NAME:PRITAM MANDAL

COURSE:COMPUTER SCIENCE HONOURS

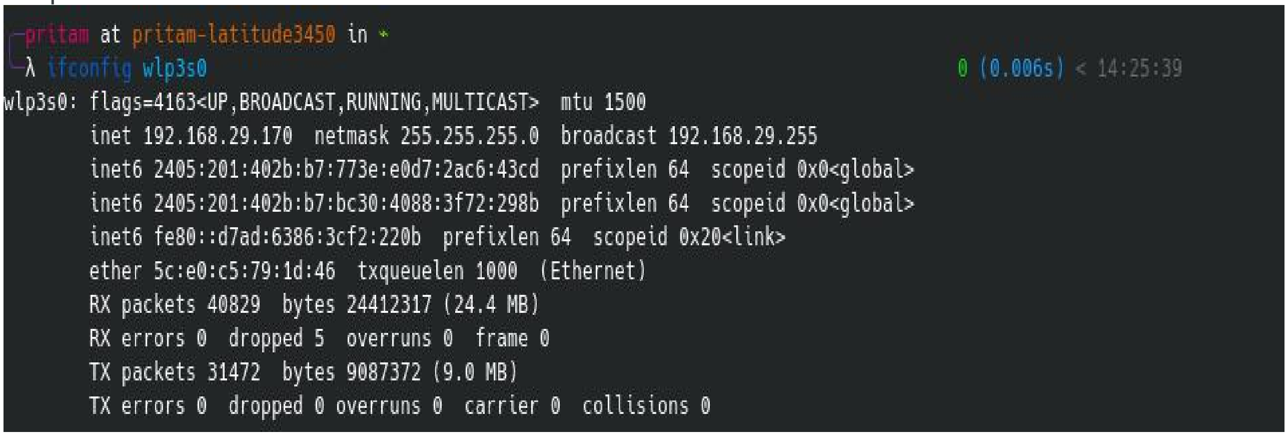
SUBJECT:INTERNET TECHNOLOGIES

ROLL NUMBER:21/18043

SUBMITTED TO: UMA MAM

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.

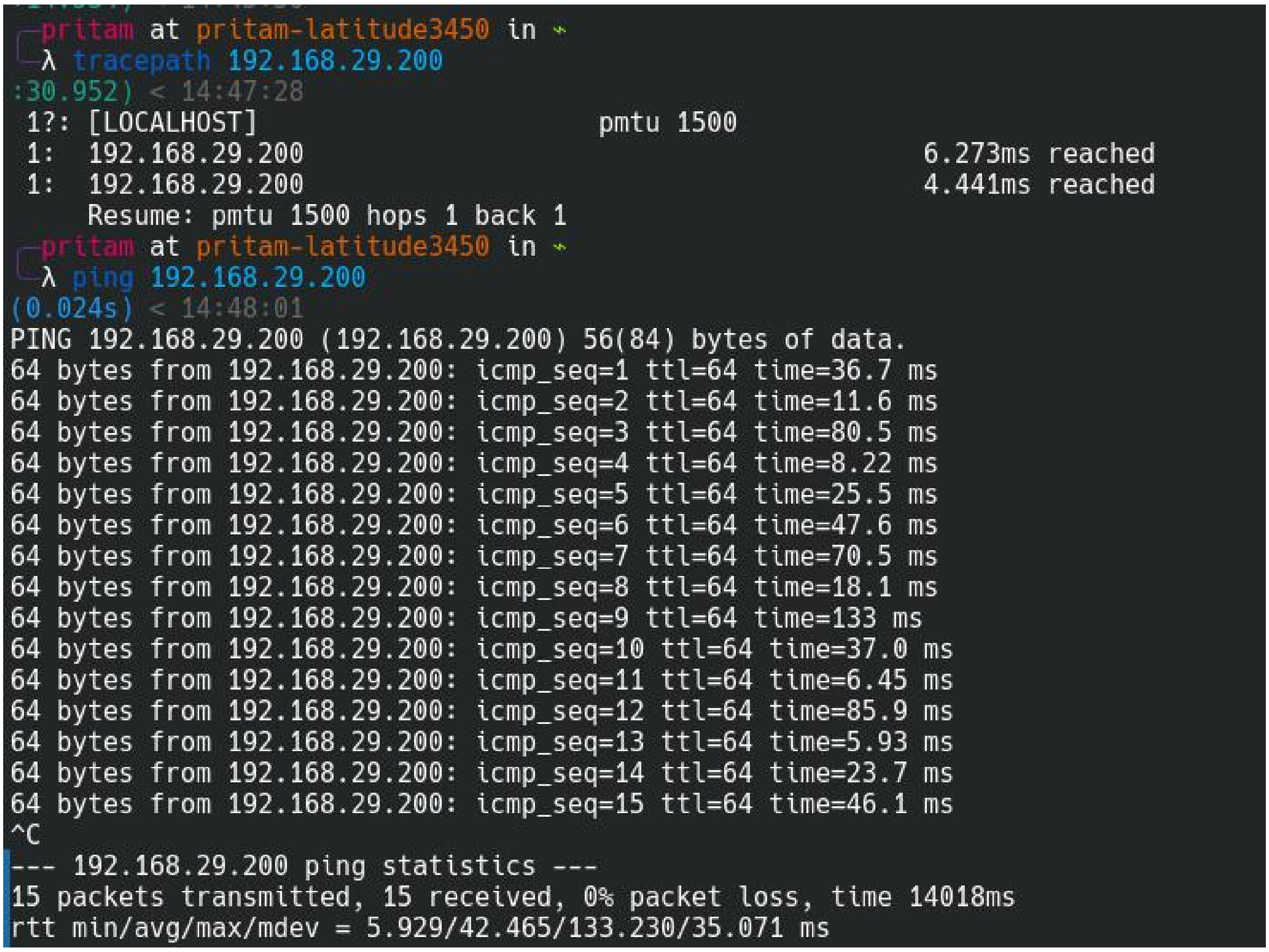
   Output:



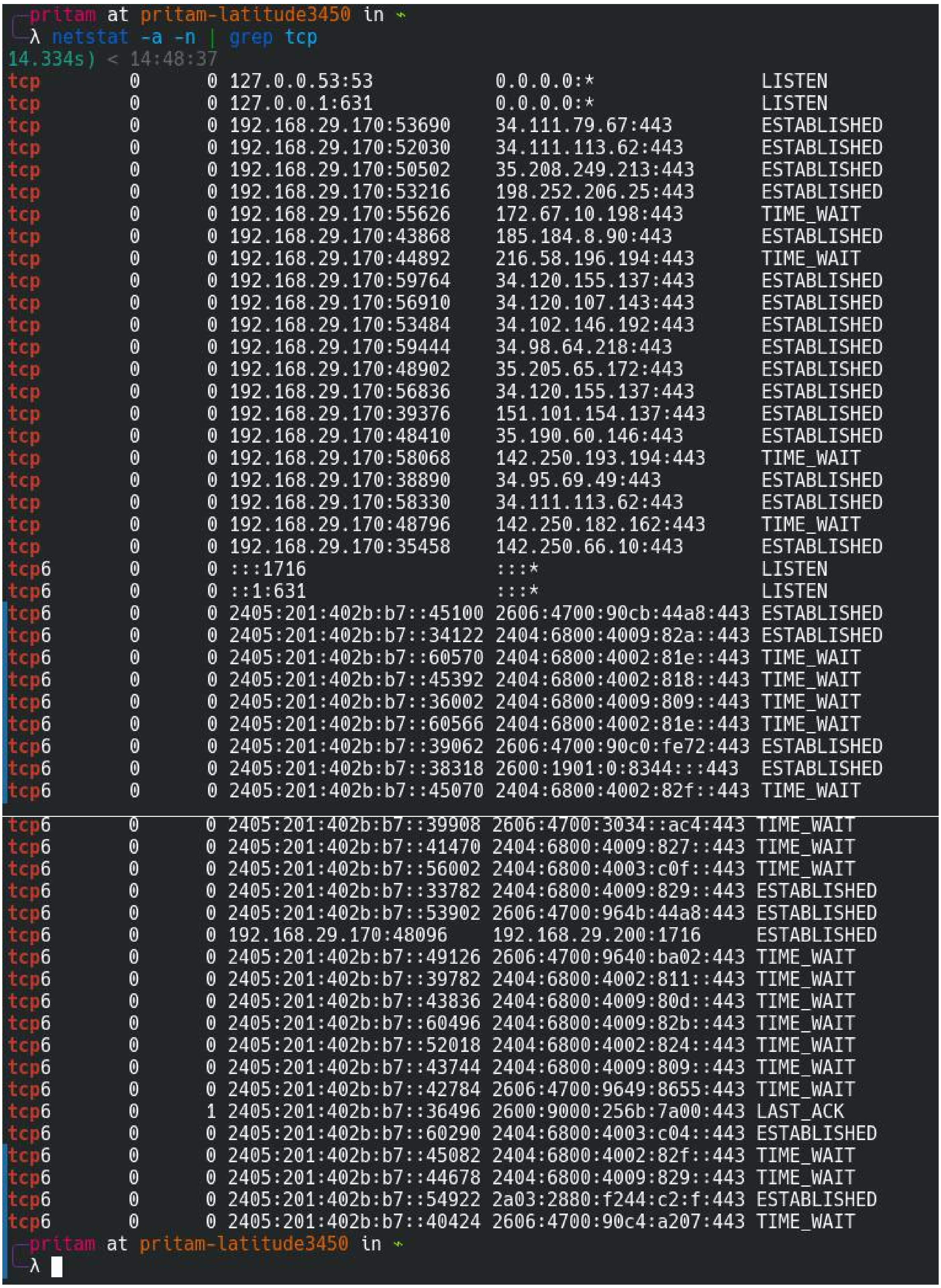
```
╭─pritam at pritam-latitude3450 in ~
╰─λ ifconfig wlp3s0                                                    0 (0.006s) < 14:25:39
wlp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.29.170  netmask 255.255.255.0  broadcast 192.168.29.255
        inet6 2405:201:402b:b7:773e:e0d7:2ac6:43cd  prefixlen 64  scopeid 0x0<global>
        inet6 2405:201:402b:b7:bc30:4088:3f72:298b  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::d7ad:6386:3cf2:220b  prefixlen 64  scopeid 0x20<link>
        ether 5c:e0:c5:79:1d:46  txqueuelen 1000  (Ethernet)
        RX packets 40829  bytes 24412317 (24.4 MB)
        RX errors 0  dropped 5  overruns 0  frame 0
        TX packets 31472  bytes 9087372 (9.0 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

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.

   Output:



```
╭─pritam at pritam-latitude3450 in ~
╰─λ tracepath 192.168.29.200
:30.952) < 14:47:28
 1?: [LOCALHOST]                        pmtu 1500
 1:  192.168.29.200                                         6.273ms reached
 1:  192.168.29.200                                         4.441ms reached
     Resume: pmtu 1500 hops 1 back 1
╭─pritam at pritam-latitude3450 in ~
╰─λ ping 192.168.29.200
(0.024s) < 14:48:01
PING 192.168.29.200 (192.168.29.200) 56(84) bytes of data.
64 bytes from 192.168.29.200: icmp_seq=1 ttl=64 time=36.7 ms
64 bytes from 192.168.29.200: icmp_seq=2 ttl=64 time=11.6 ms
64 bytes from 192.168.29.200: icmp_seq=3 ttl=64 time=80.5 ms
64 bytes from 192.168.29.200: icmp_seq=4 ttl=64 time=8.22 ms
64 bytes from 192.168.29.200: icmp_seq=5 ttl=64 time=25.5 ms
64 bytes from 192.168.29.200: icmp_seq=6 ttl=64 time=47.6 ms
64 bytes from 192.168.29.200: icmp_seq=7 ttl=64 time=70.5 ms
64 bytes from 192.168.29.200: icmp_seq=8 ttl=64 time=18.1 ms
64 bytes from 192.168.29.200: icmp_seq=9 ttl=64 time=133 ms
64 bytes from 192.168.29.200: icmp_seq=10 ttl=64 time=37.0 ms
64 bytes from 192.168.29.200: icmp_seq=11 ttl=64 time=6.45 ms
64 bytes from 192.168.29.200: icmp_seq=12 ttl=64 time=85.9 ms
64 bytes from 192.168.29.200: icmp_seq=13 ttl=64 time=5.93 ms
64 bytes from 192.168.29.200: icmp_seq=14 ttl=64 time=23.7 ms
64 bytes from 192.168.29.200: icmp_seq=15 ttl=64 time=46.1 ms
^C
--- 192.168.29.200 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14018ms
rtt min/avg/max/mdev = 5.929/42.465/133.230/35.071 ms
```



