|  |
| --- |
| Close-up image showing the leaf-sides of two oversized books side-by-side on a bookshelf, with additional books in soft focus background |
| IT Practical File  B.Sc. (Hons.) Computer Science |
| |  |  |  | | --- | --- | --- | | Abhineet Raman |  | 2002078 | |

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.

Text

Description automatically generated

As the IPv4 address starts with **192**, that means it is a Class C address. Therefore, the network address would be **192.168.1.0**. The maximum number of possible systems to use on the network would be **254**.The range of Ip addresses would be **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.

A screenshot of a computer

Description automatically generated with medium confidence

A computer screen capture

Description automatically generated with medium confidence

Text

Description automatically generated

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.

<html>

<head>

<title>Multiplication Table</title>

</head>

<body>

<p id="2" style = "font-size:20px; color:black;"></p>

<p id="3" style = "font-size:25px; color:violet;"></p>

<p id="4" style = "font-size:30px; color:indigo;"></p>

<p id="5" style = "font-size:35px; color:blue;"></p>

<p id="6" style = "font-size:40px; color:green;"></p>

<p id="7" style = "font-size:45px; color:yellow;"></p>

<p id="8" style = "font-size:50px; color:orange;"></p>

<p id="9" style = "font-size:55px; color:red;"></p>

<p id="10" style = "font-size:60px; color:black;"></p>

<script>

let table = 2;

var intv = setInterval(begin,2000);

function begin(){

var result ='';

var tmp = 1;

for(let i = 1; i<=10; i++){

tmp = table\*i;

result = result.concat(tmp.toString());

result = result.concat(' ');

}

if(table>9){

clearInterval(intv);

document.getElementById(table.toString()).innerHTML = result;

}else{

document.getElementById(table.toString()).innerHTML = result;

table++;

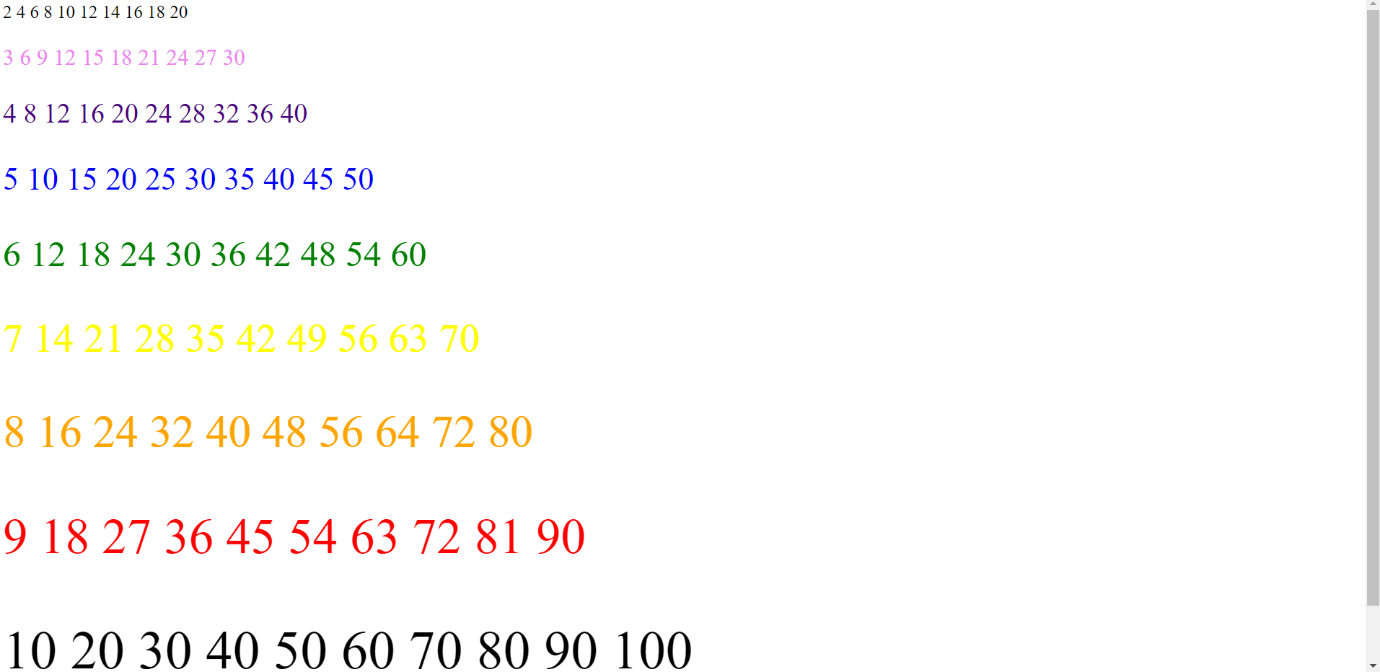
}

}

</script>

</body>

</html>



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.

