Web Technologies Lab File

LAB FILE SUBMITTED BY:

NAME: Sairanjan Subudhi
SAP ID: 500101861
ROLL NO: R2142220816
BATCH: 2 (DEVOPS)

SUBMITTED TO:
Mr. Chandramani Sharma

Experiment 1: Basic HTML Tags

Q1. Design a webpage to describe your university. The webpage should have properly aligned paragraphs to show textual information and images wherever required.

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>UPES Dehradun</title>
</head>

<body style = "background-color: lightskyblue;">

    <h1 style="background-color:DodgerBlue;text-align: center;">Welcome to
UPES Dehradun</h1>

    <img src="https://blog.upes.ac.in/wp-content/uploads/2021/06/canva-photo-
editor.png" height="700" width="700" alt="UPES Campus" style="max-width: 100%;
height: auto; display: block; margin: 30px auto;">

    <h2 style="background-color:DodgerBlue;text-align: center;">About
UPES</h2>

    <p>The University of Petroleum and Energy Studies (UPES) is a
distinguished educational institution nestled in the serene city of Dehradun,
India. Established with a focused mission on providing specialized education
in petroleum, energy, and associated fields, UPES has consistently led the way
in fostering innovation and excellence. With a commitment to meeting industry
demands, the university stands as a beacon for students aspiring to excel in
the dynamic sectors of petroleum and energy. Through its dedicated approach
and strategic location, UPES continues to play a pivotal role in shaping the
future of professionals in these crucial domains.</p>

    <img src="https://static.toiimg.com/photo/msid-102712710/102712710.jpg"
height="700" width="700" alt="UPES Campus" style="max-width: 100%; height:
auto; display: block; margin: 30px auto;">

    <h2 style="background-color:DodgerBlue;text-align: center;">Our
Campus</h2>

    <p>Nestled in Dehradun, the UPES campus is a sprawling hub of learning,
adorned with state-of-the-art facilities and an enriching atmosphere for
```

```
academic endeavors and extracurricular engagement. This expansive campus harmoniously integrates modern infrastructure with the innate splendor of nature, creating an ideal setting for a holistic educational journey. Surrounded by scenic beauty, UPES offers a conducive environment that inspires learning and personal growth. The blend of contemporary amenities and natural aesthetics provides students with a well-rounded and fulfilling educational experience, fostering a balance between academic excellence and appreciation for the serenity that surrounds this educational haven.<p>

    <img src="https://openthemagazine.com/wp-content/uploads/2022/07/UPES.jpg" height="700" width="700" alt="UPES Campus" style="max-width: 100%; height: auto; display: block; margin: 30px auto;">



    <h2 style="background-color:DodgerBlue;text-align: center;">Academic Programs</h2>

    <p>UPES offers a comprehensive range of academic programs across various disciplines, encompassing engineering, management, design, law, and more. Our meticulously crafted curriculum is designed to seamlessly align with industry standards, ensuring that students are thoroughly prepared to meet the dynamic demands of the professional landscape. Emphasizing practical relevance, our programs provide a holistic learning experience that goes beyond theoretical knowledge. Through a forward-thinking approach, UPES aims to empower students with the skills and insights needed to excel in their chosen fields, fostering a generation of professionals who are adept at navigating the complexities of the contemporary job market.</p>

    <img src="https://images.shiksha.com/mediadata/images/1510907249phpvlYi2T.png" height="700" width="700" alt="Lecture Hall" style="max-width: 100%; height: auto; display: block; margin: 30px auto;">

    <h2 style="background-color:DodgerBlue;text-align: center;">Admissions</h2>

    <p>UPES embraces a merit-based admission process, extending a warm invitation to students from diverse backgrounds who share a passion for creating a meaningful impact in the energy and related sectors. We invite prospective candidates to explore our array of programs and familiarize themselves with the admission procedure to embark on their transformative journey at UPES. Whether aspiring engineers, managers, designers, or legal professionals, we encourage individuals with a commitment to excellence and a desire to contribute to the evolving landscape of energy to discover the opportunities awaiting them at our institution. Join UPES to turn your aspirations into a reality and shape a promising future.</p>
```

```
    <h2 style="background-color:DodgerBlue;text-align: center;">Library and
Resources</h2>

    <p>Our well-equipped library stands as a vast reservoir of knowledge,
providing students with unparalleled access to a rich collection of books,
journals, and online resources. Committed to empowering our students, we have
curated an extensive repository that serves as a cornerstone for academic
excellence. This abundant stock of information caters to diverse learning
needs, fostering an environment where students can delve deep into their
subjects, conduct research, and broaden their intellectual horizons. With a
commitment to cultivating a culture of learning, our library stands as a
dedicated space where students can fuel their curiosity and lay the foundation
for scholarly success in their academic pursuits.</p>

    <img
src="https://upload.wikimedia.org/wikipedia/commons/6/66/Upes_lib1.jpg"
height="700" width="700" alt="Library" style="max-width: 100%; height: auto;
display: block; margin: 30px auto;">

    <h2 style="background-color:DodgerBlue;text-align: center;">Contact
Us</h2>

    <p>Should you have any inquiries or wish to visit our campus, please don't
hesitate to contact us. We are here to support you throughout your educational
journey at UPES.</p>

    <p style="text-align: center;">Contact Information: <a
href="tel:+91XXXXXXXXXX">+91 XXXXXXXXXX</a> | <a
href="mailto:info@upes.ac.in">info@upes.ac.in</a></p>

</body>

</html>
```

Q2. Design a webpage to show your timetable in proper format.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Timetable</title>
</head>
<body>
    <h1>Devops B-2 Timetable</h1>

    <table border="5"; align="center";>
        <thead>
            <tr>
                <th>Time</th>
                <th>Monday</th>
                <th>Tuesday</th>
                <th>Wednesday</th>
                <th>Thursday</th>
                <th>Friday</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>9:00 AM - 10:00 AM</td>
                <td style = "text-align: center">Devops automation lab</td>
                <td style = "text-align: center">Web tech lab</td>
                <td style = "text-align: center">ADBMS Lab</td>
                <td style = "text-align: center">-</td>
                <td style = "text-align: center">SEPM lab</td>

            </tr>
            <tr>
                <td>10:00 AM - 11:00 AM</td>
```

```
            <td style = "text-align: center">Devops automation lab</td>
            <td style = "text-align: center">Web tech lab</td>
            <td style = "text-align: center">ADBMS Lab</td>
            <td style = "text-align: center">-</td>
            <td style = "text-align: center">SEPM lab</td>

        </tr>
        <tr>
            <td>11:00 AM - 12:00 PM</td>
            <td style = "text-align: center">FLAT</td>
            <td style = "text-align: center">-</td>
            <td style = "text-align: center">-</td>
            <td style = "text-align: center">-</td>
            <td style = "text-align: center">Env. & SDG</td>

        </tr>
        <tr>
            <td>12:00 PM - 1:00 PM</td>
            <td style = "text-align: center">-</td>
            <td style = "text-align: center">SEPM</td>
            <td style = "text-align: center">-</td>
            <td style = "text-align: center">-</td>
            <td style = "text-align: center">SEPM</td>

        </tr>
        <tr>
            <td>1:00 PM - 2:00 PM</td>
            <td style = "text-align: center">ADBMS</td>
            <td style = "text-align: center">-</td>
            <td style = "text-align: center">-</td>
            <td style = "text-align: center">EDGE</td>
            <td style = "text-align: center">-</td>

        </tr>
        <tr>
            <td>2:00 PM - 3:00 PM</td>
            <td style = "text-align: center">Devops Automaion</td>
            <td style = "text-align: center">Web Tech</td>
            <td style = "text-align: center">FLAT</td>
            <td style = "text-align: center">-</td>
            <td style = "text-align: center">FLAT</td>

        </tr>
        <tr>
            <td>3:00 PM - 4:00 PM</td>
            <td style = "text-align: center">-</td>
            <td style = "text-align: center">Design Thinking Lab</td>
            <td style = "text-align: center">ADBMS</td>
```

```
                <td style = "text-align: center">Web Tech</td>
                <td style = "text-align: center">Devops Automation</td>

            </tr>
            <tr>
                <td>4:00 PM - 5:00 PM</td>
                <td style = "text-align: center">-</td>
                <td style = "text-align: center">Design Thinking Lab</td>
                <td style = "text-align: center">-</td>
                <td style = "text-align: center">-</td>
                <td style = "text-align: center">-</td>

            </tr>
            <tr>
                <td>5:00 PM - 6:00 PM</td>
                <td style = "text-align: center">-</td>
                <td style = "text-align: center">-</td>
                <td style = "text-align: center">Design Thinking</td>
                <td style = "text-align: center">-</td>
                <td style = "text-align: center">-</td>

            </tr>
            </tbody>
    </table>
</body>
</html>
```

Devops B-2 Timetable

Time	Monday	Tuesday	Wednesday	Thursday	Friday
9:00 AM - 10:00 AM	Devops automation lab	Web tech lab	ADBMS Lab	-	SEPM lab
10:00 AM - 11:00 AM	Devops automation lab	Web tech lab	ADBMS Lab	-	SEPM lab
11:00 AM - 12:00 PM	FLAT	-	-	-	Env. & SDG
12:00 PM - 1:00 PM	-	SEPM	-	-	SEPM
1:00 PM - 2:00 PM	ADBMS	-	-	EDGE	-
2:00 PM - 3:00 PM	Devops Automaion	Web Tech	FLAT	-	FLAT
3:00 PM - 4:00 PM	-	Design Thinking Lab	ADBMS	Web Tech	Devops Automation
4:00 PM - 5:00 PM	-	Design Thinking Lab	-	-	-
5:00 PM - 6:00 PM	-	-	Design Thinking	-	-

Q3. Create a webpage that should have a menu to show different tourist places in Dehradun. Use different fore ground, background colors, images and other properties.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=, initial-scale=1.0">
    <title>Dehradun - Tourist places</title>
</head>
<body style="background-color: rgb(255, 255, 255);">
    <h1 style="background-color:rgb(121, 220, 233);text-align: center;">
Tourism Attractions in Dehradun</h1>
    <h2 style="background-color:rgb(101, 39, 234);">Robbers Cave</h2>
    <h3>Robbers Cave State Park in Oklahoma features rugged terrain, limestone
cliffs, and a scenic woodland setting. The park is known for its unique rock
formations, including the iconic Robbers Cave, rumored to have sheltered
outlaws. Visitors enjoy hiking trails, rappelling, and exploring the historic
hideout, creating a picturesque outdoor experience..</h3>

    <img src="https://dehraduntourism.co.in/images/places-to-
visit/header/robbers-cave-dehradun-tourism-entry-fee-timings-holidays-reviews-
header.jpg" height="500" width="500" alt="Robbers Cave" style="max-width:
100%; height: auto; display: block; margin: 30px auto;">
    <a href="https://en.wikipedia.org/wiki/Robber%27s_Cave,_India">For more
information on Robbers Cave Click Here</a>
    <h2 style="background-color:rgb(101, 39, 234);"> Sahastradhara </h2>
    <h3> Sahastradhara, situated in the Indian state of Uttarakhand, is a
captivating tourist destination renowned for its myriad natural springs and
waterfalls. The name translates to "thousand-fold spring," where sulfur-
infused water flows through numerous streams, creating a therapeutic
environment. Visitors relish the scenic beauty, rejuvenating baths, and
tranquil ambiance..</h3>
    <img src="https://dehraduntourism.co.in/images/places-to-
visit/header/sahastradhara-dehradun-tourism-entry-fee-timings-holidays-
reviews-header.jpg" height="500" width="500" alt="Robbers Cave" style="max-
width: 100%; height: auto; display: block; margin: 30px auto;">
    <a href="https://en.wikipedia.org/wiki/Sahasradhara">For more information
on Sahastradhara Click Here</a>
    <h2 style="background-color:rgb(101, 39, 234);"> Buddha Monastery</h2>
    <h3>The Buddha Monastery, often exemplifying serenity, is a sacred
Buddhist retreat. Adorned with vibrant prayer flags and intricate murals, it
offers a tranquil space for meditation and worship. Nestled in scenic
surroundings, the monastery reflects the essence of Buddhist philosophy,
attracting pilgrims and seekers seeking spiritual solace and
enlightenment.</h3>
    <img src="https://dehraduntourism.co.in/images/places-to-
visit/header/buddha-tibetan-buddhist-temple-dehradun-tourism-entry-fee-
timings-holidays-reviews-header.jpg" height="500" width="500" alt="Robbers
Cave" style="max-width: 100%; height: auto; display: block; margin: 30px
auto;">
```

