**Experiment 12**

**Title :** Write a program to create and handle a session, cookie in PHP

**Aim :** To make student understand the concepts of session and cookies in PHP

**Program to create and handle session and cookie in PHP**

Sessions and cookies are two fundamental methods for storing data in PHP, often used to maintain state and manage user information across multiple pages.

**Cookies**

Definition: A cookie is a small piece of data stored on the client’s computer by the web browser. Cookies are sent to the server with each request.

Key Characteristics:

* Storage: Cookies can store data as key-value pairs and are limited in size (typically up to 4KB).
* Expiration: Cookies can have an expiration date, after which they are deleted automatically. If no expiration is set, cookies are considered session cookies and are deleted when the browser is closed.
* Scope: Cookies are accessible across different pages of the same domain.

**Creating a Cookie in PHP**:

We can set a cookie using the setcookie() function before any output is sent to the browser.

**Accessing a Cookie**:

We can access cookies using the $\_COOKIE superglobal array.

**Deleting a Cookie**:

To delete a cookie, set its expiration date to a time in the past.

**Sessions**

Definition: A session is a way to store data on the server for individual users, allowing you to track user activity and maintain user-specific data across multiple pages.

Key Characteristics:

* Storage: Sessions store data on the server, which can handle larger amounts of information compared to cookies.
* Expiration: Sessions usually expire after a period of inactivity (default is 24 minutes in PHP) or when the user closes the browser.
* Session ID: When a session is created, PHP generates a unique session ID, which is sent to the user as a cookie or URL parameter.

**Starting a Session:**

We must start a session before using session variables.

**Accessing Session Variables:**

We can access session variables using the $\_SESSION superglobal array.

**Destroying a Session:**

To end a session and clear all session variables

Difference between cookies and sessions

|  |  |  |
| --- | --- | --- |
| **Feature** | **Cookies** | **Sessions** |
| Storage | Client-side (browser) | Server-side |
| Data Size | Limited (typically 4KB) | Larger storage capacity |
| Expiration | Can be persistent or session-based | Usually expires after inactivity or when the browser is closed |
| Security | Less secure (data is stored on the client) | More secure (data is stored on the server) |
| Accessibility | Accessible across the entire domain | Accessible only while the session is active |

Program using **session**

Step 1: Start a Session and Store Data

Create a file named session\_example.php

<?php

// Start the session

session\_start();

// Store session variables

$\_SESSION["username"] = "JohnDoe";

$\_SESSION["email"] = "johndoe@example.com";

echo "Session variables are set.";

?>

Step 2: Access Session Data

Create another file named access\_session.php

<?php

// Start the session

session\_start();

// Access session variables

if (isset($\_SESSION["username"])) {

echo "Welcome back, " . $\_SESSION["username"] . "!";

echo "<br>Your email is: " . $\_SESSION["email"];

} else {

echo "No session data found. Please log in.";

}

?>

Step 3: Destroy the Session

Create a file named destroy\_session.php

<?php

// Start the session

session\_start();

// Remove all session variables

session\_unset();

// Destroy the session

session\_destroy();

echo "Session destroyed. You have been logged out.";

?>

Program using **cookies**

Step 1: Set a Cookie

Create a file named cookie\_example.php

<?php

// Set a cookie that lasts for 1 hour

setcookie("username", "JohnDoe", time() + 3600, "/"); // "/" makes it available across the whole domain

echo "Cookie has been set.";

?>

Step 2: Access the Cookie

Create another file named access\_cookie.php

<?php

// Check if the cookie is set

if (isset($\_COOKIE["username"])) {

echo "Welcome back, " . $\_COOKIE["username"] . "!";

} else {

echo "No cookie found. Please log in.";

}

?>

Step 3: Delete the Cookie

Create a file named delete\_cookie.php

<?php

// Delete the cookie by setting its expiration date to the past

setcookie("username", "", time() - 3600, "/");

echo "Cookie has been deleted.";

?>

**Conclusion:-**

Thus we have studied session and cookies in php and how they work.