



KEMENTERIAN PENDIDIKAN TINGGI
JABATAN PENDIDIKAN POLITEKNIK DAN KOLEJ KOMUNITI



PROBLEM BASED TASK 1

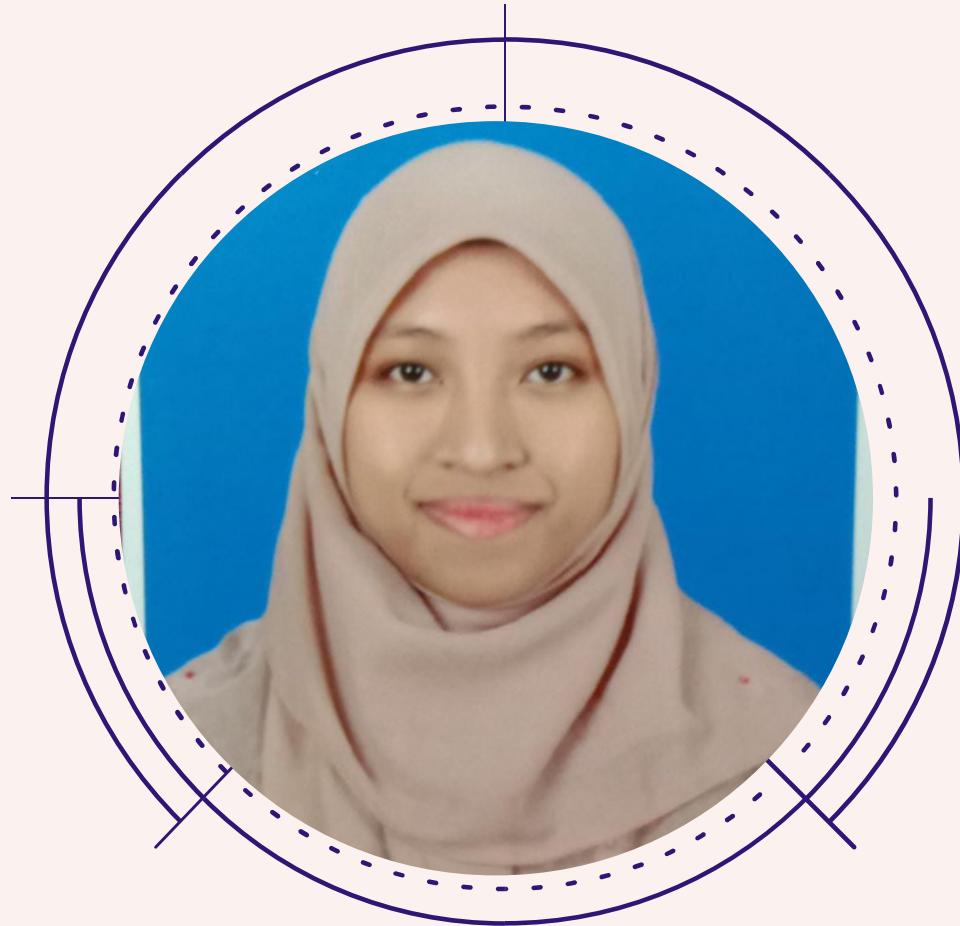
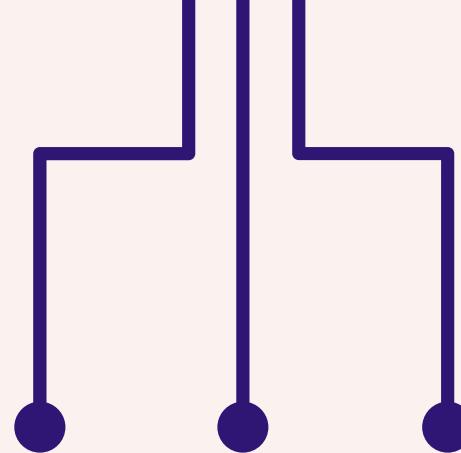
DFP50193 | WEB PROGRAMMING

