

PRACTICE REPORT
WEB PROGRAMMING LAB WORKS
MODULE 9
“Session”



Assembled by:
ONIC AGUSTINO
L200234275
X

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
TAHUN 2024/2025

PRACTICE

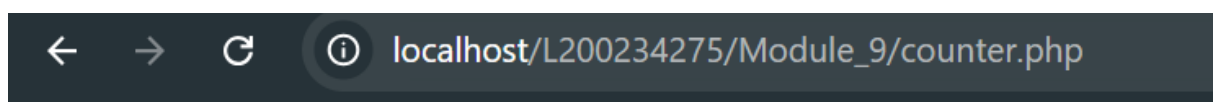
1. Trial 1 (Session)

```
Module_9 > counter.php > html
1  <?php
2  session_start();
3  error_reporting(error_level: E_ALL ^ E_NOTICE);
4  ?>
5
6  <!DOCTYPE html>
7  <html lang="en">
8  <head>
9      <meta charset="UTF-8">
10     <meta name="viewport" content="width=device-width, initial-scale=1.0">
11     <title>Accessing Session Data</title>
12 </head>
13 <body>
14     <?php
15         $_SESSION['counter']++;
16         $_SESSION['visitor_name'] = "Adul";
17         echo "Welcome " . $_SESSION['visitor_name'] . "<br>";
18         echo "You have visited this page many times: " . $_SESSION['counter'] . " kali.";
19     ?>
20 </body>
21 </html>
```

Picture 1.1 The Code.

(Explanation)

This PHP code initializes a session, suppresses notices in error reporting, and creates an HTML page where it tracks the number of times a visitor accesses the page by using a session variable while also storing their name; every time the page reloads, the counter increases, and the visitor's name ("Adul") is displayed along with the number of visits.



Welcome Adul
You have visited this page many times: 2 kali.

Picture 1.2 The Output.

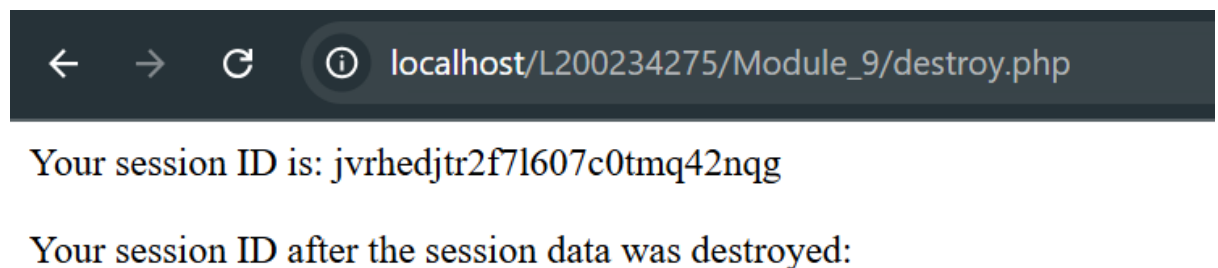
2. Trial 2 (Session Destroy)

```
Module_9 > destroy.php > ...
1  <?php
2  session_start();
3  $id_session = session_id();
4  echo "Your session ID is: " . $id_session;
5  echo "<br><br>";
6  session_destroy();
7  $id_session2 = session_id();
8  echo "Your session ID after the session data was destroyed: ".$id_session2;
9  ?>
```

Picture 2.1 the code.

(Explanation)

This PHP script starts a session and retrieves the current session ID, displaying it to the user. Afterward, the script destroys the session using `session_destroy()`, but since the session ID persists until a new session starts, the script prints the session ID again, which remains unchanged even after destroying the session data. This demonstrates that `session_destroy()` removes session data but does not alter the session ID unless a new session is initiated.



Picture 2.2 the results.

3. Attempt 3 (Login)

				username	password	name	status
<input type="checkbox"/>	Edit	Copy	Delete	admin	1234	Admin	Administrator
<input type="checkbox"/>	Edit	Copy	Delete	adul	1111	Adul Adullah	Member

