

LAB FOR SESSION 19**PHP Programming****A. OBJECTIVES**

1. Using session in PHP to update information
2. Understanding SQL Injection and XSS error.

B. STEP BY STEP EXERCISES

1. Create an account update page (update_customer.php) that allows users to change their information after he/she logs in successfully:

When clicking on the "Hi ..." link on the menu, the account update page (Update_customer.php) will be displayed with the interface below and the information of the user displayed:

The screenshot displays a web application interface for updating a user profile. The top navigation bar features a dark blue header with contact information (+84 7103 888 999, admin@gmail.com) and social media icons. Below this, a blue bar contains the 'ABC abc' logo and navigation links: Account, Cart, 'Hi, finn' (highlighted with a yellow circle), and Logout. A dark blue bar below contains links: Home, Introduction, Management, Cart, Feedback, and Contact, along with a search bar. The main content area is titled 'Update Profile' and contains a form with the following fields: Username(*) (finn), Email(*) (finn@gmail.com), Password(*) (empty), Confirm Password(*) (empty), Full name(*) (Nguyen Hung Cuong), Address(*) (Can Tho), and Telephone(*) (0903888999). An 'UPDATE' button is located at the bottom of the form.

Note:

- o “Update_customer.php” page must be embedded in the homepage
- o Username and email is not modified.
- o Password does not need to display the current password.

When users press the update button, please check the following information before updating information.

- o The password and confirm password must be the same and must be md5 encrypted before saving to the database
- o If the user does not enter a password, there is no need to update the password

Hint:

Step 1: Gets user data to display on form via session variable

```
//Get customer information
$query = "SELECT CustName, Address, email, telephone
        FROM customer
        WHERE Username = '" . $_SESSION["us"] . "'";
$result = mysqli_query($conn, $query) or die(mysqli_error($conn));
$row = mysqli_fetch_array($result, MYSQLI_ASSOC);

$us = $_SESSION["us"];
$email = $row["email"];
$fullname = $row["CustName"];
$address = $row["Address"];
$telephone = $row["telephone"];
```

Step 2: Display information on the form

```
<div class="form-group">
  <label for="lblEmail" class="col-sm-2 control-label">Email(*): </label>
  <div class="col-sm-10">
    <label class="form-control" style="font-weight:400"><?php echo $email; ?></label>
  </div>
</div>

<div class="form-group">
  <label for="lblHoten" class="col-sm-2 control-label">Full name(*): </label>
  <div class="col-sm-10">
    <input type="text" name="txtFullname" id="txtFullname" value="<?php echo $fullname; ?>"
    class="form-control" placeholder="Enter Fullname, please"/>
  </div>
</div>
```

Step 3: Write a function to check the information

```
function check(){
    if($_POST['txtFullname']==""||$_POST['txtAddress']==""){
        return "<li>Enter Fullname or Address</li>";
    }
    elseif(strlen($_POST['txtPass1'])>0 && strlen($_POST['txtPass1'])<=5){
        return "<li>Password is greater than 5 characters</li>";
    }
    elseif($_POST['txtPass1']!= $_POST['txtPass2'])
    {
        return "<li>Password and Confirm Pass do not match</li>";
    }
    else{
        return "";
    }
}
```

Step 4: Update information when the user presses the "Update" button

```
//Update information when the user presses the "Update" button
if(isset($_POST['btnUpdate'])){
    $fullname=$_POST['txtFullName'];
    $address = $_POST['txtAddress'];
    $tel = $_POST['txtTel'];

    $test = check();
    if($test==""){
        //Customer changes password
        if($_POST['txtPass1']!=""){
            $pass = md5($_POST['txtPass1']);

            $sq = "UPDATE customer
SET CustName='$fullname', Address='$address',
telephone='$telephone', Password='$pass'
WHERE Username = '" . $_SESSION['us'] . "'";

            mysqli_query($conn,$sq) or die(mysqli_error($conn));
        }
        //Customer does not change password
        else{
            $sq = "UPDATE customer
SET CustName='$fullname', Address='$address',
telephone='$telephone' WHERE Username = '" . $_SESSION['us'] . "'";
            mysqli_query($conn, $sq) or die(mysqli_error($conn));
        }
        echo '<meta http-equiv="refresh" content="0;URL=index.php"/>';
    }else{
        echo $test;
    }
}
```

2. Perform an attack on the site using SQL Injection to successfully login to the system

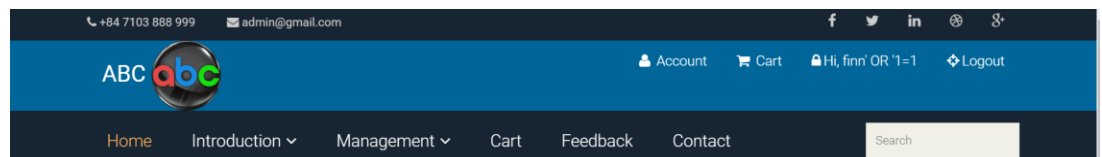
Hint:

From the login form and enter the following information:

Username: **finn' OR '1=1**

(Where: *finn* is account in database)

Password: type any thing



Fix error: Using `mysqli_real_escape_string()`

```
$us = mysqli_real_escape_string($conn, $us);
```

3. Perform XSS website attacks using `<script>` tags

Hint:

From the Add_Category.php page, enter the information as shown below:

The top screenshot shows the 'Adding Category' form with the following fields and values:

- Category ID(*): C0010
- Category Name(*): <script>alert("Hack")</script>
- Description(*): Description

Below the fields are two buttons: 'ADD NEW' and 'IGNORE'.

The bottom screenshot shows a browser window at localhost:1000/code/?page=category_management. A JavaScript alert box is displayed with the text 'localhost:1000 says Hack' and an 'OK' button.

Fix error: Prevent XSS from using the htmlspecialchars () function

```
else{
    $id = htmlspecialchars(mysqli_real_escape_string($conn, $id));
    $name = htmlspecialchars(mysqli_real_escape_string($conn, $name));
    $des = htmlspecialchars(mysqli_real_escape_string($conn, $des));
    $sq="Select * from category where Cat_ID='$id' or Cat_Name='$name'";
    $result = mysqli_query($conn,$sq);
    if(mysqli_num_rows($result)==0)
    {
        mysqli_query($conn, "INSERT INTO category (Cat_ID, Cat_Name, Cat_Des) VALUES ('$id','$name','$des')");
        echo '<meta http-equiv="refresh" content="0;URL=Category_Management.php"/>';
    }
    else
    {
        echo "<li>Duplicate category ID or Name</li>";
    }
}
```