```
╭─pritam at pritam-latitude3450 in ~
╰─λ netstat -a -n | grep tcp
14.334s) < 14:48:37
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 192.168.29.170:53690    34.111.79.67:443        ESTABLISHED
tcp        0      0 192.168.29.170:52030    34.111.113.62:443       ESTABLISHED
tcp        0      0 192.168.29.170:50502    35.208.249.213:443      ESTABLISHED
tcp        0      0 192.168.29.170:53216    198.252.206.25:443      ESTABLISHED
tcp        0      0 192.168.29.170:55626    172.67.10.198:443       TIME_WAIT
tcp        0      0 192.168.29.170:43868    185.184.8.90:443        ESTABLISHED
tcp        0      0 192.168.29.170:44892    216.58.196.194:443      TIME_WAIT
tcp        0      0 192.168.29.170:59764    34.120.155.137:443      ESTABLISHED
tcp        0      0 192.168.29.170:56910    34.120.107.143:443      ESTABLISHED
tcp        0      0 192.168.29.170:53484    34.102.146.192:443      ESTABLISHED
tcp        0      0 192.168.29.170:59444    34.98.64.218:443        ESTABLISHED
tcp        0      0 192.168.29.170:48902    35.205.65.172:443       ESTABLISHED
tcp        0      0 192.168.29.170:56836    34.120.155.137:443      ESTABLISHED
tcp        0      0 192.168.29.170:39376    151.101.154.137:443     ESTABLISHED
tcp        0      0 192.168.29.170:48410    35.190.60.146:443       ESTABLISHED
tcp        0      0 192.168.29.170:58068    142.250.193.194:443     TIME_WAIT
tcp        0      0 192.168.29.170:38890    34.95.69.49:443         ESTABLISHED
tcp        0      0 192.168.29.170:58330    34.111.113.62:443       ESTABLISHED
tcp        0      0 192.168.29.170:48796    142.250.182.162:443     TIME_WAIT
tcp        0      0 192.168.29.170:35458    142.250.66.10:443       ESTABLISHED
tcp6       0      0 :::1716                 :::*                    LISTEN
tcp6       0      0 ::1:631                 :::*                    LISTEN
tcp6       0      0 2405:201:402b:b7::45100 2606:4700:90cb:44a8:443 ESTABLISHED
tcp6       0      0 2405:201:402b:b7::34122 2404:6800:4009:82a::443 ESTABLISHED
tcp6       0      0 2405:201:402b:b7::60570 2404:6800:4002:81e::443 TIME_WAIT
tcp6       0      0 2405:201:402b:b7::45392 2404:6800:4002:818::443 TIME_WAIT
tcp6       0      0 2405:201:402b:b7::36002 2404:6800:4009:809::443 TIME_WAIT
tcp6       0      0 2405:201:402b:b7::60566 2404:6800:4002:81e::443 TIME_WAIT
tcp6       0      0 2405:201:402b:b7::39062 2606:4700:90c0:fe72:443 ESTABLISHED
tcp6       0      0 2405:201:402b:b7::38318 2600:1901:0:8344:::443  ESTABLISHED
tcp6       0      0 2405:201:402b:b7::45070 2404:6800:4002:82f::443 TIME_WAIT
tcp6       0      0 2405:201:402b:b7::39908 2606:4700:3034::ac4:443 TIME_WAIT
tcp6       0      0 2405:201:402b:b7::41470 2404:6800:4009:827::443 TIME_WAIT
tcp6       0      0 2405:201:402b:b7::56002 2404:6800:4003:c0f::443 TIME_WAIT
tcp6       0      0 2405:201:402b:b7::33782 2404:6800:4009:829::443 ESTABLISHED
tcp6       0      0 2405:201:402b:b7::53902 2606:4700:964b:44a8:443 ESTABLISHED
tcp6       0      0 192.168.29.170:48096    192.168.29.200:1716     ESTABLISHED
tcp6       0      0 2405:201:402b:b7::49126 2606:4700:9640:ba02:443 TIME_WAIT
tcp6       0      0 2405:201:402b:b7::39782 2404:6800:4002:811::443 TIME_WAIT
tcp6       0      0 2405:201:402b:b7::43836 2404:6800:4009:80d::443 TIME_WAIT
tcp6       0      0 2405:201:402b:b7::60496 2404:6800:4009:82b::443 TIME_WAIT
tcp6       0      0 2405:201:402b:b7::52018 2404:6800:4002:824::443 TIME_WAIT
tcp6       0      0 2405:201:402b:b7::43744 2404:6800:4009:809::443 TIME_WAIT
tcp6       0      0 2405:201:402b:b7::42784 2606:4700:9649:8655:443 TIME_WAIT
tcp6       0      1 2405:201:402b:b7::36496 2600:9000:256b:7a00:443 LAST_ACK
tcp6       0      0 2405:201:402b:b7::60290 2404:6800:4003:c04::443 ESTABLISHED
tcp6       0      0 2405:201:402b:b7::45082 2404:6800:4002:82f::443 TIME_WAIT
tcp6       0      0 2405:201:402b:b7::44678 2404:6800:4009:829::443 TIME_WAIT
tcp6       0      0 2405:201:402b:b7::54922 2a03:2880:f244:c2:f:443 ESTABLISHED
tcp6       0      0 2405:201:402b:b7::40424 2606:4700:90c4:a207:443 TIME_WAIT
╭─pritam at pritam-latitude3450 in ~
╰─λ █
```

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.

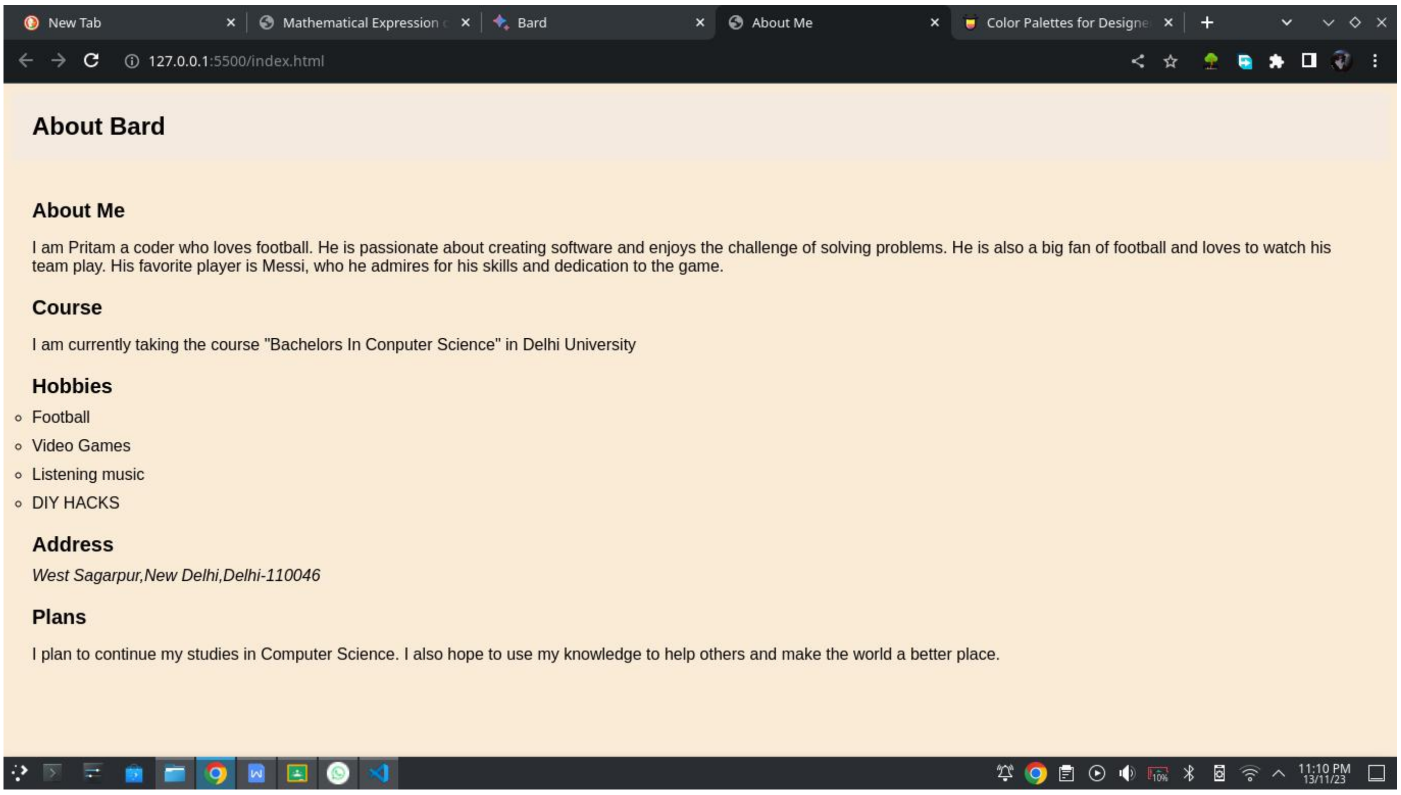
CODE:

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>About Me</title>
<style>
body {
font-family: sans-serif;
background-color: antiquewhite;
}

header {
background-color: #F4EAE0;
padding: 20px;
}

h1 {
font-size: 24px;
margin: 0;
}

main {
padding: 20px;
}

section {
margin-bottom: 20px;
}

h2 {
font-size: 20px;
margin-bottom: 10px;
}

ul {
list-style-type:circle;
margin: 0;
padding: 0;
}

li {
margin-bottom: 10px;
}
</style>
</head>
<body>
<header>
<h1>About Bard</h1>
</header>
```

```
    <main>
    <section id="about">
    <h2>About Me</h2>
    <p>I am Pritam a coder who loves football. He is passionate about creating software and enjoys the challenge
of solving problems. He is also a big fan of football and loves to watch his team play. His favorite player
is Messi, who he admires for his skills and dedication to the game.</p>
    </section>

    <section id="course">
    <h2>Course</h2>
    <p>I am currently taking the course "Bachelors In Conputer Science" in Delhi University</p>
    </section>

    <section id="hobbies">
    <h2>Hobbies</h2>
    <ul>
    <li>Football</li>
    <li>Video Games</li>
    <li>Listening music</li>
    <li>DIY HACKS</li>
    </ul>
    </section>

    <section id="address">
    <h2>Address</h2>
    <address>West Sagarpur,New Delhi,Delhi-110046</address>
    </section>

    <section id="plans">
    <h2>Plans</h2>
    <p>I plan to continue my studies in Computer Science. I also hope to use my knowledge to help others
and make the world a better place.</p>
    </section>
    </main>
    </body>
    </html>
```

OUTPUT:



New Tab | Mathematical Expression | Bard | About Me | Color Palettes for Designe

127.0.0.1:5500/index.html

About Bard

About Me

I am Pritam a coder who loves football. He is passionate about creating software and enjoys the challenge of solving problems. He is also a big fan of football and loves to watch his team play. His favorite player is Messi, who he admires for his skills and dedication to the game.

Course

I am currently taking the course "Bachelors In Conputer Science" in Delhi University

Hobbies

- Football
- Video Games
- Listening music
- DIY HACKS

Address

West Sagarpur,New Delhi,Delhi-110046

Plans

I plan to continue my studies in Computer Science. I also hope to use my knowledge to help others and make the world a better place.

