**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**

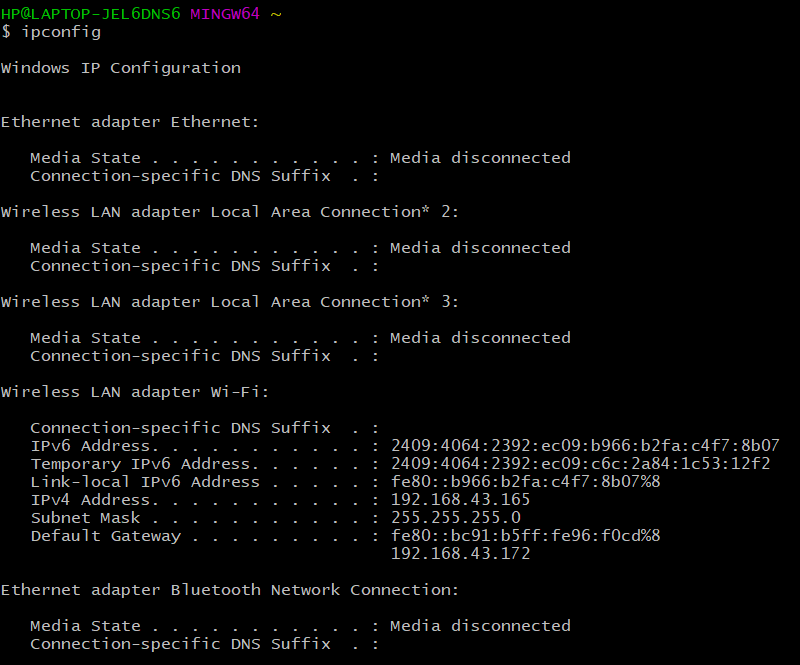
**should search for credentials in mysql database using server created in previous practical. On**

**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**

**Q1.** 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.



**● System IP Address: 192.168.43.165**

**● Subnet mask: 255.255.255.0**

**● Network ID: 192.168.43.0**

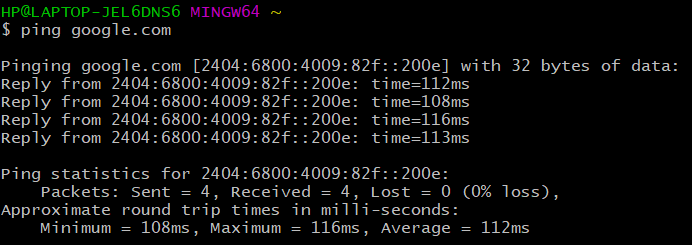
**● Broadcast ID: 192.168.43.255**

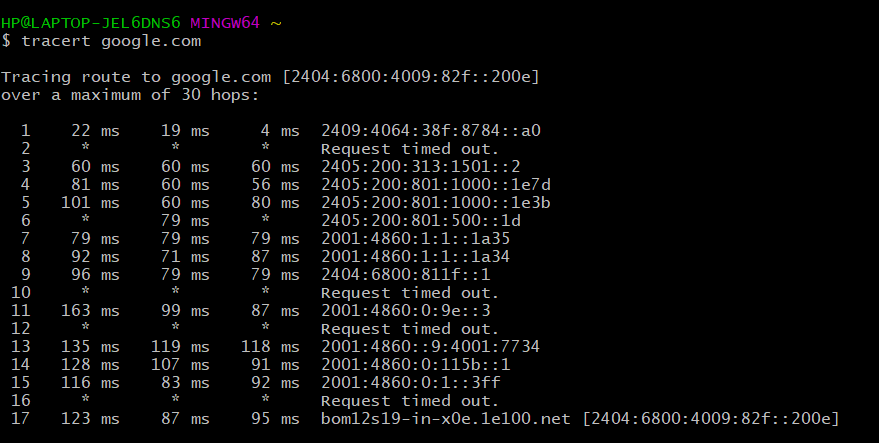
**● Maximum number of system possible on the network: 254**

