

1. Create an HTML documents to study various HTML tags, style sheets and the tag, Borders, padding, color, and the tag.

```
<html>
  <head>
    <title>My first web page</title>
    <style>
      .test
      {
        color: blue;
      }
    </style>
    <link rel="stylesheet" href= "style.css" />
  </head>
  <body>
    <p>This is a paragraph</p>
    <h1 style="color: blue">Heading 1</h1>
    <h2 class="test">Heading 2</h2>
    <h6>Heading 6</h6>
    <ul>
      <li>This is list item 1</li>
      <li>This is list item 2</li>
    </ul>
    <ol>
      <li>This is list item 1</li>
      <li>This is list item 2</li>
      <li>This is list item 3</li>
    </ol>
    
    <a href="http://www. google.com" target="_blank">Click me</a>
    <br>
    <input type="text" placeholder="Enter username" />
    <br>
    <input type="password" placeholder="Enter password" />
    <br>
    <input type="number" />
    <br>
    <button>Click me</button>
  </body>
</html>
```

style.css:

```
.test
{
  font-size: 60px;
  background-color: yellow;
  padding: 30px; margin: 80px;
  border: 5px solid red;
}
```

2. Develop a JavaScript embedded HTML file for.

a) Generating Sum of n numbers. Use alert window to display the result

b) Determine the roots of Quadratic Equation. Use document. Write to produce output.

2a)

```
<html>
  <head>
    <title>Experimant 2a</title>
  </head>
  <body>
    <h2>Sum of n numbers</h2>
    <input type="number" placeholder="number" class="num" />
    <button class="button">Sum up</button>
    <script>
      const btn = document.querySelector(".button");
      btn.addEventListener("mouseover", function ()
      {
        const enteredNumber = document.querySelector(".num").value;
        let sum = 0;
        for (let i = 1; i <= enteredNumber; i++)
        {
          sum = sum + i;
        }
        alert("sum is :" + sum);
      });
    </script>
  </body>
</html>
```

2b)

```
<html>
<head>
<title> Quadratic Equation </title>
</head>
<body>
  <input type="number" placeholder="co efficient of a" class="a" /> <br>
  <input type="number" placeholder="co efficient of b" class="b" /> <br>
  <input type="number" placeholder="co efficient of c" class="c" /> <br>
  <button class="button">Get roots</button>
  <script>
    const btn = document.querySelector(".button");
    btn.addEventListener("click", function ()
    {
      const a = Number(document.querySelector(".a").value);
      const b = Number(document.querySelector(".b").value);
      const c = Number(document.querySelector(".c").value);
      let real1, real2, imag1, imag2;
      const rootof = b * b - 4 * a * c;    // negative
      if (rootof < 0)
      {
        real1 = -b / (2 * a);
```

```
    real2 = -b / (2 * a);
    imag1 = Math.sqrt(rootof * -1) / (2 * a) + "i";
    imag2 = Math.sqrt(rootof * -1) / (2 * a) + "i";
}
// positive
if (rootof > 0)
{
    real1 = (-b + Math.sqrt(rootof)) / (2 * a);
    real2 = (-b - Math.sqrt(rootof)) / (2 * a);
    imag1 = 0 + "i";
    imag2 = 0 + "i";
}
document.write("Root1 : ", real1, "+", imag1);
document.write("Root2 : ", real2, "-", imag2);
});
</script>
</body>
</html>
```

3. Learn various array and object operations and perform the following operations:

- a) Create an empty array with name 'todoList'
- b) Use 'push' operation on the 'todoList' array to add few objects each having 'id' as key and string as value (for ex {id:"a"},{id:"b"})
- c) Use 'pop' operation to remove the last element from the 'todoList' array.
- d) Use 'filter' operation to return a new array of objects with no object having id as "a"

```
<html>
  <head>
    <title>Push, Pop, Filter Operation</title>
  </head>
  <body>
    <p> Analyze different array and object operations available in JavaScript and
    perform the following operations: <br /> a) Create ... <br /> b) Use 'push' operation .... <br
    /> c) Use 'pop' operation ... <br /> d) Use 'filter' operation .....</p>
    <script>
      const todoList = [];
      todoList.push({ id: "a" });
      todoList.push({ id: "b" });
      todoList.push({ id: "c" });
      todoList.push({ id: "d" });
      todoList.push({ id: "e" });
      console.log(todoList);
      const poppedItem = todoList.pop();
      console.log(poppedItem);
      const filteredTodoList = todoList.filter(function (item)
      {
        return item.id !== "a";    });
      console.log(filteredTodoList);
    </script>
  </body> </html>
```

