UNIT – 2 (part-1) Handling Form, Session Tracking & PHP Components

BCA SEM – 2

WEB PROGRAMMING

Code : CS-08

Presented By : Dhruvita Savaliya

Topics :

- Handling form with GET & POST
- Cookies
- Session
- Server variable
- PHP Components –
  - PHP GD Library
  - PHP Regular expression
  - Uploading file
  - Sending mail

Handling form with GET & POST :

- There are two ways the browser client can send information to the web server.

  - The **GET** Method
  - The **POST** Method

1) GET Method :

- GET is used to request data from a specified resource.
- Before the browser sends the information, it encodes it using a scheme called URL encoding.
- In this scheme, name/value pairs are joined with equal signs (=) and different pairs are separated by the ampersand (&).
- Spaces are removed and replaced with the + character and any other non-alphanumeric characters are replaced with a hexadecimal values.
- After the information is encoded it is sent to the server.

- **Some notes on GET requests:**
  - GET requests can be cached
  - GET requests remain in the browser history
  - GET requests can be bookmarked
  - GET requests should never be used when dealing with sensitive data
  - GET requests have length restrictions
  - GET requests are only used to request data (not modify)
  - The GET method is restricted to send upto 1024 characters only.
  - Never use GET method if you have password or other sensitive information to be sent to the server.
  - GET can't be used to send binary data, like images or word documents, to the server.
  - The data sent by GET method can be accessed using QUERY_STRING environment variable.
  - The PHP provides **$_GET** associative array to access all the sent information using GET method.

```
<?php
  if( $_GET["name"] || $_GET["age"] ) {
     echo "Welcome ". $_GET['name']. "<br />";
     echo "You are ". $_GET['age']. " years old.";
    exit();
  }
?>
<html>
  <body>
     <form action = "<?php $_PHP_SELF ?>" method = "GET">
        Name: <input type = "text" name = "name" />
        Age: <input type = "text" name = "age" />
        <input type = "submit" />
     </form>
  </body>
</html>
```

2) POST Method :

- The POST method transfers information via HTTP headers. The information is encoded as described in case of GET method and put into a header called QUERY_STRING.
- The POST method does not have any restriction on data size to be sent.
- The POST method can be used to send ASCII as well as binary data.
- The data sent by POST method goes through HTTP header so security depends on HTTP protocol. By using Secure HTTP you can make sure that your information is secure.
- The PHP provides **$_POST** associative array to access all the sent information using POST method.

- **Some notes on POST requests:**
- POST requests are never cached
- POST requests do not remain in the browser history
- POST requests cannot be bookmarked
- POST requests have no restrictions on data length

```
<?php
  if( $_POST["name"] || $_ POST["age"] ) {
     echo "Welcome ". $_ POST['name']. "<br />";
     echo "You are ". $_ POST['age']. " years old.";
    exit();
  }
?>
<html>
  <body>
     <form action = "<?php $_PHP_SELF ?>" method = " POST ">
        Name: <input type = "text" name = "name" />
        Age: <input type = "text" name = "age" />
        <input type = "submit" />
     </form>
  </body>
</html>
```

Points	GET	POST
BACK button/Reload	Harmless	Data will be re-submitted (the browser should alert the user that the data are about to be re-submitted)
Bookmarked	Can be bookmarked	Cannot be bookmarked
Cached	Can be cached	Not cached
Encoding type	application/x-www-form-urlencoded	application/x-www-form-urlencoded or multipart/form-data. Use multipart encoding for binary data
History	Parameters remain in browser history	Parameters are not saved in browser history
Restrictions on data length	Yes, when sending data, the GET method adds the data to the URL; and the length of a URL is limited (maximum URL length is 1024 characters)	No restrictions

