Internet Technologies

Name=Anurag singh rana

Roll.no=10828

Course=Bsc(Hons)C.S

Section=A

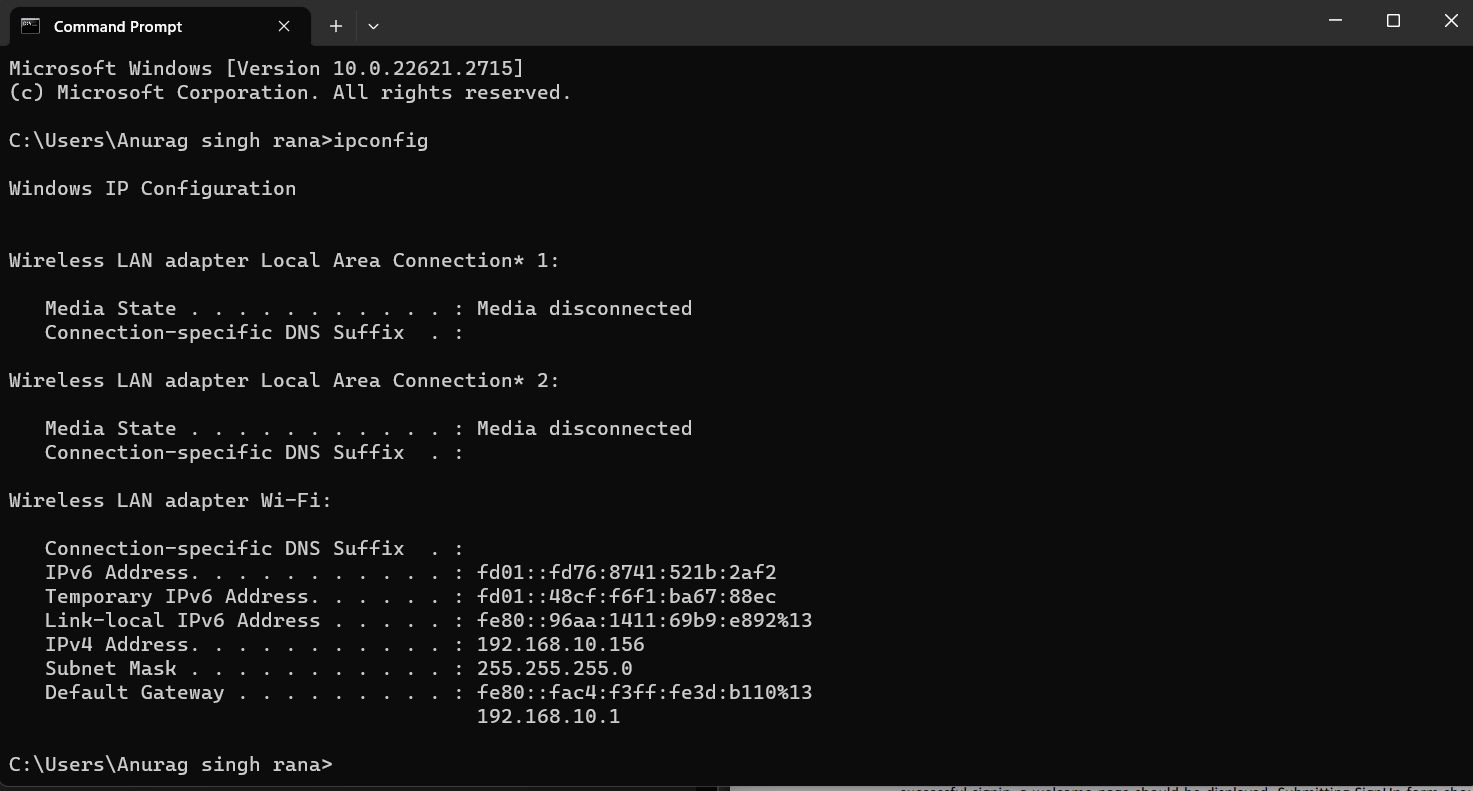
Sem=5th

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Practicals** | **Page No.** |
| **1.** | Display your systems IP Address, Subnet mask using ipconfig, and find out the network address and the maximum number of systems possible on your network and range of IP addresses available to these systems. |  |
| **2.** | With help of ping, check if you are connected to other systems of your network and find the route to connect to that system using tracert. List all the processes which are using ports for TCP protocol. |  |
| **3.** | Create an HTML page that shows information about you, your course, hobbies, address, and your plans. Use CSS for styling of HTML page so that looks nice. |  |
| **4.** | Create an HTML page with the sole purpose to show multiplication tables of 2 to 10 (row-wise) created by JavaScript. Initially, the page is blank. With help of setInterval function print a row every 5 seconds in different colors and increasing font size. |  |
| **5.** | Create an HTML page with a paragraph written on it and under which 9 buttons are placed in a 3X3 grid. The first row is for buttons labeled with colors names Red, Green, and Blue, the second row with numbers 10, 20, 30, and the third row with different font names. Click event of each of the buttons should make the appropriate change in the style of paragraph. |  |
| **6.** | Create a form that takes data about a pet. The form must be well designed and should accept the pet’s name, age , weight,type , and what it likes most. At the submission of this form create a Pet object in JavaScript filled with these values and log that object and equivalent JSON on the console. |  |
| **7.** | Store JSON data of few pets that you created in previous practical in a JSON file (copy from console output of previous program to a .json file). Using AJAX, load data from the file and display it in a presentable way using HTML and CSS. |  |
| **8.** | Create a plain HTML page for B.Sc. Hons CS course, mentioning details like fee, eligibility criteria, papers with names and credits, and future possibilities after the course. A button for styling should be there at bottom of the page. On clicking on this button JavaScript should redesign the complete page using jQuery in a nice presentable way. |  |
| **9.** | Create an HTML page for an image gallery which shows the use of BOOTSTRAP to rearrange and resize its contents on resizing the browser. |  |
| **10.** | Create an HTTP server using Node.js which handles requests on port 10000 or a free port beyond 10000. Modify the server in such a way that opening localhost:10000 will display “Hello World, This is my Node.js server” on browser |  |
| **11.** | Create index.html file containing two forms for SignIn and SignUp. Submitting SignIn form should search for credentials in mysql database using server created in previous practical. On successful signin, a welcome page should be displayed. Submitting SignUp form should insert new entry for credentials in mysql database using server created in previous practical. On successful signup, user should be returned back to index.html. |  |
| **12.** | Clock |  |
| **13.** | stack |  |
| **14.** | Mouse image resize |  |
| **15.** | Image animation |  |
| **16.** | Add data on table |  |

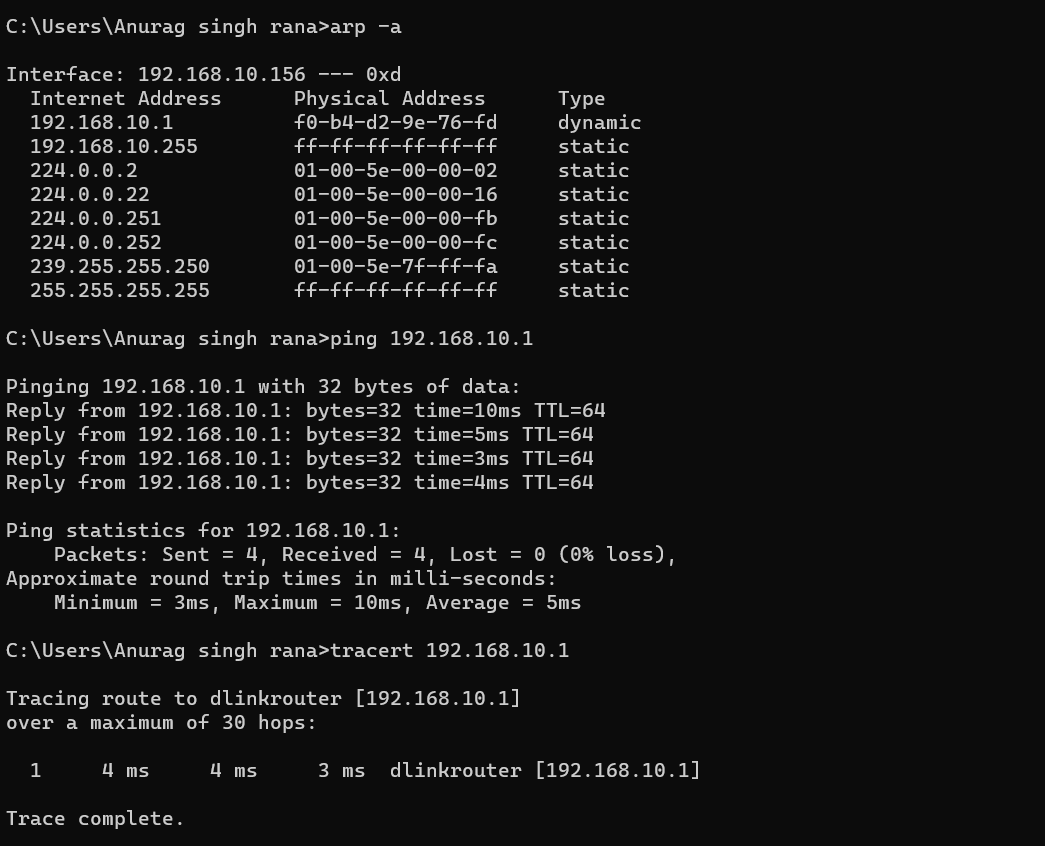
Practicals:

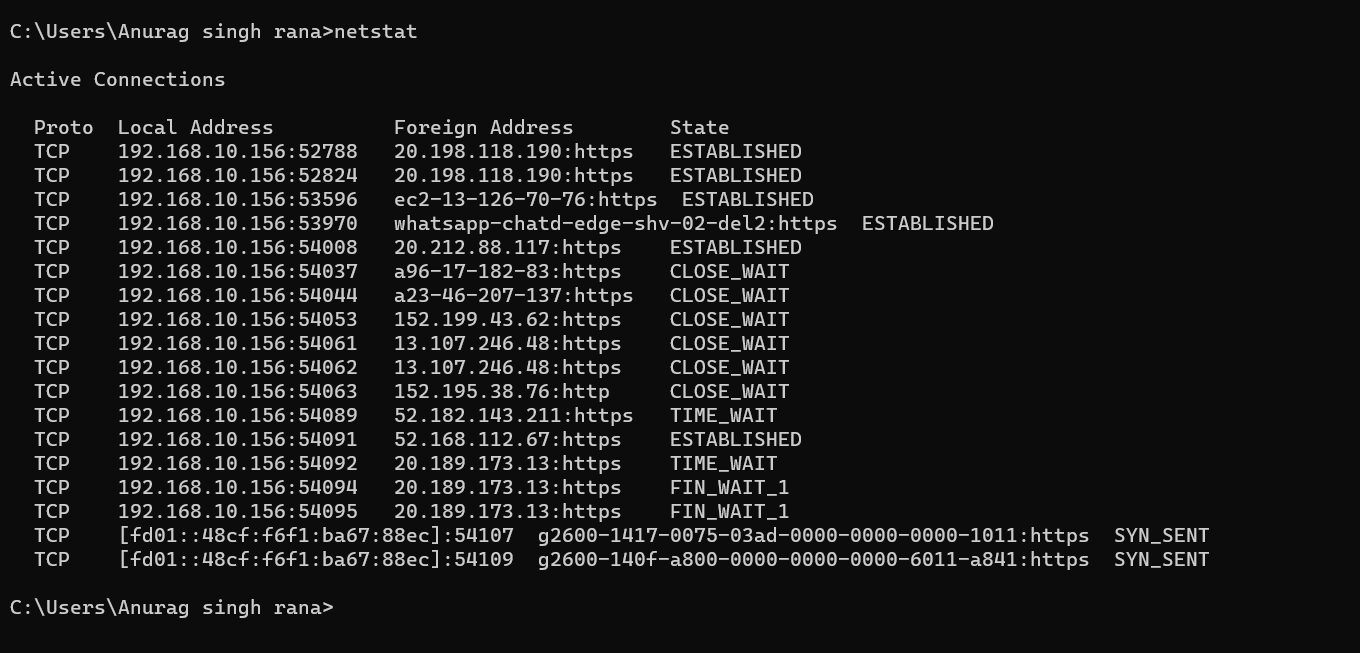
Que 1: Display your systems IP Address, Subnet mask using ipconfig, and find out the network address and the maximum number of systems possible on your network and range of IP addresses available to these systems.



Subnet number=192.168.10.0, broadcast =192.168.10.255

Que 2: With help of ping, check if you are connected to other systems of your network and find the route to connect to that system using tracert. List all the processes which are using ports for TCP protocol.





Que 3: Create an HTML page that shows information about you, your course, hobbies, address, and your plans. Use CSS for styling of HTML page so that looks nice.

<!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>About me</title>

    <link rel="stylesheet" href="./main.css" />

  </head>

  <body>

    <header class="tophead">

      <p class="tophead-intro">Hi I'm</p>

      <h1 class="tophead-heading">Anurag singh rana</h1>

    </header>

    <section class="about-section">

      <h1>About me</h1>

      <p>

        I'm a final year student of Bsc (Hons.) Computer Science at Keshav Mahavidyalaya College, University of Delhi.

      </p>

    </section>

    <section class="hobbies-section">

      <h1>My Hobbies</h1>

      <p>I love listening to music, watching tv series and reading. Playing volleyball </p>

    </section>

    <section class="course-section">

      <h1>My Course</h1>

      <p>

        <br>Things that I have learnt so far:

        <div class="course-list">

            <ul>

                <li>C++</li>

                <li>Java </li>

                <li>DSA</li>

                <li>Python</li>

                <li>Computer Networks</li>

                <li>DBMS</li>

                <li>operating system</li>

            </ul>

        </div>

    </p>

    </section>

    <section class="plan-section">

      <h1>My Plans</h1>

      <p>After the graduation I want to prepare for govt. exams</p>

    </section>

    <footer class="content-footer">

      <p>Say hi to me:</p>

      <ul class="social">

        <li>

          <a class="css-is-deranged" href="#"

            >Github</a

          >

        </li>

        <li>

          <a class="css-is-deranged" href="#"

            >Twitter</a

          >

        </li>

      </ul>

    </footer>

  </body>

</html>

Css code:

  .tophead-heading {

    margin-top: -0.2em;

    font-family: "Open Sans", "Helvetica Neue", sans-serif;

    font-weight: bold;

    font-size: 3em;

    letter-spacing: -0.02em;

    text-transform: uppercase;

  }

  .about-section > p,

  .hobbies > p,

  .content-footer > p {

    font-weight: 300;

    letter-spacing: 0.05em;

  }

  h2 {

    font-family: "Gentium Book Basic", Georgia, serif;

    font-size: 1.2em;

    font-weight: bold;

  }

  .about-section,

  .hobbies-section,

  .course-section,

  .plan-section {

    max-width: 38em;

    margin-left: auto;

    margin-right: auto;

    margin-top: 2em;

  }

  .content-footer {

    margin: 50px 0;

    padding: 20px 0;

    text-align: center;

    background-color: #73a8c2;

    text-decoration: none;

  }

  .social > li {

    display: inline-block;

    margin: 0 5px;

  }

  .content-footer > p {

    color: #f7ede1;

  }

  a {

    font-weight: bold;

    text-decoration: none;

    color: #f1c863;

  }

  @media only screen and (max-width: 500px) {

    .tophead {

      padding: 3em 0;

    }

    .tophead-heading {

      font-size: 3em;

    }

    .content-footer {

      padding: 2em 2.5em;

    }

  }

Output:



Que 4: Create an HTML page with the sole purpose to show multiplication tables of 2 to 10 (row-wise) created by JavaScript. Initially, the page is blank. With help of setInterval function print a row every 5 seconds in different colors and increasing font size.

