

ADP EXERCISES (07/06/2023)

- Create a new folder / directory (say **mywebapp**) in the xampp/htdocs/ folder.
- Keep all files in the **mywebapp** folder.
- Create a simple HTML Form and call it **simpleform.html** and save it in the **mywebapp** folder. Create a php file **getformdetails1.php** and **getformdetails2.php** save it in the mywebapp folder. Let the simpleform.html send the data through the **action** attribute to **getformdetails1.php**. **getformdetails1.php** has php echoing the form data in a HTML table and **getformdetails2.php** has php embedded in html tags.

Ex: getformdetails1.php

```
$uname=$_POST["uname"]; //assuming the method is POST
echo "<table><td><tr>uname</tr><tr>".$uname."</tr></td></table>";
```

Ex: getformdetails2.php

```
<?php $uname=$_POST["uname"]; ?> //assuming the method is POST
<html><body>
<table><td><tr>uname</tr><tr><?php echo $uname;
?></tr></td></table></body></html>
```

- Start the xampp server.
- Type **localhost** on the browser to check if the server is up and running.

/sample code snippet: and both the .html and .php should reside on the server.

//simpleform.html

```
<html>
<body>
<form action="getdetails.php" method="get">
Name: <input type="text" name="uname"><br>
Branch: <input type="text" name="branch"><br>
<input type="submit">
</form></body></html>
```

//getdetails.php

```
<?php ?>
<html>
<body>
Welcome <?php echo $_GET["uname"]; ?><br>
Your branch is: <?php echo $_GET["branch"]; ?>
</body></html>
```

//refer to php cheatsheets for more

1. Implement the client-server communication using the above two server side files (getformdetails1.php, getformdetails2.php).
2. Create a session variable using any of the values from FORM data of your choice. Save any php file (say **first.php**) with the statements as shown in red.

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

//php sessions, php include file: a session is duration of time in which a series of requests and responses are made from the same client. Maintaining the user state / remember the user between a series of requests and responses. The fundamental problem with http is statelessness (no state information is retained by the server between request-response pairs), which is overcome with session variables. Any variable can be used as a session variable. It is important to start any php page with the **session_start();** statement

```
<?php
Session_start();
$username=$_POST["uname"];
$_SESSION["uname"]=$_POST["uname"]; // setting a session variable
?>
```

3. Now create another php page say **second.php** and call it from within **first.php** with a button click in the **first.php** (the goal is to set session variables in one page while access them in another page as global variables),

```
<input type="button" name="b1" value="redirect"
onClick='document.location.href="second.php"'/>
```

second.php

```
<?php session_start();
print_r($_SESSION); ?> // prints all the session variables as name value
pairs of an array. There can be multiple session
variables for a session. here we are using print
instead of echo.
```

delete session variables by calling the two built-in functions in that order:

```
session_unset();
session_destroy();
```

4. Using include; //including one php files inside another
Ex: create first.php and include second.php inside first.php as

first.php

```
<?php include 'second.php';
echo $name."<br>".$bat;
```

second.php

```
$name = 'alpha';  
$bat = "P";
```

/