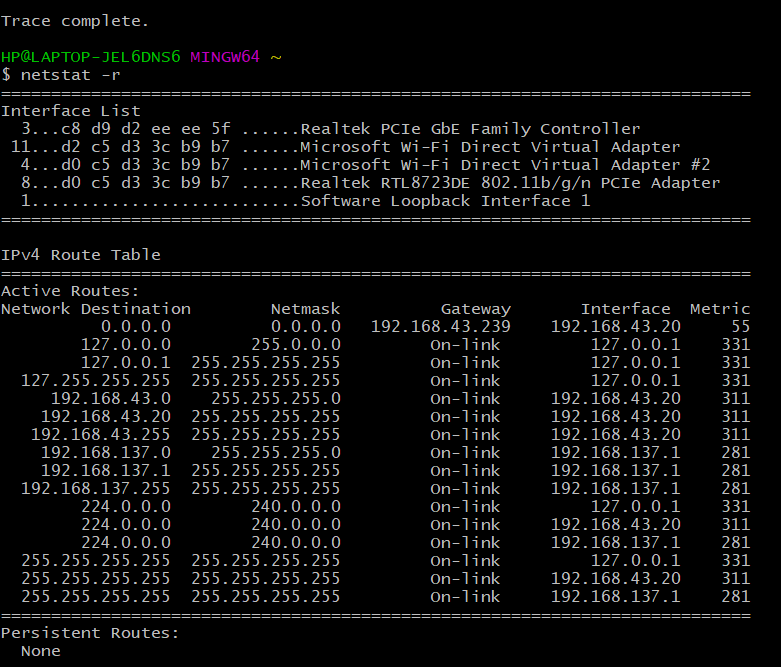
**● Range of IP addresses: 192.168.43.0 to 192.168.43.255**

