Node Js Questions

1.Create a server in node-js to accept the request from the client. On receiving request send response either in HTML format or in Text format. Display message on console that Server running on local host

server.js

```
var http = require('http'); // Import Node.js core module

var server = http.createServer(function (req, res) {   //create web server
    if (req.url == '/') { //check the URL of the current request

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

        // set response content
        res.write('<html><body><h1>This is home Page.</h1></body></html>');
        res.end();

    }
    else if (req.url == "/student") {

        res.writeHead(200, { 'Content-Type': 'text/html' });
        res.write('<html><body><p>This is student Page.</p></body></html>');
        res.end();

    }
    else if (req.url == "/admin") {

        res.writeHead(200, { 'Content-Type': 'text/html' });
        res.write('<html><body><p>This is admin Page.</p></body></html>');
        res.end();

    }
    else
        res.end('Invalid Request!');

});

server.listen(5000); //6 - listen for any incoming requests

console.log('Node.js web server at localhost:5000 is running..')
```

2.Write a program in Node js to create your own modules to perform arithmetic operations such as addition, subtraction, multiplication, division. Import these modules to create a calculator in another nodejs file

calc.js

```
// making a module to evaluate a mathematical expression
exports.add = (a,b) => {
    return a+b;
}
exports.sub = (a,b) => {
    return a-b;
}
exports.mult = (a,b) => {
    return a*b;
}
exports.div = (a,b) => {
    return a/b
}
```

temp.js

```
// importing calc module
calc = require('./calc');

console.log("3+2="+calc.add(3,2));
console.log("3-2="+calc.sub(3,2));
console.log("3*2="+calc.mult(3,2));
console.log("3/2="+calc.div(3,2));
console.log("3/0="+calc.div(3,0));
```

```
3+2=5
3-2=1
3*2=6
3/2=1.5
3/0=Infinity
```

3.Write a program in node js to read the existing file data, display on console, and write the data in the existing file

(Hint use fs module of Node js)

```
//Write a program in node js to read the existing file data, display on console,
and write the data in the existing file



//display the data in the file on console
var http=require('http')
```

```
var fs=require('fs')

const file=fs.readFileSync('data.txt');
console.log(file.toString());

//writing the data in the file
fs.writeFile('data.txt','This is the data written in the file',function(err){
    if(err) throw err;
    console.log('Data written to file');

});

//displaying the updated file
const file1=fs.readFileSync('data.txt');
console.log(file1.toString());
```

4.Write a program in node js to read the existing file data, display on console, and delete the existing file

(Hint use fs module of Node js)

```
//Write a program in node js to read the existing file data, display on console,
and delete he existing file

//display the data in the file on console
var http=require('http')
var fs=require('fs')

const file=fs.readFileSync('data.txt');
console.log(file.toString());


//delete the file
fs.unlink('data.txt',function(err){
    if(err) throw err;
    console.log('File deleted');
})
```

5.Create a server in node-js to accept the request from the client. On receiving request send HTML form in response. Display message on console that Server running on local host

server.js

```
//Create a server in node-js to accept the request from the client. On receiving request send HTML form in response. Display message on console that Server running on local host

const express = require("express");
const app = express();

app.listen(3000, () => {
  console.log("Application started and Listening on port 3000");
});

app.get("/", (req, res) => {
  res.sendFile(__dirname + "/index.html");
});
```

index.html

```
<html>
<body>

<h2>HTML Forms</h2>

<form >
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>



</body>
</html>
```

6.Create a server in node-js to accept the request from the client. On receiving request send HTML Table in response. Display message on console that Server running on local host

server.js

```
//Create a server in node-js to accept the request from the client. On receiving request send HTML table in response. Display message on console that Server running on local host

const express = require("express");
const app = express();

app.listen(3000, () => {
  console.log("Application started and Listening on port 3000");
});
```

```
app.get("/", (req, res) => {
  res.sendFile(__dirname + "/index2.html");
});
```