-MOHAMAD IQBAL HAKIM BIN CHE OMAR-

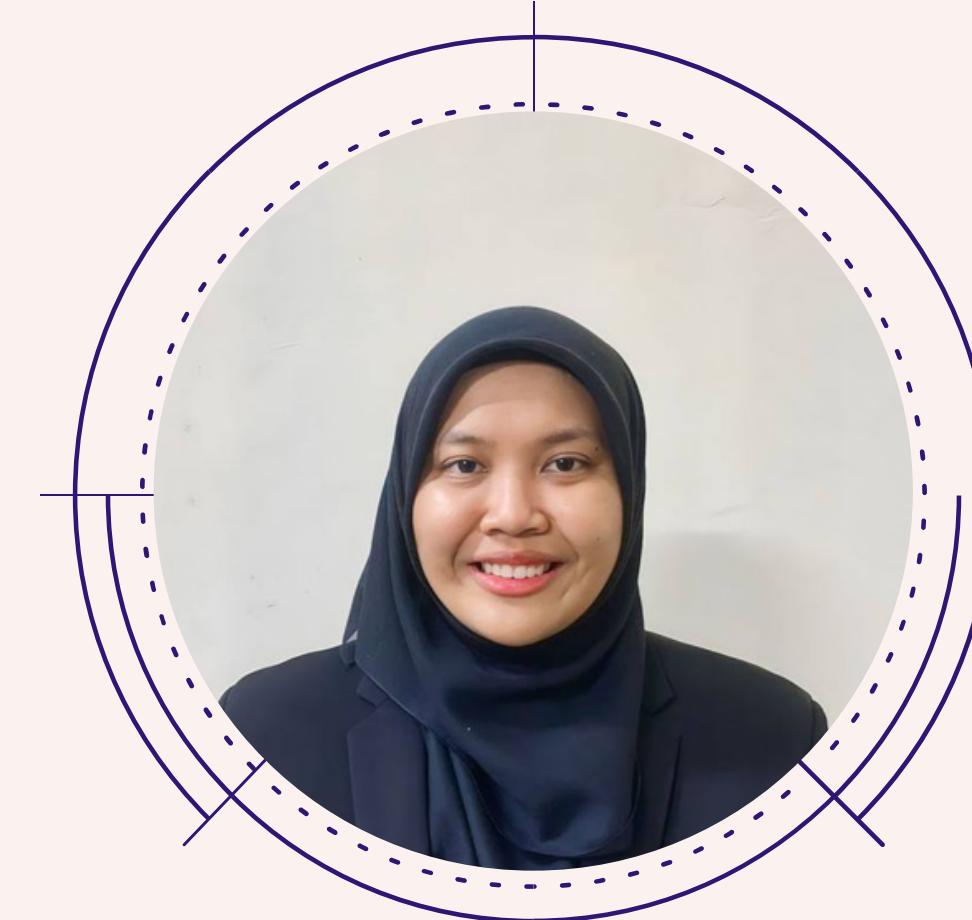
Presented by :

- 1) SITI NORAISHAH A HAIDI (25DDT21F1037)
- 2) FARAH HANIS BINTI AZAMAN (25DDT21F1023)