11:10 PM 13/11/23

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. Use clearInterval() function to stop the given task.

CODE:

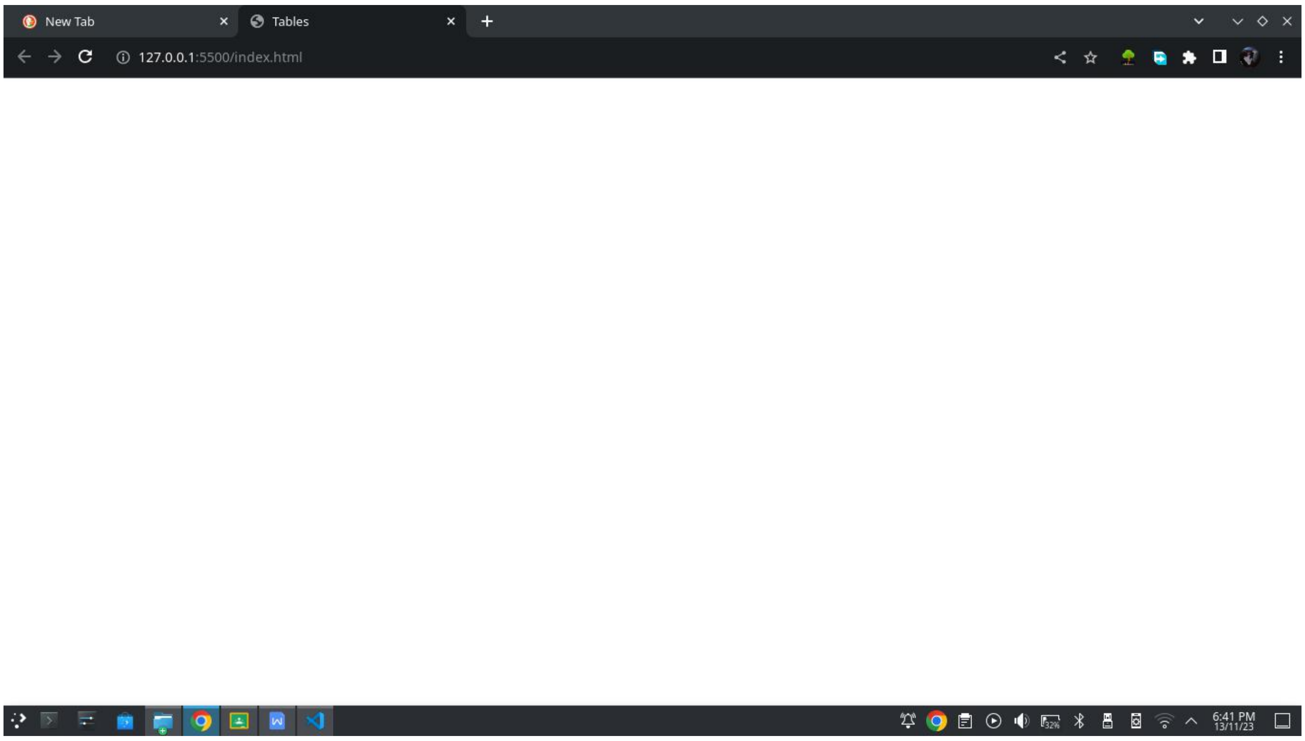
Index.html

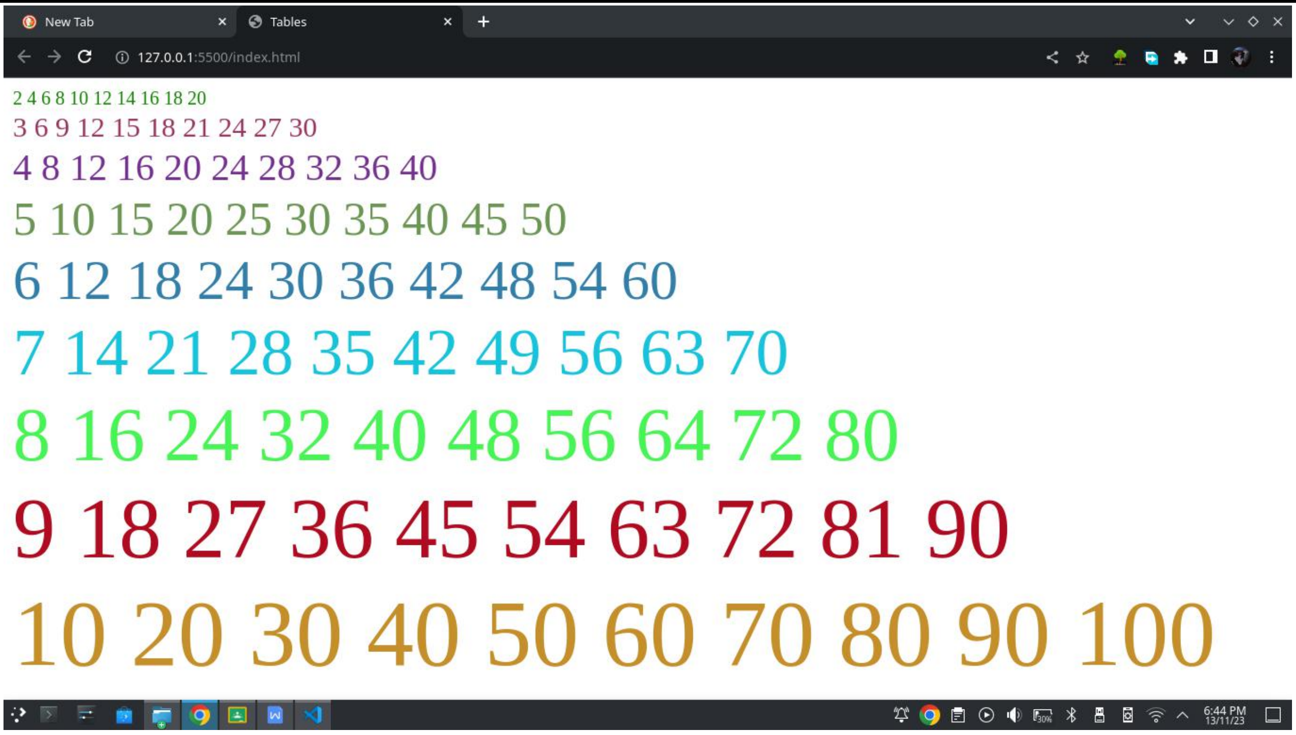
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Tables</title>
</head>
<body>
<table>
<tr class ="table"></tr>
<tr class ="table"></tr>
<tr class ="table"></tr>
<tr class ="table"></tr>
<tr class ="table"></tr>
<tr class ="table"></tr>
<tr class ="table"></tr>
<tr class ="table"></tr>
<tr class ="table"></tr>
</table>
<script src="./index.js"></script>
</body>
</html>
```

Index.js

```
var counter=2
var para =document.querySelectorAll(".table");
var numberOfPara = para.length;
var rowCount=0;
var fs=20;
function printRow(){
var randomColor=Math.floor(Math.random()*16777215).toString(16)
para[rowCount].style.fontSize=fs+'px'
para[rowCount].style.color="#"+randomColor
for(var i=1;i<=10;i++){
para[rowCount].textContent=para[rowCount].textContent+" "+(counter*i);
}
counter++;
fs=fs+10;
rowCount++;
if(rowCount>=numberOfPara){
clearInterval(intervalIdWithLimit)
}
}
printRow()
const intervalIdWithLimit = setInterval(printRow,5000)
```

OUTPUT:





New Tab
Tables
127.0.0.1:5500/index.html

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

6:44 PM
13/11/23

5. Expla setInterval function and setTimeout function with the help of an example.

Ans) **setInterval()** is a javascript method used to schedule the execution of a function at a specific interval until the interval id generated by the setInterval() is cleared.

Code Segment:

```
let timerId = setInterval(function() {
   alert('This alert will appear every 2 seconds');
}, 2000);
```

This will generate a alert every two second until the timerId is cleared.

setTimeout() is a javascript method used to schedule the execution of a function after a specific amount of time but will only run single time.

Code Segment:

```
setTimeout(function() {
   alert('This alert will appear after 5 seconds');
}, 5000);
```

This will generate a alert only a single time but after 5 seconds of delay

6. 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.

CODE:

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>Practical5</title>
<link rel="stylesheet" href="./style.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Young+Serif&display=swap" rel="stylesheet">
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Pixelify+Sans&display=swap" rel="stylesheet">
</head>
<body>
<p id="paragraph">Lorem ipsum dolor sit amet consectetur adipisicing elit.
Saepe ea deleniti cumque omnis, vero ullam amet fuga perspiciatis
dignissimos aut qui debitis
eaque eum reprehenderit odit placeat, asperiores esse quibusdam?</p>
<div id="flex">
<div class="button-grid">
<button class ="color">Red</button>
<button class ="color">Green</button>
<button class ="color">Blue</button>
<button class ="font-size">10</button>
<button class ="font-size">20</button>
<button class ="font-size">30</button>
<button class ="font-family">Courier New</button>
<button class ="font-family">Pixelify Sans</button>
<button class ="font-family">Young Serif</button>
</div>
</div>
<script src="./index.js"></script>
</body>
</html>
```

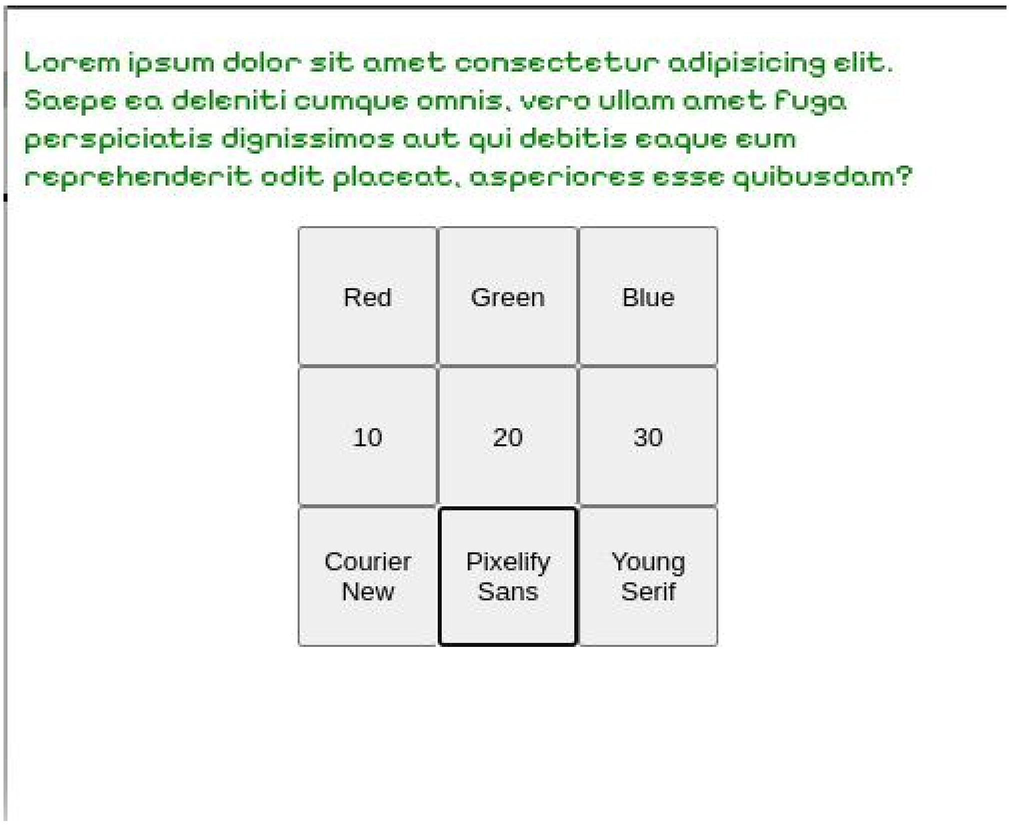
Index.js

```javascript
$("button.color").on("click",function(){
$("p").css("color",$(this).text())
})
$("button.font-size").on("click",function(){
$("p").css("fontSize",$(this).text()+"px")
})
$("button.font-family").on("click",function(){
$("p").css("font-family","'"+$(this).text()+"'")
})
```