Points	GET	POST
Restrictions on data type	Only ASCII characters allowed	No restrictions. Binary data is also allowed
Security	GET is less secure compared to POST because data sent is part of the URL Never use GET when sending passwords or other sensitive information!	POST is a little safer than GET because the parameters are not stored in browser history or in web server logs
Visibility	Data is visible to everyone in the URL	Data is not displayed in the URL

Cookies :

- Cookies are text files stored on the client computer and they are kept of use tracking purpose. PHP transparently supports HTTP cookies.
- There are three steps involved in identifying returning users –
  - Server script sends a set of cookies to the browser. For example name, age, or identification number etc.
  - Browser stores this information on local machine for future use.
  - 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.
- A **cookie** in PHP is a small file with a maximum size of 4KB that the web server stores on the client computer.
- They are typically used to keep track of information such as a username that the site can retrieve to personalize the page when the user visits the website next time.
- 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.
- **Setting Cookie In PHP**: To set a cookie in PHP, the **setcookie()** function is used. The setcookie() function needs to be called prior to any output generated by the script otherwise the cookie will not be set.

- **Syntax :**

  setcookie(name, value, expire, path, domain, security);

- **Parameters:** The setcookie() function requires six arguments in general which are:
  - **Name:** It is used to set the name of the cookie.
  - **Value:** It is used to set the value of the cookie.
  - **Expire:** It is used to set the expiry timestamp of the cookie after which the cookie can't be accessed.
  - **Path:** It is used to specify the path on the server for which the cookie will be available.
  - **Domain:** It is used to specify the domain for which the cookie is available.
  - **Security:** It is used to indicate that the cookie should be sent only if a secure HTTPS connection exists.

- **Example :**

```
<?php
   setcookie("cookie_name", "Luxury Car", time() + 2 * 24 * 60 * 60);
   if (isset($_COOKIE["cookie_name"]))     {
         echo "Auction Item is a " . $_COOKIE["cookie_name"];
   }
   else   {
         echo "No items for auction.";
   }
?>
```

NOTE : The cookie will expire after 2 days(2 days * 24 hours * 60 mins * 60 seconds). It will throw Warning:

Warning: Undefined array key " cookie_name "
in **C:\xampp\htdocs\hello.php** on line **3**

- **Deleting Cookies**:
- The setcookie() function can be used to delete a cookie. For deleting a cookie, the setcookie() function is called by passing the cookie name and other arguments or empty strings but however this time, the expiration date is required to be set in the past.
- To delete a cookie named “cookie_name”, the following code can be executed.

- **Example :**

```
<?php
   setcookie(" cookie_name ", "Luxury Car", time() + 2 * 24 * 60 *
   60);
   setcookie("cookie_name", "", time() - 60);
   echo "cookie is deleted" ;
?>
```

- **Important Points:**
- If the expiration time of the cookie is set to 0 or omitted, the cookie will expire at the end of the session i.e. when the browser closes.
- The same path, domain, and other arguments should be passed that were used to create the cookie in order to ensure that the correct cookie is deleted.

Session :

- Session refers to a frame of communication between two medium. A PHP session is used to store data on a server rather than the computer of the user.
- Session identifiers or SID is a unique number which is used to identify every user in a session based environment.
- A session creates a file in a temporary directory on the server where registered session variables and their values are stored.
- This data will be available to all pages on the site during that visit.

- **Starting a PHP Session**: The first step is to start up a session. After a session is started, session variables can be created to store information. The PHP **session_start()** function is used to begin a new session.

  It also creates a new session ID for the user.

```
<?php
   session_start();
?>
```

- **Storing Session Data**: Session data in key-value pairs using the **$_SESSION[ ]** superglobal array. The stored data can be accessed during lifetime of a session.

➢ Below is the PHP code to store a session with two session variables Rollnumber and Name:

```
<?php
session_start();
$_SESSION["Rollnumber"] = "11";
$_SESSION["Name"] = "Ajay";
?>
```

- **Accessing Session Data**: Data stored in sessions can be easily accessed by firstly calling **session_start()** and then by passing the corresponding key to the **$_SESSION** associative array.

