**CS6008 Cryptography and Network Security.**

**Assignment No.4**

**Name:** G Vignesh **Reg No:**2018103078

**Date:** 11.5.2022

**Module-3:**  EXPLOIT TECHNIQUES

**External Learning-** Implement SQL injection in PHP based websites

**Topic:**  Attack Vulnerable PHP websites using SQL commands.

**Aim:**

To check and find the vulnerabilities in the PHP based website ,use the vulnerability to exploit the database present in the server to and print all the records present in the table, modify the table, details of other tables present in the database and drop tables. Perform mitigation by using prepared statements.

**Tools-used:**

1.Xampp Apache server

2.MySQL Database

3.Any browser(Microsoft Edge)

4.VSCode (Code Editor)

**Language Used:**

1.HTML(Hypertext Markup Language)

2.PHP(Hypertext Pre-processor)

3.Sql(Structured Query Language)

4.CSS(cascading Style Sheet)

**Description:**

SQL injection attacks are performed by hacker by passing SQL based commands as input to the forms provided by developers in their PHP sites which has vulnerability. Since developers where using query statements without prepared statements and followed by its execution of prepared statements, hackers where able to pass SQL code as input to the forms because of no input validation and sanitization of the input, sql command provided by the hacker instead of being considered as a input it was capable to perform the query command in the server and able to access the database, due to which hacker was able to print all the records in the table, modify values in the table, access details about other table in the database. By using prepared statements, we can provide a countermeasure to this SQL injection problem.

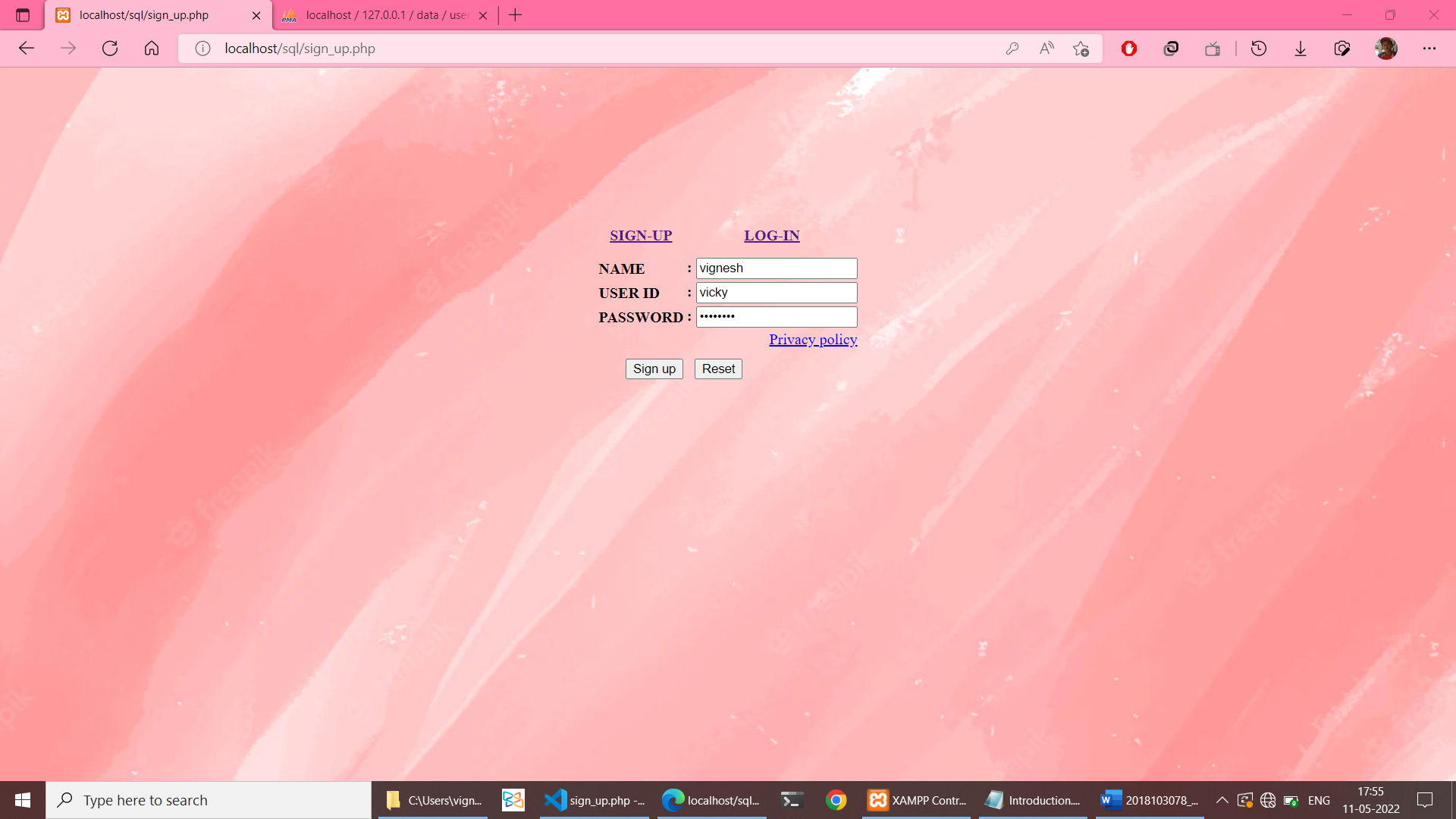
**Input:** SQL Injection commands in vulnerable PHP site.