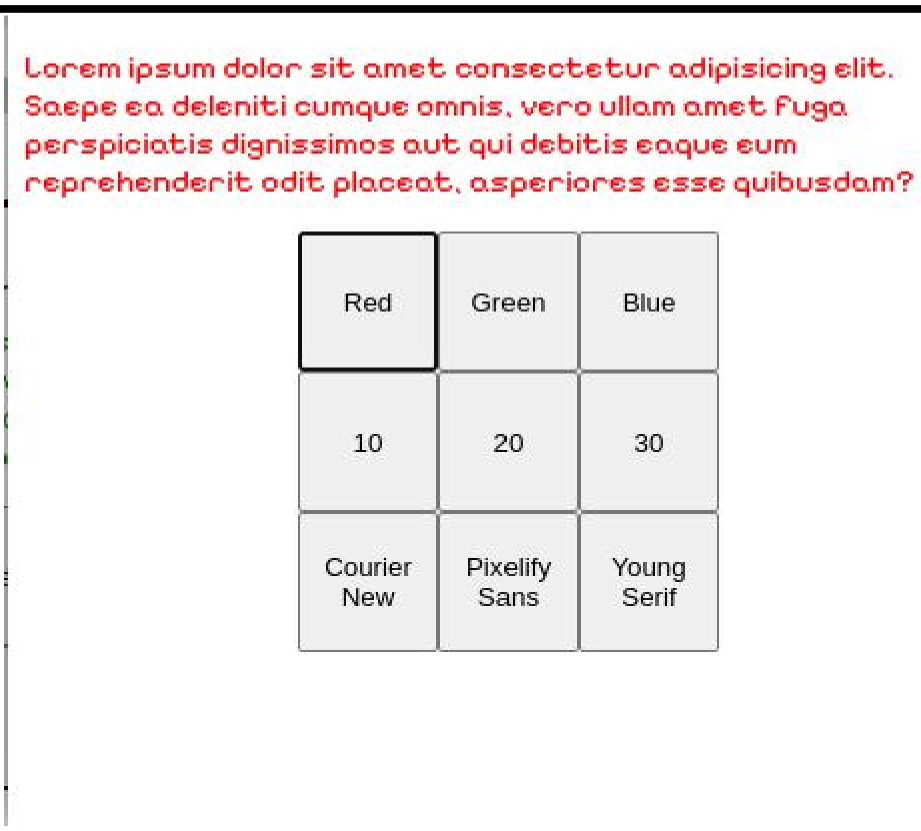
style.css

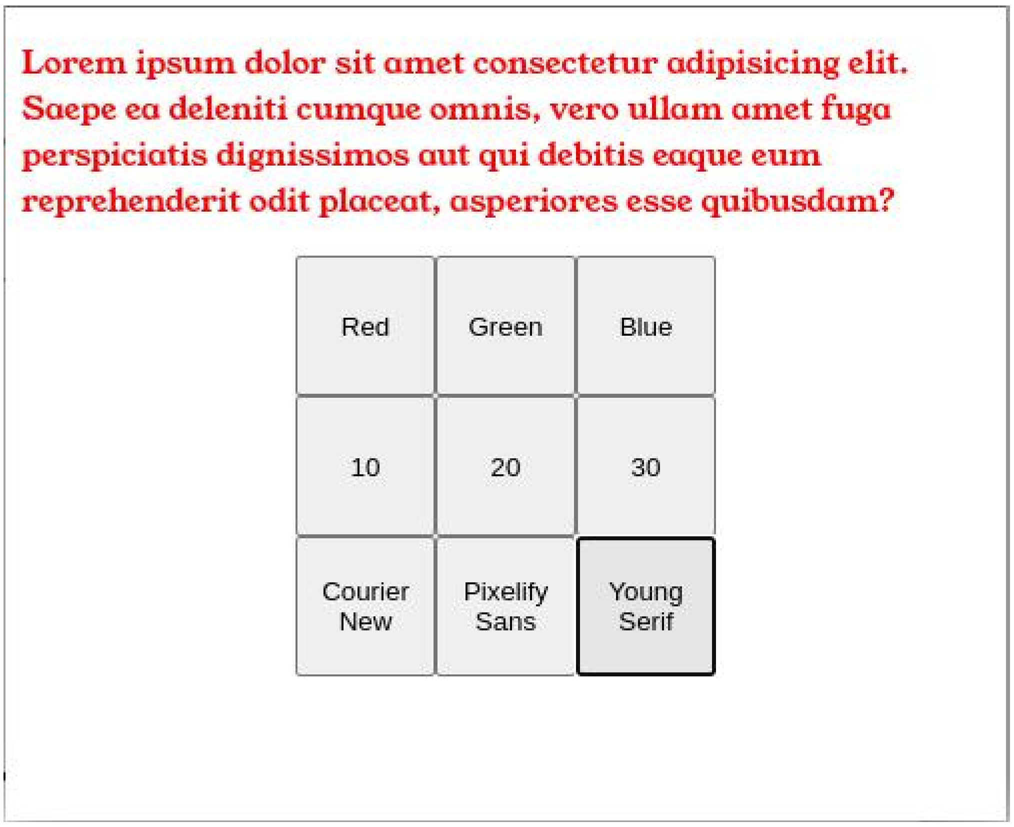
```css
.button-grid{
display: grid;
grid-template-columns: repeat(3,70px);
grid-template-rows: repeat(3,70px);
}
#flex{
display: flex;
align-items: center;
justify-content: center;
/* font-family: 'Courier New', 'Franklin Gothic Medium', 'Gill Sans', 'Gill Sans MT',; */
}
.red{
color: red;
}
```

OUTPUT:



Lorem ipsum dolor sit amet consectetur adipisicing elit. Saepe ea deleniti cumque omnis, vero ullam amet fuga perspiciatis dignissimos aut qui debitis eaque eum reprehenderit odit placeat, asperiores esse quibusdam?


Lorem ipsum dolor sit amet consectetur adipisicing elit. Saepe ea deleniti cumque omnis, vero ullam amet fuga perspiciatis dignissimos aut qui debitis eaque eum reprehenderit odit placeat, asperiores esse quibusdam?


Lorem ipsum dolor sit amet consectetur adipisicing elit. Saepe ea deleniti cumque omnis, vero ullam amet fuga perspiciatis dignissimos aut qui debitis eaque eum reprehenderit odit placeat, asperiores esse quibusdam?

7. 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.

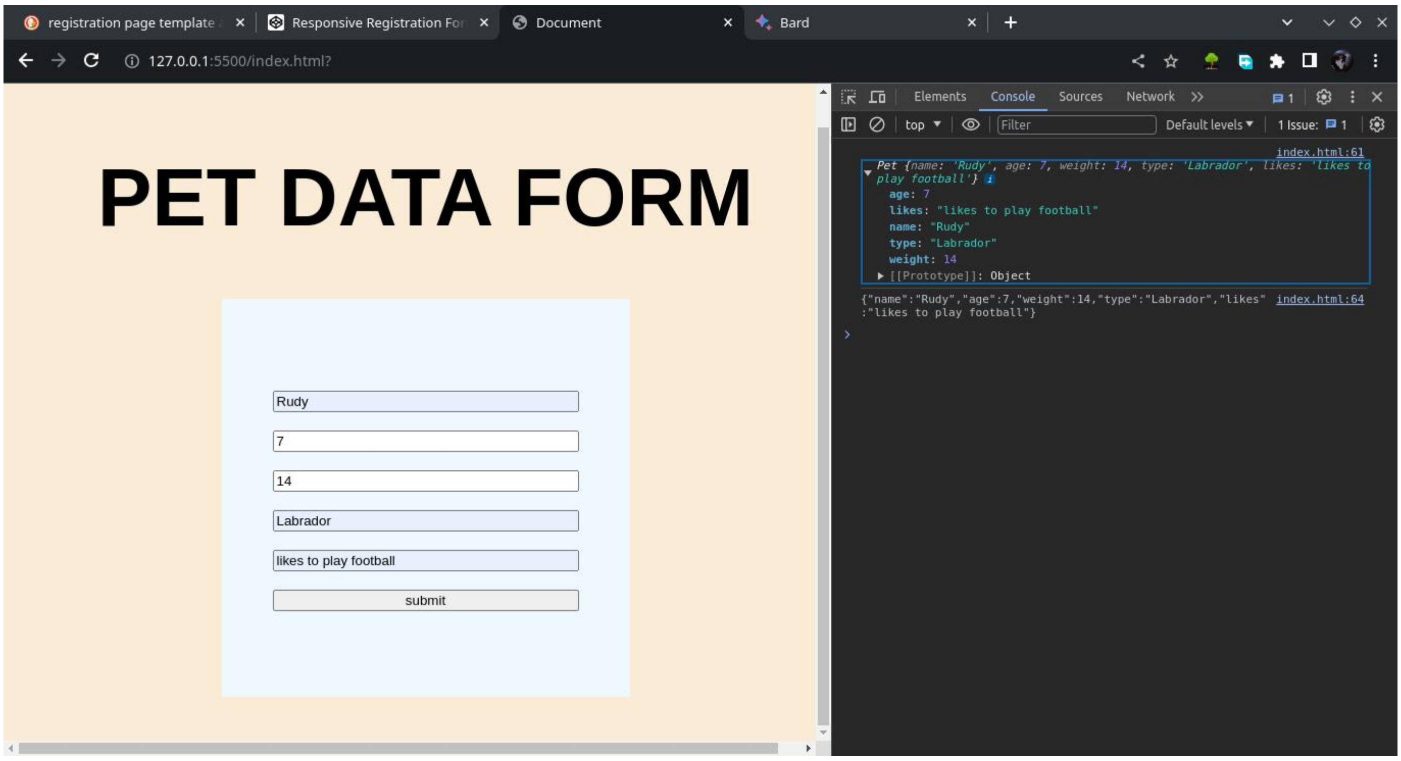
CODE:

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
<style>
body{
display: flex;
justify-content: center;
align-items: center;
height: 100vh;
width: 100vw;
background-color: antiquewhite;
flex-direction: column;
}
form{
display: flex;
flex-direction: column;

}
.container{
background-color: aliceblue;
height: 300px;
width: 300px;
padding-left: 50px;
padding-right:50px ;
padding-top: 90px;
}
h1{
font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;
font-size: 5rem;
}
</style>
</head>
<body>
<h1>PET DATA FORM</h1>
<div class="container">
<form id="petForm">
<input type="text" id="name" placeholder="Pet's Name"><br>
<input type="number" id="age" placeholder="Pet's Age"><br>
<input type="number" id="weight" placeholder="Pet's Weights(KG)"><br>
<input type="text" id="type" placeholder="Type"><br>
<input id="likings" type="text" placeholder="Linkings"><br>
<button type="submit">submit</button>

</form>
</div>
<script>
```

```
const petForm = document.getElementById('petForm')
petForm.addEventListener('submit', (event) => {
event.preventDefault();

const petName = document.getElementById('name').value;
const petAge = parseInt(document.getElementById('age').value);
const petWeight = parseFloat(document.getElementById('weight').value);
const petType = document.getElementById('type').value;
const petLikes = document.getElementById('likings').value;

const pet = new Pet(petName, petAge, petWeight, petType,petLikes);
console.log(pet);

const petJSON = JSON.stringify(pet);
console.log(petJSON);
});

class Pet {
constructor(name, age, weight, type, likes) {
this.name = name;
this.age = age;
this.weight = weight;
this.type = type;
this.likes = likes;
}
}
</script>
</body>
</html>
```

OUTPUT:



127.0.0.1:5500/index.html?

PET DATA FORM

Rudy
7
14
Labrador
likes to play football
submit

