# PRACTICAL 1



1. **Demonstration Of Web Browsers: Different components, Checking SSL Certificates, Inspect Elements, Browser Console, view Source etc.**

Components of Web Browsers:

* 1. Address Bar: Where you enter the URL of the website you want to visit.
  2. Navigation Buttons: Back, Forward, and Refresh buttons to navigate between pages and reload content.
  3. Tabs: Allow you to have multiple web pages open simultaneously within the same window.
  4. Toolbar: Contains icons for various functions like bookmarks, settings, and extensions.
  5. Menu: Access additional browser features and settings. Browser Features:

1. View Page Source:
   * Right-click on a web page and select "View Page Source" or use keyboard shortcuts (Ctrl+U or Cmd+Option+U on Mac).
   * Shows the HTML code of the webpage.
2. Inspect Elements:
   * Right-click on an element on the webpage and select "Inspect" or use keyboard shortcuts (Ctrl+Shift+I or Cmd+Option+I on Mac).
   * Opens the browser's developer tools to inspect HTML structure, CSS styles, and modify elements in real-time.
3. Browser Console:
   * Accessed through the developer tools (usually under the "Console" tab).
   * Displays messages from the webpage and JavaScript errors, allows you to execute JavaScript code.
4. Checking SSL Certificates:
   * Click on the padlock icon next to the URL in the address bar.
   * Shows details about the website's SSL certificate, including issuer, validity period, and encryption details.
5. Cookies and Site Data:
   * Manage stored cookies and site data through browser settings.
   * Important for managing login sessions and personalized content.
6. Extensions and Add-ons:
   * Enhance browser functionality with extensions for ad-blocking, password management, and more.
   * Accessible through the browser's settings or extension store. Examples and Demonstrations:

* Viewing Page Source: Open any webpage, right-click, and choose "View Page Source" to see the HTML structure.
* Inspecting Elements: Right-click on an element (like a button or text) on a webpage and select "Inspect" to see its HTML and CSS properties. You can modify these in real-time to see immediate changes.
* Using Browser Console: Open the console from the developer tools and type JavaScript commands like console.log("Hello, World!") to see output directly in the console.
* Checking SSL Certificates: Visit a secure website (https://), click on the padlock icon in the address bar, and view certificate details to check its validity and issuer information.

# PRACTICAL 2



1. **Basic HTML Programming**
   1. **Creating Sample HTML pages using tags like, headers, paragraphs, alignments, divisions, lists etc.**
   2. **Create tables in HTML.**
   3. **Displaying Images in HTML.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Pr2</title>

</head>

<body>

<header>Hello World</header>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque laborum minus inventore illum voluptate molestias nesciunt fugit molestiae vero, eum aliquid eaque ipsam, error consequatur aut quidem autem porro voluptatum odit ab illo?</p>

<h1>I am H1</h1>

<h2>I am H2</h2>

<h3>I am H3</h3>

<h4>I am H4</h4>

<h5>I am H5</h5>

<h6>I am H6</h6>

<div class="Try">

<h3>I am in "Div"</h3>

</div>

<ul>

<li>1</li>

<li>2</li>

<li>3</li>

</ul>

<ol>

<li>A</li>

<li>B</li>

<li>C</li>

</ol>

<dl>

<dt>cofee</dt>

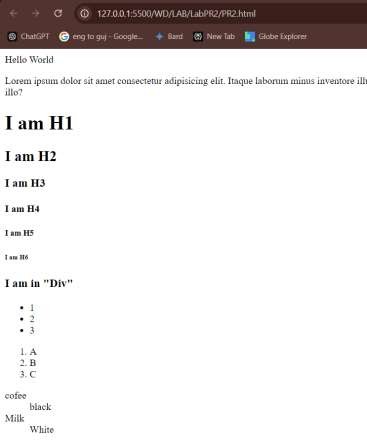
<dd>black</dd>

<dt>Milk</dt>

<dd>White</dd>

</dl>

</body>

</html>

* **Output:**

1. **HTML Forms**



PRACTICAL 3

* 1. **Basics of Form Development Practical: text fields, radios, buttons, checkboxes.**
  2. **Advanced Form Development Practical: Combo, Date, File Upload.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>PR3</title>