4. Create a modal window using absolute positioning in CSS and use JavaScript for opening and closing the modal.

```
<html>
  <head>
    <title>Create Modal Window</title>
    <style>
      .container
      {
        height: 100vh;
        width: 100vw;
        background-color: red;
        position: relative;
      }
      .modal
      {
        height: 250px;
        width: 300px;
        background-color: gold;
        position: absolute;
        top: 30%;
        left: 30%;
      }
      .hide
      {
        display: none;
      }
    </style>
  </head>
  <body>
    <div class="container">
      <button class="open">open</button>
      <div class="modal hide">
        <h1>Modal</h1>
        <button class="close">close</button>
      </div>
    </div>
    <script>
      const openBtn = document.querySelector(".open");
      const closeBtn = document.querySelector(".close");
      const modal = document.querySelector(".modal");
      openBtn.addEventListener("click", function ()
      {
        modal.classList.remove("hide");
      });
      closeBtn.addEventListener("click", function ()
      {
        modal.classList.add("hide");
      });
    </script>
  </body>
</html>
```

5. Learn basic flex commands and design a price card using flexbox for positioning of elements.

```
<html>
  <head>
    <title>Flex Box </title>
    <style>
      body
      {
        background-color: aqua;
        display: flex;
        flex-direction: row;
        align-items: center;
        justify-content: center;
      }
      .card {
        background-color: cornsilk;
        height: 425px;
        width: 350px;
        border: 4px solid black;
      }
      .order {
        display: flex;
        align-items: center;
        justify-content: space-around;
      }
    </style>
  </head>
  <body>
    <!-- <h3>
      The FlexBox was .....
    </h3> -->
    <div class="card">
      
      <h2>Burger-King</h2>
      <p>
        Pizza burgers ...
      </p>
      <div class="order">
        <h3>$2.45</h3>
        <button class="button">Order Now</button>
      </div>
    </div>
    <script>
      const btn = document.querySelector(".button");
      const box = document.querySelector(".order");
      btn.addEventListener("click", function ()
      {
        box.innerHTML = "";
        box.insertAdjacentHTML("afterbegin", "<h2>Ordered</h2>");
      });
    </script>
  </body>
</html>
```

6. Design a website which dynamically adds and removes contents (To-Do list) using flexbox.

```
<html lang="en">
<head>
  <style>
    #candidate {
      border-radius: 20%;
      border-color: aquamarine;
      box-sizing: border-box;
    }
    .buttonClass {
      border-radius: 20%;
      border-color: aqua;
      border-style: inherit;
    }
    button:hover {
      background-color: green;
    }
  </style>
</head>
<body>
  <ul id="list"></ul>
  <input type="text" id="candidate" />
  <button onclick="addItem()" class="buttonClass">Add item</button>
  <button onclick="removeItem()" class="buttonClass"> Remove item</button>
  <script>
    function addItem() {
      var a = document.getElementById("list");
      var candidate = document.getElementById("candidate");
      var li = document.createElement("li");
      li.setAttribute('id', candidate.value);
      li.appendChild(document.createTextNode(candidate.value));
      a.appendChild(li);
    }
    function removeItem() {
      var a = document.getElementById("list");
      var candidate = document.getElementById("candidate");
      var item = document.getElementById(candidate.value);
      a.removeChild(item);
    }
  </script>
</body>
</html>
```