```
                                                                  index.html:61
Pet {name: 'Rudy', age: 7, weight: 14, type: 'Labrador', likes: 'likes to
play football'}
   age: 7
   likes: "likes to play football"
   name: "Rudy"
   type: "Labrador"
   weight: 14
 ▶ [[Prototype]]: Object
{"name":"Rudy","age":7,"weight":14,"type":"Labrador","likes"   index.html:64
:"likes to play football"}
```

8. 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.

CODE:

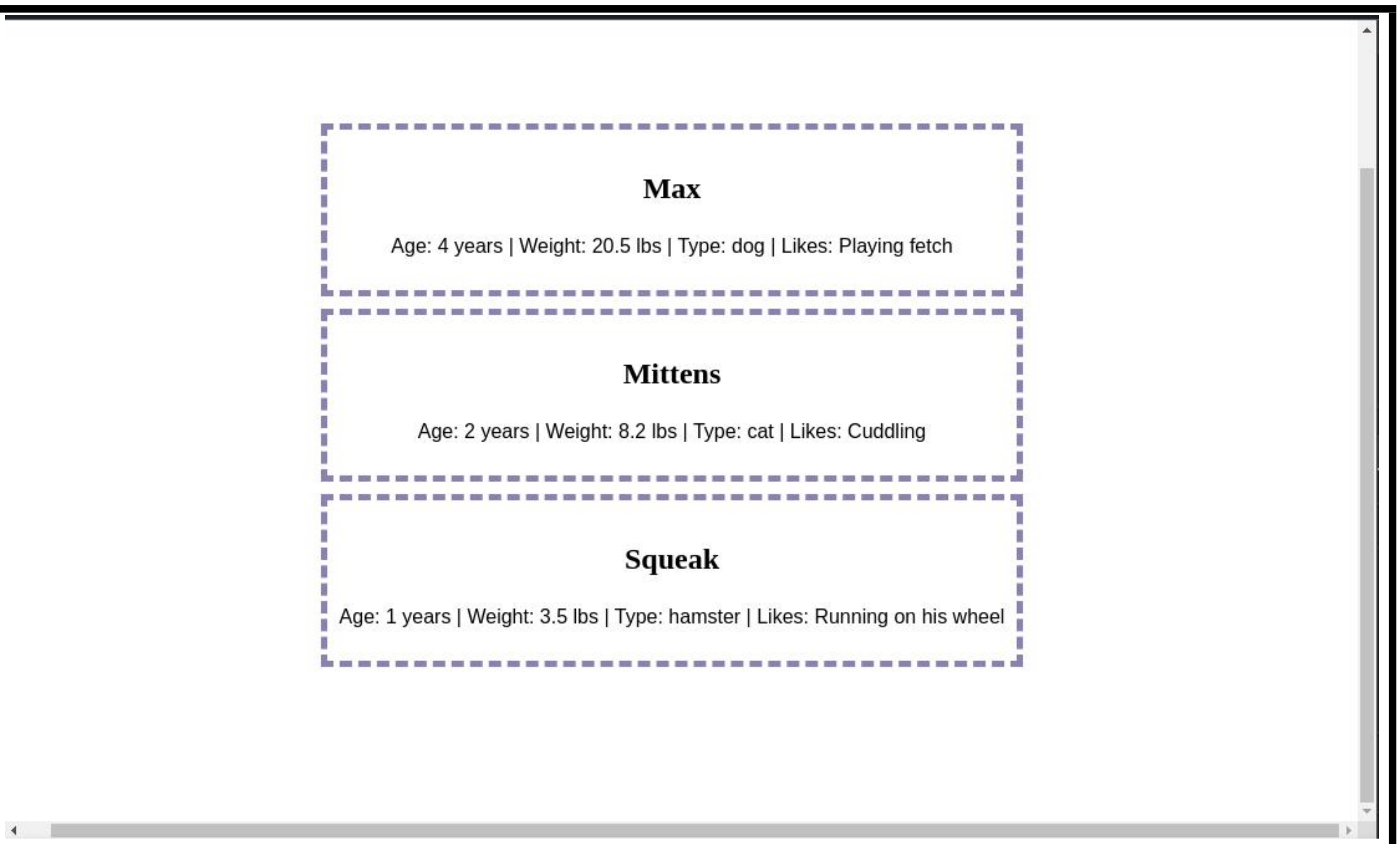
Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Pet Data</title>
<style>
body {
font-family: sans-serif;
padding-top: 100px;
display: flex;
flex-direction: column-reverse;
height: 100vh;
width: 100vw;
justify-content: center;
align-items: center;
}

.pet {
border: 5px dashed rgb(138, 130, 177);
padding: 10px;
margin-bottom: 10px;
text-align: center;
}

.pet-name {
font-weight: bold;
font-family:'Times New Roman', Times, serif;
font-size: x-large;
}
</style>
</head>
<body>
<div id="pets"></div>

<script>
const xhr = new XMLHttpRequest();
xhr.open('GET', 'pets.json');
xhr.onload = function() {
if (xhr.status === 200) {
const pets = JSON.parse(xhr.responseText);

const petList = document.getElementById('pets');
for (const pet of pets) {
const petElement = document.createElement('div');
petElement.classList.add('pet');

const petNameElement = document.createElement('h3');
petNameElement.classList.add('pet-name');
petNameElement.textContent = pet.name;
petElement.appendChild(petNameElement);

const petDetailsElement = document.createElement('p');
petDetailsElement.textContent = `Age: ${pet.age} years`;
petDetailsElement.textContent += ` | Weight: ${pet.weight} lbs`;
petDetailsElement.textContent += ` | Type: ${pet.type}`;
petDetailsElement.textContent += ` | Likes: ${pet.likes}`;
petElement.appendChild(petDetailsElement);

petList.appendChild(petElement);
}
} else {
console.error('Error loading JSON data');
}
};
xhr.send();
</script>
</body>
</html>
```

pets.json

```
[
{
"name": "Max",
"age": 4,
"weight": 20.5,
"type": "dog",
"likes": "Playing fetch"
},
{
"name": "Mittens",
"age": 2,
"weight": 8.2,
"type": "cat",
"likes": "Cuddling"
},
{
"name": "Squeak",
"age": 1,
"weight": 3.5,
"type": "hamster",
"likes": "Running on his wheel"
}
]
```

OUTPUT:



Max

Age: 4 years | Weight: 20.5 lbs | Type: dog | Likes: Playing fetch

Mittens

Age: 2 years | Weight: 8.2 lbs | Type: cat | Likes: Cuddling

Squeak

Age: 1 years | Weight: 3.5 lbs | Type: hamster | Likes: Running on his wheel

9. 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.

CODE:

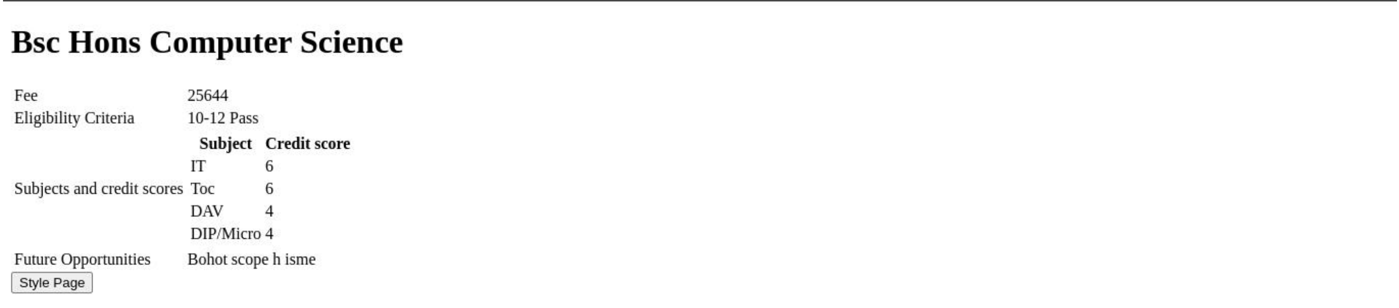
Index.html

```
<!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>Fee</td>
<td>25644</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>Bohot scope h isme</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:

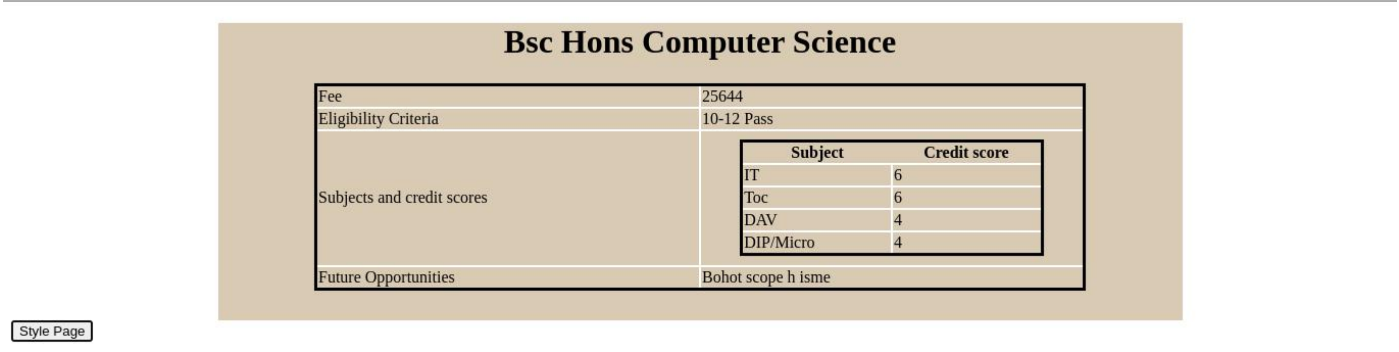


Bsc Hons Computer Science

Fee	25644
Eligibility Criteria	10-12 Pass
Subjects and credit scores	(nested table below)
Future Opportunities	Bohot scope h isme

Subject	Credit score
IT	6
Toc	6
DAV	4
DIP/Micro	4

Style Page



Bsc Hons Computer Science

Fee	25644
Eligibility Criteria	10-12 Pass
Subjects and credit scores	(nested table below)
Future Opportunities	Bohot scope h isme

Subject	Credit score
IT	6
Toc	6
DAV	4
DIP/Micro	4

Style Page

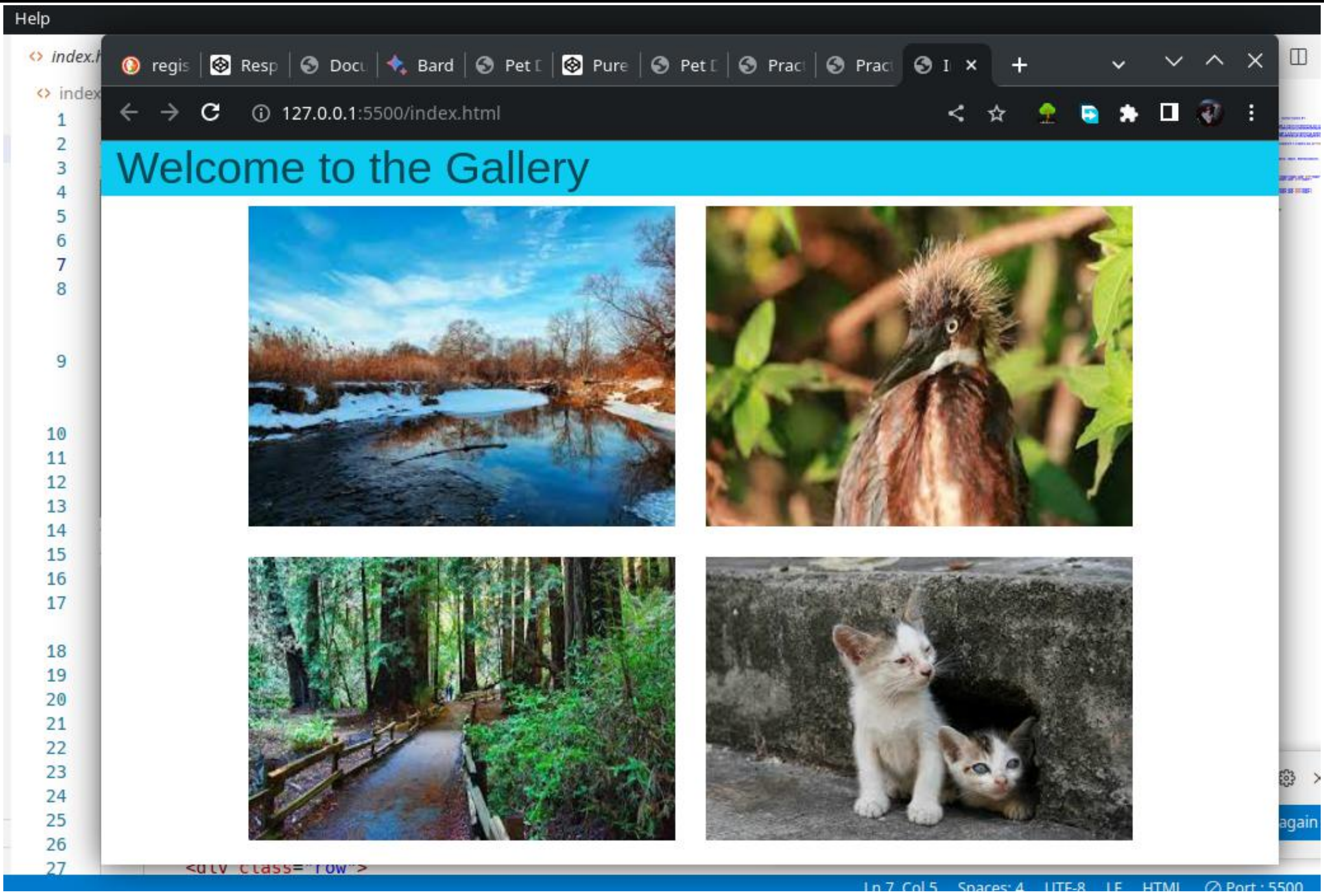
10. Create an HTML page for an image gallery, which shows the use of BOOTSTRAP to rearrange and resize its contents on resizing the browser.

CODE:

Index.html

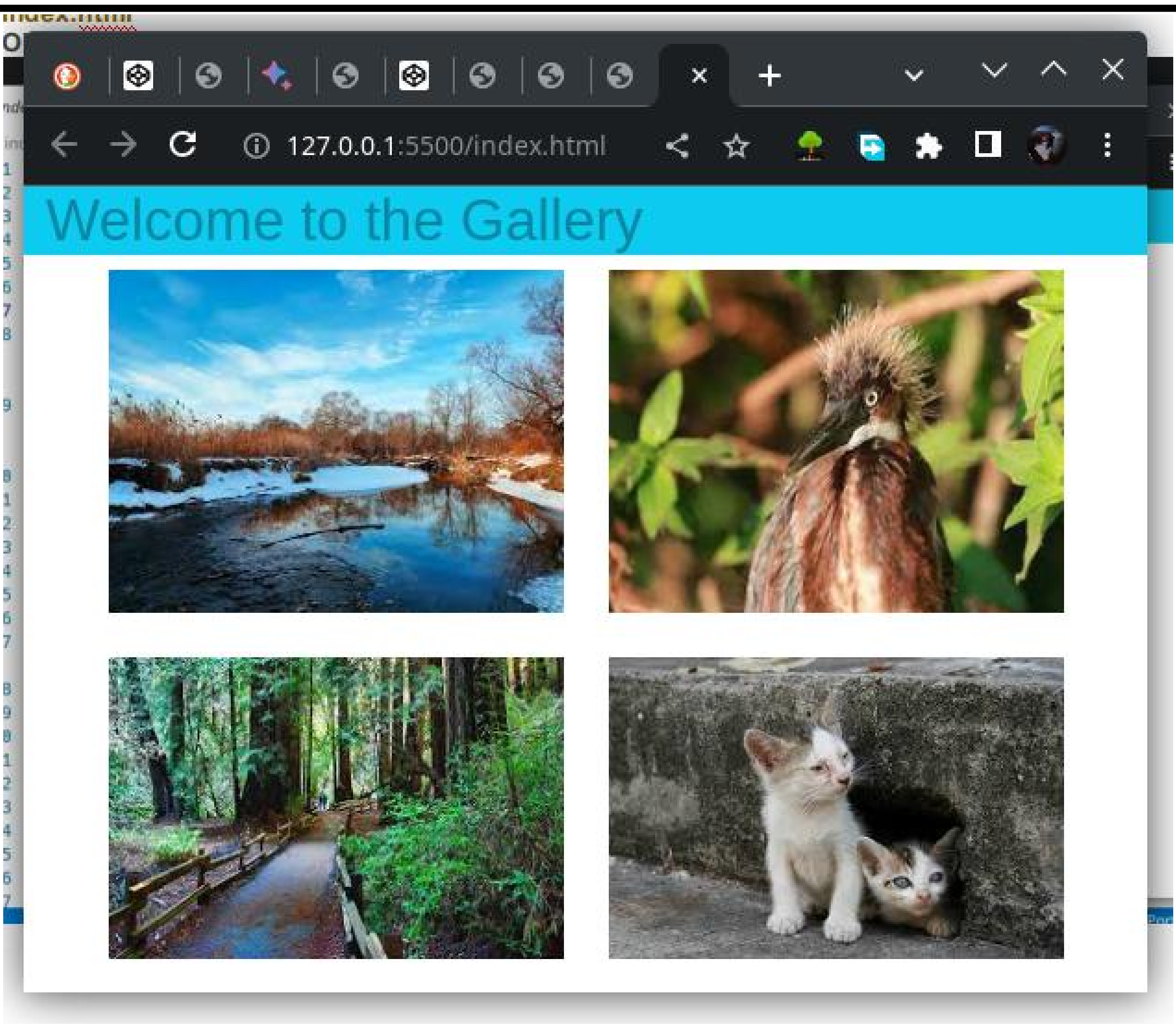
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Image Gallery</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>


</head>
<body>
<div class="container-fluid text-bg-info">
<h1 class="header text-centerw" style="font-family: Impact, Haettenschweiler, 'Arial Narrow Bold', sans-serif;">
Welcome to the Gallery
</h1>
</div>
<div class="container">
<div class="row">
<img class="img-flu col-6" src="./assests/images/image1.jpeg" alt="image1">
<img class="col-6" src="./assests/images/image2.jpeg" alt="image2">
</div>
<br/>
<div class="row">
<img class="col-6" src="./assests/images/image3.jpeg" alt="image3">
<img class="col-6" src="./assests/images/image4.jpeg" alt="image4">
</div>
</div>

<script>
$(document).ready(()=>{
setInterval(()=>{
$(".header").fadeIn(2000).fadeOut(2000)
},1000)
})
</script>
</body>
</html>
```

OUTPUT:



Welcome to the Gallery

```
<div class="row">
```



Welcome to the Gallery

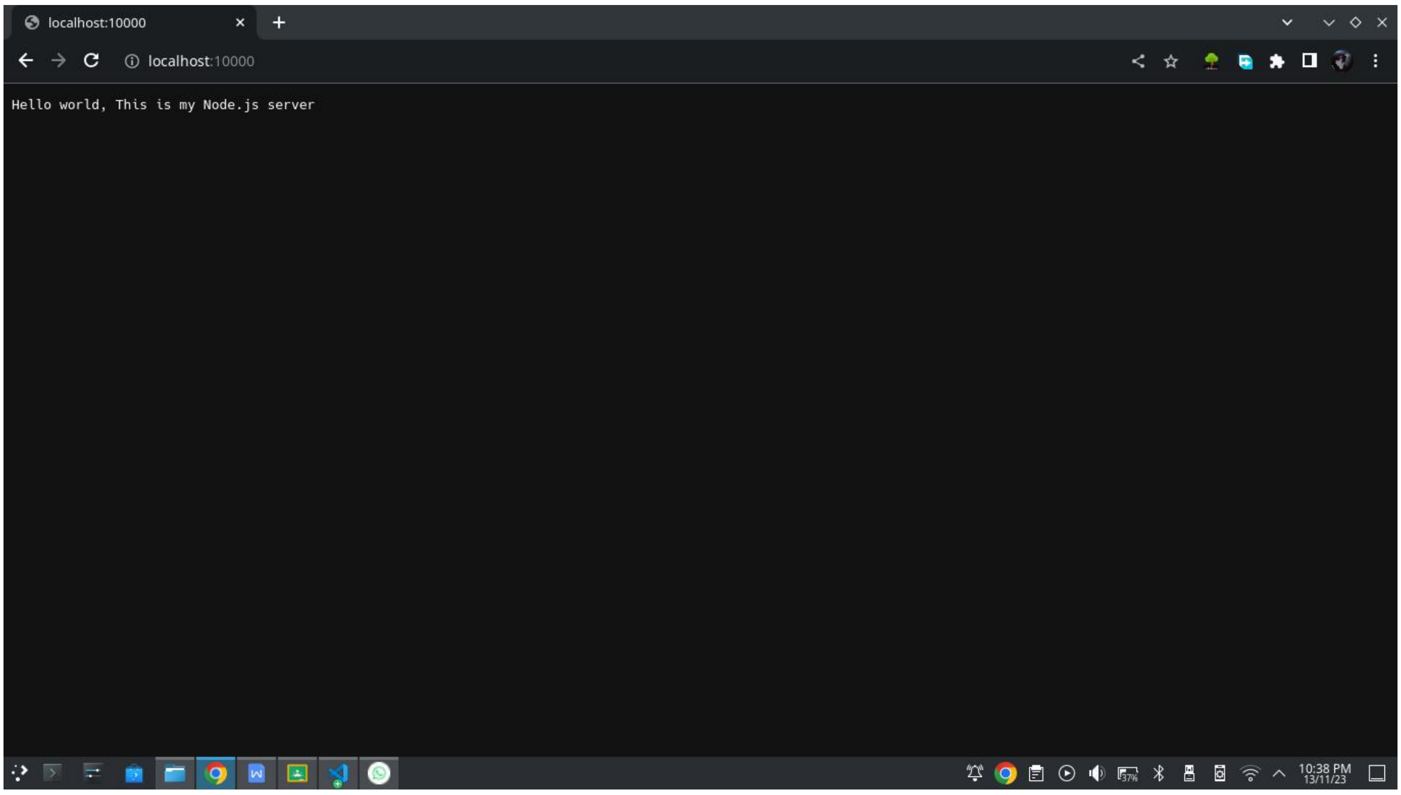
11. Create an HTTP server using Node.js, which handles requests on port 10000, or a free port beyond 10000. Modify the server in such a way that opening localhost:10000 will display “Hello world, This is my Node.js server” on browser.

CODE:

Index.js

```
const http = require('http');

// Create an HTTP server
const server = http.createServer((req, res) => {
res.writeHead(200, { 'Content-Type': 'text/plain' });
res.end('Hello world, This is my Node.js server\n');
});

// Listen on port 10000 or a free port beyond 10000
const port = 10000;
server.listen(port, () => {
console.log(`Server is running at http://localhost:${port}/`);
});
```

OUTPUT:



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

12. 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 (use Node.js for database connectivity with mysql).

