INTERNET TECHNOLOGIES

SUBMITTED BY
- VIVEK SHARMA

COURSE – B.SC(H) CS

YEAR – 3RD

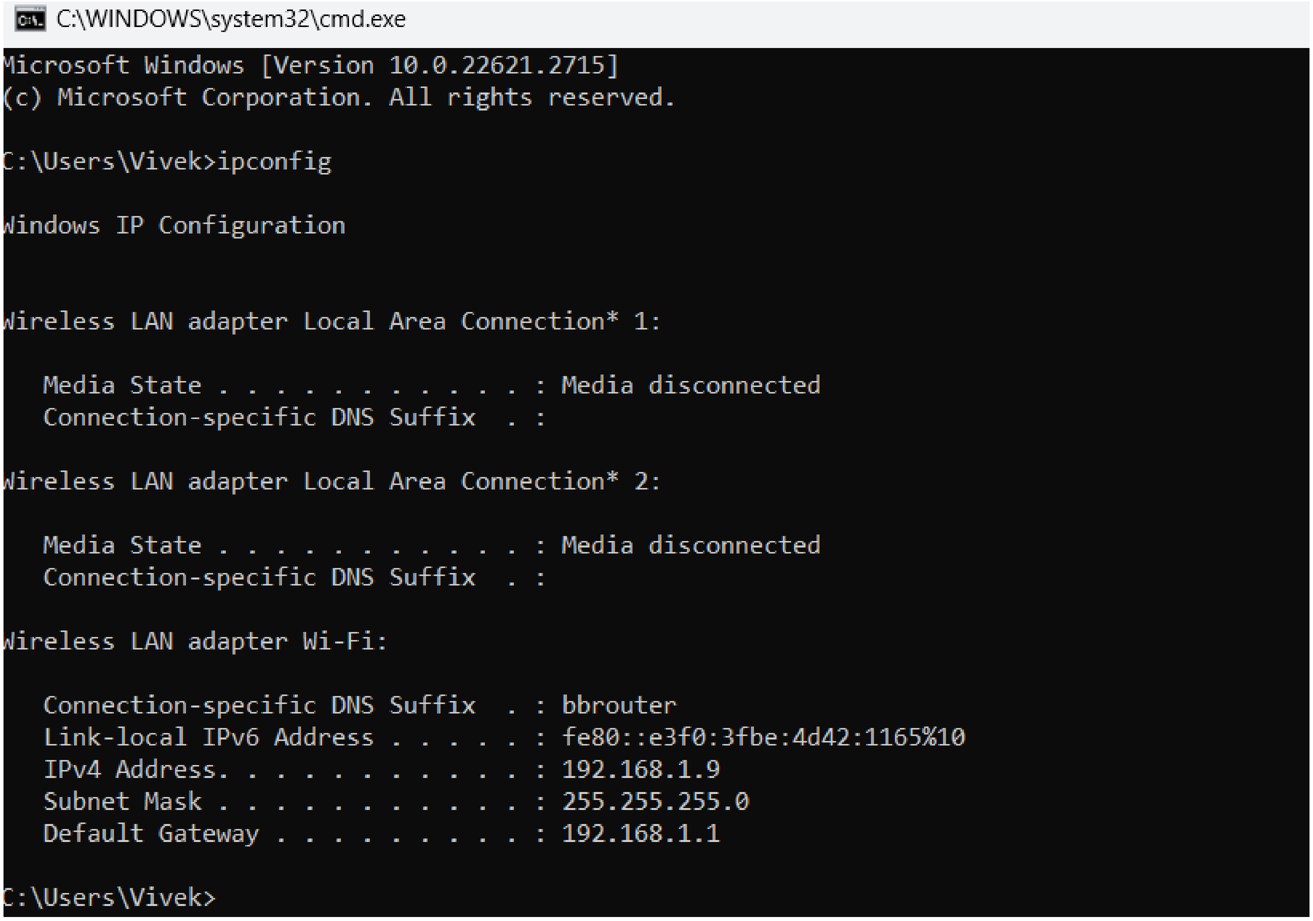
SEMESTER – 5TH

ROLL NO. – 21013570104

Clg ROLL NO. – 2K21/CS/111

SUBMITTED TO
–PARUL MA’AM

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.



C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Vivek>ipconfig

Windows IP Configuration


Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : bbrouter
   Link-local IPv6 Address . . . . . : fe80::e3f0:3fbe:4d42:1165%10
   IPv4 Address. . . . . . . . . . . : 192.168.1.9
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1

C:\Users\Vivek>
```

To determine the network address, we perform a bitwise AND operation between the IP address and the subnet mask.

- **IP Address (binary): 11000000.10101000.00000001.00001000**
- **Subnet Mask (binary): 11111111.11111111.11111111.00000000**

Performing bitwise AND:

- **Network Address (binary): 11000000.10101000.00000001.00000000**

Converting back to decimal:

- **Network Address: 192.168.1.0**

Maximum number of systems:

The subnet mask specifies that the last 8 bits of the IP address are used for hosts. Since the subnet mask is 255.255.255.0, the mask is a /24 mask. This means 24 bits are used for the network and 8 bits are used for hosts.

2^8 (number of bits used for hosts) - 2 (reserved addresses for network address and broadcast address) = 256 - 2 = 254

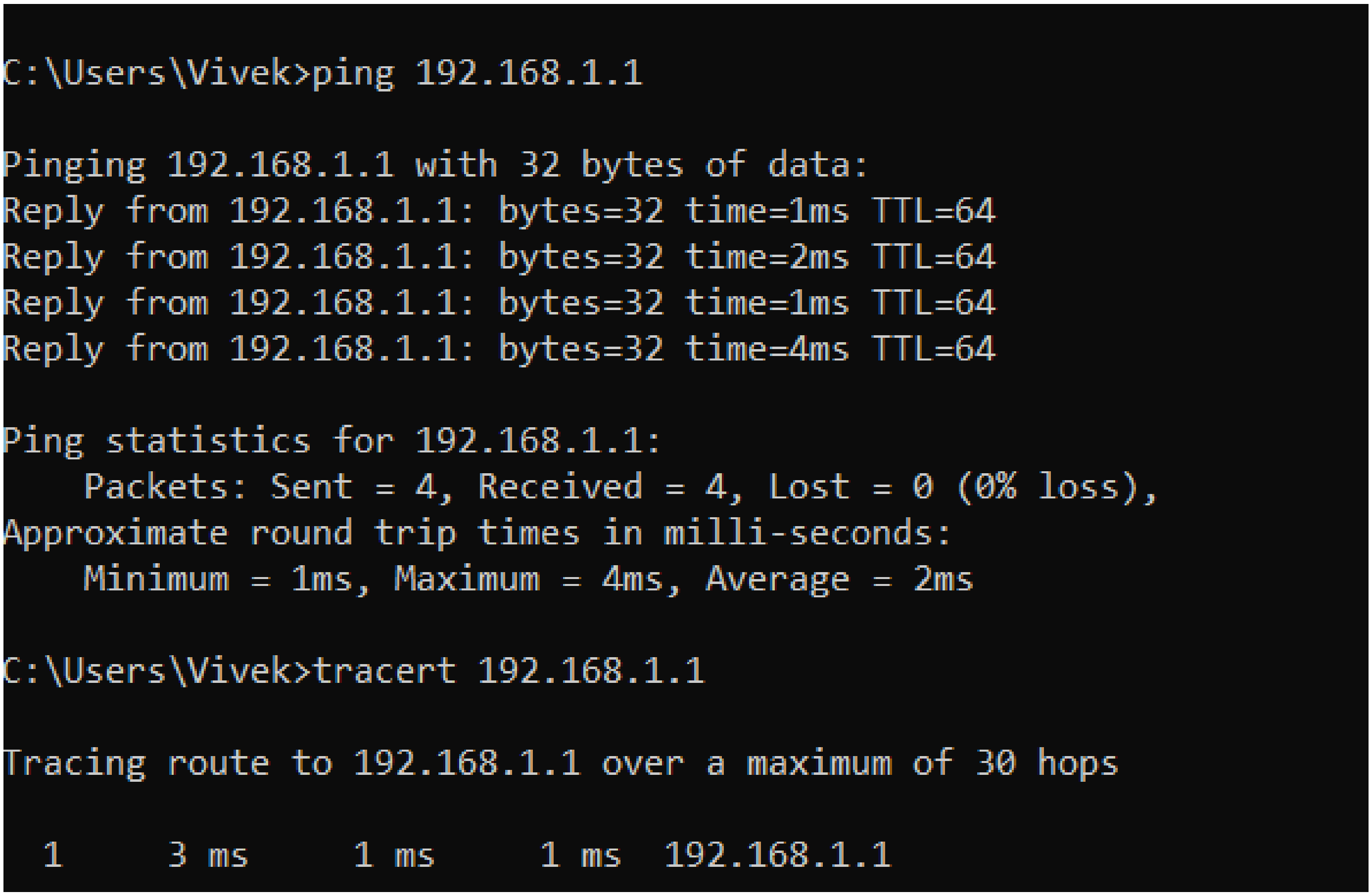
Therefore, the maximum number of systems on the network is 254.

Available IP address range:

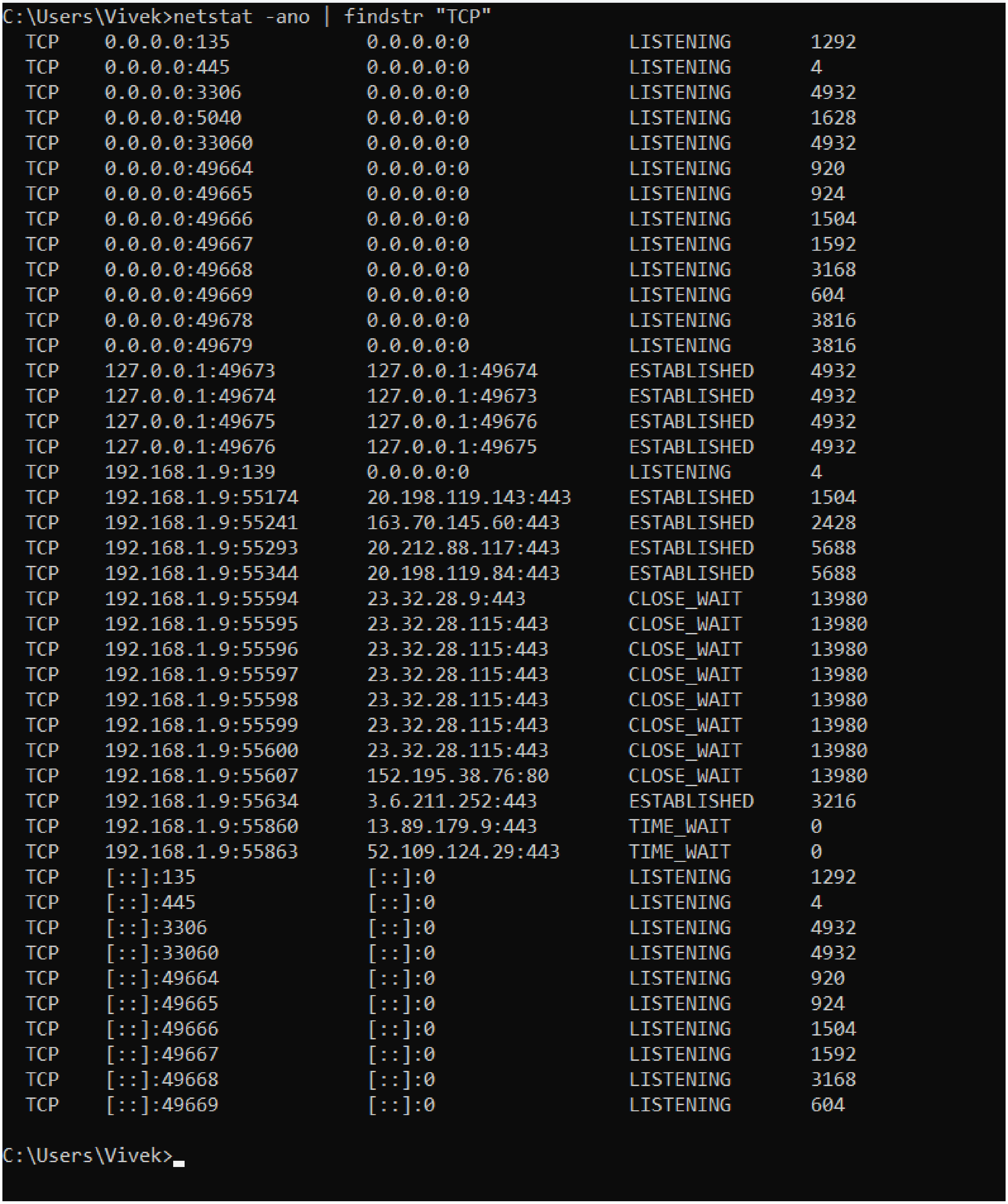
- **Start address: 192.168.1.1**
- **End address: 192.168.1.254**

Therefore, the available IP address range is: 192.168.1.1 - 192.168.1.254.

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.