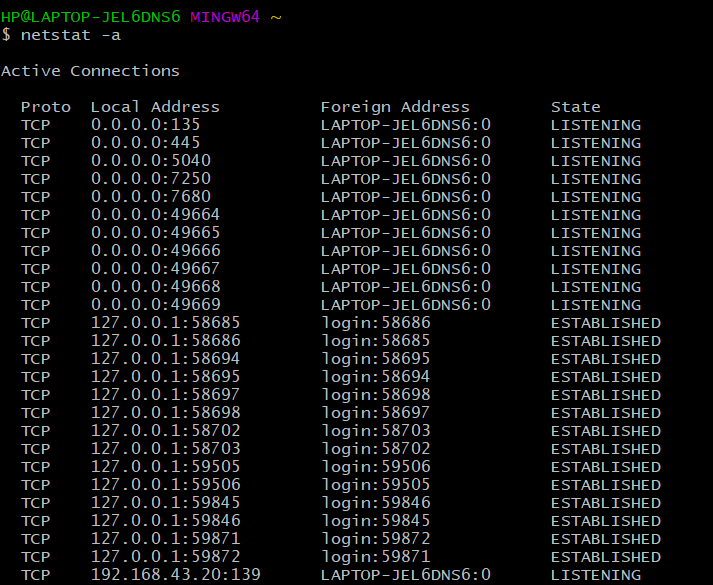
**● Class type: C**

**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.**









## Q3. 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>Question 3</title>**

**<script type="text/javascript" src="validate.js"></script>**

**<style>**

**\*{**

**margin: 5px;**

**margin-left: 20px;**

**padding: 2px;**

**}**

**.reset{**

**background:gray;**

**font-size: 24px;**

**color:black;**

**border-radius: 20px;**

**}**

**.sub{**

**background: rgb(0, 128, 75);**

**font-size: 24px;**

**color: black;**

**border-radius: 20px;**

**}**

**td{**

**font-size: 25px;**

**padding: 23px;**

**}**

**body{**

**background: rgba(0, 0, 0, 0.7) url(progra.jpg);**

**background-blend-mode: darken;**

**}**

**.form{**

**color:violet;**

**align-items: center;**

**align-content: center;**

**background: rgba(0, 0, 0, 0.9) url(progra.jpg);**

**background-blend-mode: darken;**

**padding: 32px;**

**<!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>Question 3</title>**

**<script type="text/javascript" src="validate.js"></script>**

**<style>**

**\*{**

**margin: 5px;**

**margin-left: 20px;**

**padding: 2px;**

**}**

**.reset{**

**background:gray;**

**font-size: 24px;**

**color:black;**

**border-radius: 20px;**

**}**

**.sub{**

**background: rgb(0, 128, 75);**

**font-size: 24px;**

**color: black;**

**border-radius: 20px;**

**}**

**td{**

**font-size: 25px;**

**padding: 23px;**

**}**

**body{**

**background: rgba(0, 0, 0, 0.7) url(progra.jpg);**

**background-blend-mode: darken;**

**}**

**.form{**

**color:violet;**

**align-items: center;**

**align-content: center;**

**background: rgba(0, 0, 0, 0.9) url(progra.jpg);**

**background-blend-mode: darken;**

**padding: 32px;**

**text-align: center;**

**}**

**</style>**

**</head>**

**<body>**

**<form action="#"name="Information Page"class="form" >**

**<table cellpadding="2" width="80%" align="center" cellspacing="2">**

**<tr>**

**<td colspan="2">**

**<center><font size="14">**

**<b>Information Page</b>**

**</font></center>**

**</td>**

**</tr>**

**<tr>**

**<td >**

**Name:**

**</td>**

**<td><label for="Name">Himanshu</label></td>**

**</tr>**

**<tr>**

**<td >**

**Course:**

**</td>**

**<td><label for="Course">B.sc(H)Computer SCience</label></td>**

**</tr>**

**<tr>**

**<td >**

**Hobbie:**

**</td>**

**<td><label for="Hobbie">Reading</label></td>**

**</tr>**

**<tr>**

**<td >**

**Address:**

**</td>**

**<td> <Address>Gandhi Nagar Delhi</Address> </td>**

**</tr>**

**<tr>**

**<td >**

**Plans :**

**</td>**

**<td>**

**M.C.A**

**</td>**

**</tr>**

**</table>**

**</form>**

**</body>**

**</html>**

**Q4.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>Document</title>**

**</head>**

**<style>**

**\*{**

**margin: 0;**

**padding: 0;**

**}**

**.table{**

**/\* margin: 4px; \*/**

**/\* background-color: indianred; \*/**

**width: 100%;**

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

**}**

**body{**

**background-color: rgb(56, 3, 34);**

**}**

**</style>**

**<body>**

**<p class="table" id="table1"> </p>**

**<p class="table" id="table2"> </p>**

**<p class="table" id="table3"> </p>**

**<p class="table" id="table4"> </p>**

**<p class="table"id="table5"> </p>**

**<p class="table"id="table6"> </p>**

**<p class="table" id="table7"> </p>**

**<p class="table"id="table8"> </p>**

**<p class="table" id="table9"> </p>**

**</body>**

**<script>**

**// tb=document.querySelector('.table')**

**let num=2;**

**let j=1;**

**let i=0;**

**let g=14;**

**let v=Math.floor(Math.random()\*7)**

**function table(num,ind){**

**const collection=document.getElementsByClassName("table");**

**var co=["red","blue","green","cyan","pink","yellow","orange"];**

**v=Math.floor(Math.random()\*7)**

**collection[ind].style.color=co[v];**

**let u=g+"px"**

**collection[i].style.fontSize=u;**

**for(let j=1;j<11;j++){**

**let s=num+"x"+j+"="+num\*j+"&nbsp"**