**Output:** Print all records in table, modify the values in the table, view all tables in the database.

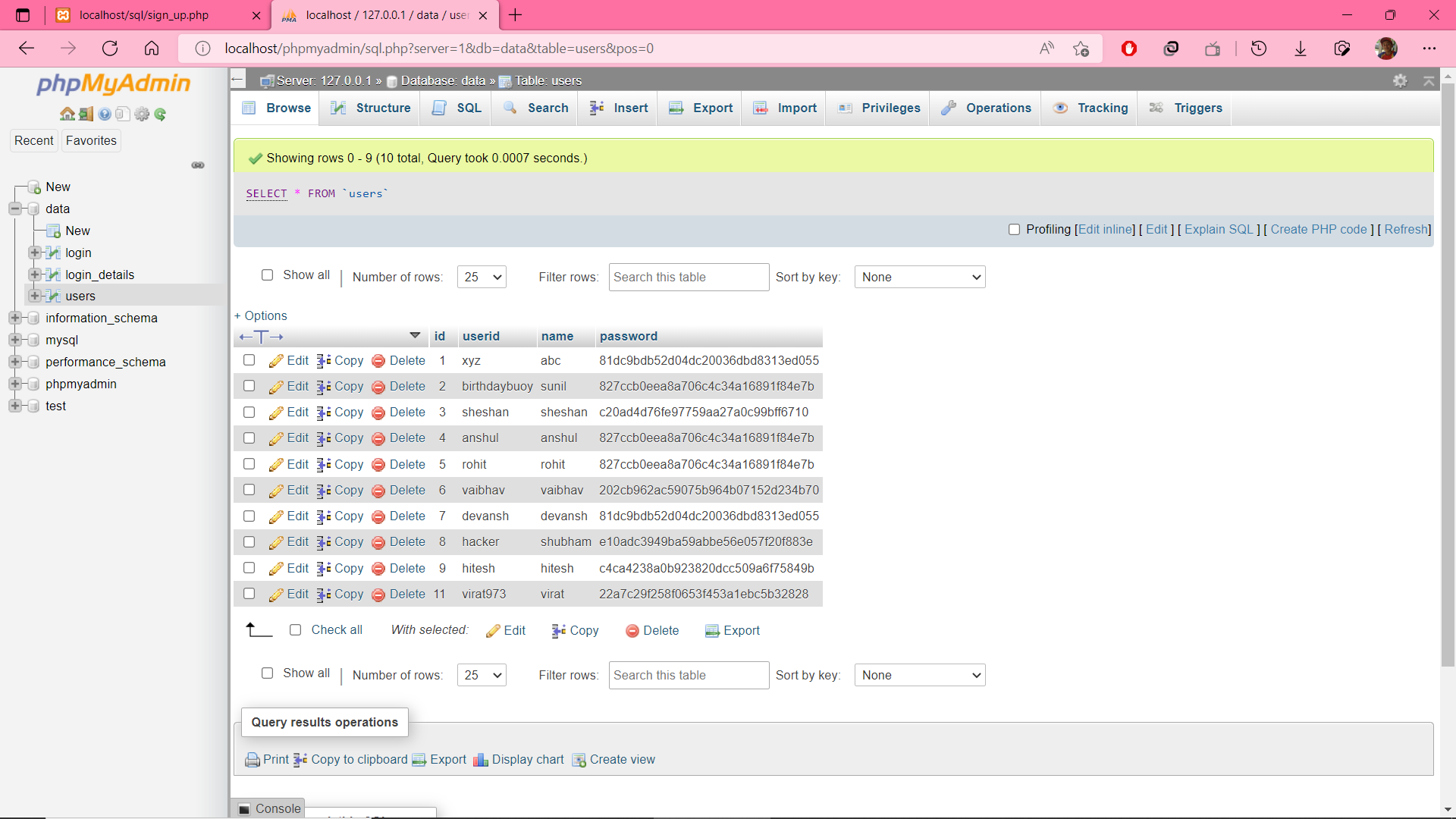