<html>

<style>

#grad1 {

background-color: red;

background-image: linear-gradient(to right, red, orange, yellow, green, blue, indigo, violet);

}

</style>

<head>

<title>About me</title>

</head>

<div id="grad1">

<body>

<div style=" text-align:center; margin:auto; font-size: 25px; font-weight:bold">

<h1 style="color: darkblue;">My Origin</h1>

<p style="color:blueviolet;">I am from Kolkata, West Bengal</p>

<h1 style="color:darkblue;">My Course</h1>

<p style="color:blueviolet;">B.Sc(H) Computer Science</p>

<h1 style="color:darkblue;">My Hobbies</h1>

<p style="color:blueviolet;">Gaming and reading</p>

<h1 style="color:darkblue;">My Address</h1>

<p style="color:blueviolet;">Saket, New Delhi</p>

<h1 style="color:darkblue;">My Plans</h1>

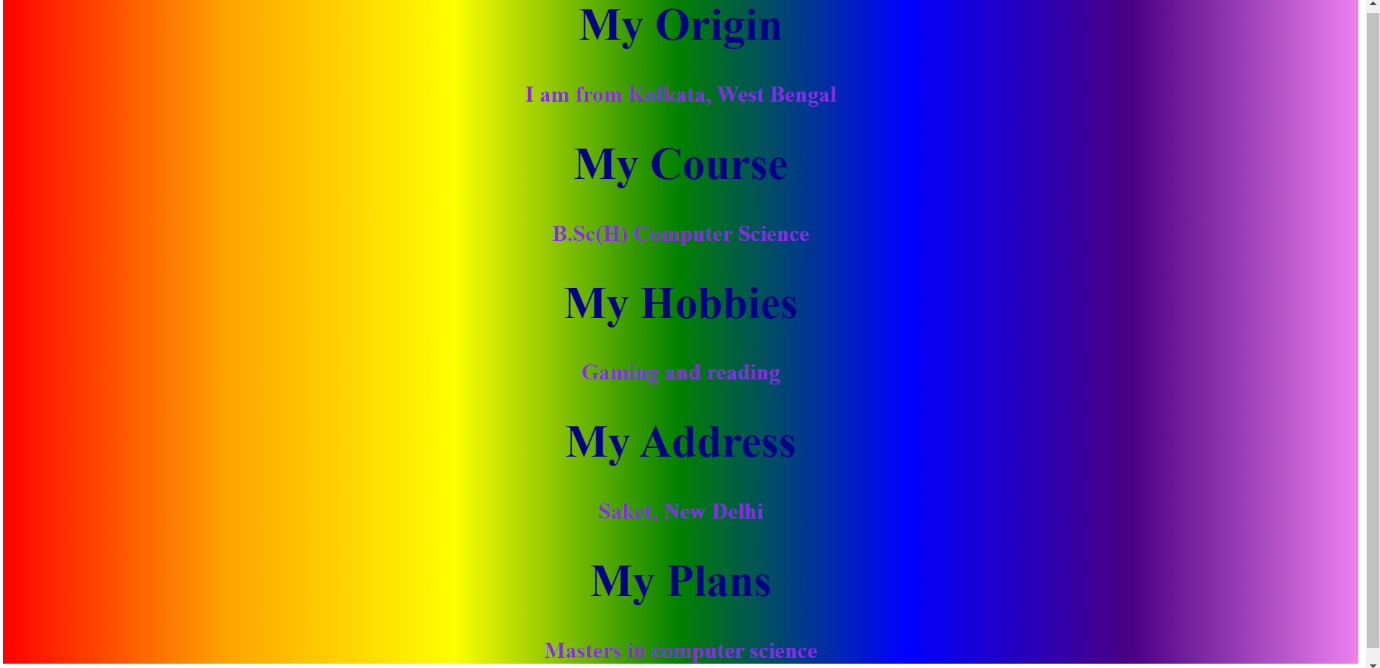
<p style="color:blueviolet;">Masters in computer science</p>

</div>

</body>

</div>

</html>



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.

<html>

<head>

<style>

.grid-container {

display: grid;

grid-template-columns: auto auto auto;

}

.grid-item {

border: 1px solid rgba(0, 0, 0, 0.8);

font-size: 30px;

text-align: center;

}

</style>

</head>

<body>

<div style='text-align: center; width: 60%;margin-left: 25%;'>

<p id='demo' style='font-size: 20px; font-family: Arial, Helvetica, sans-serif;'>

Contrary to popular belief, Lorem Ipsum is not simply random text.

It has roots in a piece of classical Latin literature from 45 BC,

making it over 2000 years old.

Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia,

looked up one of the more obscure Latin words, consectetur,

from a Lorem Ipsum passage, and going through the cites of the word in

classical literature, discovered the undoubtable source.

Lorem Ipsum comes from sections 1.10.32 and 1.10.33 of "de Finibus Bonorum

et Malorum" (The Extremes of Good and Evil) by Cicero, written in 45 BC.

This book is a treatise on the theory of ethics, very popular during the

Renaissance.

</p>

<div class="grid-container">

<div class="grid-item"><p><button onclick="change(1)" class='button buttnblue'>Blue</button>

</p></div>

<div class="grid-item"><p><button onclick="change(2)" class='button buttnviolet'>Violet</button>

</p></div>

<div class="grid-item"><p><button onclick="change(3)" class='button buttnorange'>Orange</button>

</p></div>

<div class="grid-item"> <p><button onclick="change(4)" class='button buttnsz2'>15</button>

</p></div>

<div class="grid-item"><p><button onclick="change(5)" class='button buttnasz1'>25</button>

</p></div>

<div class="grid-item"><p><button onclick="change(6)" class='button buttnsz3'>35</button>

</p></div>

<div class="grid-item"> <p><button onclick="change(7)" class='button buttnft1'>Garamond</button>

