Web Computing Question Bank

Q.1) Write the code to process online Alumni information for your college. Create form to get name, date of birth and email id. Use check boxes for taking hobbies and radio buttons for selecting branch. Write JavaScript code to validate the following.
-User has to fill all the fields prior the form submission.
-Valid e-mail id (@ and .)
-Age validation using DOB (>=22 years)

Code :

Index.html

```
<!DOCTYPE html>
<head>
  <title>Document</title>
  <style>
    label,input[type='text'],input[type='date']{
      display: block;
    }
    label{
      margin-top: 20px;
    }
    button{
      display: block;
      margin-top: 30px;
    }
  </style>
</head>
<body>
  <h1>Alumni Registration Form</h1>
    <label for="name">Name :</label>
    <input type="text" id="name" required>
    <label for="dob">DOB :</label>
    <input type="date" id="dob" required>
    <label for="email">Email :</label>
    <input type="text" id="email" required>
    <label>Hobbies :</label>
    <input type="checkbox" class="hobbies" value="Swimming"><span>Swimming</span>
    <input type="checkbox" class="hobbies" value="Reading-Books"><span>Reading Books</span>
    <input type="checkbox" class="hobbies" value="Cycling"><span>Cycling</span>
    <label>Branch :</label>
    <input type="radio" name="branch" value="A.I.D.S"><span>A.I.D.S</span>
    <input type="radio" name="branch" value="COMPS"><span>COMPS</span>
    <input type="radio" name="branch" value="ECS"><span>ECS</span>
    <input type="radio" name="branch" value="MECH"><span>MECH</span>
    <button type="submit" onclick="submit()">Submit</button>
```

```
</body>
<script src="./script.js"></script>
</html>
```

script.js

```
const namee = document.getElementById('name');
const email = document.getElementById('email');
const dob = document.getElementById('dob');
const hobbies = document.getElementsByClassName('hobbies');
const radio = document.getElementsByName('branch');

function validatgmail(mail){
   var mailformat = /^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/;
   if(mail.match(mailformat)){
      return true;
   }
   alert('Enter Proper Email Address')
   location.reload();
}

function validateage(dob){
   year = dob.slice(0,4)
   month = dob.slice(5,7)
   day = dob.slice(8,10)

   const d = new Date()
   cur_month = d.getMonth()
   cur_year = d.getFullYear()

   if(parseInt(year) < cur_year-22){
      return true
   }
   else if(parseInt(year) == cur_year-22){
      if(parseInt(month) <= cur_month){
         return true
      }
   }

   alert('Please Enter a valid Age ');
   location.reload()
}

function submit(){
   hobbiess = []
   const name_val = namee.value;
   const email_val = email.value;
```

```
    const dob_val = dob.value;

    if(hobbies[0].checked){
        hobbiess.push(hobbies[0].value);
    }
    if(hobbies[1].checked){
        hobbiess.push(hobbies[1].value);
    }
    if(hobbies[2].checked){
        hobbiess.push(hobbies[2].value);
    }

    if(radio[0].checked){
        branch = radio[0].value;
    }
    else if(radio[1].checked){
        branch = radio[1].value;
    }
    else if(radio[2].checked){
        branch = radio[2].value;
    }

    mail_validated = validatgmail(email_val)
    age_validated = validateage(dob_val)

    if(mail_validated == true && age_validated == true){
        alert(`Hi ${name_val} ! Your Form is Submitted`)
    }
    else{
        alert('Something went wrong')
        location.reload()
    }

}
```

Q.2) Write a JavaScript program to calculate multiplication and division of two numbers (input from user)

Code :

```
<!DOCTYPE html>
head>
   <title>Calculator</title>
</head>
<body>
   <h1>Calculator</h1>
   <label for="one">1st Number : </label> <br>
   <input type="number" id="one"> <br>
   <label for="two">2nd Number : </label> <br>
   <input type="number" id="two"> <br> <br>
   <button onclick="multiply()">Multiply</button>
   <button onclick="divide()">Divide</button>
   <br>
   <span id="result"></span>
</body>
<script>
   const one = document.getElementById('one')
   const two = document.getElementById('two')
   const result = document.getElementById('result')

   function multiply(){
      one_val = one.value
      two_val = two.value
      res = one_val * two_val
      result.innerHTML = `The Result is ${res}`
   }

   function divide(){
      one_val = one.value
      two_val = two.value
      res = one_val / two_val
      result.innerHTML = `The Result is ${res}`
   }

</script>
</html>
```

Q.3) Write a program that takes input as student rollno, name and marks of 3 subjects. Calculate total and average marks. If average>80 the grade is A, if average is >60 the grade is B if it is >40 then C else fail. The grade is allocated only if he is passed in all the subjects.

