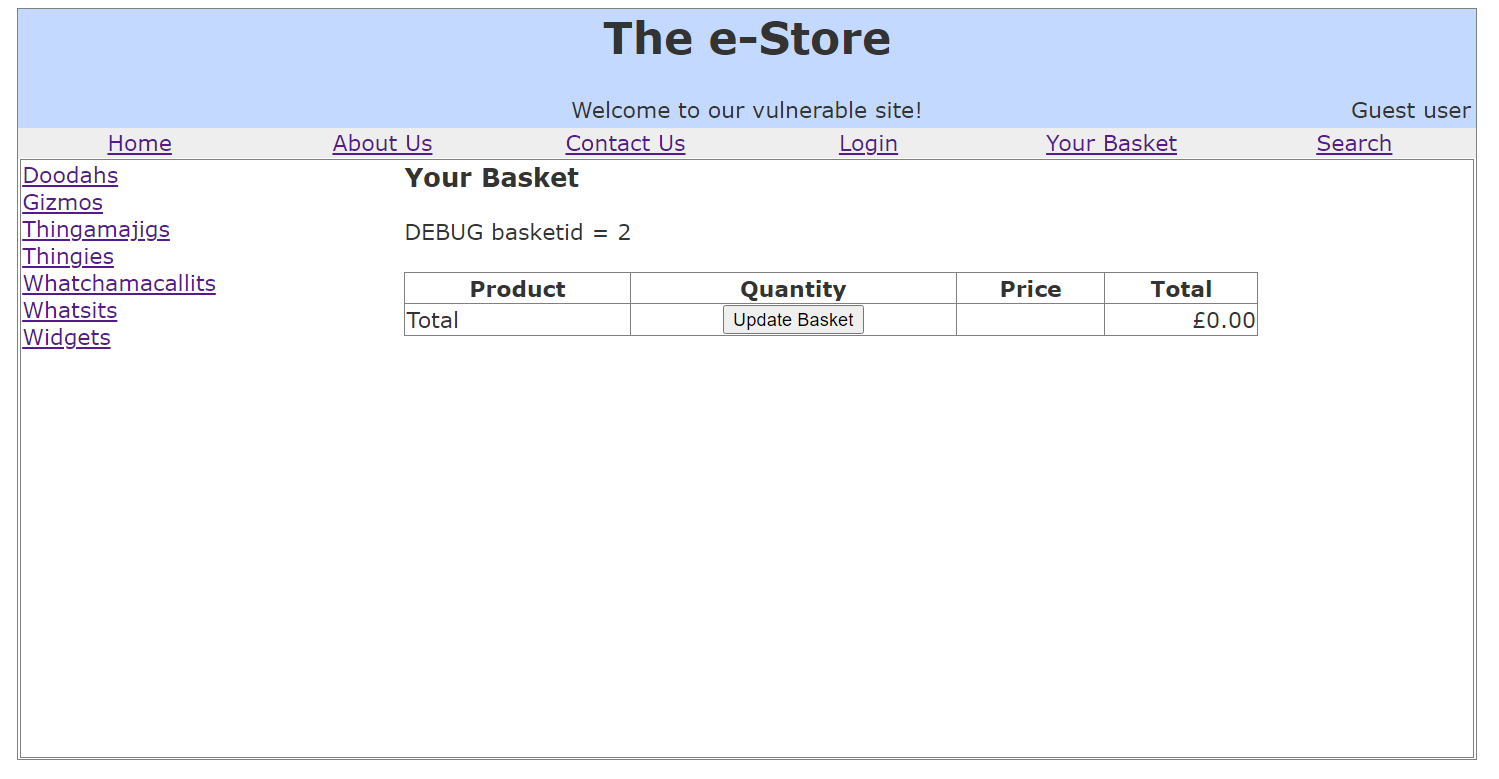
# 3: Login as admin @e-store.com

1. Navigate to the login page.
2. Enter the email that you are trying to gain authorization to. In this case we are going to be using [test@e-store.com](mailto:test@e-store.com).
3. We are going to gain access by exploiting the sql injection vulnerability by entering a Boolean statement that always returns true. 
4. Then you hit the login button.
5. You should see the login successful message. 