<!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>Multiplication Tables</title>

    <link rel="stylesheet" href="./style.css">

  </head>

  <body>

    <main id="main">

    </main>

    <script src="./script.js"></script>

  </body>

</html>

Css code:

body {

    font-family: monospace;

  }

  p {

    margin: 0;

  }

Js code:

const mainEl = document.getElementById('main');

var i = 2;

var j = 1;

var t = 1;

var colors = ['#E74C3C', '#3498DB', '#27AE60', '#8E44AD', '#000000'];

setInterval(() => {

  if (j > 10) {

    i++;

    j = 1;

  }

  if (i > 10) return;

  const child = document.createElement('p');

  child.innerHTML = `${i} \* ${j} = ${i \* j}`;

  child.style.color = colors[(j - 1) % 5];

  child.style.fontSize = `${t}rem`;

  mainEl.appendChild(child);

  window.scrollTo(0, document.body.scrollHeight);

  j++;

  t += 0.05;

}, 5e3);

Output:

Que 5: Create an HTML page with a paragraph written on it and under which 9 buttons are placed in a 3X3 grid. The first row is for buttons labeled with colors names Red, Green, and Blue, the second row with numbers 10, 20, 30, and the third row with different font names. Click event of each of the buttons should make the appropriate change in the style of paragraph.

<!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>Paragraph Styling</title>

    <link rel="stylesheet" href="./style.css">

  </head>

  <body>

    <main>

      <div>

        <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Nihil sapiente hic provident, quo quae deleniti

          obcaecati corrupti doloremque. Ipsum provident a, nesciunt necessitatibus rerum recusandae illo ipsam

          cupiditate? Itaque nemo voluptatibus deleniti velit labore quibusdam officiis laborum maiores illo

          dignissimos. Veritatis illum adipisci nemo ea eligendi nulla. Nemo aliquam impedit cupiditate odit recusandae

          eos, autem magnam placeat perspiciatis ipsam aliquid repudiandae et provident, ullam magni rem deleniti

          repellendus nihil asperiores repellat quibusdam quae inventore minima. Aperiam, natus fuga cumque commodi

          quasi non.</p>

      </div>

      <div class="grid">

        <button id="c-red">Red</button>

        <button id="c-green">Green</button>

        <button id="c-blue">Blue</button>

        <button id="s-10">10</button>

        <button id="s-20">20</button>

        <button id="s-30">30</button>

        <button id="f-mono">Monospace</button>

        <button id="f-sans">Sans-Serif</button>

        <button id="f-serif">Serif</button>

      </div>

    </main>

    <script src="./script.js"></script>

  </body>

</html>

Css code:

body, html {

    margin: 0;

    padding: 0;

  }

  body {

    background-color: #000;

    color: #fff;

    font-size: 1.2rem;

    font-family: monospace;

  }

  main {

    width: 60%;

    margin: 0 auto;

    display: flex;

    flex-direction: column;

    align-items: center;

    justify-content: center;

    position: absolute;

    top:0;

    bottom:0;

    right:0;

    left:0;

  }

  p {

    padding: 0 2.85em;

  }

  .grid {

    width: 60%;

    margin: 0 auto;

    display: grid;

    grid-template-columns: repeat(3, 1fr);

  }

  button {

    border: none;

    margin: 1em auto;

    width: 8em;

    height: 2em;

    transition-duration: 500ms;

    cursor: pointer;

    font-family: monospace;

  }

  button:hover {

    background-color: #ccc;

  }

Js code:

const paragraph = document.querySelector('p');

/\*\*

 \* set font color

 \*/

document.getElementById('c-red').addEventListener('click', () => {

  paragraph.style.color = 'red';

});

document.getElementById('c-green').addEventListener('click', () => {

  paragraph.style.color = 'green';

});

document.getElementById('c-blue').addEventListener('click', () => {

  paragraph.style.color = 'blue';

});

/\*\*

 \* set font size

 \*/

document.getElementById('s-10').addEventListener('click', () => {

  paragraph.style.fontSize = '10px';

});

document.getElementById('s-20').addEventListener('click', () => {

  paragraph.style.fontSize = '20px';

});

document.getElementById('s-30').addEventListener('click', () => {

  paragraph.style.fontSize = '30px';

});

/\*\*

 \* set font family

 \*/

document.getElementById('f-mono').addEventListener('click', () => {

  paragraph.style.fontFamily = 'monospace';

});

document.getElementById('f-sans').addEventListener('click', () => {

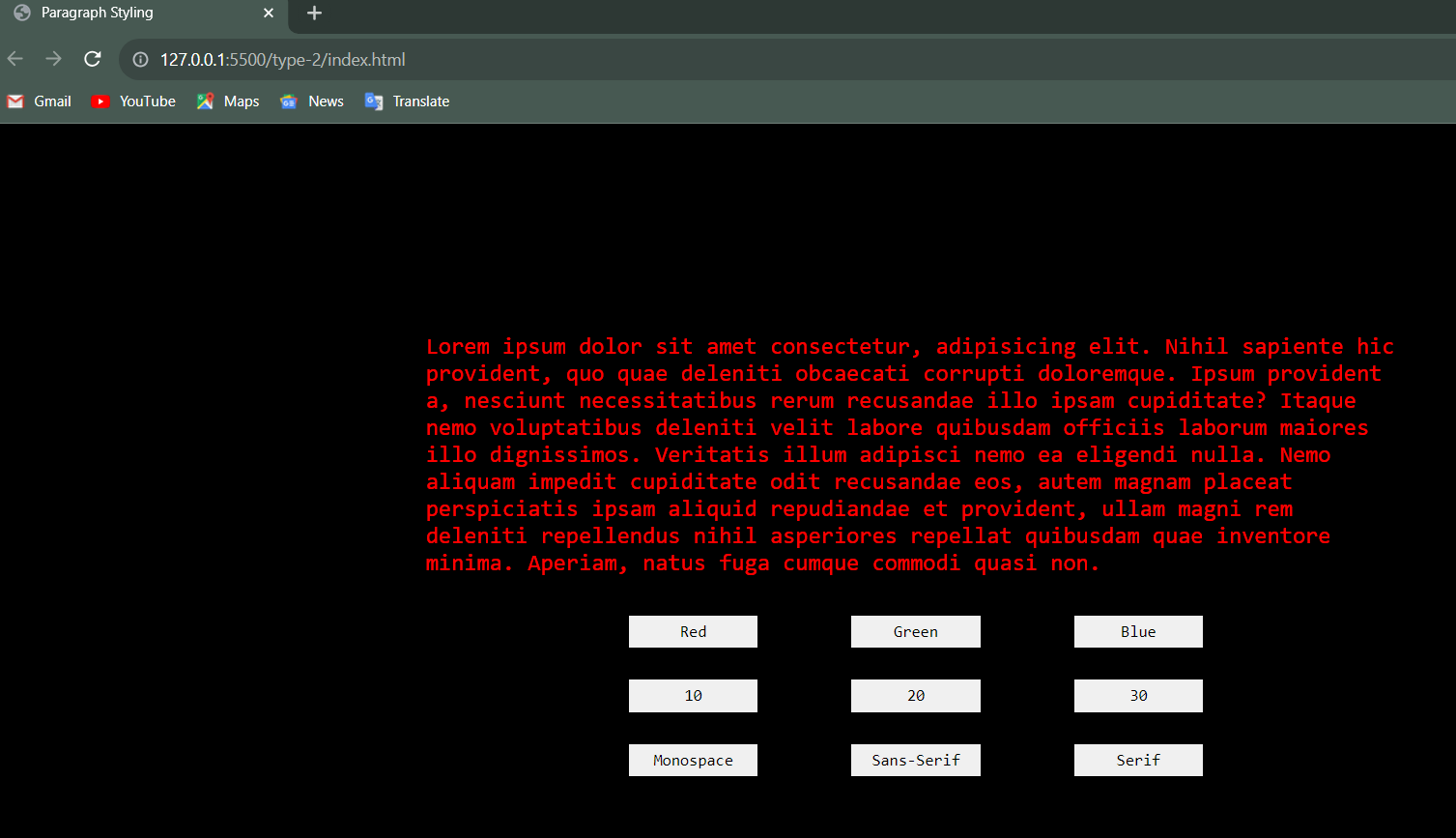
  paragraph.style.fontFamily = 'sans-serif';

});

document.getElementById('f-serif').addEventListener('click', () => {

  paragraph.style.fontFamily = 'serif';

});

Output: 

Que 6: Create a form that takes data about a pet. The form must be well designed and should accept the , and what it likes most. At the submission of this form create a Pet object in JavaScript filled with these values and log that object and equivalent JSON on the console.