```
    <a href="https://www.euttaranchal.com/tourism/buddha-temple.php">For more information on Buddha Monaestry Click Here</a>
</body>
</html>
```

Tourism Attractions in Dehradun

Robbers Cave

Robbers Cave State Park in Oklahoma features rugged terrain, limestone cliffs, and a scenic woodland setting. The park is known for its unique rock formations, including the iconic Robbers Cave, rumored to have sheltered outlaws. Visitors enjoy hiking trails, rappelling, and exploring the historic hideout, creating a picturesque outdoor experience..

For more information on Robbers Cave Click Here

Sahastradhara

Sahastradhara, situated in the Indian state of Uttarakhand, is a captivating tourist destination renowned for its myriad natural springs and waterfalls. The name translates to "thousand-fold spring," where sulfur-infused water flows through numerous streams, creating a therapeutic environment. Visitors relish the scenic beauty, rejuvenating baths, and tranquil ambiance..

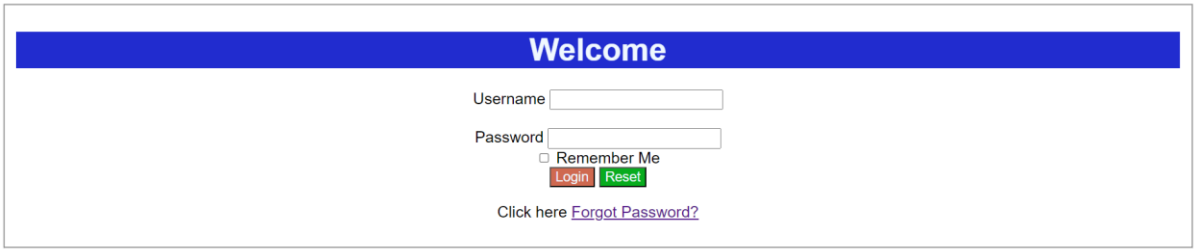
For more information on Sahastradhara Click Here

Experiment 2: Creating HTML forms using different form elements.

1. Create a login page using different form elements.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <form action="https://www.google.com/" >
    <meta charset="UTF-8">
    <meta name="viewport" content="width=, initial-scale=1.0">
    <title>Login Page</title>
</head>
<body style="text-align: center;">
    <fieldset>
    <h1 style="color: aliceblue;text-align: center;background-color: rgb(33,
44, 207);">Welcome</h1>
    <div>
        <label for="fname">Username</label>
    <input type="text" id="fname" name="fname"; required>
    </div>
    <br>

    <div>
    <label for="Password">Password</label>
    <input type="password" id="Password" name="Password"; required>
    </div>

    <div>
        <input type="checkbox" id="Remember" name="Remember" checked; required
/>
        <label for="Remember">Remember Me</label>
      </div>

     <input style="color: aliceblue;text-align: center;background-color:
rgb(207, 105, 80);" type="submit" value="Login">
     <input style="color: aliceblue;text-align: center;background-color:
rgb(8, 172, 30);" type="reset" value="Reset">

     <p style="text-align: center;">Click here <a
href="https://www.google.com/">Forgot Password?</a>


     </form>
</fieldset>
```

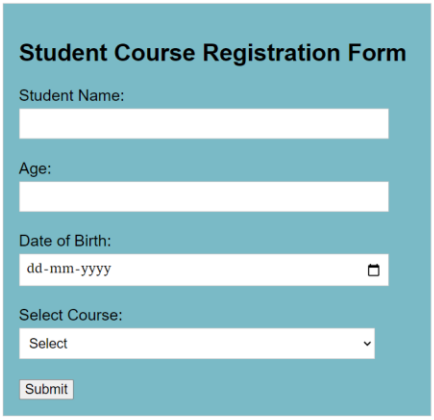
```
</body>
</html>
```



2. Create a HTML form for a student for course registration which should have following fields:
    1. Student Name (textbox)
    2. Age (textbox with numbers only)
    3. Date of Birth (Calendar)
    4. Select Course (Drop Down)
    5. Submit (Button)

```
<!DOCTYPE html>
<html>
<head>
    <title>Student Course Registration</title>
    <style>
        /* Basic styling - adjust as needed */
        form {
            width: 500px;
            margin: 0 auto;
            padding: 20px;
            margin-top: 75px;
            margin-bottom: 75px;
            border: 1px solid #ccc;
            background-color: #7abac6;
        }
        label {
            display: block;
            margin-bottom: 5px;
        }
        input[type="text"], input[type='number'],input[type='date'],select{
            width: 90%;
            padding: 8px;
            margin-bottom: 25px;
            border: 1px solid #ccc;
        }
    </style>
</head>
<body style = "background-color: #ffffff;">
```

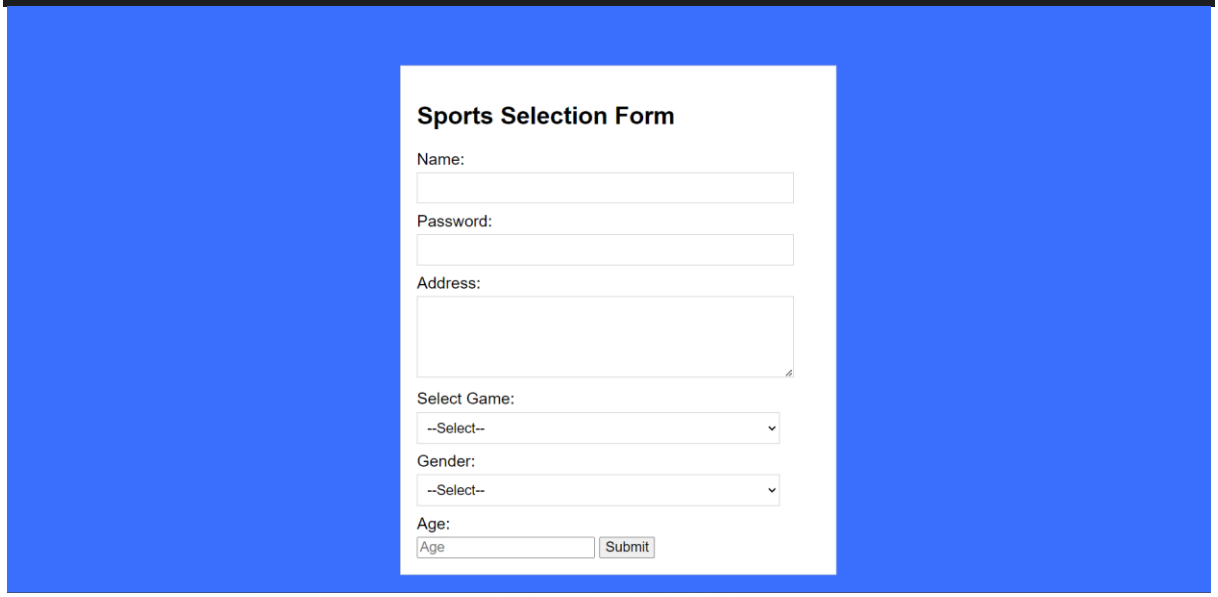
```
    <form>
        <h2>Student Course Registration Form</h2>

        <label for="student_name">Student Name:</label>
        <input type="text" id="student_name" name="student_name" >

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

        <label for="dob">Date of Birth:</label>
        <input type="date" id="dob" name="dob" >

        <label for="course">Select Course:</label>
        <select id="course" name="course" >
        <option value="">Select</option>
        <option value="Web Tech">Web Tech</option>
        <option value="ADBMS">ADBMS</option>
        <option value="FLAT">FLAT</option>
        <option value="Devops automation">Devops automation</option>
        </select>

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

</body>
</html>
```



3. Create HTML form for selecting sports in your university. The form should have fields like Name, Password, Address, Select Game, Gender, Age etc.

```html
<!DOCTYPE html>
<html>
<head>
    <title>University Sports Selection Form</title>
    <style>
        /* Basic styling - adjust as needed */
        form {
            width: 500px;
            margin: 0 auto;
            padding: 20px;
            margin-top: 75px;
            margin-bottom: 75px;
            border: 1px solid #ccc;
            background-color: #ffffff;
        }
        label {
            display: block;
            margin-bottom: 5px;
        }
        input[type="text"], input[type="password"], textarea, select{
            width: 90%;
            padding: 8px;
            margin-bottom: 10px;
            border: 1px solid #ccc;
        }
    </style>
</head>
<body style = "background-color: #3a6efd;">

    <form>
        <h2>Sports Selection Form</h2>

        <label for="name">Name:</label>
        <input type="text" id="name" name="name" required>

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

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

        <label for="game">Select Game:</label>
        <select id="game" name="game">
            <option value="">--Select--</option>
            <option value="Cricket">Cricket</option>
            <option value="football">Football</option>
            <option value="volleyball">Volleyball</option>
            <option value="Badminton">Badminton</option>
```

```
            <option value="Table Tennis">Table Tennis</option>
            <option value="Basketball">Basketball</option>
        </select>

        <label for="gender">Gender:</label>
        <select id="gender" name="gender">
            <option value="">--Select--</option>
            <option value="Male">Male</option>
            <option value="Female">Female</option>
        </select>
        <label for="age">Age:</label>
        <input type="number" id="age" name="age" min="18" required
placeholder="Age">

        <button type="submit">Submit</button>
    </form>

</body>
</html>
```



Experiment 3: Use of Inline, Internal and External stylesheets and incorporating styles in HTML document.

1. Use Inline style sheet and create a webpage.
2. Create a CSS based Sticky footer.

```
footer{
    background-color: aliceblue;
    bottom:0;
    text-align: center;
    position:sticky;
}
.box{
    height: 200px;
    width:200px;
    color:brown;
    background-color: aqua;
    margin:100px;
    animation:zoom 5s 5s infinite;
}
@keyframes zoom{
    from{
        scale:1
    }
    to{
        scale:1.5
    }
}
```

3. Using HTML and CSS, create a custom hover and focus effect for navigation items, using CSS transformations.
4. Using HTML, CSS creates a zoom in zoom out animation.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <style>
      nav ul {
        display: flex;
      }
```

```
      nav ul li {
        list-style: none;
        padding: 5px 10px 5px 10px;
      }
      nav li a {
        color: white;
        text-decoration: none;
        padding: 5px 10px 5px 10px;
      }
      nav ul li a:hover {
        background-color: grey;
      }
    </style>
    <title>Exp-3</title>
    <link rel="stylesheet" href="webTexhlab3.css" />
  </head>
  <body style="background-color: black">
    <nav>
      <ul>
        <li><a href="#" style="color: white">Option 1</a></li>
        <li><a href="#" style="color: white">Option 2</a></li>
        <li><a href="#" style="color: white">Option 3</a></li>
        <li><a href="#" style="color: white">Option 4</a></li>
      </ul>
    </nav>
    <h1 style="color: white; text-align: center">
      Paragraphs & Topic Sentences
    </h1>
    <p style="color: white; font-size: large">
      A paragraph is a series of sentences that are organized and coherent,
and
      are all related to a single topic. Almost every piece of writing you do
      that is longer than a few sentences should be organized into paragraphs.
      This is because paragraphs show a reader where the subdivisions of an
      essay begin and end, and thus help the reader see the organization of
the
      essay and grasp its main points.
    </p>
    <p style="color: white; font-size: larger">
      Paragraphs can contain many different kinds of information. A paragraph
      could contain a series of brief examples or a single long illustration
of
      a general point. It might describe a place, character, or process;
narrate
      a series of events; compare or contrast two or more things; classify
items
      into categories; or describe causes and effects. Regardless of the kind
of
```

```
      information they contain, all paragraphs share certain characteristics.
      One of the most important of these is a topic sentence.
    </p>

