

Handling Session and Cookies

- ❑ Internet is stateless place- that is, Web pages
- ❑ don't store data.
- ❑ Next time we load web page into browser is the
- ❑ same as the first.
- ❑ Web application work in multipage way: storing
- ❑ data about user that persist from page to page.

Difference between session and cookie?

- ☐ Cookie is a small amount of information stored at client side.
- ☐ Session data are stored at server side.

- ☐ cookie can store LIMITED Amount of data (max 4KB)
Session-Amount of data to be stored is NOT LIMITED

- ☐ cookies can store only STRING type of data.
- ☐ session store data as OBJECTS

- ☐ cookie data is less secure
- ☐ session data is more secure

- ☐ Cookie data is available up to expiration date.
- ☐ Session data is available for the browser run.
- ☐ We lose the session information when we close the browser.

Setting a cookie

We set cookies on the user's machine with the PHP `setcookie()` function

`setcookie(name, value, expire);`

Here , **Name** of the cookie, **Value** content that we actually want to store, **Expiry** a expiration time in seconds after which cookie will automatically expire/become inaccessible

```
<html>
<head>
<title> Setting a cookie</title>
</head>

<body>
<?php
setcookie("name","ACE Engineering college",time()+120);
echo "Done!!";
?>
</body>
</html>
```

Reading a cookie

Once cookie is set, when we load a page the cookie is sent to the server from the user's machine. We can access the cookie on the next page load with the `$_COOKIE[]` array.

```
<html>
<head>
<title> Reading a cookie</title>
</head>

<body>
<?php
$college=$_COOKIE['name'];
echo $college;
?>
</body>
</html>
```

Deleting cookies

We can set the expiration time of a cookie to some time in the past and that will make the browser delete cookie.

```
<html>
<head>
<title> Deleting a cookie</title>
</head>

<body>
<?php
setcookie("name","ACE Engineering college",time()-120);
echo "Deleted!!";
?>
</body>
</html>
```

Storing data in Sessions

- The way of storing data on the server to be used across multiple pages is to use SESSION.
- SESSION store information about the user and browser remembers the information to all pages.
- To work with session, we start session by calling session_start()
- To store data in the session we use \$_SESSION[] array

```
<html>
<head>
<title> Storing data in sessions</title>
</head>
```

```
<body>
<?php
session_start();
$_SESSION['user']='Sachin';
$_SESSION['purchase']='12000';
?>
</body>
</html>
```

Retrieving data from sessions

```
<html>
<head>
<title> Retrieving data from sessions</title>
</head>

<body>
<?php
session_start();
echo "thank you, ".$_SESSION['userID']." You have purchased ".$_SESSION['purchase']." worth";
?>
</body>
</html>
```

Writing a Hit counter using Sessions

Example of showing how session preserve data between page accesses.

How many times the user has been to a web page.

We store data in the session key "count" set to 0.

When the user reloads the page, we increment the value stored in the session.

```
<html>
<head>
<title> A Session hit counter</title>
</head>

<body>
<?php
session_start();

if(isset($_SESSION['count']))
{
    $_SESSION['count']++;
}
else
{
    $_SESSION['count']=0;
}
echo "Page views: ".$_SESSION['count'];
?>
</body>
</html>
```