7. Analyze the working of CSS grid layout and create a website using grid layout.

```
<html>
  <head>
    <title>Grid </title>
    <style>
      .item {
        background-color:plum;
      }
      .container {
        display: grid;
        background-color: blue;
        grid-template-columns: 1fr 1fr 1fr 200px;
        grid-template-rows: 50px 100px 200px 50px;
        grid-template-areas:
          "header header header header"
          "smallBox1 smallBox2 smallBox3 sidebar"
          "mainContent mainContent mainContent sidebar"
          "footer footer footer footer";
        gap: 16px;
      }
      .header {
        grid-area: header;
        padding: 20px;
        font-size: 30px;
        text-align: center
      }
      .smallBox1 {
        gap:20px;
        grid-area: smallBox1;
        padding: 30px;
        font-size: 30px;
        text-align: center
      }
      .smallBox2 {
        grid-area: smallBox2;
        padding: 40px;
        font-size: 30px;
        text-align: Top
      }
      .smallBox3 {
        grid-area: smallBox3;
        padding: 50px;
        font-size: 30px;
        text-align: right
      }
      .sidebar {
        grid-area: sidebar;
        padding: 20px;
        font-size: 30px;
        text-align: center
      }
```



```
    }
    .mainContent {
        grid-area: mainContent;
        padding: 20px;
        font-size: 30px;
        text-align: center
    }
    .footer {
        grid-area: footer;
        padding: 20px;
        font-size: 30px;
        text-align: center
    }
}
</style>
</head>
<body>
    <div class="container">
        <div class="header item"> Header</div>
        <div class="smallBox1 item"> Small box 1</div>
        <div class="smallBox2 item"> Small box 2</div>
        <div class="smallBox3 item"> Small box 3</div>
        <div class="sidebar item"> Sidebar</div>
        <div class="mainContent item"> Main content</div>
        <div class="footer item"> Footer</div>
    </div>
</body>
</html>
```

8. Develop a weather website using REST API in JavaScript and use CSS Grid for positioning.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Weather Forecast</title>
<style>
body {
    background-color: slategray; /* corrected syntax */
    display: flex;
    justify-content: center;
    align-items: center;
    color: rgb(27, 214, 83);
}
.container {
    border: 2px solid black;
    padding: 20px;
    text-align: center;
}
</style>
</head>
<body>
    <div class="container">
        <h1>Weather Forecast</h1>
        <div>
            <input type="text" placeholder="Enter the City name"
oninput="getWeatherInfo(this.value)">
        </div>
        <div class="details">Enter the city name to get the weather Forecast </div>
        </div>
    <script>
        const detailsDiv = document.querySelector('.details'); // corrected selector
        const APIkey = '3becb1759da057bd3b63a18a7d85d605';
        async function getWeatherInfo(cityName) {
            const url =
'https://api.openweathermap.org/data/2.5/weather?q=${cityName}&appid=${APIkey}`; // corrected
URL string
            const info = await fetch(url)
                .then(response => {
                    if(response.ok) {
                        return response.json();
                    }
                    return null;
                }); // corrected promise syntax
            let str = "";
            if (info == null) {
                str = 'City not found';
            } else {
```

```
        str = `Name: ${cityName}<br>
        Temp: ${info.main.temp} <br>
        Min Temp: ${info.main.temp_min}<br>
        Max Temp: ${info.main.temp_max}<br>
        Pressure: ${info.main.pressure}<br>
        Humidity: ${info.main.humidity}`;
    }
    detailsDiv.innerHTML = str;
    }
</script>
</body>
</html>
```

9. Write a PHP program to store current data-time in a COOKIE and display the Last visited on "date-time on the web page upon reopening the same page.

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Last Visit</title>
    <style>
        body {
            background-color: #87ceeb;
            text-align: center;
            font-family: Arial, sans-serif;
        }
    </style>
</head>
<body>
    <h2>Last visited time on the web page</h2>
    <br>
    <?php
        $inTwoMonths = time() + 60 * 60 * 24 * 60;
        setcookie('lastVisit', date("G:i - m/d/y"), $inTwoMonths);

        if (isset($_COOKIE['lastVisit'])) {
            $visit = $_COOKIE['lastVisit'];
            echo "Your last visit was - " . htmlspecialchars($visit);
        } else {
            echo "You've got some stale cookies!";
        }
    ?>
</body>
</html>
```

10. Run SQL queries to do the following: create a database, create table, insert rows in a table, fetch rows from a table, delete a row, and update a row.

```
<html lang="en">
  <head>
    <title>Library </title>
  </head>
  <body>
    <!-- <center> -->
    <h1>Storing Form data in Database</h1>
    <form action="insert.php" method="post">
      <p>
        <label for="anum">Accession Number:</label>
        <input type="number" name="anum" id="anum">
      </p>
      <p>
        <label for="title">Title:</label>
        <input type="text" name="title" id="title">
      </p>
      <p>
        <label for="author">Author:</label>
        <input type="text" name="author" id="author">
      </p>
      <p>
        <label for="edition">Edition:</label>
        <input type="number" name="edition" id="edition">
      </p>
      <p>
        <label for="publisher">Publisher:</label>
        <input type="text" name="publisher" id="publisher">
      </p>
      <input type="submit" value="Submit">
    </form>
    <h1>Get Book Name</h1>
    <form action="display.php" method="post">
      <label for="accessionnumber: ">Enter Accession Number :</label>
      <input type="number" name="accessionnumber" id="accessionnumber">
      <input type="submit" value="Submit">
    </form>
    <!-- </center> -->
  </body>
</html>
```

