Name: Akshay More

Roll No: 63

IP Assignment 2

Q.1) Explain how session management is done in PHP. Clearly explain how to create, access, modify session variables in PHP.

=> When you work with an application, you open it, do some changes, and then you close it. This is much like a Session. The computer knows who you are. It knows when you start the application and when you end. But on the internet there is one problem: 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, favorite color, etc). By default, session variables last until the user closes the browser.

So; Session variables hold information about one single user, and are available to all pages in one application.

A)CREATING OR STARTING A SEESION:

A session is started with the `session_start()` function.

Session variables are set with the PHP global variable: $_SESSION.

Now, let's create a new page called "demo_session1.php". In this page, we start a new PHP session and set some session variables:

Code:

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

```
<html>
<body>

<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>

</body>
</html>
```

B)Getting or Accessing PHP Session Variable:

Next, we create another page called "demo_session2.php". From this page, we will access the session information we set on the first page ("demo_session1.php").

Notice that session variables are not passed individually to each new page, instead they are retrieved from the session we open at the beginning of each page (`session_start()`).

Also notice that all session variable values are stored in the global $_SESSION variable:

Code:

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Echo session variables that were set on previous page
echo "Favorite color is " . $_SESSION["favcolor"] . ".<br>";
echo "Favorite animal is " . $_SESSION["favanimal"] . ".";
?>

</body>
</html>

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

```
<!DOCTYPE html>
<html>
<body>

<?php
print_r($_SESSION);
?>

</body>
</html>
```

C)Modyfing PHP Session Variable:

To change a session variable, just overwrite it:

Code:

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// to change a session variable, just overwrite it
$_SESSION["favcolor"] = "yellow";
print_r($_SESSION);
?>

</body>
</html>
```

Q.2) Explain the role of a cookie and differentiate it from sessions. Write a PHP script to check whether the cookie is set or not. Explain string functions in PHP.

Cookie	Session
• Cookies are client-side files that contain user information	• Sessions are server-side files which contain user information

• Cookie ends depending on the lifetime you set for it	• A session ends when a user closes his browser
• You don't need to start cookie as it is stored in your local machine	• In PHP, before using $_SESSION, you have to write session_start(); Likewise for other languages
• The official maximum cookie size is 4KB	• Within-session you can store as much data as you like. The only limits you can reach is the maximum memory a script can consume at one time, which is 128MB by default
• A cookie is not dependent on Session	• A session is dependent on Cookie
• There is no function named unsetcookie()	• Session_destroy(); is used to destroy all registered data or to unset some

C) PHP Script to check wether the Cookie is set or not:

```
<?php
setcookie("test_cookie", "test", time() + 3600, '/');
?>
<html>
<body>

<?php
if(count($_COOKIE) > 0) {
  echo "Cookies are enabled.";
} else {
  echo "Cookies are disabled.";
}
?>
```

