Web Technology Assignment

V. Jaya sai reddy

22WU0104052

AIML-A

1) List and describe the different types of form controls available.

A) Form controls in HTML allow users to enter and manipulate data. Below are the most common types:

1. Text Box (<input type="text">)

- **Purpose**: Used for single-line text input, such as names, email addresses, or other short strings of data.
- It supports various attributes like maxlength to limit the number of characters, placeholder to show a hint inside the input box, and pattern for input validation.

2. Password Field (<input type="password">)

- **Purpose**: Used to input sensitive information, such as passwords, where the input characters are masked.
- Although the input is hidden on the screen, the data is still accessible, so encryption or hashing is recommended when sending this data to a server for security.

3. Radio Buttons (<input type="radio">)

- **Purpose**: Allows users to select **one** option from a group of choices. Typically used when only one option is applicable.
- Radio buttons with the same name attribute are grouped together, ensuring that only one can be selected at a time.

4. Checkboxes (<input type="checkbox">)

- **Purpose**: Allows users to select **one or more** options from a group, often used for selecting preferences like hobbies or subscription options.
- Multiple checkboxes can be checked independently, and they can have a value attribute to specify the value that will be sent when the form is submitted.

5. Dropdown (<select> with <option>)

- **Purpose**: Provides a list of options in a dropdown menu. Users can select one (or multiple if multiple attribute is set) option(s).
- Dropdowns save space in forms and are useful when there are many predefined options, like selecting a country or a department.

6. Textarea (<textarea>)

- **Purpose**: Allows users to input multi-line text, such as comments, descriptions, or longer messages.
- Unlike the text input, the size of a <textarea> can be adjusted by setting its rows and cols attributes, or it can be resized dynamically by the user.

7. File Upload (<input type="file">)

- **Purpose**: Enables users to browse and upload files from their local machine.
- The accept attribute can be used to limit the file types that can be selected (e.g., accept="image/*" for images, or accept=".pdf" for PDFs).

8. Submit Button (<input type="submit">)

- **Purpose**: Sends the form data to the server for processing when clicked.
- It triggers the form's action (usually sending data to the server) and can be styled to look like a regular button or even an image with input type="image".

9. Reset Button (<input type="reset">)

- **Purpose**: Resets all form fields to their default values or the initial state.
- It clears the form fields, but it cannot reset fields that have been dynamically changed by JavaScript unless the page is reloaded.

10. Hidden Field (<input type="hidden">)

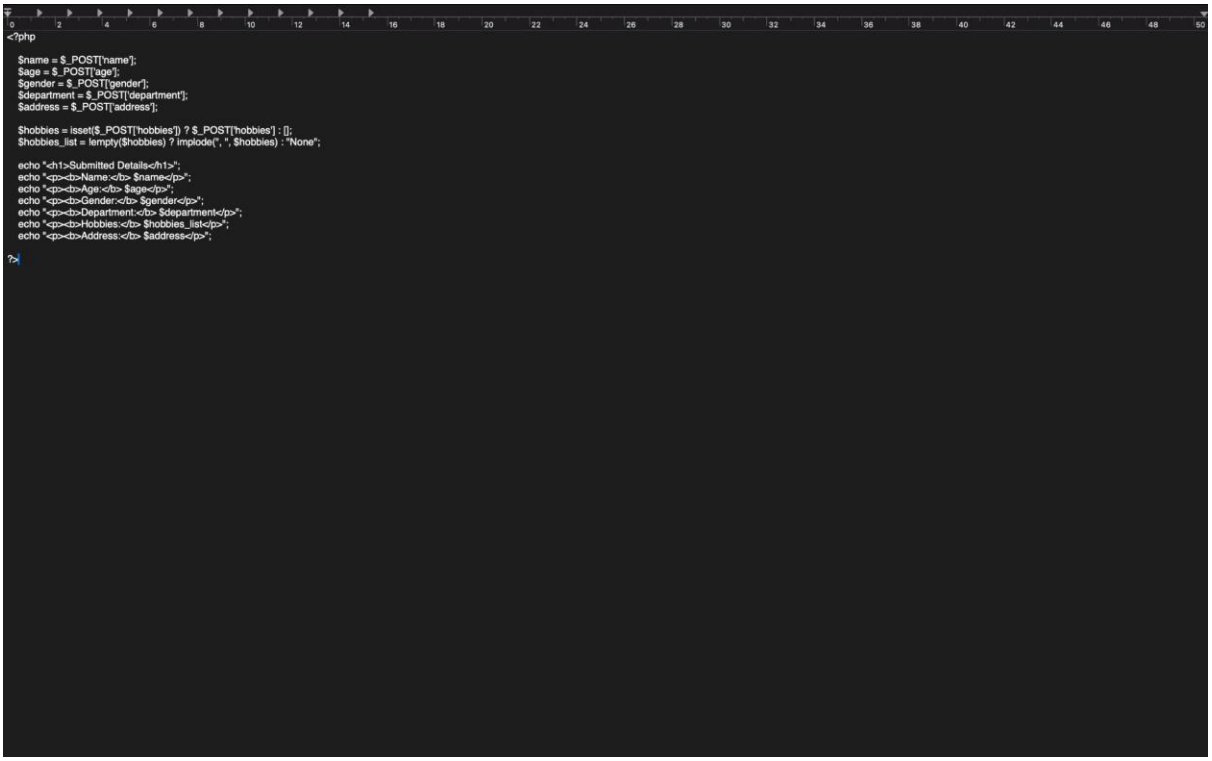
- **Purpose**: Stores hidden data that users cannot see or interact with, but is included when the form is submitted.

