**Experiment - 1**

**Aim: React application to demonstrate state.**

**Source Code:**

**Index.html:**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="utf-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <title>React App</title>

</head>

<body>

  <div id="root"></div>

</body>

</html>

**index.js**

import React from "react";

import ReactDOM from "react-dom";

import FavoriteColor from "./App";

ReactDOM.render(

  <React.StrictMode>

    <FavoriteColor />

  </React.StrictMode>,

  document.getElementById("root")

);

**App.js**

import { useState } from "react";

import ReactDOM from "react-dom/client";

function FavoriteColor() {

  const [color, setColor] = useState("red");

  return (

    <>

      <h1>My favorite color is {color}!</h1>

      <button

        type="button"

        onClick={() => setColor("blue")}

      >Blue</button>

    </>

  )

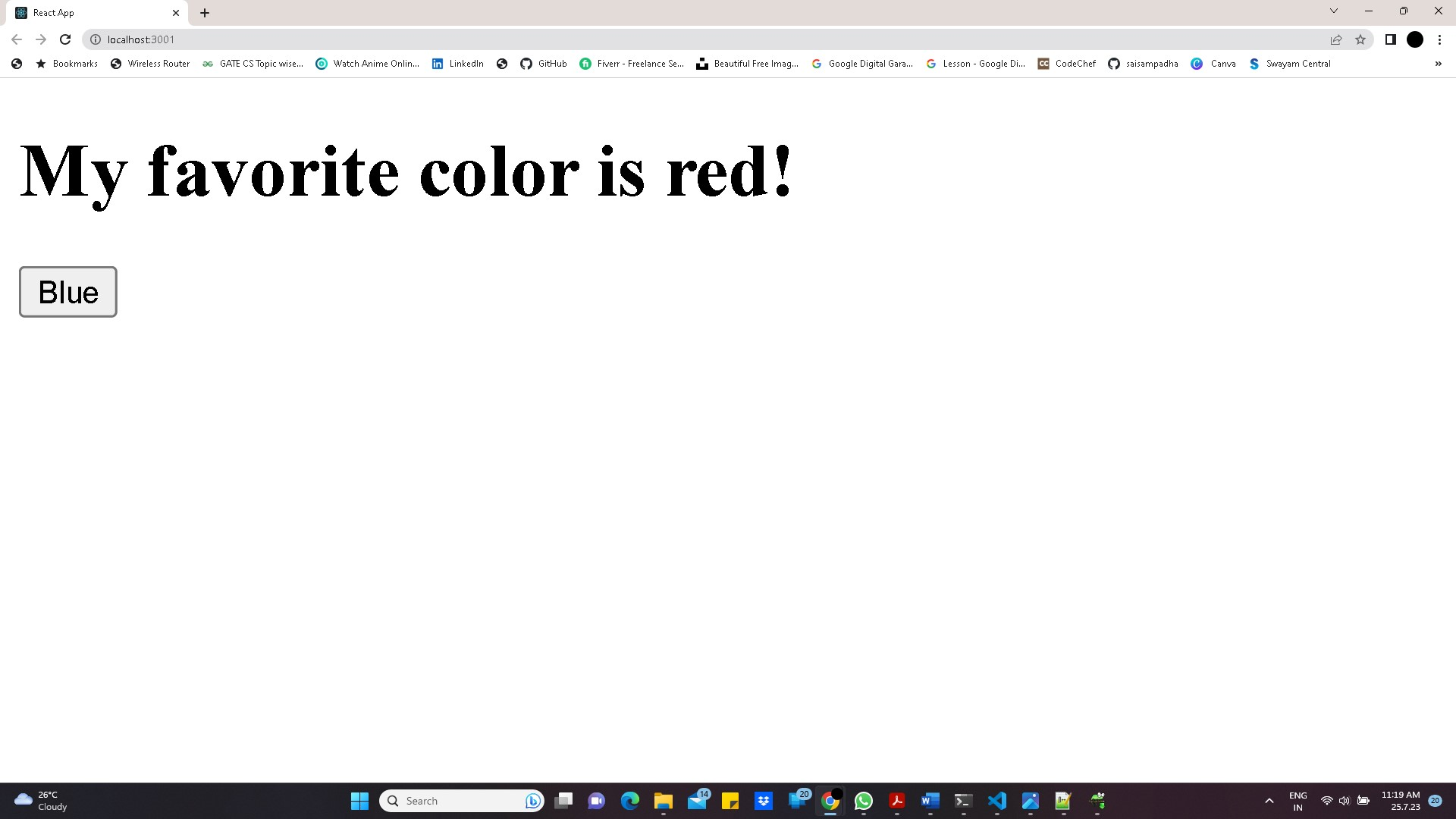
}

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

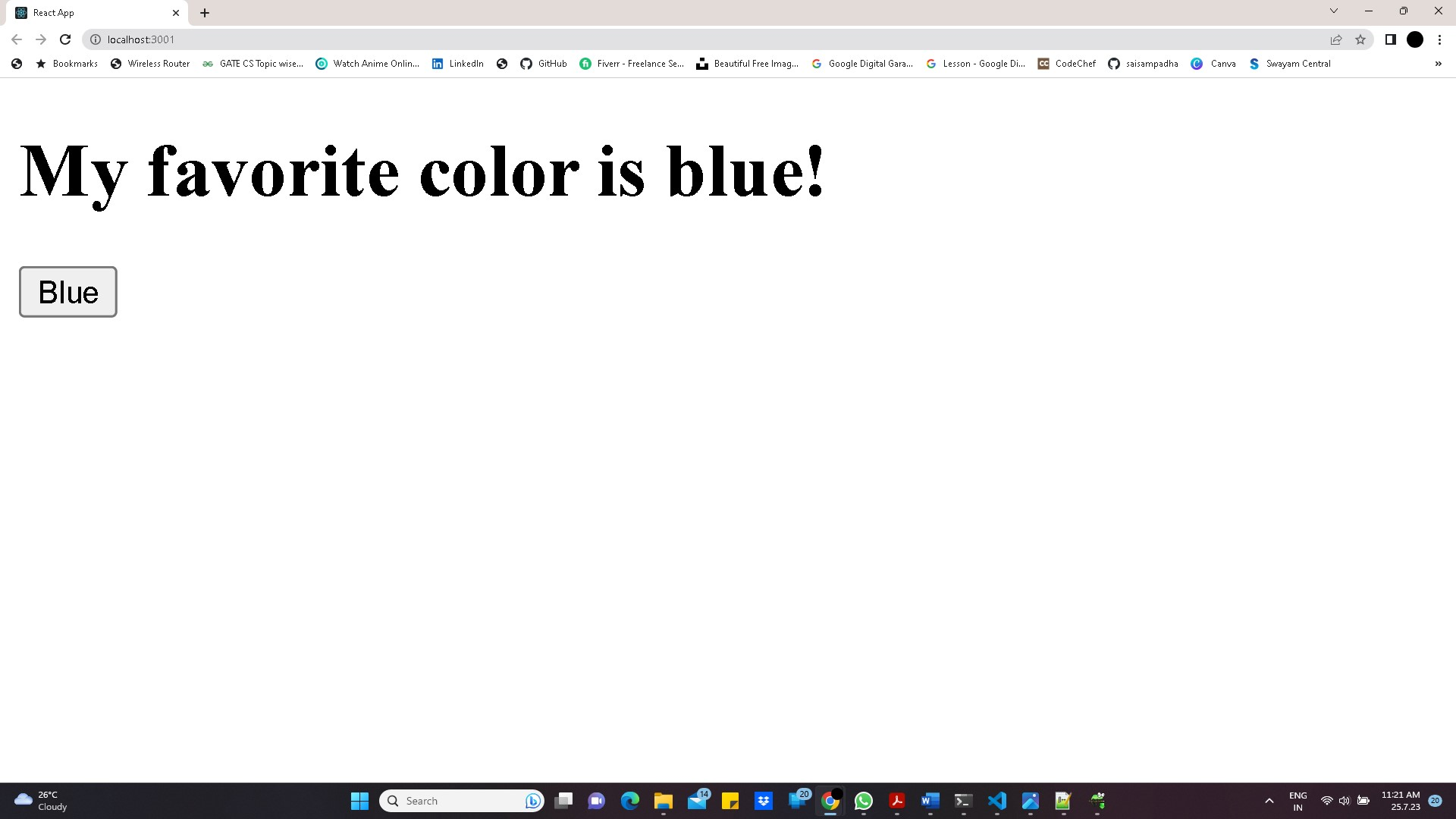
root.render(<FavoriteColor />)

export default FavoriteColor;