```
C:\Users\Vivek>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64
Reply from 192.168.1.1: bytes=32 time=2ms TTL=64
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64
Reply from 192.168.1.1: bytes=32 time=4ms TTL=64

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 4ms, Average = 2ms

C:\Users\Vivek>tracert 192.168.1.1

Tracing route to 192.168.1.1 over a maximum of 30 hops

  1     3 ms     1 ms     1 ms  192.168.1.1
```



```
C:\Users\Vivek>netstat -ano | findstr "TCP"
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING       1292
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:3306           0.0.0.0:0              LISTENING       4932
  TCP    0.0.0.0:5040           0.0.0.0:0              LISTENING       1628
  TCP    0.0.0.0:33060          0.0.0.0:0              LISTENING       4932
  TCP    0.0.0.0:49664          0.0.0.0:0              LISTENING       920
  TCP    0.0.0.0:49665          0.0.0.0:0              LISTENING       924
  TCP    0.0.0.0:49666          0.0.0.0:0              LISTENING       1504
  TCP    0.0.0.0:49667          0.0.0.0:0              LISTENING       1592
  TCP    0.0.0.0:49668          0.0.0.0:0              LISTENING       3168
  TCP    0.0.0.0:49669          0.0.0.0:0              LISTENING       604
  TCP    0.0.0.0:49678          0.0.0.0:0              LISTENING       3816
  TCP    0.0.0.0:49679          0.0.0.0:0              LISTENING       3816
  TCP    127.0.0.1:49673        127.0.0.1:49674        ESTABLISHED     4932
  TCP    127.0.0.1:49674        127.0.0.1:49673        ESTABLISHED     4932
  TCP    127.0.0.1:49675        127.0.0.1:49676        ESTABLISHED     4932
  TCP    127.0.0.1:49676        127.0.0.1:49675        ESTABLISHED     4932
  TCP    192.168.1.9:139        0.0.0.0:0              LISTENING       4
  TCP    192.168.1.9:55174      20.198.119.143:443     ESTABLISHED     1504
  TCP    192.168.1.9:55241      163.70.145.60:443      ESTABLISHED     2428
  TCP    192.168.1.9:55293      20.212.88.117:443      ESTABLISHED     5688
  TCP    192.168.1.9:55344      20.198.119.84:443      ESTABLISHED     5688
  TCP    192.168.1.9:55594      23.32.28.9:443         CLOSE_WAIT      13980
  TCP    192.168.1.9:55595      23.32.28.115:443       CLOSE_WAIT      13980
  TCP    192.168.1.9:55596      23.32.28.115:443       CLOSE_WAIT      13980
  TCP    192.168.1.9:55597      23.32.28.115:443       CLOSE_WAIT      13980
  TCP    192.168.1.9:55598      23.32.28.115:443       CLOSE_WAIT      13980
  TCP    192.168.1.9:55599      23.32.28.115:443       CLOSE_WAIT      13980
  TCP    192.168.1.9:55600      23.32.28.115:443       CLOSE_WAIT      13980
  TCP    192.168.1.9:55607      152.195.38.76:80       CLOSE_WAIT      13980
  TCP    192.168.1.9:55634      3.6.211.252:443        ESTABLISHED     3216
  TCP    192.168.1.9:55860      13.89.179.9:443        TIME_WAIT       0
  TCP    192.168.1.9:55863      52.109.124.29:443      TIME_WAIT       0
  TCP    [::]:135               [::]:0                 LISTENING       1292
  TCP    [::]:445               [::]:0                 LISTENING       4
  TCP    [::]:3306              [::]:0                 LISTENING       4932
  TCP    [::]:33060             [::]:0                 LISTENING       4932
  TCP    [::]:49664             [::]:0                 LISTENING       920
  TCP    [::]:49665             [::]:0                 LISTENING       924
  TCP    [::]:49666             [::]:0                 LISTENING       1504
  TCP    [::]:49667             [::]:0                 LISTENING       1592
  TCP    [::]:49668             [::]:0                 LISTENING       3168
  TCP    [::]:49669             [::]:0                 LISTENING       604