➢ The PHP code to access a session data with two session variables Rollnumber and Name is shown below:

```
<?php
session_start();
echo 'The Name of the student is :' . $_SESSION["Name"] . '<br>';
echo 'The Roll number of the student is :' . $_SESSION["Rollnumber"] .
   '<br>';
?>
```

- **To destroy the session there are 3 steps to be used :**
- **1. Start the session**
- **2. Clear the data**
- **3. Destroy the session**
- **Destroying Certain Session Data (Clear session data)** :

  To Clear only session data and it unset temporary session data.

  The unset feature can be used with the corresponding session variable in the **$_SESSION** associative array.

  The PHP code to unset only the "Rollnumber" session variable from the associative session array:

```
<?php
session_start();
if(isset($_SESSION["Name"])){
    unset($_SESSION["Rollnumber"]);
}
?>
```

- **Destroying Complete Session**:
  The **session_destroy()** function is used to completely destroy a session. The session_destroy() function does not require any argument.

```
<?php
session_start();
session_destroy();
?>
```

- **Important Points**
- The session IDs are randomly generated by the PHP engine .
- The session data is stored on the server therefore it doesn't have to be sent with every browser request.
- The session_start() function needs to be called at the beginning of the page, before any output is generated by the script in the browser.

Cookie	Session
Cookies are client-side files on a local computer that hold user information.	Sessions are server-side files that contain user data.
Cookies end on the lifetime set by the user.	When the user quits the browser or logs out of the programmed, the session is over.
It can only store a certain amount of info.	It can hold an indefinite quantity of data.
The browser's cookies have a maximum capacity of 4 KB.	We can keep as much data as we like within a session, however there is a maximum memory restriction of 128 MB that a script may consume at one time.
Because cookies are kept on the local computer, we don't need to run a function to start them.	To begin the session, we must use the session start() method.
Cookies are not secured.	Session are more secured compare than cookies.
Cookies stored data in text file.	Session save data in encrypted form.
Cookies stored on a limited data.	Session stored a unlimited data.
In PHP, to get the data from Cookies , $_COOKIES the global variable is used	In PHP , to set the data from Session, $_SESSION the global variable is used
We can set an expiration date to delete the cookie's data. It will automatically delete the data at that specific time.	In PHP, to destroy or remove the data stored within a session, we can use the session_destroy() function, and to unset a specific variable, we can use the unset() function.

Server Variables / Super Global Variables :

- Some predefined variables in PHP are "superglobals", which means that they are always accessible, regardless of scope - and you can access them from any function, class or file without having to do anything special.

Sr.No	Variable & Description
1	**$GLOBALS :** Contains a reference to every variable which is currently available within the global scope of the script. The keys of this array are the names of the global variables.
2	**$_SERVER :**This is an array containing information such as headers, paths, and script locations. The entries in this array are created by the web server. There is no guarantee that every web server will provide any of these. See next section for a complete list of all the SERVER variables.
3	**$_GET :** An associative array of variables passed to the current script via the HTTP GET method.
4	**$_POST :**An associative array of variables passed to the current script via the HTTP POST method.
5	**$_FILES :**An associative array of items uploaded to the current script via the HTTP POST method.
6	**$_REQUEST :**An associative array consisting of the contents of $_GET, $_POST, and $_COOKIE.
7	**$_COOKIE :** An associative array of variables passed to the current script via HTTP cookies.
8	**$_SESSION :** An associative array containing session variables available to the current script.
9	**$_ENV** : It is another superglobal associative array in PHP. It stores environment variables available to current script.
10	**$_PHP_SELF :** A string containing php script file name in which it is called

PHP Components :

- PHP GD Library
- PHP Regular expression
- Uploading file
- Sending mail

PHP GD Library :

- GD is an open-source code library that is required to create and manipulate images in PHP.
- It is used for creating PNG, JPEG, and GIF images. It is commonly used to create charts, graphics, thumbnails, etc, and website development is the most common application of GD.
- The GD library is a graphics drawing library that provides tools for manipulating image data. In Shop, the GD library is used to process images for generating gallery preview and thumbnail size images automatically.
- PHP can do much more than just serve HTML to visitors. For instance, it has the ability to manipulate images. Not only that, but you can also create your own images from scratch and then either save them or serve them to users.
- PHP can handle almost all your basic image manipulating needs using the GD library—short for Graphic Draw.