Code :

```
<!DOCTYPE html>
<head>
   <title>Document</title>
</head>
<body>
   <h1>STUDENT FORM</h1>
   <label for="rollno">Roll no: </label>
   <input type="number" id="rollno"><br>
   <label for="name">Name: </label>
   <input type="text" id="name"><br>
   <label for="sub1">Subject 1:  </label>
   <input type="number" id="sub1"><br>
   <label for="sub2">Subject 2:  </label>
   <input type="number" id="sub2"><br>
   <label for="sub3">Subject 3:  </label>
   <input type="number" id="sub3"><br>
   <button onclick="calculate()">Calculate</button>
   <span id="result"></span>
</body>
<script>
  function calculate(){
    sub1=document.getElementById('sub1').value
    sub2=document.getElementById('sub2').value
    sub3=document.getElementById('sub3').value
      const total= sub1+sub2+sub3;
      const avg= total/3;
      if(avg>=80 && avg<=60){
        result.innerHTML= 'GRADE IS A'
      }
      else if(avg>=60 && avg<=40){
        result.innerHTML= 'GRADE IS B'
      }
      else if(avg<40){
        result.innerHTML= 'GRADE IS C'
      }
      else{
        result.innerHTML= 'FAIL'
      }

  }
</script></html>
```

Q.4) Write a Javascript arrow function with parameter (= ‘Visitor’).

Code :

```
<!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>
   <h1>Javascript Arrow Function</h1>
   <p>This is demo arrow function with argument</p>
   <p id="demo"></p>
</body>
<script>
   const namee = prompt('Enter your Name :')
   const visitor = (name) =>{
      document.getElementById('demo').innerHTML = `Hi ! ${namee}`
   }
   visitor(namee)
</script>
</html>
```

Q.5) The web page displays the digital clock that updates the current time after 1000 milliseconds time interval. Write the code for webpage along with JavaScript.

Code :

```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>Document</title>
</head>
<body>
   <span id="hours"></span>
   <span id="minutes"></span>
   <span id="seconds"></span>
</body>
<script>
   setInterval(changeTime,1000)
   function changeTime(){
      const d = new Date();
      const sec = d.getSeconds();
      const min = d.getMinutes();
      const hour = d.getHours();
      document.getElementById('hours').innerHTML = hour + ':'
      document.getElementById('minutes').innerHTML = min + ':'
      document.getElementById('seconds').innerHTML = sec
   }
</script>
</html>
```

Q.6) The web page has header h2 and paragraph id=demo. Write the JavaScript with inheritance relations class Mall as super class and class shop as sub class with showshop member function

Code :

```
<!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>
   <h2>Javascript Class Inheritance</h2>
   <p id="demo"></p>
</body>
<script>
   class Mall{
      constructor(text){
         this.text = text;
      }
      show(){
         document.getElementById('demo').innerHTML = this.text;
      }
   }

   class Shop extends Mall{
      constructor(text){
         super(text);
      }
   }

   let display = new Shop('Domino is present in the Select City Walk Mall');
   display.show();

</script>
</html>
```

Q.7) Design the webpage that is dynamically changed after the onclick button event.

Code :

```
<!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>
   <style>
      body{
         text-align: center;
      }
   </style>
</head>
<body>
   <h1 id="heading">Design Thinking</h1><br>
   <img src="" alt=""><br>
   <button onclick="addimage()">Click here</button>
</body>
<script>
function addimage(){
   head_val=document.getElementById('heading')
   head_val.innerHTML='Five Steps of Design Thinking'
   head_val.style.background='yellow'
   head_val.style.display='inline-block'
   document.getElementsByTagName('img')[0].src='./img1.jpg'
}
</script>
</html>
```

Q.8) Design React front end that displays unordered list of square of an integer array elements. [1 2 3 4 5]. Illustrate the React features usage like Virtual DOM, JSX, single page application in your program.

