**Experiment No: 13**
**Write simple PHP program to**
1. **Set cookies and read it**
2. **Demonstrate session management**

**Resources required:**

| **Hardware** | **Software** |
|---|---|
| Computer System | XAMPP |

**Practical Significance:**

PHP cookie is a small piece of information which is stored at client browser. It is used to recognize the user. PHP session is used to store and pass information from one page to another temporarily (until user close the website). PHP session technique is widely used in shopping websites where we need to store and pass cart information e.g. username, product code, product name, product price etc from one page to another. PHP session creates unique user id for each browser to recognize the user and avoid conflict between multiple browsers.

**Theoretical Background:**
**Cookies:**
Cookie is created at server side and saved to client browser. Each time when client sends request to the server, cookie is embedded with request. Such way, cookie can be received at the server side. In short, cookie can be created, sent and received at server end.
A cookie is a small file with the maximum size of 4KB that the web server stores on the client computer.
A cookie can only be read from the domain that it has been issued from. Cookies are usually set in an HTTP header but JavaScript can also set a cookie directly on a browser.

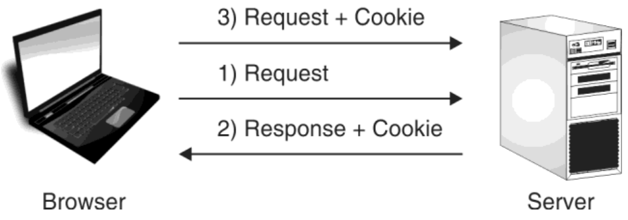


Fig. Cookies

There are three steps involved in identifying returning users :
1. Server script sends a set of cookies to the browser. For example : name, age or identification number etc.
2. Browser stores this information on local machine for future use.

3. When next time browser sends any request to web server then it sends those cookies information to the server and server uses that information to identify the user.

There are two types of cookies :

1. Session Based which expire at the end of the session.
2. Persistent cookies which are written on hard disk.

PHP provides a inbuilt function setcookie(), that can send appropriate HTTP header to create the cookie on the browser.

Only name argument is must but it is better to pass value, expires and path to avoid any ambiguity.

Syntax : setcookie(name, value, expire, path, domain, secure, HttpOnly);

**Attributes of Cookie:**

| Attribute of cookies | Description |
|---|---|
| Name | The unique name is given to a particular cookie. |
| Value | The value of the cookie. |
| Expires | The time when a cookie will get expire. When it reaches to its expiration period cookies is deleted from browser automatically. If value is set to zero, it will only last till the browser is running its get deleted when the browser exits. |
| Path | The path where browser to send the cookies back to the server. If the path is specified, it will only send to specified URL else if it is stored with “/” the cookie will be available for all the URL’s on the server. |
| Domain | The browser will send the cookie only for URLs within this specified domain. By default is the server host name. |
| Secure | If this field is set, the cookie will only be sent over https connection. By default it is set to false, means it is okay to send the cookie over an insecure connection. |
| HttpOnly | This field, if present, tells the browser that it should only make the cookie assessable only to scripts that run on the Web server (that is, via HTTP). Attempts to access the cookie through JavaScript will be rejected. |

**Session:**

Session are used to store important information such as the user id more securely on the server where malicious users cannot temper with them.

To pass values from one page to another.

Sessions are the alternative to cookies on browsers that do not support cookies.

You want to store global variables in an efficient and more secure way compared to passing them in the URL

You are developing an application such as a shopping cart that has to temporary store information with a capacity larger than 4KB.

Set Session Variables

```
<?php
    session_start();
$_SESSION["username"] = "abc";
?>
```

A PHP function session_unset() is used to remove all session variables and session_destroy() is used to destroy session.

**Practical related questions:**

1. How to initiate a session in PHP ?
   By using the method session_start()

2. How to register a variable in PHP session ?
   Use _SESSION array.
   e.g. $_SESSION["variable"] = "Data"

3. What is the difference between PHP session and Cookie ?
   Session data is stored on the server
   Cookie data is stored on the visitors browsers

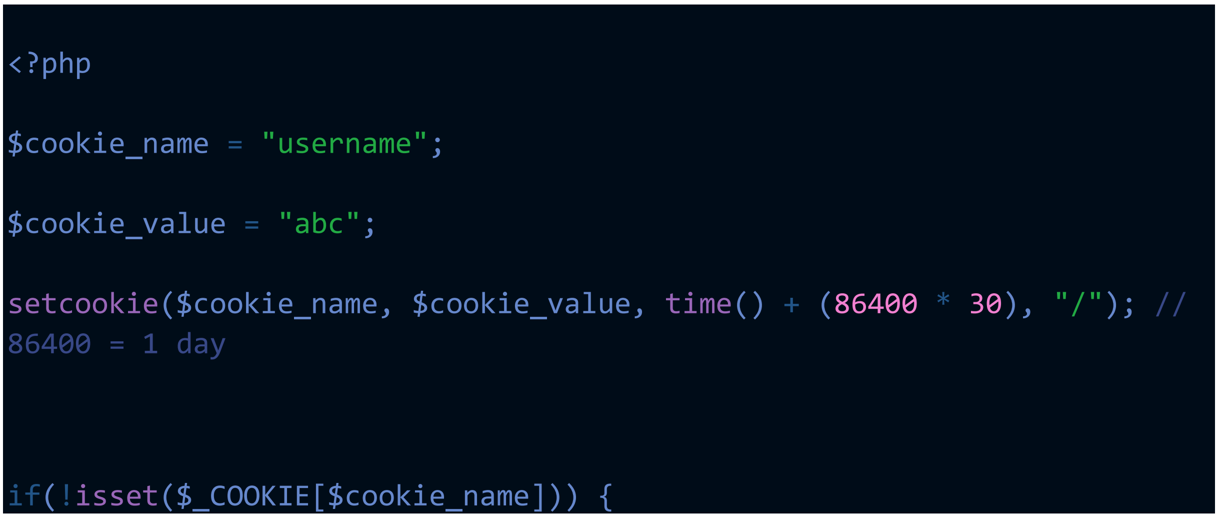
4. What is the difference between session_unregister () and session_unset()?
   Session_unregister is used to unregister one global variable
   Session_unset is used to free all session variables

5. What is the default session time in PHPH?
   24 minutes

6. Is it possible to destroy a cookie ()?
   Yes, by using the unset() method

**Exercise:**

1. Write a program that demonstrate use of cookies.



```
<?php

$cookie_name = "username";

$cookie_value = "abc";

setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/"); //
86400 = 1 day



if(!isset($_COOKIE[$cookie_name])) {
    echo "Cookie name '" . $cookie_name . "' is not set!";

} else {

    echo "Cookie '" . $cookie_name . "' is set!<br>";

    echo "Value is: " . $_COOKIE[$cookie_name];

}

?>

<!-- Output -->
<!-- Cookie 'username' is set!
Value is: abc -->
```

2. Write a PHP program that demonstrate use of session.

```
<?php

session_start();
$_SESSION["username"] = "abc";

    if(isset($_SESSION["username"])) {
        echo "Session variable is set";
        echo "<br> variable is " .$_SESSION["username"];
    } else {
        echo "Session variable is not set";
    }

?>

<!-- output -->
<!-- Session variable is set
variable is abc  -->
```