</head>

<body>

<form>

<label for="fname">First Name:</label>

<input type="text" id="fname" name="fname"><br><br>

<label for="lname">Last name:</label>

<input type="text" id="lname" name="fname"><br><br>

<input type="submit" value="submit"><br><br>

</form>

<button type="button">Click Me!</button><br>

<input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">

<label for="vehicle1"> I have a bike</label><br>

<input type="checkbox" id="vehicle2" name="vehicle2" value="Car">

<label for="vehicle2"> I have a car</label><br>

<input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">

<label for="vehicle3"> I have a boat</label><br>

<input type="date" id="date" name="date">

<label for="date">Date</label>

<form>

<input type="file" id="myFile" name="filename">

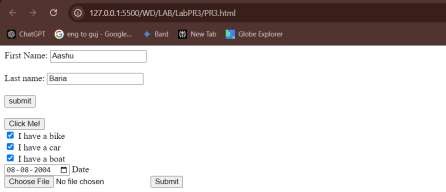
<input type="submit">

</form>

</body>

</html>

## Output:



1. **CSS Programming**



**PRACTICAL 4**

* 1. **Basics of CSS programming Practical: Class, Id, changing properties like color, size, background etc.**
  2. **CSS 3 Programming Practical: shadow, orientation, transformation, gradient etc.**
  3. **Positioning practical in CSS: Absolute and relative positioning,**

**Z-index.**

## HTML:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>PR4</title>

<link rel="stylesheet" href="PR4.css">

</head>

<body>

<div id="MyID">Hello I am ID1</div>

<!-- <div id="MyID">Hello I am ID2</div> -->

<div class="MyClass">Hello I am in Class 1</div>

<div class="MyClass">Hello I am in Class 2</div>

<div class="shadow">My Shadow</div>

<div class="a">

<h3>Mixed oriantation</h3>

<p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Fugit alias similique fuga dignissimos mollitia maxime repellendus aut inventore nesciunt odio.</p>

</div>

<div class="b">

<h3>Mixed oriantation</h3>

<p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Fugit alias similique fuga dignissimos mollitia maxime repellendus aut inventore nesciunt odio.</p>

</div>

<div class="Transform1">

<p>Hello I am Right</p>

</div>

<div class="Transform2">

<p>Hello I am Left</p>

</div>

<div class="position">

<p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Consequuntur consectetur quisquam eaque beatae reprehenderit error eos sunt corporis in doloremque maxime mollitia veritatis iure impedit, exercitationem iusto hic quam inventore.</p>

</div>

<div class="x">

<p>X</p>

</div>

<div class="y">

<p>Y</p>

</div>

<div class="z">

<p>Z</p>

</div>

</body>

</html>

## CSS:

\*{

margin: 0%;

padding: 0%;

}

#MyID{

background-color: red;

background-image: linear-gradient(red,yellow); font-size: 36px;

text-align: center; border: 2px solid black;

/\* padding: 10px 10px; \*/

}

.MyClass{

background-color:aquamarine ; padding: 10px 10px;

}

.shadow{

text-shadow: 2px 2px;

}

div.a{

writing-mode: sideways-lr; text-orientation: mixed; background-color: aqua; border: 2px solid black; height: 350px;

width: 300px; margin: 5px;

}

div.b{

writing-mode: vertical-lr; /\*most imp \*/ text-orientation: upright; /\*most imp \*/ background-color: aqua;

border: 2px solid black; height: 350px;

width: 300px; margin: 5px;

}

.Transform1{ width: 200px; height: 200px;

background-color: blue; transform: rotate(20deg);

}

.Transform2{ width: 100px; height: 100px;

background-color: blue; transform:skewX(1.5);

}

.position{ width: 500px;

height: 500px;

border: solid black 2px; position:sticky;

}

.x{

background-color: yellow;

border: 2px solid black; width: 50px;

height: 50px; z-index: 1;

position: absolute;

/\* left: 0px; top: 0px; \*/

}

.y{

background-color: yellow; border: 2px solid black; width: 50px;

height: 50px; z-index: 2;

position: absolute;

}

.z{

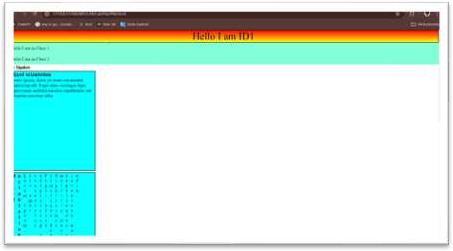
background-color: yellow; border: 2px solid black; width: 50px;

height: 50px; z-index: 3;

position: absolute;