Code :

App.js

```
function App() {
  const names = [1,2,3,4,5]
    return (
      <ul>
      {names.map(name => <li>{name*name}</li>)}
        </ul>
    )
}

export default App;
```

Q.9) Design one of the client side application using React Components shown below. Discuss the State and Properties of the two types of React components.

Code :

NewComp.js

```
import React, { Component } from 'react';
import bell1 from './bell.png';
import bell2 from './bell2.png';

class NewComp extends Component {
   constructor(props) {
     super(props)

     this.state = {
        message : "Subscribe to MyShopWebsite",
        sub: "subscribe",
        imageURL: bell1
     };
   }
   styles = {
      color : "purple",
      fontstyle : "italic"
   }
   Buttonchange = ()=>{
      this.setState({
```

```
        message: "Hit the bell icon to get Updates",
        sub : "Subscribed",
      })
  };
  imageChange=()=>{
    this.setState({
      message: "Thank you. Do Shopping",
      imageURL : bell2
    })
  };
  render() {
    return (
      <div className='App'>
        <h3 style={this.styles}>{this.state.message}</h3>
        <button onClick={this.Buttonchange}>{this.state.sub}</button>
        <p/>
        <img
          style={{width:"30px", height:"30px"}}
          src={this.state.imageURL}
          alt="Image is not present"
          onClick={this.imageChange} />
      </div>
    );
  }}
export default NewComp;
```

App.js

```
import './App.css';
import React from 'react';
import NewComp from './NewComp';

class App extends React.Component {
 styles = {
  fontstyle:"bold",
  color : "teal"
 };
 render(){
 return (
  <div className='App'>
   <h2 style={this.styles}>Welcome</h2>
   <NewComp />
  </div>
 );
}}

export default App;
```

Q.10) React – Form event handling

Code :

Live_data.js

```
import React,{useState} from 'react';
export default function Form_Validation(){
   const [data,setData] = useState(null)
   function getData(val){
      setData(val.target.value)
   }
    return(
      <>
         <span>Enter Text : </span><input type='text' onChange={getData}/>
         <br/>
         <span>Text is : {data}</span>
      </>
    )
}
```

App.js

```
import logo from './logo.svg';
import './App.css';
import Live from './live_data.js'
function App() {
  return (
    <Live>
    </Live>
  );
}

export default App;
```

Q.11) Develop an Asynchronous and Synchronous back end file reading node.js program.

Code :

App.js

```
var fs = require("fs");
fs.readFile('data.txt', function (err, data) {
   if(err){
      return res.send(err);
   }
   console.log('File Readed Async : ',data.toString());
});

data = fs.readFileSync('data.txt')
console.log('File Readed Sync : ',data.toString())
```

Q.12) Node.js – http module

Code :

App.js

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

http.createServer((req,res) => {
   res.writeHead(200,{'content-Type':'application/json'});
   res.write('Hello World !')
   res.end()
}).listen(3000)
```

Q.13) Develop Client-Server communication Networking NodeJS Program.

Code :

Client.js

```
const net = require('net');
const client = net.connect({port: 50448}, () => {//use same port of server
  console.log('connected to server!');
  client.write('world!\r\n');
});
client.on('data', (data) => {
  console.log(data.toString());
  client.end();
});
client.on('end', () => {
  console.log('disconnected from server');
});
```

Server.js

```
const net = require('net');
var server = net.createServer((socket) => {
  socket.end('goodbye\n');
}).on('error', (err) => {
  // handle errors here
  throw err;
});
server.listen(() => {
  address = server.address();
  console.log('opened server on %j', address);
});
```

Q.14) Calculator that does addition and subtraction using node.js

Code :

App.js

```
const process = require('process');
a = process.argv[2]
operator = process.argv[3]
b = process.argv[4]

if(operator == '+'){
   console.log(a+' + '+b+' = ',parseInt(a)+parseInt(b));
}
else if(operator == '-'){
   console.log(a+' - '+b+' = ',parseInt(a)-parseInt(b));
}
```

Input : node app.js 10 + 20

Q.15) Node.js Event Emitter Program-

Code :

```
var events = require('events');
var eventEmitter = new events.EventEmitter();
var connectHandler = function connected() {
   console.log('connection succesful.');
   eventEmitter.emit('data_received');
}
eventEmitter.on('connection', connectHandler);
eventEmitter.on('data_received', function(){
   console.log('data received succesfully.');
});
eventEmitter.emit('connection');
console.log("Program Ended.");
```