ADL - Lab Test

1) Design HTML form for student registration and validate it using Java script.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Student Registration Form</title>
  </head>
  <body>
    <h1>Student Registration Form</h1>
    <form id="registrationForm" onsubmit="validateForm()">
      <label for="name">Name:</label>
      <input type="text" id="name" name="name" required><br><br>

      <label for="email">Email:</label>
      <input type="email" id="email" name="email" required><br><br>

      <label for="password">Password:</label>
      <input type="password" id="password" name="password"
required><br><br>

      <label for="confirmPassword">Confirm Password:</label>
      <input type="password" id="confirmPassword"
name="confirmPassword" required><br><br>

      <label for="phone">Phone:</label>
      <input type="tel" id="phone" name="phone" required><br><br>

      <label for="gender">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="birthdate">Date of Birth:</label>
      <input type="date" id="birthdate" name="birthdate" required><br><br>

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

    <script>
      function validateForm() {
        var name = document.getElementById("name").value;
        var email = document.getElementById("email").value;
        var password = document.getElementById("password").value;
        var confirmPassword =
document.getElementById("confirmPassword").value;
        var phone = document.getElementById("phone").value;
        var gender =
document.querySelector('input[name="gender"]:checked');
        var birthdate = document.getElementById("birthdate").value;

        // Name Validation
        if (name == "") {
          alert("Name must be filled out");
          return false;
        }

        // Email Validation
        if (email == "") {
          alert("Email must be filled out");
          return false;
        }
        // Password Validation
        if (password == "") {
          alert("Password must be filled out");
          return false;
        }
        if (password != confirmPassword) {
          alert("Passwords do not match");
          return false;
        }

        // Phone Validation
        if (phone == "") {
          alert("Phone number must be filled out");
          return false;
        }
        if (phone.length != 10) {
          alert("Invalid phone number");
          return false;
        }

        // Gender Validation
        if (!gender) {
          alert("Gender must be selected");
          return false;
        }

        // Birthdate Validation
        var currentDate = new Date();
        var userDate = new Date(birthdate);
        var age = (currentDate.getTime() - userDate.getTime()) / (1000 * 60 *
    60 * 24 * 365);
        if (age < 18) {
          alert("You must be at least 18 years old to register");
          return false;
        }
```

```
      }
    </script>
  </body>
</html
```

2) Write a JavaScript function that generates all combinations of a string. Example string: 'dog' Expected Output: d,o,do,g,dg,og,dog

```
function generateCombinations(str) {
  let result = [];
  for (let i = 0; i < str.length; i++) {
    for (let j = i + 1; j <= str.length; j++) {
      result.push(str.slice(i, j));
    }
  }
  return result.join(',');
}

// Example usage:
console.log(generateCombinations('dog')); // d,o,do,g,dg,og,dog
```

3) Write a JavaScript function that returns a passed string with letters in
alphabetical order.
Example string : 'webmaster'
Expected Output : 'abeemrstw'
Assume punctuation and numbers symbols are not included in the passed string.

```
function sortAlphabetically(str) {
  // Convert string to array, sort the array alphabetically, and join back to a
string
  return str.split('').sort().join('');
}

// Example usage
var str = 'webmaster';
var sortedStr = sortAlphabetically(str);
console.log(sortedStr); // Output: abeemrstw
```

4) Write an HTML page that has one input, which can take multi-line text and a submit button. Once the user clicks the submit button, it should show the number of characters, lines and words in the text entered using an alert message. Words are separated with white space and lines are separated with new line character.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Multi-line Text Analysis</title>
  </head>
  <body>
    <h1>Multi-line Text Analysis</h1>
    <form>
      <label for="text-input">Enter text:</label><br>
      <textarea id="text-input" name="text-input" rows="4"
cols="50"></textarea><br>
      <button type="submit">Analyze Text</button>
    </form>

    <script>
      const form = document.querySelector('form');
      const textInput = document.getElementById('text-input');

      form.addEventListener('submit', (event) => {
        event.preventDefault();
        const text = textInput.value;
        const numChars = text.length;
        const numLines = text.split(/\r?\n/).length;
        const numWords = text.trim().split(/\s+/).length;

        const message = `Number of characters: ${numChars}\nNumber of lines: ${numLines}\nNumber of words: ${numWords}`;
        alert(message);
      });
    </script>
  </body>
</html>
```