<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8">

    <title>Practical 6</title>

    <style type="text/css">

        .container{

            width: 60%;

            margin: auto;

            border: 1px solid black;

            border-radius: 8px;

            padding: 50px;

        }

        .btn-submit{

            border-radius: 5px;

            color: white;

            background: greenyellow;

            font-weight: bold;

            font-size: 1rem;

            margin: 20px;

        }

        @media(width<=575){

            .container{

                width: 84%;

            }

        }

    </style>

</head>

<body>

    <div class="container">

        <h1>Pet's Information</h1>

        <hr>

        <label for="name">Pet's Name: </label>

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

        <label for="age">Age: </label>

        <input type="number" name="age">

        <label for="weight">Weight: </label>

        <input type="number" name="weight" class=""><br><br>

        <label for="type">Pet type: </label>

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

        <label for="likes">Likes: </label>

        <input type="text" name="likes"><br>

        <button class="btn-submit" onclick="display()">Submit</button>

    </div>

    <script type="text/javascript">

        function display(){

            // event.preventDafault();

            var pet = {};

            var input\_fields = document.getElementsByTagName('input');

            for (var i = 0; i < input\_fields.length; i++) {

                pet[input\_fields[i].name] = input\_fields[i].value;

            }

            console.log(pet);

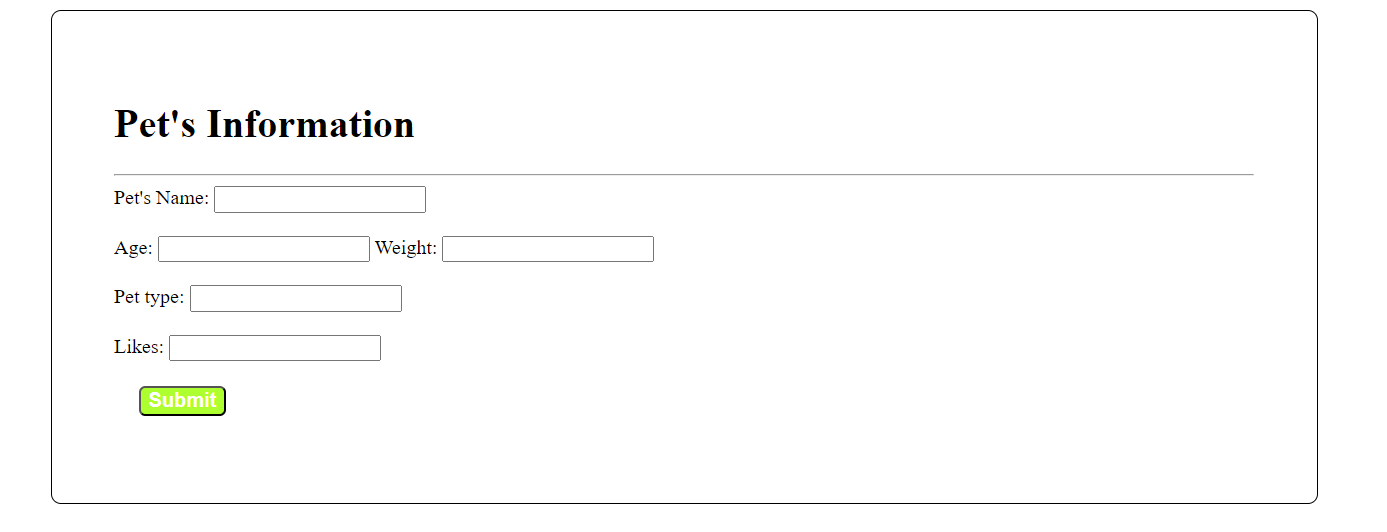
        }

    </script>

</body>

</html>

Output:



Que 7: Store JSON data of few pets that you created in previous practical in a JSON file (copy from console output of previous program to a .json file). Using AJAX, load data from the file and display it in a presentable way using HTML and CSS

<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8">

    <title>Practical 7</title>

    <style type="text/css">

        #pet-data{

            border: 1px solid black;

            border-radius: 10px;

            border-collapse: collapse;

        }

        td{

            border: 1px solid black;

            border-collapse: collapse;

        }

        #btn-fetch{

            margin-top: 20px;

            font-size: 24px;

            font-weight: bold;

            background-color: black;

            color: white;

            border-radius: 8px;

        }

    </style>

</head>

<body>

    <div id="content">

    </div>

    <button id="btn-fetch">Fetch Data</button>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <script type="text/javascript">

        var btnFetch = document.getElementById('btn-fetch');

        var content = document.getElementById('content');

        btnFetch.addEventListener('click', ()=>{

            const xhr = new XMLHttpRequest();

            xhr.open("GET",'/pet.json',true);

            xhr.onload = ()=>{

                console.log(xhr.responseText);

                renderHtml(JSON.parse(xhr.responseText));

            }

            xhr.send();

        });

        function renderHtml(data){

            content.innerHTML = "";

            for (var i = 0; i <= data.length; i++) {

                let p = document.createElement('p');

                let htmlpart = "";

                htmlpart += data[i].name+" is a "+data[i].type+" with age "+data[i].age+" years and weight "+data[i].weight+"kg and likes "+data[i].likes;

                p.innerHTML = htmlpart;

                content.append(p);

                // console.log(tr);

                htmlpart="";

            }

        }

    </script>

</body>

</html>

Json file:

[

    {

        "name":"Pluto",

        "age":3,

        "weight":12,

        "type":"Pavellion",

        "likes":"eating, playing with ball"

    },

    {

        "name":"Hulk",

        "age":4,

        "weight":22,

        "type":"German Sefford",

        "likes":"Biting, eating flesh"

    },

    {

        "name":"Jerry",

        "age":2.5,

        "weight":8,

        "type":"cat",

        "likes":"sleeping"

    },

    {

        "name":"Tom",

        "age":0.5,

        "weight":0.7,

        "type":"Mouse",

        "likes":"running, eating cheese"

    },

    {

        "name":"Chiku",

        "age":1,

        "weight":1.2,

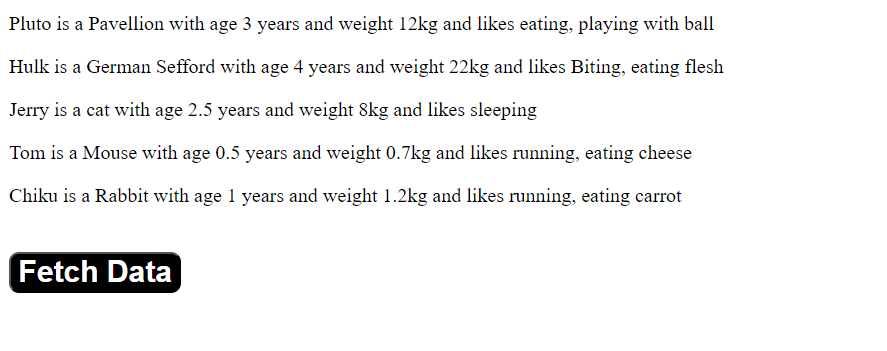
        "type":"Rabbit",

        "likes":"running, eating carrot"

    }

]

Output:



Que 8: Create a plain HTML page for B.Sc. Hons CS course, mentioning details like fee, eligibility criteria, papers with names and credits, and future possibilities after the course. A button for styling should be there at bottom of the page. On clicking on this button JavaScript should redesign the complete page using jQuery in a nice presentable way.