**collection[ind].innerHTML+=s;**

**}**

**collection[ind].innerHTML+=('<br>');**

**}**

**function print (){**

**table(num,i);**

**num++;**

**i++;**

**g=g+2;**

**// if(num>10){**

**// clearInterval(printer);**

**// }**

**// table(num,i);**

**// num++;**

**// i++;**

**// g=g+4;**

**// if(j==11){**

**// g=g+4;**

**// j=1;**

**// num++;**

**// i++;**

**// // g=(Math.floor(Math.random()\*20+15));**

**// // g=g+(Math.floor(Math.random()\*50+15));**

**// }**

**// if(i==0 && num==2 &&j==1){**

**// // g=(Math.floor(Math.random()\*20+15));**

**// // g=g+(Math.floor(Math.random()\*50+15));**

**// v=Math.floor(Math.random()\*7)**

**// }**

**// // let g=(Math.floor(Math.random()\*20+15));**

**// // g=g+(Math.floor(Math.random()\*50+15));**

**// // let v=Math.floor(Math.random()\*5)**

**// j++;**

**}**

**const printer=setInterval(print,2000);**

**print()**

**</script>**

**</html>**

**Q5. 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>Document</title>**

**</head>**

**<style>**

**.grid-container {**

**display: grid;**

**grid-template-columns: auto auto auto;**

**background-color: rgb(8, 35, 61);**

**margin-top: 100px;**

**padding: 10px;**

**}**

**.sub {**

**font-size: 78px;**

**background-color: aqua;**

**border: 1px solid rgba(0, 0, 0, .7);**

**}**

**</style>**

**<body>**

**<div class="paragraph" id="para">Computer programming or coding is the composition of sequences of instructions,**

**called programs, that computers can follow to perform tasks.[1][2] It involves designing and implementing**

**algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages.**

**Programmers typically use high-level programming languages that are more easily intelligible to humans than**

**machine code, which is directly executed by the central processing unit. Proficient programming usually requires**

**expertise in several different subjects, including knowledge of the application domain, details of programming**

**languages and generic code libraries, specialized algorithms, and formal logic.**

**Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging**

**(investigating and fixing problems), implementation of build systems, and management of derived artifacts, such**

**as programs' machine code. While these are sometimes considered programming, often the term software development**

**is used for this larger overall process – with the terms programming, implementation, and coding reserved for**

**the writing and editing of code per se. Sometimes software development is known as software engineering,**

**especially when it employs formal methods or follows an engineering design process.</div>**

**<div class="grid-container">**

**<input type="button" value="RED" class="sub" onclick="color('red')">**

**<input type="button" value="Green" class="sub" onclick="color('green')">**

**<input type="button" value="Blue" class="sub" onclick="color('Blue')">**

**<input type="button" value="10" class="sub" onclick="colors(10)">**

**<input type="button" value="20" class="sub" onclick="colors(20)">**

**<input type="button" value="30" class="sub" onclick="colors(30)">**

**<input type="button" value="Arial" class="sub" onclick="colorf('arial')">**

**<input type="button" value="Impact" class="sub" onclick="colorf('Impact')">**

**<input type="button" value="Monospace" class="sub" onclick="colorf('Monospace')">**

**</div>**

**</body>**

**<script>**

**function color(i) {**

**var t = document.getElementById("para");**

**t.style.color = i;**

**}**

**function colors(i) {**

**var t = document.getElementById("para");**

**var p = i + "px"**

**t.style.fontSize = p;**

**}**

**function colorf(i) {**

**var t = document.getElementById("para");**

**var p =**

**t.style.fontFamily = i;**

**}**

**</script>**

**</html>**

**Q6. 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.**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="utf-8">**

