Cookies and Session

- A cookie is a small file that the server embeds on the user's computer to identify a user
- Each time the same computer requests a page with a browser, it will send the cookie too
- With PHP, you can both create and retrieve cookie values
- provides a general approach to storing information about sessions on the browser system itself
- The server is given this information when the browser makes subsequent requests for Web resources from the server
- allows the server to present a customized interface to the client
- also allow the server to connect requests from a particular client to previous requests, thereby connecting sequences of requests into a session

Cookies and Session

- At the time it is created, a cookie is assigned a lifetime. When the time a cookie has existed reaches its associated lifetime, the cookie is deleted from the browser's host machine
- a particular cookie is information that is exchanged exclusively between one specific browser and one specific server
- A cookie is set in PHP with the setcookie function. This function takes one or more parameters.

setcookie(*name, value, expire, path*)

Mandatory, string

New value, string

Time in seconds, integer, default 0 (end of current session)

Cookies and Session

- **setcookie("valid", "true", time() + 86400);**

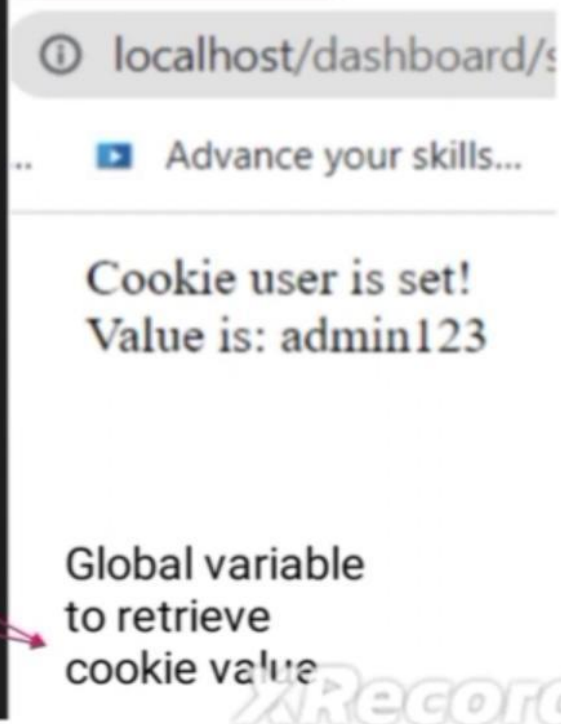
creates a cookie named "valid" whose value is "true" and whose lifetime is one day (86,400 is the number of seconds in a day)

KTU CS TUTOR

```php
<?php
$cookie_name ="user";
$cookie_value ="admin123";
setcookie($cookie_name, $cookie_value, time() + (86400*30),"/");
?>
<html>
    <body>
        <?php
        if(!isset($_COOKIE[$cookie_name]))
        {
            echo "Cookie named ". $cookie_name."is not set!";
        }
        else
        {
            echo "Cookie $cookie_name  is set!<br>";
            echo "Value is: ".$_COOKIE[$cookie_name];
        }
        ?>
    </body>
</html>
```

Available in entire website

To check if cookie is set



Global variable to retrieve cookie value

Cookies and Session

- **setcookie("valid", "true", time() -3600);**

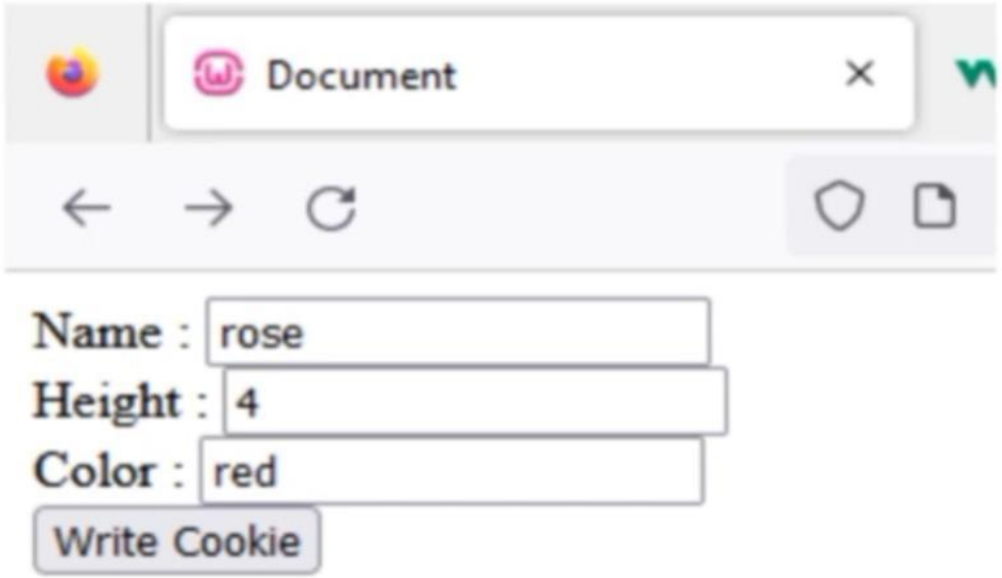
To delete a cookie, use setcookie() function with an expiration date in the past:

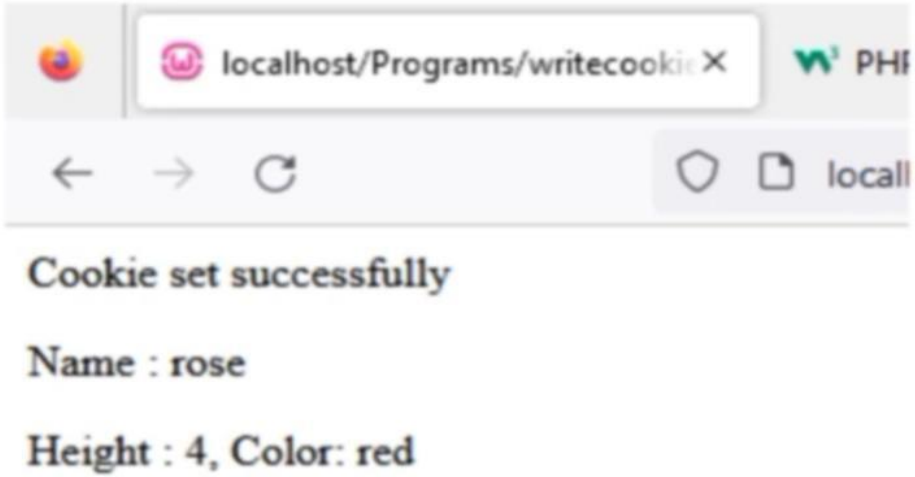
- cookie values are treated much like form values
- All cookies that arrive with a request are placed in the implicit $_COOKIES array, which has the cookie names as keys and the cookie values as values
- most browsers have a limit on the number of cookies that will be accepted from a particular server site

Cookies and Session

- Rather than using one or more cookies, a single session array can be used to store information about the previous requests of a client during a session
- session arrays often store a unique session ID for a session

Create cookie using form data





Create cookie using form data

```php
<?php
if(isset($_POST["submit"]))
{
  setcookie("name",$_POST["name"],time()+(86400*5));
  setcookie("height",$_POST["height"],time()+(86400*5));
  setcookie("color",$_POST["color"],time()+(86400*5));
  $name=$_COOKIE["name"];
  $height=$_COOKIE["height"];
  $color=$_COOKIE["color"];
  echo "Cookie set successfully";
  echo "<p>Name : $name</p> Height : $height, Color: $color";
}
else
{
?>
```

```php
<!DOCTYPE html>
<html>
<head>
<title>Document</title>
</head>
<body>
<form method="post" action="writecookie.php">
<label>Name : </label>
<input type="text" name="name"><br>
<label>Height : </label>
<input type="text" name="height"><br>
<label>Color : </label>
<input type="text" name="color"><br>
<input type="submit" name="submit" value="Write Cookie">
<br>
</form>
</body>
</html>
<?php
}
?>
```

Session

- on the internet the web server does not know who you are or what you do, because the HTTP address doesn't maintain state
- Session variables solve this problem by storing user information to be used across multiple pages (e.g. username ,pwd etc)
- By default, session variables last until the user closes the browser
- Session variables hold information about one single user, and are available to all pages in one application
- differ from cookies is that they can be stored on the server, whereas cookies are stored on the client

Session

- session ID is an internal value that identifies a session
- Session IDs need not be known or handled in any way by PHP scripts
- session_start function causes a session ID to be created and recorded
- On subsequent calls to session_start in the same session, the function retrieves the $_SESSION array, which stores any session variables and their values that were registered in previously executed scripts in this session
- A session is started with the session_start() function.
- Session variables are set with the PHP global variable: $_SESSION

Session

```php
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html><body>
    <?php

    $_SESSION["uname"] ="admin";
    $_SESSION["pwd"] ="admin123";
    echo "Session variables are set.";
    ?>
</body></html>
```

```php
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html><body>
    <?php

    // to change a session variable, just overwrite it
    $_SESSION["uname"] ="administrator";
    print ("welcome ".$_SESSION["uname"] );
    ?>
</body></html>
```

Session – destroy a session

- session_unset() and session_destroy() removes all global session variables and destroy the session

```php
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html><body>
    <?php
     // remove all session variables
    session_unset();
    // destroy the session
        session_destroy();
        echo "session cleared";
    ?>
</body></html>
```