<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8">

    <title>Practical 8</title>

    <style type="text/css">

        .container {

            width: 70%;

            margin: auto;

            align-items: center;

            background-color: #D9CAB3;

            padding-bottom: 10px;

        }

        .info-table {

            width: 80%;

            margin: auto;

            border: 3px solid black;

            border-collapse: collapse;

            margin-top: 2%;

            margin-bottom: 2%;

        }

        .table-row {

            width: 100%;

            margin: auto;

        }

        .table-data {

            width: 50%;

            border: 2px solid white;

            border-collapse: collapse;

        }

    </style>

</head>

<body>

    <div>

        <h1 class="heading">Bsc Hons Computer Science</h1>

        <table>

            <tr>

                <td>Fees</td>

                <td>24240</td>

            </tr>

            <tr>

                <td>Eligibility Criteria</td>

                <td>10-12 Pass</td>

            </tr>

            <tr>

                <td>Subjects and credit scores</td>

                <td>

                    <table>

                        <tr>

                            <th>Subject</th>

                            <th>Credit score</th>

                        </tr>

                        <tr>

                            <td>IT</td>

                            <td>6</td>

                        </tr>

                        <tr>

                            <td>Toc</td>

                            <td>6</td>

                        </tr>

                        <tr>

                            <td>DAV</td>

                            <td>4</td>

                        </tr>

                        <tr>

                            <td>DIP/Micro</td>

                            <td>4</td>

                        </tr>

                    </table>

                </td>

            </tr>

            <tr>

                <td>Future Opportunities</td>

                <td>After doing this course we have very good opportunities in the feild of IT.</td>

            </tr>

        </table>

    </div>

    <button id="btn-style">

        Style Page

    </button>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <script type="text/javascript">

        $(document).ready(function () {

            $('#btn-style').click(function () {

                $("div").addClass('container');

                $("table").addClass('info-table');

                $("tr").addClass('table-row');

                $("td").addClass('table-data');

                $(".heading").css({

                    "textAlign": 'center'

                });

            });

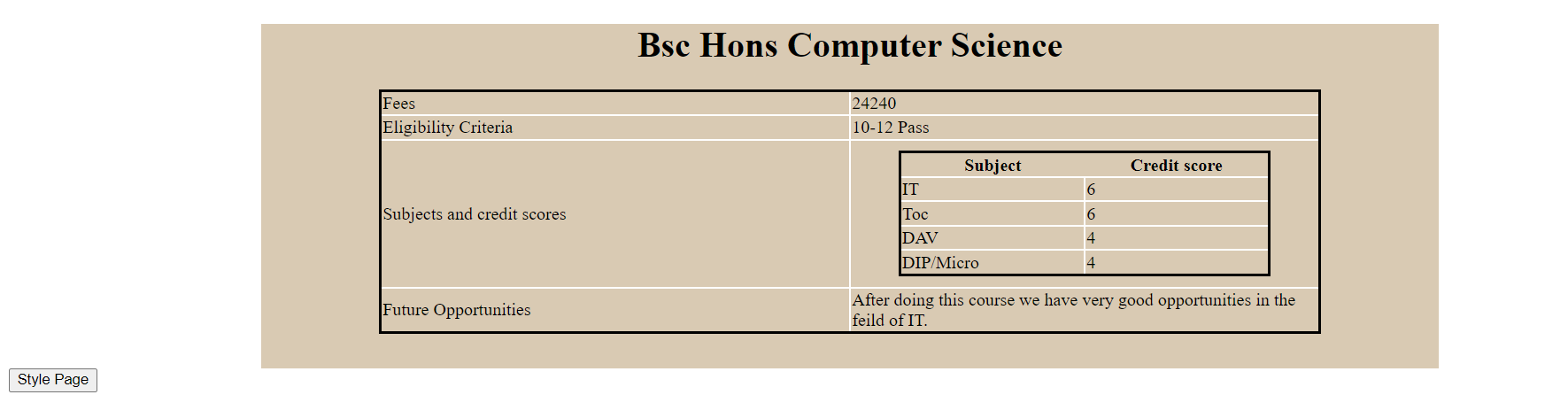
        });

    </script>

</body>

</html>

Output:



Que 9: Create an HTML page for an image gallery which shows the use of BOOTSTRAP to rearrange and resize its contents on resizing the browser.

<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

    <title>Practical 9</title>

    <style type="text/css">

        /\*img{

            margin: 20px;

        }\*/

    </style>

</head>

<body>

    <div class="jumbotron text-center">

      <h1>IMAGE GALLERY</h1>

      <p>Responsive Image gallery using bootstrap.</p>

    </div>

    <div class="container">

        <img class="col-sm-4" src="https://picsum.photos/200/"></img>

        <img class="col-sm-4" src="https://picsum.photos/200/"></img>

        <img class="col-sm-4" src="https://picsum.photos/200/"></img>

        <img class="col-sm-4" src="https://picsum.photos/200/"></img>

        <img class="col-sm-4" src="https://picsum.photos/200/"></img>

        <img class="col-sm-4" src="https://picsum.photos/200/"></img>

    </div>

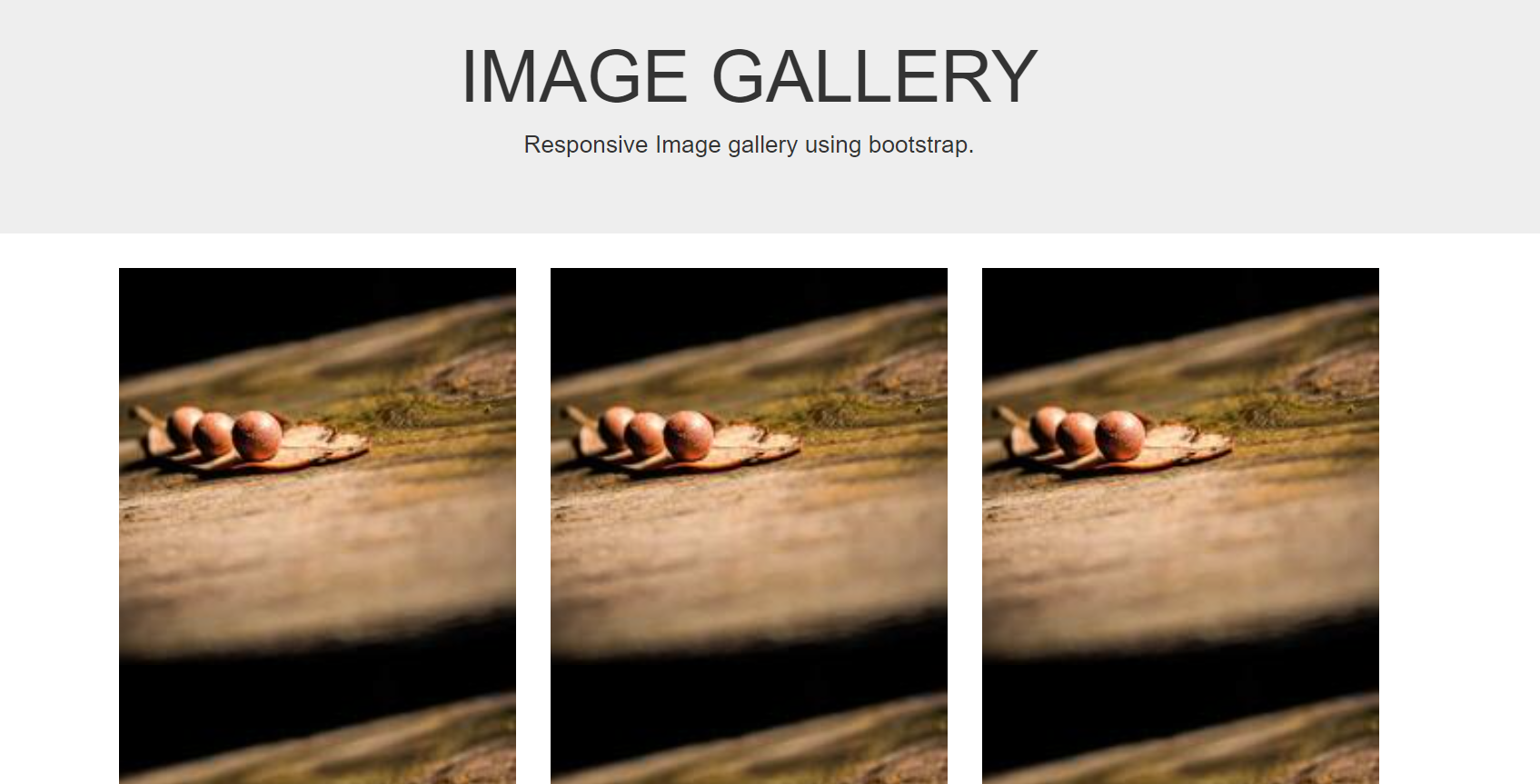
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

</body>

</html>

Output:



Que 10: Create an HTTP server using Node.js which handles requests on port 10000 or a free port beyond 10000. Modify the server in such a way that opening localhost:10000 will display “hello world,This is my Node.js server” on browser.