**Execution:**

**Step-1:**

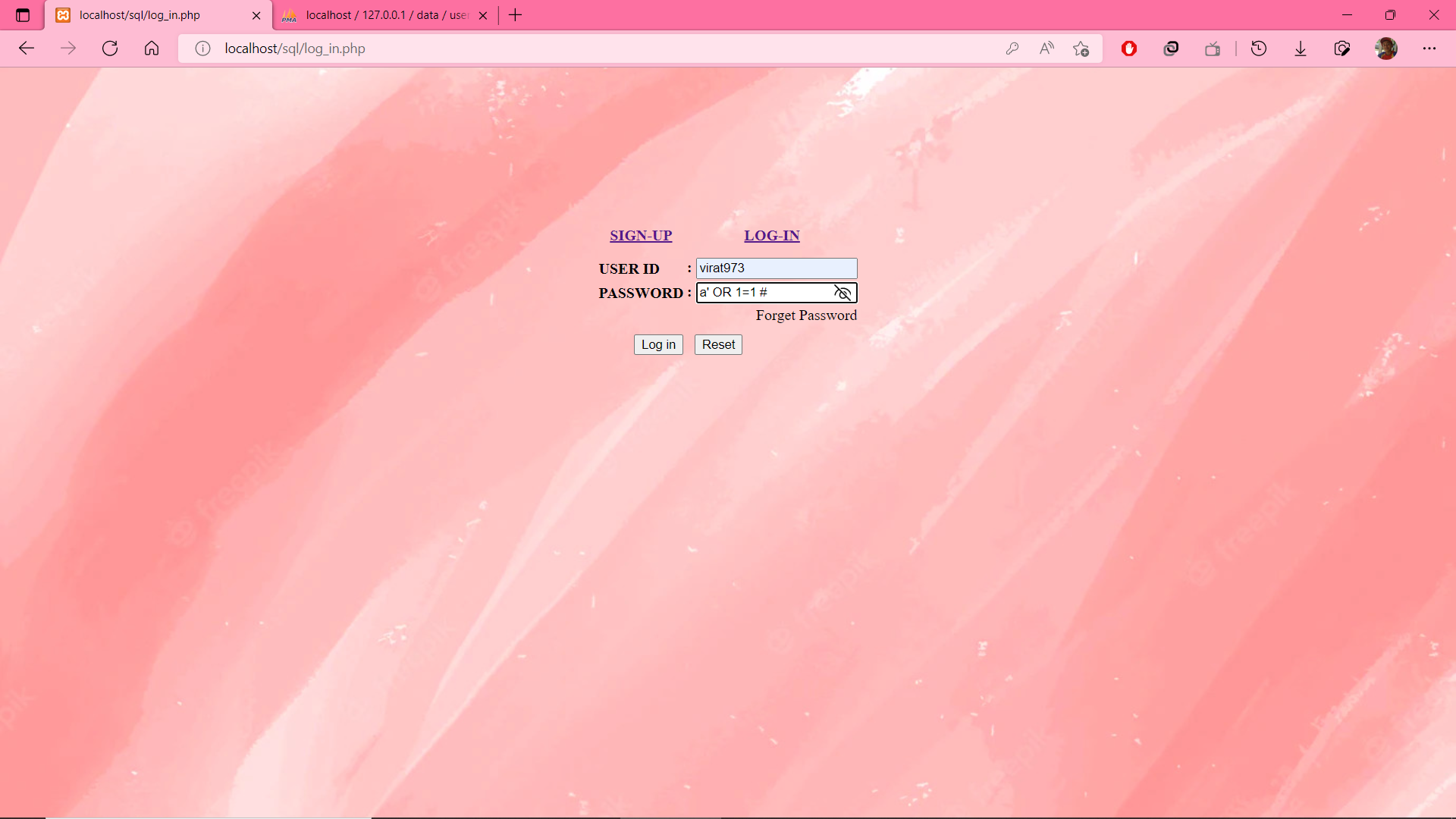
**Login without knowing password.**



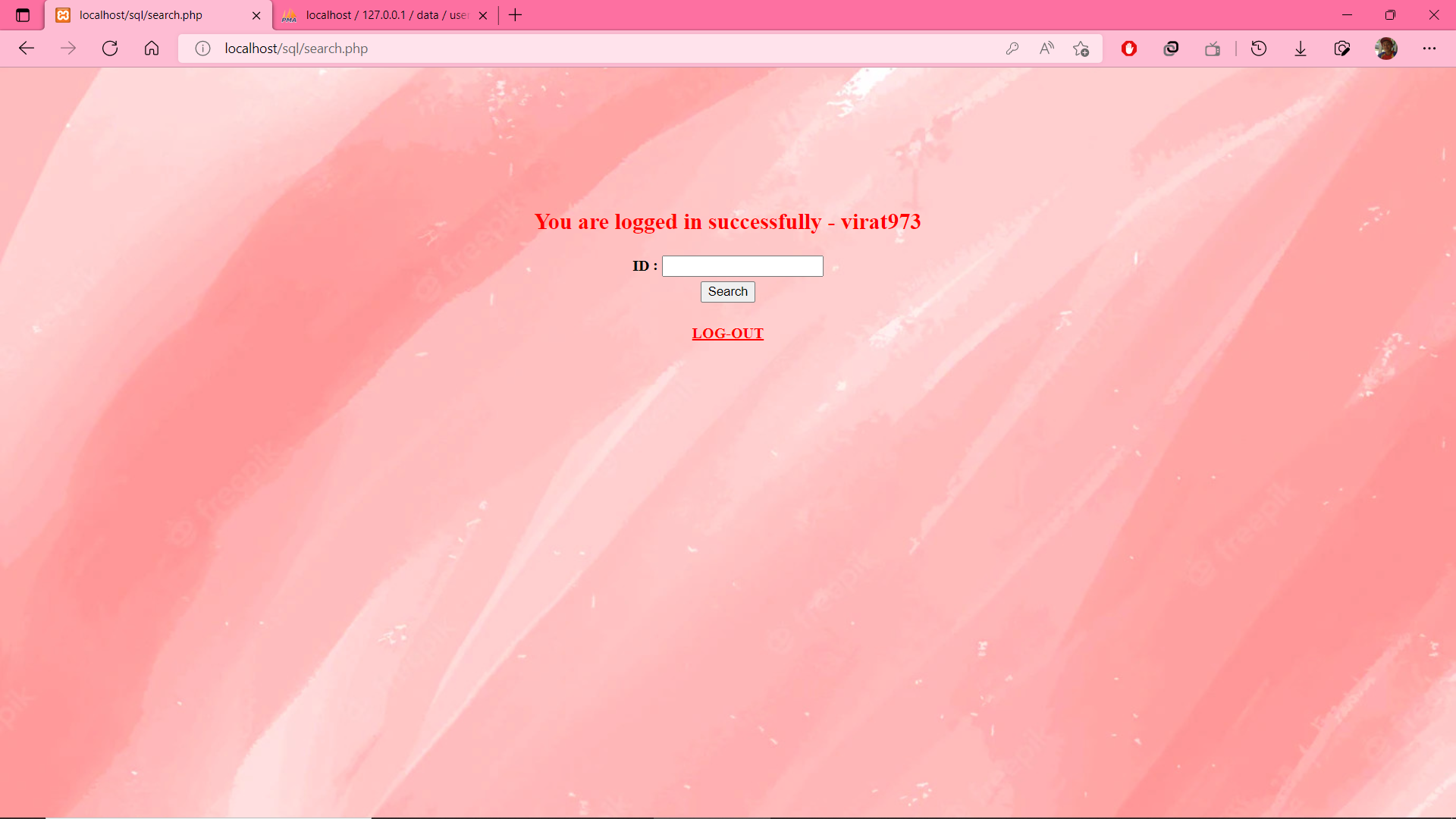
This the registration site created by us where our details gets stored in **users table** in MYSQL database which is administrated by PHPMYadmin.



