

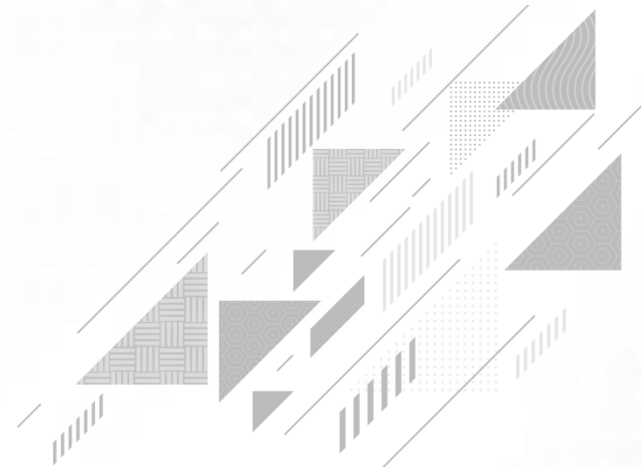
## Unit-5

# Session and State Management using PHP

**Prof. Mayur Prajapati**

Computer Engineering Department  
SVBIT, Gandhinagar

---





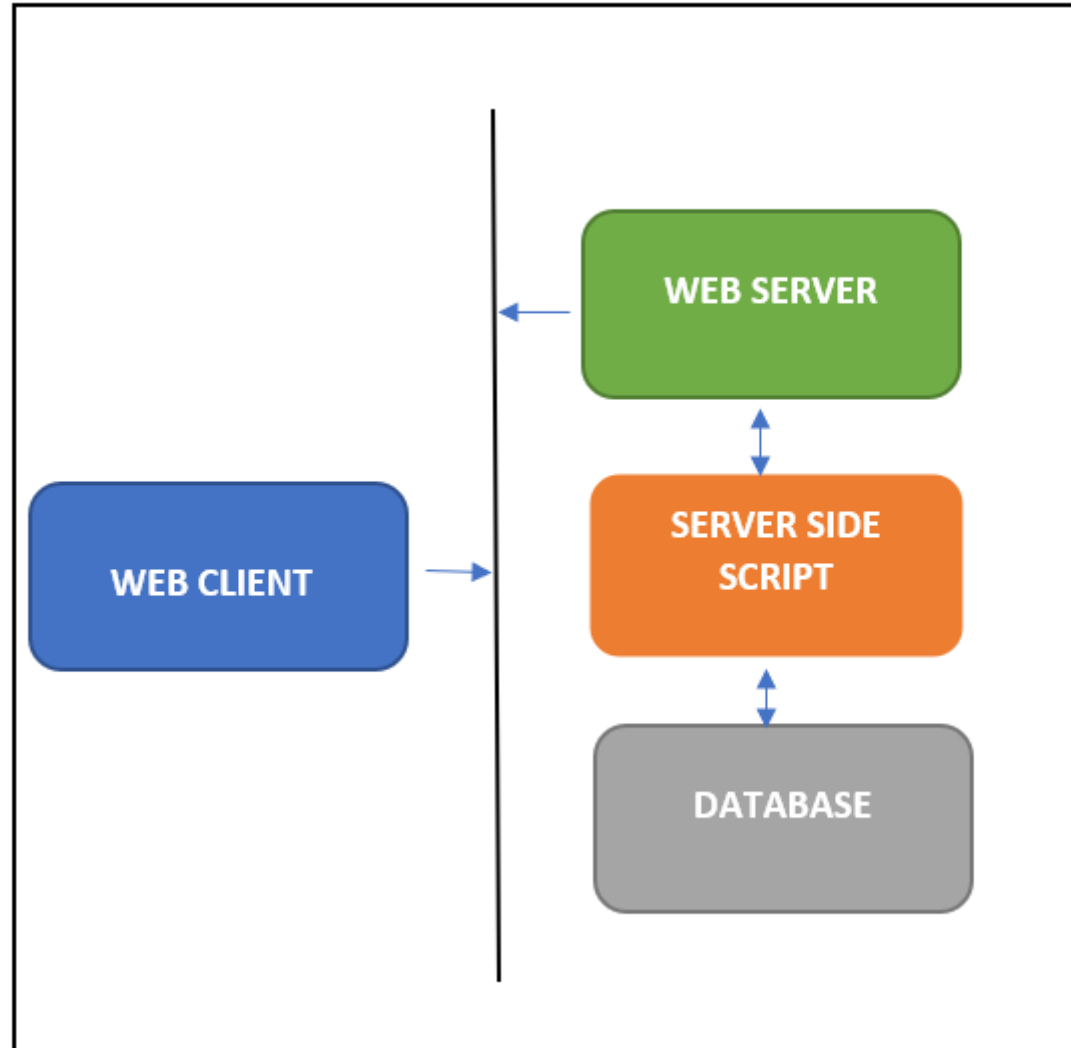
## Outline

- ✓ Need of session management
- ✓ Various techniques for state and session management like: Hidden Fields, Query String, Cookie and Session
- ✓ HTML vs HTML5

# State management System in PHP

- ▶ The basic foundation of a data communication system for the internet is HTTP (Hypertext Transfer Protocol) which is an application layer protocol, distributive and collaborative, and hypermedia information system.
- ▶ As we know that HTTP is a generic and stateless protocol to manage state in applications like E-commerce, Social Media, Blog sites, many commercial sites with the help of sessions and cookies.
- ▶ HTTP is a stateless protocol due to which it is also known to be connectionless.
- ▶ The server and client are aware of each other only when the current request after that client and server forgot about each other, so that browser cannot get information between different request across the web pages.