var http = require('http');

const server = http.createServer((request,response)=>{

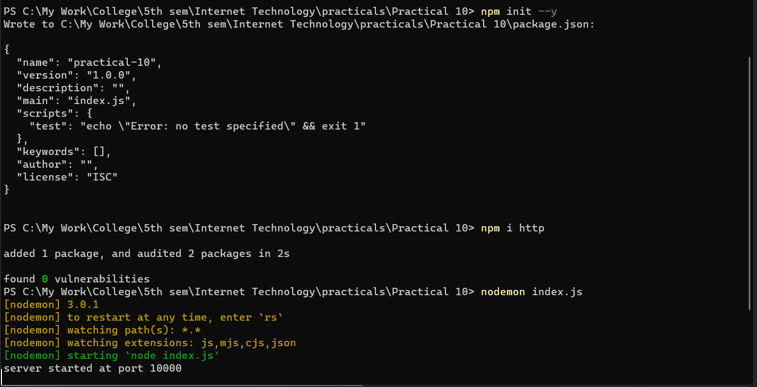
    response.write("Hello World, This is my Node.js server");

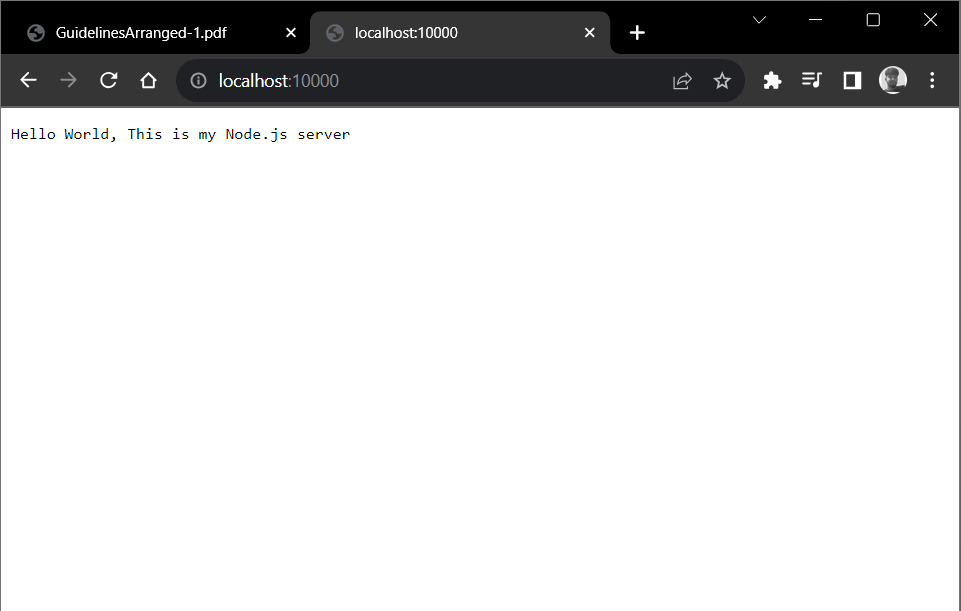
    response.end();

})

server.listen(10000,()=>console.log('server started at port 10000'));

output:





Que 11: Create index.html file containing two forms for SignIn and SignUp. Submitting SignIn form should search for credentials in mysql database using server created in previous practical. On successful signin, a welcome page should be displayed. Submitting SignUp form should insert new entry for credentials in mysql database using server created in previous practical. On successful signup, user should be returned back to index.html.

Before writing code first build packages using command prompt

Run command prompt :

**npm init –y**

**npm install http**

**npm install fs**

**npm install mysql**

**npm install querrystring**

now make login.html, signup.html, server.js and db.js file

**login.html:**

<html>

<head>

<title>Login</title>

    <style>

.container{

    border: 2px solid black;

    width: fit-content;

    height: fit-content;

    margin: auto;

    padding: 20px;

    text-align: center;

    border-radius: 30px;

    background-color: rgb(241, 241, 211);

}

form div{

    display: flex;

    justify-content: space-between;

    gap: 10px;

    /\* border-bottom: 2px solid black; \*/

    padding: 10px;

    margin: 10px auto;

    font-weight: bolder;

    font-size: 20px;

}

form button{

    border: 2px solid black;

    font-size: 20px;

    border-radius: 10px;

    background-color: green;

    margin: auto;

}

.btn-container a{

    border: 2px solid black;

    font-size: 20px;

    font-weight: bolder;

    border-radius: 10px;

    padding: 10px;

    margin: 10px;

    background-color: aqua;

    text-decoration: none;

}

    </style>

</head>

<body>

    <div class="container">

        <h1>Login</h1>

        <form action="/loginpost" method="post">

            <div>

                <label for="userid">User Id  </label>

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

            </div>

            <div>

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

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

            </div>

            <div>

                <button type="submit">Login</button>

            </div>

        </form>

        <div class="btn-container">

            <!-- <a href="/login">Login</a> -->

            <a href="/signup">SignUp</a>

        </div>

    </div>

</body>

</html>

**signup.html :**

<html>

<head>

    <title>Sign-up</title>

    <style>

.container{

    border: 2px solid black;

    width: fit-content;

    height: fit-content;

    margin: auto;

    padding: 20px;

    text-align: center;

    border-radius: 30px;

    background-color: rgb(241, 241, 211);

}

form div{

    display: flex;

    justify-content: space-between;

    gap: 10px;

    /\* border-bottom: 2px solid black; \*/

    padding: 10px;

    margin: 10px auto;

    font-weight: bolder;

    font-size: 20px;

}

form button{

    border: 2px solid black;

    font-size: 20px;

    border-radius: 10px;

    background-color: green;

    margin: auto;

}

.btn-container a{

    border: 2px solid black;

    font-size: 20px;

    font-weight: bolder;

    border-radius: 10px;

    padding: 10px;

    margin: 10px;

    background-color: aqua;

    text-decoration: none;

}

    </style>

</head>

<body>

    <div class="container">

        <h1>Sign Up</h1>

        <form action="/signuppost" method="post">

            <div>

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

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

            </div>

            <div>

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

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

            </div>

            <div>

                <label for="mobile">Mobile No.  </label>

                <input type="number" name="mobile" id="mobile" required>

            </div>

            <div>

                <label for="userid">UserID  </label>

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

            </div>

            <div>

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

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

            </div>

            <div>

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

            </div>

        </form>

        <div class="btn-container">

            <a href="/login">Login</a>

            <!-- <a href="/singup">SignUp</a> -->

        </div>

    </div>

</body>

</html>

**db.js**

const mysql = require('mysql')

const con = mysql.createConnection({

    host: 'localhost',

    user: 'root',

    port: 3306,

    password: '' ,

    // database: 'students'

});

con.connect((err) => {

    if (err) {

        console.error('Error connecting to the database:', err);

        return;

    }

    const createDatabaseQuery = `CREATE DATABASE IF NOT EXISTS students`;

    con.query(createDatabaseQuery, (err, result) => {

      if (err) {

        console.error('Error creating database:', err);

      } else {

        console.log('Database created successfully');

        con.changeUser({ database: `students` }, (err) => {

          if (err) {

            console.error('Error switching to the new database:', err);

            return con.end();

          }

          createTables();

        })

      }

    });

});

const createTables = ()=>{

    const credentials = `CREATE TABLE IF NOT EXISTS credentials(

        userid varchar(40) NOT NULL PRIMARY KEY,

        name varchar(30) NOT NULL,

        email varchar(30) NOT NULL,

        mobile bigint(12) NOT NULL,

        password varchar(30) NOT NULL

    )`;

    con.query(credentials,(err,result)=>{

        if(err){

            return console.log("create table error ",err);

        }

        console.log("Database and table created successfully...")

    })

}

module.exports = con;

**server.js**

const http = require('http');

const con = require('./db');

const fs = require('fs');

const url = require('url');

const querystring = require('querystring');

let login ='';

let signup ='';

fs.readFile('login.html',(err,data)=>{

    if(err){

        return console.log("File Read Error : ",err);

    }

    login = data;

});

fs.readFile('signup.html',(err,data)=>{

    if(err){

        return console.log("File Read Error : ",err);

    }

    signup = data;

});

const server = http.createServer((req,res)=>{

    const parsedUrl = url.parse(req.url, true);

    const path = parsedUrl.pathname;

    res.writeHead(200, {'Content-Type': 'text/html'});

    if(req.method === 'GET'){

        if (path === '/' || path ==='/login'){

            res.write(login);

            res.end();

        }

        else if (path === '/signup') {

            res.end(signup);

        }

        else {

            res.writeHead(404, {'Content-Type': 'text/html'});

            res.end('404 Not Found');

        }

    }

    else if(req.method === 'POST'){

        let body = '';

        req.on('data', (chunk) => {

            body += chunk.toString();

        });

        req.on('end', () => {

            const formData = querystring.parse(body);

            if (path === '/loginpost') {

                const {userid,password} = formData;

                const fetch = `SELECT \* FROM credentials WHERE userid='${userid}'`;

                con.query(fetch,(err,result)=>{

                    if(err){

                        console.log("Error in fetching Login data  : ",err);

                        return res.end(`There is some error in login <br> ${err}`)

                    }

                    if(result[0]){

                        if(password === result[0].password){

                            res.end(`Login successfully...`);

                        }

                        else{

                            res.end('Please with valid Credential <br> or First create new account..')