C:\Users\Vivek>
```

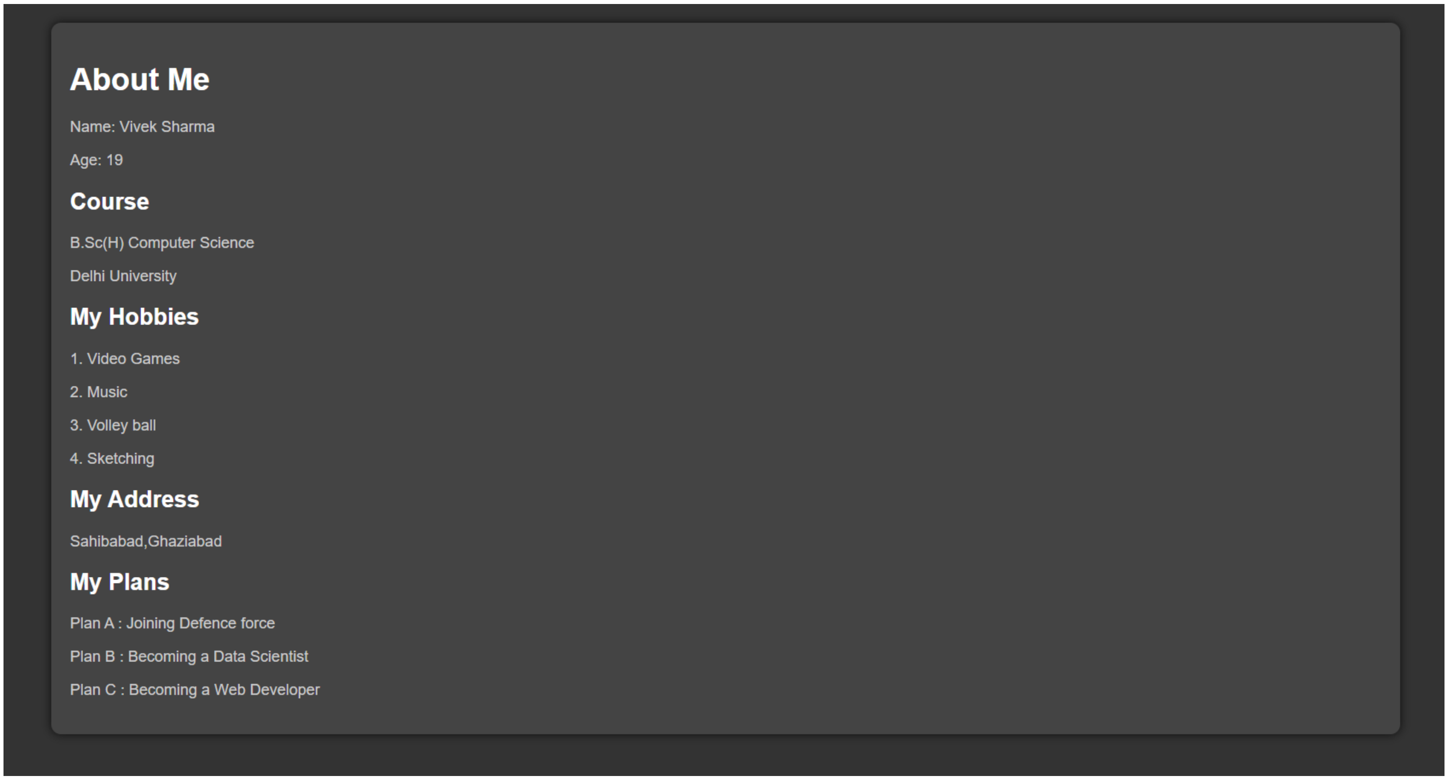
3. Create an HTML page that shows information about you, your course, hobbies, address, and your plans. Use CSS for styling of HTML page so that looks nice.

Index.html

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initialscale=1.0">
    <link rel="stylesheet" type="text/css" href="style.css">
</head>
<title>Document</title>
</head>

<body>
    <div class="container">

        <h1>About Me</h1>
        <p>Name: Vivek Sharma</p>
        <p>Age: 19</p>

        <h2>Course</h2>
        <p>B.Sc(H) Computer Science</p>
        <p>Delhi University</p>

        <h2>My Hobbies</h2>
        <p>1. Video Games</p>
        <p>2. Music</p>
        <p>3. Volley ball</p>
        <p>4. Sketching</p>

        <h2>My Address</h2>
        <p>Sahibabad,Ghaziabad</p>

        <h2>My Plans</h2>
        <p>Plan A : Joining Defence force</p>
        <p>Plan B : Becoming a Data Scientist</p>
        <p>Plan C : Becoming a Web Developer</p>
    </div>
</body>

</html>
```

Style.css

```css
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #333;
    color:
        #f0f0f0;
}

.container {
    width: 80%;
    margin: 20px auto;
    padding:
        20px;
    background-color: #444;
    border-radius:
        10px;
    box-shadow: 0px 0px 10px 0px rgba(0, 0, 0,0.75);
}

h1,
h2 {
    color: #fff;
}

p {
    color: #ccc;
}
```



About Me

Name: Vivek Sharma

Age: 19

Course

B.Sc(H) Computer Science

Delhi University

My Hobbies

1. Video Games

2. Music

3. Volley ball

4. Sketching

My Address

Sahibabad,Ghaziabad

My Plans

Plan A : Joining Defence force

Plan B : Becoming a Data Scientist

Plan C : Becoming a Web Developer

4. Create an HTML page with the sole purpose to show multiplication tables of 2 to 10 (row-wise) created by JavaScript. Initially, the

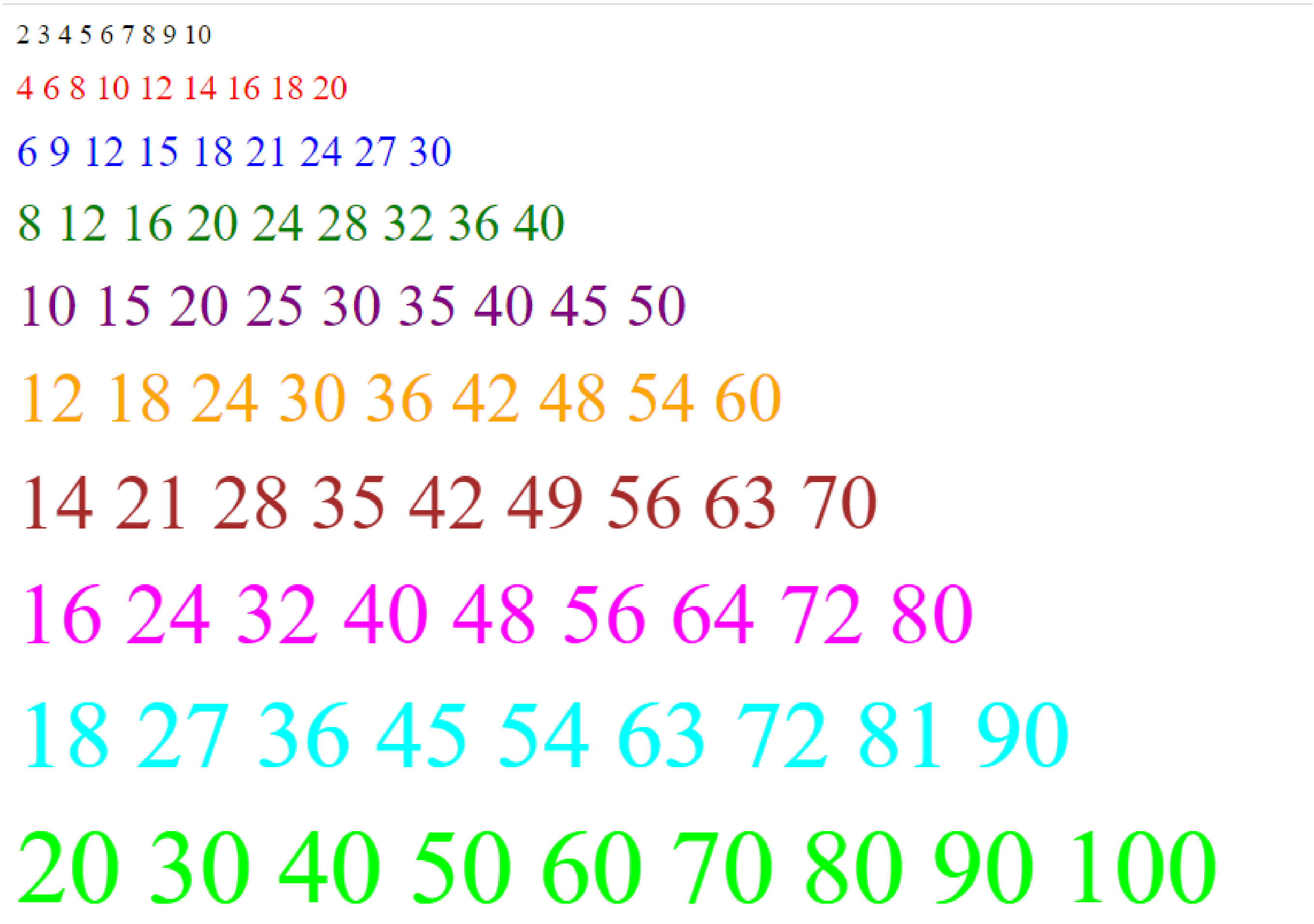
page is blank. With help of setInterval function print a row every 5 seconds in different colors and increasing font size.

Index.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Multiplication Tables</title>
    <style>
        .row {
            display: block;
            margin-bottom: 10px;
        }
    </style>
</head>
<body>
    <div id="multiplicationTable"></div>
    <script src="table.js"></script>
</body>
</html>
```

Table.js

```javascript
let colors = ['red', 'blue', 'green', 'purple', 'orange', 'brown', 'magenta', 'cyan', 'lime', 'pink'];
        let fontSizes = [20, 25, 30, 35, 40, 45, 50, 55, 60, 65];

        let table = document.getElementById('multiplicationTable');

        let counter = 1;

        function printRow() {
            let row = '';
            for (let i = 2; i <= 10; i++) {
                row += i * counter + ' ';
            }
            let rowElement = document.createElement('div');
            rowElement.textContent = row;
            rowElement.style.color = colors[counter - 2];
            rowElement.style.fontSize = fontSizes[counter - 2] + 'px';
            rowElement.classList.add('row');
            table.appendChild(rowElement);

            counter++;

            if (counter > 10) {
                clearInterval(interval);
            }
        }

        let interval = setInterval(printRow, 5000);
```



```
2 3 4 5 6 7 8 9 10
4 6 8 10 12 14 16 18 20
6 9 12 15 18 21 24 27 30
8 12 16 20 24 28 32 36 40
10 15 20 25 30 35 40 45 50
12 18 24 30 36 42 48 54 60
14 21 28 35 42 49 56 63 70
16 24 32 40 48 56 64 72 80
18 27 36 45 54 63 72 81 90
20 30 40 50 60 70 80 90 100
```

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.

Index.html

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

</head>
<body>
```

```html

    <p id="para">HI, I am Vivek</p>

    <div style="display: grid; grid-template-columns: repeat(3, 1fr); gap: 10px; margin-top: 
20px;">
        <button onclick="changeColor('red')">Red</button>
        <button onclick="changeColor('green')">Green</button>
        <button onclick="changeColor('blue')">Blue</button>
        <button onclick="changeFontSize(10)">10</button>
        <button onclick="changeFontSize(20)">20</button>
        <button onclick="changeFontSize(30)">30</button>
        <button onclick="changeFont('Arial')">Arial</button>
        <button onclick="changeFont('Courier New')">Courier New</button>
        <button onclick="changeFont('Times New Roman')">Times New Roman</button>
    </div>

    <script src="q5.js"></script>
</body>
</html>
```

Q5.js

```javascript
function changeColor(color) {
    document.getElementById('para').style.color = color;
}
function changeFontSize(fontSize) {
    document.getElementById('para').style.fontSize = fontSize + 'px';
}
function changeFont(font) {
    document.getElementById('para').style.fontFamily = font;
}
```

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.