Insert.php:

```
<!DOCTYPE html>
<head>
  <title>Insert Page page</title>
</head>
<body>
  <?php
```

```

$conn = mysqli_connect("localhost", "root", "", "lab13");
if($conn){
    echo "";
}
else{
    die("ERROR: Could not connect. ". mysqli_connect_error());
}
$num = $_POST['anum'];
$title = $_REQUEST['title'];
$author = $_REQUEST['author'];
$edition = $_REQUEST['edition'];
$publisher=$_REQUEST['publisher'];
$sql = "insert into books values('$num','$title','$author','$edition','$publisher')";
if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
mysqli_close($conn);
?>
</body>
</html>

```

Display.php:

```

<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "lab13";
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$accessionnumber = $_REQUEST['accessionnumber'];
$sql = "SELECT anum,title,author,edition,publisher FROM books where anum = '$accessionnumber'";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        echo "Accession no : " . $row["anum"]. "<br>Title : " . $row["title"]. "<br>Author : " . $row["author"].
"<br>Edition : ".$row['edition']. "<br>Publisher : ".$row['publisher'];
    }
} else {
    echo "0 results";
}
$conn->close();
?>

```

11. On any HTML page, include a link for Login. Write a login page having login/password fields. Write JavaScript code to validate the login-id and password for the following: both are properly formed and at least 6 bytes long; the password contains at least one special case, one capital and one numeric character; convert the password into its MD5 hash use table created in experiment

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
input {
    width: 100%;
    padding: 12px;
    border: 1px solid #ccc;
    border-radius: 4px;
    box-sizing: border-box;
    margin-top: 6px;
    margin-bottom: 16px;
}

input[type=submit] {
    background-color: #04AA6D;
    color: white;
    border: none;
    cursor: pointer;
}

input[type=submit]:hover {
    background-color: #45a049;
}

.container {
    background-color: #f1f1f1;
    padding: 20px;
}

#message {
    display: none;
    background: #f1f1f1;
    color: #000;
    position: relative;
    padding: 20px;
    margin-top: 10px;
}

#message p {
    padding: 10px 35px;
    font-size: 18px;
}

.valid {
```



```
var capital = document.getElementById("capital");
var number = document.getElementById("number");
var length = document.getElementById("length");

myInput.onfocus = function() {
    document.getElementById("message").style.display = "block";
}

myInput.onblur = function() {
    document.getElementById("message").style.display = "none";
}

myInput.onkeyup = function() {
    var lowerCaseLetters = /[a-z]/g;
    if(myInput.value.match(lowerCaseLetters)) {
        letter.classList.remove("invalid");
        letter.classList.add("valid");
    } else {
        letter.classList.remove("valid");
        letter.classList.add("invalid");
    }
}

var upperCaseLetters = /[A-Z]/g;
if(myInput.value.match(upperCaseLetters)) {
    capital.classList.remove("invalid");
    capital.classList.add("valid");
} else {
    capital.classList.remove("valid");
    capital.classList.add("invalid");
}

var numbers = /[0-9]/g;
if(myInput.value.match(numbers)) {
    number.classList.remove("invalid");
    number.classList.add("valid");
} else {
    number.classList.remove("valid");
    number.classList.add("invalid");
}

if(myInput.value.length >= 8) {
    length.classList.remove("invalid");
    length.classList.add("valid");
} else {
    length.classList.remove("valid");
    length.classList.add("invalid");
}
}
</script>

<?php
```

```
if (isset($_POST['submit'])) {  
    $password = $_POST['psw'];  
    echo "Entered Password: " . htmlspecialchars($password) . "<br>";  
    $pass = md5($password);  
    echo "MD5 Hashed Password: " . $pass;  
}  
?>  
  
</body>  
</html>
```