**Output:**



**After clicking on the button blue we get**



**Experiment - 2**

**Aim:**. **Develop react application using function & class components.**

**Function Component:**

**Source Code:**

**Index.html:**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="utf-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <title>React App</title>

</head>

<body>

  <div id="root"></div>

</body>

</html>

**Index.js:**

import React from "react";

import ReactDOM from "react-dom";

import HelloWorld from "./App";

ReactDOM.render(

  <React.StrictMode>

    <HelloWorld />

  </React.StrictMode>,

  document.getElementById("root")

);

**App.js:**

import React from 'react';

// Define a function component named 'HelloWorld'

function HelloWorld() {

  return (

    <div>

      <h1>Hello, World!</h1>

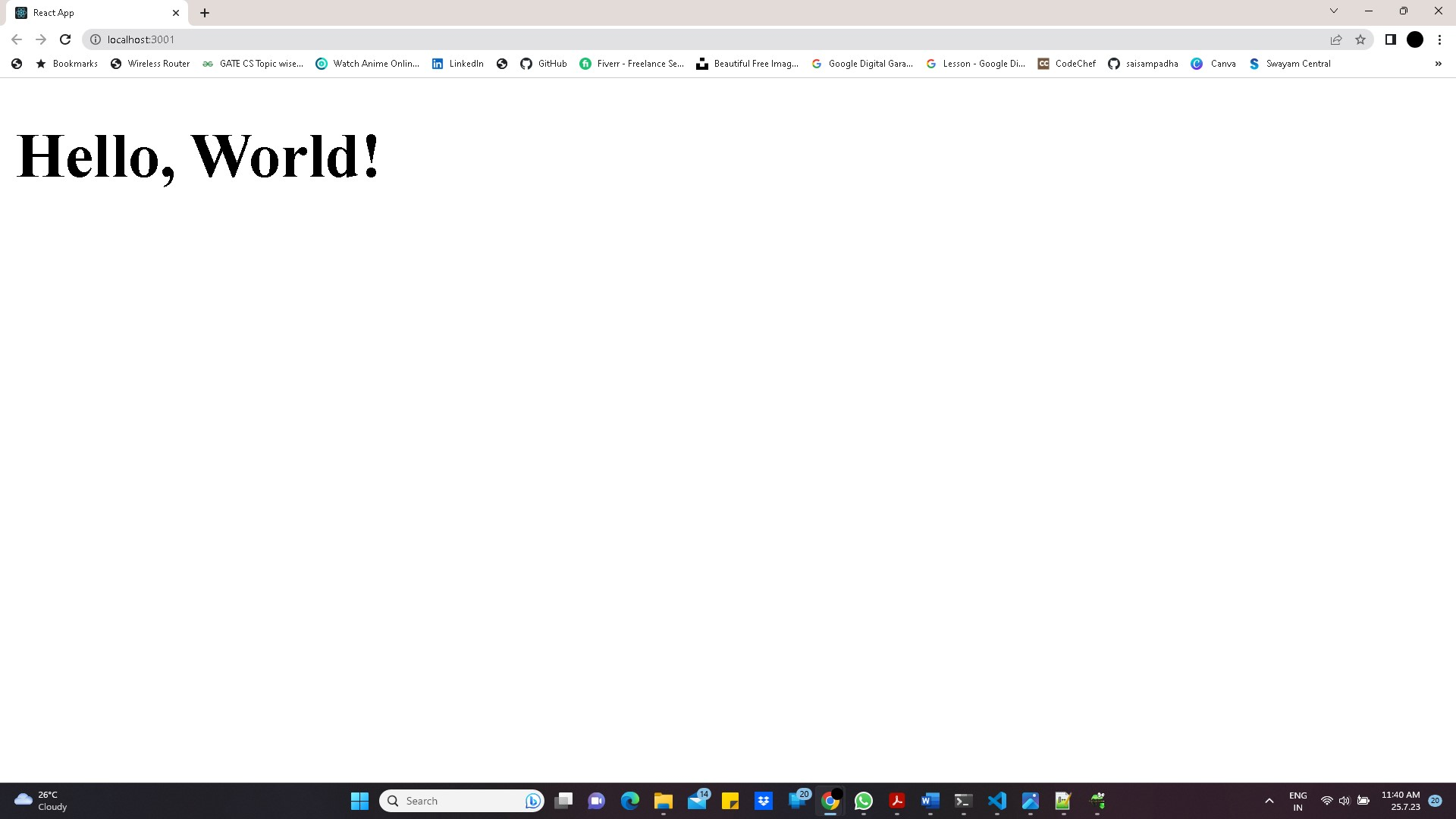
    </div>

  );

}

export default HelloWorld;

**Output:**



**Class Component:**

**Source Code:**

**Index.html:**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="utf-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <title>React App</title>

</head>

<body>

  <div id="root"></div>

</body>

</html>

**Car.js:**

import React from 'react';

class Car extends React.Component {

  constructor() {

    super();

    this.state = { color: "red" };

  }

  render() {

    return <h2>I am a {this.state.color} Car!</h2>;

  }

}

export default Car;