Index.html

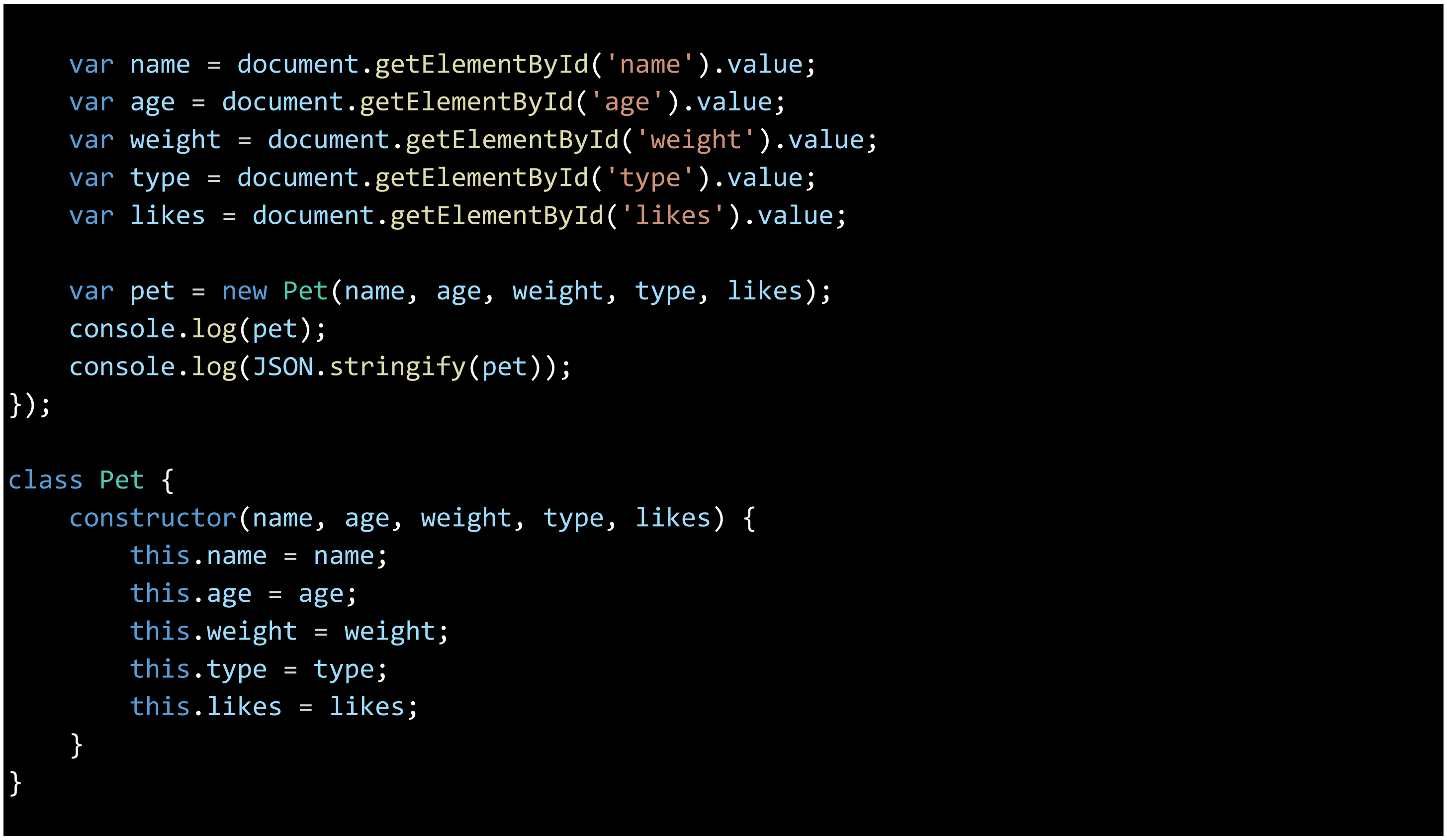
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Pet Form</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <form id="petForm">
        <label for="name">Name</label>
        <input type="text" id="name" name="name" required>

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

        <label for="weight">Weight</label>
        <input type="number" id="weight" name="weight" required>

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

        <label for="likes">What it likes most</label>
        <textarea id="likes" name="likes" required></textarea>

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

    <script src="q6.js"></script>
</body>
</html>
```

Style.css

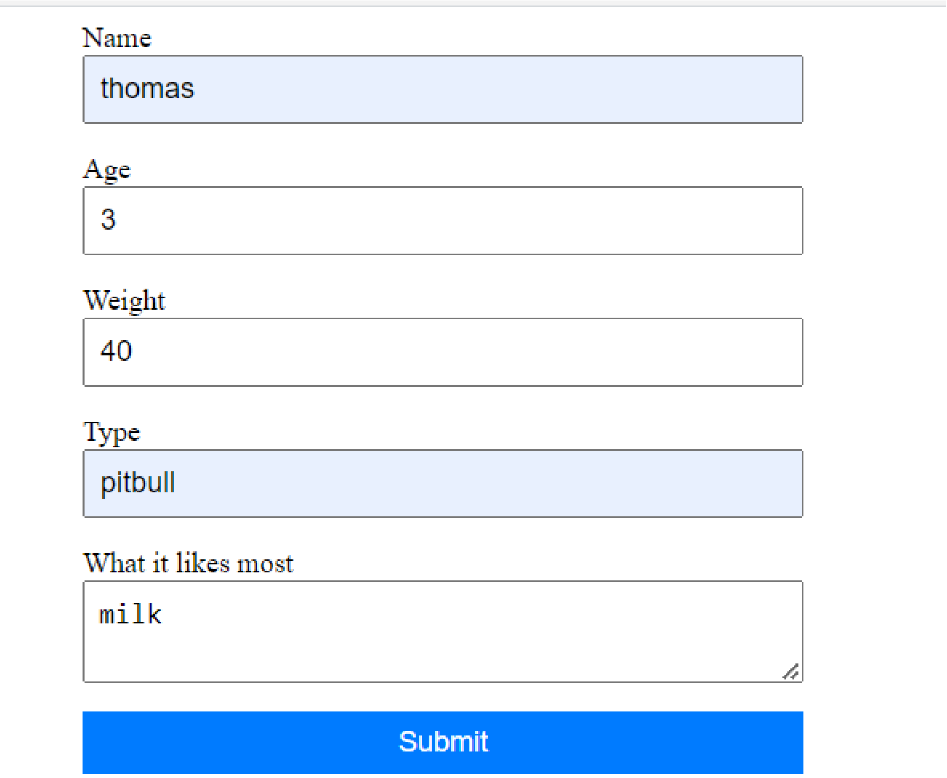
```
form {
    display: flex;
    flex-direction: column;
    max-width: 400px;
    margin: 0 auto;
}

input, textarea, select {
    margin-bottom: 1rem;
    padding: .5rem;
    font-size: 1rem;
}

button {
    background-color: #007BFF;
    color: white;
    padding: .5rem 1rem;
    font-size: 1rem;
    border: none;
    cursor: pointer;
}

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

Q6.js

```
document.getElementById('petForm').addEventListener('submit', function(event) {
    event.preventDefault();
```



```
    var name = document.getElementById('name').value;
    var age = document.getElementById('age').value;
    var weight = document.getElementById('weight').value;
    var type = document.getElementById('type').value;
    var likes = document.getElementById('likes').value;

    var pet = new Pet(name, age, weight, type, likes);
    console.log(pet);
    console.log(JSON.stringify(pet));
});

class Pet {
    constructor(name, age, weight, type, likes) {
        this.name = name;
        this.age = age;
        this.weight = weight;
        this.type = type;
        this.likes = likes;
    }
}
```



Name

thomas

Age

3

Weight

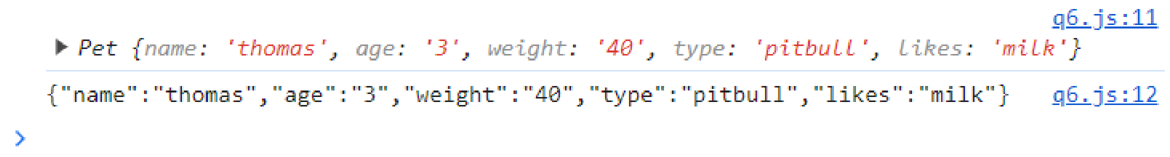
40

Type

pitbull

What it likes most

milk



```
                                                                       q6.js:11
▶ Pet {name: 'thomas', age: '3', weight: '40', type: 'pitbull', likes: 'milk'}
{"name":"thomas","age":"3","weight":"40","type":"pitbull","likes":"milk"}   q6.js:12
>
```

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.