index2.html

```
<!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>Time Table</title>
    <style>
        th,
        td {
            padding: 15px;
        }
    </style>
</head>

<body>

    <h1 style="text-align:center">Timetable for AY 2021-2022 (Odd Sem)</h1>

    <h3 style="text-align:center">Pushkar Waykole <br> SAP ID:60009200039</h3>


    <table border="3" cellspacing="0" align="center">

        <tr align="center">

            <td>Time</td>

            <td>Monday</td>

            <td>Tuesday</td>

            <td>Wednesday</td>
```

```
            <td>Thursday</td>

            <td>Friday</td>

            <td>Saturday</td>

        </tr>

        <tr align="center">

            <td>9 am-10 am</td>

            <td>DSA[HN]</td>

            <td> </td>

            <td rowspan="2">

                DBMS [K1 & K2] [AV]<br>[Online]<br>DSA [K3 & K4] [HN]<br>[Online]

            </td>

            <td> </td>

            <td>DSA [HN]</td>

            <td rowspan="2">

                DBMS [K3 & K4] [AV]<br>[Online]<br>DSA [K1 & K2] [HN]<br>[Online]

            </td>

        </tr>



        <tr align="center">

            <td>10:10 am-11:10 am</td>

            <td>MIS[AB]</td>

            <td rowspan="2">

                PP [K1 & K2] [PB]<br>[SPM]<br>FDA [K3 & K4] [KRS]<br>[COM]

            </td>
```

```
        <td rowspan="2">

            SDS [K1 & K2] [SS]<br>[COM]<br>PP [K3 & K4] [PB]<br>[SPM]

        </td>

        <td>FDA[KRS]</td>

    </tr>



    <tr align="center">

        <td>11:20 am-12:20 am</td>

        <td>FDA[KRS]</td>

        <td>FDA[KRS]</td>

        <td>MIS[AB]</td>

        <td>SDS[VS]</td>

    </tr>

    <tr align="center">

        <td>12:30 am -1pm</td>

        <td colspan="6" align="center">Break</td>

    </tr>

    <tr align="center">

        <td>1pm-2pm</td>

        <td>SDS[VS]</td>

        <td rowspan="2">

            FDA [K1 & K2] [KRS]<br>[COM]<br>SDS [K3 & K4] [VS]<br>[SPM]

        </td>
```

```
            <td>MIS[AB]</td>

            <td>MIS[AB] [TUT] [PM]</td>

            <td>COI[RK]</td>

            <td>DSA[HN]</td>

        </tr>

        <tr align="center">

            <td>2:10pm-3:10pm</td>

            <td>DBMS[AV]</td>

            <td>SDS[VS]</td>

            <td>IPD Discussion</td>

            <td>DBMS[AV]</td>

            <td>DBMS[AV]</td>

        </tr>

    </table>

    <table align="center">
        <tr>
            <td>Subject Names</td>
            <td>Lab Names</td>
            <td>Faculty Names</td>
        </tr>
        <tr>
            <td> MIS: Mathematics for Intelligent System <br> DSA: Data Structures
and Algorithms <br> FDA: Foundations
                of Data Analysis <br> DBMS: Database Management System <br> SDS:
Statistics for Data Science <br> PP:
                Programming with Python <br> COL: Constitution of India</td>
            <td>SPM: Software Project Management [3* Floor] <br> COM: Computing Lab
[3 Floor]</td>
            <td>AB: Prof. Alisha Bant <br>HN: Prof. Harish Narula <br>KRS: Prof.
Kriti Srivastava <br>AV: Prof. Anusha
                Vegesna <br>V5: Dr. Vaibhavi Sonetha/<br>SS: Prof. Shilan Singh
<br>PB: Prof. Pranit Bari <br>RK: Prof.
                Rupali Karande</td>
        </tr>
```

```
    </table>
</body>

</html>
```

Express Questions

1.Create a server in express js to accept the request from the client. Based on the route specified by user send the response (Hint use get method)

If route is '/'- sen d response as information which will be displayed on browser

If route is '/books' - send response as books information which will be displayed on browser

server.js

```
const express=require('express')

const app=express();

app.get('/',function(req,res){
    res.send('Hello World');
})
app.get('/books',function(req,res){
    res.send('Information about Books');
})

app.listen(3000,function(){
    console.log('Server started on port 3000');
})
```

2.Create server in express js to accept the request from the client. Based on the route specified by user send the response (Hint use both get and post method and body parser)

If route is '/'- send response as HTML form.

On form submit use post method, get the data field in form and display in on the browser

server.js

```
var express = require('express');
var app = express();

var bodyParser = require("body-parser");
app.use(bodyParser.urlencoded({ extended: false }));
```

```
app.get('/', function (req, res) {
    res.sendFile(__dirname + '/index.html');
});

app.post('/submit-student-data', function (req, res) {
    var name = req.body.firstName + ' ' + req.body.lastName;

    res.send(name + ' Submitted Successfully!');
});

var server = app.listen(3000, function () {
    console.log('Node server is running..');
});
```

index.html

```
<!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>Form </title>
</head>
<body>

    <form action="/submit-student-data" method="post">
    First name:
    <input type="text" placeholder="First name"  name="firstName" >
    <br>
    Last name
    <input type="text" placeholder="Last name" name="lastName">
    <br>
    <button type="submit">Submit</button>
</form>
</body>
</html>
```

Mongo DB Questions

Harry OP https://www.codewithharry.com/blogpost/mongodb-cheatsheet

1. **Create Student Database, create collection student information and perform insert, update, remove operation.**