                        }

                    }

                    else{

                        res.end('Please with valid Credential <br> or First create new account..')

                    }

                })

            }

            else if(path==='/signuppost'){

                const {userid,password,name,email,mobile} = formData;

                const insert = `INSERT INTO credentials(userid, name, email, mobile, password) VALUES ('${userid}','${name}','${email}',${mobile},'${password}')`;

                con.query(insert,(err,result)=>{

                    if(err){

                        console.log("Insertin error  : ",err)

                        res.end(`There is some error in signup <br> ${err}`)

                    }

                    res.end(`Sing-Up successfully....<br> Now You can Login with your userid and password...<br><a href='/login'>Login</a>`)

                })

            }

        });

    }

});

server.listen(8000,()=>{

    console.log(`http://localhost:8000`)

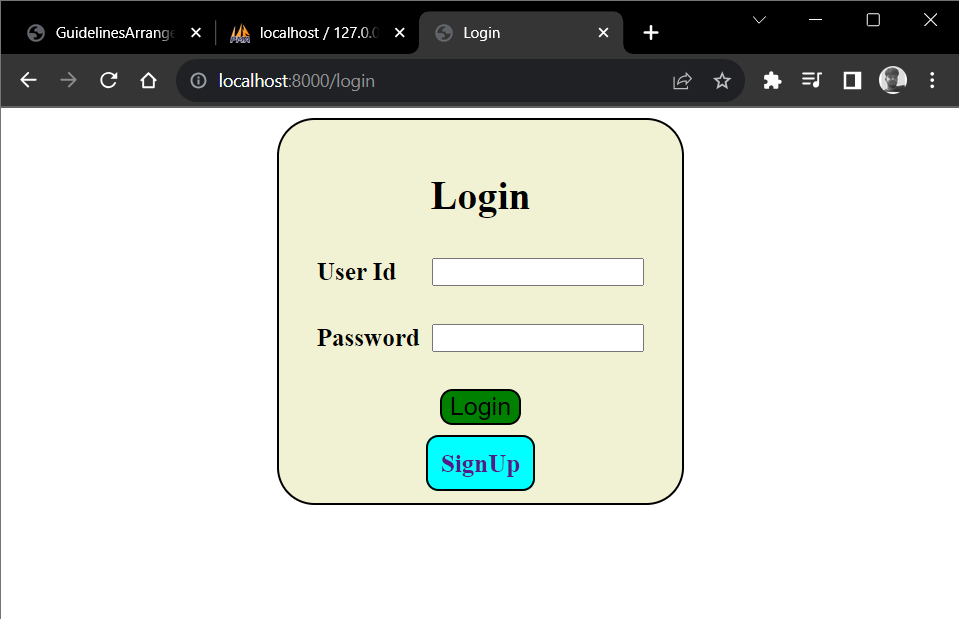
})

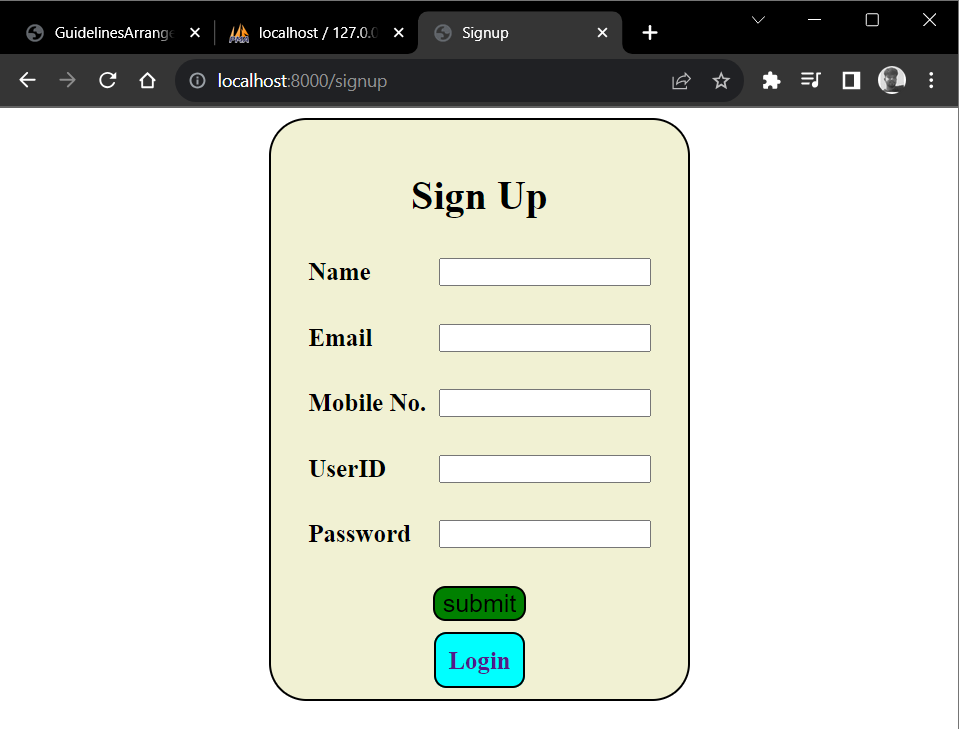
Now run

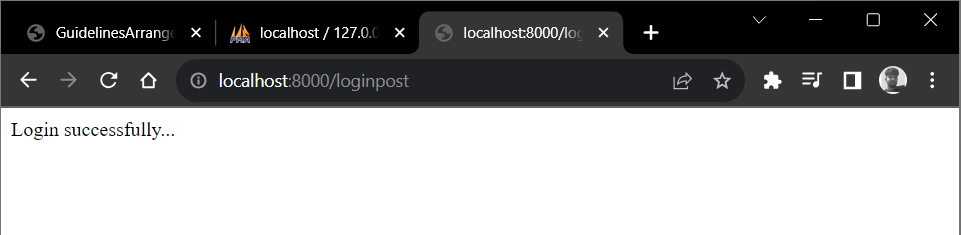
**node server.js**

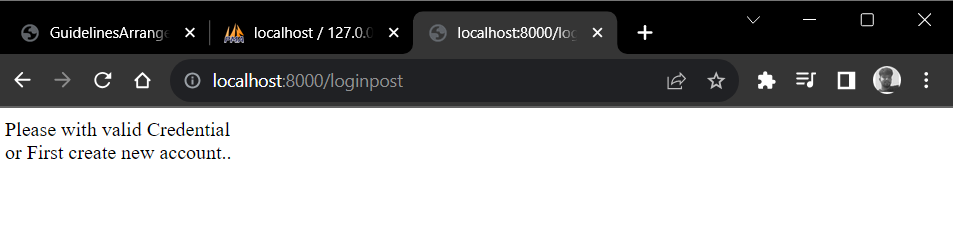
**Output :**

**Login route**



**signup route**

If login with valid credentials…

If login with invalid credentials..