**Index.js:**

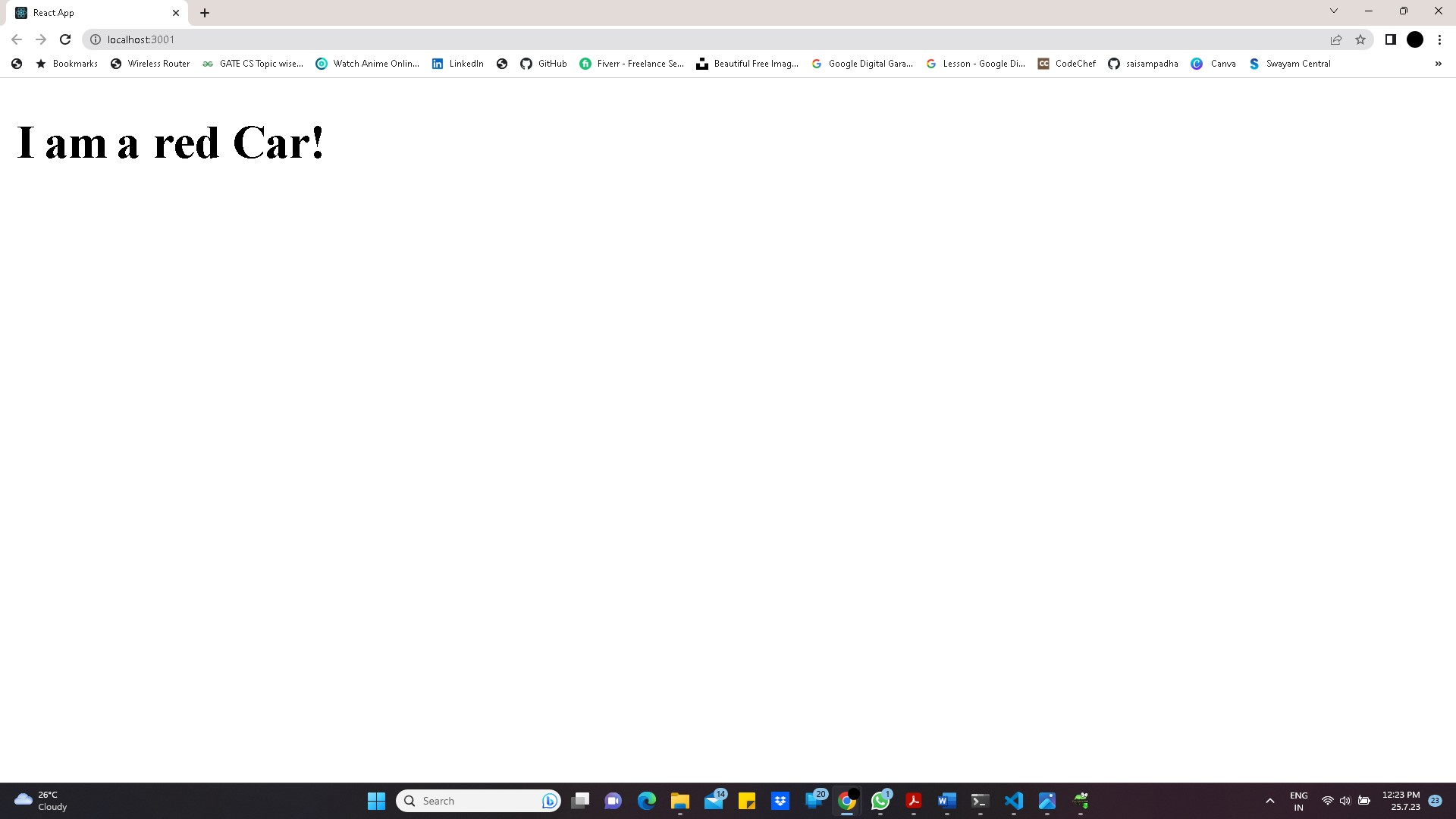
import React from 'react';

import ReactDOM from 'react-dom';

import Car from './Car';

ReactDOM.render(<Car />, document.getElementById('root'));

**Output:**



**Experiment – 3**

**Aim: Develop react application using props.**

**Source Code:**

**App.js:**

import { useState } from 'react'

import reactLogo from './assets/react.svg'

import viteLogo from '/vite.svg'

import './App.css'

function App() {

const [count, setCount] = useState(0)

return (

<div className="App">

</div>

)

}

export default App

**Index.js:**

import React from 'react'

import ReactDOM from 'react-dom/client'

import './index.css'

function Person(props){

return <h1>{props.name} is living</h1>

}

function Family(){

return(

<>

<h1>who is living in my Home</h1>

<Person name="Man"/>

</>

);

}

ReactDOM.createRoot(document.getElementById('root')).render(<Family/>

)

**App.CSS:**

#root {

max-width: 1280px;

margin: 0 auto;

padding: 2rem;

text-align: center;

}

.logo {

height: 6em;

padding: 1.5em;

will-change: filter;

transition: filter 300ms;

}

.logo:hover {

filter: drop-shadow(0 0 2em #646cffaa);

}

.logo.react:hover {

filter: drop-shadow(0 0 2em #61dafbaa);

}

@keyframes logo-spin {

from {

transform: rotate(0deg);

}

to {

transform: rotate(360deg);

}

}

@media (prefers-reduced-motion: no-preference) {

a:nth-of-type(2) .logo {

animation: logo-spin infinite 20s linear;

}

}

.card {

padding: 2em;

}

.read-the-docs {

  color: #888;

}

**Output:**

**A screenshot of a computer

Description automatically generated**

**Experiment – 4**

**Aim: Develop a Calculator React Application.**

**Source Code:**

**Index.html:**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="utf-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <meta name="theme-color" content="#000000" />

  <title>React App</title>

</head>

<body>

  <div id="root"></div>

</body>

</html>

**Index.js:**

import React from "react";

import ReactDOM from "react-dom";

import App from "./App";

const rootElement = document.getElementById("root");

ReactDOM.render(

  <React.StrictMode>

    <App />

  </React.StrictMode>,

  rootElement

);

**App.js:**

import React, { Component } from 'react';

import './styles.css';

import ResultComponent from './components/ResultComponent';

import KeyPadComponent from './components/KeyPadComponent';

class App extends Component {

  state = {

    result: ""

  }

  onClick = button => {

    if (button === "=") {

      this.calculate();

    }

    else if (button === "C") {

      this.reset();

    }

    else if (button === "CE") {

      this.backspace();

    }

    else {

      this.setState({

        result: this.state.result + button

      })

    }

  };

  calculate = () => {

    var checkResult = ''

    if (this.state.result.includes('--')) {

      checkResult = this.state.result.replace('--', '+')

    } else {

      checkResult = this.state.result;

    }

    try {

      this.setState({

        result: (eval(checkResult) || "") + ""

      })

    } catch (e) {

      this.setState({

        result: "error"

      })

    }

  };

  reset = () => {

    this.setState({

      result: ""

    })

  };

  backspace = () => {

    this.setState({

      result: this.state.result.slice(0, -1)

    })

  };

  render() {

    return (

      <div>

        <div className="calculator-body">

          <h1>Simple Calculator</h1>

          <ResultComponent result={this.state.result} />

          <KeyPadComponent onClick={this.onClick} />

        </div>

      </div>

    )

  }

}

export default App;

**Styles.css:**

.result {

    height: 60px;

    background-color: #bbb;

    width: 100%;

  }

  .result p {

    font-size: 40px;

    margin: 5px;

  }

  .calculator-body {

    max-width: 400px;

    margin: auto;

  }

  .button {

    display: block;

    background-color: #bbb;

  }

  button {

    width: 25%;

    height: 60px;

    font-size: 30px;

  }

**Components:**

**KeypadComponent.js**

import React, { Component } from 'react';

class KeyPadComponent extends Component {

  render() {

    return (

      <div className="button">

        <button name="(" onClick={e => this.props.onClick(e.target.name)}>(</button>

        <button name="CE" onClick={e => this.props.onClick(e.target.name)}>CE</button>

        <button name=")" onClick={e => this.props.onClick(e.target.name)}>)</button>

        <button name="C" onClick={e => this.props.onClick(e.target.name)}>C</button><br/>

        <button name="1" onClick={e => this.props.onClick(e.target.name)}>1</button>

        <button name="2" onClick={e => this.props.onClick(e.target.name)}>2</button>

        <button name="3" onClick={e => this.props.onClick(e.target.name)}>3</button>

        <button name="+" onClick={e => this.props.onClick(e.target.name)}>+</button><br/>

        <button name="4" onClick={e => this.props.onClick(e.target.name)}>4</button>

        <button name="5" onClick={e => this.props.onClick(e.target.name)}>5</button>

        <button name="6" onClick={e => this.props.onClick(e.target.name)}>6</button>

        <button name="-" onClick={e => this.props.onClick(e.target.name)}>-</button><br/>

        <button name="7" onClick={e => this.props.onClick(e.target.name)}>7</button>

        <button name="8" onClick={e => this.props.onClick(e.target.name)}>8</button>

        <button name="9" onClick={e => this.props.onClick(e.target.name)}>9</button>

        <button name="\*" onClick={e => this.props.onClick(e.target.name)}>x</button><br/>

        <button name="." onClick={e => this.props.onClick(e.target.name)}>.</button>

        <button name="0" onClick={e => this.props.onClick(e.target.name)}>0</button>

        <button name="=" onClick={e => this.props.onClick(e.target.name)}>=</button>

        <button name="/" onClick={e => this.props.onClick(e.target.name)}>÷</button><br/>

      </div>

    )

  }

}

export default KeyPadComponent;

**ResultComponent.js**

import React, { Component } from 'react';

class ResultComponent extends Component {

  render() {

    let { result } = this.props;

    return (

      <div className="result">

        <p>{ result }</p>

      </div>

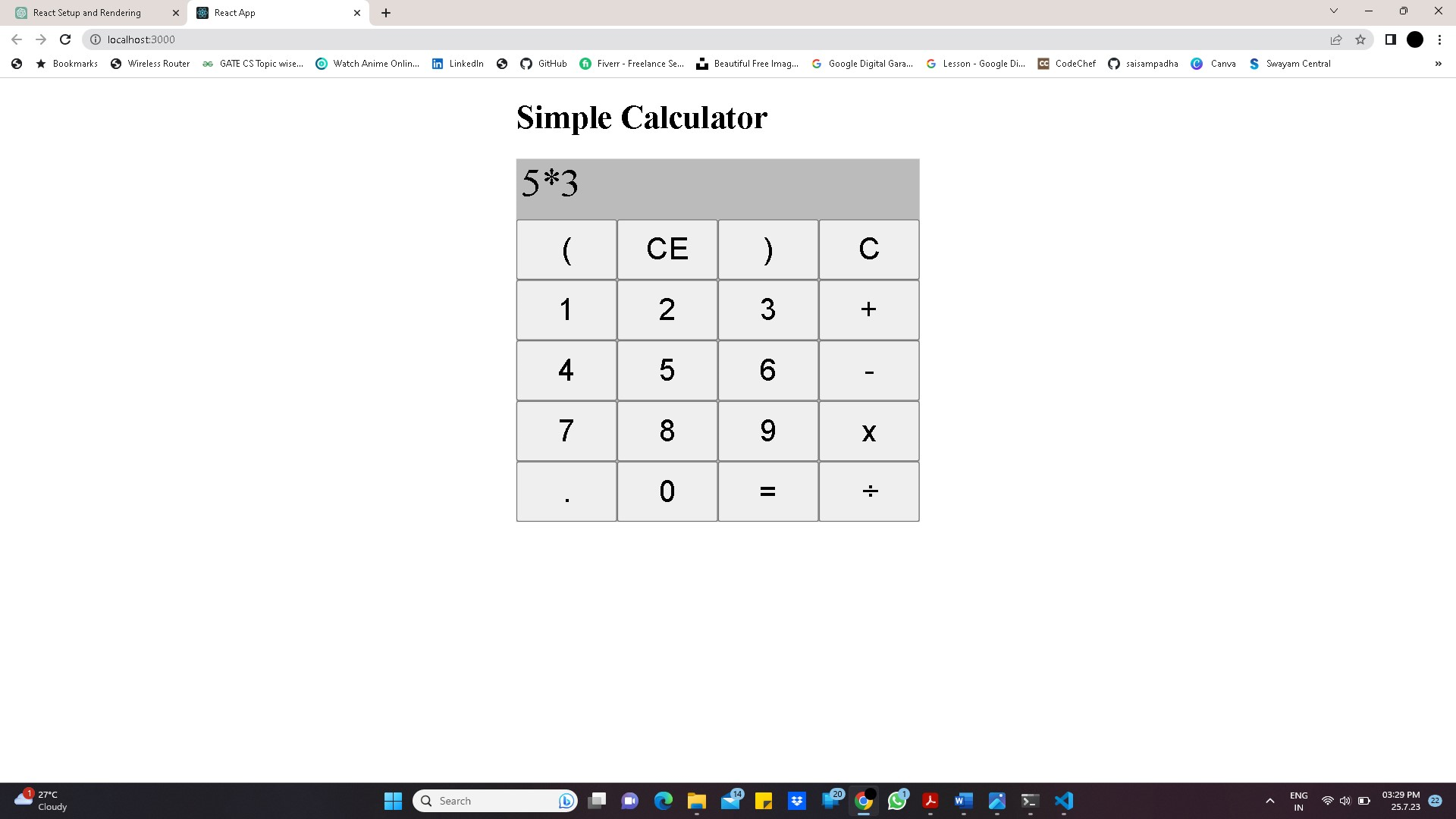
    )

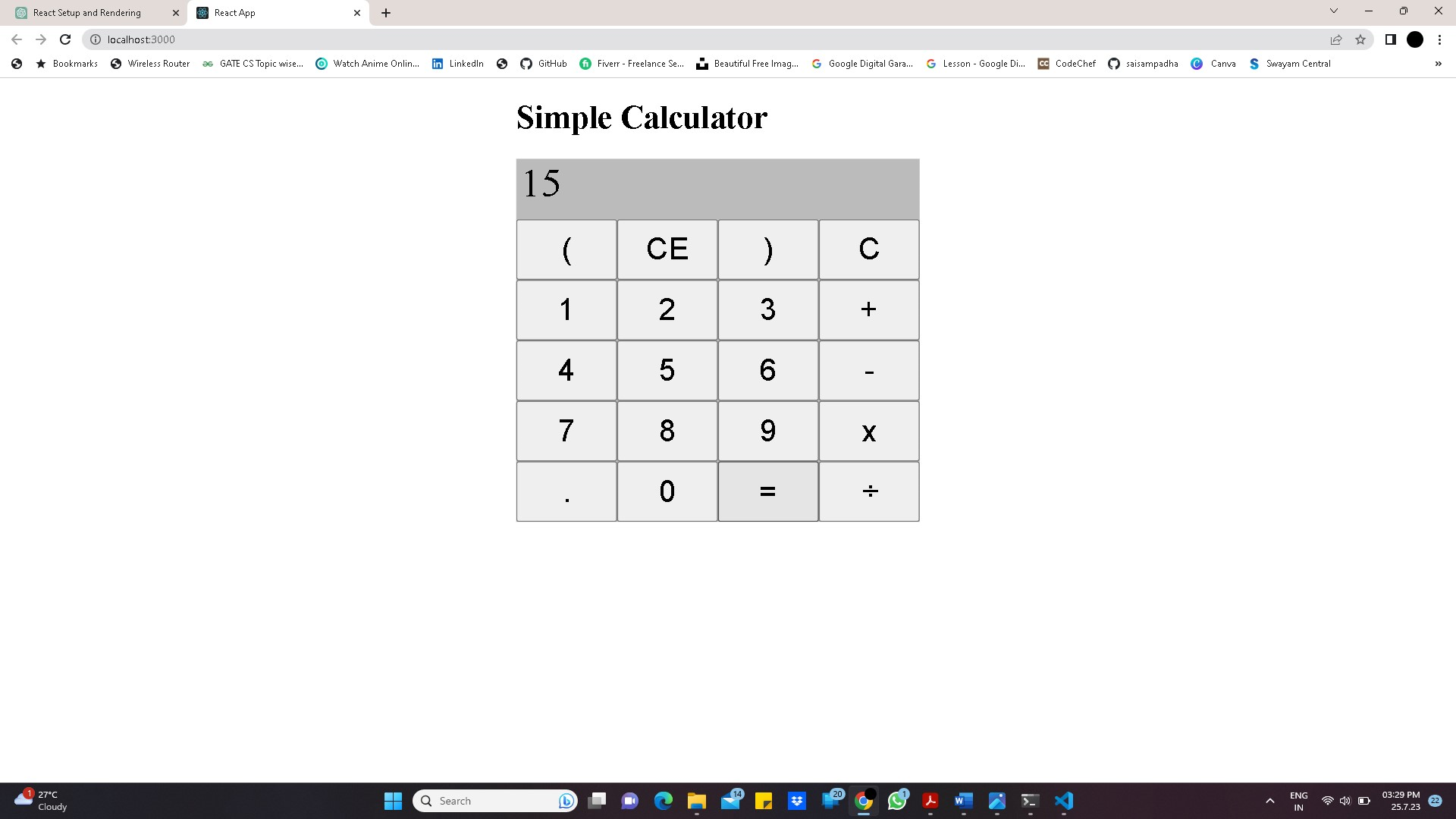
  }

}

export default ResultComponent;

**Output:**





**Experiment - 5**

**Aim: Develop React application using Forms**.

**Source Code:**

**App.js:**

import { useState } from "react";

import "./App.css";

function App() {

  const [age, setage] = useState();

  const [name, setname] = useState("");

  const [email, setemail] = useState("");

  const [ssc\_marks, setsscmarks] = useState();

  const [inter\_marks, setintermarks] = useState();

  const [phone, setphone] = useState();

  const [course1, setcourse1] = useState(false);

  const [course2, setcourse2] = useState(false);

  const [course3, setcourse3] = useState(false);

  const [gender, setgender] = useState("Male");

  const [submitted, setsubmitted] = useState(false);

  const handle\_submit = (e) => {

    e.preventDefault();

    setsubmitted(true);

  };

  return (

    <>

      {submitted ? (

        <div className="result">

          Name: {name}

          <br />

          <br />

          Age: {age}

          <br />

          <br />

          Email: {email} <br />

          <br />

          SSC marks: {ssc\_marks}

          <br />

          <br />

          Inter marks: {inter\_marks}

          <br />

          <br />

          phone no: {phone}

          <br />

          <br />

          Gender: {gender}

          <br />

          <br />

          Courses: {course1 ? "Java, " : ""}

          {course2 ? "C, " : ""}

          {course3 ? "Python" : ""}

        </div>

      ) : (

        <center>

          <form onSubmit={handle\_submit} className="forms">

            Name:{" "}

            <input

              type="text"

              value={name}

              onChange={(e) => setname(e.target.value)}

            />

            <br />

            <br />

            Age:{" "}

            <input

              type="number"

              value={age}

              onChange={(e) => setage(e.target.value)}

            />

            <br />

            <br />

            Email:{" "}

            <input

              type="email"

              value={email}

              onChange={(e) => setemail(e.target.value)}

            />{" "}

            <br />

            <br />

            Password: <input type="password" /> <br />

            <br />

            SSC marks:

            <input

              type="number"

              value={ssc\_marks}

              onChange={(e) => setsscmarks(e.target.value)}

            />

            <br />

            <br />

            Intermediate marks:

            <input

              type="number"

              value={inter\_marks}

              onChange={(e) => setintermarks(e.target.value)}

            />

            <br />

            <br />

            Phone no:

            <input

              type="number"

              value={phone}

              onChange={(e) => setphone(e.target.value)}

            />

            <br />

            <br />

            Select Courses:{" "}

            <input

              type="checkbox"

              name="java"

              checked={course1}

              onChange={(e) => setcourse1(e.target.checked)}

            />

            Java

            <input

              type="checkbox"

              name="c"

              checked={course2}

              onChange={(e) => setcourse2(e.target.checked)}

            />

            C

            <input

              type="checkbox"

              name="python"

              checked={course3}

              onChange={(e) => setcourse3(e.target.checked)}

            />

            Python

            <br />

            <br />

            <input

              type="radio"

              value="Male"

              name="Gender"

              onChange={(e) => setgender(e.target.value)}

            />

            Male

            <input

              type="radio"

              value="FeMale"

              name="Gender"

              onChange={(e) => setgender(e.target.value)}

            />

            Female

            <br />

            <br />

            <button> submit</button>

          </form>

        </center>

      )}

    </>

  );

}

export default App;

**App.CSS:**

.result{

  color:rgb(41, 69, 93);

  font-size:20px;

  text-align: center;

  margin-left:600px;

}

.forms{

  width:600px;

  background: #f4f290;

  color:black;

  font-size: 20px;

  text-align: center;

  margin-left:400px;

}

**Index.js:**

import React from 'react'

import ReactDOM from 'react-dom/client'

import App from './App.js'

import './App.css'

ReactDOM.createRoot(document.getElementById('root')).render(

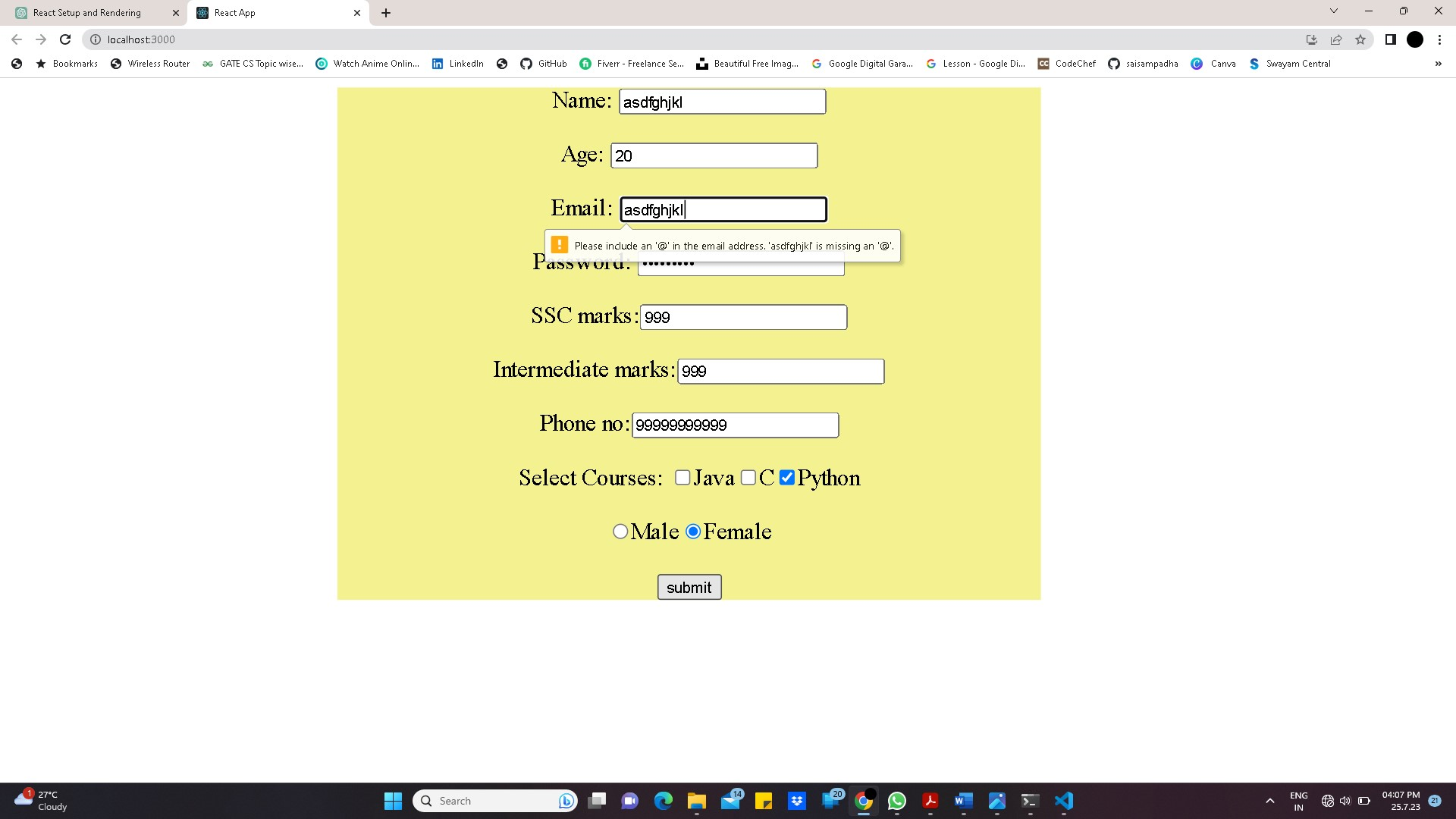
  <React.StrictMode>

    <App />

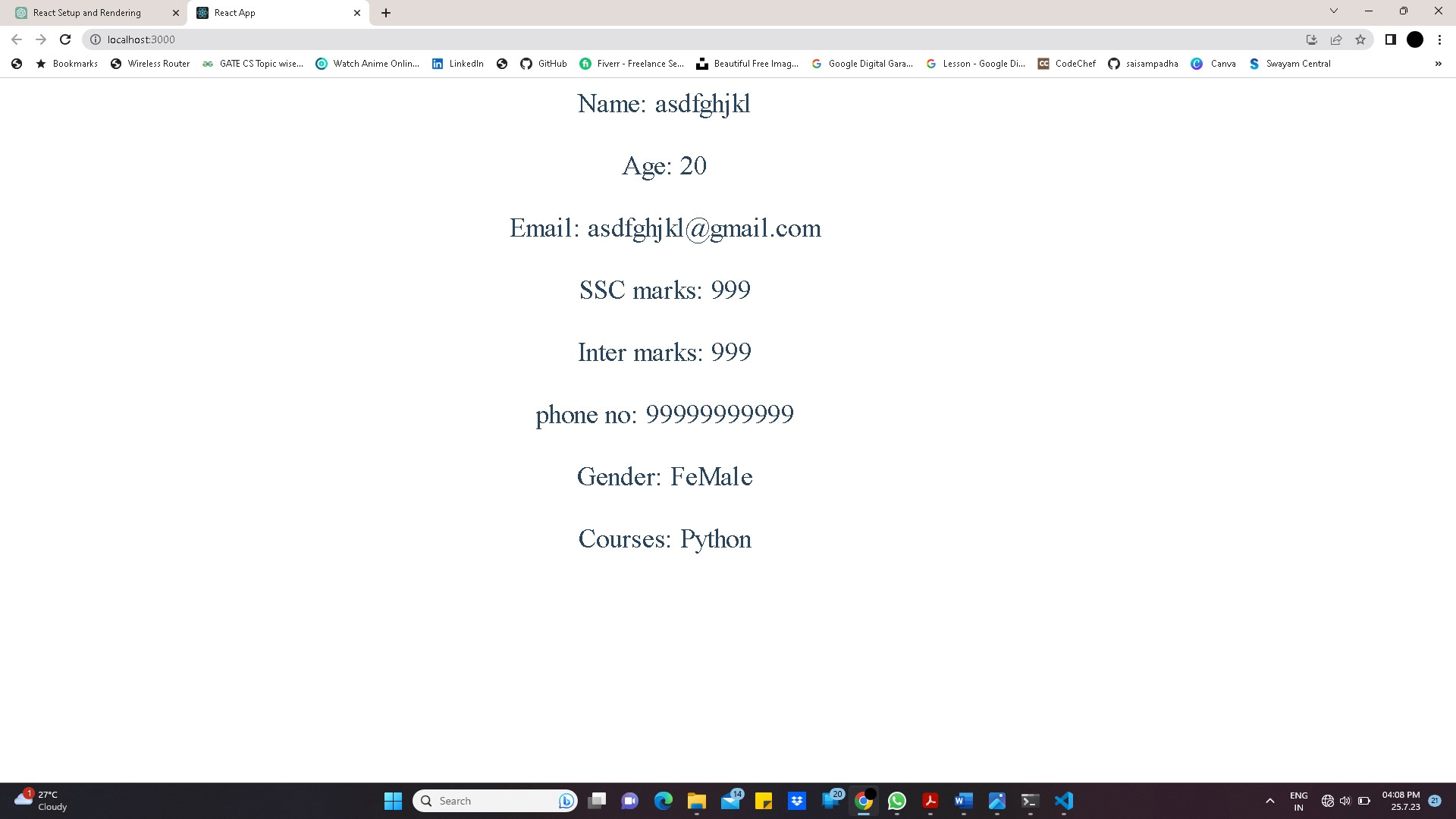
  </React.StrictMode>,

)

**Output:**



After clicking on submit button we get



**Experiment - 6**

**Aim: Develop react application using CSS style.**

**Source Code:**

**App.js:**

import React from 'react';

import List from './List';

function App() {

return (

<div className="App">

<List />

</div>

);

}

export default App;

**List.js:**

import React from 'react';

import './List.css'; // Import the CSS file

const List = () => {

const items = [

'Item 1',

'Item 2',

'Item 3',

'Item 4',

'Item 5',

];

return (

<div className="list-container">

<h2>List of Items</h2>

<ul className="item-list">

{items.map((item, index) => (

<li key={index}>{item}</li>

))}

</ul>

</div>

);

};

export default List;

**List.css:**

.list-container {

max-width: 300px;

margin: 0 auto;

padding: 20px;

border: 1px solid #ccc;

border-radius: 5px;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

.item-list {

list-style-type: none;

padding: 0;

margin: 0;

}

.item-list li {

padding: 10px;

border-bottom: 1px solid #ccc;