}

## Output:





1. **JavaScript**



**PRACTICAL 5**

* 1. **Basic JS Practical: script tags, alerts, documents, functions, arrays, loops, and conditions**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Basic JavaScript Practical</title>

</head>

<body>

<h1>JavaScript Basics</h1>

<p id="output"></p>

<script>

alert("Welcome to Basic JavaScript!");

function greetUser(name){ return "Hello, "+ name +"!";

}

let userName=prompt("Enter your name:"); document.getElementById('output').innerHTML=greetUser(userName);

let numbers=[1,2,3,4,5];

for(let i=0;i<numbers.length;i++){ console.log("Number: "+numbers[i]);

}

if(numbers.length>3){

console.log("The array has more than 3 elements.");

}else{

console.log("The array has 3 or fewer element.");

}

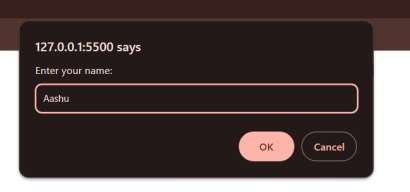
</script>

</body>

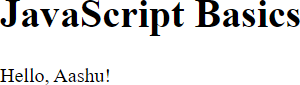
</html>

## Output:









* 1. **Advanced JS Practical: Objects, DOM references: getElementById, InnerHTML etc.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Advanced JavaScript Practical</title>

</head>

<body>

<h1>Advanced JavaScript DOM Manipulation</h1>

<div id="person-info">

<p id="name"></p>

<p id="age"></p>

<p id="location"></p>

</div>

<button onclick="updatePerson()">Update Person Info</button>

<script>

let person = {

name: "Aashu Baria", age: 21,

location: "Godhra"

};

function updatePerson() {

document.getElementById('name').innerHTML = "Name: " + person.name; document.getElementById('age').innerHTML = "Age: " + person.age; document.getElementById('location').innerHTML = "Location: " + person.location;

}

</script>

</body>

</html>



* **Output:**

# PRACTICAL 6



1. **JavaScript and Event Listeners**
   1. **Listeners and JavaScript Practical: Mouse motion, movement, keyboard.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>JavaScript Event Listeners</title>

<style>

#mouse-box { width: 300px; height: 200px;

background-color: lightblue; border: 2px solid black; margin: 20px;

text-align: center; line-height: 200px;

}

#key-box {

margin-top: 20px; padding: 10px;

border: 2px solid #ccc;

}

</style>

</head>

<body>

<h1>JavaScript Event Listeners</h1>

<div id="mouse-box">Move your mouse here!</div>

<p id="mouse-output">Mouse position will be shown here</p>

<input type="text" id="key-box" placeholder="Type something...">

<p id="key-output">Typed key will appear here</p>

<script>

// Mouse movement event listener

const mouseBox = document.getElementById('mouse-box'); const mouseOutput = document.getElementById('mouse-output');

mouseBox.addEventListener('mousemove', function (event) {

mouseOutput.innerHTML = `Mouse X: ${event.clientX}, Mouse Y: ${event.clientY}`;

});

// Keyboard event listener

const keyBox = document.getElementById('key-box'); const keyOutput = document.getElementById('key-output');

keyBox.addEventListener('keydown', function (event) { keyOutput.innerHTML = `You pressed: ${event.key}`;

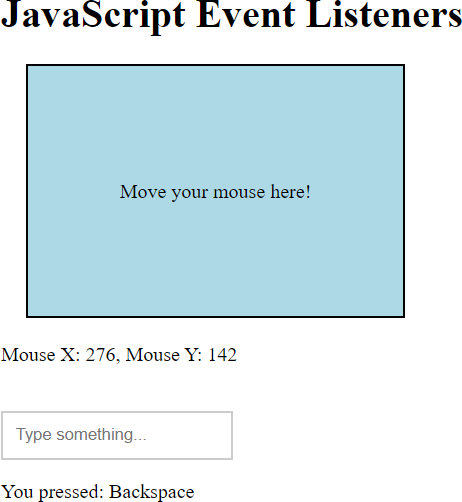
});

</script>

</body>

</html>

* **Output:**



* 1. **Perform Regular Expressions and validation using JavaScript.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial- scale=1.0">