- **Note:** On Windows servers, php_gd2.dll is included in a standard PHP installation, but is not enabled by default. To enable it, uncomment the extension=php_gd2.dll line in your php.ini file (remove the # from the beginning of that line) and restart the PHP extension.
- **Note:** If you are working on Windows, you can include the php_gd2.dll file as an extension in php.ini. If you're using something like XAMPP, you will find the php_gd2.dll file in the directory xampp\php\ext. You can also check if GD is installed on your system using the function phpinfo();. If you scroll through the resulting output, you will find something similar to the following.
- **There are 5 steps for graphics image :**
- 1. Create Canvas
- 2. Allocate background color for Canvas.
- 3. Allocate foreground color for Shapes.
- 4. Draw shapes on the Canvas.
- 5. Place the canvas on the web-page.

* Canvas Function:

1. Imagecreate()

It is the first step to draw the image. This function is used for creating the canvas. To create the canvas fixed width and height is to be given.

Syntax

imagecreate(integer width, integer height)

Example:

```
<?php
$canvas=imagecreate(500,500); imagepng($canvas);
?>
```

* Color Allocation Function:

2. imagecolorAllocate()

it is the next step for drawing the image on the canvas. This function is used for allocating the color to background of canvas and the shapes which are drawn. The colors which are to be used are in RGB format.

Syntax

imagecolorallocate(canvas_handler,int red, int green, int blue)

Example

```
<?php
$canvas=imagecreate(500,500);
$bk=imagecolorallocate($canvas,0,0,0); imagepng($canvas);
?>
```

* Shapes Function

3. imageline()

This function is used to draw line on the canvas. It has 6 parameters to be taken as input.

Syntax

imageline(canvas variable,int x1, int y1, int x2, int y2,color)

Example:

```
<?php
$canvas=imagecreate(500,500);
$bk=imagecolorallocate($canvas,0,0,0);
$fg=imagecolorallocate($canvas,1,1,1); imageline($canvas,200,200,250,200,$fg);
  imagepng($canvas);
?>
```

* Draw a rectangle

4. imagerectangle() or imagefilledrectangle()

imagerectangle() creates a rectangle of color col in image image starting at upper left coordinate x1, y1 and ending at bottom right coordinate x2, y2. 0, 0 is the top left corner of the image.

Syntax:

int imagerectangle(resource image, int x1, int y1, int x2, int y2, int col)

int imagefilledrectangle(resource image, int x1, int y1, int x2, int y2, int col)

Example:

```
<?php
$canvas=imagecreate(500,500);
$bk=imagecolorallocate($canvas,0,0,0);
$fg=imagecolorallocate($canvas,1,1,1);
   imagefilledrectangle($canvas,200,200,250,200,$fg); imagepng($canvas);
?>
```

*** Draw a filled ellipse**

5. imageellipse() or imagefilledellipse()

Syntax: bool imagefilledellipse (resource image, int cx, int cy, int w, int h, int color)

imagefilledellipse() draws an ellipse centered at cx, cy (top left is 0, 0) in the image represented by image. W and h specifies the ellipse's width and height respectively. The ellipse is filled using color.

