WEB SYSTEMS AND TECHNOLOGIES

Midterm MODULE 7

PHP COOKIES AND SESSIONS

Learning Outcomes

AT THE END OF THIS MODULE, STUDENTS SHOULD BE ABLE TO:

1. Identify and apply cookies and sessions in storing user’s data in a web page.

Lesson 1. Cookies

What are Cookies?

- Small files created by the server and stored on the client’s browser.
- Used to identify returning users.
- Automatically sent along with HTTP requests to the server.

Lesson 1. Cookies

Steps for Identifying Returning Users

1. Server sends cookies to the browser (e.g., name, ID).
2. Browser stores them locally.
3. On future visits, the browser sends stored cookies back to the server.

Lesson 1.1: Anatomy of a Cookie

- Cookies are set through HTTP headers or JavaScript.
- Typical header example:
- Set-Cookie: name=xyz; expires=Fri, 04-Feb-07 22:03:38 GMT; path=/; domain=tutorialspoint.com

Lesson 1.1: Anatomy of a Cookie

Cookie Components

- Name–Value Pair → The stored data.
- Expires → When the cookie should be deleted.
- Path & Domain → Define where the cookie is valid.
- Browser automatically sends cookies matching a page’s path/domain.

Lesson 1.2: Setting Cookies in PHP

- Function used: setcookie()
- Must appear before HTML tags.

Lesson 1.2: Setting Cookies in PHP

- setcookie() Parameters
- Name – Identifier for the cookie.
- Value – Content stored.
- Expire – Lifetime in seconds from Jan 1, 1970 (UNIX time).
- Path – Directory where cookie is valid.
- Domain – Domain scope (must have at least two dots).
- Security – 1 for HTTPS; 0 for HTTP.

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

Lesson 1.3: Creating and Retrieving Cookies

- Create a cookie: setcookie("user", "Ren", time() + (86400 * 30), "/");
- Retrieve cookie: $_COOKIE["user"]
- Use isset() to check if a cookie exists.
- Note: Cookie is available after refreshing or revisiting the page.

Lesson 1.4: Modifying and Deleting Cookies

- Modify: Recreate the cookie using the same name with a new value.
- Delete: Use setcookie() with an expired date.
- Example: setcookie("user", "", time() - 3600);

Lesson 2: Sessions

What are Sessions?

- Used to store user information temporarily across pages.
- Unlike cookies, session data is stored server-side.
- Each session has a unique ID (sent as a cookie or in the URL).
- Commonly used for features like shopping carts or login systems.

Lesson 2.1: Starting a Session

- Start session: session_start();
- Set variables: $_SESSION["name"] = "value";
- Must appear before any HTML output.

Lesson 2.2: Retrieving a Session

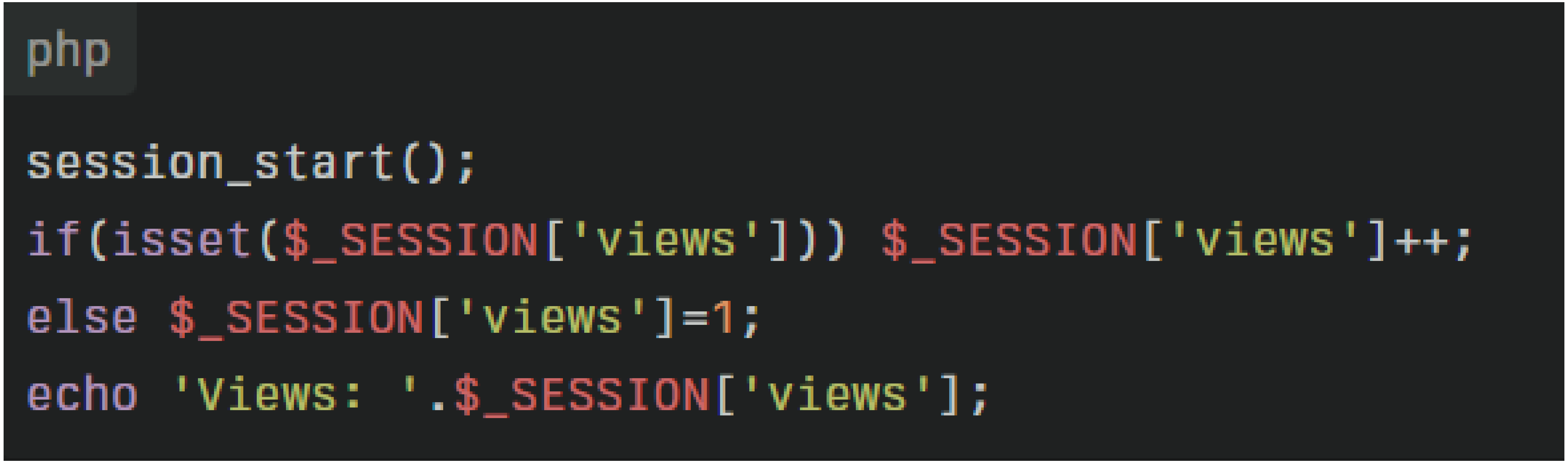
- Access existing session data: $_SESSION["variable"]
- Use session_start() on each page where sessions are needed.