TEAM MEMBERS



SITI NORAISSAH A HAIDI
25DDT21F1037



FARAH HANIS BINTI AZAMAN
25DDT21F1023

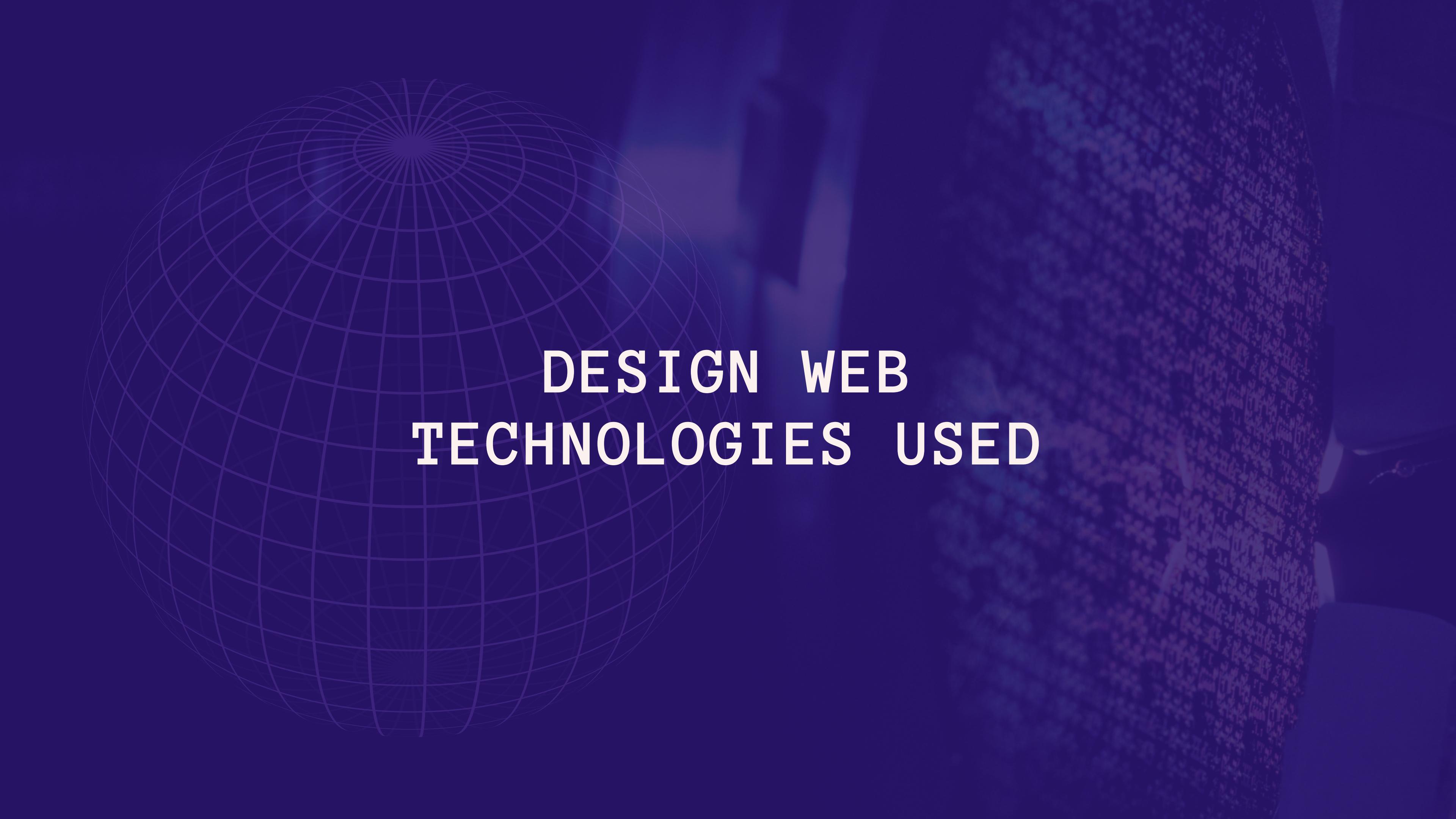
INSTRUCTION

Instruction:

Present the completed Login with marks calculation and biodata page in PBT 1 that involve: (CLO2, A3)

- a. Design web technologies used.
- b. Client-side scripting and validation.
- c. Sessions - purpose, process of creating and destroy session and setting session variables in PHP.
- d. Differences between Sessions and Cookies method.
- e. Store and retrieve data.
- f. Best practices to follow when developing web applications to requirement and function correctly without error and bug.

*If plagiarism (submitting the works of others as your own or allowing someone else to copy and submit your work as their own) occurs, all group members will receive zero marks for the project.