Example:

```
<?php
$canvas=imagecreate(500,500);
$bk=imagecolorallocate($canvas,0,0,0);
$fg=imagecolorallocate($canvas,1,1,1);
   imagefilledellipse($canvas,200,200,10,50,$fg); imagepng($canvas);
?>
```

*** Draw a partial ellipse**

6. imagearc() Syntax:

int imagearc(resource image, int cx, int cy, int w, int h, int s, int e, int color)

imagearc() draws a partial ellipse centered at cx, cy (top left is 0, 0) in the image represented by image.

and h specifies the ellipse's width and height respectively while the start and end points are specified in degrees indicated by the s and e arguments. 0° is located at the three-o'clock position, and the arc is drawn counter- clockwise.

Example:

```
<?php
$canvas=imagecreate(500,500);
$bk=imagecolorallocate($canvas,0,0,0);
$fg= imagecolorallocate($canvas,1,1,1);
imagearc($canvas,200,200,15,5,0,180,$color); imagepng($canvas);
?>
```

7. imagepng() or imagejpeg() or imagegif()

The last step is to place this canvas on the web-page.This image can be in any format that is either PNG, JPEG or GIF.

If PNG image is to be created, then the function imagepng() is used.

If JPEG image is to be created, then the function imagejpeg() is used.

If GIF image is to be created, then the function imagegif() is used.

Syntax:

imagepng(canvas variable)

imagejpeg(canvas variable)

imagegif(canvas variable)

Example :

```
<?php
   header("content-type:image/png");
   $image=imagecreate(100,70);
            //It create box which has 100 px width and 70 px height
   $bg=imagecolorallocate($image,50,2,1);
          //It gives background color in box
   $t_color=imagecolorallocate($image,233,14,91);
           //It gived text color
   imagestring($image,1,5,5,"hello",$t_color);
         //imagestring("image","font","x position","y position","text","text color")
   imagepng($image);
           //use to display image in browser
?>
```

Regular Expression :

- Regular Expression means the formula or the general format of any string or expression.
- It is used for pattern matching.
- It is also known as regexis.
- It checks any characters or digits which are specified in the format.
- It is used generally for the validation on Server-side scripting.
- **Advantage :**
- It prevents the data to be entered in invalid format.
- It is a portable language as it has general format.
- It is easy to learn and use.
- Regular expression helps the programmers to validate text string.
- It is helpful in user input validation testing like email address, mobile number, and IP address.

- **Types of Regular Expression :**

1. POSIX Extended Regular Expression
2. PERL Compatible Regular Expression

1. **POSIX Extended Regular Expression :**

- When this type of regular expression is used then the function ereg() is required to be used for the pattern matching.
- The function ereg() is abbreviated as **e**xtended **reg**ular expression.
- It is the oldest type of regular expression in use.

Symbol	Description
^	For starting the format string
$	For ending the format string
[a-z]	Only lowercase alphabets
[A-Z]	Only Uppercase alphabets
[a-z A-Z]	Only alphabets
[0-9]	Only digits
[a-z A-Z 0-9]	Alphanumeric characters
\	Used when any special characters are to be used in the string
( )	Used when more than one value is to be checked
|	For or Condition
{n}	Used when the lengths of the digits are known. Where n is an integer value
{n,m}	n for minimum value and m for maximum value

- General symbols the other symbols used in the format are :

  * One or more times

  + Zero or more times

  ? Zero or one time

- **Note : ereg() is removed from php7. Replace with preg_match().**

2. PERL Compatible Regular Expression :

- This type of regular expression is developed using the Perl Language.
- This function is used for the given regular expression is preg_match() that is **P**erl **reg**ular expression.
- The symbols are used as the above ones only.
- It also return boolean value.
- If the format is matched , then it returns true(1) value otherwise it will return false(0) value.