</p></div>

<div class="grid-item"><p><button onclick="change(8)" class='button buttnft2'>Courier New</button>

</p></div>

<div class="grid-item"> <p><button onclick="change(9)" class='button buttnft3'>Georgia</button>

</p>

</div>

</div>

<script>

function change(buttonId){

if(buttonId==1){

document.getElementById('demo').style.color = 'blue';

}

if(buttonId==2){

document.getElementById('demo').style.color = 'violet';

}

if(buttonId==3){

document.getElementById('demo').style.color = 'orange';

}

if(buttonId==4){

document.getElementById('demo').style.fontSize = '15px';

}

if(buttonId==5){

document.getElementById('demo').style.fontSize = '25px';

}

if(buttonId==6){

document.getElementById('demo').style.fontSize = '35px';

}

if(buttonId==7){

document.getElementById('demo').style.fontFamily = 'Garamond';

}

if(buttonId==8){

document.getElementById('demo').style.fontFamily = 'Courier New';

}

if(buttonId==9){

document.getElementById('demo').style.fontFamily = 'Georgia';

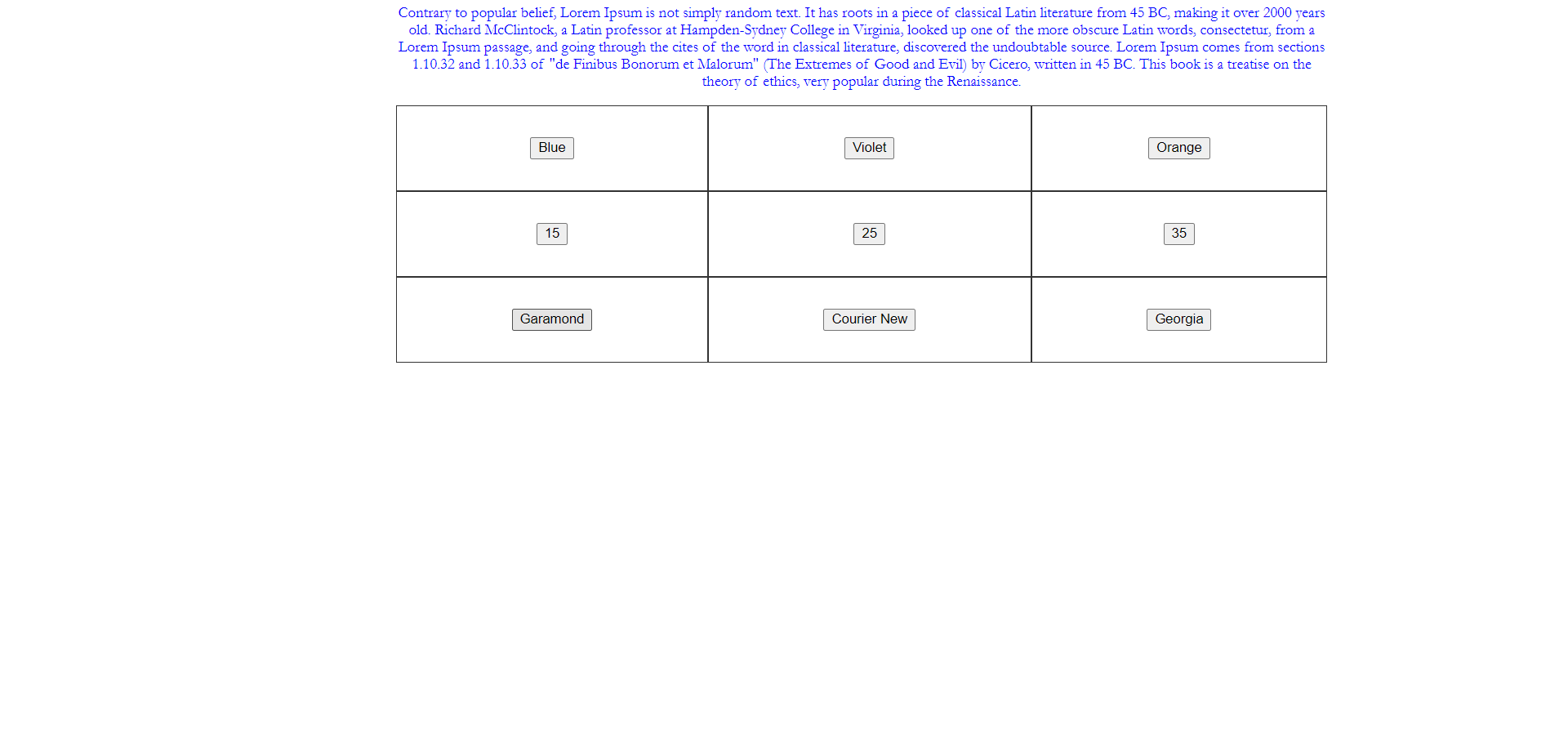
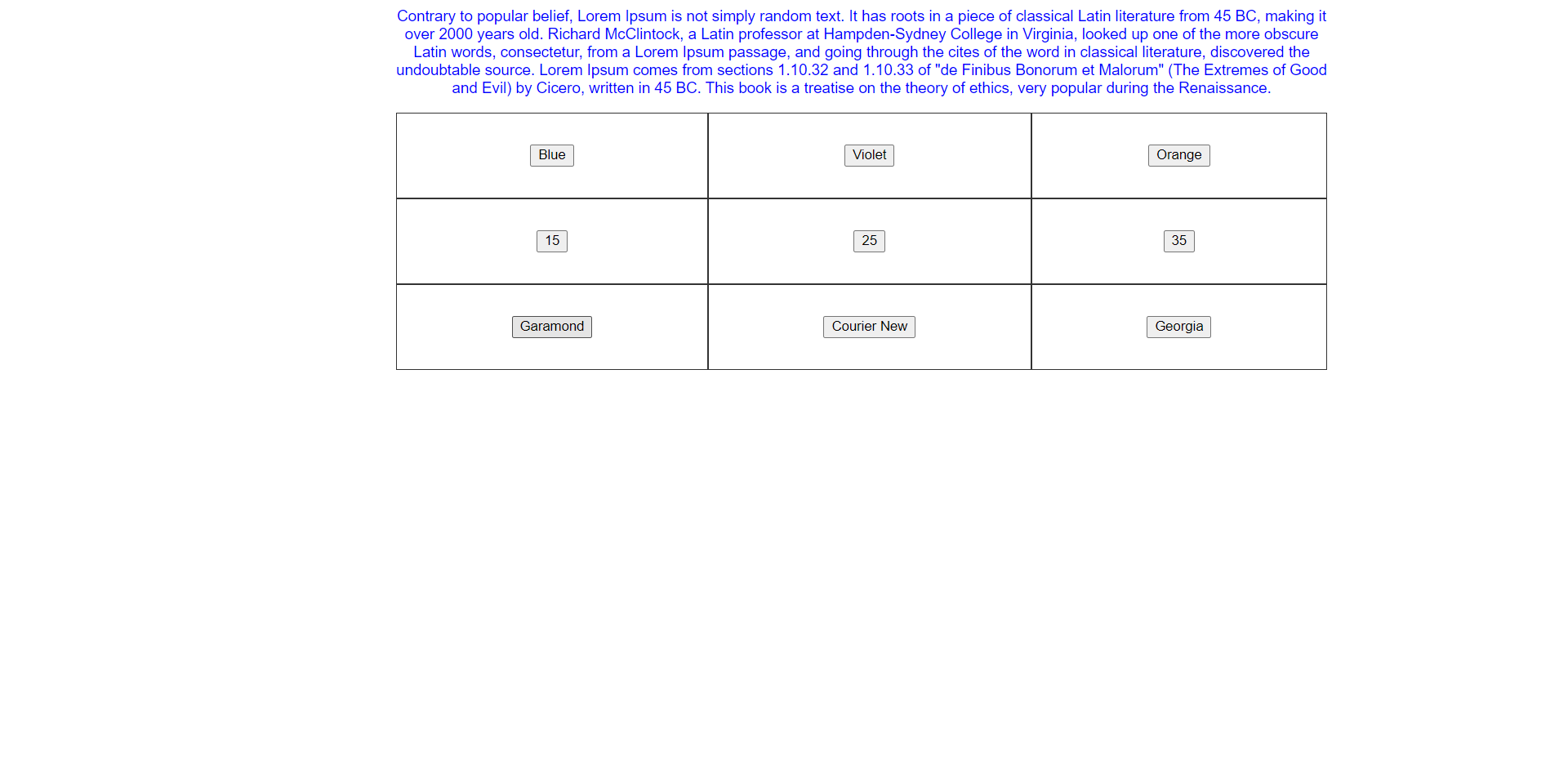
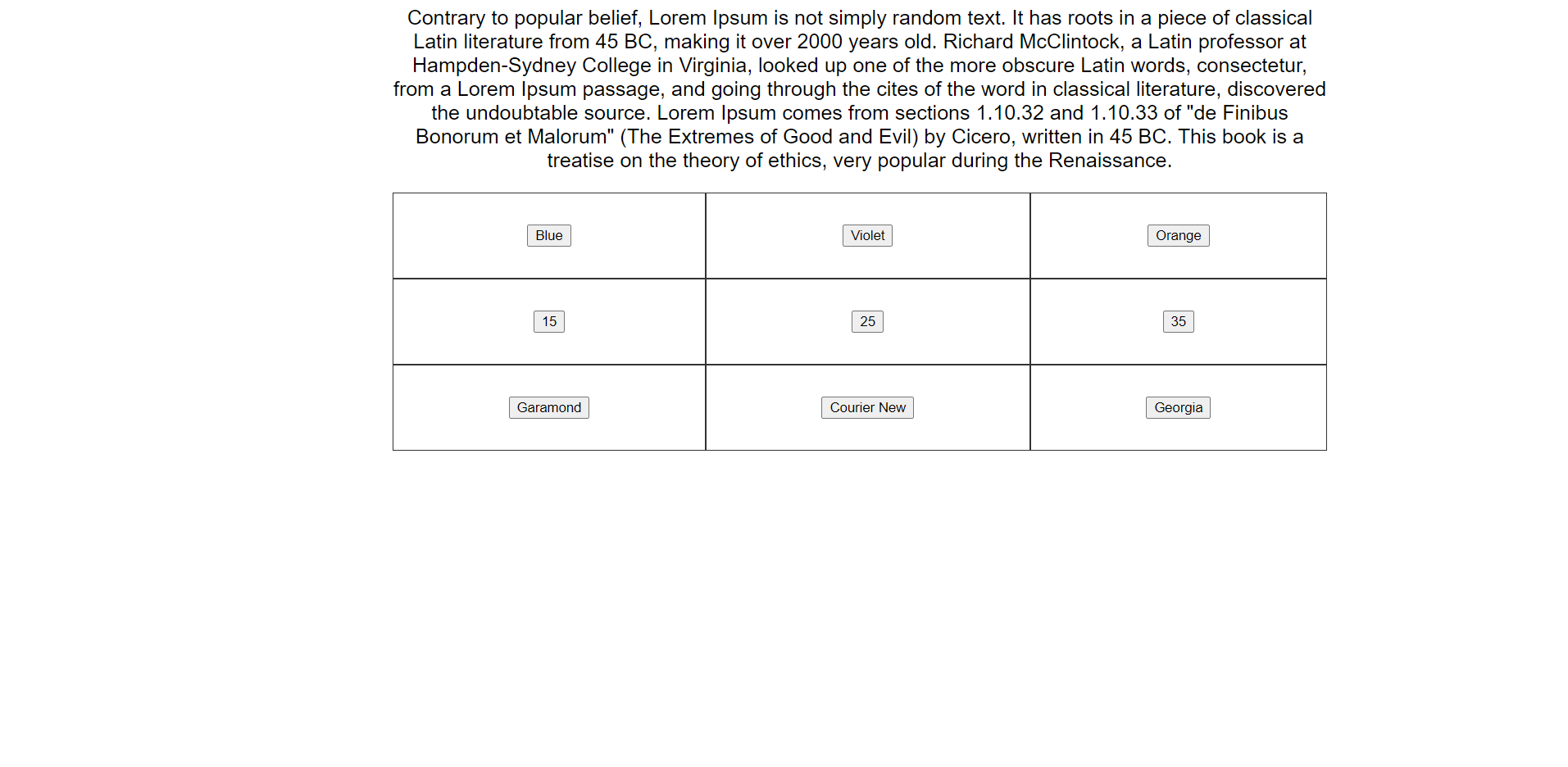
}