    <p style="color: white; font-size: larger">
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet fugiat
      sapiente doloremque dolores, voluptates, quisquam accusantium inventore
      quas nisi ratione autem expedita, id sint corrupti aperiam deserunt.
      Laborum quasi explicabo cum autem officiis, esse dignissimos tenetur
      dolor! Maiores fugit magnam quos libero illum, perspiciatis error quod
      quia consequuntur, nobis qui doloribus, voluptates reprehenderit neque
at
      ad. Rerum odio at eius aperiam, dicta, inventore asperiores provident
      assumenda animi, iusto adipisci deserunt libero ab dolores fuga nostrum.
      Consectetur amet non necessitatibus labore quaerat illo voluptas, ut
      temporibus beatae soluta quia, cumque iure delectus nulla aliquid quam
      sint. Quidem voluptatibus minus tempora velit ipsam, fugit perspiciatis
      placeat eius accusantium libero? Quidem libero similique ratione maiores
      numquam fugiat inventore repudiandae eveniet possimus quae facere
      voluptatem accusantium vitae quaerat amet iste, molestias suscipit
      voluptate alias non magni tempora expedita fugit! Doloribus accusamus
      recusandae explicabo exercitationem modi ipsam praesentium cumque.
      Dolorum, debitis recusandae reprehenderit assumenda repellat iusto eos
      veniam. Asperiores beatae perferendis provident aperiam architecto,
      inventore ut fugit illo voluptates accusantium labore dolores ea dolor
      molestiae tempora amet. Laudantium aut eum nostrum atque deserunt odio,
      corporis optio? Rem incidunt iure alias, fugit, quae corporis,
reiciendis
      facilis sit iusto perspiciatis ab ipsa nulla cupiditate beatae autem
      blanditiis sed quos nihil excepturi voluptates? Nostrum culpa aliquam
      sequi eligendi quo voluptatum, ipsum quaerat est consequuntur quasi
      accusamus saepe perspiciatis voluptates earum provident blanditiis
neque,
      voluptate, iusto tenetur itaque ex sint. Veritatis quia consequatur
      exercitationem velit molestias, quasi quam architecto autem asperiores
sit
      quaerat, nesciunt ratione natus inventore quis officia consectetur, qui
      iusto unde consequuntur beatae.
    </p>
    <div class="box">Box</div>
    <footer>&copy; Sai 2024</footer>
  </body>
</html>
```

Option 1 Option 2 Option 3 Option 4

Paragraphs & Topic Sentences

A paragraph is a series of sentences that are organized and coherent, and are all related to a single topic. Almost every piece of writing you do that is longer than a few sentences should be organized into paragraphs. This is because paragraphs show a reader where the subdivisions of an essay begin and end, and thus help the reader see the organization of the essay and grasp its main points.

Paragraphs can contain many different kinds of information. A paragraph could contain a series of brief examples or a single long illustration of a general point. It might describe a place, character, or process; narrate a series of events; compare or contrast two or more things; classify items into categories; or describe causes and effects. Regardless of the kind of information they contain, all paragraphs share certain characteristics. One of the most important of these is a topic sentence.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet fugiat sapiente doloremque dolores, voluptates, quisquam accusantium inventore quas nisi ratione autem expedita, id sint corrupti aperiam deserunt. Laborum quasi explicabo cum autem officiis, esse dignissimos tenetur dolor! Maiores fugit magnam quos libero illum, perspiciatis error quod quia consequuntur, nobis qui doloribus, voluptates reprehenderit neque at ad. Rerum odio at eius aperiam, dicta, inventore asperiores provident assumenda animi, iusto adipisci deserunt libero ab dolores fuga nostrum. Consectetur amet non necessitatibus labore quaerat illo voluptas, ut temporibus beatae soluta quia, cumque iure delectus nulla aliquid quam sint. Quidem voluptatibus minus tempora velit ipsam, fugit perspiciatis placeat eius accusantium libero? Quidem libero similique ratione maiores numquam fugiat inventore repudiandae eveniet possimus quae facere voluptatem accusantium vitae quaerat amet iste, molestias suscipit voluptate alias non magni tempora expedita fugit! Doloribus accusamus recusandae explicabo exercitationem modi ipsam praesentium cumque. Dolorum, debitis recusandae reprehenderit assumenda repellat iusto eos veniam. Asperiores beatae perferendis provident aperiam architecto, inventore ut fugit illo voluptates accusantium labore dolores ea dolor molestiae tempora amet. Laudantium aut eum nostrum atque deserunt odio, corporis optio? Rem incidunt iure alias, fugit, quae corporis, reiciendis facilis sit iusto perspiciatis ab ipsa nulla cupiditate beatae autem blanditiis sed quos nihil excepturi voluptates? Nostrum culpa aliquam sequi eligendi quo voluptatum, ipsum quaerat est consequuntur quasi accusamus saepe perspiciatis voluptates earum provident blanditiis neque, voluptate, iusto tenetur itaque ex sint. Veritatis quia consequatur exercitationem velit molestias, quasi quam architecto autem asperiores sit quaerat, nesciunt ratione natus inventore quis officia consectetur, qui iusto unde consequuntur beatae.

Experiment 4: Responsive Design with Media Queries

1. Add a CSS media query and appropriate styles so that the webpage looks similar even when resized to smaller widths. Specifically:
   The sidebar should be hidden.
   The body should have no padding.
   The images shouldn't exceed the width of the window.
   The navigation items should each be on their own line.
   The header should be fixed, so that it stays at the top after scrolling.

```
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" href="style4.css" >
    <title>EXP-4</title>
</head>
<body>

  <div class="container">
    <div class="sidebar">
        <a href="#">Blogs</a>
        <a href="#">Achievements</a>
        <a href="projects.html">Works</a>
        <a href="about.html">Skills</a>
    </div>


  <div class="navbar_about">
    <a style="position: absolute; left: 8%; top: -0.5%; font-size: 200%
!important;" href="#">Sai</a>
      <a href="contact.html">Contact</a>
      <a href="projects.html">Projects</a>
      <a href="about.html">About</a>
      <a href="index.html">Home</a>
</div>
<h1 style="text-align: center; color: #000000; font-weight: 600; font-size:
48px;">Contact Me</h1>
<div class="box">
  <div class="info">
    <h2 style="position: relative; z-index: 1; margin-left: 20%;">Sai</h2>
```

```
      <img class="icon" src="location.png"><h3 class="text">University of
Petroleum & Energy Studies </h3><br>
      <img class="icon" src="location.png"><h3 class="text">Dehradun </h3><br>
      <img class="icon" src="email-logo.png"><h3 class="text"><a
href="www.ranjansai1214@gmail.com">ranjansai1214@gmail.com</a></h3><br>
      <img class="icon" src="github-logo.png"><h3 class="text"><a
target="_blank" href="https://github.com/Saitech5">sai-tech5</a> </h3><br>
      <img class="icon" src="linkedin-logo.png"><h3 class="text"><a
target="_blank" href= "#"> sai</a></h3>
  </div>
  <div class="text-content">
    <h2>Let's Connect</h2>

    <div class="row">
      <div class="rectangle">
        <input type="text" id="name" class="input-field" placeholder="Enter
your name">
        <label class="form-label">Enter your name</label>
      </div>

    </div>
    <div class="row">
      <div class="rectangle">
        <input type="email" id="email" class="input-field" placeholder="Enter
your email">
        <label class="form-label">Enter your E-mail</label>
      </div>

    </div>
    <div class="row">
      <div class="rectangle3">
        <textarea id="message" class="input-field" placeholder="Type your
message"></textarea>
        <label class="form-label" id="message">Type your message</label>
      </div>

    </div>
    <div style="display:flex; justify-content: center; width: 500px;">
        <button type="button" id="sendButton">Send</button>
    </div>
    </div>
    </div>

    <div class="footer">
        © 2004 Sai. All rights reserved.
    </div>
  </body>
  </html>
```


Blogs
Sai
Achievements
Works
Skills
Home
About
Projects
Contact
Contact Me
Sai
University of Petroleum & Energy Studies
Dehradun
ranjansai1214@gmail.com
sai-tech5
Let's Connect
Enter your name
Enter your E-mail
Type your message
© 2004 Sai. All rights reserved.

EXPERIMENT – 5

1. Write a JavaScript program that displays the largest integer among two integers.

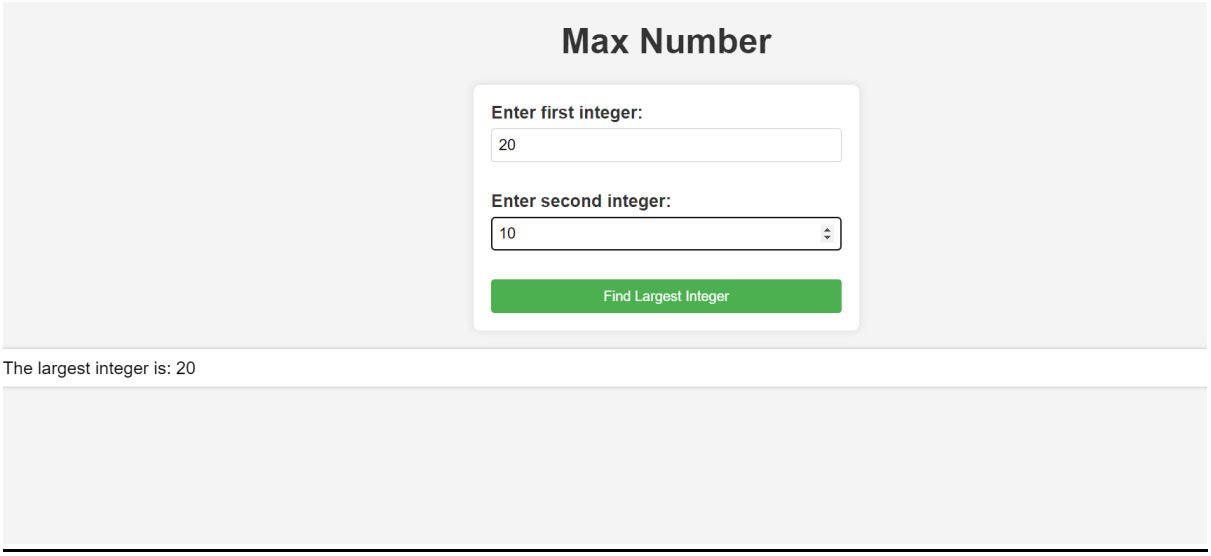
CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Display Largest Integer</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        h1 {
            text-align: center;
            color: #333;
        }

        #integerForm {
            max-width: 400px;
            margin: 0 auto;
            background-color: #fff;
            padding: 20px;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        }

        label {
            font-weight: bold;
            color: #333;
        }

        input[type="number"] {
            width: 100%;
```

```
            padding: 8px;
            margin-top: 6px;
            margin-bottom: 10px;
            border: 1px solid #ccc;
            border-radius: 4px;
            box-sizing: border-box;
        }

        button {
            width: 100%;
            background-color: #4CAF50;
            color: white;
            padding: 10px 20px;
            border: none;
            border-radius: 4px;
            cursor: pointer;
            font-size: 16px;
        }

        button:hover {
            background-color: #45a049;
        }

        #result {
            margin-top: 20px;
            padding: 10px;
            background-color: #fff;
            border: 1px solid #ccc;
            border-radius: 4px;
            box-shadow: 0 0 5px rgba(0, 0, 0, 0.1);
        }
    </style>
</head>
<body>
    <h1>Max Number</h1>
    <form id="integerForm">
        <label for="num1">Enter first integer:</label>
        <input type="number" id="num1" name="num1" required><br><br>
        <label for="num2">Enter second integer:</label>
        <input type="number" id="num2" name="num2" required><br><br>
        <button onclick="findlargest(event)">Find Largest Integer</button>
    </form>

    <div id="result"></div>

    <script>
        function displayLargestInteger(num1, num2) {
            if (num1 > num2) {
```

```
                return num1;
            } else {
                return num2;
            }
        }

        function findlargest(e){
            e.preventDefault();
            var num1 = parseInt(document.getElementById("num1").value);
            var num2 = parseInt(document.getElementById("num2").value);
            var result = displayLargestInteger(num1, num2);
            document.getElementById("result").innerHTML = "The largest integer
is: " + result; // Displaying the result
        }
    </script>
</body>
</html>
```

OUTPUT:



2. Write a JavaScript function that accepts a string as a parameter and converts the first letter of each word into upper case.