5) Write a JavaScript function which accepts an argument and returns the type. Note : There are six possible values that typeof returns: object, boolean, function, number, string, and undefined.

```
function getType(val) {
 return typeof val;
}

// Example usage
console.log(getType("Hello World")); // Output: string
console.log(getType(42)); // Output: number
console.log(getType(true)); // Output: boolean
console.log(getType({})); // Output: object
console.log(getType(function() {})); // Output: function
console.log(getType(undefined)); // Output: undefined
```

6) Using HTML, CSS create a page that displays a text on top of an image using an overlay.

```
<!DOCTYPE html>
<html>
```

```
<head>
  <title>Image Overlay Example</title>
  <style>
    /* Style for the overlay */
    .overlay {
      position: absolute;
      top: 0;
      bottom: 0;
      left: 0;
      right: 0;
      background-color: rgba(0, 0, 0, 0.5); /* Semi-transparent black */
      display: flex;
      align-items: center;
      justify-content: center;
      color: white;
      font-size: 3rem;
      text-align: center;
    }

    /* Style for the image */
    .image {
      width: 100%;
      height: auto;
      display: block;
    }

    /* Style for the container */
    .container {
      position: relative;
    }
  </style>
</head>
<body>
  <div class="container">
    <img class="image" src="image.jpg" alt="Your Image">
    <div class="overlay">
      <p>Hello World</p>   // any text
    </div>
  </div>
</body>
</html>
```

7) Using HTML, CSS creates a page applying a perspective transform with a hover animation to an element.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Perspective Transform with Hover Animation</title>
  <style>
    .box {
      width: 200px;
      height: 200px;
      background-color: #4CAF50;
      perspective: 1000px;
      transition: transform 0.5s;
    }

    .box:hover {
      transform: rotateY(30deg);
    }

    .content {
      transform: translateZ(50px);
      text-align: center;
      color: white;
      font-size: 24px;
    }
  </style>
```

```
</head>
<body>
  <div class="box">
    <div class="content">
      Hover Me!
    </div>
  </div>
</body>
</html>
```

8) Using HTML, CSS, JavaScript create a page that displays a typewriter effect animation.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Typewriter Effect Animation</title>
  <style>
    /* Hide the text initially */
    .typewriter {
      overflow: hidden;
      font-size: 24px;
      font-family: monospace;
      white-space: nowrap;
      border-right: 3px solid black;
      animation: typing 1s steps(30, end) forwards;
    }

    /* Typing animation */
    @keyframes typing {
      from {
        width: 0;
      }
      to {
        width: 100%;
      }
    }
  </style>
</head>
<body>
  <div class="typewriter" id="typewriter">Hello, World!</div>

  <script>
    // Get the element that displays the text
    const typewriter = document.getElementById('typewriter');
    // Get the text to be displayed
    const text = typewriter.innerHTML;
    // Clear the text from the element
    typewriter.innerHTML = '';

    // Set up a loop to animate the text
    let i = 0;
    function type() {
      if (i < text.length) {
        typewriter.innerHTML += text.charAt(i);
        i++;
        setTimeout(type, 50);
      }
    }

    // Start the animation
    type();
  </script>
</body>
</html>
```

9) Using HTML, CSS create a page that animates with underline effect when the
user hovers over the text.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Underline Effect on Hover</title>
    <style>
      /* style for the link */
      a {
        text-decoration: none;
        position: relative;
        color: #000;
      }

      /* style for the underline */
      a::before {
        content: "";
        position: absolute;
        width: 100%;
        height: 2px;
        bottom: 0;
        left: 0;
        background-color: #000;
        visibility: hidden;
        transform: scaleX(0);
        transition: all 0.3s ease-in-out 0s;
      }