```
</body>
</html>
```

D)String Funtions in PHP:

Following are some of the most commonly used string functions in PHP:

A) strlen() - Return the Length of a String

B) str_word_count() - Count Words in a String

C) strrev() - Reverse a String

D) strpos() - Search For a Text Within a String

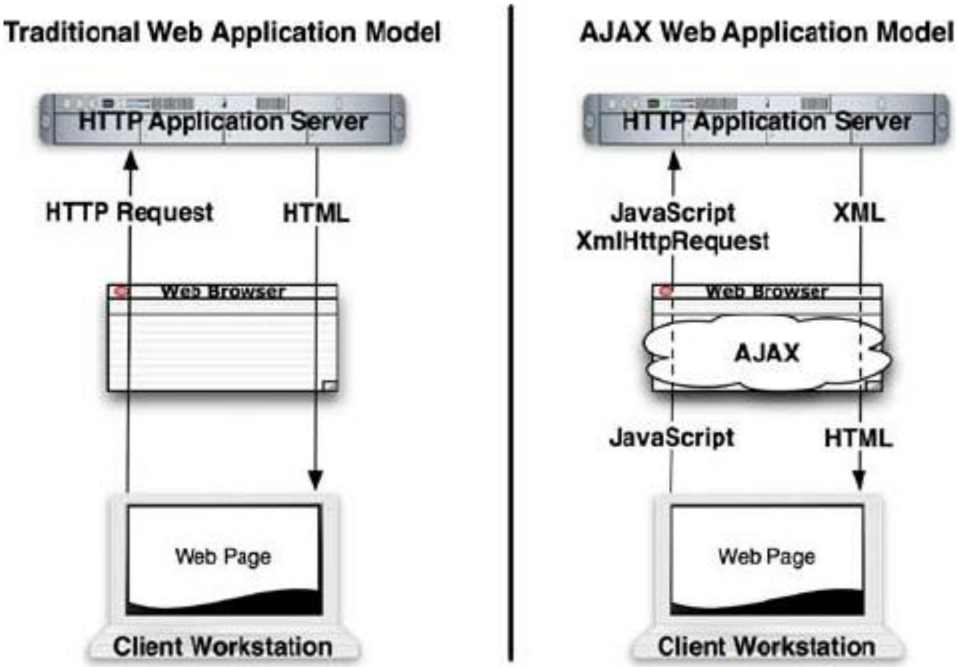
E) str_replace() – Replaces some characters with some other characters in a string

Q.3) What are the characteristics of Rich Internet Application?

Following are some of its characteristics:

- ***Performance*** - RIAs can often perform better than traditional applications on the basis of the characteristics of network and applications, performance of server also improved by offloading possible processing work to the client system and also perceived performance in terms of UI responsiveness and smoother visual transitions and animations are key aspects of any RIA.
- ***Offline use*** - When connectivity is unavailable, it might still be possible to use an RIA. An RIA platform let the user work with the application without connecting to the Internet and synchronizing it automatically when the user goes live.
- ***Security*** - RIAs should be as secure as any other web application, and the framework should be well equipped to enforce limitations appropriately when the user lacks the required privileges, especially when running within a constrained environment such as a sandbox.
- ***Rapid Development*** - An RIA Framework should facilitate rapid development of a rich user experience through its easy-to-use interfaces in ways that help developers.
- ***Direct Interaction*** - An RIA can use a wider range of controls that allow greater efficiency and enhance the user experience. In RIAs, for example, users can interact directly with page elements through editing or drag-and-drop tools. They can also do things like pan across a map or other image.
- ***Better Feedback*** - Because of their ability to change parts of pages without reloading, RIAs can provide the user with fast and accurate feedback, real-time confirmation of actions and choices, and informative and detailed error messages.

Q.4) Draw the diagram for AJAX Web application model and Traditional Web application model and compare them.

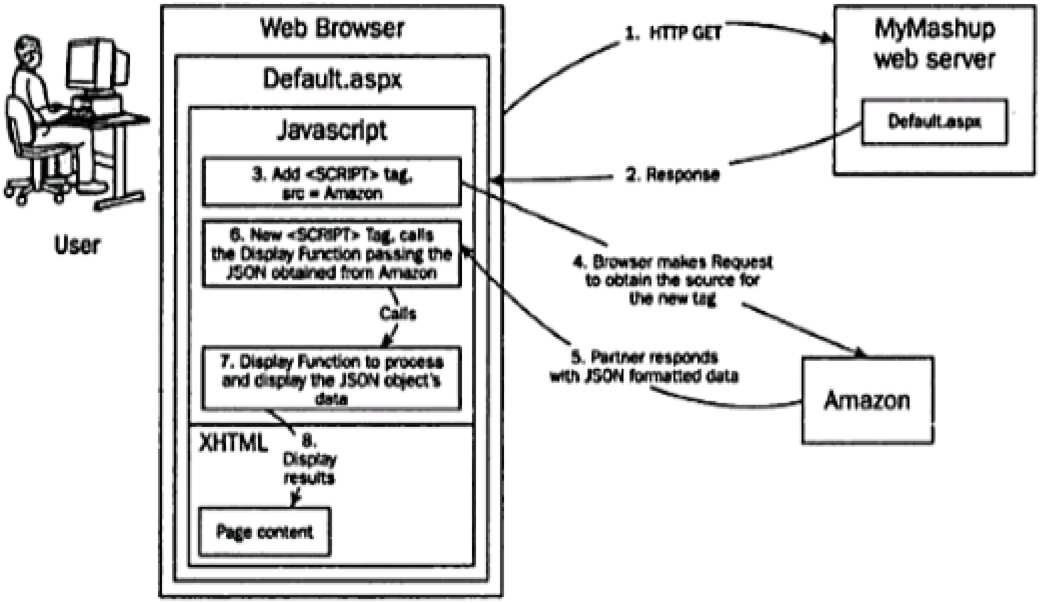


Difference between both:

AJAX	Traditional
• Application on client and server	• Application only on server
• Javascript receives user input,issues function calls to server when needed -Get map tile -Save location data	• Entire form sent to server -User fills in input items -Clicks on submit

- Server returns individual data items	- Server returns new page -Presentation+Data
- Javascript incorporates data items into existing page	

Q.5) Explain in detail JSON mash ups with neat diagram.


User
Web Browser
Default.aspx
Javascript
3. Add <SCRIPT> tag, src = Amazon
6. New <SCRIPT> Tag, calls the Display Function passing the JSON obtained from Amazon
Calls
7. Display Function to process and display the JSON object's data
XHTML
8. Display results
Page content
1. HTTP GET
MyMashup web server
Default.aspx
2. Response
4. Browser makes Request to obtain the source for the new tag
5. Partner responds with JSON formatted data
Amazon

The readability and simplicity features of JSON have made it one of the most popular notations in the mashup community. JSON, being a notation, provides a way in which objects are written so that human beings can read it easily. It has built-in JavaScript feature that made JSON a viable mashup technology. The architecture of mashups in JSON implementation is shown in above figure. The flow of JSON mashups that use dynamic Script method goes in the following steps:

1. The flow of the process starts with the browser sending request to the server by using HTTP GET.
2. The Web Server responds with a page that includes the following couple of important JavaScript functions:

   a. A parsing function that expects JavaScript objects to be parameters.

   b. The Dynamic Script method is the core of the initiation script through which a new script tag is added to the page, specifying the source for that script tag to be the Uniform Resource Locator (URL) at some partner site.

3. The source code for the new script tag gets loaded by the browser.
4. Amazon receives an HTTP Get request sent from the browser using the loaded script.
5. A JavaScript object, after being serialized into a JSON Object, is served by the partner site.
6. A function call to the render function wraps the JSON script, and the JavaScript entirely becomes the content for the script tag.
7. The new piece of JavaScript is tried for execution by the browser, which calls the render method from step 2(a).
8. The server invokes the render method and evaluates the JSON script, which is converted into a JavaScript object. The data contained in the render method is pushed into the page after the render method, which uses the new JavaScript object in its execution