The background features a large, semi-transparent globe on the left side, showing a grid of latitude and longitude lines. To the right, a person's profile is visible, looking towards the left. The overall color palette is dark, with deep blues and blacks.

DESIGN WEB TECHNOLOGIES USED



CSS

Cascading Style Sheets

Design the interface to make it more interesting and more understandable



HTML

Hypertext Markup Language

It is designed to be displayed in a web browser. Structure the contents



SCRIPT

Create calculation function and display average marks

Flow



login.php

Welcome, Mr Lee as guest

[Log Out](#)

Full Name: [...](#)

Age:

Email:

[Update](#)

Lab Task for Web Programming

Lab Task 1:

Lab Task 2:

Lab Task 3:

[Calculate Average](#)

Your Average marks is: 92.67 - EXCELLENT

main.php

Your Biodata

Name: Mr Lee Ah Chong

Age: 30

Email: lee@gmail.com

[Back to Main Page](#)

[Log Out](#)

biodata.php

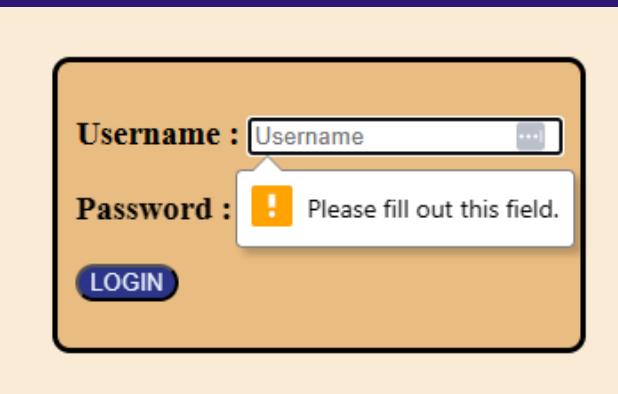


CLIENT SIDE SCRIPTING AND VALIDATION



CLIENT SIDE SCRIPTING AND VALIDATION

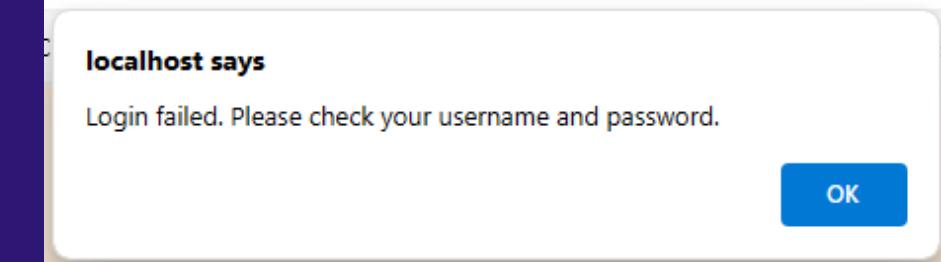
USERNAME & PASSWORD -CLIENT SIDE VALIDATION-



```
<h3>Username :  
<input type="text" name="username" placeholder="Username" required="true" id="username"></input></h3>  
<h3>Password :  
<input type="password" placeholder="Password" required="true" name="password" id="password"></input></h3>
```

The HTML form includes the `required="true"` attribute for the username and password fields. Means that the user must fill in these fields before submitting the form.

WRONG USERNAME & PASSWORD -CLIENT SIDE SCRIPTING-



```
if ($enteredUsername === $correctUsername && $enteredPassword === $correctPassword) {  
    // Set the username and password in session variables  
    $_SESSION['username'] = $enteredUsername;  
    $_SESSION['password'] = $enteredPassword;  
    header("Location: main.php");  
    exit();  
} else {  
    $loginError = '<script>alert("Login failed. Please check your username and password.")</script></p>';  
}
```

JavaScript alert is used to show a pop-up message if the login credentials entered by the user do not match the expected values. This JavaScript code is embedded in the `$loginError` variable, and it's displayed as an alert message.

CLIENT SIDE SCRIPTING AND VALIDATION

FULL NAME

Log Out

Full Name:

Age: ! Please fill out this field.