# 4: Find hidden content as a non admin user

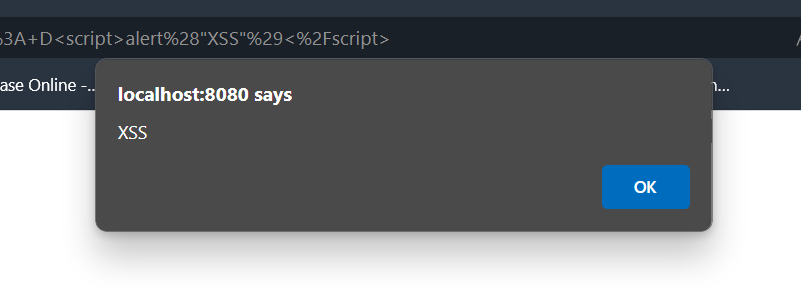
1. To find hidden admin content navigate to the admin page by visiting the /admin.jsp endpoint
2. We have gotten access to the hidden content displayong valuable database information. 

# 5: Find diagnostic data

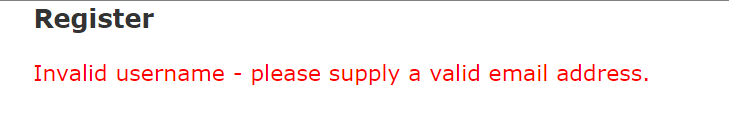
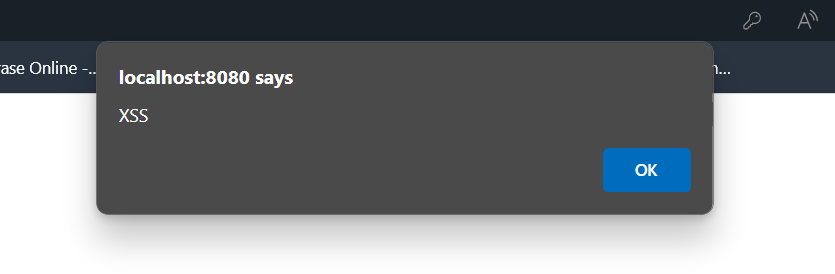
1. We want to get info from the database such as errors so we can know how the dev implemented it.
2. We navigate to the basket page.
3. We add a URL parameter, since most devs forget to validate their URLs when using JavaScript, we add the “?debug=true” parameter to the back of the page URL. 
4. We just found some useful information on how the database identifies baskets. 