CODE:

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Practical 12</title>
<style>
body{
display: inline-flex;
flex-direction: column;
justify-content: center;
align-items: center;
height: 100vh;
width: 100vw;
}
.signin{
padding: 20px;
margin: auto;
background-color: rgb(182, 182, 182);
box-shadow: 1px 1px 1px 1px;
}
.signup{
padding: 20px;
```

```html
margin:auto;
background-color: rgb(182, 182, 182);
box-shadow: 1px 1px 1px 1px;

}
button{
padding: 10px;
border-radius: 45%;
border-style:groove;
margin-top: 10px;
font-weight: bold;
}
</style>
</head>
<body>
<!-- Sign IN -->
<div class="signin">
<h2>Sign In</h2>
<form action="/signin" method="post">
<label for="username">Username: </label>
<input name="username" type="text">
<br/>
<br/>
<label for="password">Password: </label>
<input name="password" type="password">
<br/>
<button type="submit">SignIn</button>
</form>
</div>
<br/>
<br/>
<!-- Sign Up -->
<div class="signup">
<h2>Sign Up</h2>
<form action="/signup" method="post">
<label for="username">Username: </label>
<input name="username" type="text">
<br/>
<br/>
<label for="password">Password: </label>
<input name="password" type="password">
<br/>
<button type="submit">SignUp</button>
</form>
</div>

</body>
</html>
```

welcome.html

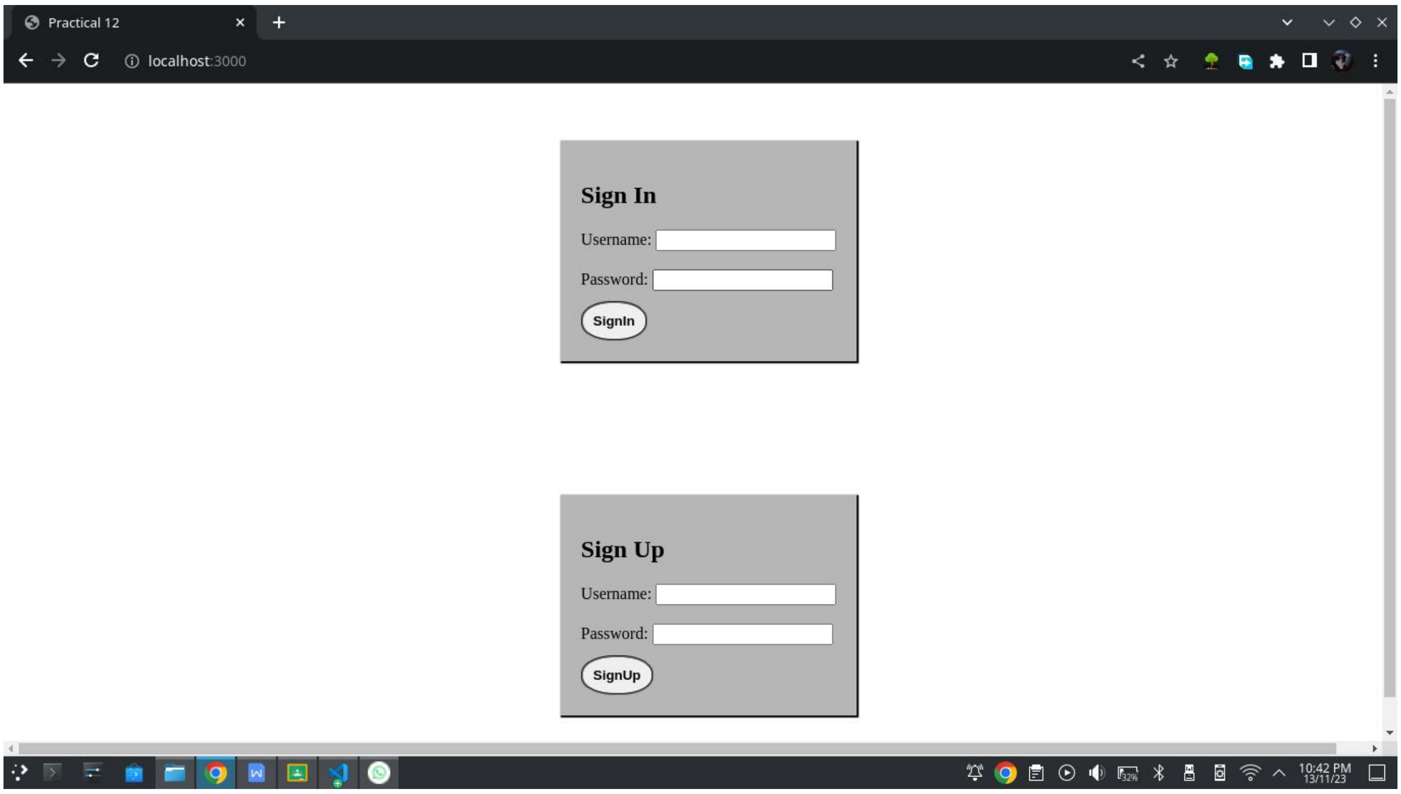
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Welcome</title>
<style>
@keyframes gradientAnimation {
0% {
background-position: 0% 50%;
}
50% {
background-position: 100% 50%;
}
100% {
background-position: 0% 50%;
}
}

h1,h3,body {
font-size: 3em;
background: linear-gradient(45deg, #ff6b6b, #b1f2b2, #6b6bff, #f2b266);
background-size: 400% 400%;
color: transparent;
-webkit-background-clip: text;
background-clip: text;
animation: gradientAnimation 10s ease infinite;
}
div{
position: relative;
left: 17%;
top:10%;
}
</style>
</head>
<body>
<div style="background-color: white;text-align: center;display: inline-block;padding-left: 
10px;padding-right: 10px;">
<h1 style="font-size: 8rem;font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida 
Sans Unicode', Geneva, Verdana, sans-serif;">Welcome User</h1>
<h3 style="font-family:Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;font-size: 
3rem;padding-top: 10px;margin-top:10px;">You logged in successfully</h3>
</div>
</body>
</html>
```

script.js

```
const http=require('http')
const mysql=require('mysql')
const URL=require('url')
const fs = require('fs')
const db=mysql.createConnection({
host:'localhost',
user:'root',
password:'',
database:'assignment'
})
db.connect((err)=>{
if (err) console.log('Could not connect')
else{
console.log("Connection successful")}
})
const server = http.createServer((req,res)=>{
const url_path=URL.parse(req.url,true)
const pathname= url_path.pathname
if(req.method==="GET" && pathname==="/"){
fs.readFile("./index.html",(err,data)=>{
if(err){
res.writeHead(500);
res.end("Internal Server Error");
}
else{
res.writeHead(200, { "Content-Type": "text/html" });
res.end(data);
}
})
}
if(req.method==="GET" && pathname==="/welcome"){
fs.readFile("./welcome.html",(err,data)=>{
if(err){
res.writeHead(500);
res.end("Internal Server Error");
}
else{
res.writeHead(200, { "Content-Type": "text/html" });
res.end(data);
}
})
}
if(req.method==="POST" && pathname==="/signup"){
let body=''
req.on('data',(chunk)=>{
body+=chunk
})
req.on('end',()=>{
var formData = new URLSearchParams(body)
var name = formData.get('username')
var password=formData.get('password')
db.query(`Insert into users(username,password) Values('${name}','${password}')`,(err)=>{
```

```
if(err){
console.log(err)
res.writeHead(500).end("Error inserting the vale")
}
else{
console.log("Data insserted successfully")
res.writeHead(302,{"location":"/"})
res.end()
}
})
})
}
if(req.method==="POST" && pathname==="/signin"){
let body=''
req.on("data",(chunk)=>{
body+=chunk
})
req.on('end',()=>{
var formData=new URLSearchParams(body);
var name = formData.get('username')
var password = formData.get('password')
console.log(name," ",password)
db.query(`Select * from users where username ='${name}' and password='${password}'`,(err,results)=>{
if(err){
console.log("cant find records")
res.writeHead(404)
res.end("Cant find any records for the username and password")
}
else if(results.length>1){
console.log("Invalid users")
res.writeHead("500")
res.end("Invalid user")
}
else{
console.log("recod find successfully")
res.writeHead(302,{"location":"/welcome"})
res.end()
}
})
})
}
})
server.listen(3000,(err)=>{
if(err) console.log("Error Occured")
else{ console.log(`website running in port 3000`)}
})
```

OUTPUT:



Sign In

Username:

Password:

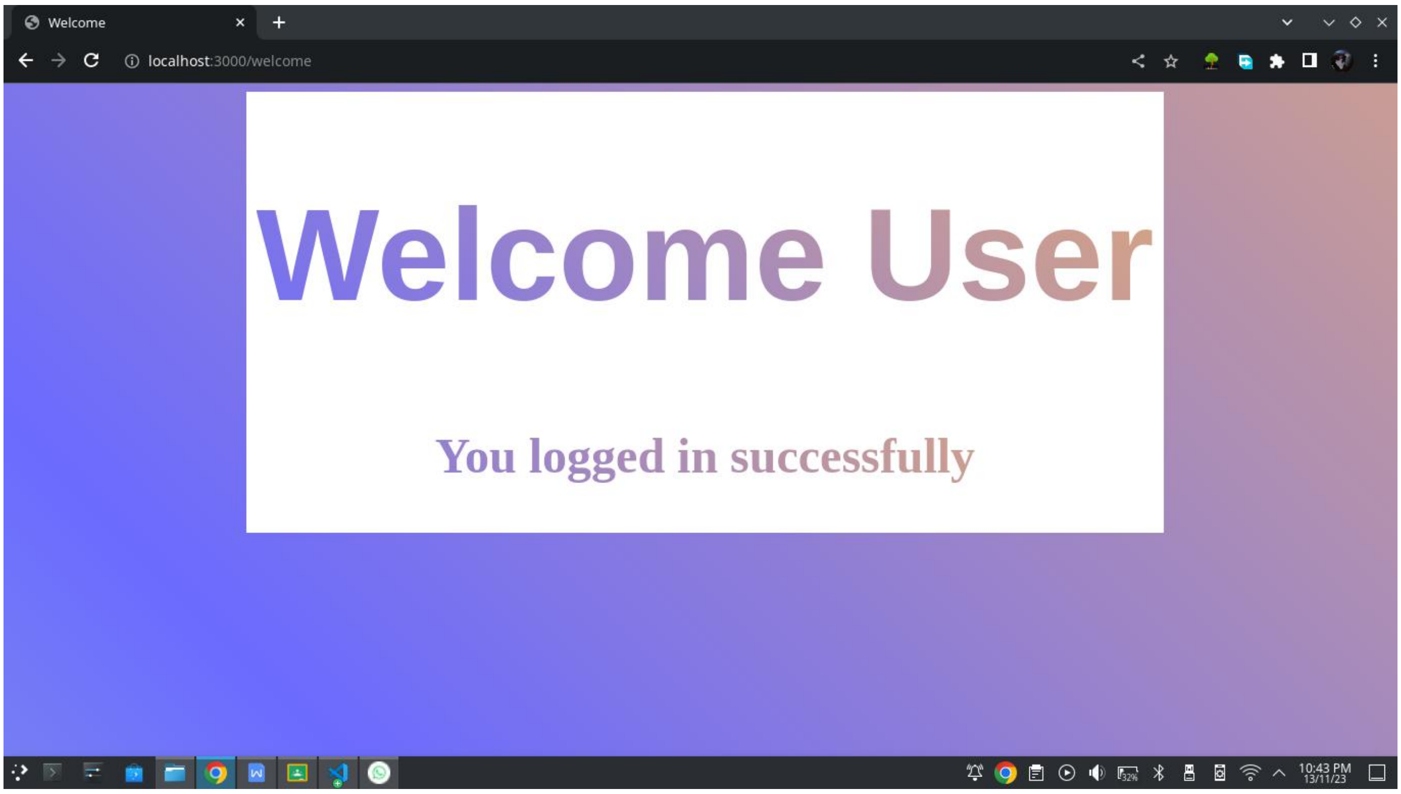
SignIn

Sign Up

Username:

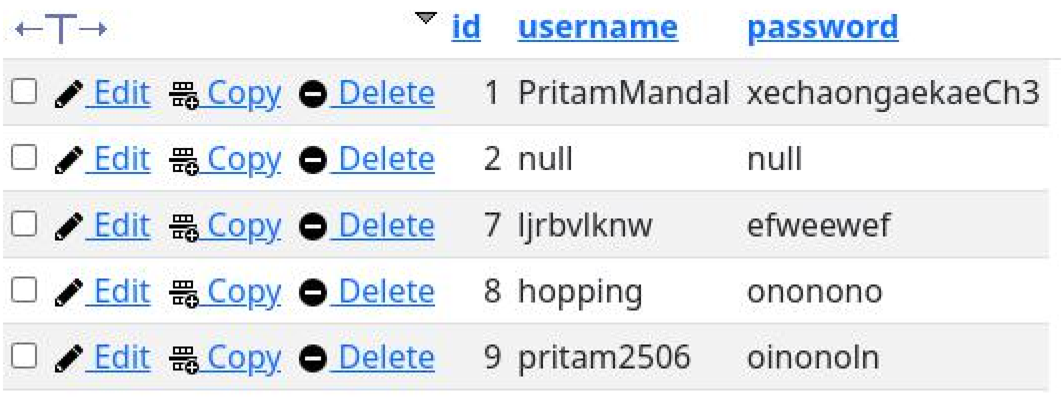
Password:

SignUp



Welcome User

You logged in successfully



	id	username	password
☐ Edit Copy Delete	1	PritamMandal	xechaongaekaeCh3
☐ Edit Copy Delete	2	null	null
☐ Edit Copy Delete	7	ljrbvlknw	efweewef
☐ Edit Copy Delete	8	hopping	ononono
☐ Edit Copy Delete	9	pritam2506	oinonoln

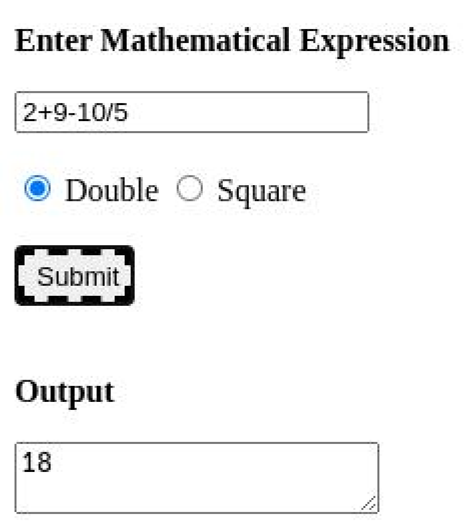
13. Create an HTML page with one input field, one radio button and a text field for display. The first input field will take a mathematical expression as input. The two radio buttons will be displayed as SQUARE and DOUBLE. The user selects whichever option, the result of the mathematical expression as entered by the user, will be squared or doubled and the corresponding answer should be displayed in the text field.

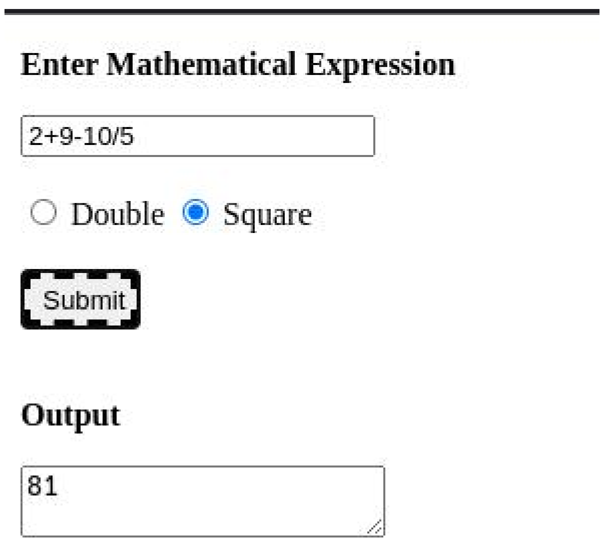
CODE:

Index.html

```
 <!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Mathematical Expression calculator</title>
<style>
p{
font-weight: 900;
}
</style>
</head>
<body>
<p>Enter Mathematical Expression</p>
<input type="text" id="mathermaticalexpression" placeholder="Add Mathematical expression">
<br>
<br>
<input type="radio" id="calculatordouble" value="double" name="operation">
<label for="calculatordouble">Double</label>
<input type="radio" id="calculatorsquare" value="square" name="operation">
<label for="calculatordouble">Square</label>
<br>
<br>
<button style="border: 5px dashed black;height: 30px;width: 60px;"
onclick="calculate()">Submit</button>
<br>
<br>
<p>Output</p>
<textarea id="output" placeholder="Your output will be shown here"></textarea>
<script src="./script.js"></script>
</body>
</html>
```

OUTPUT:





14. Create a form that takes data from a customer. The form must be well designed and should accept the customer's FirstName, LastName, Age, Birthday and FoodPreferences. At the submission of this form, create a Customer object in JavaScript using the above values and an equivalent JSON object. Print both these objects on the console. Using AJAX, displays the data of two customers in a presentable way.

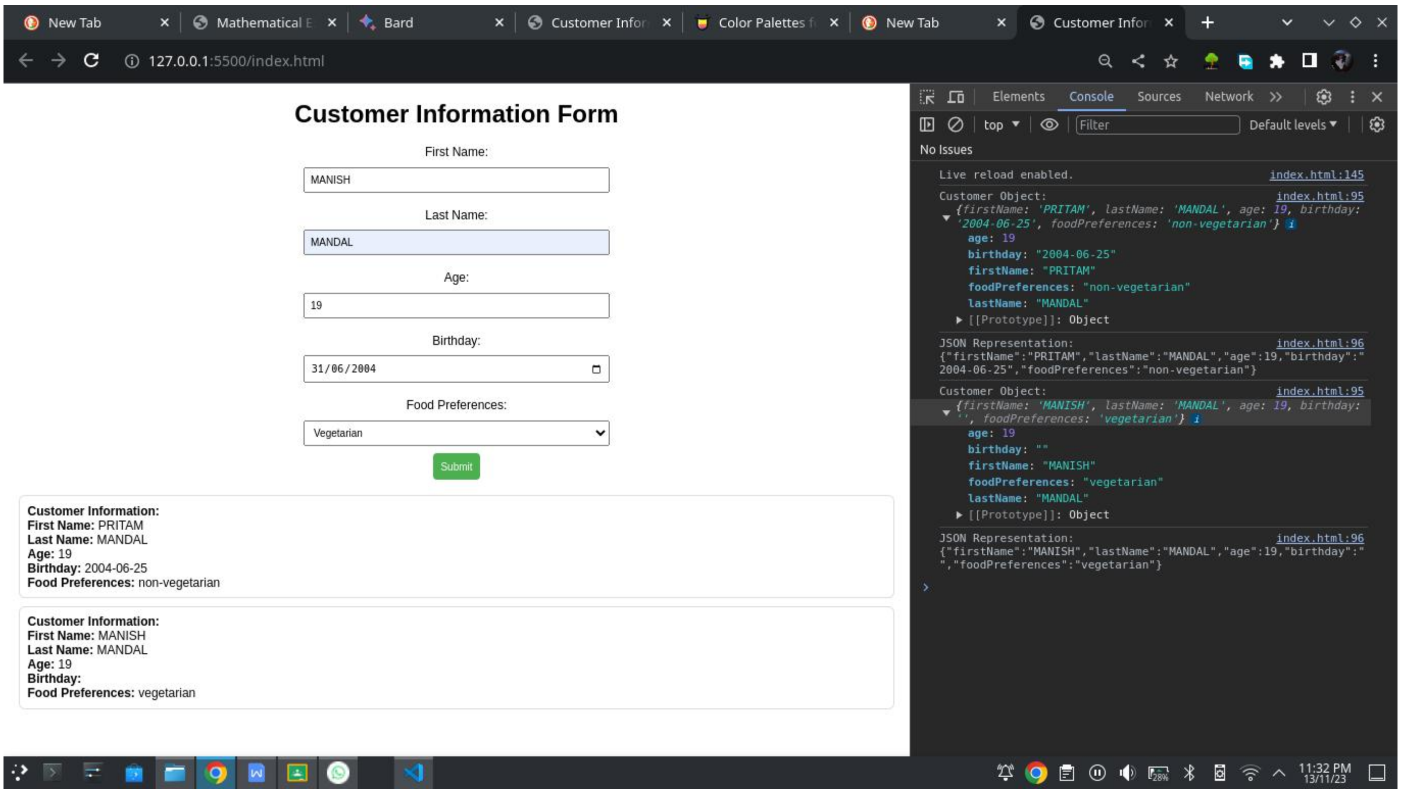
CODE:

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Customer Information Form</title>
<style>
body {
font-family: Arial, sans-serif;
text-align: center;
margin: 20px;
}

form {
max-width: 400px;
margin: 0 auto;
}

label {
display: block;
margin: 10px 0;
}

input, select {
width: 100%;
padding: 8px;
margin-bottom: 10px;
box-sizing: border-box;
}
```

```
button {
padding: 10px;
background-color: #4CAF50;
color: white;
border: none;
border-radius: 5px;
cursor: pointer;
}

#customerData {
margin-top: 20px;
}

#customerData div {
border: 1px solid #ddd;
border-radius: 8px;
padding: 10px;
margin: 10px 0;
text-align: left;
}
</style>
</head>
<body>
<h1>Customer Information Form</h1>
<form id="customerForm">
<label for="firstName">First Name:</label>
<input type="text" id="firstName" name="firstName" required>

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

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

<label for="birthday">Birthday:</label>
<input type="date" id="birthday" name="birthday" required>

<label for="foodPreferences">Food Preferences:</label>
<select id="foodPreferences" name="foodPreferences" required>
<option value="vegetarian">Vegetarian</option>
<option value="non-vegetarian">Non-Vegetarian</option>
</select>

<button type="button" onclick="submitForm()">Submit</button>
</form>

<div id="customerData"></div>

<script>
function submitForm() {
const firstName = document.getElementById('firstName').value;
const lastName = document.getElementById('lastName').value;
const age = parseInt(document.getElementById('age').value);
const birthday = document.getElementById('birthday').value;
const foodPreferences = document.getElementById('foodPreferences').value;

const customerObject = {
firstName: firstName,
lastName: lastName,
age: age,
birthday: birthday,
foodPreferences: foodPreferences
};

console.log("Customer Object:", customerObject);
console.log("JSON Representation:", JSON.stringify(customerObject));

displayCustomerData(customerObject);
}

function displayCustomerData(customer) {
const customerDataDiv = document.getElementById('customerData');
const customerDiv = document.createElement('div');

customerDiv.innerHTML = `
<strong>Customer Information:</strong><br>
<strong>First Name:</strong> ${customer.firstName}<br>
<strong>Last Name:</strong> ${customer.lastName}<br>
<strong>Age:</strong> ${customer.age}<br>
<strong>Birthday:</strong> ${customer.birthday}<br>
<strong>Food Preferences:</strong> ${customer.foodPreferences}
`;

customerDataDiv.appendChild(customerDiv);
}
</script>
</body>
</html>
```

OUTPUT:



Customer Information Form

First Name: MANISH

Last Name: MANDAL

Age: 19

Birthday: 31/06/2004

Food Preferences: Vegetarian

Submit

Customer Information:
First Name: PRITAM
Last Name: MANDAL
Age: 19
Birthday: 2004-06-25
Food Preferences: non-vegetarian

Customer Information:
First Name: MANISH
Last Name: MANDAL
Age: 19
Birthday:
Food Preferences: vegetarian

```
Live reload enabled.                                            index.html:145
Customer Object:                                                index.html:95
▼ {firstName: 'PRITAM', lastName: 'MANDAL', age: 19, birthday:
  '2004-06-25', foodPreferences: 'non-vegetarian'} i
    age: 19
    birthday: "2004-06-25"
    firstName: "PRITAM"
    foodPreferences: "non-vegetarian"
    lastName: "MANDAL"
  ▶ [[Prototype]]: Object
JSON Representation:                                            index.html:96
{"firstName":"PRITAM","lastName":"MANDAL","age":19,"birthday":"
2004-06-25","foodPreferences":"non-vegetarian"}
Customer Object:                                                index.html:95
▼ {firstName: 'MANISH', lastName: 'MANDAL', age: 19, birthday:
  '', foodPreferences: 'vegetarian'} i
    age: 19
    birthday: ""
    firstName: "MANISH"
    foodPreferences: "vegetarian"
    lastName: "MANDAL"
  ▶ [[Prototype]]: Object
JSON Representation:                                            index.html:96
{"firstName":"MANISH","lastName":"MANDAL","age":19,"birthday":"
","foodPreferences":"vegetarian"}
>
```