Index.html

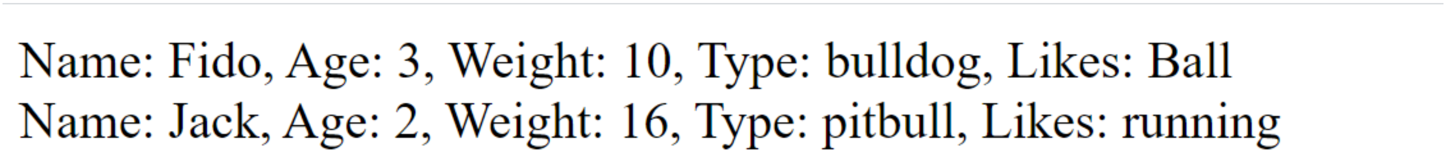
```
<!DOCTYPE html>
<html>
<head>
    <title>Pet App</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
</head>
<body>
    <div id="app"></div>
    <script src="app.js"></script>
</body>
</html>
```

data.json

```
{
    "pets": [
        {
            "name": "Fido",
            "age": 3,
            "weight": 10,
            "type": "bulldog",
            "likes": "Ball"
        },
        {
            "name": "Jack",
            "age": 2,
            "weight": 16,
            "type": "pitbull",
            "likes": "running"
        }
    ]
}
```

app.js

```
$.ajax({
    url: "data.json",
    success: function(petData) {
        let app = document.getElementById('app');

        for (let i = 0; i < petData.pets.length; i++) {
            let pet = petData.pets[i];
            let div = document.createElement('div');
            div.innerHTML = `Name: ${pet.name}, Age: ${pet.age}, Weight: ${pet.weight}, Type: 
${pet.type}, Likes: ${pet.likes}`;
            app.appendChild(div);
        }
    }
});
```

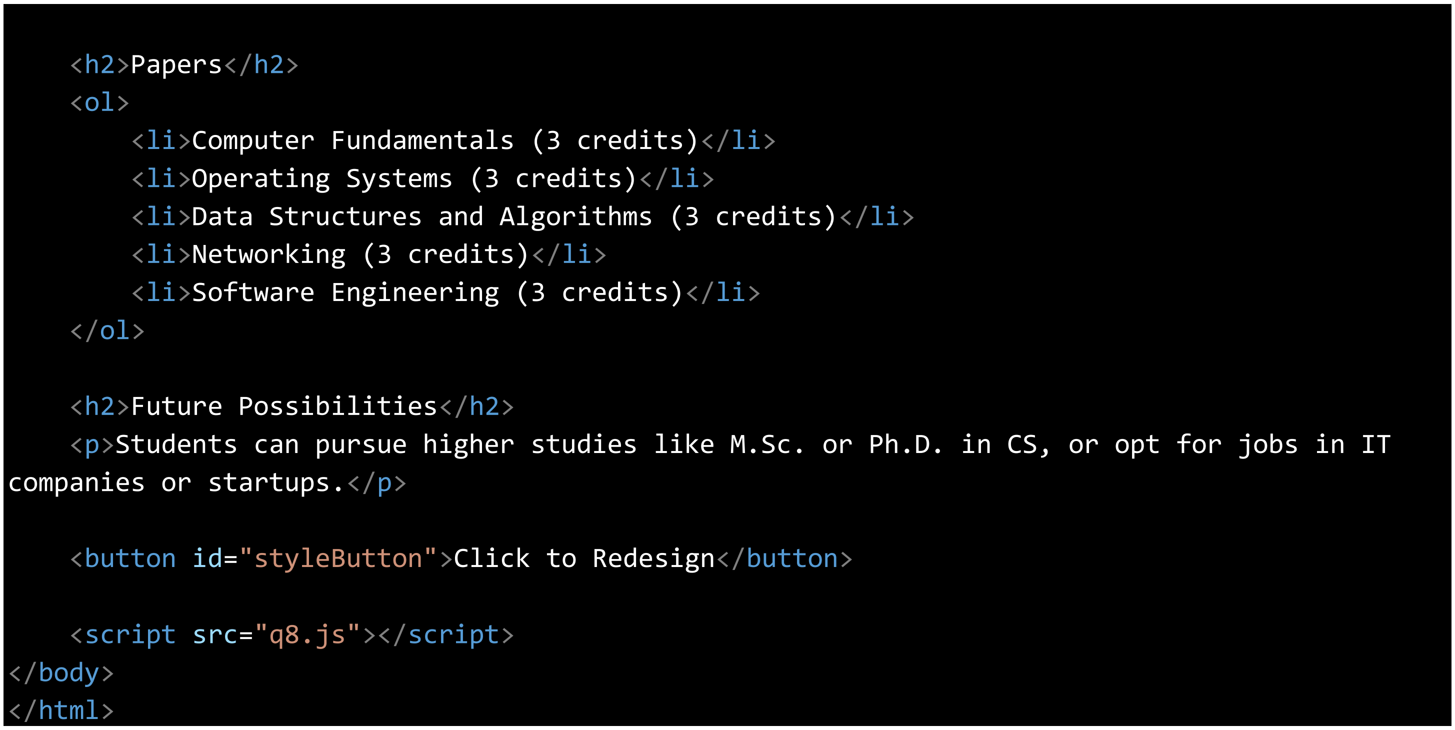


```
Name: Fido, Age: 3, Weight: 10, Type: bulldog, Likes: Ball
Name: Jack, Age: 2, Weight: 16, Type: pitbull, Likes: running
```

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.

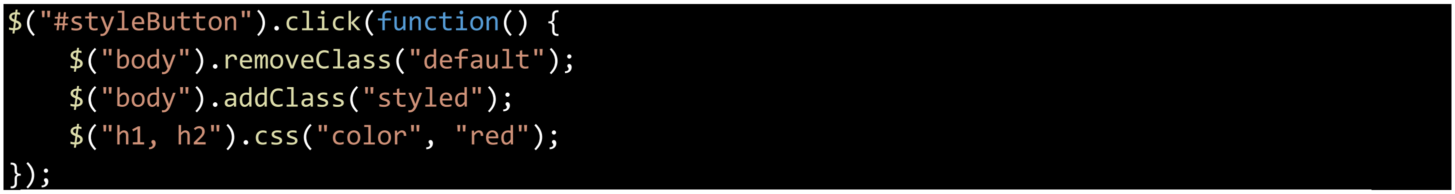
Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>B.Sc. Hons CS Course</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <style>
        .default {
            font-family: Arial, sans-serif;
            font-size: 16px;
        }
        .styled {
            font-family: 'Courier New', monospace;
            font-size: 24px;
            color: blue;
        }
    </style>
</head>
<body class="default">
    <h1>B.Sc. Hons CS Course</h1>
    <h2>Fee</h2>
    <p>INR 50,000 per year</p>

    <h2>Eligibility Criteria</h2>
    <p>10+2 with Mathematics or Physics as one of the subjects</p>
```

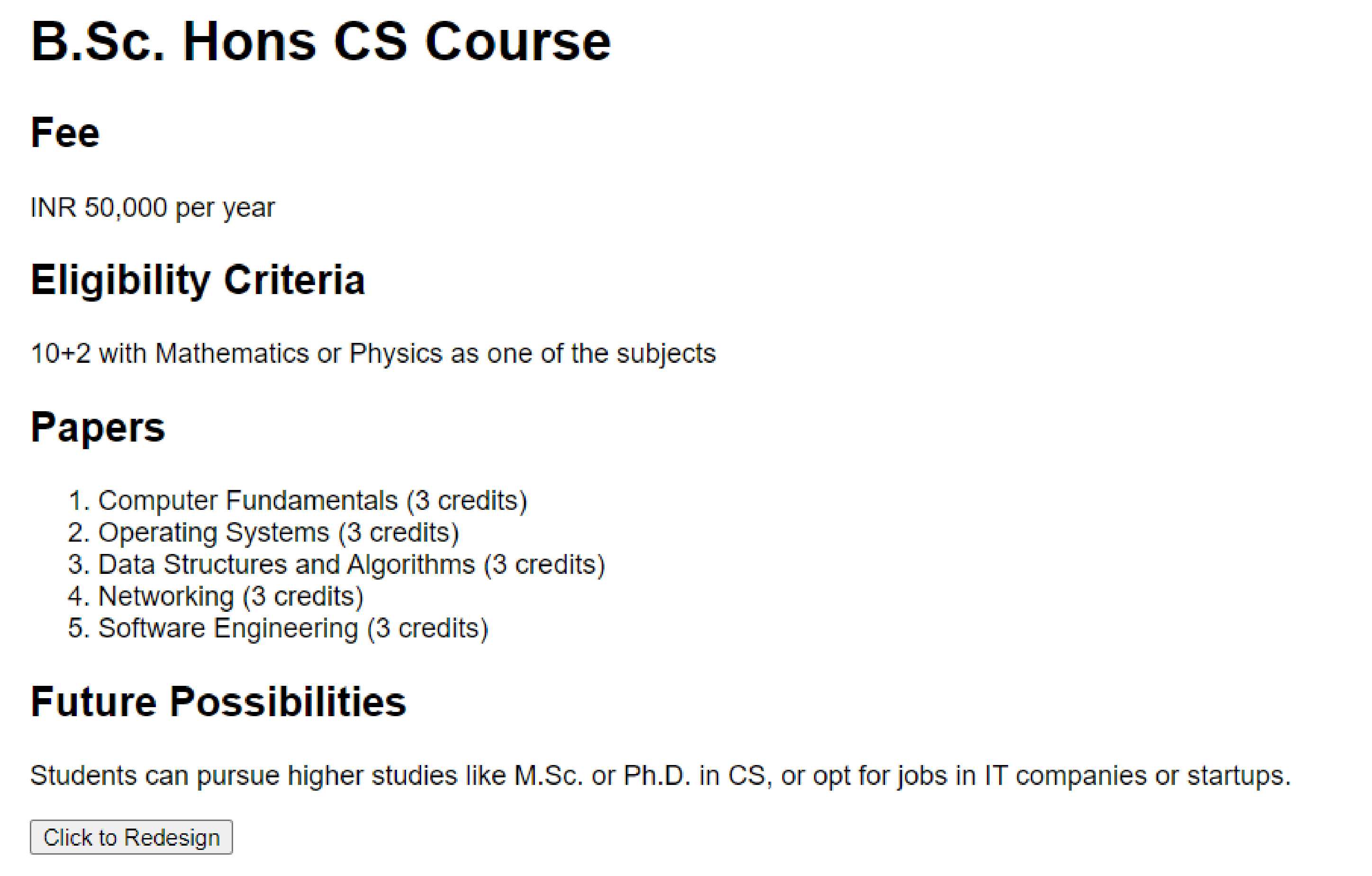