# 6: Level 1: Display a popup using: <script>alert("XSS")

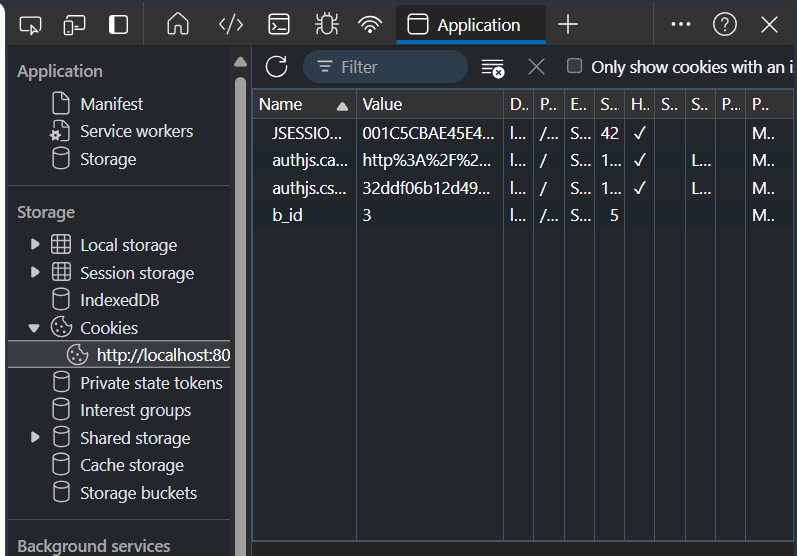
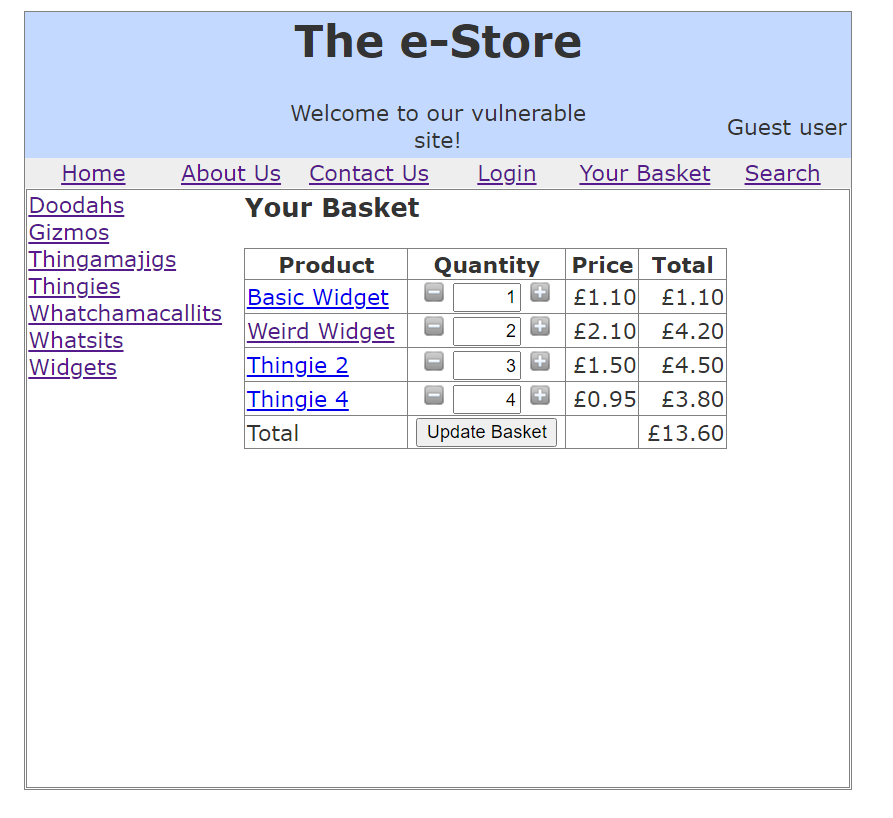
# </script>