MySQL Integration-Connecting to MySQL with PHP

- The most popular ways to connect a PHP script to MySQL are MySQli and PDO
- MySQL is an open-source relational database management system (RDBMS) used with PHP
- data in a MySQL database are stored in tables that consist of columns and rows
- MySQL is a database system that runs on a server ideal for both small and large applications which is very fast, reliable, and easy-to-use
- uses standard SQL and compiles on a number of platforms
- PHP 5 and later can work with a MySQL database using:
  - MySQLiextension (the 'i' is abbreviation for improved)
  - PDO (PHP Data Objects)

MySQL Integration-Connecting to MySQL with PHP

- **Connection to MySQL using MySQLi**
- PHP provides mysql_connect() function to open a database connection
- To disconnect from the MySQL database use mysql_close().
- There is also a procedural approach of MySQLito establish a connection to MySQL database from a PHP script

MySQL Integration-Connecting to MySQL with PHP

- The basic syntax for a connection to MySQL is as follows:

$mysqli= mysqli_connect("hostname", "username", "password", "database");

- Eg.

$mysqli= mysqli_connect("localhost", “root", “ ", "test");

MySQL Integration-Connecting to MySQL with PHP

```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname="test";
// Connection
$conn= mysqli_connect($servername,$username, $password,$dbname);
// For checking if connection is successful or not
if ($conn->connect_error) {
die("Connection failed: ". $conn->connect_error);
}
echo "Connected to ".$dbname." Establised successfully";
?>
```

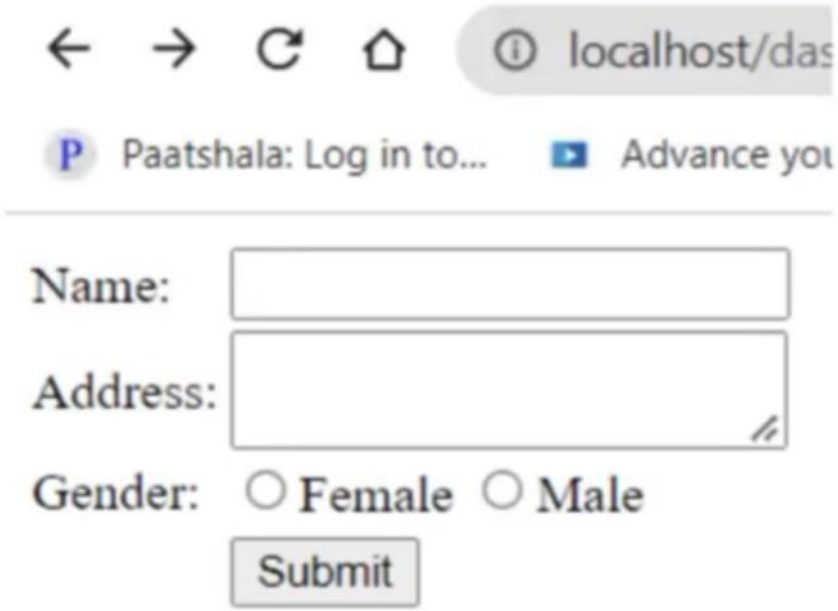
MySQL

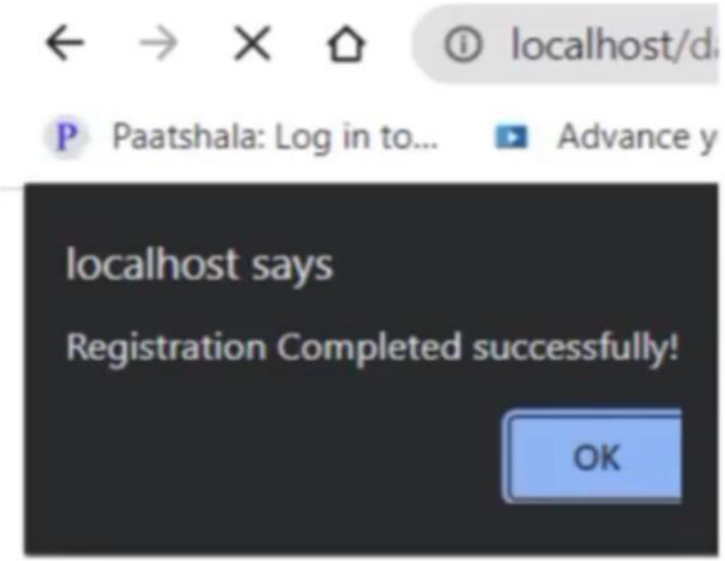
Just like exit()

```php
<?php
$conn = mysqli_connect("localhost", "root", "", "test");
// For checking if connection is successful or not
if ($conn->connect_error) {
die("Connection failed: ". $conn->connect_error);
}
echo "Connection to DB Establised successfully";
 $sql = "INSERT INTO testTable(testField) VALUES ('$_POST[testfield]')";
        $res = mysqli_query($conn, $sql);
        //if ($conn->query($sql) === TRUE)
        if($res===TRUE)
            {
             echo '<br/>Data Inserted successfully!';
             }
        else
            {
            echo "Error: " . $sql . "<br>" . $conn->error;
            }
         $conn->close();
```