```
    <h2>Papers</h2>
    <ol>
        <li>Computer Fundamentals (3 credits)</li>
        <li>Operating Systems (3 credits)</li>
        <li>Data Structures and Algorithms (3 credits)</li>
        <li>Networking (3 credits)</li>
        <li>Software Engineering (3 credits)</li>
    </ol>

    <h2>Future Possibilities</h2>
    <p>Students can pursue higher studies like M.Sc. or Ph.D. in CS, or opt for jobs in IT
companies or startups.</p>

    <button id="styleButton">Click to Redesign</button>

    <script src="q8.js"></script>
</body>
</html>
```

Q8.js



```
$("#styleButton").click(function() {
    $("body").removeClass("default");
    $("body").addClass("styled");
    $("h1, h2").css("color", "red");
});
```



B.Sc. Hons CS Course

Fee

INR 50,000 per year

Eligibility Criteria

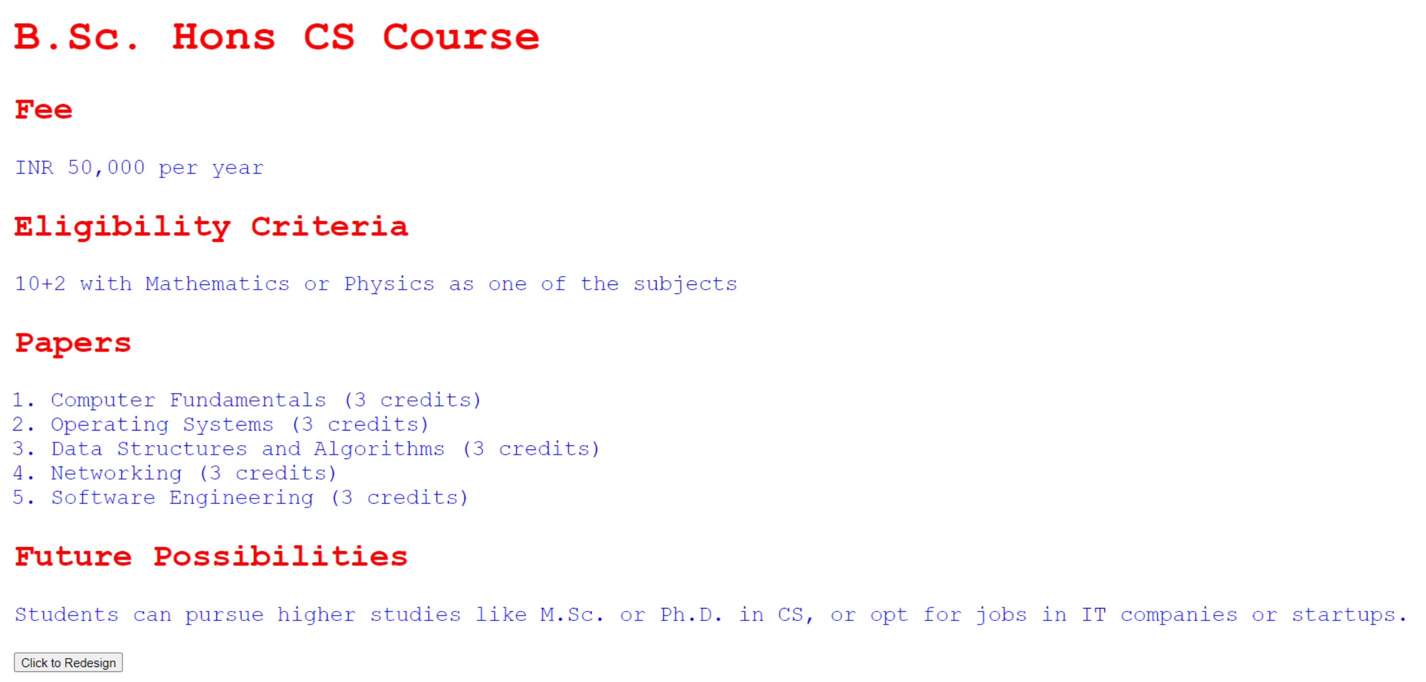
10+2 with Mathematics or Physics as one of the subjects

Papers

1. Computer Fundamentals (3 credits)
2. Operating Systems (3 credits)
3. Data Structures and Algorithms (3 credits)
4. Networking (3 credits)
5. Software Engineering (3 credits)

Future Possibilities

Students can pursue higher studies like M.Sc. or Ph.D. in CS, or opt for jobs in IT companies or startups.



B.Sc. Hons CS Course

Fee

INR 50,000 per year

Eligibility Criteria

10+2 with Mathematics or Physics as one of the subjects

Papers

1. Computer Fundamentals (3 credits)
2. Operating Systems (3 credits)
3. Data Structures and Algorithms (3 credits)
4. Networking (3 credits)
5. Software Engineering (3 credits)

Future Possibilities

Students can pursue higher studies like M.Sc. or Ph.D. in CS, or opt for jobs in IT companies or startups.

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.

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Image Gallery</title>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</head>
<body>
    <div class="container">
        <h1 class="text-center">Image Gallery</h1>
        <div class="row">
            <div class="col-md-4">
                <img
src="https://imgs.search.brave.com/9HMwN6WQgYZdiIQGli8YbEHeavS0UW499b9fS1QE3tM/rs:fit:860:0:0
/g:ce/aHR0cHM6Ly9tZWRp/YS5pc3RvY2twaG90/by5jb20vaWQvMTA0/MTk4NzQ4OC9waG90/by9jdXRlLWRvZy1w/dX
QtaGlzLWZhY2Ut/b24taGlzLWtuZWVz/LXRvLXRoZS1tYW4t/YW5kLXNtaWxpbmct/ZnJvbS10aGUtaGFu/ZHMtc2NyYX
RjaGlu/Zy1oZXItZWFyLmpw/Zz9zPTYxMng2MTIm/dz0wJms9MjAmYz1O/S0dmOG5tWFZka3Nt/TlMwQXk2OTZjVlBO/U
0lmQ0pKMXl1X3k5/akZHQnNNPQ" class="img-fluid">
            </div>
            <div class="col-md-4">
                <img
src="https://imgs.search.brave.com/9HMwN6WQgYZdiIQGli8YbEHeavS0UW499b9fS1QE3tM/rs:fit:860:0:0
/g:ce/aHR0cHM6Ly9tZWRp/YS5pc3RvY2twaG90/by5jb20vaWQvMTA0/MTk4NzQ4OC9waG90/by9jdXRlLWRvZy1w/dX
QtaGlzLWZhY2Ut/b24taGlzLWtuZWVz/LXRvLXRoZS1tYW4t/YW5kLXNtaWxpbmct/ZnJvbS10aGUtaGFu/ZHMtc2NyYX
RjaGlu/Zy1oZXItZWFyLmpw/Zz9zPTYxMng2MTIm/dz0wJms9MjAmYz1O/S0dmOG5tWFZka3Nt/TlMwQXk2OTZjVlBO/U
0lmQ0pKMXl1X3k5/akZHQnNNPQ" class="img-fluid">
            </div>
            <div class="col-md-4">
                <img
src="https://imgs.search.brave.com/9HMwN6WQgYZdiIQGli8YbEHeavS0UW499b9fS1QE3tM/rs:fit:860:0:0
/g:ce/aHR0cHM6Ly9tZWRp/YS5pc3RvY2twaG90/by5jb20vaWQvMTA0/MTk4NzQ4OC9waG90/by9jdXRlLWRvZy1w/dX
QtaGlzLWZhY2Ut/b24taGlzLWtuZWVz/LXRvLXRoZS1tYW4t/YW5kLXNtaWxpbmct/ZnJvbS10aGUtaGFu/ZHMtc2NyYX
RjaGlu/Zy1oZXItZWFyLmpw/Zz9zPTYxMng2MTIm/dz0wJms9MjAmYz1O/S0dmOG5tWFZka3Nt/TlMwQXk2OTZjVlBO/U
0lmQ0pKMXl1X3k5/akZHQnNNPQ" class="img-fluid">
            </div>
        </div>
        <div class="row">
            <div class="col-md-4">
                <img
src="https://imgs.search.brave.com/9HMwN6WQgYZdiIQGli8YbEHeavS0UW499b9fS1QE3tM/rs:fit:860:0:0
/g:ce/aHR0cHM6Ly9tZWRp/YS5pc3RvY2twaG90/by5jb20vaWQvMTA0/MTk4NzQ4OC9waG90/by9jdXRlLWRvZy1w/dX
QtaGlzLWZhY2Ut/b24taGlzLWtuZWVz/LXRvLXRoZS1tYW4t/YW5kLXNtaWxpbmct/ZnJvbS10aGUtaGFu/ZHMtc2NyYX
RjaGlu/Zy1oZXItZWFyLmpw/Zz9zPTYxMng2MTIm/dz0wJms9MjAmYz1O/S0dmOG5tWFZka3Nt/TlMwQXk2OTZjVlBO/U
0lmQ0pKMXl1X3k5/akZHQnNNPQ" class="img-fluid">
            </div>
            <div class="col-md-4">
                <img
src="https://imgs.search.brave.com/9HMwN6WQgYZdiIQGli8YbEHeavS0UW499b9fS1QE3tM/rs:fit:860:0:0
/g:ce/aHR0cHM6Ly9tZWRp/YS5pc3RvY2twaG90/by5jb20vaWQvMTA0/MTk4NzQ4OC9waG90/by9jdXRlLWRvZy1w/dX
QtaGlzLWZhY2Ut/b24taGlzLWtuZWVz/LXRvLXRoZS1tYW4t/YW5kLXNtaWxpbmct/ZnJvbS10aGUtaGFu/ZHMtc2NyYX
RjaGlu/Zy1oZXItZWFyLmpw/Zz9zPTYxMng2MTIm/dz0wJms9MjAmYz1O/S0dmOG5tWFZka3Nt/TlMwQXk2OTZjVlBO/U
0lmQ0pKMXl1X3k5/akZHQnNNPQ" class="img-fluid">
            </div>
            <div class="col-md-4">
                <img
src="https://imgs.search.brave.com/9HMwN6WQgYZdiIQGli8YbEHeavS0UW499b9fS1QE3tM/rs:fit:860:0:0
/g:ce/aHR0cHM6Ly9tZWRp/YS5pc3RvY2twaG90/by5jb20vaWQvMTA0/MTk4NzQ4OC9waG90/by9jdXRlLWRvZy1w/dX
QtaGlzLWZhY2Ut/b24taGlzLWtuZWVz/LXRvLXRoZS1tYW4t/YW5kLXNtaWxpbmct/ZnJvbS10aGUtaGFu/ZHMtc2NyYX
RjaGlu/Zy1oZXItZWFyLmpw/Zz9zPTYxMng2MTIm/dz0wJms9MjAmYz1O/S0dmOG5tWFZka3Nt/TlMwQXk2OTZjVlBO/U
0lmQ0pKMXl1X3k5/akZHQnNNPQ" class="img-fluid">
            </div>
        </div>
    </div>
</body>
</html>
```