1. We navigate to the search page.
2. And then enter a script into the search bar.
3. We saw a browser alert. 

# 7: Level 2: Display a popup using: <script>alert("XSS")

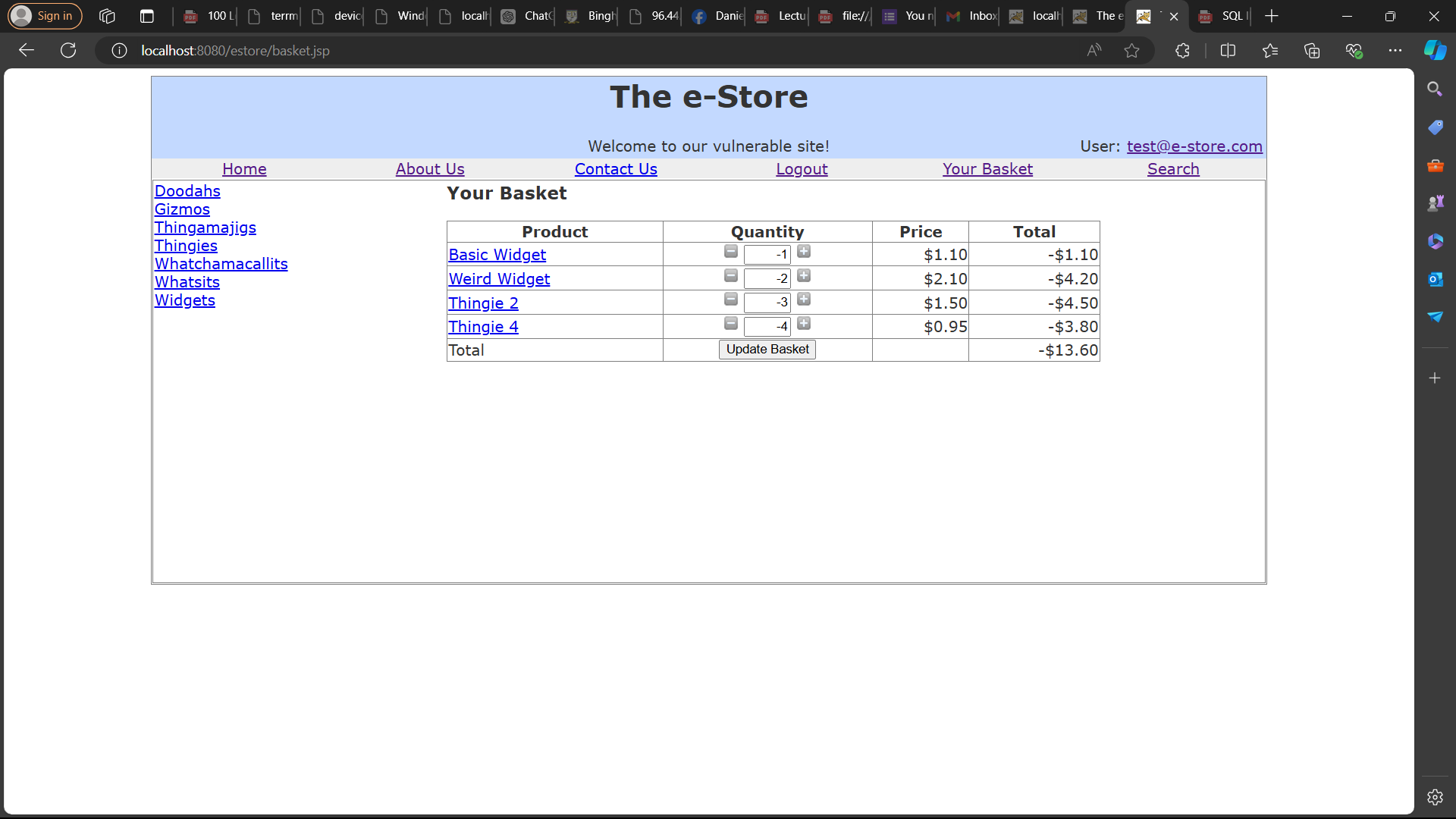
1. We navigate to the register page.
2. And attempt to register as a user but instead we add some code into the username input box.
3. It denied us access because it used a regex statement to determine whether we entered an email. 
4. So we trick it by adding a string in front “test@”.
5. We have pawned injected the script successfully. 

# 8: Access someone else’s basket

1. We navigate to the basket page.
2. We know from adding that URL parameter that the website probably stores the data on the client with cookies so we trick it by modifying the cookie.
3. We access the dev tools page and move to the applications tab. 
4. Notice the “b\_id” row, the value is stored there so we are going to change it to the value of “1” in other ro get access to another users basket. 