In Login page instead of giving the giving the correct password, which may not be known by the attackers if he inputs **a' OR 1=1 # ,**



This value 1=1 is always true anything followed by this is commented so password field becomes true always and we are able to login.



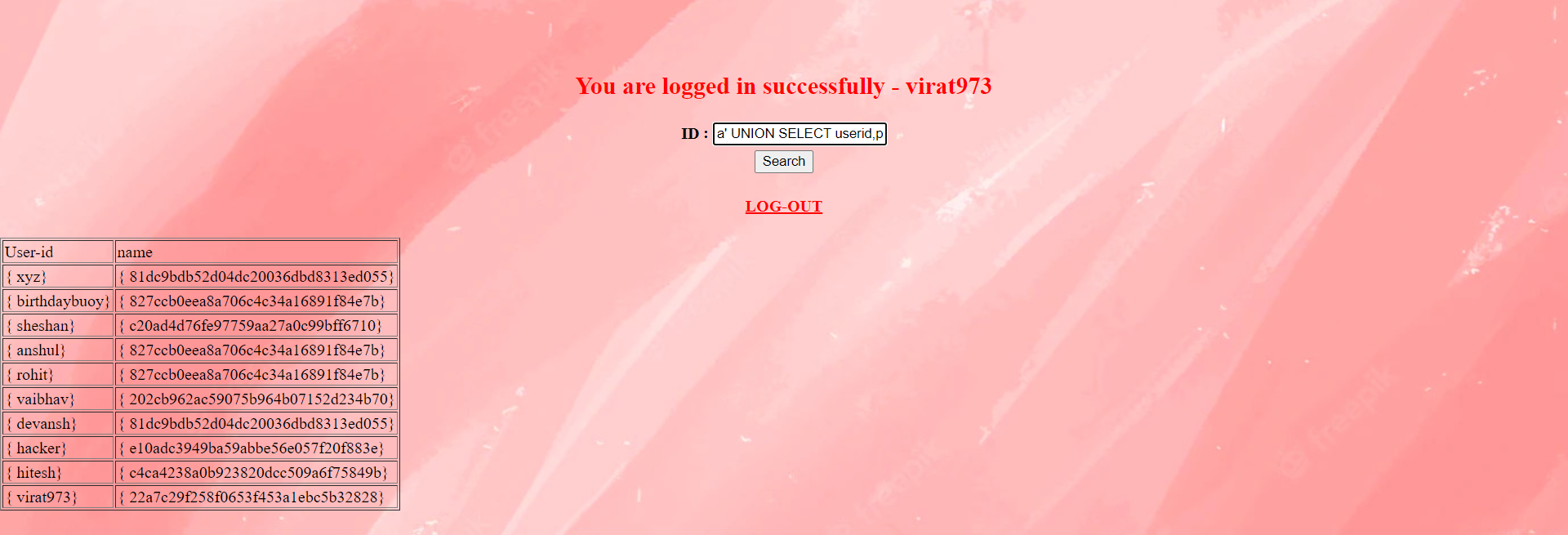
**Step-2:**

The Search bar in the website when entered our userid it prints details about us.



But when attacker enters **a' UNION SELECT userid,password FROM users#**

The sql query makes the php site to print userid and password of all users. Confidential data such as passwords of other users can be leaked by this kind of UNION based sql injection attacks.