Symbol	Description
\d	Any number
\D	Anything other than a number
\s	Kind of whitespace
\S	Anything other than a whitespace
\W	Any word character (including the underscore character)
\w	Anything other than a word character
\A	Beginning of string
\b	word boundary
\B	Not word boundary
\Z	End of string (Match before final new line or at end of string)
\z	End of string (matches only at the very end of the string)

- **Syntax :**

Boolean preg_match(‘format’, Variable \ Value to be checked)

- **Example :**

```
<?php
   $dt="16/01/2023";
   if(preg_match('/^\d{2}\/\d{2}\/\d{4}$/',$dt))
   {
         echo $dt;
   }
   else
   {
         echo "invalid";
   }
?>
```

Uploading File :

- A PHP script can be used with a HTML form to allow users to upload files to the server. Initially files are uploaded into a temporary directory and then relocated to a target destination by a PHP script.
- Information in the **phpinfo.php** page describes the temporary directory that is used for file uploads as **upload_tmp_dir** and the maximum permitted size of files that can be uploaded is stated as **upload_max_filesize**. These parameters are set into PHP configuration file **php.ini**
- **Approach:** In your "php.ini" file, search for the "file_uploads" parameter and set it to "On" as mentioned below.

  file_uploads = On
- Steps are:
  - The user opens the page containing a HTML form featuring a text files, a browse button and a submit button.
  - The user clicks the browse button and selects a file to upload from the local PC.
  - The full path to the selected file appears in the text filed then the user clicks the submit button.
  - The selected file is sent to the temporary directory on the server.
  - The PHP script that was specified as the form handler in the form's action attribute checks that the file has arrived and then copies the file into an intended directory.
  - The PHP script confirms the success to the user.

move_uploaded_file (string $filename , string $destination)

- **Html File :**

```
<form action="form.php" method="post" enctype="multipart/form-data">
    Select File:
    <input type="file" name="fileToUpload"/>
    <input type="submit" value="Upload Image" name="submit"/>
</form>
```

- **Php File:**

```
<?php
$target_path = "e:/";
$target_path = $target_path.basename( $_FILES['fileToUpload']['name']);
if(move_uploaded_file($_FILES['fileToUpload']['tmp_name'], $target_path)) {
    echo "File uploaded successfully!";
} else{
    echo "Sorry, file not uploaded, please try again!";
}
?>
```

Sending mail :

- PHP mail() function is used to send email in PHP.
- You can send text message, html message and attachment with message using PHP mail() function.
- **Syntax**

  bool mail (string $to , string $subject , string $message
  [, string $additional_headers
  [, string $additional_parameters]])
- It's a cost effective way of notifying users on important events.
- Let users contact you via email by providing a contact us form on the website that emails the provided content.
- Developers can use it to receive system errors by email
- You can use it to email your newsletter subscribers.
- You can use it to send password reset links to users who forget their passwords
- You can use it to email activation/confirmation links. This is useful when registering users and verifying their email addresses

Sr.No	Parameter & Description
1	**to** Required. Specifies the receiver / receivers of the email
2	**subject** Required. Specifies the subject of the email. This parameter cannot contain any newline characters
3	**message** Required. Defines the message to be sent. Each line should be separated with a LF (\n). Lines should not exceed 70 characters
4	**headers** Optional. Specifies additional headers, like From, Cc, and Bcc. The additional headers should be separated with a CRLF (\r\n)
5	**parameters** Optional. Specifies an additional parameter to the send mail program

```php
<?php
        ini_set("sendmail_from", "sonoojaiswal@javatpoint.com");
        $to = "sonoojaiswal1987@gmail.com";//change receiver address
        $subject = "This is subject";
        $message = "This is simple text message.";
        $header = "From:sonoojaiswal@javatpoint.com \r\n";

        $result = mail ($to,$subject,$message,$header);

        if( $result == true ){
           echo "Message sent successfully...";
        }else{
           echo "Sorry, unable to send mail...";
        }
?>
```

Assignment Questions :

1. Explain Session in detail
2. Explain Cookie in detail
3. What is GD Library explain in detail with all functions
4. What is FileUploading Explain in detail
5. What is sending main explain in detail
6. What is Regular expression Explain in detail with preg_match().
7. What is get and post method in detail
8. Explain Server variable.