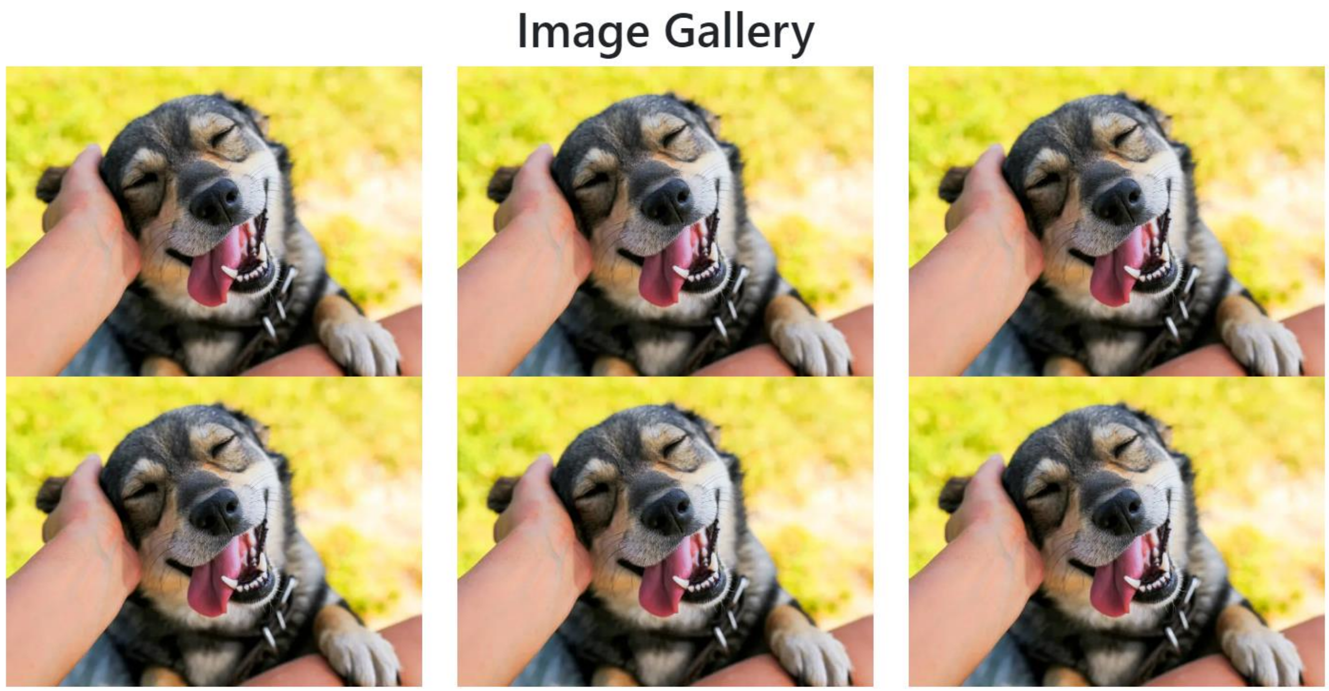
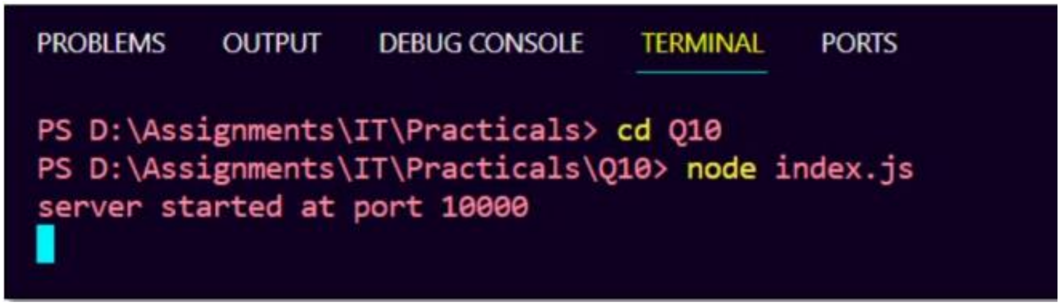


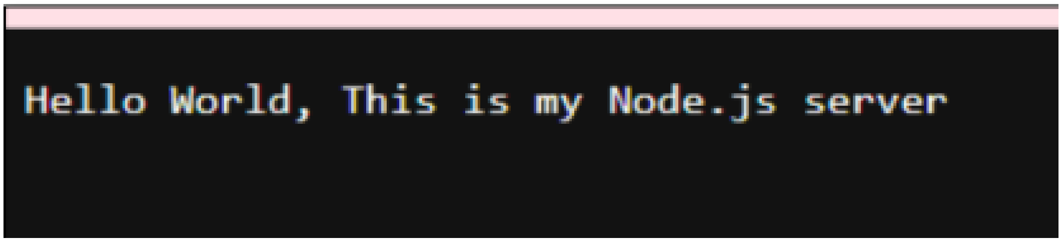
Image Gallery

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.

```
varhttp = require('http');
http.createServer((request,response)=>{
response.write("Hello World, This is my Node.js server");
response.end();
})
.listen(10000,()=>console.log('server started at port 10000'));
```



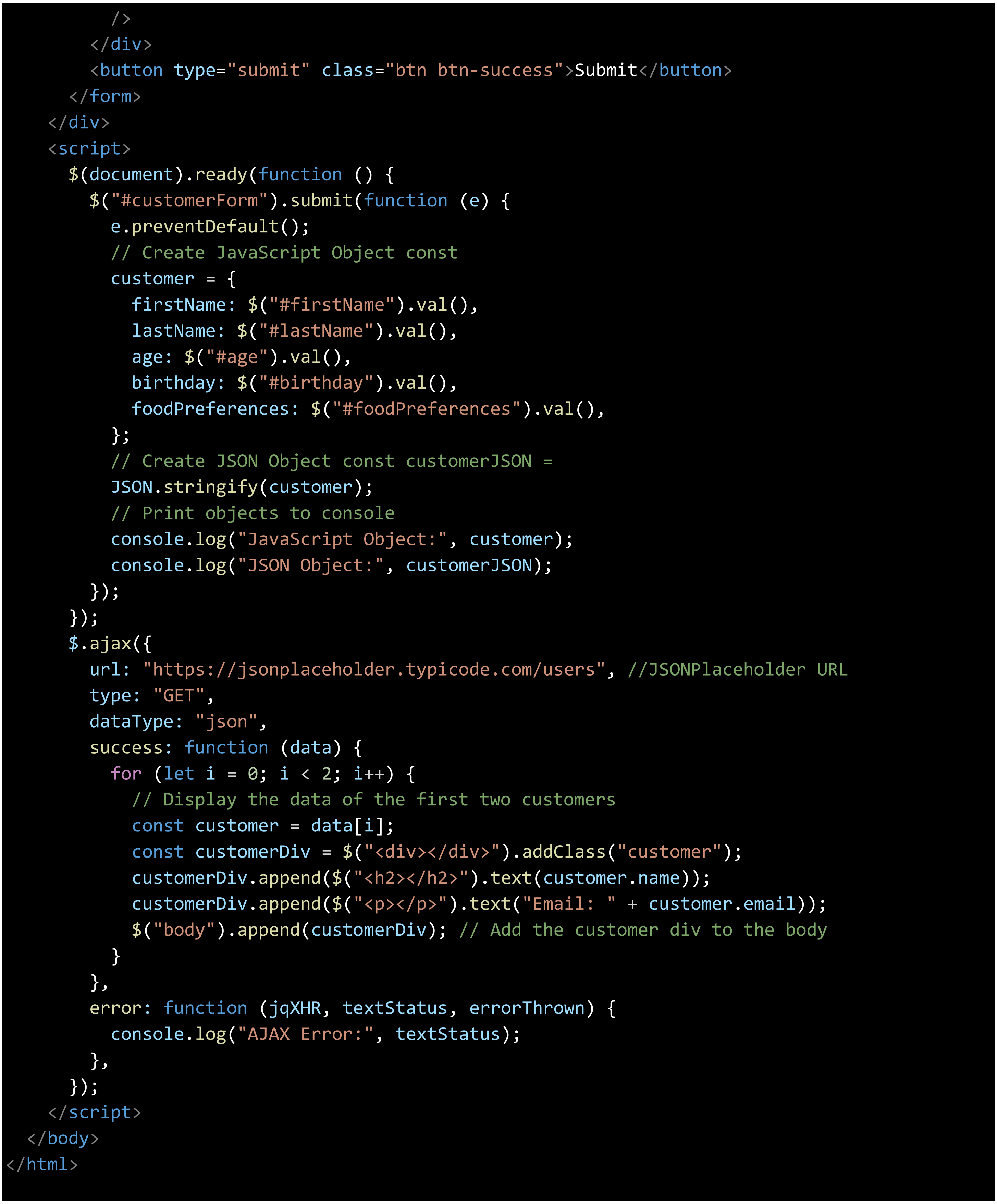
```
PS D:\Assignments\IT\Practicals> cd Q10
PS D:\Assignments\IT\Practicals\Q10> node index.js
server started at port 10000
```