Email:

Update

AGE

Log Out

Full Name: Mr Lee Ah Chong

Age:

Email: ! Please fill out this field.

Update

EMAIL

Log Out

Full Name: Mr Lee Ah Chong

Age:

Email: ! Please fill out this field.

Update

```
<h3>Full Name:  
<input type="text" placeholder="enter FullName" required="true" name="name" id="name" value=<?php echo $fullname;  
</h3>  
<h3>Age:  
<input type="text" placeholder="enter Age" required="true" name="age" id="age" value=<?php echo $age; ?>></h3>  
<h3>Email:  
<input type="text" placeholder="enter your Email" required="true" name="email" id="email" value=<?php echo $email;  
<button type="submit" name="update" id="update">Update</button>
```

For updating user information includes the `required="true"` attribute for fields like "Full Name," "Age," and "Email." This attribute ensures that users must fill in these fields before they can submit the form. If any of these required fields are left empty, the browser itself prevents the form from being submitted.

CLIENT SIDE SCRIPTING AND VALIDATION

FUNCTION CALCULATE AVERAGE

Lab Task for Web Programming

Lab Task 1:

Lab Task 2:

Lab Task 3:

Calculate Average

Your Average marks is: 90.00 - EXCELLENT

```
// Get the result element
const resultElement = document.getElementById('result');
var message = "";
var bgColor = "";

if (average >= 80 && average <= 100) {
    message = 'EXCELLENT';
    bgColor = "#00cc00"; // Green
} else if (average >= 60 && average <= 79) {
    message = 'NEED MORE EXERCISE';
    bgColor = "#ffcc00"; // Yellow
} else if (average >= 40 && average <= 59) {
    message = 'NEED MORE EXERCISE AND FOCUS IN THE CLASS';
    bgColor = "#ff3300"; // Orange
} else if (average >= 0 && average <= 39) {
    message = 'PLEASE SEE ME';
    bgColor = "#990000"; // Red
} else {
    message = 'INVALID DATA';
    bgColor = "#990000"; // Red
}
```

- JavaScript function called calculateAverage that calculates the average of three lab task scores entered by the user.
- The script checks whether the values entered are valid numbers, and if not, it displays an alert to the user.
- It also changes the background color of a result element based on the calculated average.



SESSIONS

PURPOSE, PROCESS OF CREATING AND DESTROY SESSIONS AND SETTING SESSIONS VARIABLES IN PHP

PURPOSE OF SESSION

To store important information more securely on the server where malicious users cannot temper with them. It is a temporarily storing and making data accessible across all the website pages. It will create a temporary file that stores various session variables and their values. This will be destroyed when you close the website.(logout)

CREATING SESSION

Start a session at the beginning of PHP script using **session_start()**. Located at the top of your PHP script before any HTML or content is sent to the browser

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

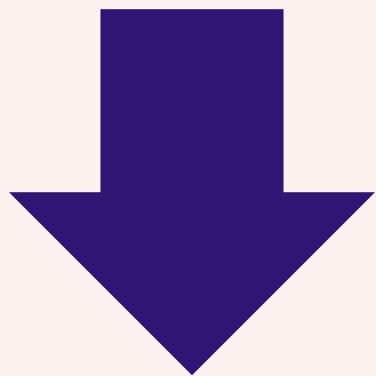
DESTROY SESSION

```
<?php  
// Start the session  
session_start();  
  
// Clear cookies by setting their expiration time 1 hour from the current time  
setcookie('fullname', '', time() - 3600);  
setcookie('age', '', time() - 3600);  
setcookie('email', '', time() - 3600);  
  
// Destroy the session to log the user out  
session_destroy();  
  
// Redirect to login.php after clearing cookies and sessions  
header('Location: login.php');  
exit();  
?>
```

destroy a session when a user logs out or when a session expires.