<title>Form Validation with Regular Expressions</title>

</head>

<body>

<h1>Form Validation</h1>

<form id="registration-form" onsubmit="return validateForm()">

<label for="username">Username (alphanumeric, 5-12

characters):</label><br>

<input type="text" id="username"><br><br>

<label for="email">Email (example@domain.com):</label><br>

<input type="email" id="email"><br><br>

<label for="password">Password (at least 6 characters):</label><br>

<input type="password" id="password"><br><br>

<input type="submit" value="Register">

</form>

<p id="error-message" style="color: red;"></p>

<script>

function validateForm() {

let username = document.getElementById('username').value; let email = document.getElementById('email').value;

let password = document.getElementById('password').value; let errorMessage = document.getElementById('error-message'); errorMessage.innerHTML = "";

// Regular Expressions for validation

let usernameRegex = /^[a-zA-Z0-9]{5,12}$/;

let emailRegex = /^[a-zA-Z0-9.\_-]+@[a-zA-Z0-9.-]+\.[a-zA- Z]{2,}$/;

let passwordRegex = /^.{6,}$/;

// Username validation

if (!usernameRegex.test(username)) { errorMessage.innerHTML += "Invalid username.<br>"; return false;

}

// Email validation

if (!emailRegex.test(email)) {

errorMessage.innerHTML += "Invalid email format.<br>"; return false;

}

// Password validation

if (!passwordRegex.test(password)) {

errorMessage.innerHTML += "Password must be at least 6 characters long.<br>";

return false;

}

alert("Form submitted successfully!"); return true;

}

</script>

## Output:

</body>

</html>





1. **PHP**
2. **Write a PHP script to upload a file to server.**



**PRACTICAL 7**

* + **HTML:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial- scale=1.0">

<title>File Upload</title>

</head>

<body>

<h1>Upload a File</h1>

<form action="Upload.php" method="post" enctype="multipart/form-data">

<label for="file">Choose a file:</label>

<input type="file" name="file" id="file" required>

<br><br>

<input type="submit" value="Upload">

</form>

</body>

</html>

## PHP:

<?php

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

// Directory where the file will be uploaded

$target\_dir = "LabProgram/Pr7/Upload";

$target\_file = $target\_dir . basename($\_FILES["file"]["name"]);

// Check if the upload directory exists; if not, create it if (!is\_dir($target\_dir)) {

mkdir($target\_dir, 0777, true);

}

// Check if file is a valid upload

if (move\_uploaded\_file($\_FILES["file"]["tmp\_name"],

$target\_file)) {

echo "The file " . htmlspecialchars(basename($\_FILES["file"]["name"])) . " has been uploaded.";

} else {

echo "Sorry, there was an error uploading your file.";

}

}?>



## Output:



1. **Write a PHP script to get the value from one form and display it into other form.**
   * **HTML:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial- scale=1.0">

<title>Form 1</title>

</head>

<body>

<h1>Form 1</h1>

<form action="form2.php" method="post">

<label for="value">Enter a value:</label>

<input type="text" name="value" id="value" required>

<br><br>

<input type="submit" value="Submit">

</form>

</body>

</html>

## PHP:

<?php

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$value = htmlspecialchars($\_POST['value']); echo "<h1>Form 2</h1>";

echo "<form action='form2.php' method='post'>";

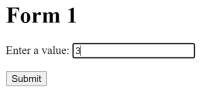
echo "<label for='displayValue'>You entered:</label>";

echo "<input type='text' name='displayValue' id='displayValue' value='$value' readonly>";

echo "<br><br>";

echo "<input type='submit' value='Go Back'>"; echo "</form>";

}

?>

## Output:



1. **Write a PHP script for login authentication. Design an html form which takes username and password from user and validate against stored username and password in file.**
   * **HTML:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Login</title>

</head>

<body>

<h1>Login</h1>

<form action="login.php" method="post">

<label for="username">Username:</label>

<input type="text" name="username" id="username" required>

<br><br>

<label for="password">Password:</label>

<input type="password" name="password" id="password" required>

<br><br>

<input type="submit" value="Login">

</form>

</body>

</html>

## PHP:

<?php

// Replace with your own username and password for validation

$stored\_username = 'admin';

$stored\_password = 'password123';

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$username = htmlspecialchars($\_POST['username']);

$password = htmlspecialchars($\_POST['password']);

if ($username === $stored\_username && $password

=== $stored\_password) {

echo "Login successful! Welcome, $username.";

} else {

echo "Invalid username or password.";