Code:

```
> use Student
switched to db Student
> show dbs
Sample               0.000GB
admin                0.000GB
config               0.000GB
local                0.000GB
practical            0.000GB
user-authentication  0.000GB
> db.createCollection('test')
{ "ok" : 1 }
> db.test.insert({"name":})
uncaught exception: SyntaxError: expected expression, got '}' :
@(shell):1:23
>
> db.test.insert({"name":"op"})
WriteResult({ "nInserted" : 1 })
> show collections
test
> db.test.insert({"name":"opewrerw"})
WriteResult({ "nInserted" : 1 })
> db.test.update({"name":"op"},{"name":"Harry op","skills":"everything"},{upsert:true})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.test.insert({"Name":"carry"})
WriteResult({ "nInserted" : 1 })
> db.remove({"Name":"carry"})
uncaught exception: TypeError: db.remove is not a function :
@(shell):1:1
> db.test.remove({"Name":"carry"})
WriteResult({ "nRemoved" : 1 })
```

2. **Create Student Database, create collection student information and perform insert operation. Write the following queries:**

Mai ye wala kar rahe hu mongo db pe ss delta hu: okay

Creating database:

```
> use exam
switched to db exam
> show collections
test
> db.test.insertOne({"name":"ram","marks":93,"lives":"andheri"})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("62a361b1bffc45e092fc2fd6")
}
> db.test.find()
{ "_id" : ObjectId("62a361b1bffc45e092fc2fd6"), "name" : "ram", "marks" : 93, "lives" : "andheri" }
> db.test.insertOne({"name":"shyam","marks":91,"lives":"bandra"})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("62a361e0bffc45e092fc2fd7")
}
> db.test.insertOne({"name":"raju","marks":50,"lives":"andheri"})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("62a361f4bffc45e092fc2fd8")
}
> db.test.insertOne({"name":"ramu","marks":10,"lives":"vile parle"})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("62a3621abffc45e092fc2fd9")
}
```

Display student information who has secured more than 90%.
Display student information who failed the examination
Display student information who stays in Andheri

Queries:

```
> db.test.find()
{ "_id" : ObjectId("62a361b1bffc45e092fc2fd6"), "name" : "ram", "marks" : 93, "lives" : "andheri" }
{ "_id" : ObjectId("62a361e0bffc45e092fc2fd7"), "name" : "shyam", "marks" : 91, "lives" : "bandra" }
{ "_id" : ObjectId("62a361f4bffc45e092fc2fd8"), "name" : "raju", "marks" : 50, "lives" : "andheri" }
{ "_id" : ObjectId("62a3621abffc45e092fc2fd9"), "name" : "ramu", "marks" : 10, "lives" : "vile parle" }
> db.test.find({marks:{$gte:90}})
{ "_id" : ObjectId("62a361b1bffc45e092fc2fd6"), "name" : "ram", "marks" : 93, "lives" : "andheri" }
{ "_id" : ObjectId("62a361e0bffc45e092fc2fd7"), "name" : "shyam", "marks" : 91, "lives" : "bandra" }
> db.test.find({marks:{$lte:35}})
{ "_id" : ObjectId("62a3621abffc45e092fc2fd9"), "name" : "ramu", "marks" : 10, "lives" : "vile parle" }
> db.test.find({lives:"andheri"})
{ "_id" : ObjectId("62a361b1bffc45e092fc2fd6"), "name" : "ram", "marks" : 93, "lives" : "andheri" }
{ "_id" : ObjectId("62a361f4bffc45e092fc2fd8"), "name" : "raju", "marks" : 50, "lives" : "andheri" }
>
```

ref: https://www.tutorialspoint.com/mongodb/mongodb_query_document.htm#

Mongosh

cheetsheet mongoose cheetsheet.pdf
Youtube video Mongoose Crash Course - Beginner Through Advanced **Chota hi hai**

3. Create Mongo dB Schema using mongoose module and insert data into database
   ref: NEW LINK IS HERE

script.js

```
const mongoose = require('mongoose');
// console.log("start");
const connectionParams = {
    useNewUrlParser: true,
    useUnifiedTopology: true,
};
try {
```

```
    mongoose.connect('mongodb://0.0.0.0:27017/temp', connectionParams);
    console.log("Connected to database successfully");
} catch (error) {
    console.log(error);
    console.log("Could not connect database!");
}
// console.log("end");

const kittySchema = new mongoose.Schema({
    name: String
});

const Kitten = mongoose.model('Kitten', kittySchema);

const silence = new Kitten({ name: 'ho ja yaar' });
console.log(silence.name);

silence.save();
const silence2 = new Kitten({ name: 'ho ja yaar please' });
console.log(silence2.name);

silence2.save();
```

Isse naya database banta hai temp naam ka and kitten nam ka collection banta hai,and "`ho ja yaar`" and '`ho ja yaar please`' add hota hai

4. Create Mongo dB Schema using mongoose module and Find All data from database and display on browser

React Questions

1. Create a react application for rendering single element and rendering component having multiple elements

https://codesandbox.io/s/react-practice-question1-w3d7xb?file=/src/Components/Temp.js

Rendering single element

```
import React from "react"
import ReactDOM from "react-dom"

const element = <h1>Hello from Create React App</h1>

ReactDOM.render(element, document.getElementById("root"))
```

Rendering multiple elements

App.js

```
import logo from './logo.svg';
```

```
import './App.css';
import Temp from './Temp.js';

function Temp2(){
  return (<h3>This is temp2 <Temp /></h3>)
}
function App() {
  return (
    <div className="App">
      <Temp />
      <Temp />
      <Temp />
      <Temp />
      <Temp2 />
    </div>
  );
}

export default App;
```

Temp.js

```
import React from 'react'

const Temp = () => {
  return (
    <div>This is a temp variable</div>
  )
}

export default Temp
```

2. Create a react application for rendering components having multiple elements and reusing the components at multiple places.