**Step-3:**

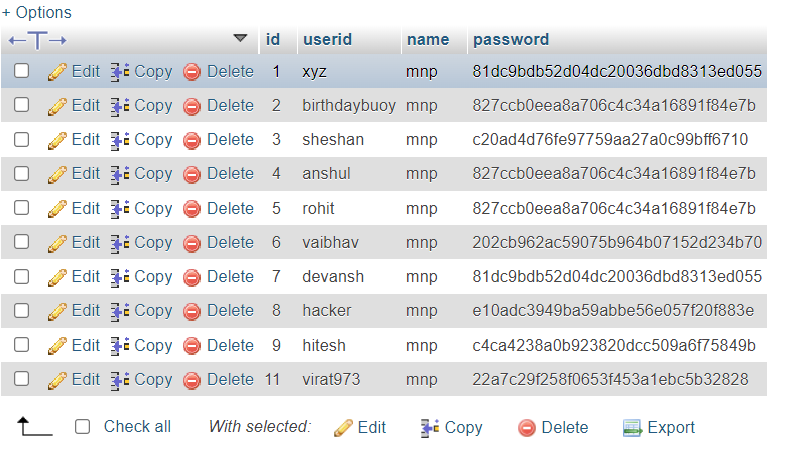
**Modify the data in the database.**

If the attacker uses this command **1';UPDATE users SET name='mnp' #**

All the users name will be changed to mnp , which leads to information lose and several loss to the organization.



Since it’s an update command nothing prints in the website but when we look at the our database all the names get changed to mnp. i.e This query is used to update the 'users' table and set name of all users to 'mnp'.



**Step-4:**

**Extracting details about other tables in the database.**

If the attacker uses this command **a' UNION SELECT table\_name,null FROM information\_schema.tables #**, in the search tab  **,**then all the tables names present in the database will be printed in the website.



prints all the table names present in database at the moment.

**Step-5:**

**Mitigation: using prepared statements.**

We have changed our SQL query statement using prepared statements, where the value entered in the search query is sanitized , stored in a variable and then binded with the SQL query prepared statement.

**The prepared statement used as countermeasure:**

if(isset($\_POST['search']))

    {

        if(!empty($\_POST['search']))

        {

        $search= mysqli\_real\_escape\_string($con,"%{$\_POST['search']}%");

        //mysqli\_real\_escape\_string is used to santize the output.

        // by inserting ? we are seperating user input from query both won't know same information

        $stmt =$con->prepare("SELECT userid,name FROM users WHERE id LIKE ?");

        //The bind\_param() method is where you attach variables to the dummy values in the prepare template.

        // s means just a piece of data

        $stmt->bind\_param("s",$title);

        //tells which param inside the bind\_param

        $title = $search;

        // execute the stmt

        $stmt->execute();

}

When we use prepared statement, where the variable entered by the user is sanitized if any unwanted symbols present in it other than string characters will be avoided and the variable is binded to the query statement, which provided prevention against using comments in the query.

After using prepared statements when a user tries to input anything invalid such as SQL injection codes in the input form it throws an error.

### Here it shows: \* Please enter your own userid.Don't mess around by trying to hack using sql injection .

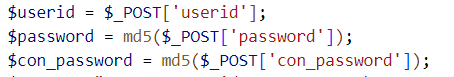
This is one of the mitigation provided by us.



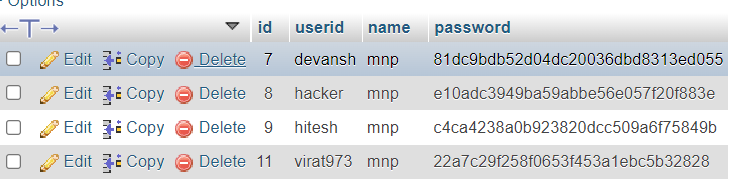
**Using cryptography algorithm to encrypt sensitive data**

We can use any cryptography algorithm say **MD5** to encrypt our password so that even if accessible by attacker he cannot perform anything with it without decrypting using keys.

**MD5 Encrpytion OF password:**



so the password in the database will be encrypted so no use for the attacker.



**Conclusion:**

SQL injection attack in our PHP websites can we mitigated by using prepared statements i.e binding variables in query statements, encrypting confidential information in our Database, Validating the input given by the user in the form, sanitizing the input, limiting the privileges of the user so that we can protect our data in the database from the SQL injection attackers.