}

</script>

</body>

</html>



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.

<!DOCTYPE html>

<html>

<head>

<!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>

<body>

</body>

</html>

<style>

html {

background-color: #BFF4BF;

}

input[type=text],

select,

textarea,

input[type=number] {

width: 70%;

padding: 12px;

border: 1px solid whitesmoke;

border-radius: 4px;

box-sizing: border-box;

margin-top: 6px;

margin-left: 10%;

margin-bottom: 16px;

resize: vertical

}

button {

background-color: green;

color: white;

padding: auto;

border-radius: 4px;

cursor: pointer;

margin-left: 30%;

}

label {

font-size: xx-large;

margin-left: 20%;

padding: auto;

}

.container {

width: 30%;

margin-left: 25%;

padding: 20px;

}

</style>

</head>

<body>

<div class="container">

<form id="pet">

<label for="petName">Name : </label><br />

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

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

<input type="number" id="Age" /><br />

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

<input type="number" id="Weight" /><br />

<label for="type">Type : </label><br />

<input type="text" id="Type" /><br />

<label for="thing">Favourite Thing : </label><br />

<input type="text" id="Thing" /><br /><br />

<button type="button" onclick="submit()">Submit</button>

</form>

</div>

<script>

function submit(){

const pt = {

name: document.getElementById("Name").value,

age: document.getElementById("Age").value,

weight: document.getElementById("Weightt").value,

type: document.getElementById("Type").value,

favourite: document.getElementById("Thing").value,

};

console.log(pt);

const jSONPet = JSON.stringify(pt);

console.log(jSONPet);

document.getElementById("pet").reset;

}

</script>

</body>

</html>



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.

JSON File:

{

"jSONPet": [

{

"name": "Zophia",

"age": "5",

"weight": "11",

"type": "Bird",

"thing": "Spice"

},

{

"name": "Kiira",

"age": "5",

"weight": "22",

"type": "Dog",

"thing": "Chicken"

}

]

}

HTML File:

<!DOCTYPE html>

<html>

<head>

<!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>Pet</title>

<style>

html {

background-color: orangered;

}

table {

text-align: center;

width: 100%;

border: 1px solid;

}

th {

font-size: xx-large;

text-decoration: underline;

}

tbody tr:hover {

background-color: tomato;

font-size: x-large;

}

tr {

font-size: large;

}

.container {

width: 50%;

margin-left: 25%;

margin-top: 5%;

border-radius: 5px;

background-color: yellow;

padding: 20px;

}

</style>

</head>

<body>

<p style="text-align: center; font-size: 100px;">PET</p>

<div class="container">

<table><thead>

<th>Name</th>

<th>Age</th>

<th>Weight</th>

<th>Type</th>

<th>Favourite Thing</th>

</thead>

<tbody id="a">

</tbody>

</table>

</div>

<script>

setTimeout(loadDoc(), 0);

function loadDoc() {

var xhr = new XMLHttpRequest();

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

xhr.send();

xhr.onreadystatechange = function () {

if (this.readyState == 4 && this.status == 200) {

var x = JSON.parse(this.response);

var temp = "";

for (var i = 0; i < Object.keys(x.jSONPet).length; i++) {

temp = temp.concat("<tr><td>" + x.jSONPet[i].name + "</td><td>" +

x.jSONPet[i].age + "</td><td>" + x.jSONPet[i].weight + "</td><td>" +

x.jSONPet[i].type + "</td><td>" + x.jSONPet[i].thing +

"</td></tr>");

document.getElementById("a").innerHTML = temp;

}

}

};

}

</script>

</body>

</html>



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.