Temp.js

```
import React from 'react'

function A() {
  return (
    <div>This is element 1</div>
  )
}

function B() {
  return (
    <div>This is element 2</div>
  )
```

```
}
const Temp = () => {
  return (
    <div>This is a temp variable
        <A />
        <B />
        <A />

    </div>
  )
}

export default Temp
```

3. Create a react application to build user defined component, export the component and import user defined component

Same as 1 and 2

4. Create a react application to Import and use CSS in react application

Temp.js

```
import React from 'react'
import './Temp1.css'
function A() {
  return (
    <div className='red'>This is element 1</div>
  )
}

function B() {
  return (
    <div className='blue'>This is element 2</div>
  )
}
const Temp = () => {
  return (
    <div>This is a temp variable
        <A />
        <B />
        <A />

    </div>
  )
}
```

```
export default Temp
```

Temp.css

```
.red{
    color: red;
}

.blue{
    color: blue;
}
```

5. Create a react application to implement props in react application

In this link I have used CSS as well as props. This link is solution to

2- reusing list
3- bullshit
4- CSS
5- Props

Temp.js

```
import React from 'react'
import './Temp1.css'
function A() {
  return (
    <div className='red'>This is element 1</div>
  )
}

function B() {
  return (
    <div className='blue'>This is element 2</div>
  )
}
const Temp = (props) => {
  return (
    <div>
        <h3>My name is {props.name}</h3>
        This is a temp variable
        <A />
        <B />
        <A />

    </div>
  )
}
```

```
export default Temp
```

App.js

```
import logo from './logo.svg';
import './App.css';
import Temp from './Temp.js';

function Temp2(){
  return (<h3>This is temp2 <Temp /></h3>)
}
function App() {
  return (
    <div className="App">
      <Temp name="Pushkar"/>


      <Temp name="JKB"/>



    </div>
  );
}

export default App;
```

6. Create a react application for Raising and handling events.

Link for question 6 and 7

Counter.js

```
import {useState} from 'react';

export default function Change() {
  const [isActive, setIsActive] = useState(false);

  const handleClick = () => {
    // 👇 toggle
    setIsActive(current => !current);

    // 👇 or set to true
    // setIsActive(true);
  };

  return (
```

```
    <div>
      <button
        style={{
          backgroundColor: isActive ? 'salmon' : '',
          color: isActive ? 'white' : '',
        }}
        onClick={handleClick}
      >
        Click
      </button>
    </div>
  );
}
```

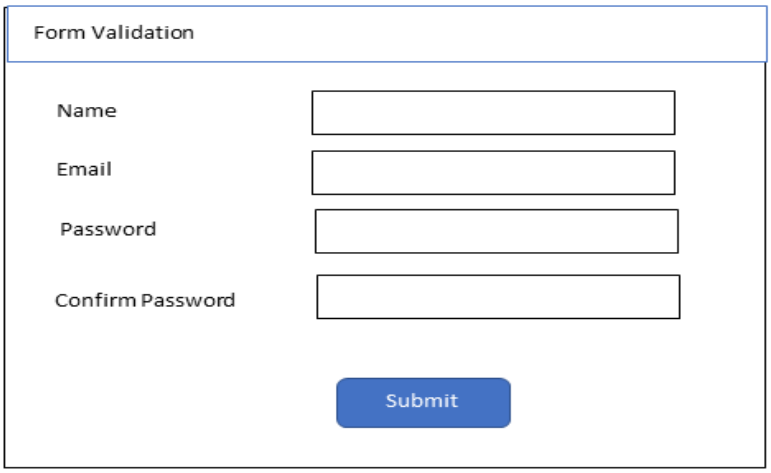
7. Create a react application to Use of react using State hook to increment and decrement value.

