

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

ANS-Q1 NODE

```
const http=require('http')
const fs=require('fs')
const path=require('path')
const port=8000
const server=http.createServer((req,res)=>{
  if(req.url=='/'){
    res.end("hello world")
  }
  else if(req.url=='/home'){
    res.end("hello home")
  }
})
server.listen(port,()=>{
  console.log("listennig")
})
```

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

ANS- //Q2

```
//practice.js
exports.add=function(x,y){
  return x+y;
};
exports.sub=function(x,y){
  return x-y;
};
exports.mul=function(x,y){
  return x*y;
};
exports.div=function(x,y){
  return x/y;
};
//any other js file

var calc=require('./practice');
var a=10,b=5;
console.log(calc.add(a,b));
console.log(calc.sub(a,b));
console.log(calc.mul(a,b));
console.log(calc.div(a,b));
```

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)

ANS-

Q3 NODE

```
var fs=require("fs");
var data=fs.readFileSync('input.txt');
console.log(data.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)

ANS-

Q4 NODE

```
var fs=require("fs");
var data=fs.readFileSync('input.txt');
console.log(data.toString());
fs.unlink("input.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

```
ANS- const http=require('http')
const fs=require('fs')
const path=require('path')
const port=3000
const server=http.createServer((req,res)=>{
    if(req.url=='/'){
        res.end("hello world")
    }
    else if(req.url=='/form'){
        fs.readFile('form.html','utf-8',(err,data)=>{
            res.end(data)
        })
    }
})
server.listen(port,()=>{
    console.log("listen")
})
```

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

ANS-Q6 NODE

```
const http=require('http')
const fs=require('fs')
const path=require('path')
const port=3000
const server=http.createServer((req,res)=>{
  if(req.url=='/'){
    res.end("hello world")
  }
  else if(req.url=='/table'){
    fs.readFile('timetable.html','utf-8',(err,data)=>{
      res.end(data)
    })
  }
})
server.listen(port,()=>{
  console.log("listen")
})
```

Express Questions

1. Create 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 '/' - send response as information which will be displayed on browser

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

ANS- Q1 EXPRESS

```
const http=require('http')
const fs=require('fs')
const express=require('express')
const app=express()
const path=require('path')
const port=5000
app.get('/',(req,res)=>{
  res.end('this is homepage')
})
app.get('/books',(req,res)=>{
  res.end('this is book')
})
app.listen(port,()=>{
  console.log("listening")
})
```

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

ANS-Q2 EXPRESS

```
var express=require('express');
var app=express();
const port=8000
var bodyParser=require("body-parser");
app.use(bodyParser.urlencoded({extended:false}));
app.get('/form1',function(req,res){
    res.sendFile(__dirname+'/input.html')
});
app.post('/form',function(req,res){
    var name=req.body.Fname+' '+req.body.Lname;
    res.send(name+' submitted succesfully');
});
app.listen(port,function(){
});
```

Mongo DB Questions

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

Display student information who has secured more than 90%.

Display student information who failed the examination

Display student information who stays in Andheri

3. Create Mongo dB Schema using mongoose module and insert data into database
4. Create Mongo dB Schema using mongoose module and Find All data from data base and display on browser
5. Create Mongo dB Schema using mongoose module and find first data from data base and display on the browser

```
ANS-const mongoose = require('mongoose');
const { Course } = require('./model.js');
const express = require('express')
const bodyParser = require('body-parser')
```

```

const app = express()
const parser = bodyParser.urlencoded({extended: true})

mongoose.connect('mongodb://localhost:27017/test-db',
  {
    useNewUrlParser: true,
    useUnifiedTopology: true
  });

var Schema = mongoose.Schema
const Peoples = mongoose.model('People_names', new Schema(
  {
    fname: String,
    lname: String,
  }
), 'people');

app.get('/', (req, res) => {
  console.log(__dirname)
  res.sendFile(__dirname + '/index.html')
})

app.get('/data', (req, res) => {
  Peoples.find({}, function(err, collection){
    res.send(collection)
  });
})

app.post('/fill-data', parser, (req, res) => {
  const data = {
    fname: req.body.fname,
    lname: req.body.lname
  }
  Peoples.insertMany(data, (err, value) => {
    if (err) {
      console.log(err)
    } else {
      console.log("Saved succesfully")
    }
  })
  res.redirect('/data')
})

app.listen(8000)

```

React Questions

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

ANS-

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

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

4. Create a react application to Import and use CSS in react application
5. Create a react application to implement props in react application
6. Create a react application for Raising and handling event.
7. Create a react application to Use of react using State hook to increment and decrement value.

```
ANS- import './App.css';

import React,{useState} from 'react';

function App() {
  const[count,setCount]=useState(0);
  function decrease(){
    setCount(count-1)
  }
  function increase(){
    setCount(count+1)
  }
  return (
    <div className="App">
      <button onClick={decrease}>-</button>
      {count}
      <button onClick={increase}>+</button>
    </div>

  );
}

export default App;
```

Java Script Questions

1. Write program to perform form validation using JavaScript

Form Validation

Name

Email

Password

Confirm Password

Submit

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

Search and Replace String

Input String

Enter String to search

No of occurances

Replace String With

New String

Search

Replace

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

Display Even odd

Enter List

Even nos

Odd nos

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

5. Program to design a calculator using JavaScript

C

7

8

9

/

4

5

6

*

1

2

3

*

0

.