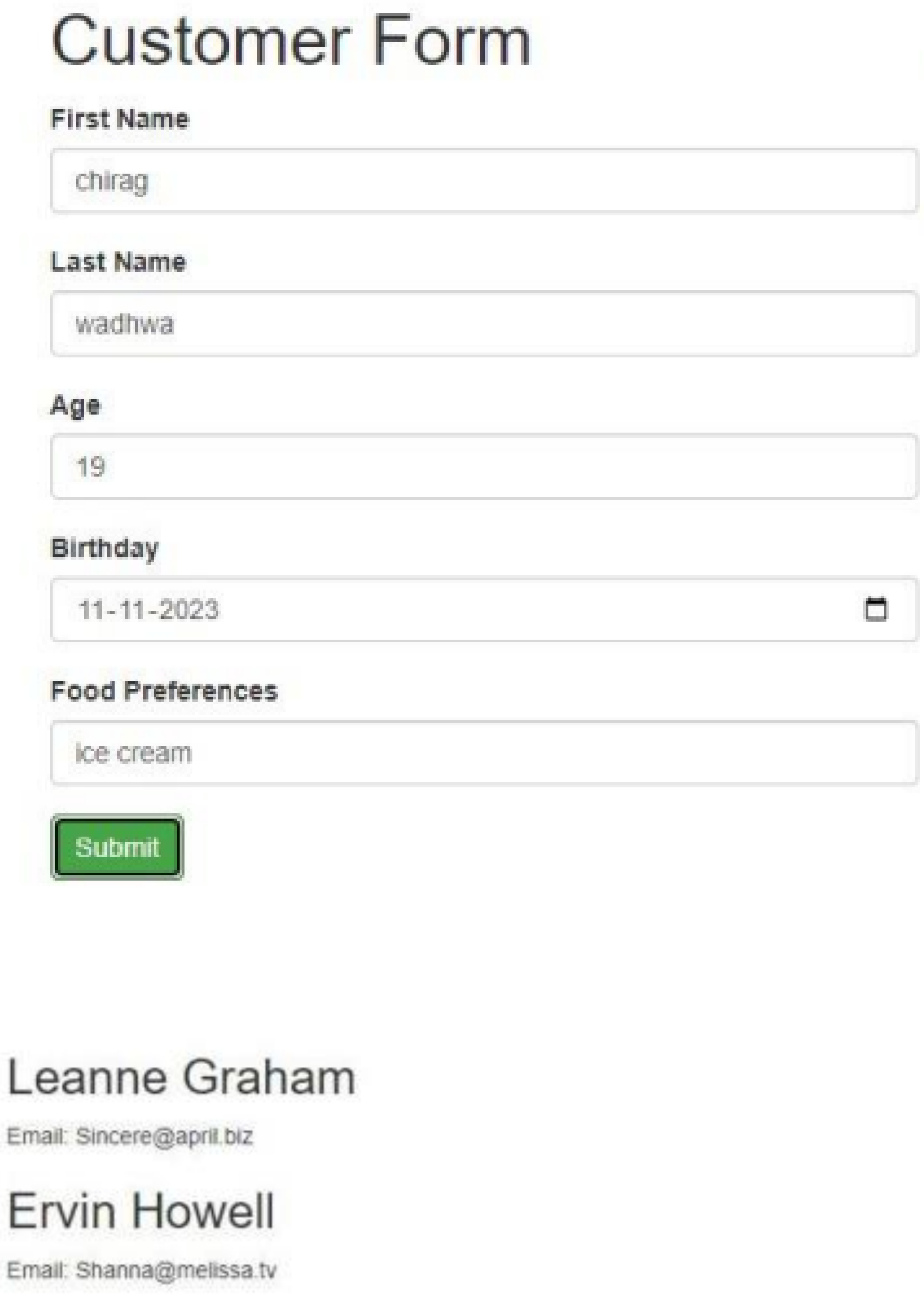
```
Hello World, This is my Node.js server
```

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

```
  <head>
    <title>Customer Form</title>
    <link
      rel="stylesheet"
      href="//netdna.bootstrapcdn.com/bootstrap/3.0.3/css/bootstrap.min.css"
    />
    <script src="//ajax.googleapis.com/ajax/libs/jquery/2.0.3/jquery.min.js"></script>
  </head>
  <body>
    <div class="col-sm-6 col-sm-offset-3">
      <h1>Customer Form</h1>
      <form id="customerForm">
        <div class="form-group">
          <label for="firstName">First Name</label>
          <input
            type="text"
            class="form-control"
            id="firstName"
            placeholder="First Name"
          />
        </div>
        <div class="form-group">
          <label for="lastName">Last Name</label>
          <input
            type="text"
            class="form-control"
            id="lastName"
            placeholder="Last Name"
          />
        </div>
        <div class="form-group">
          <label for="age">Age</label>
          <input
            type="number"
            class="form-control"
            id="age"
            placeholder="Age"
          />
        </div>
        <div class="form-group">
          <label for="birthday">Birthday</label>
          <input type="date" class="form-control" id="birthday" />
        </div>
        <div class="form-group">
          <label for="foodPreferences">Food Preferences</label>
          <input
            type="text"
            class="form-control"
            id="foodPreferences"
            placeholder="Food Preferences"
```



```
          />
        </div>
        <button type="submit" class="btn btn-success">Submit</button>
      </form>
    </div>
    <script>
      $(document).ready(function () {
        $("#customerForm").submit(function (e) {
          e.preventDefault();
          // Create JavaScript Object const
          customer = {
            firstName: $("#firstName").val(),
            lastName: $("#lastName").val(),
            age: $("#age").val(),
            birthday: $("#birthday").val(),
            foodPreferences: $("#foodPreferences").val(),
          };
          // Create JSON Object const customerJSON =
          JSON.stringify(customer);
          // Print objects to console
          console.log("JavaScript Object:", customer);
          console.log("JSON Object:", customerJSON);
        });
      });
      $.ajax({
        url: "https://jsonplaceholder.typicode.com/users", //JSONPlaceholder URL
        type: "GET",
        dataType: "json",
        success: function (data) {
          for (let i = 0; i < 2; i++) {
            // Display the data of the first two customers
            const customer = data[i];
            const customerDiv = $("<div></div>").addClass("customer");
            customerDiv.append($("<h2></h2>").text(customer.name));
            customerDiv.append($("<p></p>").text("Email: " + customer.email));
            $("body").append(customerDiv); // Add the customer div to the body
          }
        },
        error: function (jqXHR, textStatus, errorThrown) {
          console.log("AJAX Error:", textStatus);
        },
      });
    </script>
  </body>
</html>
```



Customer Form

First Name

chirag

Last Name

wadhwa

Age

19

Birthday

11-11-2023

Food Preferences

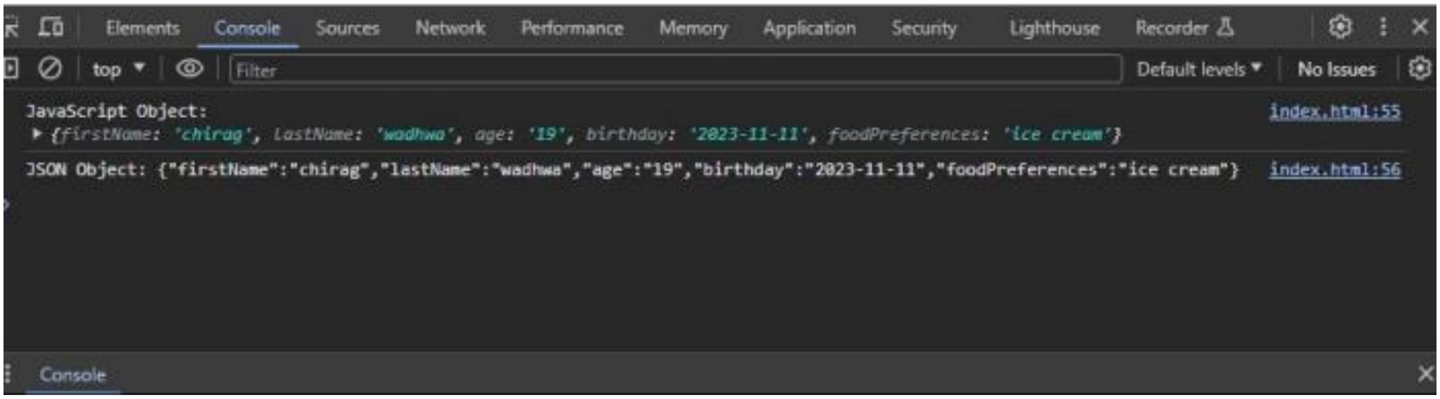
ice cream

Leanne Graham

Email: Sincere@april.biz

Ervin Howell

Email: Shanna@melissa.tv



```
JavaScript Object:                                                                                              index.html:55
 ▸ {firstName: 'chirag', lastName: 'wadhwa', age: '19', birthday: '2023-11-11', foodPreferences: 'ice cream'}
JSON Object: {"firstName":"chirag","lastName":"wadhwa","age":"19","birthday":"2023-11-11","foodPreferences":"ice cream"}  index.html:56
```