**<title>Practical 6</title>**

**<style type="text/css">**

**.container {**

**width: 80%;**

**margin: auto;**

**border: 1px solid black;**

**border-radius: 8px;**

**padding: 50px;**

**background-color: brown;**

**}**

**.btn-submit {**

**border-radius: 5px;**

**color: white;**

**background: rgb(78, 47, 255);**

**font-weight: bold;**

**text-align: center;**

**font-size: 1rem;**

**margin: 20px;**

**position: relative;**

**left: 380px;**

**}**

**hr{**

**height: 6px;**

**background-color: rgb(255, 94, 0);**

**}**

**.label1{**

**text-align: right;**

**margin-top: 15px;**

**margin-left: 250px;**

**}**

**input{**

**text-align: right;**

**margin-left: 100px;**

**}**

**.age{**

**margin-left: 170px;**

**}**

**.name{**

**margin-left: 130px;**

**}**

**.weight{**

**margin-left: 150px;**

**}**

**.pet{**

**margin-left: 145px;**

**}**

**.like{**

**margin-left: 160px;**

**}**

**h1{**

**text-align: center;**

**}**

**@media(width<=575) {**

**.container {**

**width: 90%;**

**}**

**}**

**</style>**

**</head>**

**<body>**

**<div class="container">**

**<h1>Pet's Information</h1>**

**<hr> <label for="name" class="label1">Pet's Name:</label>**

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

**<label for="age" class="label1">Age:</label>**

**<input type="number" name="age" class="age"> <br> <br>**

**<label for="weight" class="label1">Weight:</label> <input type="number" name="weight"**

**class="weight"><br><br>**

**<label for="type" class="label1">Pet type:</label> <input type="text" name="type" class="pet"><br><br>**

**<label for="likes" class="label1">Likes:</label> <input type="text" name="likes" class="like"><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>**