CSS File:

.htmlClass {

background: linear-gradient(to right, #d95d39, #946569);

}

.divClass {

text-align: center;

}

.tableClass {

margin-left: auto;

margin-right: auto;

}

.th {

font-size: 30px;

color: #474350;

font-style: Georgia;

}

.td {

font-size: 30px;

}

.heading {

font-size: 70px;

font-family: Cambria, serif;

text-decoration:blanchedalmond;

}

.fee {

font-size: 60px;

font-family: Times,serif;

}

.future{

font-size: 40px;

text-decoration:underline dotted;

}

.futureClass{

font-size: 40px;

text-decoration:underline dotted;

}

.element.criteria {

font-family: 'Courier New', Courier, monospace;

font-weight: 1000;

font-style: oblique;

font-size: x-large;

color: firebrick;

}

.element.eligibility {

font-family: 'Courier New', Courier, monospace;

font-weight: 1000;

font-style: oblique;

font-size: large;

}

HTML File:

<!DOCTYPE html>

<head>

<!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>Computer Science</title>

<link rel="stylesheet" href="design.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

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

</head>

<body>

<div id="main">

<h1 id="heading">B.Sc. Hons CS</h1>

<p id="fee">

Fee : ₹ 80,000

</p>

<p> <element id="criteria">Eligibility Criteria :</element><element id="eligibility"> A total of no less than 60% marks in English, Mathematics and best of the two

subjects

from Physics, Chemistry, Computer Science/Informatics Practices

OR

A total of no less than 60% marks in English, Mathematics and two subjects (other than Physics, Chemistry,

Computer Science/Informatics Practices) from List B with a deduction of 1% per subject in the aggregate.

</element>

</p>

<p>

<div id="tableContainer">

<table>

<thead>

<th>S. No.</th>

<th>Name</th>

<th>Credit</th>

</thead>

<tbody>

<tr>

<td>1.</td>

<td>

Computer System Architecture

</td>

<td>

4

</td>

</tr>

<tr>

<td>2.</td>

<td>

C++ Programming

</td>

<td>

4

</td>

</tr>

<tr>

<td>3.</td>

<td>

Programming in Java

</td>

<td>

4

</td>

</tr>

<tr>

<td>4.</td>

<td>

Discrete Structure

</td>

<td>

4

</td>

</tr>

<tr>

<td>5.</td>

<td>

Data Structures

</td>

<td>

5

</td>

</tr>

<tr>

<td>6.</td>

<td>

Operating System

</td>

<td>

5

</td>

</tr>

<tr>

<td>7.</td>

<td>

Computer Networks

</td>

<td>

5

</td>

</tr>

<tr>

<td>8.</td>

<td>

Design And Analysis of Algorithm

</td>

<td>

5

</td>

</tr>

<tr>

<td>9.</td>

<td>

Software Engineering

</td>

<td>

5

</td>

</tr>

<tr>

<td>10.</td>

<td>

Database Management Systems

</td>

<td>

5

</td>

</tr>

<tr>

<td>11.</td>

<td>

Internet Technologies

</td>

<td>

4

</td>

</tr>

<tr>

<td>12.</td>

<td>

Theory of Computation

</td>

<td>

5

</td>

</tr>

<tr>

<td>13.</td>

<td>

Artificial Intelligence

</td>

<td>

4

</td>

</tr>

<tr>

<td>14.</td>

<td>

Computer Graphics

</td>

<td>

5

</td>

</tr>

</tbody>

</table>

</div>

</p>

<p id="future">

<element id="futureElement">Future Possibilities :

</element>

You can become someone in field of Computer.