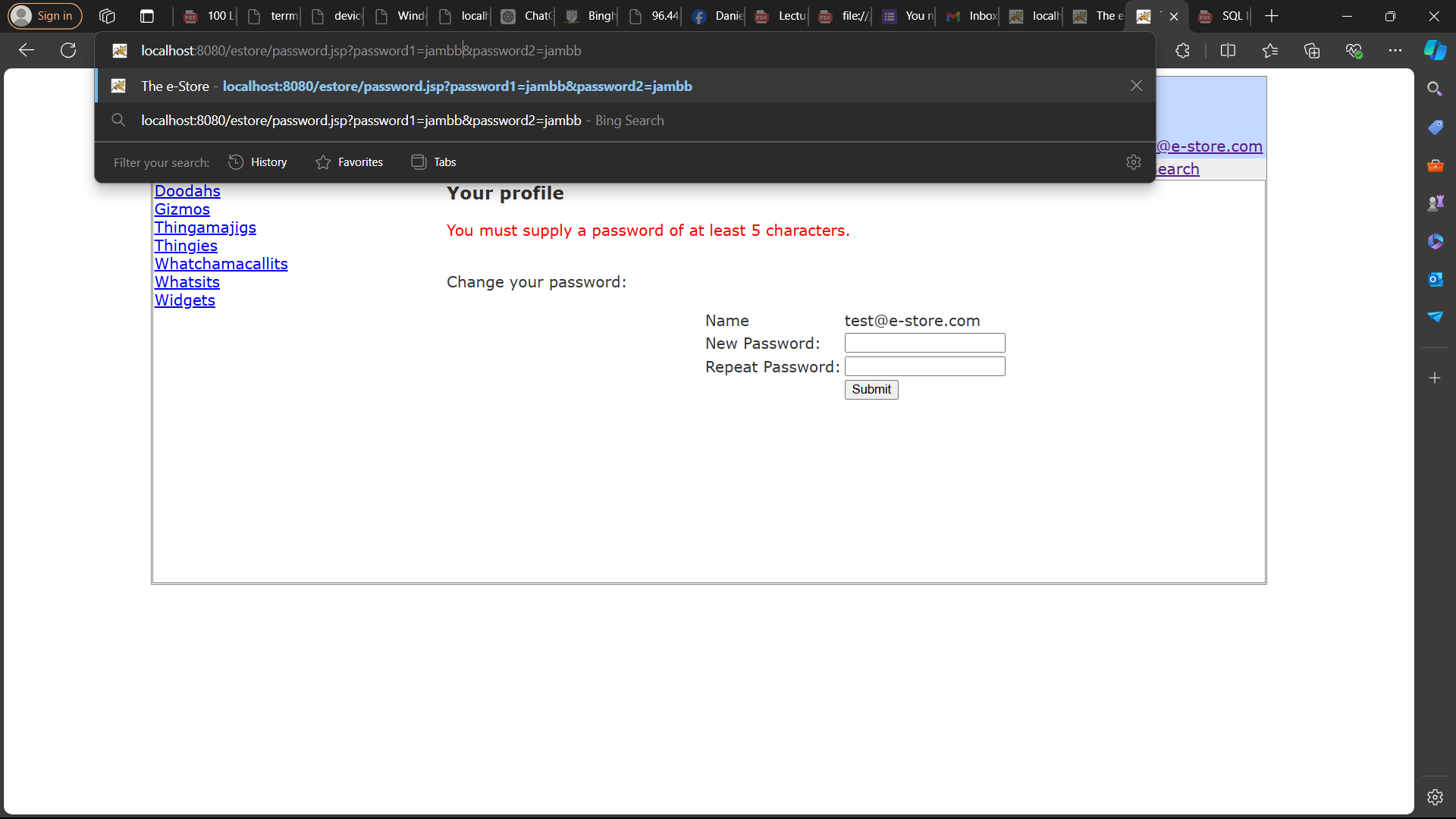
# 9: Get the store to owe you money

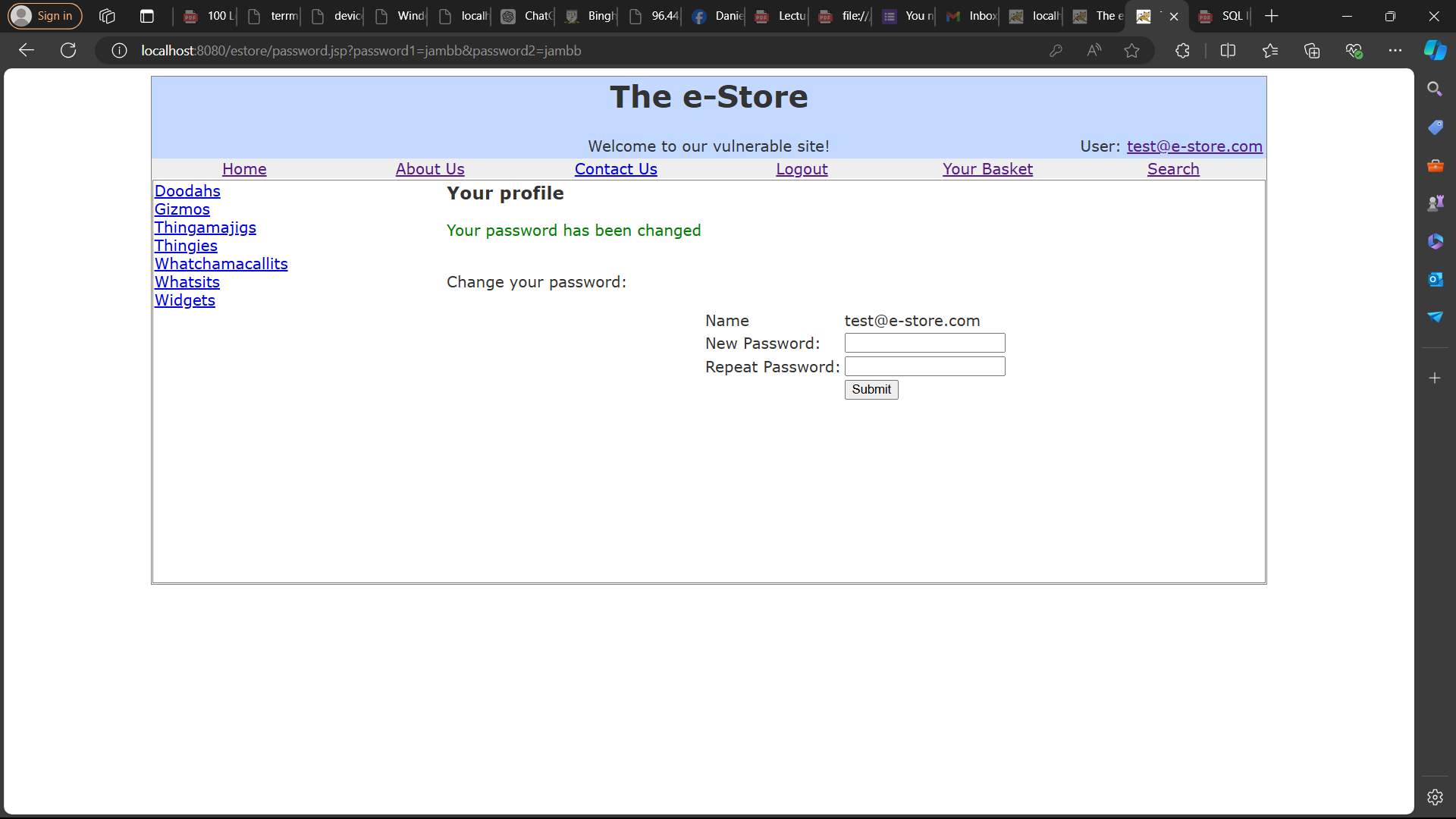
1. Add an item to your baskets page.
2. Navigate to the baskets page.
3. Inspect on the quantity element in the table.
4. Edit the value from the pages dom in the dev tools elements tab. Change it to a negative value.
5. Then tap the update basket button.



# 10: Change your password via a GET request

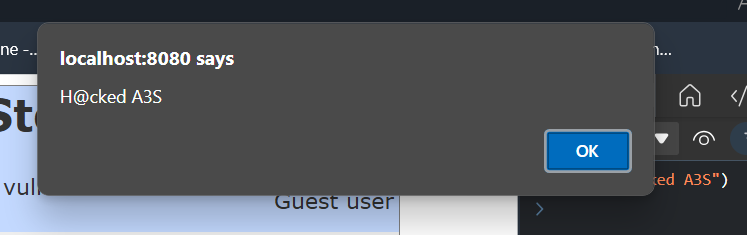
1. Login as a us
2. Navigate to the password change page (Localhost:8080/estore/password.jsp)
3. Append some url parameters to change the password (password1, password2)





# 11: Conquer AES encryption, and display a popup using:

# <script>alert("H@cked A3S")</script>

1. Navigate to the advanced search page.
2. Using burp we intercept the javascript and modify the validateForm function. 

# 12: Conquer AES encryption and append a list of table names to the normal results.

1. Navigate to the advanced search page.
2. We use burp to intercept the connection and modify the code and then append tables from the database.
3. Using this sql code “anything%' UNION SELECT (SELECT LIMIT 3 1 table\_name FROM information\_schema.system\_tables ORDER BY table\_name),2,3,4,5 FROM products –”, we enter it into the types box and then hit search.