**Q7. 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: rgb(70, 147, 167);**

**color: rgb(14, 9, 9);**

**border-radius: 8px;**

**}**

**#content{**

**font-family: Impact, Haettenschweiler, 'Arial Narrow Bold', sans-serif;**

**font-size: 26px;**

**background: yellow;**

**}**

**</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;**

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

**// c.innerHTML=htmlpart;**

**content.append(p);**

**htmlpart = ""; }**

**} </script>**

**</body>**

**</html>**

## Q8. 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 lang="en">**

**<head>**

**<meta charset="UTF-8">**

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

**<title>Practial 8</title>**

**<script src="jquery-3.7.1.js"></script>**

**<script>**

**$(document).ready(function() {**

**$("#button").click(function() {**

**$("h1").css({**

**"color": "red",**

**"text-align": "center",**

**"background": "cyan",**

**"border-radius": "20%"**

**})**

**$("body").css({**

**"background": "purple"**

**})**

**$("td").css({**

**"color": "red",**

**"font-size": "20px",**

**"background-color": "yellow"**

**})**

**$("h2").css({**

**"color": "Blue",**

**"text-align": "center"**

**})**

**$("hr").css({**

**"color": "blue",**

**"height": "3px",**

**"background": "blue"**

**})**

**$(".list1").css("text-align", "center")**

**$("li").css({**

**"list-style": "none"**

**})**

**});**

**});**

**</script>**

**</head>**

**<style>**

**\* {**

**margin: 0;**

**padding: 0;**

**}**

**table,**

**th,**

**td {**

**border: 1px solid black;**

**border-collapse: collapse;**

**color: blue;**

**height: 40%;**

**}**

**table {**

**margin-left: auto;**

**margin-right: auto;**

**margin-top: 5px;**

**margin-bottom: 10px;**

**}**

**.btn {**

**text-align: center;**

**}**

**.btn {**

**font-size: 34px;**

**}**

**</style>**

**<body>**

**<H1>B.SC(H) Computer Science</H1>**

**<h2>course detail</h2>**

**<ul class="list1">**

**<li>Annual Fee:35000</li>**

**<li>Eligibility: <b>10+2 at least 60% with Mathematics</b></li>**

**</ul>**

**<hr>**

**<h1>Course subject detail</h1>**

**<hr>**

**<table style="width:70%">**

**<tr>**

**<th style="width:10%">Semester</th>**

**<th>Paper</th>**

**<th>credits</th>**

**</tr>**

**<tr>**

**<td rowspan="4">1</td>**

**<td>c++</td>**

**<td>6</td>**

**</tr>**

**<tr>**

**<td>CSA</td>**

**<td>6</td>**

**</tr>**

**<tr>**

**<td>AECC</td>**

**<td>4</td>**

**</tr>**

**<tr>**

**<td>GE</td>**

**<td>4</td>**

**</tr>**

**<tr>**

**<td rowspan="4">2</td>**

**<td>java</td>**

**<td>6</td>**

**</tr>**

**<tr>**

**<td>Discrete Mathematics</td>**

**<td>6</td>**

**</tr>**

**<tr>**

**<td>AECC-2</td>**

**<td>4</td>**

**</tr>**

**<tr>**

**<td>GE-2</td>**

**<td>4</td>**

**</tr>**

**<tr>**

**<td rowspan="4">3</td>**

**<td>Data Structure</td>**

**<td>6</td>**

**</tr>**

**<tr>**

**<td>Operating System</td>**

**<td>6</td>**

**</tr>**

**<tr>**

**<td>Computer Networking</td>**

**<td>6</td>**

**</tr>**

**<tr>**

**<td>GE-3</td>**

**<td>4</td>**

**</tr>**

**<tr>**

**<td rowspan="4">4</td>**

**<td>Design and Analysis Algorithms</td>**

**<td>6</td>**

**</tr>**

**<tr>**

**<td>Software Engineering</td>**

**<td>6</td>**

**</tr>**

**<tr>**

**<td>Database Management System </td>**

**<td>6</td>**

**</tr>**

**<tr>**

**<td>GE-4</td>**

**<td>4</td>**

**</tr>**

**<tr>**

**<td rowspan="4">5</td>**

**<td>Internet Technology</td>**

**<td>6</td>**

**</tr>**

**<tr>**

**<td>Data Visualization</td>**

**<td>6</td>**

**</tr>**

**<tr>**

**<td>Theory of Computation</td>**

**<td>4</td>**

**</tr>**

**<tr>**

**<td>Digital Image Processing</td>**

**<td>6</td>**

**</tr>**

**<tr>**

**<td rowspan="4">6</td>**

**<td>Artificial Intelligence</td>**

**<td>6</td>**

**</tr>**

**<tr>**

**<td>Computer Graphics</td>**

**<td>6</td>**

**</tr>**

**<tr>**

**<td>Advance Algorithms</td>**

**<td>6</td>**

**</tr>**

**<tr>**

**<td>Project Work</td>**

**<td>4</td>**

**</tr>**

**</table>**

**<hr>**

**<div class="btn">**

**<button class="btn" id="button">Redesign</button>**

**</div>**

**</body>**

**</html>**

**<Q9. 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 lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width,

initial-scale=1, shrink-to-fit=no">

<link rel="stylesheet"

href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css

/bootstrap.min.css">

<title>Bootstrap Image Gallery</title>

<style>

body {

padding: 20px;

}

.gallery-item {

margin-bottom: 20px;

}

</style>

</head>

<body>

<div class="container-fluid">

<h2 class="text-center">Bootstrap Image Gallery</h2>

<div class="row">

<div class="col-md-4 gallery-item">

<img src="pd.jpg" class="img-fluid"