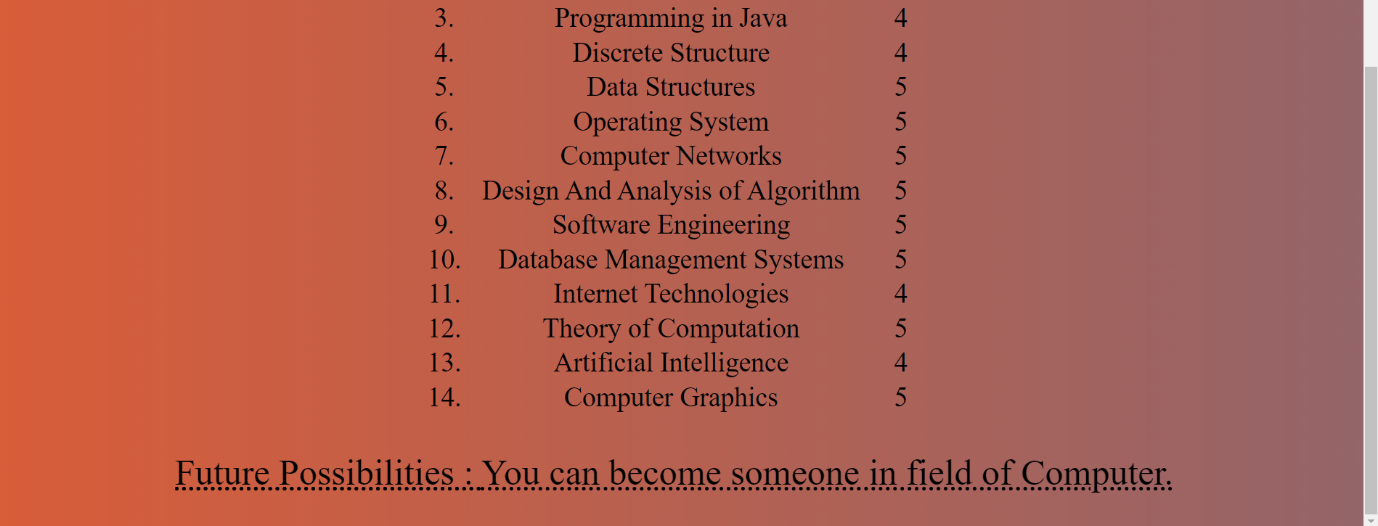
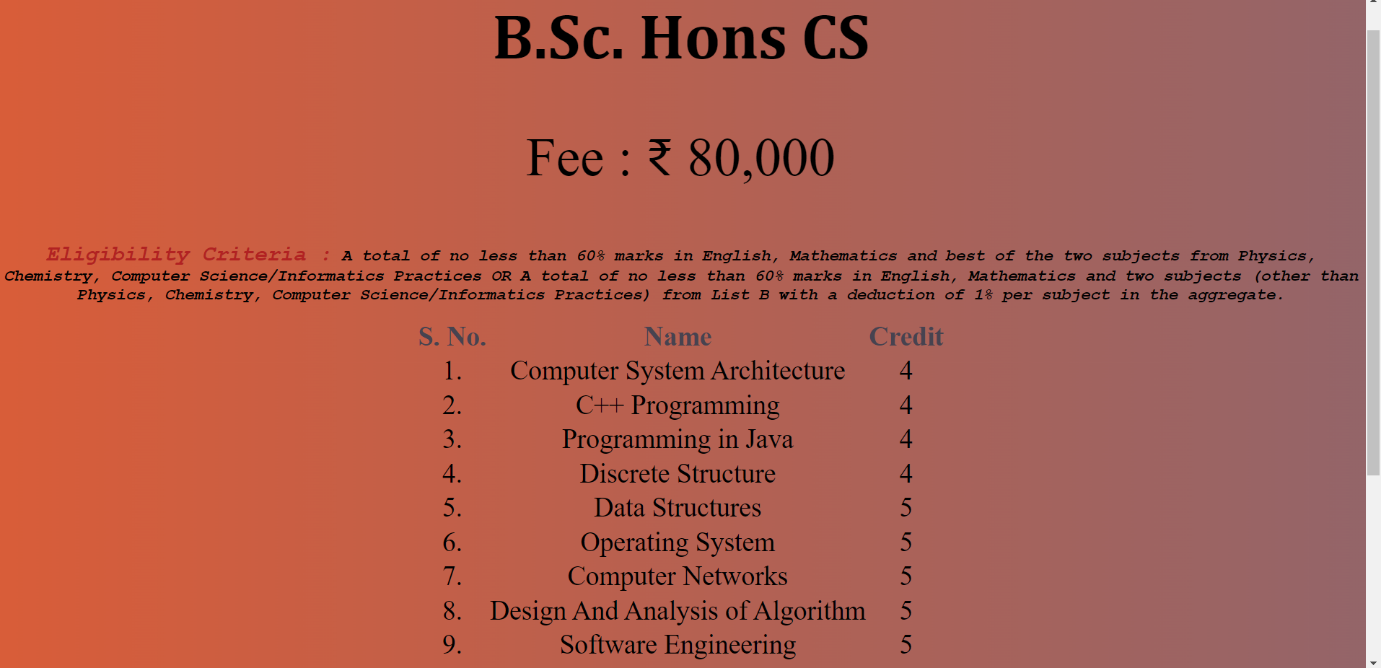
</p>

<p><button onclick="letsChange()">Re-Design</button></p>

</div>

</body>

</html>



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

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">

<title>Practical</title>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css">

</head>

<body><div class="row">

<div class="col"><div>

<h1 class="display-2 fw-semibold text-center text-primary">My Gallery</h1>

</div>

</div></div>

<div class="row" style="text-align: center;">

<div class="col text-center m-auto">

<img class="rounded" src="assets/img/b.jpg">

</div>

<div class="col text-center align-self-center">

<img class="rounded" src="assets/img/d.jpg">

</div>

<div class="col text-center">

<img class="rounded" src="assets/img/f.jpg" style="text-align: center;">

</div>

</div>

<div class="row" style="text-align: center;">

<div class="col text-center align-self-center">

<img class="rounded" src="assets/img/a.jpg">

</div>

<div class="col text-center">

<img class="rounded" src="assets/img/c.jpg">

</div>

<div class="col align-self-center">

<img class="rounded" src="assets/img/e.jpg">

</div>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>



A picture containing text, screenshot, different, person

Description automatically generated

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

var http = require('http');

http.createServer(function (req, res) {

res.write('Hi , This is a Node.js server');

res.end();

}).listen(10000);

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.