Counter.js

```
import React, { useState } from 'react';

const Counter = () => {

    const [bell, setBell] = useState(0);

    function increase() {
        setBell((oldbell)=>oldbell + 1);
    }
    function decrease() {
        setBell((oldbell)=>oldbell - 1);
    }
    return (
        <div>

            This is a counter

            <h3>The value of counter is {bell}</h3>
            <button onClick={increase}>Increase</button>
            <button onClick={decrease}>Decrease</button>
        </div>
    )
}

export default Counter;
```

Java Script Questions

1. Write program to perform form validation using JavaScript



index.js

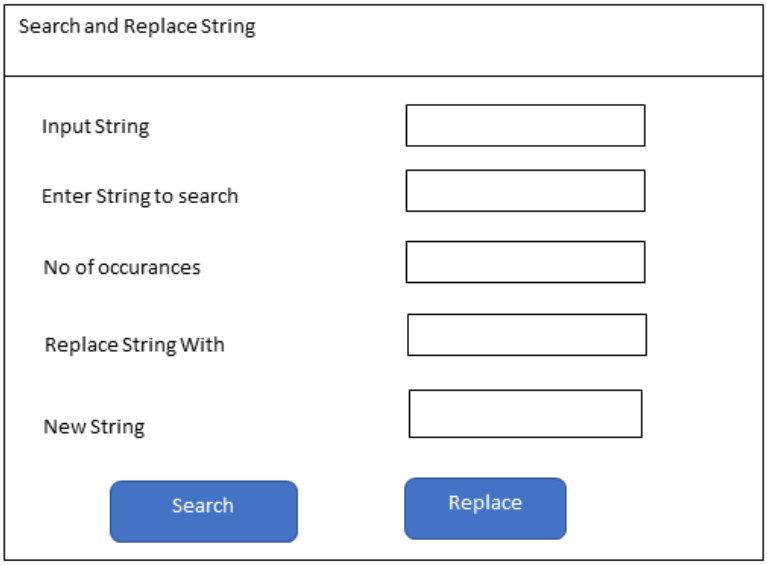
```
console.log("Hello world");
function validate() {
 if (document.myForm.name.value === "") {
   alert("Please provide a name");
   return false;
 }
 if (document.myForm.email.value === "") {
   alert("Please provide a email");
   return false;
 }
 if (document.myForm.password.value === "") {
   alert("Please provide a password");
   return false;
 }
 if (document.myForm.confirmpassword.value === "") {
   alert("Please provide a confirm password");
   return false;
 }
 if (
   document.myForm.password.value !== document.myForm.confirmpassword.value
```

```
 ) {
   alert("Your password and confirm password does not match.");
 }

 return true;
}
```

index.html

```
<!DOCTYPE html>
<html lang="en">
 <head>
   <meta charset="UTF-8" />
   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
   <meta http-equiv="X-UA-Compatible" content="ie=edge" />
   <script src="./index.js"></script>
   <title>Static Template</title>
 </head>
 <body>
   <h1>Form validation in js</h1>
   <form name="myForm" onsubmit="return validate()">
     Name:<input type="text" name="name" /><br />
     Email:<input type="email" name="email" /><br />
     Password:<input type="text" name="password" /><br />
     Confirm Password:<input type="text" name="confirmpassword" /><br />
     <input type="submit" value="submit" />
   </form>
 </body>
</html>
```

2. Write a program to search the string in the given program, display number of occurrences of string and replace the string with new string



html code:

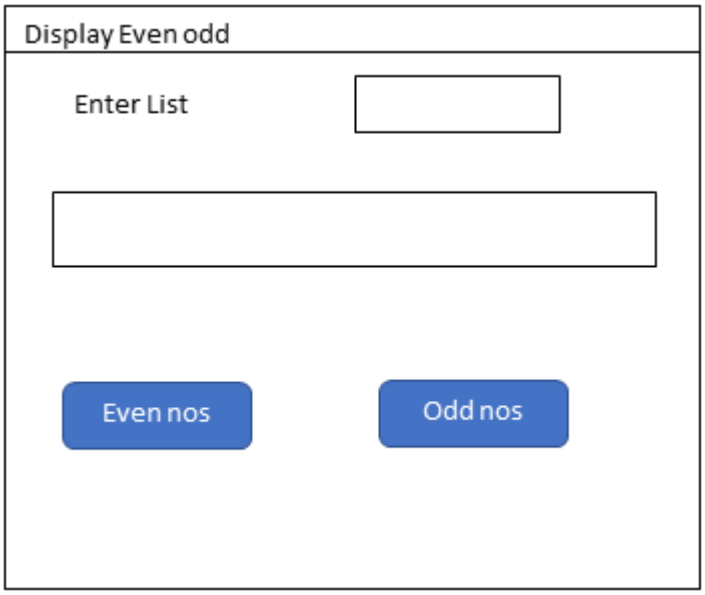
```
Input <input type="text" id="in" value="abababababcd" /> <br />
   Enter string to search <input type="text" id="pattern" value="ab" /> <br
/>
   No of occurances: <input type="text" id="occurances" /> <br />
   Replace str with <input type="text" id="replace" value="op" /> <br />
   new str <input type="text" id="answer" /> <br />
   <button onclick="solve()">Click to see magic</button>
```