      /* hover style for the underline */
      a:hover::before {
        visibility: visible;
        transform: scaleX(1);
      }
    </style>
  </head>
  <body>
    <h1>
      <a href="#">Hover over me!</a>
    </h1>
  </body>
</html>
```

10) Write a JavaScript conditional statement to find the sign of product of three numbers. Display an alert box with the specified sign.
Sample numbers : 3, -7, 2
Output : The sign is –

```
let num1 = 3;
let num2 = -7;
let num3 = 2;

let product = num1 * num2 * num3;

if (product > 0) {
  alert("The sign is +");
} else if (product < 0) {
  alert("The sign is -");
} else {
  alert("The product is 0");
}
```

11) Write a JavaScript function to get the last day of a month.
Test Data :
document.write(lastday(2014,0));
document.write(lastday(2014,1));

```
document.write(lastday(2014,11));
```

Output :

```
31
28
31
```

```
function lastday(year, month) {
  return new Date(year, month + 1, 0).getDate();
}
```

In this example, we use the Date object to create a new date for the first day of the next month (month + 1) in the specified year. We then subtract one day from this date by setting the day of the month to 0 using the getDate() method, which returns the last day of the specified month.
To test this function, you can call it with different year and month values:

```
document.write(lastday(2014, 0)); // output: 31
document.write(lastday(2014, 1)); // output: 28
document.write(lastday(2014, 11)); // output: 31
```

12.Create a simple calculator application using servlet.

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/calculator")
public class CalculatorServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        doPost(request, response);
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        // Set the content type of the response
        response.setContentType("text/html");

        // Get the user inputs for calculation
        int num1 = Integer.parseInt(request.getParameter("num1"));
        int num2 = Integer.parseInt(request.getParameter("num2"));
        String op = request.getParameter("op");

        // Calculate the result based on the operator
        int result = 0;
        switch (op) {
        case "+":
            result = num1 + num2;
            break;
        case "-":
            result = num1 - num2;
            break;
        case "*":
            result = num1 * num2;
            break;
        case "/":
            result = num1 / num2;
            break;
        }

        // Generate the HTML response with the result
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Calculator Servlet</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h2>Calculator Servlet</h2>");
        out.println("<p>Result: " + result + "</p>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

13) Create a servlet for a login page. If the username and password are correct then
it says message “Hello ” else a message “login failed”.

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class LoginServlet extends HttpServlet {
  private String username = "myusername";
  private String password = "mypassword";

  public void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    // Get the username and password from the request parameters
    String enteredUsername = request.getParameter("username");
    String enteredPassword = request.getParameter("password");

    // Check if the username and password are correct
    if (enteredUsername.equals(username) &&
enteredPassword.equals(password)) {
      // Login successful
      response.setContentType("text/html");
      PrintWriter out = response.getWriter();
      out.println("<html><body>");
      out.println("<h2>Hello " + enteredUsername + "</h2>");
      out.println("</body></html>");
    } else {
      // Login failed
      response.setContentType("text/html");
      PrintWriter out = response.getWriter();
      out.println("<html><body>");
      out.println("<h2>Login failed</h2>");
      out.println("</body></html>");
    }
  }
}
```

14) Create a servlet that uses Cookies to store the number of times a user has visited servlet.

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class VisitCounterServlet extends HttpServlet {
  private int visitCount;

  public void init() {
    // Initialize the visit count to 0
    visitCount = 0;
  }

  public void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    // Get the current visit count from the cookie
    Cookie[] cookies = request.getCookies();
    if (cookies != null) {
     for (Cookie cookie : cookies) {
      if (cookie.getName().equals("visitCount")) {
       visitCount = Integer.parseInt(cookie.getValue());
      }
     }
    }

    // Increment the visit count
    visitCount++;

    // Store the updated visit count in a new cookie
    Cookie visitCountCookie = new Cookie("visitCount",
Integer.toString(visitCount));
    visitCountCookie.setMaxAge(60 * 60 * 24 * 365); // Cookie lasts for one
year
    response.addCookie(visitCountCookie);

    // Send the response with the updated visit count
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html><body>");
    out.println("<h2>You have visited this servlet " + visitCount + "
times</h2>");
    out.println("</body></html>");
  }
}
```