right click > inspect > application

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Partition Key	Priority
PHPSESSID	oliddpiiheq7gl7rheakh9ipne	localhost	/	Session	35					Medium
remember_token	55401ce8a21992524610c06cc681d210aee49f4bf827e327ac6c5096c4a2c66d	localhost	/	2023-11-03T10:24:55.981Z	78					Medium
age	30	localhost	/F1037_F1023_PBT1	2023-10-23T15:48:28.722Z	5					Medium
email	lee%40gmail.com	localhost	/F1037_F1023_PBT1	2023-10-23T15:48:28.722Z	20					Medium
fullname	Mr%20Lee%20Ah%20Chong	localhost	/F1037_F1023_PBT1	2023-10-23T15:48:28.719Z	29					Medium



Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Partition Key	Priority
PHPSESSID	oliddpiiheq7gl7rheakh9ipne	localhost	/	Session	35					Medium
remember_token	55401ce8a21992524610c06cc681d210aee49f4bf827e327ac6c5096c4a2c66d	localhost	/	2023-11-03T10:24:55.981Z	78					Medium

SESSION VARIABLE

store username
and password
using variables

call by variables

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    // Pre-determined username and password  
    $correctUsername = "Mr Lee";  
    $correctPassword = "1003";  
  
    $enteredUsername = $_POST["username"];  
    $enteredPassword = $_POST["password"];  
  
    // Check if the entered credentials match the pre-determined values  
    if ($enteredUsername === $correctUsername && $enteredPassword === $correctPassword) {  
        // Set the username and password in session variables  
        $_SESSION['username'] = $enteredUsername;  
        $_SESSION['password'] = $enteredPassword;  
        header("Location: main.php");  
        exit();  
    }  
}
```

set session variables to store data that you want to persist across pages.



DIFFERENCES BETWEEN SESSION AND COOKIES METHOD

SESSION	COOKIES
Session store data on the server	Cookies store data on the client's browser
<ul style="list-style-type: none">• Session typically expire when a user close the browser or log out• Suitable for temporary data storage	<p>Cookies can have both session-only</p> <ul style="list-style-type: none">• Delete when the browser is closed• lasting for specified duration
Suitable for sensitive information such as password	Suitable for non-sensitive data such as user preferences and tracking information

APPLY METHOD SESSION & COOKIES

Method Session : For store sensitive data that should not be exposed to the client. We decide to use session method to store username & password.

Method Cookies: For store data such as Full Name, Age and Email all of this in main.php we decide to use cookies and then the data will be display to biodata.php.

A large, semi-transparent globe with a grid pattern of latitude and longitude lines is positioned on the left side of the frame. To its right, a person's face is shown in profile, looking towards the left. The background is a dark, textured surface.

STORE AND RETRIEVE DATA

sessions

Username :

Password :

LOGIN



- Retrieve username inserted from ‘login.php’ using session

```
// Check if the user is logged in and retrieve their username from the session
if (isset($_SESSION['username'])) {
    $username = $_SESSION['username'];
```

Welcome, **Mr Lee** as guest

- Store value of username inserted from ‘login.php’ at ‘main.php’

```
<h1>Welcome, <?php echo $username; ?> as guest</h1>
```

cookies

Welcome, Mr Lee as guest

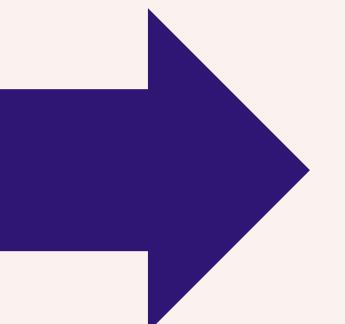
[Log Out](#)

Full Name: [...]

Age:

Email:

[Update](#)



Your Biodata

Name: Mr Lee Ah Chong

Age: 30

Email: lee@gmail.com

[Back to Main Page](#)

[Log Out](#)

```
<h3>Full Name:  
<input type="text" placeholder="enter FullName" required="true" name="name" id="name" value=<?php echo $fullname; ?>>  
</h3>
```