CSS File:

html {

background-color: darkslateblue;

}

input[type=text],

select,

textarea,

input[type=number],

input[type=email],input[type=password] {

width: 70%;

padding: 12px;

border: 1px solid #ccc;

border-radius: 4px;

box-sizing: border-box;

margin-top: 6px;

margin-left: 10%;

margin-bottom: 16px;

resize: vertical

}

button {

background-color: blueviolet;

color: white;

padding: 12px 20px;

border: none;

border-radius: 4px;

cursor: pointer;

margin-left: 10%;

}

.buttonGroup{

margin-left: auto;

margin-right: auto;

}

button:hover {

background-color: darkblue;

}

label {

font-size: xx-large;

margin-left: 10%;

}

.container {

width: 50%;

margin-left: 25%;

margin-top: 10%;

border-radius: 5px;

background-color: lightblue;

padding: 20px;

}

JS File:

var signInForm = true;

var signUp = '<label for="firstName">First Name : </label><br /> <input type="text" id="firstName" /><br /> <label for="lastName">Last Name : </label><br /> <input type="text" id="lastName" /><br /> <label for="emailID">Email ID : </label><br /><input type="email" id="emailID" autocomplete="username"/><br /><label for="password">Password : </label><br /><input type="password" id="password" autocomplete="new-password"/><br /><label for="confirmPassword">Confirm Password : </label><br /><input type="password" id="confirmPassword" autocomplete="new-password"/><br /><button type="button" onClick="createAccount()">Create Account</button><button type="button" onclick="changeForm()">Login</button>';

var signIn = '<label for="emailID">Email ID : </label><br /><input type="email" id="emailID" autocomplete="username"/><br /><label for="password">Password : </label><br /><input type="password" id="password" autocomplete="current-password"/><br /><div class="buttonGroup"><button type="button" onClick="login()">Login</button><button type="button" onclick="changeForm()">Sign Up</button></div>'

$(function () {

document.getElementById('changeState').innerHTML = signIn;

});

const HTTP = new XMLHttpRequest();

const url = 'http://localhost:10000/';

HTTP.open("get", url + 'welcome');

HTTP.send();

function changeForm() {

if (signInForm) {

document.getElementById('changeState').innerHTML = signUp;

signInForm = false;

} else {

document.getElementById('changeState').innerHTML = signIn;

signInForm = true;

}

}

function login() {

var password = document.getElementById("password").value;

var emailId = document.getElementById("emailID").value;

console.log(password, emailId);

HTTP.open("GET", url + 'authenticate/' + emailId + '/' + password, false);

HTTP.send();

console.log(HTTP.responseText);

if (HTTP.responseText === 'true') {

welcomeUser();

}

else {

confirm("Credential Invalid");

}

}

function createAccount() {

var password = document.getElementById("password").value;

var emailId = document.getElementById("emailID").value;

var confirmPassword = document.getElementById("confirmPassword").value;

var name = document.getElementById("firstName").value + " " + document.getElementById("lastName").value;

if (confirmPassword === password) {

HTTP.open("POST" , url+'new/' + name +'/' +emailId +'/' + password , false );

console.log( url+'new/' + name +'/' +emailId +'/' + password);

HTTP.send();

console.log(HTTP.responseText);

if(HTTP.responseText === 'true'){

confirm("Account Created Successfully");

location.href = "index.html";

}else{

confirm("Email ID already in use.Use different Email ID.")

}

}

else {

confirm("Password didn't match");

}

}

function welcomeUser() {

var welcome = '<p style = "font-size: xx-large">Welcome User !!</p>';

document.getElementById('changeState').innerHTML = welcome;

}

NodeJs File:

var mysql = require('mysql');

var express = require('express');

var app = express();

app.listen(10000, (err) => {

if (err) throw err;

console.log("Serever listening at http://127.0.0.1:10000");

})

app.use(function (\_req, res, next) {

res.setHeader('Access-Control-Allow-Origin', '\*');

res.setHeader('Access-Control-Allow-Methods', 'GET, POST, PUT, DELETE');

res.setHeader('Access-Control-Allow-Headers', 'Origin, X-Requested-With, Content-Type, Accept');

res.setHeader('Access-Control-Allow-Credentials', true);

next();

});

app.get('/welcome', (\_req, res) => {

res.write("API Working");

res.end();

})

app.get('/authenticate/:userID/:password', function (req, res) {

var email = req.params.userID.toString();

var password = req.params.password.toString();

var customQuery = "select password from userdata where email = '" + email + "'";

con.query(customQuery, function (\_err, result) {

if (result.length > 0) {

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

res.send(true);

res.end();

} else {

res.send(false);

res.end();

}

} else {

res.send(false);

res.end();

}

})

})

app.post('/new/:name/:userID/:password', function (req, res) {

var name = req.params.name.toString();

var email = req.params.userID.toString();

var password = req.params.password.toString();

var customQuery = "insert into userdata (name , email , password) values ('" + name + "','" + email + "','" + password + "')";

con.query(customQuery, function (err, result) {

if (err) {

res.send(false);

} else {

res.send(true);

console.log("User Created Successfully");

}

res.end();

})

})

var con = mysql.createConnection({

host: "hosting",

user: "serverroot",

password: "123456"

});

con.connect(function (err) {

if (err) throw err;

con.query("use prog11", function (\_err, result) {

console.log(result.message)

})

console.log("Connected!");

});

HTML File:

<!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>Index

</title>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

<script src="prog11.js"></script>

<link rel="stylesheet" href="prog11.css">

</head>

<body>

<form class="container" id="changeState"></form>

</html>