2) Design an HTML form containing all major input controls: text box, password field, radio buttons, checkboxes, dropdown, text area, and file upload. Create a PHP script to display the submitted values.

A) HTML Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>User Details Form</title>
</head>
<body>
    <h1>Enter Your Details</h1>
    <form action="#" method="POST">
        <label for="name">Name:</label>
        <input type="text" id="name" name="name"><br><br>

        <label for="age">Age:</label>
        <input type="number" id="age" name="age"><br><br>

        <label>Gender:</label>
        <input type="radio" id="male" name="gender" value="male">
        <label for="male">Male</label>
        <input type="radio" id="female" name="gender" value="female">
        <label for="female">Female</label>
        <input type="radio" id="other" name="gender" value="other">
        <label for="other">Other</label><br><br>

        <label for="department">Department:</label>
        <select id="department" name="department">
            <option value="cse">Computer Science</option>
            <option value="aiml">AI & ML</option>
            <option value="cyber">Cyber Security</option>
        </select><br><br>

        <label>Hobbies:</label>
        <input type="checkbox" id="reading" name="hobbies" value="reading">
        <label for="reading">Reading</label>
        <input type="checkbox" id="traveling" name="hobbies" value="traveling">
        <label for="traveling">Traveling</label>
        <input type="checkbox" id="sports" name="hobbies" value="sports">
        <label for="sports">Sports</label>
        <input type="checkbox" id="music" name="hobbies" value="music">
        <label for="music">Music</label>
        <input type="checkbox" id="cooking" name="hobbies" value="cooking">
        <label for="cooking">Cooking</label><br><br>

        <label for="address">Address:</label><br>
        <textarea id="address" name="address" rows="4" cols="50"></textarea><br><br>

        <input type="submit" value="Register">
    </form>
</body>
</html>
```

PHP Code :



```php
<?php

    $name = $_POST['name'];
    $age = $_POST['age'];
    $gender = $_POST['gender'];
    $department = $_POST['department'];
    $address = $_POST['address'];

    $hobbies = isset($_POST['hobbies']) ? $_POST['hobbies'] : [];
    $hobbies_list = !empty($hobbies) ? implode(", ", $hobbies) : "None";

    echo "<h1>Submitted Details</h1>";
    echo "<p><b>Name:</b> $name</p>";
    echo "<p><b>Age:</b> $age</p>";
    echo "<p><b>Gender:</b> $gender</p>";
    echo "<p><b>Department:</b> $department</p>";
    echo "<p><b>Hobbies:</b> $hobbies_list</p>";
    echo "<p><b>Address:</b> $address</p>";

?>
```

OUTPUT

Enter Your Details

Name: Jaya sai reddy

Age: 19

Gender: (•) Male ( ) Female ( ) Other

Department: AI & ML

Hobbies: [x] Reading [ ] Traveling [x] Sports [x] Music [x] Cooking

Address:

Kadapa

Register

3. What are the different file handling functions in PHP?

A) PHP provides various functions to work with files:

- fopen($filename, $mode) – Opens a file in the specified mode (r, w, a, etc.).
- fclose($file) – Closes an open file.
- fwrite($file, $content) – Writes content to a file.
- fread($file, $length) – Reads a file up to a specified length.
- file_get_contents($filename) – Reads the entire file into a string.
- file_put_contents($filename, $data) – Writes data to a file (overwrites if exists).
- unlink($filename) – Deletes a file.
- rename($oldname, $newname) – Renames a file.

Example: Writing and Reading a File php

CopyEdit

<?php

$file = fopen("example.txt", "w"); fwrite($file, "Hello, PHP File

Handling!"); fclose($file);

$content = file_get_contents("example.txt"); echo "File Content: " .

$content;

?>

4) Compare and contrast cookies and sessions in PHP. Which one is more secure? How can you delete a cookie in PHP?
What happens when a cookie expires? How do you destroy a session in PHP? Explain the use of session unset and session destroy.

A) Cookies vs. Sessions in PHP

Feature	Cookies	Sessions
Storage	Stored on the client's browser.	Stored on the server.
Lifetime	Can persist after closing the browser Ends when the browser is closed (based on expiration time). or session expires.	
Security	Less secure (can be accessed or modified More secure as data is stored on by the user). the server.	
Usage	Suitable for storing small amounts of data, Used for storing sensitive data like like user preferences. login details.	

Which is More Secure?

Sessions are more secure than cookies because they are stored on the server, making them less vulnerable to user manipulation.

Deleting a Cookie in PHP php CopyEdit setcookie("user", "",
time() - 3600, "/");
This sets the cookie's expiration time in the past, effectively deleting it.

What Happens When a Cookie Expires?

The browser automatically removes the expired cookie, and it is no longer accessible.

Destroying a Session in PHP
php CopyEdit session_start();
session_destroy();

This completely removes all session data from the server.

Difference Between session_unset() and session_destroy()

- session_unset(); – Clears all session variables but keeps the session active.
- session_destroy(); – Ends the session and deletes session data on the server.

Example: Using Sessions

php
CopyEdit <?php
session_start();
$_SESSION['username'] = "JohnDoe";

echo "Username: " . $_SESSION['username'];

session_unset(); // Unsets session variables session_destroy(); //
Destroys the session ?>