Picture 3.1 the database

```

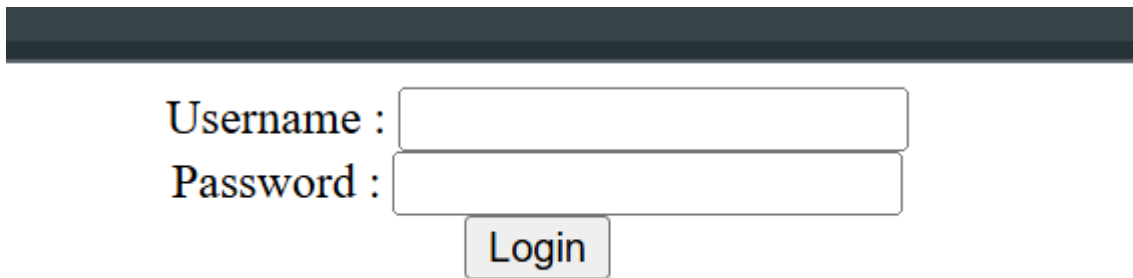
Module_9 > login.php > ...
1  <?php
2  session_start();
3  error_reporting(error_level: E_ALL ^ E_NOTICE ^ E_DEPRECATED);
4  $conn = mysqli_connect(hostname: 'localhost', username: 'root', password: '', database: 'informatika');
5
6  $username = $_POST['username'] ?? '';
7  $password = $_POST['password'] ?? '';
8  $submit = $_POST['submit'] ?? '';
9
10 if ($submit) {
11     $sql = "SELECT * FROM users WHERE username='$username' AND password='$password'";
12     $query = mysqli_query(mysql: $conn, query: $sql);
13     $row = mysqli_fetch_array(result: $query); // Changed $result to $row
14     if ($row['username'] != "") { // Corrected $result to $row
15         $_SESSION['username'] = $row['username'];
16         $_SESSION['status'] = $row['status'];
17     }
18     <script language script="javascript">
19         alert('You are logged in as <?php echo $row['username']; ?>');
20         document.location = 'loginresult.php';
21     </script>
22 <?php
23 } else {
24     <script language script="javascript">
25         alert('Login Failed');
26         document.location = 'login.php';
27     </script>
28 <?php
29 }
30 }
31 }
32 ?>
33 <form action="login.php" method="post">
34     <p align="center">Username : <input type="text" name="username"><br>
35     Password : <input type="password" name="password"><br>
36     <input type="submit" name="submit" value="Login">
37 </p>
38 </form>
39

```

Picture 3.2 the code login

(Explanation)

This PHP script manages a simple login system using sessions and a MySQL database. It starts a session, suppresses certain error types, and establishes a connection to a database named "informatika." It retrieves user inputs for username and password through the POST method and, upon form submission, queries the database for matching credentials. If a user is found, their session is set with username and status, followed by a JavaScript alert confirming the login and redirecting to `loginresult.php`. If login fails, a JavaScript alert informs the user and redirects them to `login.php`. The HTML form at the end allows users to input their credentials and submit the login request.



Username :

Password :

Login

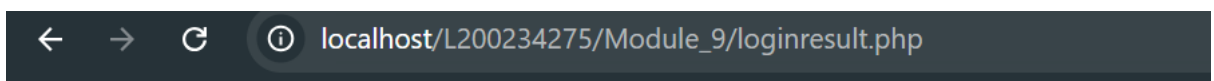
Picture 3.3 the result code for login.

```
Module_9 > loginresult.php > a
1  <?php
2  session_start();
3  echo "You have successfully logged in as " . $_SESSION['username'] . " And you are registered as " . $_SESSION['status'] . "<br>";
4
5  ?>
6  <br>
7  Please log out by clicking the link <a href="logout.php">Here</a>
```

Picture 3.4 the code loginresults.

(Explanation)

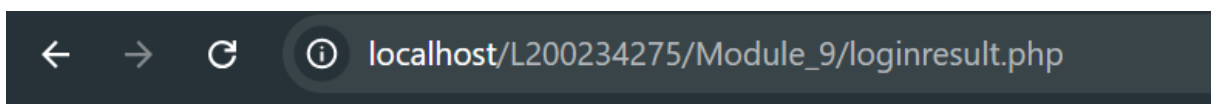
This PHP script starts a session and retrieves the stored session variables username and status, which were set during the login process. It then displays a message confirming the user's successful login, along with their registration status. A logout link (logout.php) is provided to allow the user to end their session. If no session data exists, the displayed message may be incomplete.



← → ↻ ⓘ localhost/L200234275/Module_9/loginresult.php

You have successfully logged in as admin And you are registered as Administrator

Please log out by clicking the link [Here](#)



← → ↻ ⓘ localhost/L200234275/Module_9/loginresult.php

You have successfully logged in as adul And you are registered as Member

Please log out by clicking the link [Here](#)

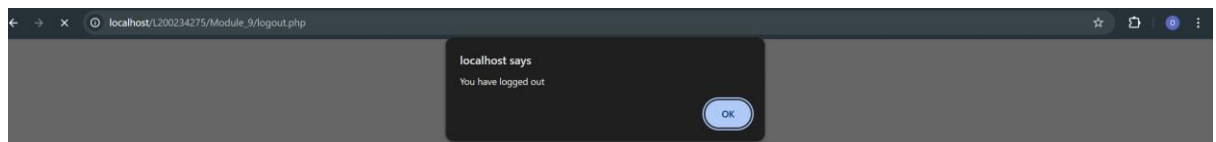
Picture 3.5 the result of loginresult.

```
Module_9 > 🐘 logout.php > ...
1  <?php
2  session_start();
3  session_destroy();
4  ?>
5  <script language script="javascript">
6      alert("You have logged out");
7      document.location = 'login.php';
8  </script>
```

Picture 3.6 the code logout.