Script code:

```
const solve = () => {
       const input = document.getElementById("in").value;
       console.log(input);

       const pattern = document.getElementById("pattern").value;
       console.log(pattern);

       const occur = document.getElementById("occurances");
       console.log(occur);

       const num = input.split(pattern).length - 1;
       console.log(num);
       occur.value = num;

       const re = document.getElementById("replace").value;
```

```
        console.log(re);

        const ans = document.getElementById("answer");
        let result = input.replaceAll(pattern, re);
        ans.value = result;
        console.log(result);
      };
```

3. Write JavaScript program to display the even nos and odd nos from the given list



html code

```
Enter the array <input type="text" id="in" />
    <input type="text" id="out" />
    <!-- <button onclick="sort()">Sort</button> -->
    <button onclick="even()">Even no</button>
    <button onclick="odd()">Odd no</button>
```

Script code:

```
const odd = () => {
        const arr = document.getElementById("in").value;
        console.log(arr);
        const a = arr.split(" ");
        console.log(a);
        const nums = a.map((str) => {
          return Number(str);
        });
        console.log(nums);
        const n = nums.filter((num) => {
```

```
        return num % 2 != 0;
      });
      const output = document.getElementById("out");
      output.value = n;
    };
    const even = () => {
      const arr = document.getElementById("in").value;
      console.log(arr);
      const a = arr.split(" ");
      console.log(a);
      const nums = a.map((str) => {
        return Number(str);
      });
      console.log(nums);
      const n = nums.filter((num) => {
        return num % 2 == 0;
      });
      const output = document.getElementById("out");
      output.value = n;
    };
```

4. Write a program in JavaScript to take a list of numbers from user and double all the numbers and display the doubled list

Html code:

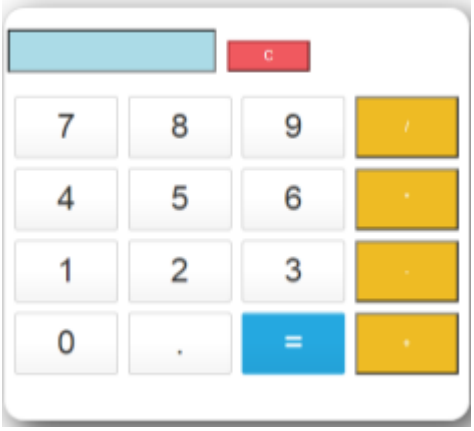
```
Input <input type="text" id="in" value="1 2 3 4" /><br />
   doubled <input type="text" id="ans" /><br />
   <button onclick="double()">Double</button>
```

Script code:

```
const double = () => {
      const input = document.getElementById("in").value;
      const arr = input.split(" ");
      const nums = arr.map((str) => {
        return Number(str);
      });
      console.log(nums);
      const ans = nums.map((n) => {
        return n * 2;
```

```
      });
      console.log(ans);

      document.getElementById("ans").value = ans;
    };
```

5. Program to design a calculator using JavaScript



calc.js

```
function display(val) {

  document.getElementById("result").value += val;

  return val;

}

function solve() {

  let x = document.getElementById("result").value;

  let y = 0;

  if (x.includes("Sqrt")) {

    let number = x.split("Sqrt");

    let result = parseInt(number[1]);

    console.log(number);

    y = Math.sqrt(result);

  } else if (x.includes("log")) {
```

```
    let result = x.split("log");
    let number = parseInt(result[1]);
    y = Math.log(number);
  } else if (x.includes("^")) {
    let result = x.split("^");
    y = Math.pow(parseInt(result[0]), parseInt(result[1]));
  } else {
    y = eval(x);
  }
  document.getElementById("result").value = y;
  return y;
}
function clearScreen() {
  document.getElementById("result").value = "";
}
```

index.html

```
<!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">
```

```
    <script src="./calc.js" type="text/javascript"></script>

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

    <title>Calculator-JS</title>

</head>

<body>



    <h1 style="text-align:center">Calculator App</h1>

    <div class="container">

    <br>

    <table>

        <tr>

          <td colspan="3"><input type='text' id='result' class ='screen'
style="text-align: right;"></td>

          <td>



          </td>

</tr>

    </table>

    <div class="keys">

        <input type="button" value="^" class="button"
onClick="display('^')"></input>

        <input type="button" value="log" class="button"
onClick="display('log')"></input>

        <input type="button" value="Sqrt" class="button"
onClick="display('Sqrt')"></input>
```

```
        <input type='button'  value = 'C' onclick="clearScreen()"
class="button c-button"/>

    <input type="button" value="7" class="button"
onClick="display('7')"></input>

    <input type="button" value="8" class="button "
onClick="display('8')"></input>

    <input type="button" value="9" class="button"
onClick="display('9')"></input>

    <input type="button" value="/" class="operator"
onClick="display('/')"></input>

    <input type="button" value="4" class="button"
onClick="display('4')"></input>

    <input type="button" value="5" class="button"
onClick="display('5')"></input>

    <input type="button" value="6" class="button"
onClick="display('6')"></input>

    <input type="button" value="*" class="operator"
onClick="display('*')"></input>

    <input type="button" value="1" class="button"
onClick="display('1')"></input>

    <input type="button" value="2" class="button"
onClick="display('2')"></input>

    <input type="button" value="3" class="button"
onClick="display('3')"></input>

    <input type="button" value="-" class="operator" onClick="display('-
')"></input>

    <input type="button" value="0" class="button"
onClick="display('0')"></input>
```

```
    <input type="button" value="." class="button"
onClick="display('.')"></input>

    <input type="button" value= "=" class="button equal-sign"
onClick="solve()"></input>

    <input type="button" value="+" class="operator"
onClick="display('+')"></input>

</div>

</div>

</body>

</html>
```