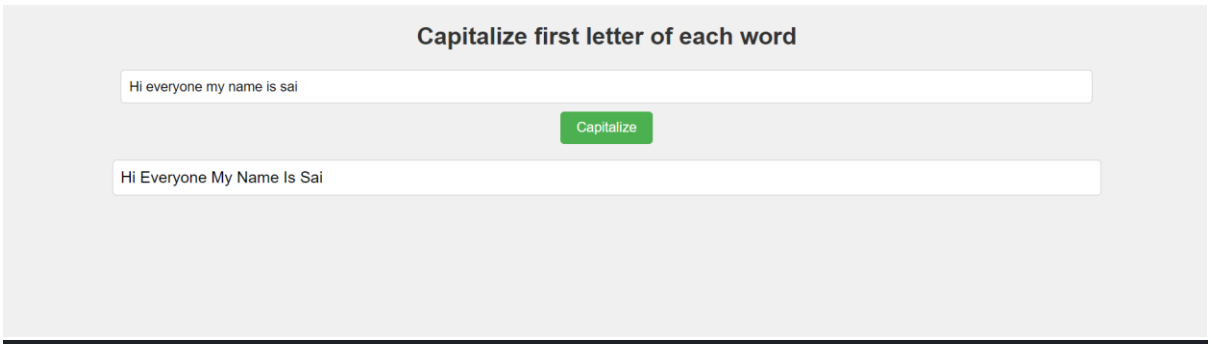
CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Capitalize</title>
<style>
    body {
        font-family: Arial, sans-serif;
        background-color: #f0f0f0;
        margin: 0;
        padding: 0;
    }

    h2 {
        text-align: center;
        color: #333;
    }

    input[type="text"] {
        width: 80%;
        padding: 10px;
        margin: 10px auto;
        display: block;
        border: 1px solid #ccc;
        border-radius: 5px;
        box-sizing: border-box;
    }

    button {
        padding: 10px 20px;
        margin: 10px auto;
        display: block;
        background-color: #4CAF50;
        color: white;
        border: none;
        border-radius: 5px;
        cursor: pointer;
    }

    button:hover {
        background-color: #45a049;
    }

    #output {
        width: 80%;
        margin: 20px auto;
        padding: 10px;
        background-color: #fff;
        border: 1px solid #ccc;
        border-radius: 5px;
    }
</style>
```

```
<script>
function capitalizeFirstLetterOfEachWord(str) {
    return str.replace(/\b\w/g, function(char) {

 /* \b\w matches the first character of each word in the string. The \b
ensures that it matches at the beginning of a word, and \w matches any word
character.
 /g ensures that this pattern is applied globally, so it finds all occurrences
of the pattern */

        return char.toUpperCase();
    });
}

function capitalizeAndDisplay() {
    let inputText = document.getElementById("inputText").value;
    let capitalizedText = capitalizeFirstLetterOfEachWord(inputText);
    document.getElementById("output").innerText = capitalizedText;
}
</script>
</head>
<body>

<h2>Capitalize first letter of each word</h2>

<input type="text" id="inputText" placeholder="Enter a sentence">
<button onclick="capitalizeAndDisplay()">Capitalize</button>
<div id="output"></div>

</body> </html>
```

OUTPUT:



3. Write a Java Script to create a simple calculator.

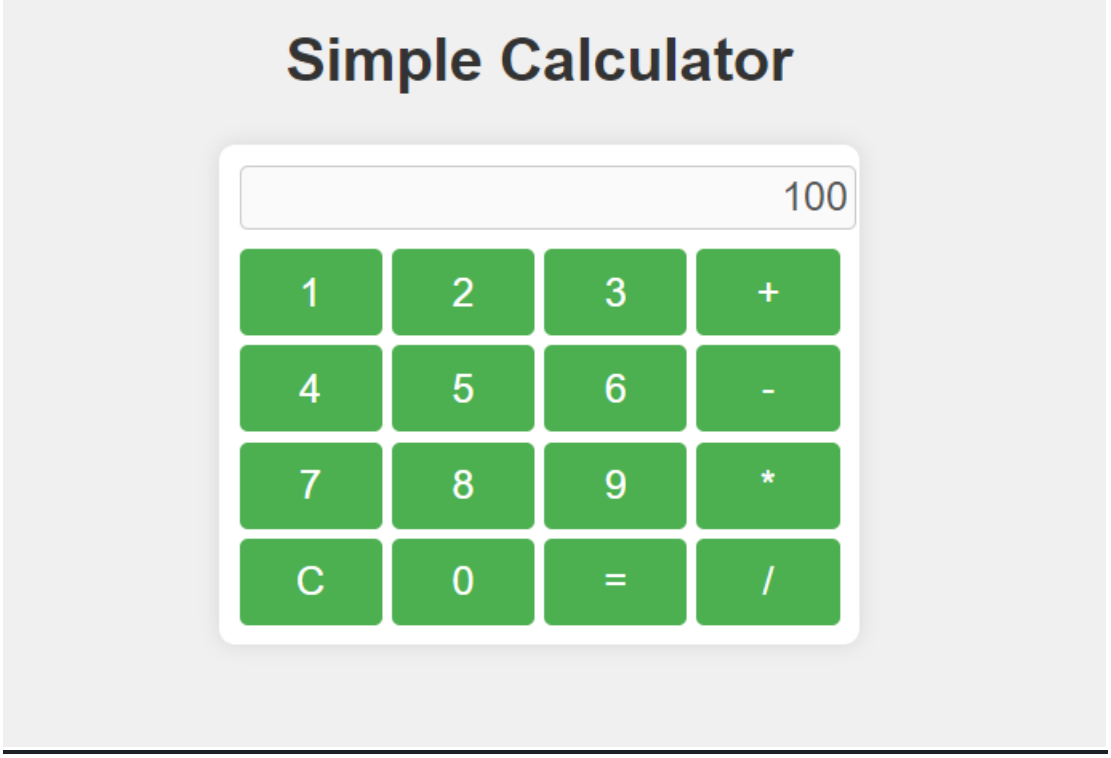
CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Simple Calculator</title>
<style>
    body {
        font-family: Arial, sans-serif;
        background-color: #f0f0f0;
        margin: 0;
        padding: 0;
    }
    h2 {
        text-align: center;
        color: #333;
    }
    .calculator {
        width: 300px;
        margin: 20px auto;
        padding: 10px;
        background-color: #fff;
        border-radius: 8px;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }
    #display {
        width: 100%;
        margin-bottom: 10px;
        padding: 3.5px;
        font-size: 20px;
        text-align: right;
        border: 1px solid #ccc;
        border-radius: 4px;
        margin-right: 30px;
    }
    .btn-container {
        display: grid;
        grid-template-columns: repeat(4, 1fr);
        grid-gap: 5px;
    }
    input[type="button"] {
        padding: 10px;
        font-size: 20px;
```

```
        background-color: #4CAF50;
        color: white;
        border: none;
        border-radius: 4px;
        cursor: pointer;
    }
    input[type="button"]:hover {
        background-color: #45a049;
    }
</style>
</head>
<body>

<h2>Simple Calculator</h2>

<div class="calculator">
    <input type="text" id="display" disabled>
    <div class="btn-container">
        <input type="button" value="1" onclick="appendToDisplay('1')">
        <input type="button" value="2" onclick="appendToDisplay('2')">
        <input type="button" value="3" onclick="appendToDisplay('3')">
        <input type="button" value="+" onclick="appendToDisplay('+')">
        <input type="button" value="4" onclick="appendToDisplay('4')">
        <input type="button" value="5" onclick="appendToDisplay('5')">
        <input type="button" value="6" onclick="appendToDisplay('6')">
        <input type="button" value="-" onclick="appendToDisplay('-')">
        <input type="button" value="7" onclick="appendToDisplay('7')">
        <input type="button" value="8" onclick="appendToDisplay('8')">
        <input type="button" value="9" onclick="appendToDisplay('9')">
        <input type="button" value="*" onclick="appendToDisplay('*')">
        <input type="button" value="C" onclick="clearDisplay()">
        <input type="button" value="0" onclick="appendToDisplay('0')">
        <input type="button" value="=" onclick="calculate()">
        <input type="button" value="/" onclick="appendToDisplay('/')">
    </div>
</div>

<script>
function appendToDisplay(value) {
    document.getElementById('display').value += value;
}

function clearDisplay() {
    document.getElementById('display').value = '';
}

function calculate() {
    let expression = document.getElementById('display').value;
```

```
    let result = eval(expression); /*built in JS func */
    document.getElementById('display').value = result;
}
</script>

</body>
</html>
```

OUTPUT:



4. Write a JavaScript function that accepts a string as a parameter and finds the longest word within the string.

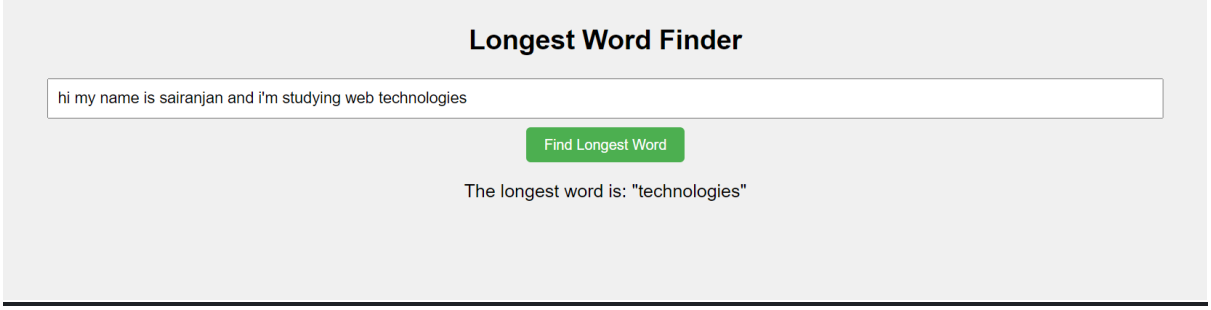
CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Longest Word Finder</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f0f0f0;
```

```
      margin: 0;
      padding: 0;
    }
    #container {
      width: 80%;
      margin: 50px auto;
      text-align: center;
    }
    input[type="text"] {
      width: 100%;
      padding: 10px;
      margin-bottom: 10px;
      box-sizing: border-box;
    }
    button {
      padding: 10px 20px;
      font-size: 16px;
      background-color: #4CAF50;
      color: white;
      border: none;
      border-radius: 5px;
      cursor: pointer;
    }
    button:hover {
      background-color: #45a049;
    }
    #result {
      margin-top: 20px;
    }
  </style>
</head>
<body>

<div id="container">
  <h2>Longest Word Finder</h2>
  <input type="text" id="inputString" placeholder="Enter your sentence here">
  <button type="button" onclick="findLongestWord()">Find Longest Word</button>
  <p id="result"></p>
</div>

<script>
  function findLongestWord() {
    const inputString = document.getElementById("inputString").value.trim();

    if (!inputString) {
      alert("Please enter a sentence.");
      return;
    }
```

```
    const words = inputString.split(/\s+/); /*\s+ matches one or more
whitespace characters */

    let longestWord = "";
    let longestWordLength = 0;

    for (const word of words) {
      if (word.length > longestWordLength) {
        longestWord = word;
        longestWordLength = word.length;
      }
    }
    const resultElement = document.getElementById("result");
    resultElement.textContent = `The longest word is: "${longestWord}"`;
  }
</script>
</body>
</html>
```

OUTPUT:



5. Write a JavaScript program to find odd and even numbers from 1 to 100.

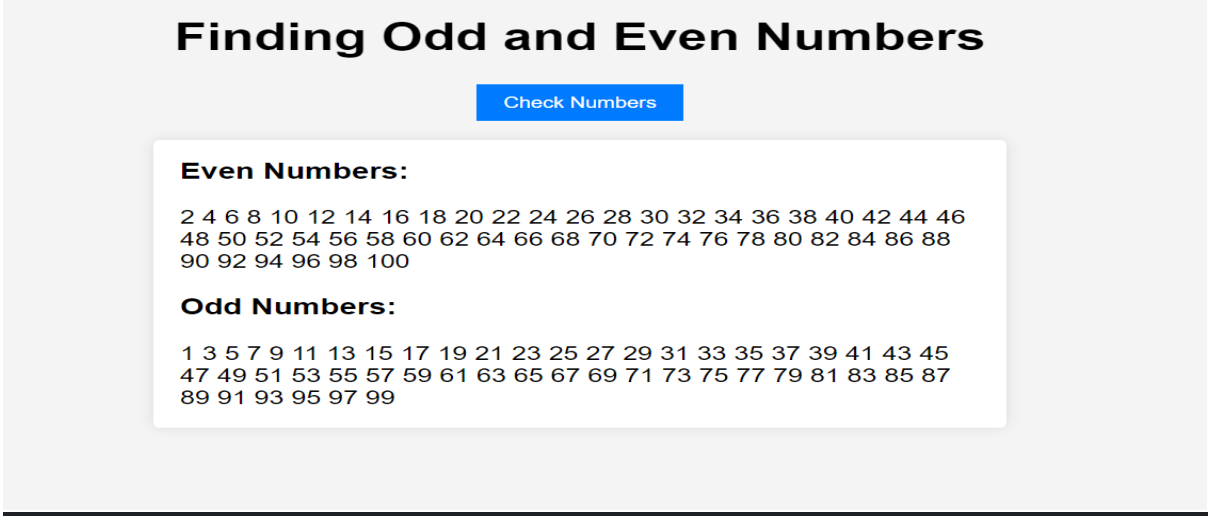
CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Odd and Even Numbers</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
```

```
      padding: 0;
      background-color: #f4f4f4;
    }

    h1 {
      text-align: center;
    }

    button {
      display: block;
      margin: 20px auto;
      padding: 10px 20px;
      font-size: 16px;
      border: none;
      background-color: #007bff;
      color: #fff;
      cursor: pointer;
    }

    button:hover {
      background-color: #0056b3;
    }

    #results {
      max-width: 600px;
      margin: 0 auto;
      padding: 20px;
      background-color: #fff;
      border-radius: 5px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }

    h3 {
      margin-top: 0;
    }

    p {
      margin: 0;
    }
  </style>
</head>
<body>
  <h1>Finding Odd and Even Numbers</h1>
  <button onclick="checkNumbers()">Check Numbers</button>
  <div id="results"></div>
  <script>
    function checkNumbers() {
      const results = document.getElementById("results");
```

```
      results.innerHTML = ""; // Clear previous results

      let evenNumbers = "";
      let oddNumbers = "";

      for (let i = 1; i <= 100; i++) {
        if (i % 2 === 0) {
          evenNumbers += i + " ";
        } else {
          oddNumbers += i + " ";
        }
      }

      results.innerHTML = `<h3>Even Numbers:</h3> <p>${evenNumbers}</p> <br>
<h3>Odd Numbers:</h3> <p>${oddNumbers}</p>`;
    }
  </script>
</body>
</html>
```

OUTPUT:



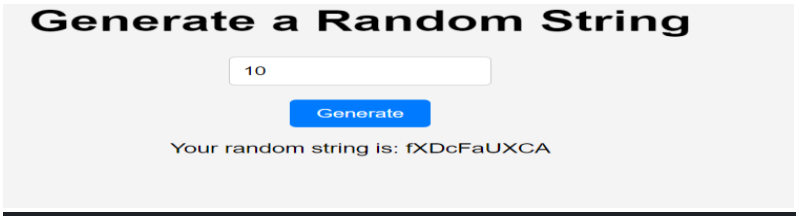
6. Write a JavaScript program to generate a random string.

CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Random String Generator</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #f4f4f4;
    }

    h1 {
      text-align: center;
    }

    input[type="number"] {
      display: block;
      margin: 20px auto;
      padding: 10px;
      font-size: 16px;
      width: 200px;
      border: 1px solid #ccc;
      border-radius: 5px;
      box-sizing: border-box;
    }

    button {
      display: block;
      margin: 10px auto;
      padding: 10px 20px;
      font-size: 16px;
      border: none;
      background-color: #007bff;
      color: #fff;
      cursor: pointer;
      border-radius: 5px;
    }

    button:hover {
```

```
      background-color: #0056b3;
    }

    p#result {
      text-align: center;
      margin-top: 20px;
      font-size: 18px;
    }
  </style>
</head>
<body>
  <h1>Generate a Random String</h1>
  <input type="number" id="stringLength" placeholder="Enter String Length">
  <button onclick="generateString()">Generate</button>
  <p id="result"></p>
  <script>
    function generateString() {
      const stringLength = document.getElementById("stringLength").value;
      const result = document.getElementById("result");

      if (stringLength === "" || isNaN(stringLength) || stringLength <= 0) {
        result.textContent = "Please enter a valid positive number for string
length.";
        return;
      }

      const characters =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
      let randomString = "";

      for (let i = 0; i < stringLength; i++) {
        const randomIndex = Math.floor(Math.random() * characters.length);
        randomString += characters.charAt(randomIndex);
      }
      result.textContent = "Your random string is: " + randomString;
    }
  </script>
</body>
</html>
```

OUTPUT:



7. Write a JavaScript Program to Print All Prime Numbers in an Interval.

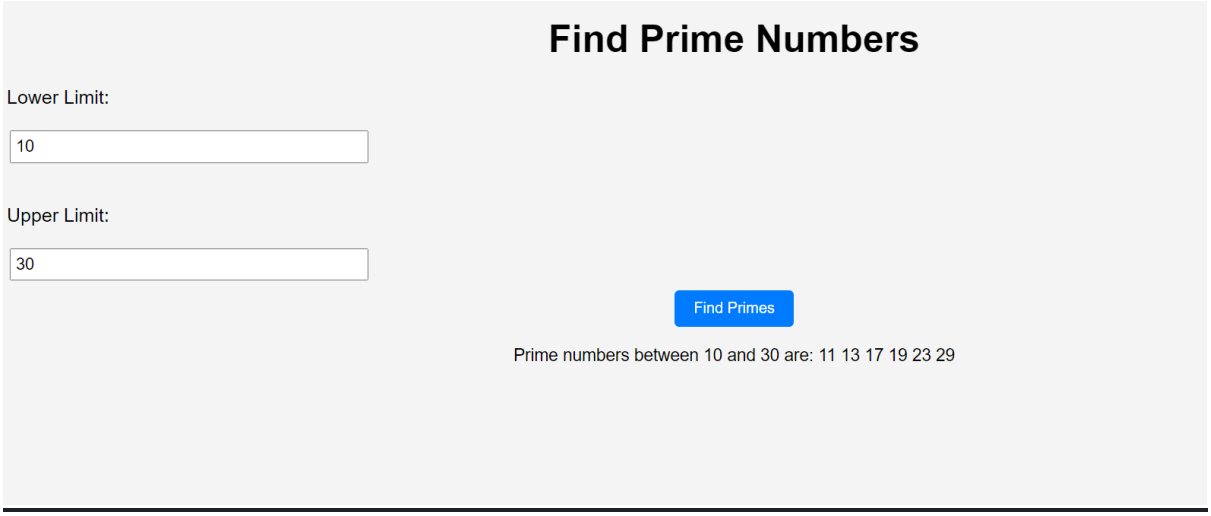
CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Prime Number Finder</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #f4f4f4;
      margin-left: 4px;
    }

    h1 {
      text-align: center;
    }

    label {
      display: block;
      margin-top: 10px;
    }

    input[type="number"] {
      display: block;
      margin-bottom: 10px;
      padding: 5px;
      width: calc(25% - 5px);
      box-sizing: border-box;
      margin-left:3px ;
    }

    button {
      display: block;
      margin: 10px auto;
      padding: 10px 20px;
      font-size: 16px;
      border: none;
      background-color: #007bff;
      color: #fff;
      cursor: pointer;
```

```
      border-radius: 5px;
    }

    button:hover {
      background-color: #0056b3;
    }

    p#result {
      text-align: center;
      margin-top: 20px;
      font-size: 18px;
    }
  </style>
</head>
<body>
  <h1>Find Prime Numbers</h1>
  <label for="lowerLimit">Lower Limit:</label> <br>
  <input type="number" id="lowerLimit" placeholder="Enter lower limit"> <br>
  <label for="upperLimit">Upper Limit:</label> <br>
  <input type="number" id="upperLimit" placeholder="Enter upper limit">
  <button onclick="findPrimes()">Find Primes</button>
  <p id="result"></p>
  <script>
    function isPrime(num) {
      if (num <= 1) {
        return false;
      }
      for (let i = 2; i <= Math.sqrt(num); i++) {
        if (num % i === 0) {
          return false;
        }
      }
      return true;
    }

    function findPrimes() {
      const lowerLimit =
parseInt(document.getElementById("lowerLimit").value);
      const upperLimit =
parseInt(document.getElementById("upperLimit").value);
      const result = document.getElementById("result");

      result.textContent = "Prime numbers between " + lowerLimit + " and " +
upperLimit + " are:";

      for (let i = lowerLimit; i <= upperLimit; i++) {
        if (isPrime(i)) {
          result.textContent += " " + i;
```

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

OUTPUT:



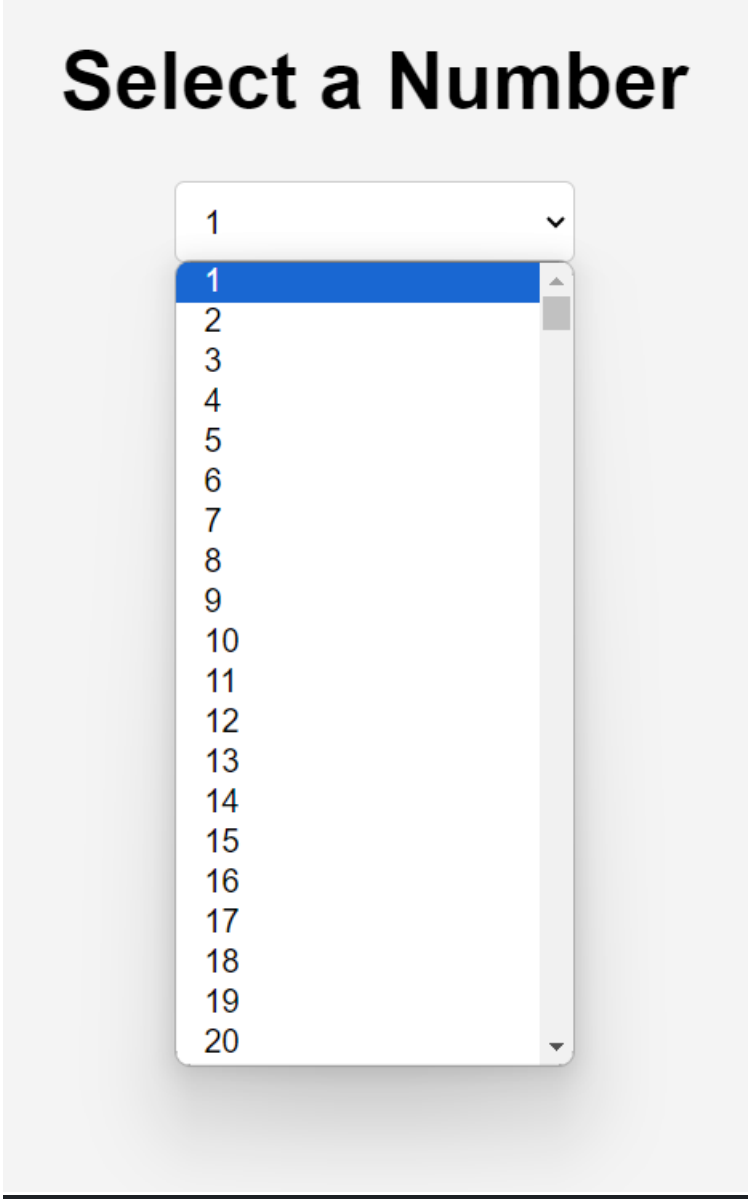
8. Write a JavaScript program to populate a drop-down box from 1 to 1000.

CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Number Dropdown</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #f4f4f4;
    }

    h1 {
      text-align: center;
```

```
    }

    select {
      display: block;
      margin: 20px auto;
      padding: 10px;
      font-size: 16px;
      width: 200px;
      border: 1px solid #ccc;
      border-radius: 5px;
      box-sizing: border-box;
    }
  </style>
</head>
<body>
  <h1>Select a Number</h1>
  <select id="numberDropdown"></select>
  <script>
    const dropdown = document.getElementById("numberDropdown");

    function populateDropdown() {
      for (let i = 1; i <= 1000; i++) {
        const option = document.createElement("option");
        option.value = i;
        option.text = i;
        dropdown.appendChild(option);
      }
    }

    populateDropdown();
  </script>
</body>
</html>
```

OUTPUT:

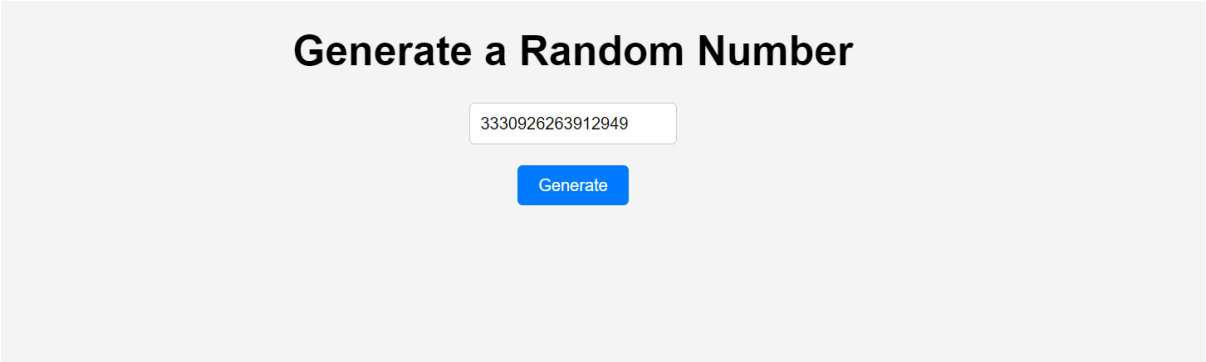


Experiment 6: Java Script Event Handling and Functions

1. Write a Java script program to Generate a Random Number and display it in a textbox.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Random Number Generator</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      margin: 0;
      padding: 0;
    }

    h1 {
      text-align: center;
    }

    input[type="text"] {
      display: block;
      margin: 20px auto;
      padding: 10px;
      font-size: 16px;
      width: 200px;
      border: 1px solid #ccc;
      border-radius: 5px;
      box-sizing: border-box;
    }

    button {
      display: block;
      margin: 10px auto;
      padding: 10px 20px;
      font-size: 16px;
      border: none;
      background-color: #007bff;
      color: #fff;
      cursor: pointer;
      border-radius: 5px;
    }
```

```
    button:hover {
      background-color: #0056b3;
    }
  </style>
</head>
<body>
  <h1>Generate a Random Number</h1>
  <input type="text" id="randomNumber" readonly>
  <button onclick="generateRandomNumber()">Generate</button>
  <script>
    function generateRandomNumber() {
      const randomNumber = Math.floor(Math.random() *
Number.MAX_SAFE_INTEGER);
      document.getElementById("randomNumber").value = randomNumber;
    }
  </script>
</body>
</html>
```