(Explanation)

This PHP script starts a session and then immediately destroys all session data using `session_destroy()`, effectively logging the user out. After that, a JavaScript alert notifies the user that they have logged out, and the page redirects them to `login.php`, where they can log in again if needed. This ensures a smooth logout experience by both clearing session variables and guiding the user back to the login page.



Picture 3.7 the result of logout.

ASSIGNMENT

1. Using the “user” table above, create a login application whgere the appearance of the website between users with status as “Administrator” will be different from the appearance of the website for users with status as “Member”.

```
Module_9 > task > task.php > ...
1  <?php
2  session_start();
3  error_reporting(error_level: E_ALL ^ E_NOTICE ^ E_DEPRECATED);
4  $conn = mysqli_connect(hostname: "localhost", username: "root", password: "", database: "informatika");
5  $username = $_POST['username'] ?? '';
6  $password = $_POST['password'] ?? '';
7  $submit = $_POST['submit'] ?? '';
8
9  if ($submit) {
10     // Query to check user credentials
11     $query = "SELECT * FROM users WHERE username='$username' AND password='$password'";
12     $result = mysqli_query(mysql: $conn, query: $query);
13     $row = mysqli_fetch_assoc(result: $result);
14     if ($row) {
15         $_SESSION['username'] = $row['username'];
16         $_SESSION['status'] = $row['status'];
17
18         if ($row['status'] == 'Administrator') {
19             header(header: "Location: admin_dashboard.php");
20         } elseif ($row['status'] == 'Member') {
21             header(header: "Location: member_dashboard.php");
22         }
23         exit;
24     } else {
25         echo "<script>alert('Login Failed');</script>";
26     }
27 }
28 ?>
29 <!DOCTYPE html>
30 <html lang="en">
31 <head>
32     <meta charset="UTF-8">
33     <meta name="viewport" content="width=device-width, initial-scale=1.0">
34     <title>Login</title>
35 </head>
36 <body>
37     <form action="task.php" method="post">
38         <p align="center">
39             Username: <input type="text" name="username"><br>
40             Password: <input type="password" name="password"><br>
41             <input type="submit" name="submit" value="Login">
42         </p>
43     </form>
44 </body>
45 </html>
```

Picture 1.1 the code.

(Explanation)

This PHP script implements a login system using sessions and a MySQL database. It begins by starting a session and suppressing certain error messages. Then, it establishes a connection to the database named **"informatika"** and retrieves user input for **username** and **password** through the POST method. If the form is submitted, it queries the database to check whether the credentials match an existing user. If a valid user is found, their session is set with username and status. Based on the user's status, the script redirects them to either **"admin_dashboard.php"** for administrators or

"**member_dashboard.php**" for regular members, ensuring that different user roles are directed accordingly. If login fails, a JavaScript alert notifies the user. The HTML form at the end allows users to enter their credentials and attempt login again.

Username:

Password:

Picture 1.2 the results.

```
Module_9 > task > admin_dashboard.php > ...
1  <?php
2  session_start();
3  if ($_SESSION['status'] != 'Administrator') {
4      header(header: "Location: task.php");
5      exit;
6  }
7  ?>
8  <!DOCTYPE html>
9  <html lang="en">
10
11  <head>
12      <meta charset="UTF-8">
13      <meta name="viewport" content="width=device-width, initial-scale=1.0">
14      <title>Admin Dashboard</title>
15  </head>
16
17  <body>
18      <h1>Welcome, Administrator!</h1>
19      <p>This is the admin dashboard.</p>
20      <a href="logout.php">Logout</a>
21  </body>
22
23  </html>
```

Picture 1.3 the code dashboard admin.