```
<h3>Age:  
<input type="text" placeholder="enter Age" required="true" name="age" id="age" value=<?php echo $age; ?>></h3>
```

```
<h3>Email:  
<input type="text" placeholder="enter your Email" required="true" name="email" id="email" value=<?php echo $email; ?>></h3>
```

```
if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST['update'])) {  
    // Store the form data as cookies  
    setcookie('fullname', $_POST['name'], time() + 3600); // Update 'name' to 'fullname'  
    setcookie('age', $_POST['age'], time() + 3600);  
    setcookie('email', $_POST['email'], time() + 3600);  
  
    // Redirect to the biodata.php page  
    header("Location: biodata.php");  
    exit();  
}
```

```
<h1>Your Biodata</h1>  
<p>Name: <?php echo $fullname; ?></p>  
<p>Age: <?php echo $age; ?></p>  
<p>Email: <?php echo $email; ?></p>
```

- Retrieve full name, age and email inserted from ‘main.php’ using cookies and it store at browsing

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Partition Key	Priority
PHPSESSID	oliddpiiheq7gl7rheakh9ipne	localhost	/	Session	35					Medium
remember_token	55401ce8a21992524610c06cc681d210aee49f4bf827e327ac6c5096c4a2c66d	localhost	/	2023-11-03T10:24:55.981Z	78					Medium
age	30	localhost	/F1037_F1023_PBT1	2023-10-23T15:48:28.722Z	5					Medium
email	lee%40gmail.com	localhost	/F1037_F1023_PBT1	2023-10-23T15:48:28.722Z	20					Medium
fullname	Mr%20Lee%20Ah%20Chong	localhost	/F1037_F1023_PBT1	2023-10-23T15:48:28.719Z	29					Medium



**BEST PRACTICES TO FOLLOW WHEN
DEVELOPING WEB APPLICATION TO
REQUIREMENT AND FUNCTION
CORECCTLY WIHTOUT ERROR AND BUG**

BEST PRACTICE WHEN DEVELOPING WEB APPLICATION

1. SESSION MANAGEMENT

We use `session_start()` at the beginning of PHP files (`login.php`) is a best practice for managing user sessions. It enables the use of session variables to maintain user data across multiple pages.

2. PROPER USE OF COOKIES

We implement Cookies to store user data like full name, age, and email. This is a common practice for maintaining user information between sessions.

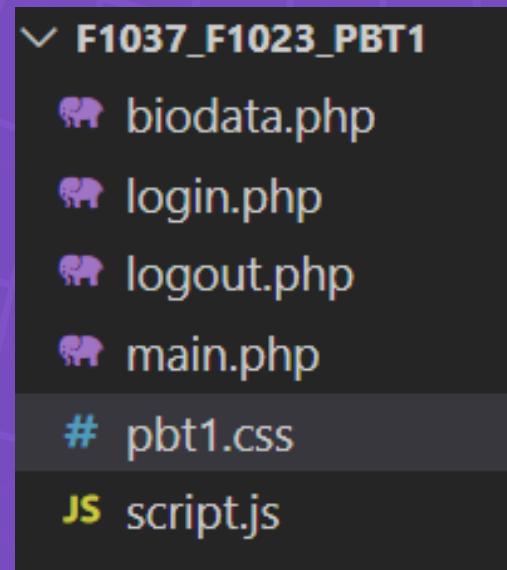
3. REDIRECTION FOR UNAUTHENTICATED USERS

In `main.php`, if a user is not logged in, they are redirected back to the login page. A good security practice to ensure that unauthorized users can't access sensitive information.

BEST PRACTICE WHEN DEVELOPING WEB APPLICATION

4. SEPARATION OF CONCERNS

We organize HTML, PHP, and JavaScript into separate files, promoting a clean separation of concerns and making the code easier to maintain and understand.



5. CLIENT-SIDE VALIDATION

HTML forms, the required attribute is used for input fields, enforcing basic client-side validation by ensuring that users fill in necessary fields. This provides a better user experience by preventing empty submissions.

THANK YOU!