alt="Gallery Item 1">

</div>

<div class="col-md-4 gallery-item">

<img src="x.jpg" class="img-fluid" alt="Gallery

Item 2">

</div>

<div class="col-md-4 gallery-item">

<img src="pd.jpg" class="img-fluid" alt="Gallery

Item 3">

</div>

</div>

</div>

<script src="https://code.jquery.com/jquery3.3.1.slim.min.js"></script>

<script

src="https://stackpat.bootstrapcdn.com/bootstrap/4.3.1/js/b

ootstrap.min.js"></script>

</body>

</html>

**Q10. 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');**

**//create a server object:**

**http.createServer(function (req, res) {**

**res.write('hello ,this is my NOde.js server!'); //write a response to the client**

**res.end(); //end the response**

**}).listen(10000);**

**Q11)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.**

==> index.html file :-

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Sign In / Sign Up</title>

</head>

<body>

<h2>Sign In</h2>

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

<label for="signin-username">Username:</label>

<input type="text" id="signin-username" name="username" required>

<br>

<label for="signin-password">Password:</label>

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

<br>

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

</form>

<h2>Sign Up</h2>

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

<label for="signup-username">Username:</label>

<input type="text" id="signup-username" name="username" required>

<br>

<label for="signup-password">Password:</label>

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

<br>

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

</form>

</body>

</html>

==> JS file :-

const http = require('http');

const fs = require('fs');

const url = require('url');

const mysql = require('mysql');

const path = require('path');

// Create a MySQL connection

const db = mysql.createConnection({

host: 'localhost',

user: 'root',

password: 'K1@2r3t4i5k6e7y8',

database: 'kartikey\_18',

});

// Connect to the database

db.connect((err) => {

if (err) {

console.error('Database connection error:', err);

} else {

console.log('Connected to the database');

}

});

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

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

const pathname = parsedUrl.pathname;

const filePath = path.join(\_\_dirname, pathname);

if (req.method === 'GET') {

if (pathname === '/') {

// Serve index.html for the root path

fs.readFile(path.join(\_\_dirname, 'index11.html'), (err, data) => {

if (err) {

res.writeHead(500, { 'Content-Type': 'text/plain' });

res.end('Internal Server Error');

} else {

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

res.end(data);

}

});} else if (pathname === '/welcome') {

    // Serve a welcome page for successful signin or signup

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

    res.end('<h1>Welcome!</h1>');

    } else {

    // Serve static files

    fs.readFile(filePath, (err, data) => {

    if (err) {

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

    res.end('Not Found');

    } else {

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

    res.end(data);

    }

    });

    }

    } else if (req.method === 'POST') {

    if (pathname === '/signin') {

    // Handle Sign In form submission

    let body = '';

    req.on('data', (chunk) => {

    body += chunk;

    });

    req.on('end', () => {

    const formData = new URLSearchParams(body);

    const username = formData.get('username');

    const password = formData.get('password');

    // Query the database to check credentials

    db.query(

    'SELECT \* FROM users WHERE username = ? AND password = ?',

    [username, password],

    (err, results) => {

    if (err) {

    console.error('Database query error:', err);

    res.writeHead(500, { 'Content-Type': 'text/plain' });

    res.end('Internal Server Error');

    } else if (results.length > 0) {

    // Successful sign-in

    res.writeHead(302, { Location: '/welcome' });

    res.end();

    } else {

    // Invalid credentials

    res.writeHead(401, { 'Content-Type': 'text/plain' });

    res.end('Unauthorized');

    }

    }

    );

    });

    } else if (pathname === '/signup') {

    // Handle Sign Up form submission

    let body = '';

    req.on('data', (chunk) => {

    body += chunk;

    });

    req.on('end', () => {

    const formData = new URLSearchParams(body);

    const username = formData.get('username');

    const password = formData.get('password');

    // Insert new entry into the database

    db.query('INSERT INTO users (username, password) VALUES (?, ?)', [username, password], (err) => {

    if (err) {

    console.error('Database insertion error:', err);

    res.writeHead(500, { 'Content-Type': 'text/plain'

    }); res.end('Internal Server Error');

    } else {

    // Successful signup

    res.writeHead(302, { Location: '/' });

    res.end();

    }

    });

    });

    }

    }

    });

    const PORT = 10000;

    server.listen(PORT, () => {

    console.log(`Server is running at http://localhost:${PORT}/`);

    });