=

*

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

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.

ANS-`<!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>
  multiple:<input type="number" id="multiple">
  length:<input type="number" id="length">
  <input type="submit" onclick="arraymulti()">
  <p id="answer" value=" "></p>
  <script>
    function arraymulti(){
      let a=document.getElementById("multiple").value
      let b=document.getElementById("length").value
      let arr=[]
      let x=1
      for(let i=1; i<=b;i++){
        arr.push(a*x);
        x++
      }
      document.getElementById("answer").innerHTML=arr
      console.log(arr)
    }
  </script>

</body>
</html>
```

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".

ANS-`<!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>
  ENter a number:<input type="number" value="0" id="num">
  <input type="submit" onclick="tocheck()">
  <p id="result" value=" "></p>
```



```

<script>
    function tocheck(){
        let n=document.getElementById("num").value
        let a=n.toString();
        let b=0
        for(let i=0;i<a.length;i++){
            b+=parseInt(a[i]);
        }
        if(b%2===1){
            document.getElementById("result").innerHTML="Oddish";
            console.log("oddish")
        }
        else{
            document.getElementById("result").innerHTML="Evenish"
            console.log("evenish")
        }
    }
};
</script>
</body>
</html>

```

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

```

ANS-<!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>
    Enter number:<input type="number" id="number">
    <input type="submit" onclick="tosum()">
    <br>
    Sum of digits:<p id="sum" value=" "></p>
    Number of digits:<p id="length" value=" "></p>
    <script>
        function tosum(){
            let n=document.getElementById("number").value
            var a=n.toString();
            let sum=0;
            for(let i=0;i<a.length;i++){
                sum+=parseInt(a[i]);
            }
            document.getElementById("sum").innerHTML=sum
            document.getElementById("length").innerHTML=a.length
        }
    </script>
</body>
</html>

```

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

ANS-`<!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>
  Enter string:<input type="text" id="string">
  <input type="submit" onclick="toCheck()">
  <p id="answer" value=" "></p>
  <script>
    function toCheck(){
      let str=document.getElementById("string").value
      let strch=/^[A-Z]/;
      if(strch.test(str)){
        document.getElementById("answer").innerHTML="First element
uppercase"
        console.log("Uppercase")
      }
      else{
        document.getElementById("answer").innerHTML="First element
lowercase"
        console.log("Lowercase")
      }
    }
  </script>
</body>
</html>
```

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

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: 10 th 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 Bang
HN: Prof. Harish Narula
KRS: Prof. Kriti Srivastava
AV: Prof. Anusha Yegesna
VS: Dr. Vaibhavi Sonetha/ SS: Prof. Shilani Singh
PB: Prof. Pranit Bari
RK: Prof. Rupali Karande.

3. Create a registration form using HTML.

VIP Registration

Prefix <input style="width: 100%;" type="text"/>	Where do you want to be picked up? <input style="width: 100%;" type="text"/>
Name <input style="width: 45%;" type="text"/> First Name <input style="width: 45%;" type="text"/> Last Name	
Preferred Pronouns <input style="width: 100%;" type="text"/>	Where do you want to be dropped off? <input style="width: 100%;" type="text"/>
Email <input style="width: 100%;" type="text" value="sample@example.com"/>	
Work Phone <input style="width: 100%;" type="text"/>	Will your spouse attend? Leave name and contact info if so. <input style="width: 100%;" type="text"/>
Cell Phone <input style="width: 100%;" type="text"/>	
Job Title <input style="width: 100%;" type="text"/>	Are you interested in attending the VIP Networking event? <input type="radio"/> Yes <input type="radio"/> No
Company <input style="width: 100%;" type="text"/>	
Flight arrival time (to be picked up) <input style="width: 100%;" type="text"/>	Would you like to be sent event gifts? <input type="radio"/> Yes <input type="radio"/> No
Flight departure time (to be dropped off) <input style="width: 100%;" type="text"/>	

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

Web Technology C Programming Contact

Search

Web Technology

HTML stands for HyperText Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document's structure which defines the structure of web pages. HTML is a markup language that is used by the browser to transform text, images, and other content to display it in the required format.

C Programming

C is a procedural programming language. It was initially developed by Dennis Ritchie as a system programming language to write operating system. The main features of C language include low-level access to memory, simple set of keywords, and clean style, these features make C language suitable for system programming like operating system or compiler development.

Java

Java has been one of the most popular programming languages for many years. Java is Object Oriented. However it is not considered as pure object oriented as it provides support for primitive data types (like int, char, etc.) The Java codes are first compiled into byte code (machine-independent code). Then the byte code is run on Java Virtual Machine (JVM) regardless of the underlying architecture.

Copyright © All rights are reserved

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

On devices with maximum width of 500px and maximum width of 700px, the background color will be black

On the other hand, devices with less than the maximum width of 500px will have the body be displayed in blue

On devices with minimum width of 500px and maximum width of 700px, the background color will be black

On the other hand, devices with less than the minimum width of 500px will have the body be displayed in blue

6. Design a web page using Bootstrap.

7. Design a resume using Bootstrap.