}

}

?>



## Output:





# PRACTICAL 8



1. **Write PHP script for storing and retrieving user information from MySql table.**
   1. **Design an Html page which takes Name, Address, Email and phone from user. (registration.php).**
   2. **Store this data in MySQL database.**
   3. **Next page displays all user in html table using PHP (display.php).**

**HTML:**

<!-- registration.php -->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>User Registration</title>

</head>

<body>

<h2>User Registration</h2>

<form action="store.php" method="POST">

<label for="name">Name:</label>

<input type="text" id="name" name="name" required><br><br>

<label for="address">Address:</label>

<input type="text" id="address" name="address" required><br><br>

<label for="email">Email:</label>

<input type="email" id="email" name="email" required><br><br>

<label for="phone">Phone:</label>

<input type="text" id="phone" name="phone" required><br><br>

<input type="submit" value="Register">

</form>

</body>

</html>

## PHP-1:

<!-- store.php -->

<?php

// Database connection

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "users\_db"; // Create this database in MySQL

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Get form data

$name = $\_POST['name'];

$address = $\_POST['address'];

$email = $\_POST['email'];

$phone = $\_POST['phone'];

// Insert user data into the table

$sql = "INSERT INTO users (name, address, email, phone) VALUES ('$name', '$address', '$email', '$phone')";

if ($conn->query($sql) === TRUE) {

echo "New record created successfully. <a href='display.php'>View Users</a>";

} else {

echo "Error: " . $sql . "<br>" . $conn->error;

}

$conn->close();

?>

## PHP-2:

<!-- display.php -->

<?php

// Database connection

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "users\_db";

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Fetch all users from the database

$sql = "SELECT id, name, address, email, phone FROM users";

$result = $conn->query($sql);

if ($result->num\_rows > 0) {

echo "<h2>Registered Users</h2>";

echo "<table

border='1'><tr><th>ID</th><th>Name</th><th>Address</th><th>Email</th><th>Phone</t h></tr>";

// Output data of each row

while($row = $result->fetch\_assoc()) {

echo "<tr><td>" . $row["id"]. "</td><td>" . $row["name"]. "</td><td>" .

$row["address"]. "</td><td>" . $row["email"]. "</td><td>" . $row["phone"]. "</td></tr>";

}

echo "</table>";

} else {

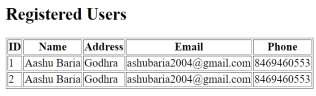
echo "0 results";

}

$conn->close();

?>

**OUTPUT:**



# PRACTICAL 9



9.Write a PHP program to store page views count in SESSION, to increment the count on each refresh, and to show the count on web page.

<?php

// Start the session session\_start();

// Check if the 'views' session variable exists if(isset($\_SESSION['views'])) {

// If it exists, increment the count

$\_SESSION['views'] = $\_SESSION['views'] + 1;

} else {

// If it doesn't exist, initialize it to 1

$\_SESSION['views'] = 1;

}

// Display the current view count

echo "<h2>Page Views Count: ".$\_SESSION['views']."</h2>";

?>

* **Output:**



# PRACTICAL 10



**10 Create REST API using PHP to send the form data like username, email into the database and test that API.**

<?php

// api.php

// Set headers to allow the API to receive JSON data and handle CORS (for testing) header("Access-Control-Allow-Origin: \*");

header("Content-Type: application/json; charset=UTF-8"); header("Access-Control-Allow-Methods: POST"); header("Access-Control-Max-Age: 3600");

header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With");

// Database connection details

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "api\_test\_db";

// Create connection

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Get the posted data from the request body

$data = json\_decode(file\_get\_contents("php://input"));

// Validate data

if(isset($data->username) && isset($data->email)) {

// Sanitize input

$username = htmlspecialchars(strip\_tags($data->username));

$email = htmlspecialchars(strip\_tags($data->email));

// Prepare SQL query

$sql = "INSERT INTO users (username, email) VALUES ('$username', '$email')";

if ($conn->query($sql) === TRUE) {

// Send a success response http\_response\_code(201); // Created

echo json\_encode(array("message" => "User created successfully."));

} else {

// Send an error response http\_response\_code(503); // Service unavailable

echo json\_encode(array("message" => "Unable to create user."));

}

} else {

// Send an error response for incomplete data http\_response\_code(400); // Bad request

echo json\_encode(array("message" => "Incomplete data. Username and email are required."));

}

// Close the database connection

$conn->close();

?>

* **Output:**