How It Works

- PHP creates a unique session ID (e.g., 765487cf34ert8dede5a562e4f3a7e12).
- Stored in browser as a temporary cookie.
- Server retrieves the session based on this ID.

Lesson 2.3: Session Counters

- Track page views or visits using a session variable.
- Example:



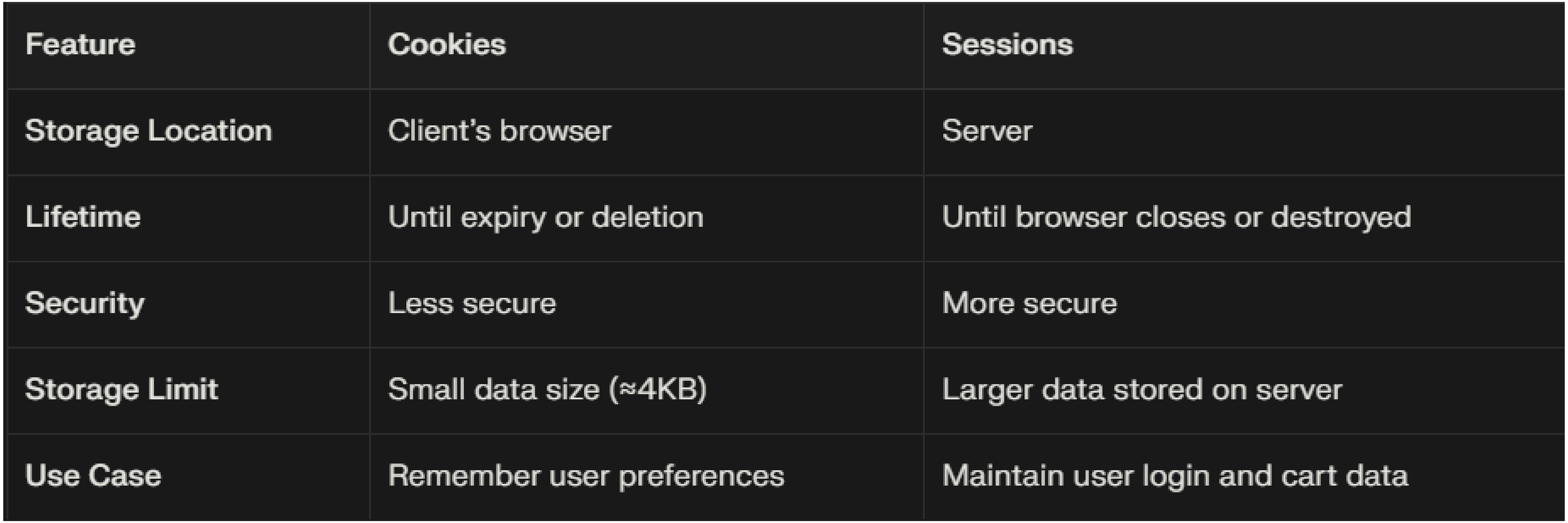
php

```php
session_start();
if(isset($_SESSION['views'])) $_SESSION['views']++;
else $_SESSION['views']=1;
echo 'Views: '.$_SESSION['views'];
```

Lesson 2.4: Destroying a Session

- Remove session variables: session_unset();
- Destroy session completely: session_destroy();
- Clears all user data and ends the session.

Cookies vs Sessions



Feature	Cookies	Sessions
Storage Location	Client's browser	Server
Lifetime	Until expiry or deletion	Until browser closes or destroyed
Security	Less secure	More secure
Storage Limit	Small data size (≈4KB)	Larger data stored on server
Use Case	Remember user preferences	Maintain user login and cart data

THANK YOU !!!

Prepaired By: John Wilmar Ludovice