



Department of Electrical and Electronics Communication Engineering

Subject: Full Stack Development

Semester: VI AY 2025-26

Name: Samyak Bakliwal
Roll No: 1032221792

Class: T. Y. B. Tech.
Batch: G3

Experiment No: 3

Name of the Experiment: JavaScript for PHP Sessions, Cookies, and DOM real-time update.

Performed on:

Submitted on:

Marks	Teacher's Signature with date

Aim: JavaScript for PHP Sessions, Cookies, and DOM real-time update.

Objective:

- Understanding how JavaScript interacts with PHP sessions and cookies for user authentication and data persistence.
- Implementing JavaScript techniques to read, create, and modify cookies for session management.
- Using AJAX to fetch and update PHP session data without refreshing the page.
- Dynamically updating the DOM in real time based on session or cookie changes.

Platform: VS code, PHP, Code runner extension

THEORY:

Instructions: you are requested to write answer of following questions in A4 size pages.

1. What are PHP sessions? And how JavaScript can access PHP session data?
2. What are cookies, and how do they differ from sessions? And how can JavaScript create, read, and delete cookies?
3. What is the Document Object Model (DOM), and how does JavaScript manipulate it?
4. How can JavaScript update the DOM based on changes in PHP session data?

Problem Statement 1:

- Write a program to design Student registration form by using HTML, CSS having following fields: Username, Email, Phone number, Password, Confirm Password and write external javascript code to achieve following validations
 - Fields should not be empty. If spaces are entered those should be considered empty
 - Phone number must accept only numeric values and it should be 10 digits
 - Password length must be at least 7 and it should contain at least one capital letter, one digit and one special character from the set (&,\$,#@)
 - Value entered in password field and confirm password fields must match
 - Email address must contain @ sign and a ., there should be few letters before the @ sign, there should be three letters between @ sign and a . There must be 3 or 2 letters after the . (hint: Use regular expression)

Problem Statement 2:

- Write a client-side script with JavaScript to access and manipulate Document Object Model (DOM) objects in an HTML web page. Develop a dynamic web page using javascript and DOM. Make use of the following for accessing elements
 - getElementById, getElementsByTagName, getElementsByClassName
 - Change the text using innerHTML property
 - Change the CSS properties like color, position of a particular element on the page
 - Change the image source after clicking on a button
 - Add a text node and attach it to a parent node
 - Delete a node
- Include jQuery to perform following operations:
 - Change button text using jQuery.
 - Set background-image using jQuery CSS property.
 - Access HTML form data using jQuery.
 - Add attribute using jQuery
- Use this reference link for jQuery: <https://www.w3resource.com/jquery-exercises/part1/index.php>

Problem Statement 3:

Real-Time User Activity Tracker Scenario:

- Track user activity (e.g., last visited page, current time) using cookies and display it dynamically.
- Requirements:
 - Use PHP to store the user's last visited page and visit timestamp in a cookie.
 - Retrieve the cookie using JavaScript and display the information on the page.
 - Update the cookie whenever the user visits a new page.

Output:

1. Screenshots of the output of problem statement to be attached.
2. Share GitHub repository link of above project

Conclusion:

This experiment demonstrated how JavaScript interacts with PHP sessions, cookies, and the DOM for real-time updates. We implemented secure session management, client-side cookie handling, form validation, AJAX-based data retrieval, and dynamic DOM manipulation using JavaScript and jQuery. The integration of these technologies enhances user experience by enabling seamless, interactive web applications.

Additional Links:

https://youtu.be/SbAfPJj0H4g?si=lzDgy3vcX_CwyV41

<https://youtu.be/SbAfPJj0H4g?si=zVFp4SBVMwAxx95>

https://youtu.be/7zcXPCt8Ck0?si=S_oBHp_ErmlQPbCY

Post lab question:

1. How can AJAX be used to retrieve PHP session data without reloading the page?
2. What is the difference between document.cookie in JavaScript and setcookie() in PHP?
3. What are some best practices for efficiently updating the DOM without causing performance issues?

Name - samyak bakiwal (SD) - Ty elec 9/ comp.
PRN - 1032221792

experiment no -- 3

Theory:-

Q1) what are PHP sessions? how Javascript can access PHP session data?

PHP sessions store data on the server & associate it with a session id. Javascript can't directly access PHP session data, but it can make AJAX requests to PHP script, that returns session data.

Q2) what are cookies & how do they differ from session? and how can javascript create, read & delete cookies?

cookies store data on the client-side and persist across sessions. sessions store data server-side and typically expire when the browser is closed. javascript can create using document cookies, read them & delete them by setting an expiration date in past.

Q3) what is DOM or how does javascript manipulate it?

→ The document object model (DOM) represents the HTML structure of a webpage as a tree of objects. JavaScript manipulates it by selecting elements or changing their properties, attributes or content.

Q4) How can javascript update the DOM based on changes in PHP session data?

→ To update DOM based on PHP session use AJAX to send PHP session data to the client side then update the DOM with javascript based on the response.

⇒ Post lab Question:-

Q1) How can AJAX be used to retrieve PHP session data without reloading the page?

⇒ AJAX can send request to PHP scripts that return session data as JSON or HTML, enabling dynamic content updates without reloading the page.

Q2)

What is the difference bⁿ document, cookie in javascript & setcookie () in PHP?

⇒

document : cookie is used in Java script to read & write cookies directly in the browser.

set cookie () in PHP sends cookies to the browser via HTTP header.

Q3)

What are some best practices for efficiently updating the DOM without causing performance issues?

⇒

Some best practices for efficiently updating DOM are:-

- Minimize DOM manipulation.
- Use document fragment.
- Defer / throttle events.
- Avoid layout thrashing.
- Use request animation frame.
- Leverage CSS for animation.
- Batch event handlers.
- Limit reflow & repaints.

Dr.
h.2.25