2. Write a JavaScript function that changes the background color of an element on a particular event.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Change Background Color</title>
</head>
<body>

<div id="myElement" style="width: 200px; height: 200px; background-color:
yellow; text-align: center; line-height: 200px;">Click me to change
color!</div>
```

```
<script>
function changeColor() {
  var element = document.getElementById("myElement");
  var randomColor = '#' + Math.floor(Math.random()*16777215).toString(16); //
generates a random hex color
  element.style.backgroundColor = randomColor;
}

// Attaching the event listener
document.getElementById("myElement").addEventListener("click", changeColor);
</script>

</body>
</html>
```

color!

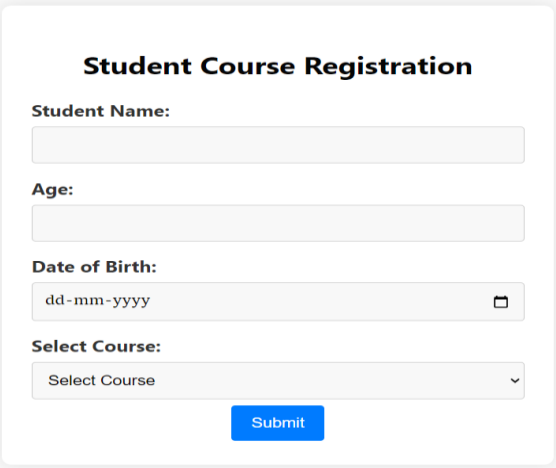
3. Write a Java script to validate course registration form.

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Student Course Registration</title>
    <style>
```

```css
body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    background-color: #f9f9f9;
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
}

form {
    background-color: #fff;
    padding: 30px;
    border-radius: 10px;
    box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);
    text-align: center;
    width: 80%;
    max-width: 500px;
}

h2 {
    color: #050506;
}

label {
    display: block;
    text-align: left;
    margin-top: 10px;
    font-weight: bold;
    color: #333;
}

input, select {
    width: 100%;
    padding: 12px;
    margin: 8px 0;
    box-sizing: border-box;
    border: 1px solid #ccc;
    border-radius: 4px;
    background-color: #f8f8f8;
}

input:focus, select:focus {
    outline: none;
    border-color: #007bff;
    background-color: #fff;
}
```

```
        button {
            background-color: #007bff;
            color: #fff;
            padding: 12px 20px;
            border: none;
            border-radius: 4px;
            cursor: pointer;
            transition: background-color 0.3s ease;
        }

        button:hover {
            background-color: #0056b3;
        }
    </style>
</head>

<body>
    <form id="registrationForm">
        <h2>Student Course Registration</h2>

        <label for="studentName">Student Name:</label>
        <input type="text" id="studentName" name="studentName" required>

        <label for="age">Age:</label>
        <input type="number" id="age" name="age" min="1" required>

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

        <label for="course">Select Course:</label>
        <select id="course" name="course" required>
            <option value="">Select Course</option>
            <option value="computerScience">Computer Science</option>
            <option value="engineering">Engineering</option>
            <option value="aeronautics">Aeronautics</option>
            <option value="business">Business Administration</option>
        </select>

        <button type="submit">Submit</button>
    </form>

    <script>
        document.getElementById('registrationForm').addEventListener('submit',
function(event) {
            // Prevent default form submission
            event.preventDefault();
```

```
            // Validate inputs
            var studentName = document.getElementById('studentName').value;
            var age = document.getElementById('age').value;
            var dob = document.getElementById('dob').value;
            var course = document.getElementById('course').value;

            if (studentName.trim() === '') {
                alert('Please enter student name');
                return;
            }

            if (isNaN(age) || age < 1) {
                alert('Please enter a valid age');
                return;
            }

            if (dob.trim() === '') {
                alert('Please enter date of birth');
                return;
            }

            if (course === '') {
                alert('Please select a course');
                return;
            }

            // If all inputs are valid, submit the form
            this.submit();
        });
    </script>
</body>

</html>
```



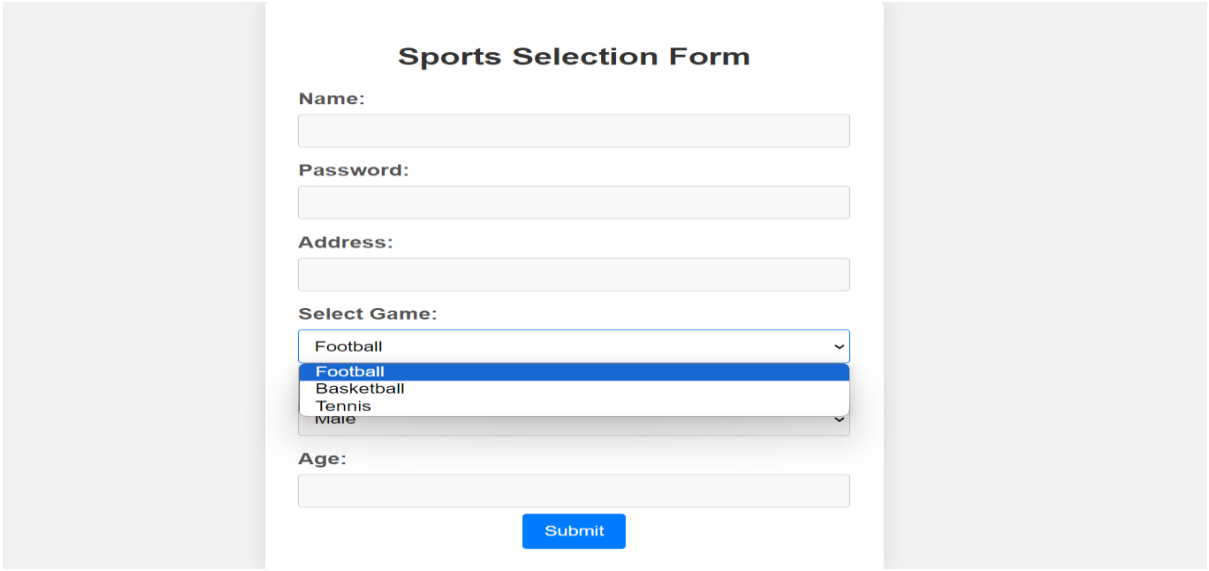
4. Write a JavaScript program that adds a keydown event listener to a text input to detect when the "Enter key" is pressed.

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sports Selection Form</title>
    <style>
        body {
            font-family: 'Arial', sans-serif;
            background-color: #f2f2f2;
            margin: 0;
            padding: 0;
            display: flex;
            justify-content: center;
            align-items: center;
            min-height: 100vh;
        }

        form {
            background-color: #fff;
            padding: 30px;
            border-radius: 8px;
            box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);
            text-align: center;
            width: 80%;
            max-width: 500px;
        }
```

```
        h2 {
            color: #333;
        }

        label {
            display: block;
            text-align: left;
            margin-top: 10px;
            font-weight: bold;
            color: #555;
        }

        input, select {
            width: 100%;
            padding: 10px;
            margin: 8px 0;
            box-sizing: border-box;
            border: 1px solid #ccc;
            border-radius: 4px;
            background-color: #f8f8f8;
        }

        input:focus, select:focus {
            outline: none;
            border-color: #007bff;
            background-color: #fff;
        }

        button {
            background-color: #007bff;
            color: #fff;
            padding: 12px 20px;
            border: none;
            border-radius: 4px;
            cursor: pointer;
            transition: background-color 0.3s ease;
        }

        button:hover {
            background-color: #0056b3;
        }
    </style>
</head>

<body>
    <form id="sportsForm">
        <h2>Sports Selection Form</h2>
```

```
        <label for="name">Name:</label>
        <input type="text" id="name" name="name" required>

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

        <label for="address">Address:</label>
        <input type="text" id="address" name="address" required>

        <label for="selectGame">Select Game:</label>
        <select id="selectGame" name="selectGame" required>
            <option value="football">Football</option>
            <option value="basketball">Basketball</option>
            <option value="tennis">Tennis</option>
        </select>

        <label for="gender">Gender:</label>
        <select id="gender" name="gender" required>
            <option value="male">Male</option>
            <option value="female">Female</option>
            <option value="other">Other</option>
        </select>

        <label for="age">Age:</label>
        <input type="number" id="age" name="age" min="1" required>

        <button type="submit">Submit</button>
    </form>

    <script>
        const textInputs = document.querySelectorAll('input[type="text"],
input[type="password"], input[type="number"]');
        textInputs.forEach(input => {
            input.addEventListener('keydown', function(event) {
                // Check if the Enter key is pressed (key code 13)
                if (event.keyCode === 13) {
                    alert("Are you sure you want to submit")
                    const nextInputIndex =
Array.from(textInputs).indexOf(this) + 1;
                    if (nextInputIndex < textInputs.length) {
                        textInputs[nextInputIndex].focus();
                    } else {
                        document.getElementById('sportsForm').submit();
                    }
                }
            });
        });
    </script>
```

```
</body>

</html>
```



5. Write a program to show the use of alert, prompt and confirm dialog boxes.

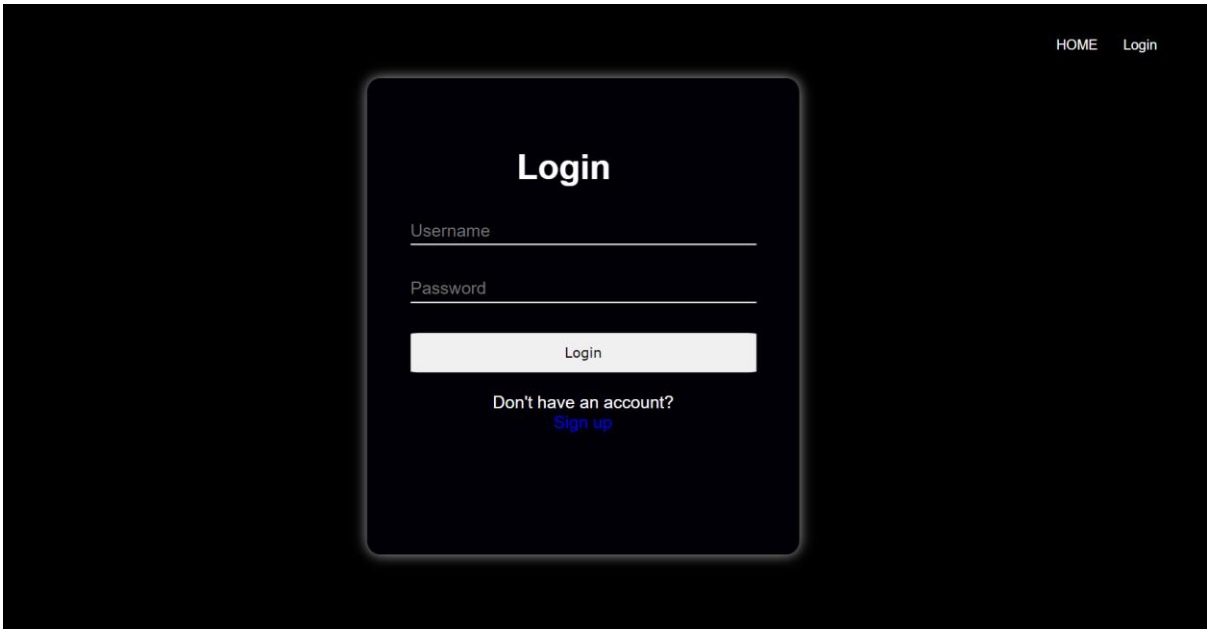
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Page</title>
    <link rel="stylesheet" href="login.css">
</head>
<style>
    *{
        font-family: Arial, Helvetica, sans-serif;
        margin: 0;
        padding: 0;
        text-decoration: none;
        box-sizing: border-box;
    }

    body{
        min-height: 100vh;
        width: 100%;
        background: url(2.jpeg);
        background-color: black;
```

```
        background-position: center;
        background-size: cover;
        position: relative;
    }

    form{
        background: rgba(6, 1, 30,0.2);
        width: 500px;
        height: 550px;
        padding: 80px 50px;
        position: absolute;
        left: 50%;
        top: 50%;
        transform: translate(-50%,-50%);
        border-radius: 3%;
        box-shadow: 0px 0px 15px rgba(255, 255, 255, 0.692);

    }

    h1 {
         text-align: center;
         margin-right: 45px;
         color: white;
         font-size: 40px;
    }

    .text{
        border-bottom: 2px solid white;
        position: relative;
        margin: 35px 0;
    }

    .text input{
        background: none;
        border: none;
        outline: none;
        width: 100;
        color: white;
        height: 30px;
        font-size: 20px;
    }
    .login{
        height: 45px;
        width: 100%;
        border: none;
        background-size: 200% ;
        font-size: 16px;
        border-radius: 3%;
```

```
        font-family:Verdana, Geneva, Tahoma, sans-serif, Arial, sans-serif;
    }

    .login:hover{
        background-position: right;
        font-size: 18px;
        background: orangered;
    }
    .su{
        color: white;
        text-align: center;
    }

    .header{
        min-height: 100vh;
        width: 100%;
        background-image: url(1.jpeg);
        background-color: black;
        background-position: center;
        background-size: cover;
        position: relative;
    }
    nav{
        display: flex;
        padding: 2% 6%;
        justify-content: space-between;
        align-items: center;

    }

    .navbar{
        flex: 1;
        text-align: right;
    }
    .navbar ul li {
        list-style: none;
        display: inline-block;
        padding: 8px 12px;
        position: relative;

    }

    .navbar ul li a {
        color: white ;
        text-decoration: none;
        font-size: 15px;
    }
```

```
    .navbar ul li::after{
        content: '';
        width: 0% ;
        height: 2px ;
        background: rgb(54, 183, 209);
        display: block;
        margin: auto;
        transition: 0.5s;
    }
</style>
<body>
    <section class="header">
        <nav>
            <div class="navbar" id="navbar">
                <ul>
                    <li><a href="/">HOME</a></li>
                    <li><a href="/signin">Login</a></li>
                </ul>
            </div>
        </nav>
    </section>
    <form id="loginForm">
        <h1>Login</h1>
        <div class="text">
            <input type="text" id="username" name="username"
placeholder="Username">
        </div>
        <div class="text">
            <input type="password" id="password" name="password"
placeholder="Password">
        </div>
        <input type="button" value="Login" class="login"
onclick="validateLogin()">
        <br><br>
        <div class="su">
            Don't have an account?
            <br>
            <a href="/signup">Sign up</a>
        </div>
    </form>

    <script>
        function validateLogin() {
            var username = document.getElementById("username").value;
            var password = document.getElementById("password").value;
            if (username.trim() === '' || password.trim() === '') {
                alert("Please enter both username and password.");
                return;
```

```
            }

            //confirm dialog for login confirmation
            var confirmed = confirm("Are you sure you want to login?");
            if (confirmed) {
                //prompt dialog to verify login
                var loginConfirmation = prompt("Please type 'yes' to
confirm.");
                if (loginConfirmation === 'yes') {
                    document.getElementById("loginForm").submit();
                } else {
                    return;
                }
            } else {
                return;
            }
        }
    </script>
</body>
</html>
```