# State management System in PHP



# Type of state management system

1. **Server-side state management system:** In server-side state management system where we used to store user specific information to identify user on server and information is available in every web pages.

Example: session variables

2. **Client side State management System:** In a client-side state management system, the user information is stored by the browser.

Example: cookies

# Various techniques for state and session management

## ▶ Cookies:

- ▶ Cookies are used for client-side state management system.
- ▶ Cookies are data by the browser, cookies are sent to the web server as a piece of header information with every HTTP request.
- ▶ Cookies can contain 1KB (1024B) size of data.

## ▶ **Uses of Cookies:**

- ▶ To store information about visitors regarding the accessed website's page.
- ▶ Number of visit and views.
- ▶ Store first visit information and update it in every visit that pointed towards better experience of user.

# Various techniques for state and session management

## ▶ Type of Cookies:

▶ **1. Session Cookie:** This is a type of cookie that expires when the session will destroy.

▶ **2. Persistent Cookie:** Persistent cookie is a kind of cookie that is stored permanently on browser's system and expires on some specific time.

▶ **Creation of Cookies:** In PHP, we can create and set cookie by [setcookie\(\)](#)

## ▶ Syntax:

```
setcookie( name, value, expire, path, domain, secure );
```

# Various techniques for state and session management

- ▶ **name:** It is mandatory for the time of creation, other arguments are optional.
- ▶ **secure:** If it is set to 1, it means it is available and sent to PHP.
- ▶ Example:

```
<?php  
Setcookie("username", "abc", time() + 60 * 60);  
?>
```

- ▶ **Note:**
- ▶ To retrieve cookies data in PHP use `$_COOKIES`.
- ▶ To check if it is set or not, use `isset()` function.



# Various techniques for state and session management

## ► Example:

```
<?php
Setcookie("username", "abc", time() + 60 * 60);
?>
<html>
<body>
    if(isset($_COOKIE["username"])) {
        echo "Cookie is set";
    }
    else {
        echo "Cookie is not set";
    }
</body>
</html>
```

# Various techniques for state and session management

- ▶ For updating cookies only, we need to change the argument by calling **setcookie()** function. We change name “abc” to “xyz”.

```
<?php  
setcookie("username", "xyz", time() + 60 * 60);  
?>
```

- ▶ For deleting cookies we need to set expiry time in negative.

```
<?php  
Setcookie("username", "xyz", time() - 3600);  
?>
```

- ▶ **Note:** Drawback of using cookies is it can easily retrieve and also easily deleted. It is not secure.

# Various techniques for state and session management

## ▶ **Session:**

- ▶ Session stores server-side information, so that all the information are accessible to all the webpages.
- ▶ It is more secure than cookies.
- ▶ We know that HTTP is a stateless protocol so that previously performed task cannot be remembered by current request.
- ▶ **For example**, when we want to buy something online, we visit many e-commerce websites and compare products.
- ▶ Some of them are added to cart for future reference.
- ▶ After few days when we are ready to buy it, the information is still available as the cart session is set earlier.
- ▶ Session is size independent to store as many data the user wants.

# Various techniques for state and session management

- ▶ **Uses of Session:**
- ▶ It provides login and logout functionality to store and display relevant information.
- ▶ It maintains cart of e-commerce.

# Various techniques for state and session management

- **Creation of Session:** In PHP, session starts from [session\\_start\(\)](#) function and data can be set and get by using global variables `$_SESSION`.

```
<?php
    session_start();
    $_SESSION["username"] = "abc";
    $_SESSION["userid"] = "1";
?>
<html>
<body>
    <?php
        echo "Session variable is set";
    ?>
    <button type="submit" href="logout.html"></button>
</body>
</html>
```

# Various techniques for state and session management

- ▶ Retrieve information to another pages like below example:

```
<?php  
  
echo "Username:" . " " . $username;  
  
echo "Userid:" . " " . $userid;  
  
?>
```

- ▶ Output:

```
Username abc  
Userid 1
```

# Various techniques for state and session management

- ▶ For updating the value of session variable
- ▶ `$_SESSION["userid"]="1024";`
- ▶ To destroy session in PHP, first unset all the session variable using `session_unset()` and call `session_destroy()` methods.

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

session_unset();
session_destroy();
?>
</body>
</html>
```

# Various techniques for state and session management