Que 12:clock

<html>

<head>

    <title>Digital Clock</title>

    <style>

        body{

            background-color: rgb(127, 176, 255);

        }

        #clock {

            font-size: 200px;

            text-align: center;

        }

    </style>

</head>

<body>

    <div id="clock"></div>

    <script>

        function updateClock() {

            var now = new Date();

            var hr = now.getHours();

            var min = now.getMinutes();

            var sec = now.getSeconds();

            var timeString = `${hr}:${min}:${sec}`;

            document.getElementById('clock').textContent = timeString;

        }

        function abc(){

            setInterval(updateClock, 1000);

        }

        abc();

    </script>

</body>

</html>

Output:



Que 13: Stack

<html>

    <head>

        <title>

            STACK

        </title>

    </head>

    <style>

        body{background-color:rgb(249, 246, 250)}

    </style>

    <body>

            enter text:<input type="number" id="0"/>

            <button id="b1" onclick="f1()">PUSH</button>

            <button id="b2" onclick="f2()">POP</button><br><br>

            <label id="1">Stack</label>

            <textarea rows="10" id="2"></textarea><br>

        <script>

            const arr=new Array();

            function f1(){

                var v1=document.getElementById("0").value;

                if(v1==""){

                    alert("value is empty");

                }

                else{

                arr.push(v1);

                document.getElementById("0").value="";

                var s="";

                for(var i=arr.length-1;i>=0;i--){

                    s=s+arr[i]+"\n";

                }

                document.getElementById("2").value=s;}

            }

            function f2(){

                if(arr.length==0){

                    alert("stack is empty");

                }

                else{

                arr.pop();

                var s="";

                for(var i=arr.length-1;i>=0;i--){

                    s=s+arr[i]+"\n";

                }

                document.getElementById("2").value=s;

                }

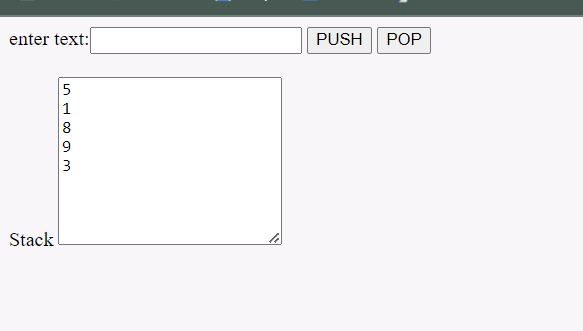
            }

         </script>

    </body>

</html>

Output:



Que 14:Image resize

<html>

    <head>

        <title>

            Resize

        </title>

    </head>

    <body>

        <img src="joshua-hanson-e616t35Vbeg-unsplash.jpg" width="400px" height="400px" id="1" name="img" onmouseenter="abc()" onmouseout="xyz()">

    </body>

    <script>

        function abc(){

            document.getElementById("1").style.width="700px".height="700px";

        }

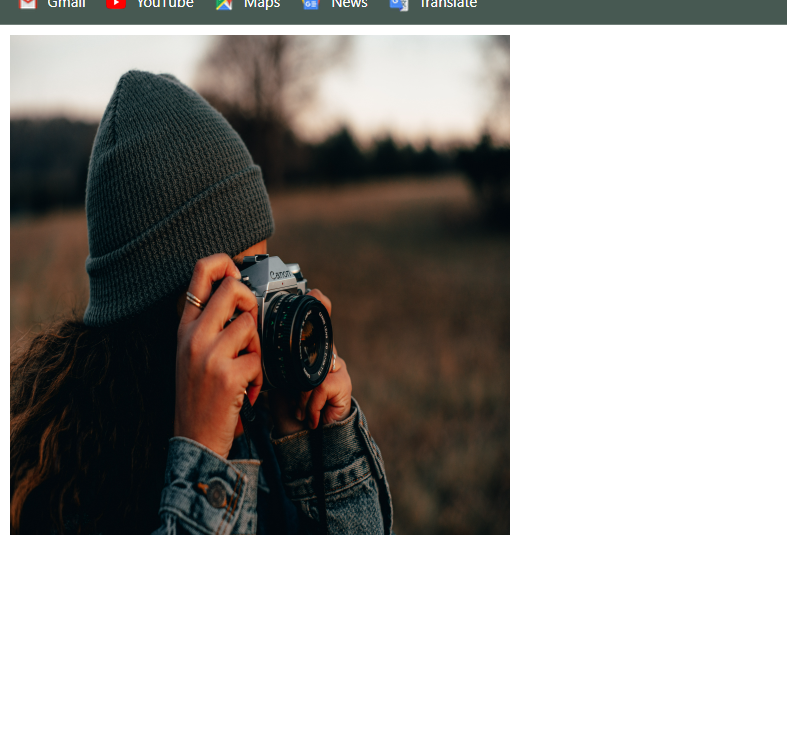
        function xyz(){

            document.getElementById("1").style.width="400px".height="400px";

        }

    </script>

</html>

Output: 

Que 15:Image Animation

<!DOCTYPE html>

<html>

<head>

    <title>Continuous Image Animation</title>

    <style>

        #movingImage {

            position: absolute;

            width: 100px;

            height: 100px;

            transition: 2s;         }

    </style>

</head>

<body>

    <img id="movingImage" src="joshua-hanson-e616t35Vbeg-unsplash.jpg">

    <script>

        function moveImage() {

            var windowWidth = window.innerWidth;

            var windowHeight = window.innerHeight;

            var randomX = Math.random() \* (windowWidth - 100);

            var randomY = Math.random() \* (windowHeight - 100);

            document.getElementById("movingImage").style.left = randomX + "px";

            document.getElementById("movingImage").style.top = randomY + "px";

            setTimeout(moveImage, 2000);

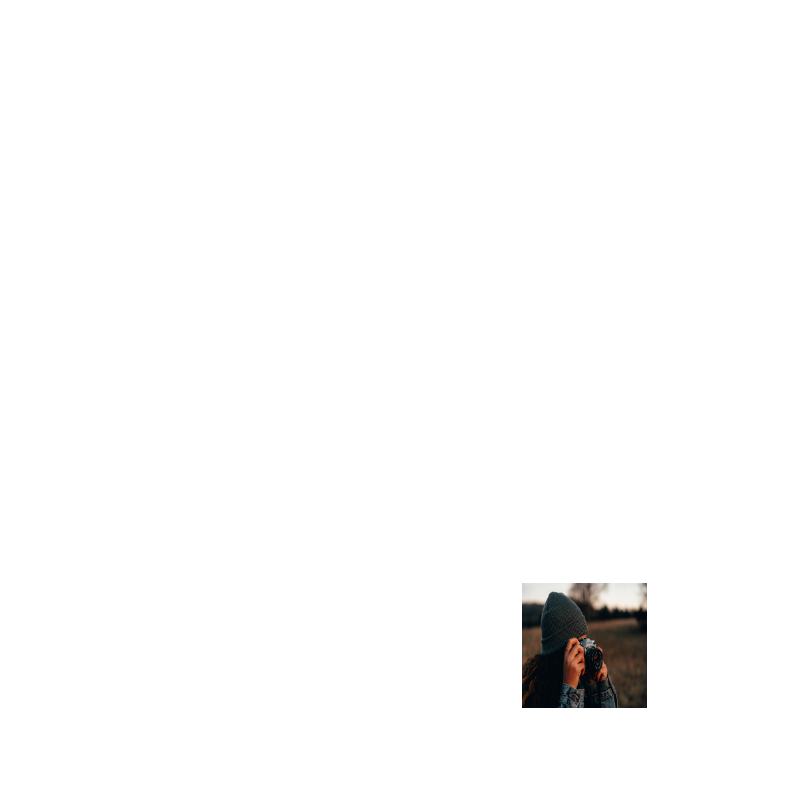
        }

        window.onload = moveImage;

    </script>

</body>

</html>

Output: 

Que 16: Add data into the table

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Table</title>

    <style>

        tr{

            margin: 20px 20px;

            border-bottom: 2px solid black;

        }

        table{

            border: 2px solid black;

        }

    </style>

</head>

<body>

    <label for="name">Name : </label>

    <input type="text" id="name"><br>

    <label for="age">Age : </label>

    <input type="number" name="age" id="age"><br><br>

    <button onclick="addData()">Add to table</button><br><br>

    <table id="table">

        <thead>

            <tr>

                <th>Name</th>

                <th>Age</th>

            </tr>

        </thead>

    </table>

    <script>

        const name = document.getElementById('name');

        const age = document.getElementById('age');

        const table = document.getElementById('table');

        const addData = ()=>{

            console.log(name.value)

            console.log(age.value)

            if(name.value && age.value){

                const tr = document.createElement('tr')

                const td1 = document.createElement('td')

                const td2 = document.createElement('td')

                td1.innerHTML = name.value;

                td2.innerHTML = age.value;

                tr.appendChild(td1)

                tr.appendChild(td2)

                table.appendChild(tr)

                name.value = '';

                age.value = '';

            }

            else{

                alert("Input field may be empty")

            }

        }

    </script>

</body>

</html>

Output:

