1. Write program in Typescripts to Convert the given numbers(Decimal,Hexa-decimal,Otal and binary numbers)

```
let first:number=123;

let second:number=0x37CF;

let third:number=0o25;

let fourth:number=0b1111;

console.log(first);

console.log(second);

console.log(third);

console.log(fourth);
```

output:

Signature of Student   Signature of Faculty

2. Write program in Typescripts using number methods.

```
let mynum:number=12345678;

let mynum1:number=10.87654;

let mynum2:number=10667.987;

mynum2.toLocaleString();

mynum1.toFixed();

mynum1.toFixed(1);

mynum1.toPrecision();

mynum1.toPrecision(1);

mynum.toExponential();

mynum.toExponential(1);

console.log("the input num is:",+mynum);

console.log(mynum.toExponential());

console.log(mynum.toExponential(1));

console.log("the input number is :",+mynum1);

console.log(mynum1.toFixed());

console.log(mynum1.toFixed(1));

console.log("the input number is",+mynum2);

console.log(mynum2.toLocaleString());

console.log("my number is input is" ,+mynum1);

console.log(mynum1.toPrecision());

console.log(mynum1.toPrecision(1));
```

output:

Signature of Student Signature of Faculty

3. Write program in Typescripts to find the sum of series of a given numbers.

```
  function addNumbers() {
  var nums = [];
  for  (var  n = 0;  n < arguments.length;  n++)  {
    nums[n - 0] = arguments[n];
  }
  var i;
  var sum = 0;
  for (i = 0;  i < nums.length;  i++)  {
    sum = sum + nums[i];
  }
  console.log("sum of the numbers", sum);
}
addNumbers(1, 2, 3,4);
addNumbers(10, 30, 20, 60, 35);
```

output:

Signature of Student   Signature of Faculty

4. Write program in Typescripts to find the Factorial of a given numbers.

```
function factorial(number) {
          if (number <= 0) {
    return 1;
    } else {
       return (number * factorial(number - 1));
     }
  };
console.log(factorial(6));
```

output:

Signature of Student   Signature of Faculty

5. Write program in Typescripts using String methods.

```
let str1:string="Hello hmspt";

let str2:string="TUMKUR";

str1.charAt(0);

str1.charAt(5);

str1.conCat(str2);

str1.toUpperCase();

str2.toLowerCase();

str2.indexOf(0);

str2.indexOf(4);

console.log("the input string is:",+str1);

console.log(str1.charAt(0));

console.log(str1.charAt(0));

console.log(str1.conCat(str2));

console.log(str1.toUpperCase());

console.log(str2.toLowerCase());

console.log(str2.indexOf(0));

console.log(str2.indexOf(4));
```

output:

Signature of Student   Signature of Faculty

6. Write program in Typescripts In the array-element is adding (push) and deleting (Pop) elements and sorting using array method.

```
let fruits:array<string>;

fruits=['apple','orange','banana'];

console.log(fruits);

fruits.push('papaya');

fruits.push('mango');

console.log(fruits);

fruits.pop();

console.log(fruits);

fruits.sort();

console.log(fruits);
```

output:

Signature of Student   Signature of Faculty

7. Write program in Typescripts In the array-element is adding (push) and deleting (Pop) elements and sorting using tuple method.

```
var employee:[number,string[] ];

employee=[[1,"steve"],[2,"bill"],[3,"jeff"]];

console.log(employee);

employee.push([5,"gate"]);

employee.push([4,"door"]);

console.log(employee);

employee.pop();

console.log(employee);

employee.sort();

console.log(employee);
```

output:

Signature of Student   Signature of Faculty

1. a. ReactJS installation and setup.

b. Display a message "Hello world" in Visual Studio Code ,Online editor Codepen.io and Html format.

a. ReactJS installation and setup.

(1).NPM will be installed along with Nodejs. Node.js can be downloaded and installed from the official NodeJs website.

https://nodejs.org

Once the Installation of Node is complete. Open Node.Js Command Prompt and we can check the version as well.

```
C:/User>node  -v
```

(2). Install Create-React-App Tool

```
npm install -g create-react-app
```

Creating a new react project

I want to create the project or application in D:\React_Programs. I will create this folder and let our command prompt point to it by using the change directory command.

```
create-react-app test-project
```

Running the React Application

```
cd test-project
npm start
```

The Project we have created and run it locally on our system using npm start. Launch the browser and visit http://localhost:3000.

(3)Install Visual Studio Code

Download and install Visual Studio Code from the following URL

https://code.visualstudio.com/download

After the installation, open the Project we have created earlier using the VS Code. The Project has the following 3 folders

- Node_modules

- Public
- src

In Public folder index.html we have one div tag with id as root.

```
<div id="root"></div>
```

(4). React online editors

we want to create react project using CodePen. In the browser, navigate to

https://codepen.io and click on Start Coding.

Create a simple div in HTML section.

```
<div id="root"></div>
```

Followed by writing some JavaScript Code :

```
ReactDOM.render(
              <h1> Hello world</h1>,
          document.getElementById('root')
        );
```

This code will throw an error as we are missing the references to two Javascript files.

Go to Pen Settings section of Js and add,

https://unpkg.com/react/umd/react.development.js

https://unpkg.com/react-dom/umd/react-dom.development.js

React Directly in HTML

React is to write React directly in your HTML files. Start by including three scripts, the first two let us write React code in our JavaScripts, and the third, Babel, allows us to write JSX syntax

```
<html>

  <head>

 <script src="https://unpkg.com/react/umd/react.development.js" crossorigin></script>

<script src="https://unpkg.com/react-dom/umd/react-dom.development.js"
crossorigin></script>

<script src="https://unpkg.com/@babel/standalone/babel.min.js"> </script>

  </head>

  <body>

    <div id="mydiv"></div>

    <script type="text/babel">

      function Hello() {

        return <h1>Hello World!</h1>;

      }

      ReactDOM.render(<Hello />, document.getElementById('mydiv'))

    </script>

  </body>

</html>
```

Output:

Signature of Student   Signature of Faculty

2.Write a ReactJS program using in (1)React Element.(2)React Element using CreateElement Method.

1)

```
  const element = (
      <div>
    <h1>Welcome to React Programming World</h1>
    <h2>Understanding React Rendering…</h2>
  </div>
      );
         ReactDOM.render(element, document.getElementById('root'));
```

Output:

2)

```
  const element=React.createElement("div",{className:"testClass"},
        React.createElement("h1",null,'Welcome to Pragim Technologies'),
                 React.createElement('h2',null,'I am from h2 Tag'));
  ReactDOM.render(element,document.getElementById("root"));
```

Output:

Signature of Student   Signature of Faculty

3.Write a ReactJS program using in Function method /(Component)in React

1)Using One Component. 2) Using two Components.

1)Using One Component:-

```
function Employee(data) {
  return <div><p>Name : {data.name}</p>
    <p>Salary : {data.salary}</p></div>;
    }
   const element = <Employee name="Sara" salary="12345" />;
  ReactDOM.render(element,  document.getElementById('root'));
```

Output:-

2) Using two Components.

```
const Employee=(data)=> {
  return (<div><p>Name : {data.name}</p>
    <p>Salary : {data.salary}</p>
  <Department dept={data.dept} head={data.head}/>
  </div>);
  }
```

```
 const Department=(deptInfo)=>{
  return <div><p>Dept Name is : <b>{deptInfo.dept}</b></p>
    <p>Dept Head is : <b>{deptInfo.head}</b></p>
    </div>;
}
const element = <Employee name="Sara" salary="12345" dept="Test"
                 head="Head" />;
ReactDOM.render(element,  document.getElementById('root'));
```

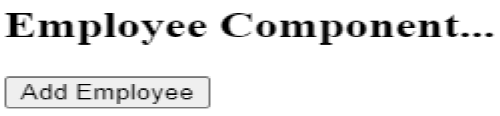
Output:

Signature of Student   Signature of Faculty

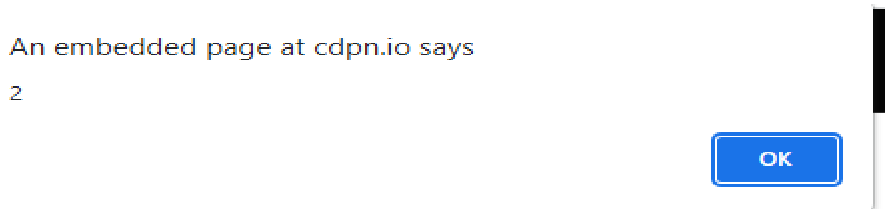
4. (1).Write a ReactJS program to how many times the button is being Clicked. Lets create a counter variable and initialize with zero. In AddEmployee function, using state component.

```
class Employee extends React.Component {
 count=0;
   addEmployee = () => {
       this.count=this.count+1;
       alert(this.count);
       alert('Clicked on addEmployee Method');
   }
render() {
     return <div>
         <h2>Employee Component...</h2>
         <button onClick={this.addEmployee}>Add Employee</button>
     </div>
   }
  }
  const element1=<Employee></Employee>
  ReactDOM.render(element1,document.getElementById("root"));
```

Output:



Employee Component...



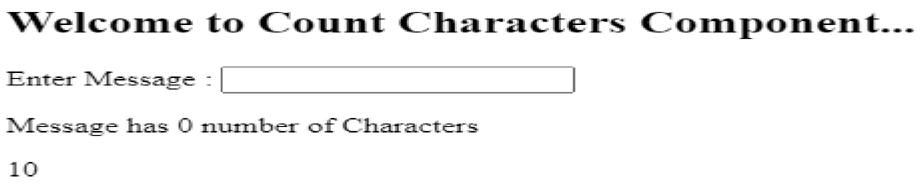
An embedded page at cdpn.io says

2

(2).Write a ReactJS program to count the Characters using State in ClassComponent in React.

```
class Employee extends React.Component {

   state={count:0};

   addEmployee = () => {

      this.setState({counter:this.state.counter+1});

   }

   render() {

     return <div>

        <h2>Employee Component...</h2>

        <button onClick={this.addEmployee}>Add Employee</button>

        <p>   <label>Add Employee Button is Clicked :
<b>{this.state.count}</b></label></p>

     </div>

   }

 }

class CountCharacters extends React.Component{

   constructor(props){

      super(props);

      this.state={

         message:'',

         counter:10

      };

   }

   onMessageChange(text){

      this.setState({

         message:'Message has '+text.length+' number of Characters'

      });
   }
   render(){
       return <div>
           <h2>Welcome to Count Characters Component...</h2>
           <p>   <label>Enter Message : <input type="text"

onChange={e=>this.onMessageChange(e.target.value)}></input></label>
           </p>
           <p>   <label>{this.state.message}</label>     </p>
           <p>   <label>{this.state.counter}</label>     </p>
       </div>
   }
}
const element=<CountCharacters></CountCharacters>
ReactDOM.render(element,document.getElementById("root"));
```

Output:



```
Welcome to Count Characters Component...
Enter Message :
Message has 0 number of Characters
10
```

Signature of Student   Signature of Faculty

5. Write a ReactJS program to Interaction between Components in React and how to pass the data from Parent to Child and child to parent components

(1)In this communication data passing from parents to child's.

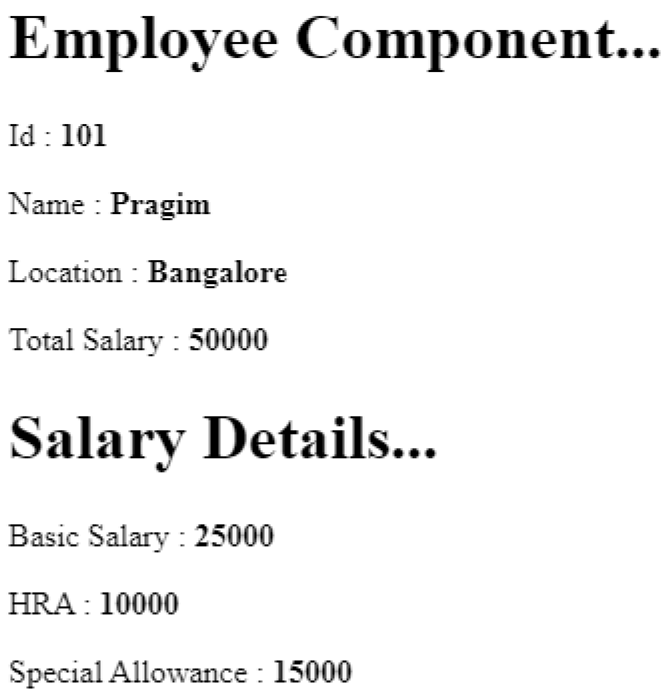
(2).In this communication data passing from child to parent.

(1)In this communication data passing from parents to child's.

Employee is the Parent and Salary Component is the Child. Parent Component is passing the Data to Child Components through properties.

```
class Employee extends React.Component{
  constructor(props){
    super(props);
  }
  render(){
    return <div>
      <h1>Employee Component...</h1>
    <p>    <label>Id : <b>{this.props.Id}</b></label>     </p>
      <p>    <label>Name : <b>{this.props.Name}</b></label>     </p>
      <p>    <label>Location : <b>{this.props.Location}</b></label>     </p>
      <p>   <label>Total Salary : <b>{this.props.Salary}</b></label>     </p>
<Salary BasicSalary={this.props.BasicSalary} HRA={this.props.HRA}
 SpecialAllowance={this.props.SpecialAllowance}></Salary>
      </div>
  }
}
class Salary extends React.Component{
  constructor(props){
    super(props);
  }
 render(){
   return <div>
     <h1>Salary Details...</h1>
<p> <label>Basic Salary : <b>{this.props.BasicSalary}</b></label>   </p>
<p>  <label>HRA : <b>{this.props.HRA}</b></label>  </p>
   <p>   <label>Special Allowance : <b>{this.props.SpecialAllowance}</b></label>
   </p>
     </div>
 }
}
const element=<Employee Id="101" Name="Pragim" Location="Bangalore"
Salary="50000"
BasicSalary="25000" HRA="10000" SpecialAllowance="15000"></Employee>
 ReactDOM.render(element,document.getElementById("root"));
```

Output:

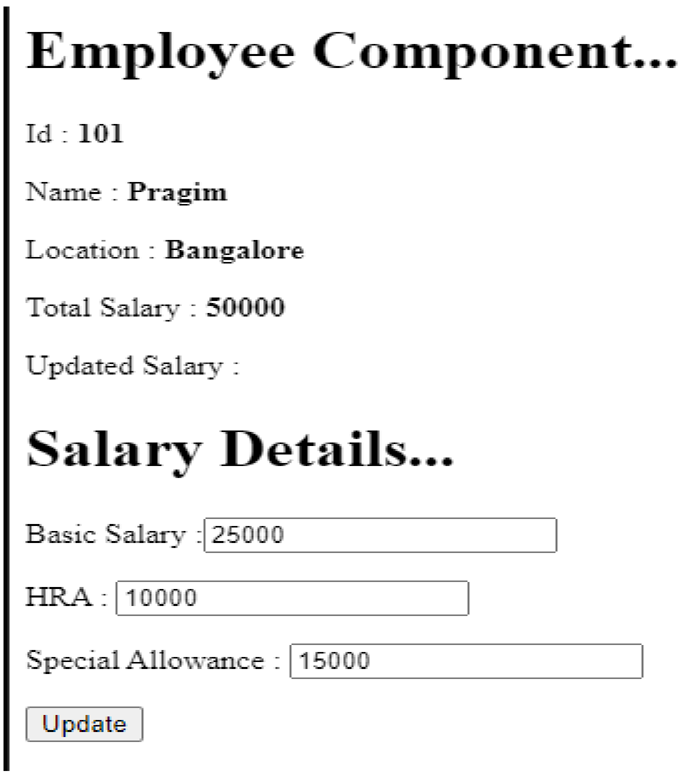


```
Employee Component...

Id : 101

Name : Pragim

Location : Bangalore

Total Salary : 50000

Salary Details...

Basic Salary : 25000

HRA : 10000

Special Allowance : 15000
```

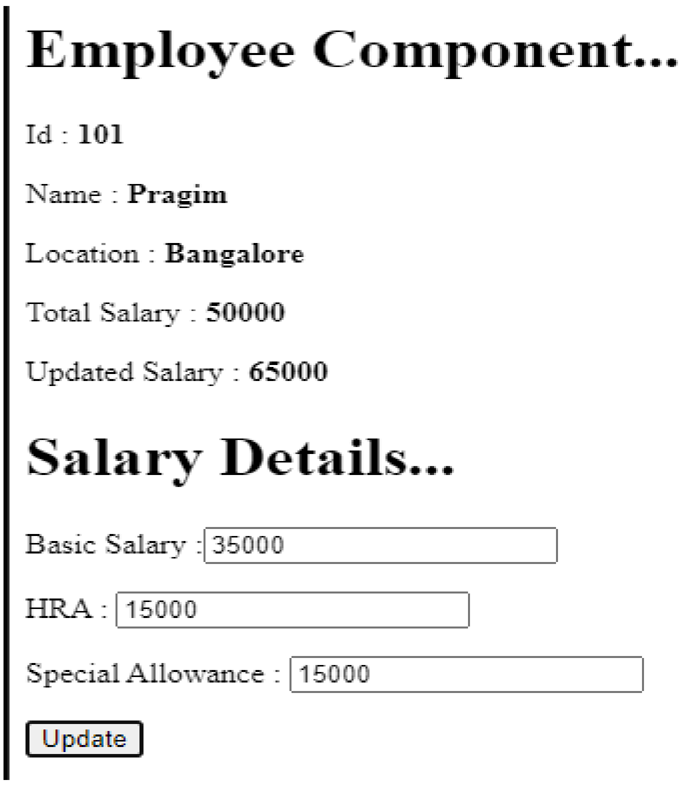
(2).In this communication data passing from child to parent.

```
class Employee extends React.Component{
  constructor(props){
    super(props);
    this.state={updatedSalary:null};
  }
  getUpdatedSalary = (salary) => {
this.setState({updatedSalary:salary});
}
  render(){
    return <div>
  <h1>Employee Component...</h1>
 <p> <label>Id : <b>{this.props.Id}</b></label>  </p>
<p> <label>Name : <b>{this.props.Name}</b></label>   </p>
 <p> <label>Location : <b>{this.props.Location}</b></label>     </p>
  <p> <label>Total Salary : <b>{this.props.Salary}</b></label>     </p>
 <p> <label>Updated Salary : <b>{this.state.updatedSalary}</b></label>
</p>
<Salary BasicSalary={this.props.BasicSalary} HRA={this.props.HRA}
SpecialAllowance={this.props.SpecialAllowance} onSalaryChanged={this.
getUpdatedSalary }></Salary>
        </div>
  }
}
class Salary extends React.Component{
 constructor(props){
    super(props);
    this.state={
      basic:this.props.BasicSalary,

      hra:this.props.HRA,

      sa:this.props.SpecialAllowance

    };

 }

 updateSalary=()=>{

let salary=parseInt(this.refs.BasicSalary.value)+parseInt(this.refs.HRA.value)+
parseInt(this.refs.SpecialAllowance.value);

this.props.onSalaryChanged(salary);

 }

  render(){

    return <div>

      <h1>Salary Details...</h1>

   <p>

 <label>Basic Salary :<input type="text" ref="basic"  defaultValue={this.state.basic}
ref="BasicSalary"/></label>  </p>

<p> <label>HRA : <input type="text"  ref="hra" defaultValue={this.state.hra}
ref="HRA"/></label>     </p>

 <p> <label>Special Allowance : <input type="text" ref="sa"
defaultValue={this.state.sa} ref="SpecialAllowance"/></label>     </p>

<button onClick={this.updateSalary}>Update</button>

         </div>

  }

}

const element=<Employee Id="101" Name="Pragim" Location="Bangalore"
Salary="50000"  BasicSalary="25000" HRA="10000"
SpecialAllowance="15000"></Employee>

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

Output:



```
Employee Component...

Id : 101

Name : Pragim

Location : Bangalore

Total Salary : 50000

Updated Salary :

Salary Details...

Basic Salary : 25000

HRA : 10000

Special Allowance : 15000

Update
```



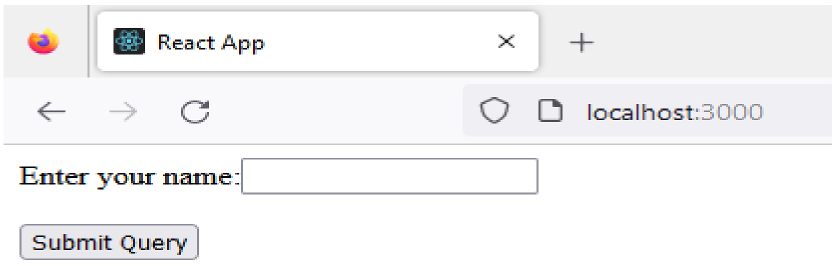
```
Employee Component...

Id : 101

Name : Pragim

Location : Bangalore

Total Salary : 50000

Updated Salary : 65000

Salary Details...

Basic Salary : 35000

HRA : 15000

Special Allowance : 15000

Update
```

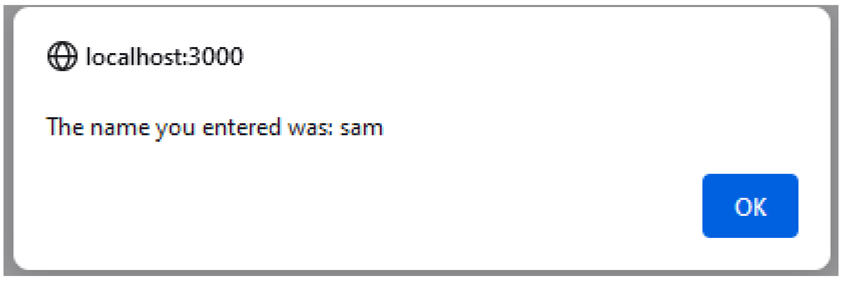
Signature of Student Signature of Faculty

6.Write a React-program using Submitting forms :

```
import { useState } from 'react';
import ReactDOM from 'react-dom/client';
function MyForm() {
  const [name, setName] = useState("");
  const handleSubmit = (event) => {
        event.preventDefault();
          alert(`The name you entered was: ${name}`)
  }
  return ( <div>
    <form onSubmit={handleSubmit}>
      <label>Enter your name: <input type="text" value={name}
         onChange={(e) => setName(e.target.value)}/>
                  </label><br></br><br></br>
      <input type="submit" />
    </form></div>  )
}
  const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<MyForm/>);
```

Output





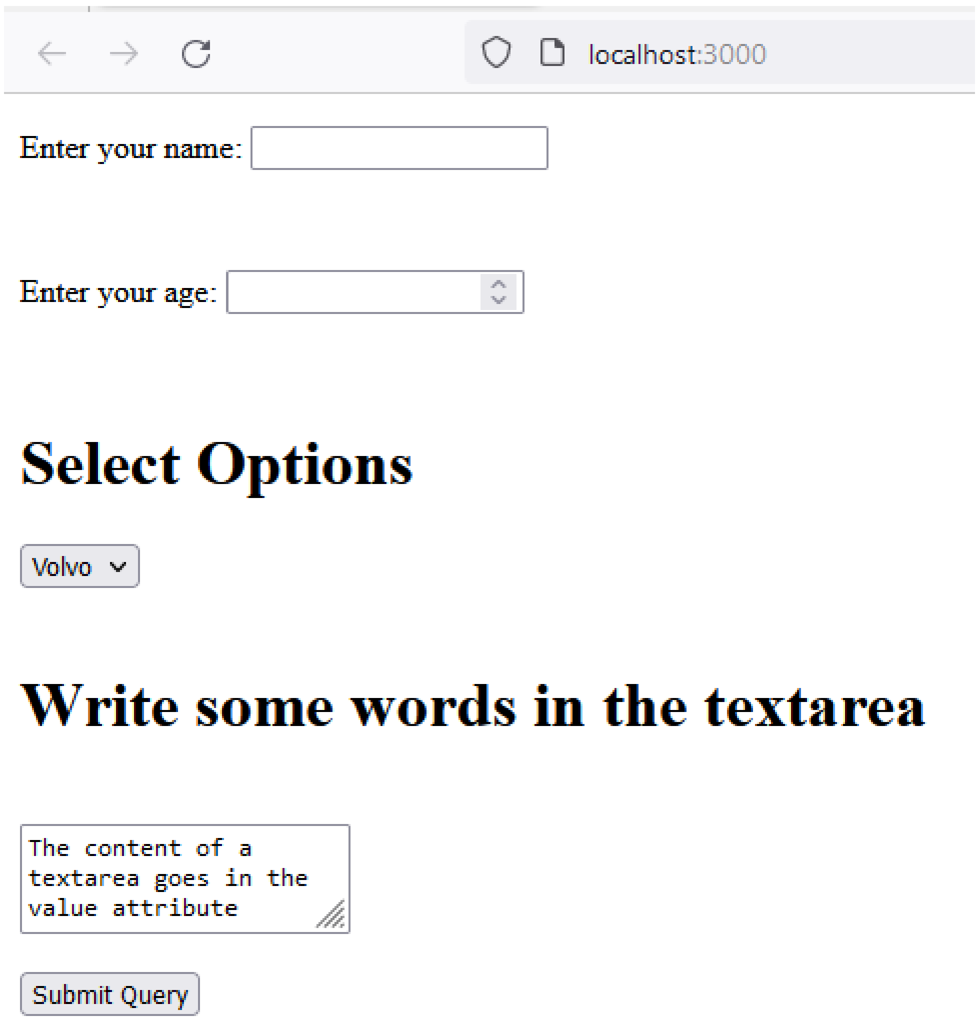
localhost:3000

The name you entered was: sam

7.Write a React-program using Multiple input fields (textarea, Select) Submitting forms.

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

import ReactDOM from 'react-dom/client';

function MyForm() {

  const [inputs, setInputs] = useState({});

  const [textarea, setTextarea] = useState(

                                  "The content of a textarea goes in the value attribute"  );

 const [myCar, setMyCar] = useState("Volvo");

const handleChange = (event) => {

    const name = event.target.name;

    const value = event.target.value;

    setInputs(values => ({...values, [name]: value}))

    setTextarea(event.target.value)

    setMyCar(event.target.value)

    }

  const handleSubmit = (event) => {

    event.preventDefault();

    alert(inputs);

  }

  return ( <div>

    <form onSubmit={handleSubmit}>
<p>

<label>Enter your name: <input type="text"  name="username"
value={inputs.username || ""}   onChange={handleChange}/> </label></p><br></br>

<p><label>Enter your age: <input  type="number" name="age"

              value={inputs.age || ""}    onChange={handleChange} />
</label></p><br></br>

<h1>Select Options</h1>

<select value={myCar} onChange={handleChange}>

        <option value="Ford">Ford</option>

        <option value="Volvo">Volvo</option>

        <option value="Fiat">Fiat</option>

      </select>

      <br></br> <br></br>

      <h1>Write some words in the textarea </h1><br></br>

  <textarea value={textarea} onChange={handleChange} /><br></br><br></br>

  <input type="submit" />

    </form></div>

  )

}

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<MyForm />);
```

Output:



localhost:3000

Enter your name:

Enter your age:

Select Options

Volvo

Write some words in the textarea

```
The content of a
textarea goes in the
value attribute
```

Submit Query

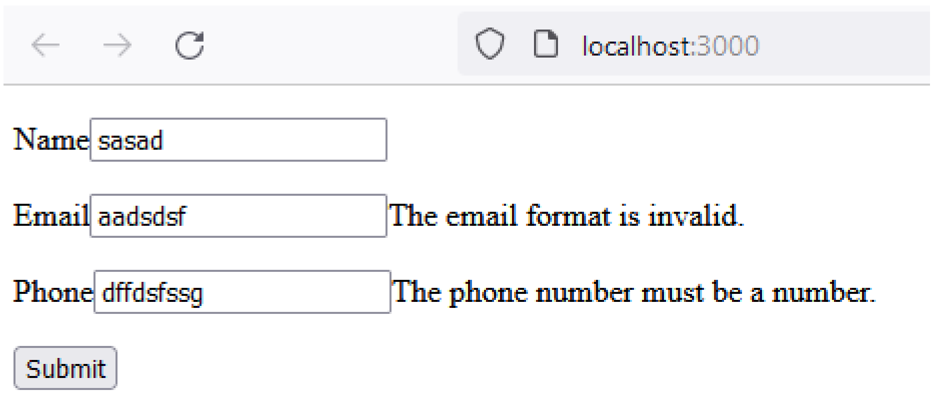
8.Write a React-program using Form Input Validation using

1. **Class component**
2. **Functional Component**

1.Class component

```
import React from 'react';

import { render } from "react-dom";

import ReactDOM from 'react-dom';

import { useForm } from "react-hook-form";

import ReactFormInputValidation from "react-form-input-validation";

class ValidationForm extends React.Component {

  constructor(props) {

    super(props);

    this.state = {

      fields: {  name: "",   email: "",   phone_number: ""

      },

      errors: {}

    };

    this.form = new ReactFormInputValidation(this);

    this.form.useRules({  name: "required",  email: "required|email",

        phone_number: "required|numeric|digits_between:10,12",

    });

    this.form.onformsubmit = (fields) => {

    }

  }

  render() {

    return (<React.Fragment>
        <form onSubmit={this.form.handleSubmit}>

            <p>    <label>   Name    <input    type="text"     name="name"
onBlur={this.form.handleBlurEvent}    onChange={this.form.handleChangeEvent }

                  value={this.state.fields.name}  />  </label>

   <label className="error">

                {this.state.errors.name ? this.state.errors.name : ""}   </label>

            </p>

           <p>   <label>     Email    <input   type="email"    name="email"

       onBlur={this.form.handleBlurEvent}   onChange={this.form.handleChangeEvent}

                  value={this.state.fields.email}  />  </label>

              <label className="error">

                {this.state.errors.email ? this.state.errors.email : ""}  </label>

            </p>

            <p>   <label>   Phone  <input    type="tel"  name="phone_number"

                  onBlur={this.form.handleBlurEvent}

                  onChange={this.form.handleChangeEvent}

                  value={this.state.fields.phone_number}  />  </label>

              <label className="error">

                {this.state.errors.phone_number ? this.state.errors.phone_number : ""}

              </label>   </p>

            <p>   <button type="submit">Submit</button>    </p>

        </form>

    </React.Fragment>)

  }

}

const element=<ValidationForm></ValidationForm>

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

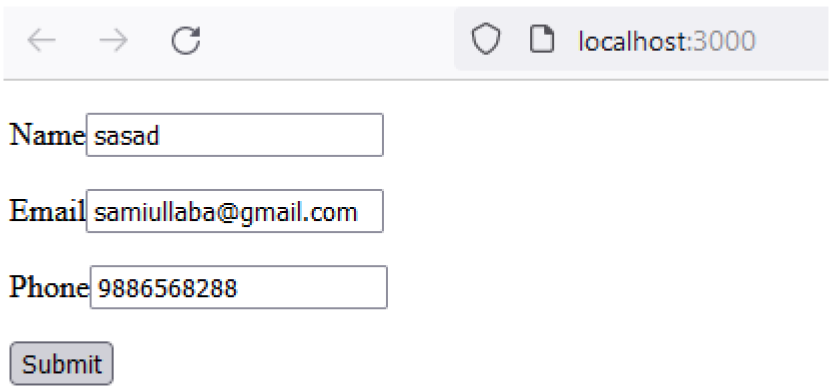
Output:-


localhost:3000

Name sasad

Email aadsdsf The email format is invalid.

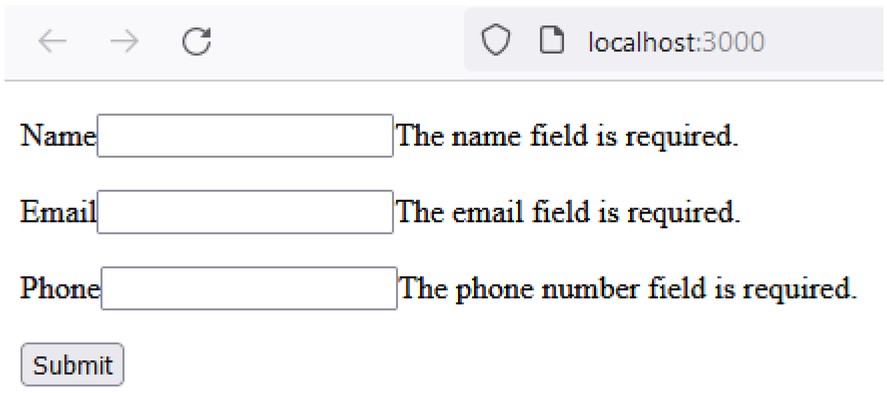
Phone dffdsfssg The phone number must be a number.


localhost:3000

Name sasad

Email samiullaba@gmail.com

Phone 9886568288


localhost:3000

Name The name field is required.

Email The email field is required.

Phone The phone number field is required.

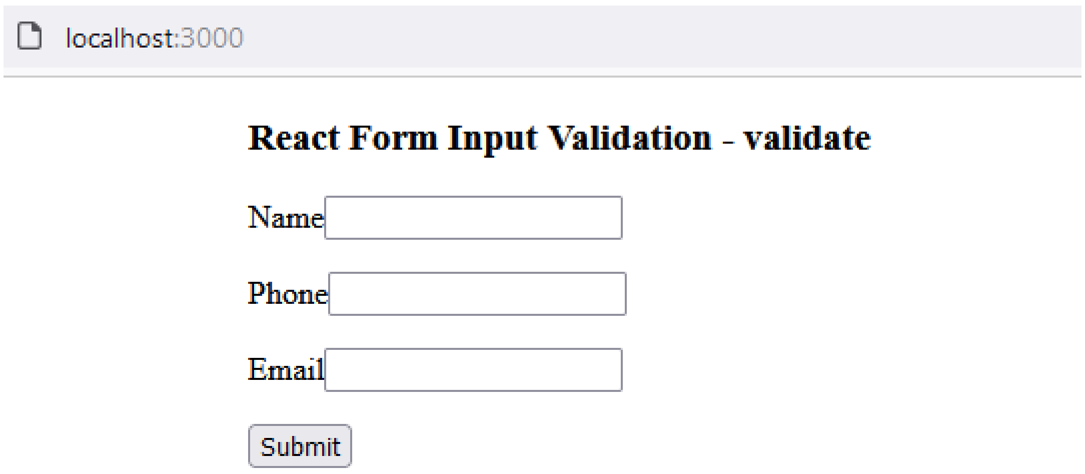
2. Functional Component

```
import React from 'react';
import { render } from "react-dom";
import ReactDOM from 'react-dom';
import { useForm } from "react-hook-form";
import ReactFormInputValidation from "react-form-input-validation";
import { useFormInputValidation } from "react-form-input-validation";
const ValidationForm = () => {
  const [fields, errors, form] = useFormInputValidation({
    customer_name: "",    email_address: "",   phone_number: "",  }, {
    customer_name: "required",  email_address: "required|email",
    phone_number: "required|numeric|digits_between:10,12"
  });

  const onSubmit = async (event) => {
    const isValid = await form.validate(event);
    if (isValid) {
    }
  }
    return <div style={{maxWidth: "600px", margin: "0 auto"}}>
  <h3>React Form Input Validation - validate</h3>
  <form
    className="myForm"
    noValidate    autoComplete="off"    onSubmit={onSubmit}  >
    <p>
      <label>   Name   <input   type="text"   name="customer_name"
```

```
          onBlur={form.handleBlurEvent}    onChange={form.handleChangeEvent}
          value={fields.customer_name}  />
      </label>
      <label className="error">
        {errors.customer_name   ? errors.customer_name   : ""}
      </label>    </p>
   <p>
      <label>    Phone   <input   type="tel"  name="phone_number"
          onBlur={form.handleBlurEvent}    onChange={form.handleChangeEvent}
          value={fields.phone_number}  />
      </label>
      <label className="error">
 {errors.phone_number    ? errors.phone_number    : ""}
      </label>     </p>
    <p>  <label>   Email    <input    type="email"    name="email_address"
          onBlur={form.handleBlurEvent}  onChange={form.handleChangeEvent}
          value={fields.email_address}  />
      </label>
      <label className="error">
        {errors.email_address  ? errors.email_address   : ""}
      </label>     </p>
    <p>
      <button type="submit">Submit</button>    </p>
  </form>
</div>
}
export default ValidationForm;
```

```
const element=<ValidationForm></ValidationForm>

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

Output:-



9. Write a React-program using signup-form

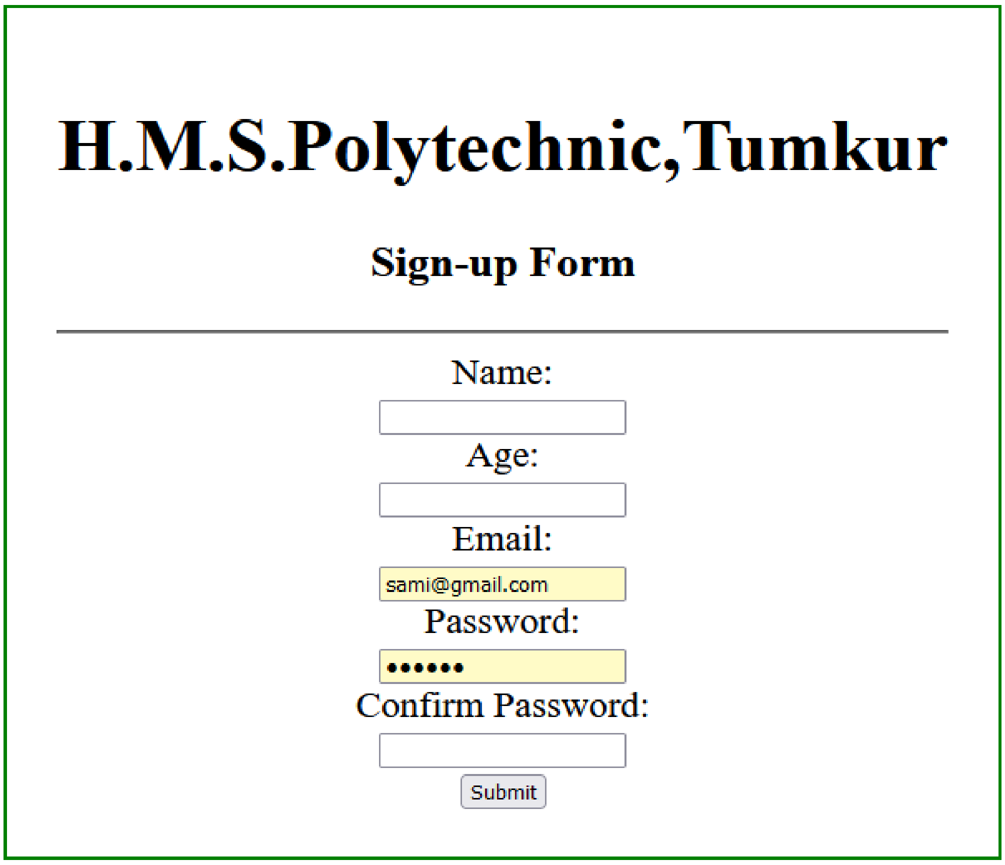
```
import React, {useState} from 'react';
import ReactDOM from 'react-dom';
import './App.css';

function App() {
  const [name , setName] = useState('');
  const [age , setAge] = useState('');
  const [email , setEmail] = useState('');
  const [password , setPassword] = useState('');
  const [confPassword , setConfPassword] = useState('');
  const handleChange =(e)=>{
   setName(e.target.value);
  }
  const handleAgeChange =(e)=>{
   setAge(e.target.value);
  }
  const handleEmailChange =(e)=>{
   setEmail(e.target.value);
  }
  const handlePasswordChange =(e)=>{
   setPassword(e.target.value);
  }
  const handleConfPasswordChange =(e)=>{
   setConfPassword(e.target.value);
  }
  const handleSubmit=(e)=>{
   if(password!=confPassword)
   {
    // if 'password' and 'confirm password'
    // does not match.
    alert("password Not Match");
   }
   else{
    // display alert box with user
```

```
        // 'name' and 'email' details .
        alert('A form was submitted with Name :"' + name +
        '" ,Age :"'+age +'" and Email :"' + email + '"');
      }
      e.preventDefault();
     }
    return (
      <div className="App">
      <header className="App-header">
      <form onSubmit={(e) => {handleSubmit(e)}}>
           <h2> Geeks For Geeks </h2>
      <h3> Sign-up Form </h3>
          <label>   Name:  </label><br/>
   <input type="text" value={name} required onChange={(e)=>
{handleChange(e)}}/><br/>
             <label>  Age:   </label><br/>
   <input type="text" value={age} required onChange={(e)=>
{handleAgeChange(e)}}/><br/>

          <label>  Email:  </label><br/>
  <input type="email" value={email} required onChange={(e)=>
{handleEmailChange(e)}}/><br/>

          <label> Password:   </label><br/>
  <input type="password" value={password} required onChange={(e)  =>
{handlePasswordChange(e)}}/><br/>

      <label> Confirm Password:   </label><br/>
  <input type="password" value={confPassword} required onChange={(e) =>
{handleConfPasswordChange(e)}}/><br/>

  <input type="submit" value="Submit"/>
        </form>
      </header>
      </div>
    );
  }
     export default App;
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<App/>);
```

Seprate file name App.css

```
.App {
  text-align: center;
}
form {
 border:2px solid green;
 padding: 30px;
}
img{
 height: 120px;
 margin-left: 90px;
 margin-bottom: 10px;
 display: block;
 border:1px solid black;
 border-radius: 50%;
}
.App-header {
 background-color: white;
 min-height: 100vh;
 display: flex;
 flex-direction: column;
 align-items: center;
 justify-content: center;
 font-size: calc(10px + 2vmin);
 color: black;
}
```

Output:



H.M.S.Polytechnic,Tumkur

Sign-up Form

Name:

Age:

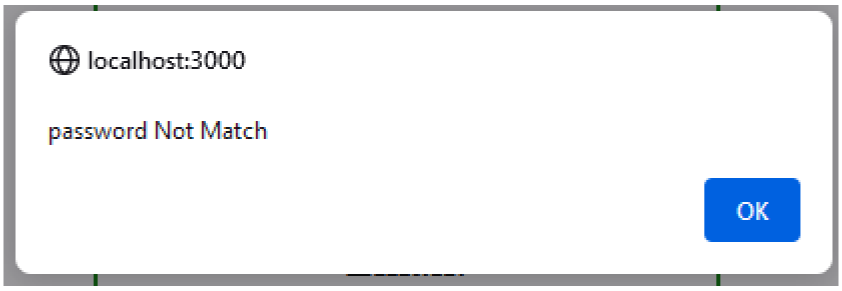
Email:

sami@gmail.com

Password:

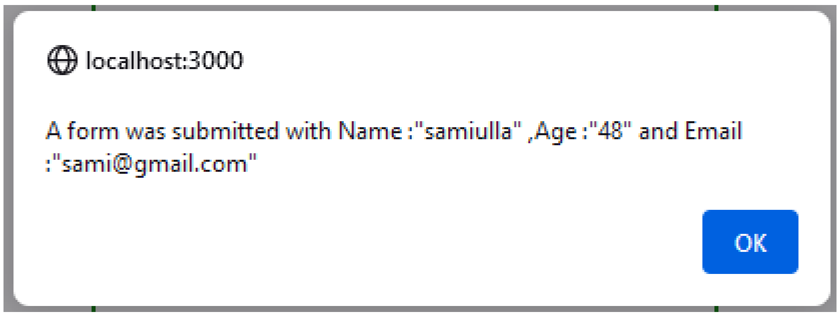
••••••

Confirm Password:



localhost:3000

password Not Match



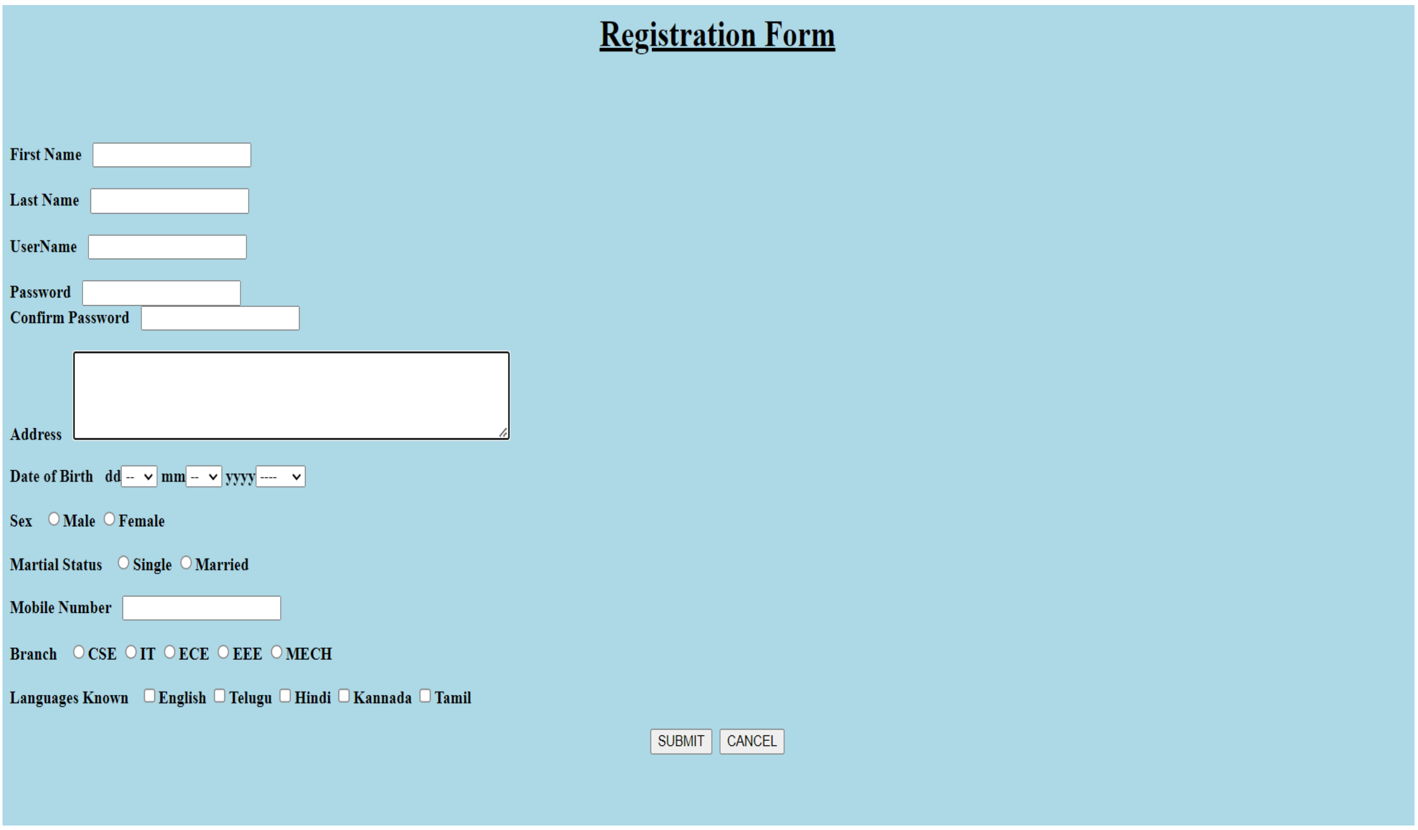
localhost:3000

A form was submitted with Name :"samiulla" ,Age :"48" and Email :"sami@gmail.com"

1. Write a HTML program to develop a static Registration Form.

```
   <html>
   <head>    <title>Registration</title>
   </head>
  <body bgcolor=lightblue>
 <h1 align=center><u>Registration Form</u></h1>
 <br><br><br>
 <div>
 <strong>
 First Name &nbsp <input type=text value=" " name="txt1"><br><br>
 Last Name &nbsp <input type=text value=" " name="txt2"><br><br>
 UserName &nbsp <input type=text value="" name="txt3"><br><br>
 Password &nbsp <input type=password value="" name="pwd1"><br>
 Confirm Password &nbsp <input type=password value="" name="pwd2"><br><br>
 Address &nbsp <textarea rows=3 cols=60></textarea><br><br>
 Date of Birth &nbsp
 dd<select name="sel1">
 <option>--</option>
 <option>01</option>
 <option>02</option>
 <option>03</option>
 <option>04</option>
 <option>05</option>
 <option>27</option>
 <option>28</option>
 <option>29</option>
 <option>30</option>
 <option>31</option>
 </select>
mm<select name="sel2">
 <option>--</option>
 <option>01</option>
 <option>02</option>
 <option>03</option>
 <option>04</option>
 <option>05</option>
 <option>06</option>
 <option>07</option>
 <option>08</option>
 <option>09</option>
 <option>10</option>
 <option>11</option>
 <option>12</option>
 </select>
 yyyy<select name="sel3">
 <option>----</option>
 <option>1987</option>
 <option>1988</option>
 <option>1989</option>
 <option>1990</option>
 <option>1991</option>
 <option>1992</option>
 <option>1993</option>
 <option>1994</option>
 <option>1995</option>
 <option>1996</option>
 <option>1997</option>
 <option>1998</option>
 <option>1999</option>
<option>2000</option>
 <option>2001</option>
 <option>2002</option>
 <option>2003</option>
 <option>2004</option>
 <option>2005</option>
 <option>2006</option>
 <option>2007</option>
 <option>2008</option>
 <option>2009</option>
 <option>2010</option>
 <option>2011</option>
 <option>2012</option>
 <option>2013</option>
 <option>2014</option>
 <option>2015</option>
 <option>2016</option>
 <option>2017</option>
 </select><br><br>
 Sex &nbsp
 <input name="rb1" type="radio" value="radiobutton">Male
 <input name="rb1" type="radio" value="radiobutton">Female
 <br><br>
 Martial Status &nbsp
 <input name="rb2" type="radio" value="radiobutton">Single
 <input name="rb2" type="radio" value="radiobutton">Married
 <br><br>
 Mobile Number &nbsp <input type=text name="txt4"><br><br>
 Branch &nbsp
 <input name="rb3" type="radio" value="radiobutton">CSE
<input name="rb3" type="radio" value="radiobutton">IT
 <input name="rb3" type="radio" value="radiobutton">ECE
 <input name="rb3" type="radio" value="radiobutton">EEE
 <input name="rb3" type="radio" value="radiobutton">MECH
 <br><br>
 Languages Known &nbsp
 <input name="cb1" type="checkbox" value="checkbox">English
 <input name="cb1" type="checkbox" value="checkbox">Telugu
 <input name="cb1" type="checkbox" value="checkbox">Hindi
 <input name="cb1" type="checkbox" value="checkbox">Kannada
 <input name="cb1" type="checkbox" value="checkbox">Tamil
 <br><br>
 <center>
 <input type=submit value="SUBMIT" name="btn1">&nbsp
 <input type=reset value="CANCEL" name="btn1">
 </center>
</strong>
</body>
</html>
```

Output:



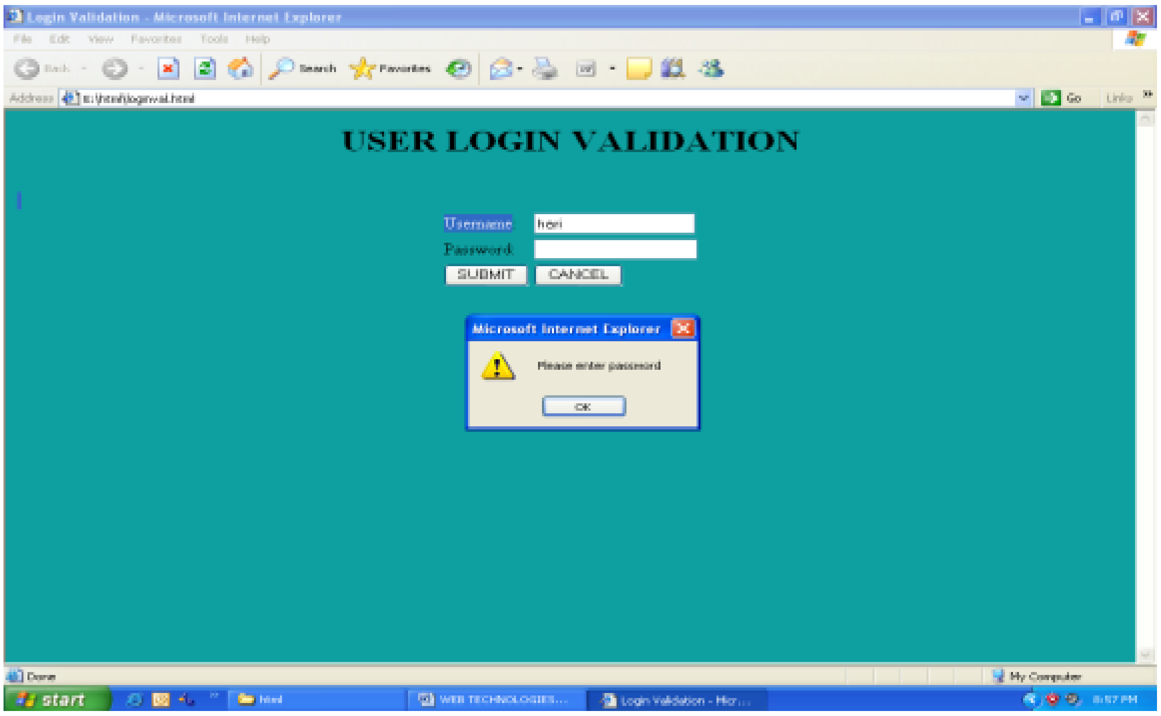
Registration Form

First Name	
Last Name	
UserName	
Password	
Confirm Password	
Address	
Date of Birth	dd -- mm -- yyyy ----
Sex	○ Male ○ Female
Martial Status	○ Single ○ Married
Mobile Number	
Branch	○ CSE ○ IT ○ ECE ○ EEE ○ MECH
Languages Known	☐ English ☐ Telugu ☐ Hindi ☐ Kannada ☐ Tamil

2. Write a javascript program to validate USER LOGIN page.

```
<html>
 <head>
 <title>Login Validation</title>
 <script language="javascript">
 function formValidator()
 {
 var username=document.getElementById('uname');
 var password=document.getElementById('pwd');
 if(isEmpty(username)&&isEmpty(password))
 {
 alert("enter something");
 document.form1.uname.focus();
 }
 if(!isEmpty(username)&&isEmpty(password)&&isAlphabet(username))
 {
 alert("Please enter password");
 document.form1.pwd.focus();
 }
 if(!isEmpty(username)&&!isEmpty(password)&&isAlphabet(username))
 {
 return true;
 }
 else
 {
 if(!isEmpty(username)&&!isEmpty(password)&&!isAlphabet(username))
 {
 alert("Please Enter only alphabets for username");
 document.form1.uname.focus();
 }

}
 return false;
 }
 function isEmpty(elem)
 {
 if(elem.value.length==0)
 {
 return true;
 }
 return false;
 }
 function isAlphabet(elem)
 {
 var alphaExp=/^[a-z A-Z]+$/;
 if(elem.value.match(alphaExp))
 {
 return true;
 }
}
</script>
</head>
<body bgColor=megastar>
<h1 align=center>USER LOGIN VALIDATION</h1>
<br><br>
<form name="form1" onSubmit="return formValidator()">
<center>
<table border=0 colsSpacing=4>
<tr>
<td>Username:</td>
<td><input type=text value="" name="uname"></td>
</tr>
<tr>
<td>Password:</td>
<td><input type=password value="" name="pwd"></td>
</tr>
<tr>
<td><input type=submit value="SUBMIT" name="btn1"></td>
<td><input type=reset value="CANCEL" name="btn2"></td>
</tr>
</table>
</center>
</form>
</body>
</html>
```

Output:


Login Validation - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address e:\html\loginval.html
Go Links
SUBMIT
CANCEL
Microsoft Internet Explorer
OK
Done
My Computer
start

USER LOGIN VALIDATION

Username hari
Password

Please enter password

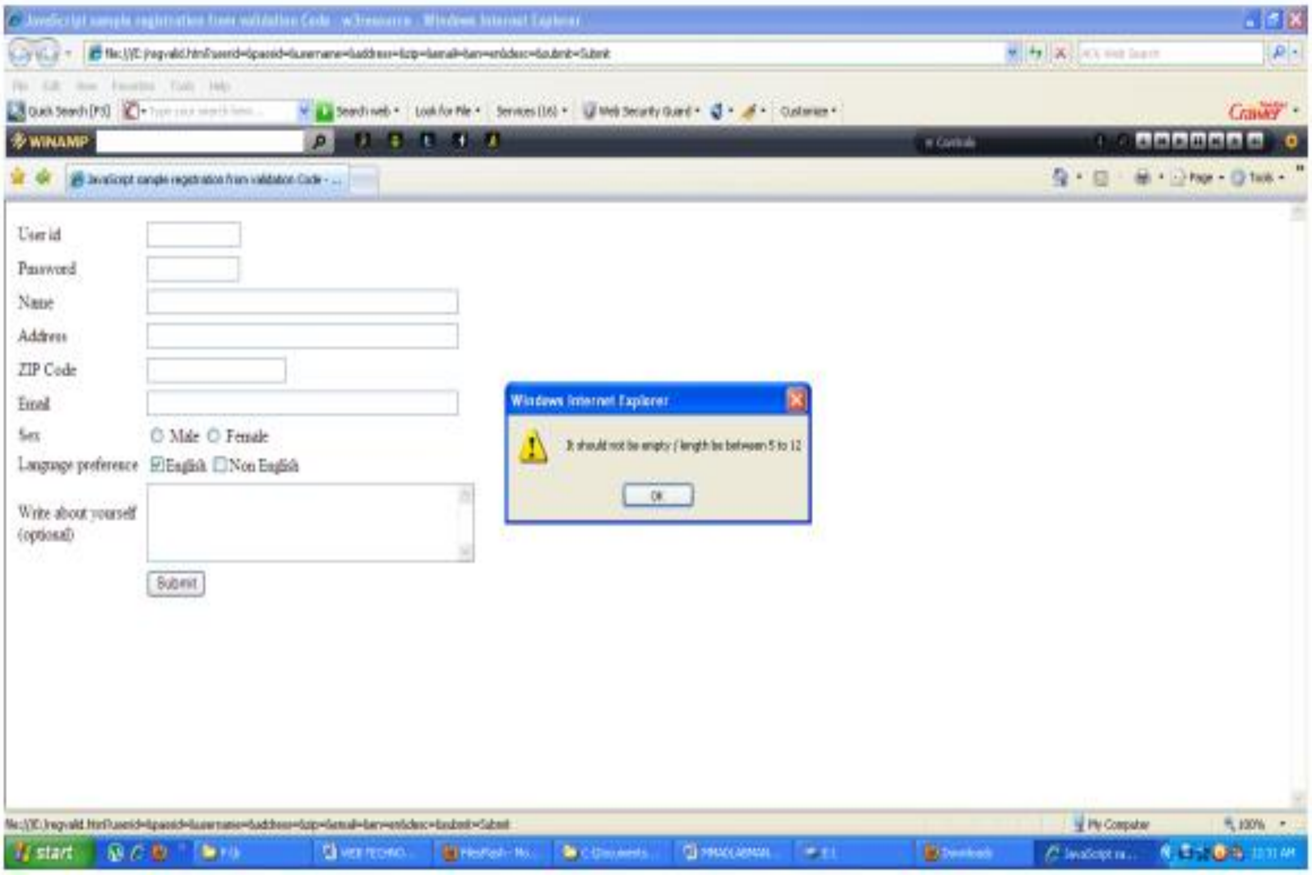
3. Write a javascript program for validating REGISTRATION FORM.

```
<html>
<head>
<title>JavaScript sample registration from validation </title>
<script type='text/javascript'>
function formValidation()
{
var uid = document.form1.userid;
var passid = document.form1.passid;
var uname = document.form1.username;
var uadd = document.form1.address;
var uzip = document.form1.zip;
var uemail = document.form1.email;
var umsex = document.form1.msex;
var ufsex = document.form1.fsex;
if(userid_validation(uid,5,12))
{
if(userid_validation(passid,7,12))
{
if(allLetter(uname))
{
if(alphanumeric(uadd))
{
if(allnumeric(uzip))
{
if(ValidateEmail(uemail))
{
if(validsex(umsex,ufsex))
{
  }    }
}   }     }
}      }
return false;
} function userid_validation(uid,mx,my)
{
var uid_len = uid.value.length;
if (uid_len == 0 || uid_len >= my || uid_len < mx)
{
alert("It should not be empty / length be between "+mx+" to "+my);
uid.focus();
return false;
}
return true;
}
function allLetter(uname)
{
var letters = /^[A-Za-z]+$/;
if(uname.value.match(letters))
{
return true;
}
else
{
alert('Please input alphabet characters only');
uname.focus();
return false;
}
}
function alphanumeric(uadd)
{
var letters = /^[0-9a-zA-Z]+$/;
if(uadd.value.match(letters))
{
return true;
}
else
{
alert('Please input alphanumeric characters only');
uadd.focus();
return false;
}
}
function allnumeric(uzip)
{
var numbers = /^[0-9]+$/;
if(uzip.value.match(numbers))
{
return true;
}
else
{
alert('Please input numeric characters only');
uzip.focus();
return true;
}
}

function ValidateEmail(uemail)
{
var mailformat = /^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/;
if(uemail.value.match(mailformat))
{
return true;
}
else
{
alert("You have entered an invalid email address!");
uemail.focus();
return false;
}
```

```
} function validsex(umsex,ufsex)
{
x=0;
if(umsex.checked)
{
x++;
} if(ufsex.checked)
{
x++;
}
if(x==0)
{
alert('Select Male/Female');
umsex.focus();
return false;
}
else
{
return true;
}
}
</script>
</head>
<body>
<form name='form1' onsubmit='return formValidation()' >
<table width="500" cellpadding="3" style="border-collapse: collapse;">
<tr>
<td>User id </td>
<td><input type="text" name="userid" size="12" /></td>
</tr>
<tr>
<td>Password</td>
<td><input type="password" name="passid" size="12" /></td>
</tr>
<tr>
<td>Name</td>
<td><input type="text" name="username" size="50" /></td>
</tr>
<tr>
<td>Address</td>
<td><input type="text" name="address" size="50" /></td>
</tr>
<tr>
<td>ZIP Code </td>
<td><input type="text" name="zip" /></td>
</tr>
<tr>
<td>Email</td>
<td><input type="text" name="email" size="50" /></td>
</tr>
```

```
<tr>
<td>Sex</td>
<td><input type="radio" name="msex" value="Male" /> Male
<input type="radio" name="fsex" value="Female" /> Female</td>
</tr>
<tr>
<td>Language preference</td>
<td><input type="checkbox" name="en" value="en" checked />English
<input type="checkbox" name="nonen" value="noen" />Non English</td>
</tr>
<tr>
<td>Write about yourself<br>
(optional)</td>
<td><textarea name="desc" rows="4" cols="40"></textarea></td>
</tr>
<tr>
<td>&nbsp;</td>
<td><input type="submit" name="submit" value="Submit" /></td>
<td>&nbsp;</td>
</tr>
</table>
</form>
</body>
</html>
```

Output:



User id
Password
Name
Address
ZIP Code
Email
Sex ○ Male ○ Female
Language preference [✓ English] ☐ Non English
Write about yourself (optional)
Submit

Windows Internet Explorer
It should not be empty / length be between 5 to 12
OK