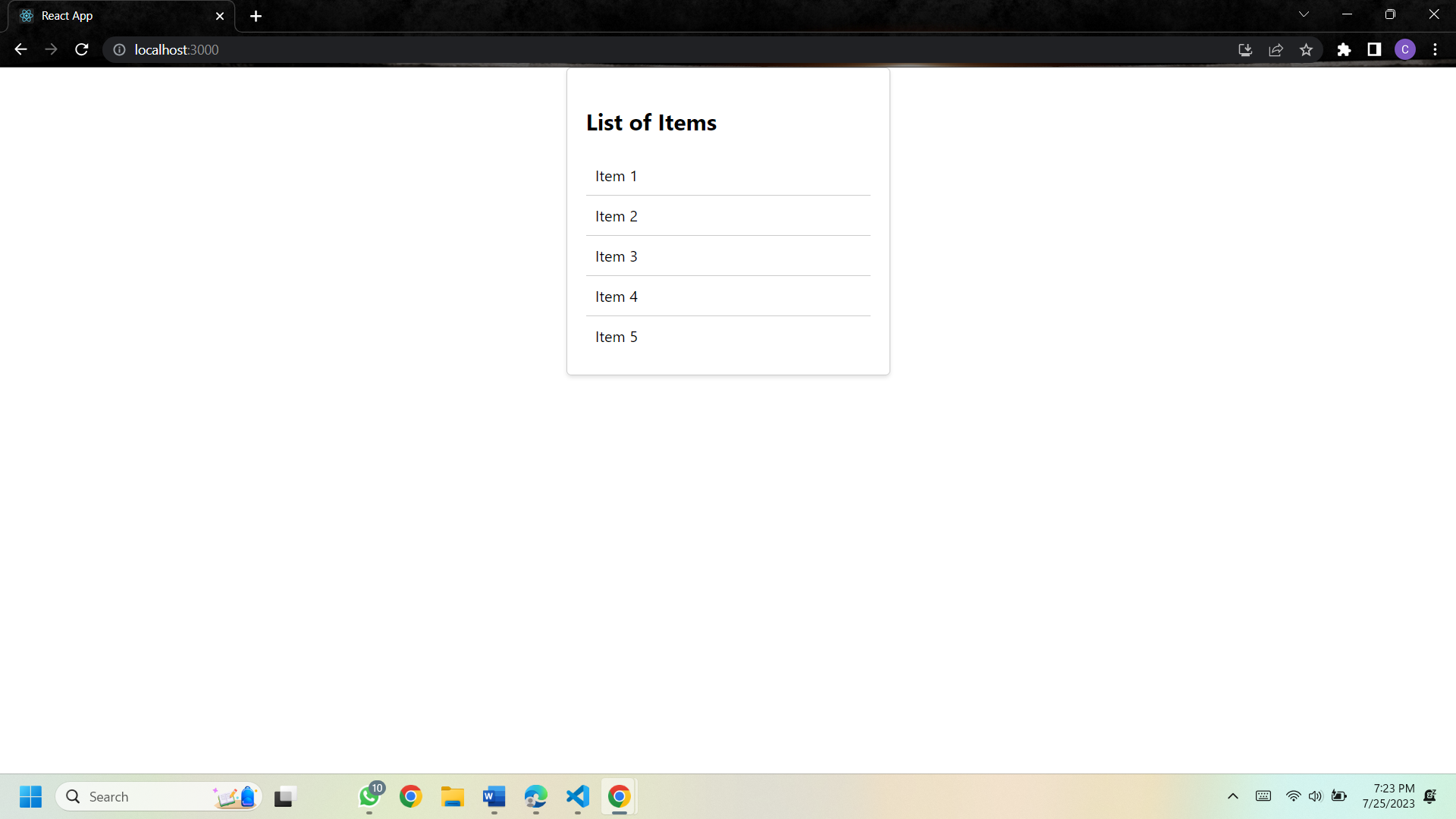
}

.item-list li:last-child {

border-bottom: none;

}

**Output:**



**Experiment - 7**

**Aim: Develop react application using events.**

**Source Code:**

**App.js:**

import React, { useState } from 'react';

function App() {

const [showMessage, setShowMessage] = useState(false);

const handleButtonClick = () => {

setShowMessage((prevState) => !prevState);

};

return (

<div className="App">

<h1>React Event Handling Example</h1>

<button onClick={handleButtonClick}>

{showMessage ? 'Hide Message' : 'Show Message'}

</button>

{showMessage && <p>This is the hidden message that becomes visible when you click the button.</p>}

</div>

);

}

export default App;

**Index.js:**

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import App from './App';

import reportWebVitals from './reportWebVitals';

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

root.render(

<React.StrictMode>

<App />

</React.StrictMode>

);

reportWebVitals();

**Output:**

**A screenshot of a computer

Description automatically generated**

**After cicking “Show Message” button**

**A screenshot of a computer

Description automatically generated**

**Experiment - 8**

**Aim: Node js application to display current date and time.**

**Source Code:**

**App.js:**

import React from 'react';

import CurrentDateTime from './CurrentDateTime';

const App = () => {

return (

<div>

<h1>My React App</h1>

<CurrentDateTime />

</div>

);

};

export default App;

**Index.js:**

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import App from './App';

import reportWebVitals from './reportWebVitals';

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

root.render(

<React.StrictMode>

<App />

</React.StrictMode>

);

reportWebVitals();

**Output:**

**A screenshot of a computer

Description automatically generated**

**Experiment - 9**

**Aim: Develop node.js application using modules.**

**Source Code:**

buf = new Buffer.allocUnsafe(26);

for (var i = 0 ; i < 26 ; i++)

{

buf[i] = i + 97;

}

console.log("Buffer Reading example...")

console.log( buf.toString('ascii')); // outputs: abcdefghijklmnopqrstuvwxyz

console.log( buf.toString('ascii',0,5));

//Buffer Writing

console.log("")

console.log("Buffer Writing example...")

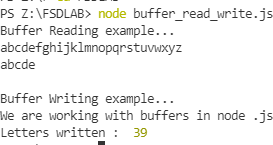
buf = new Buffer.allocUnsafe(256);

len = buf.write("We are working with buffers in node .js");

console.log(buf.toString())

console.log("Letters written : ",len)