KTU CS TUTOR

Registration form



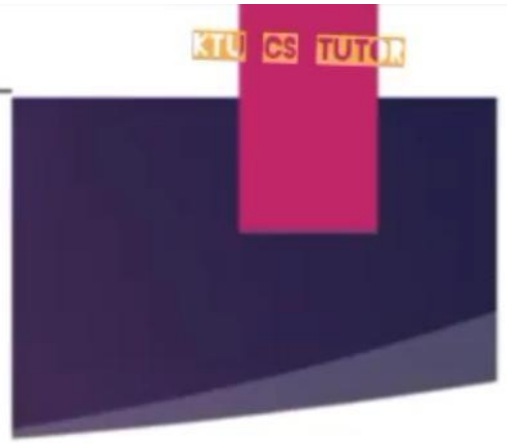


KTU CS TUTOR

```php
<?php
 if(isset($_POST["submit"]))
    { // Check if we click on SUBMIT BUTTON
    $servername = "localhost";
    $username = "root";
    $password = "";
    $dbname = "test";
    $conn = new mysqli($servername, $username, $password, $dbname);
    if ($conn->connect_error)
        {
           die("Connection failed: " . $conn->connect_error);
        }// insert Query
        $sql = "INSERT INTO ttest(name,address,gender) VALUES ('$_POST[name]', '$_POST[address]', '$_POST[gender]')";
        $res = mysqli_query($conn, $sql);
        //if ($conn->query($sql) === TRUE)
        // {
            if($res===TRUE)
            {
             echo '<script>alert("Data Inserted successfully!");</script>';
             }
        else
            {
            echo "Error: " . $sql . "<br>" . $conn->error;
            }
         $conn->close();
    }
    else
    {
    ?>
```

```
<form method="post" action="register.php">
<table>
    <tr>
        <td>Name: </td>
        <td><input type="text" name="name"></td>
    </tr>
    <tr><td>Address:</td>
        <td><textarea name="address"></textarea></td>
    </tr>
    <tr>
            <td>Gender:</td>
            <td><input type="radio" name="gender" value="female">Female
            <input type="radio" name="gender" value="male">Male</td>
    </tr>
    <tr><td></td><td><input type="submit" name="submit" value="Submit"></td>
    </tr>
    </table>
</form>
 <?php
}
?>
```

A Script to Create a Table




```
<?php
$mysqli = mysqli_connect("localhost", "testuser", "somepass", "testDB");

if (mysqli_connect_errno()) {
    printf("Connect failed: %s\n", mysqli_connect_error());
    exit();
} else {
     $sql = "CREATE TABLE testTable
            (id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
            testField VARCHAR(75))";
     $res = mysqli_query($mysqli, $sql);

     if ($res === TRUE) {
            echo "Table testTable successfully created.";
     } else {
         printf("Could not create table: %s\n", mysqli_error($mysqli));
     }

     mysqli_close($mysqli);
 }
 ?>
```

Select Data From a MySQL Database

Select Data With MySQLi can also be done in two ways

- MySQLi Object-oriented
- MySQLi Procedural

```php
<?php
$conn = mysqli_connect("localhost", "root", "", "test");
// For checking if connection is successful or not
if ($conn->connect_error) {
die("Connection failed: ". $conn->connect_error);
}
echo "Connection to DB Estabilised successfully";
$sql = "SELECT * FROM testTable";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
  // output data of each row
  echo "<br/><b>.....id....My Message.....</b><br/>" ;
  while($row = $result->fetch_assoc()) {
    echo  "....".$row["id"]. "......." . $row["testField"]. "<br>";
  }
} else {
  echo "0 results";
}
$conn->close();
?>
```

```php
<?php
$conn = mysqli_connect("localhost", "root", "", "test");
// For checking if connection is successful or not
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
  }
$sql = "SELECT * FROM testTable";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
  // output data of each row
  echo "<table border='border' width='300'><tr><th>id</td><td width='250'>My Message</td></tr>" ;
  while($row = mysqli_fetch_assoc($result)) {
    echo  "<tr><td>".$row["id"]. "</td><td>" . $row["testField"]. "</td></tr>";
  }
  echo "</table>";
} else {
  echo "0 results";
}
$conn->close();
?>
```

id	My Message
3	cccc
4	wwwww
5	god bless you
6	my name is smitha
8	I am from Pala