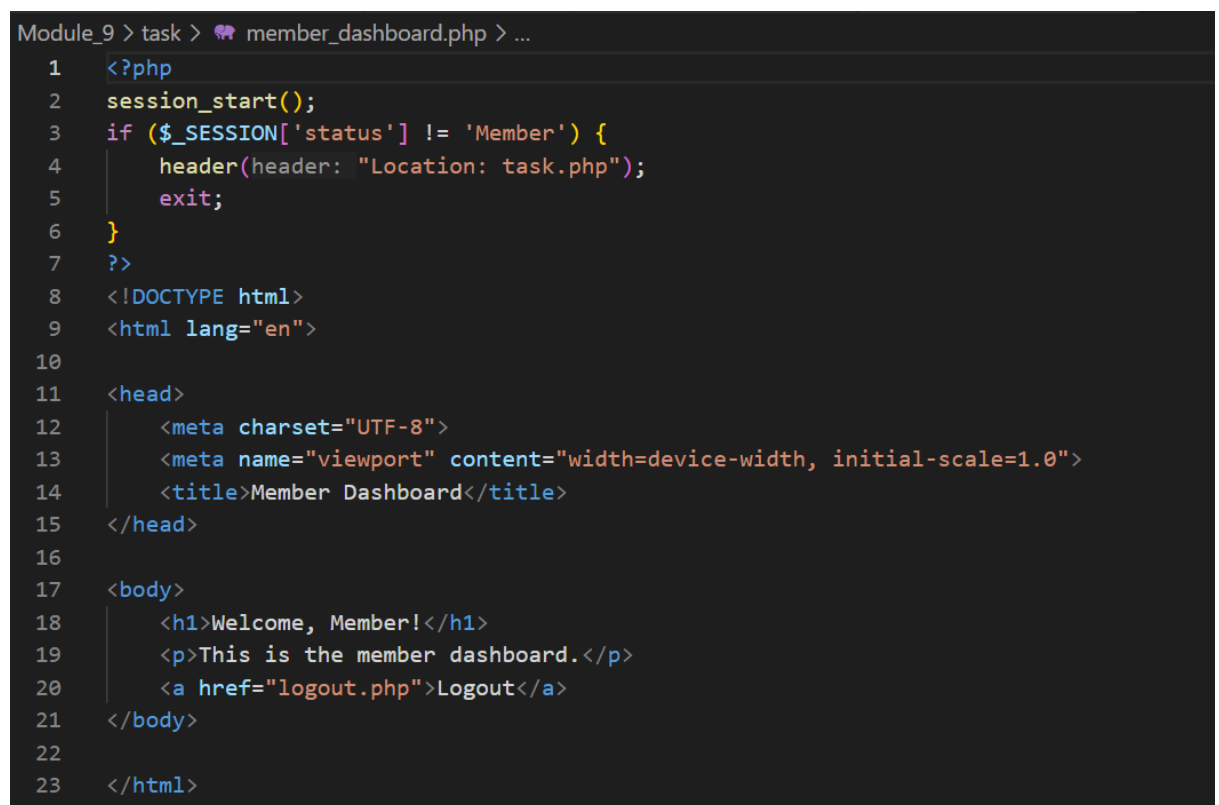
(Explanation)

This PHP script ensures that only users with the "Administrator" status can access the admin dashboard. It starts a session and checks the status session variable—if the user is not an administrator, they are redirected to task.php and the script exits to prevent further execution. If access is granted, the script generates an HTML page displaying a

welcome message for the administrator, along with a logout link (logout.php). This helps maintain security by restricting access based on user roles.



Picture 1.4 the results of dashboard admin.

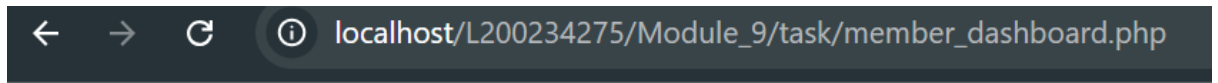


Picture 1.5 the code dashboard member.

(Explanation)

This PHP script ensures that only users with the "Member" status can access the member dashboard. It starts a session and checks the status session variable—if the user is not a member, they are redirected to task.php and the script exits to prevent further

execution. If access is granted, the script generates an HTML page displaying a welcome message for the member, along with a logout link (logout.php). This helps maintain security by restricting access based on user roles.



Welcome, Member!

This is the member dashboard.

[Logout](#)

Picture 1.7 the results of dashboard member.

```
Module_9 > task > 🐘 logout.php
1  <?php
2  session_start();
3  session_destroy();
4  header(header: "Location: task.php");
5  exit;
6  |
```

Picture 1.8 the code.

(Explanation)

This PHP script starts a session and then immediately destroys all session data using `session_destroy()`, effectively logging the user out. Afterward, it redirects them to `task.php` using `header("Location: task.php");`, ensuring a seamless transition to another page. The `exit;` statement is used to stop further script execution after the redirection, preventing any unintended output.

The result is back to menu login or back to `task.php`