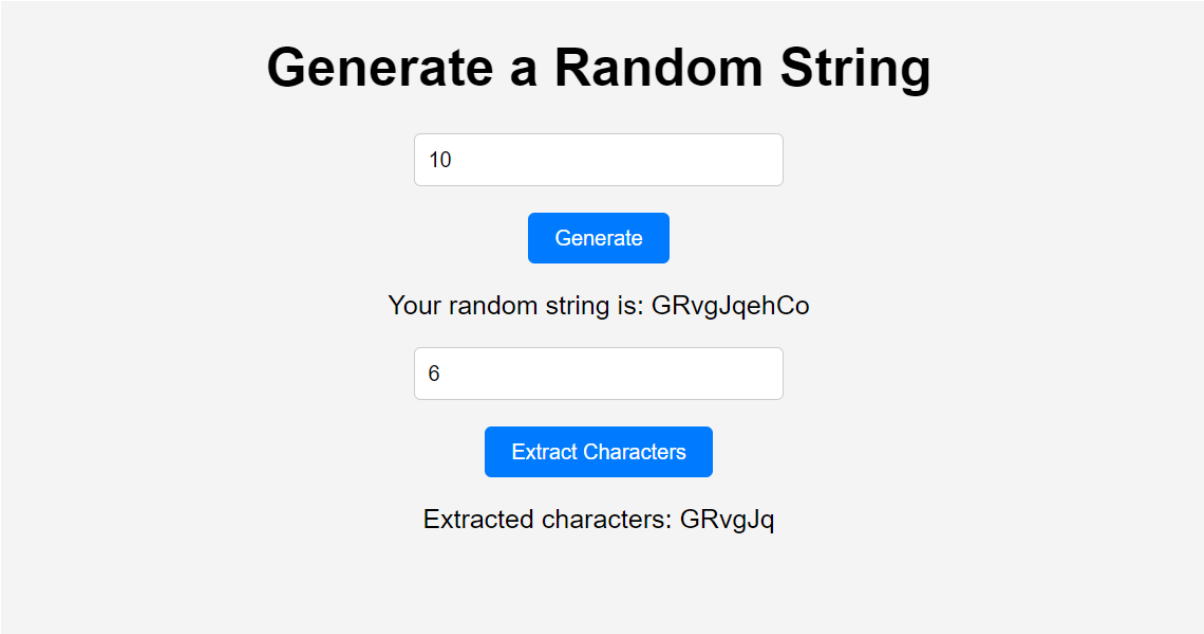
6. Write a JavaScript function to extract a specified number of characters from a string.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Random String Generator</title>
  <style>
        body {
          font-family: Arial, sans-serif;
          margin: 0;
          padding: 0;
          background-color: #f4f4f4;
        }

        h1 {
          text-align: center;
        }

        input[type="number"] {
          display: block;
          margin: 20px auto;
          padding: 10px;
          font-size: 16px;
          width: 280px;
          border: 1px solid #ccc;
          border-radius: 5px;
          box-sizing: border-box;
        }

        button {
          display: block;
          margin: 10px auto;
          padding: 10px 20px;
          font-size: 16px;
          border: none;
          background-color: #007bff;
          color: #fff;
          cursor: pointer;
          border-radius: 5px;
        }

        button:hover {
          background-color: #0056b3;
        }
        #randomStringMessage,#extractedCharactersMessage {
            text-align: center;
        }
  </style>
</head>
<body>
  <h1>Generate a Random String</h1>
  <input type="number" id="stringLength" placeholder="Enter String Length">
```

```
  <button onclick="generateString()">Generate</button>
  <p id="randomStringMessage"></p>
  <input type="number" id="charNumber" placeholder="Number of characters to
extract">
  <button onclick="extractCharacters()">Extract Characters</button>
  <p id="extractedCharactersMessage"></p>

  <script>
    let randomString = "";

    function generateString() {
      const stringLength = document.getElementById("stringLength").value;
      const result = document.getElementById("randomStringMessage");

      if (stringLength === "" || isNaN(stringLength) || stringLength <= 0) {
        result.textContent = "Please enter a valid positive number for string
length.";
        return;
      }

      const characters =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
      randomString = ""; // Reset randomString before generating a new one

      for (let i = 0; i < stringLength; i++) {
        const randomIndex = Math.floor(Math.random() * characters.length);
        randomString += characters.charAt(randomIndex);
      }
      result.textContent = "Your random string is: " + randomString;
    }

    function extractCharacters() {
      const charNumber = document.getElementById("charNumber").value;
      const result = document.getElementById("extractedCharactersMessage");

      if (charNumber === "" || isNaN(charNumber) || charNumber <= 0) {
        result.textContent = "Please enter a valid positive number for number
of characters to extract.";
        return;
      }

      if (randomString === "") {
        result.textContent = "Please generate a random string first.";
        return;
      }

      if (charNumber > randomString.length) {
```

```
        result.textContent = "Number of characters to extract exceeds the
length of the random string.";
        return;
      }

      const extractedCharacters = randomString.substring(0, charNumber);
      result.textContent = "Extracted characters: " + extractedCharacters;
    }
  </script>
</body>
</html>
```



7. Write a function to accept date of birth from a user and calculate the difference till current date.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Age Calculator</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      margin: 0;
```

```
      padding: 0;
    }

    h1 {
      text-align: center;
      margin-top: 20px;
    }

    label {
      display: block;
      text-align: center;
      margin-top: 20px;
    }

    input[type="date"] {
      display: block;
      margin: 0 auto;
      padding: 10px;
      font-size: 16px;
      border: 1px solid #ccc;
      border-radius: 5px;
    }

    button {
      display: block;
      margin: 20px auto;
      padding: 10px 20px;
      font-size: 16px;
      border: none;
      background-color: #007bff;
      color: #fff;
      cursor: pointer;
      border-radius: 5px;
    }

    button:hover {
      background-color: #0056b3;
    }

    #result {
      text-align: center;
      margin-top: 20px;
      font-size: 18px;
    }
  </style>
</head>
<body>
  <h1>Age Calculator</h1>
```

```
  <label for="dobInput">Enter your date of birth:</label> <br>
  <input type="date" id="dobInput">
  <button id="calculateButton">Calculate Age</button>
  <div id="result"></div>

  <script>
    document.getElementById("calculateButton").addEventListener("click",
function() {
      const dobInput = document.getElementById("dobInput").value;
      const ageDifference = calculateAgeDifference(dobInput);
      const resultElement = document.getElementById("result");

      resultElement.innerHTML = `<p>Your age is: ${ageDifference.years} years,
${ageDifference.months} months, and ${ageDifference.days} days.</p>`;
    });



    function calculateAgeDifference(dateOfBirth) {
      const dob = new Date(dateOfBirth);
      const currentDate = new Date();
      let difference = currentDate - dob;
      const daysDifference = Math.floor(difference / (1000 * 60 * 60 * 24));
      //difference in days by dividing the difference (which is in
milliseconds) by the number of milliseconds
      //in a day (1000 milliseconds * 60 seconds * 60 minutes * 24 hours).
      const years = Math.floor(daysDifference / 365);
      const months = Math.floor((daysDifference % 365) / 30);
      const days = daysDifference - (years * 365) - (months * 30);
      if (dob.getMonth() === currentDate.getMonth() && dob.getDate() ===
currentDate.getDate()) {
        alert("Happy Birthday, dear user!")
      }
      return {
        years: years,
        months: months,
        days: days
      };
    }
  </script>
</body>
</html>
```

Age Calculator

Enter your date of birth:

05-09-2004

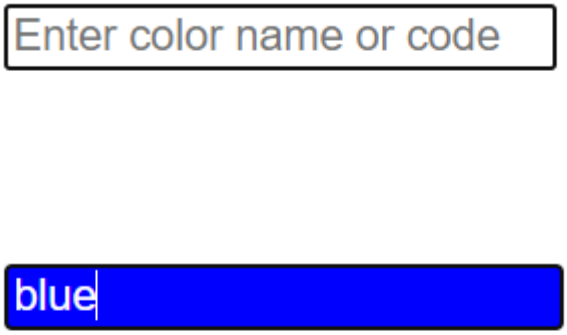
Calculate Age

Your age is: 19 years, 7 months, and 29 days.

Experiment 7: Using Angular JS Implement Input Validation

1. Write a code to change the background of the textbox based on the color mentioned by the user in the same text box.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Input Validation and Background Color Change</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
    <style>
        .error {
            background-color: red;
            color: white;
        }
    </style>
</head>

<body ng-app="validationApp">
    <div ng-controller="validationCtrl">
        <input type="text" ng-model="color" ng-style="{ 'background-color':
color }" ng-class="{ 'error': !isValidColor(color) }" placeholder="Enter color
name or code">
    </div>

    <script>
        var app = angular.module('validationApp', []);
        app.controller('validationCtrl', function($scope) {
            $scope.isValidColor = function(color) {
                if (!color) return true; // If no color entered, no error
                return /(^#[0-9A-F]{6}$)|(^#[0-9A-F]{3}$)/i.test(color); //
Regex for valid hex color codes
            };
        });
    </script>
</body>

</html>
```



2. Write a code to create a cost calculator.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Cost Calculator</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
</head>
<body ng-app="costCalculatorApp">
    <div ng-controller="costCalculatorCtrl">
        <label>Quantity:</label>
        <input type="number" ng-model="quantity">
        <label>Price per item:</label>
        <input type="number" ng-model="price">
        <p>Total Cost: {{ calculateCost() }}</p>
    </div>

    <script>
        var app = angular.module('costCalculatorApp', []);

        app.controller('costCalculatorCtrl', function($scope) {
            $scope.quantity = 0;
            $scope.price = 0;

            $scope.calculateCost = function() {
                return $scope.quantity * $scope.price;
            };
        });
    </script>
```

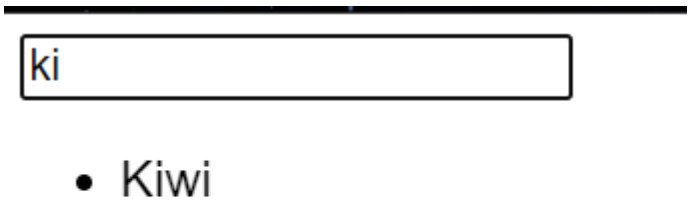
```
</body>
</html>
```

Quantity: 10 Price per item: 5

Total Cost: 50

3. Write a code to build a simple search filter functionality to display a filtered list based on the search query entered by the user.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Simple Search Filter</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
</head>
<body ng-app="searchFilterApp">
    <div ng-controller="searchFilterCtrl">
        <input type="text" ng-model="searchQuery" placeholder="Search...">
        <ul>
            <li ng-repeat="item in items | filter: searchQuery">{{ item
}}</li>
        </ul>
    </div>

    <script>
        var app = angular.module('searchFilterApp', []);

        app.controller('searchFilterCtrl', function($scope) {
            $scope.items = ['Apple', 'Banana', 'Orange', 'Grapes', 'Mango',
'Pineapple' , 'Kiwi', 'Strawberry'];
        });
    </script>
</body>
</html>
```



4. Use AngularJS Tables to perform the following.

   Display a Table

   Display contents of table with Order by Filter

   Display Table with even and odd.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>AngularJS Tables</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
    <style>
        table {
            width: 100%;
            border-collapse: collapse;
            margin-bottom: 20px;
        }

        th, td {
            padding: 8px;
            text-align: left;
            border-bottom: 1px solid #ddd;
        }

        tr:nth-child(even) {
            background-color: #f2f2f2;
        }

        th {
            background-color: #4CAF50;
            color: white;
        }
    </style>
</head>
<body ng-app="tableApp">
```

```
    <div ng-controller="tableCtrl">
        <h2>Display a Table</h2>
        <table>
            <thead>
                <tr>
                    <th>ID</th>
                    <th>Name</th>
                    <th>Age</th>
                </tr>
            </thead>
            <tbody>
                <tr ng-repeat="person in people">
                    <td>{{ person.id }}</td>
                    <td>{{ person.name }}</td>
                    <td>{{ person.age }}</td>
                </tr>
            </tbody>
        </table>

        <h2>Display contents of table with Order by Filter</h2>
        <table>
            <thead>
                <tr>
                    <th><a href="" ng-click="orderByField='id'; reverseSort =
!reverseSort">ID</a></th>
                    <th><a href="" ng-click="orderByField='name'; reverseSort
= !reverseSort">Name</a></th>
                    <th><a href="" ng-click="orderByField='age'; reverseSort =
!reverseSort">Age</a></th>
                </tr>
            </thead>
            <tbody>
                <tr ng-repeat="person in people |
orderBy:orderByField:reverseSort">
                    <td>{{ person.id }}</td>
                    <td>{{ person.name }}</td>
                    <td>{{ person.age }}</td>
                </tr>
            </tbody>
        </table>

        <h2>Display Table with even and odd</h2>
        <table>
            <thead>
                <tr>
                    <th>ID</th>
                    <th>Name</th>
                    <th>Age</th>
```

```
                </tr>
            </thead>
            <tbody>
                <tr ng-repeat="person in people" ng-class-odd="'odd'" ng-
class-even="'even'">
                    <td>{{ person.id }}</td>
                    <td>{{ person.name }}</td>
                    <td>{{ person.age }}</td>
                </tr>
            </tbody>
        </table>
    </div>

    <script>
        var app = angular.module('tableApp', []);

        app.controller('tableCtrl', function($scope) {
            $scope.people = [
                { id: 1, name: 'John', age: 25 },
                { id: 2, name: 'Alice', age: 30 },
                { id: 3, name: 'Bob', age: 22 },
                { id: 4, name: 'Eve', age: 27 },
                { id: 5, name: 'Carol', age: 35 }
            ];

            $scope.orderByField = 'id';
            $scope.reverseSort = false;
        });
    </script>
</body>
</html>
```

Display a Table

ID	Name	Age
1	John	25
2	Alice	30
3	Bob	22
4	Eve	27
5	Carol	35

Display contents of table with Order by Filter

ID	Name	Age
1	John	25
2	Alice	30
3	Bob	22
4	Eve	27
5	Carol	35

Display Table with even and odd

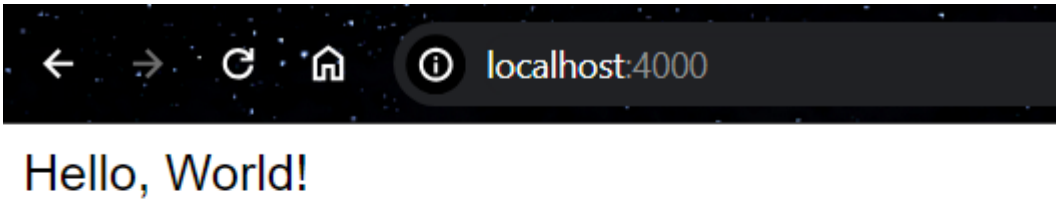
ID	Name	Age
1	John	25
2	Alice	30
3	Bob	22
4	Eve	27
5	Carol	35

Experiment 8: NodeJS basic exercises

1. Create a simple “Hello, World!” server using Node.js and Express.

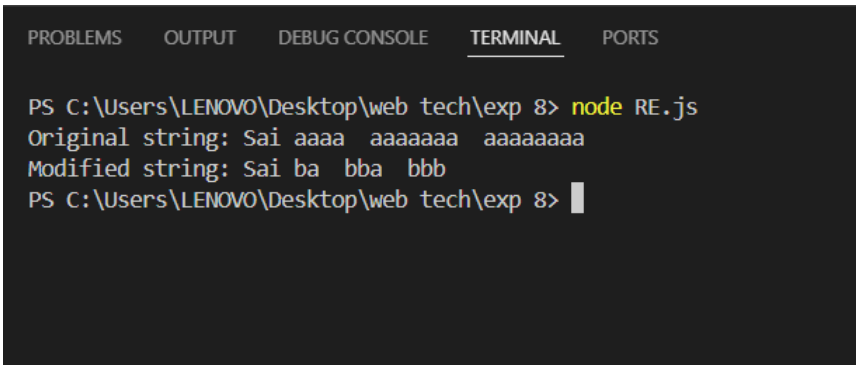
```
const express = require('express');
const app = express();
const port = 4000;
app.get('/', (req, res) => {
  res.send('Hello, World!');
});
app.listen(port, () => {
  console.log(`Server is running on http://localhost:${port}`);
});
```

```
PS C:\Users\LENOVO\Desktop\web tech\exp 8> node HelloWorld.js
Server is running on http://localhost:4000
```



2. Write a node.js program to replace two or more a's with the letter b on the given string using Regular Expression.

```
const inputString = "Sai aaaa  aaaaaaa  aaaaaaaa";
const regex = /a{2,3}/g;
const replacedString = inputString.replace(regex, 'b');

console.log("Original string:", inputString);
console.log("Modified string:", replacedString);
```



3. Create a basic calculator that can perform arithmetic operations (addition, subtraction, multiplication, and division) through HTTP requests.

```
const express = require('express');
const app = express();
app.get('/calculate', (req, res) => {

    const { operation, num1, num2 } = req.query;
    const operand1 = parseFloat(num1);
    const operand2 = parseFloat(num2);
    let result;
    switch(operation) {
        case 'add':
            result = operand1 + operand2;
            break;
        case 'subtract':
            result = operand1 - operand2;
            break;
        case 'multiply':
            result = operand1 * operand2;
            break;
        case 'divide':
            if (operand2 === 0) {
                return res.status(400).send('Cannot divide by zero');
            }
            result = operand1 / operand2;
            break;
        default:
            return res.status(400).send('Invalid operation');
    }
    res.status(200).send(`Result: ${result}`);
});
const port = 3000;
app.listen(port, () => {
```

```
    console.log(`Server running at http://localhost:${port}/`);
});
PS C:\Users\LENOVO\Desktop\web tech\exp 8> node Calculator.js
Server running at http://localhost:3000/
```

4. Write a node.js code to iterate over the given array.

```
const myArray = [1, 2, 3, 4, 5 , 6 , 7];
console.log("Using forEach loop:");
myArray.forEach((element, index) => {
  console.log(`Element at index ${index}: ${element}`);
});
console.log("\nUsing for...of loop:");
for (const [index, element] of myArray.entries()) {
  console.log(`Element at index ${index}: ${element}`);
}
console.log("\nUsing for loop:");
for (let i = 0; i < myArray.length; i++) {
  console.log(`Element at index ${i}: ${myArray[i]}`);
}
```

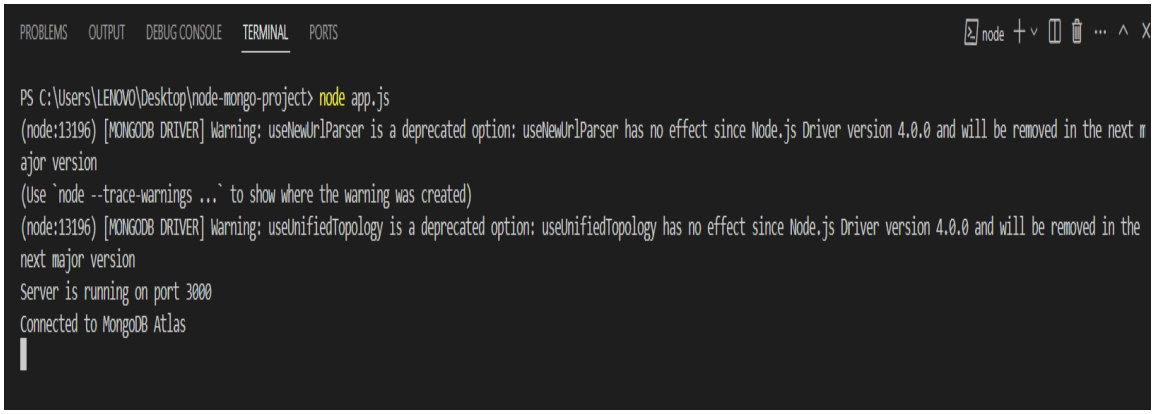
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\LENOVO\Desktop\web tech\exp 8> node Array.js
Using forEach loop:
Element at index 0: 1
Element at index 1: 2
Element at index 2: 3
Element at index 3: 4
Element at index 4: 5
Element at index 5: 6
Element at index 6: 7

Using for...of loop:
Element at index 0: 1
Element at index 1: 2
Element at index 2: 3
Element at index 3: 4
Element at index 4: 5
Element at index 5: 6
Element at index 6: 7

Using for loop:
Element at index 0: 1
Element at index 1: 2
Element at index 2: 3
Element at index 3: 4
Element at index 4: 5
Element at index 5: 6
Element at index 6: 7
PS C:\Users\LENOVO\Desktop\web tech\exp 8>
```

Experiment 9: Working with Data

1. Create a NodeJS application to connect to a MongoDB database.

```
// Import dependencies
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');

// Create Express app
const app = express();

// Set up middleware
app.use(bodyParser.urlencoded({ extended: true }));
app.set('view engine', 'ejs');

// Connect to MongoDB Atlas
mongoose.connect('mongodb+srv://Sai:Sairanjan123@cluster0.81suskm.mongodb.net/
?retryWrites=true&w=majority&appName=Cluster0', {
  useNewUrlParser: true,
  useUnifiedTopology: true
}).then(() => {
  console.log('Connected to MongoDB Atlas');
}).catch((err) => {
```

```
  console.error('Error connecting to MongoDB Atlas:', err.message);
});

// Define Mongoose schema and model
const userSchema = new mongoose.Schema({
  username: String,
  email: String
});

const User = mongoose.model('User', userSchema);

// Routes
app.get('/', (req, res) => {
  res.render('index');
});

app.post('/users', async (req, res) => {
  const { username, email } = req.body;
  try {
    const newUser = new User({ username, email });
    await newUser.save();
    res.redirect('/users');
  } catch (err) {
    console.error('Error saving user:', err.message);
    res.redirect('/');
  }
});

app.get('/users', async (req, res) => {
  try {
    const users = await User.find({});
    res.render('users', { users });
  } catch (err) {
    console.error('Error finding users:', err.message);
    res.redirect('/');
  }
});

// Start the server
const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
```


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
node
PS C:\Users\LENOVO\Desktop\node-mongo-project> node app.js
(node:13196) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:13196) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server is running on port 3000
Connected to MongoDB Atlas