css

```
.container {

  border: 1px solid #cccccc;

  background: white;



  box-shadow: 10px 10px 30px 0px rgba(0, 0, 0, 0.75);



  border-radius: 20px;



  position: absolute;



  top: 55%;



  left: 50%;
```

```
  transform: translate(-50%, -50%);

  width: 450px;

  height: 600px;
}

.keys {
  display: grid;

  grid-template-columns: repeat(4, 1fr);

  grid-gap: 10px;

  padding: 10px;
  margin: auto;
}

.button {
  height: 60px;

  padding: 3px;
  font-weight: bolder;
```

```
  border-radius: 50%;


  background-color: cadetblue;
  border: none;
  font-size: 2rem;


  color: #333;


  background-image: linear-gradient(
    to bottom,
    transparent,
    transparent 50%,
    rgba(0, 0, 0, 0.04)
  );


  text-shadow: 0 1px rgba(255, 255, 255, 0.4);
}


.button:hover {
  background-color: rgb(163, 235, 238);
}


.operator {
```

```
  color: black;


  background-color: rgb(62, 62, 201);
  border: none;
  font-size: 2rem;
  border-radius: 25px;
}
.operator:hover {
  background-color: rgb(139, 228, 231);
}


.clear {
  background-color: white;


  border-color: orange;


  color: black;
  display: inline-block;
  width: 80px;
  height: 40px;
  margin-left: 135px;
  margin-bottom: 12px;
  /* border: none; */
```

```
  border-radius: 30px;
}

.clear:hover {
  background-color: orange;
}

.equal-sign {
  background-color: orange;
  border: none;
  border-radius: 15px;
  color: #fff;
}

.equal-sign:hover {
  background-color: #4e9ed4;
}

.screen {
  background-color: rgb(171, 219, 231);

  justify-content: left;
```

```
  color: black;

  font-size: large;

  width: 113%;

  height: 100px;

  cursor: default;

  padding: 10px;

  padding-left: 40%;

  margin: auto;

  margin-bottom: 10px;
  border: none;
  margin-left: 10px;
  border-radius: 15px;
}
.c-button {
  background-color: orange;
```

```
  border-radius: 30px;

  color: white;

}

.c-button:hover {

  background-color: orange;

}
```

6. Write a JavaScript program to sort the items of an array.

index.html

```
<!DOCTYPE html>
<html lang="en">
 <head>
   <meta charset="UTF-8" />
   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
   <meta http-equiv="X-UA-Compatible" content="ie=edge" />
   <script>
     function sort() {
       const arr = document.getElementById("in").value;
       console.log(arr);
       const a = arr.split(" ");
       console.log(a);
       const nums = a.map((str) => {
         return Number(str);
       });
       console.log(nums);
       console.log(nums.sort());
       var g = document.getElementById("out");
       g.value = nums.sort();
     }
   </script>
   <title>Static Template</title>
 </head>
 <body>
   Enter the array <input type="text" id="in" />
```

```
   <input type="text" id="out" />
   <button onclick="sort()">Sort</button>
 </body>
</html>
```

7. Create a function that takes two numbers as arguments (number, length) and returns an array of multiples of number until the array length reaches length.

```
const arrayOfMultiples = (a, b) => {
       let arr = [];
       let x = 1;
       for (let i = 1; i <= b; i++) {
         arr.push(a * x);
         x++;
       }
       return arr;
     };

     console.log(arrayOfMultiples(7, 5));
```

8. Create a function that determines whether a number is Oddish or Evenish. A number is Oddish if the sum of all its digits is odd, and a number is Evenish if the sum of all its digits is even. If a number is Oddish, return "Oddish". Otherwise, return "Evenish".

```
const oddOrEven = (n) => {
       let a = n.toString();
       let b = 0;
       for (let i = 0; i < a.length; i++) {
         b += parseInt(a[i]);
       }
       if (b % 2 === 1) {
         return "Oddish";
       } else return "Evenish";
     };

     console.log(oddOrEven(45));
```

9. Create a function that will return the total number of digits in the given no as 234123 has 6 digits and Sum of all the digits

```
const q9 = (n) => {
       let a = n.toString();
       let b = 0;
       let count = a.length;
       for (let i = 0; i < a.length; i++) {
         b += parseInt(a[i]);
       }
       console.log(b, count);
     };
```

10. Write a JavaScript program to test whether the first character of a string is uppercase or not.

```
const check = (n) => {
       let a = n.toString();
       let f = a[0];
       if (f == f.toUpperCase()) {
         console.log("Yes");
       } else {
         console.log("No");
       }
     };

     console.log(check("Sdfsd"));
```

11. Write a JavaScript program to count and display the items of a dropdown list, in an alert window

html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset=utf-8 />
    <style type="text/css">
        body {
            margin: 30px;
        }
```

```
    </style>
    <title>Count and display items of a dropdown list - w3resource</title>
</head>
<body>
    <form>
        Select your favorite Color :
        <select id="mySelect">
            <option>Red</option>
            <option>Green</option>
            <option>Blue</option>
            <option>White</option>
        </select>
        <input type="button" onclick="getOptions()" value="Count and Output all
items">
    </form>
</body>
</html>
```

js

```
function getOptions() {
  var x = document.getElementById("mySelect");
  var txt1 = "No. of items in dropdown is : ";
  var i;
  l = document.getElementById("mySelect").length;
  txt1 = txt1 + l;
  for (i = 0; i < x.length; i++) {
    txt1 = txt1 + "\n" + x.options[i].text;
  }
  alert(txt1);
}
```

ref : [codepen]

HTML and CSS Questions

1. Create a static web page using HTML.

2. Create a class timetable using HTML.

Class: S.E (Sem-III)

Timetable for AY 2021-2022 (Odd Sem)

WEF: 10th Nov 2021

Time	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
9 am-10am	DSA (HN)		DBMS [K1 &K2] (AV) [Online]		DSA (HN)	DBMS [K3 &K4] (AV) [Online]
10:10 am– 11: 10am	MIS (AB)	PP [K1 & K2] (PB) [SPM]	DSA [K3 & K4] (HN) [Online]	SDS [K1 & K2] [SS] [COM]	FDA (KRS)	DSA [K1 & K2] (HN) [Online]
11:20am – 12:20pm	FDA (KRS)	FDA [K3 & K4] (KRS) [COM]	FDA (KRS)	PP [K3 & K4] (PB) [SPM]	MIS (AB)	SDS (VS)
12: 30pm– 1pm	Break					
1pm- 2pm	SDS (VS)	FDA [K1 & K2] (KRS) [COM]	MIS (AB)	MIS (AB)[TUT] [PM]	COI (RK)	DSA (HN)
2:10pm-3:10pm	DBMS (AV)	SDS [K3 & K4] (VS) [SPM]	SDS (VS)	IPD Discussion	DBMS (AV)	DBMS (AV)

Subject Names
MIS: Mathematics for Intelligent System
DSA: Data Structures and Algorithms
FDA: Foundations of Data Analysis
DBMS: Database Management System
SDS: Statistics for Data Science
PP: Programming with Python
COI: Constitution of India

Lab Names
SPM: Software Project Management [3rd Floor]
COM: Computing Lab [3rd Floor]

Faculty Names
AB: Prof. Alisha Banz
HN: Prof. Harish Narula
KRS: Prof. Kriti Srivastava
AV: Prof. Anusha Vegesna
VS: Dr. Vaibhavi Sonetha/ SS: Prof. Shilank Singh
PB: Prof. Pranit Bari
RK: Prof. Rupali Karande

3. Create a registration form using HTML.

VIP Registration

Prefix

Name First Name Last Name

Preferred Pronouns

Email sample@example.com

Work Phone

Cell Phone

Job Title

Company

Flight arrival time (to be picked up)

Flight departure time (to be dropped off)

Where do you want to be picked up?

Where do you want to be dropped off?

Will your spouse attend? Leave name and contact info if so.

Are you interested in attending the VIP Networking event?

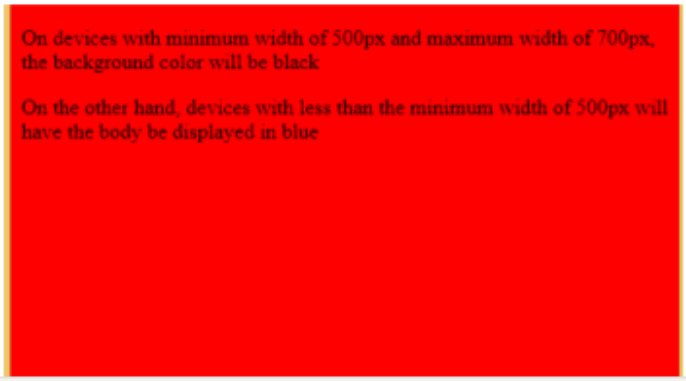
Yes No

Would you like to be sent event gifts?

Yes No

4. Design a web page using External or Embedded Style Sheet.

5. Design a responsive web page using media queries and CSS3.



6. Design a web page using Bootstrap.

7. Design a resume using Bootstrap.

8. Design the admission form using Bootstrap.