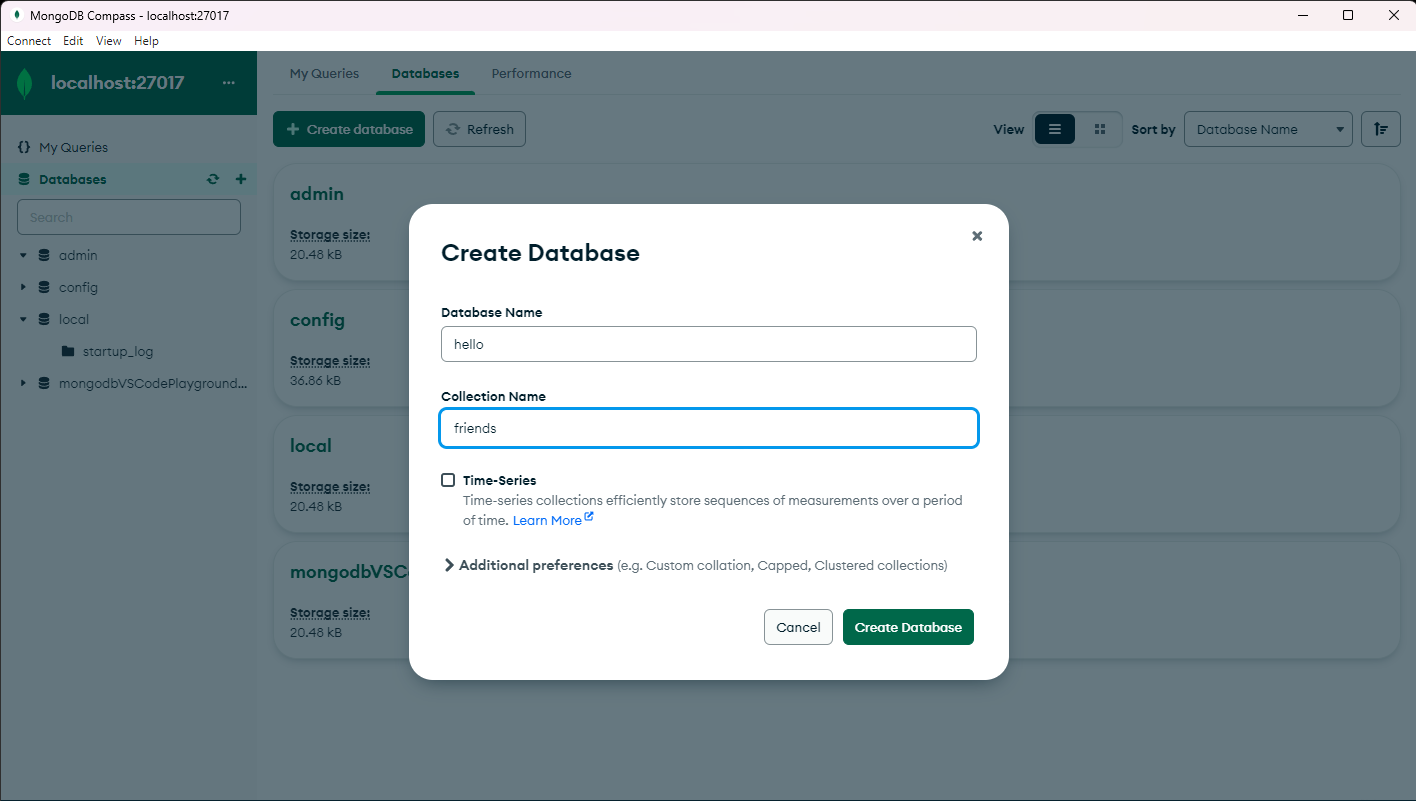
**OUTPUT:**



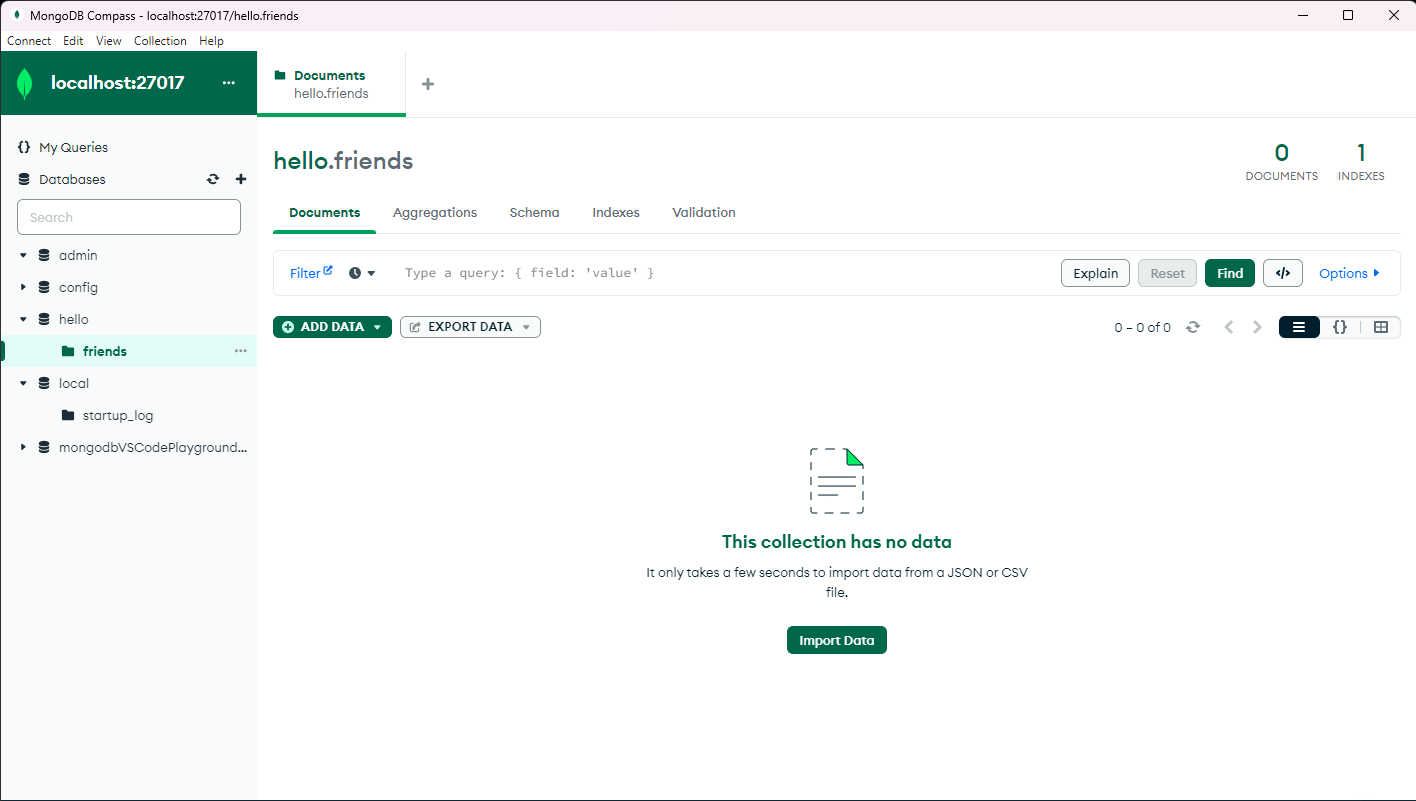
**Experiment - 10**

**Aim: Demonstrate Working with MongoDB using MongoDB Compass.**

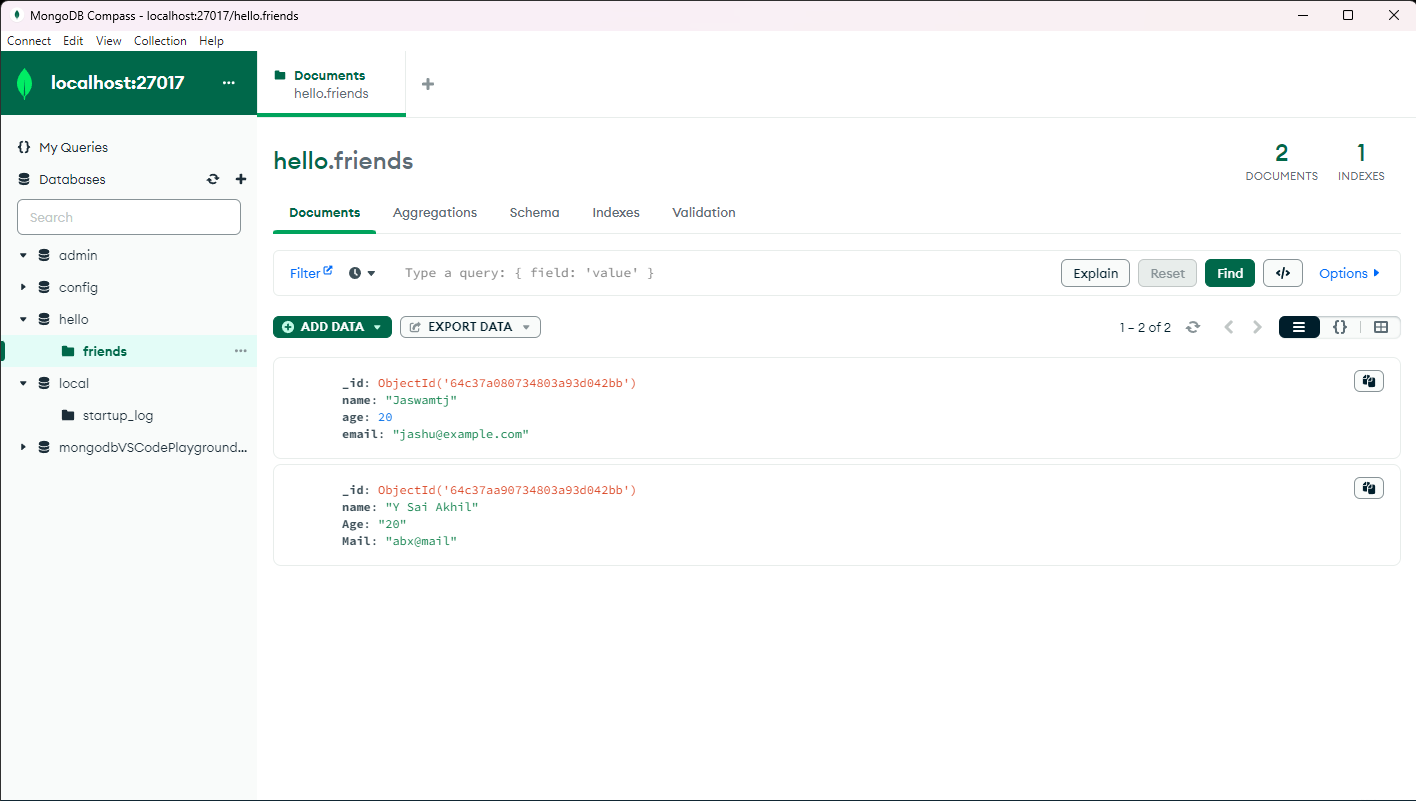
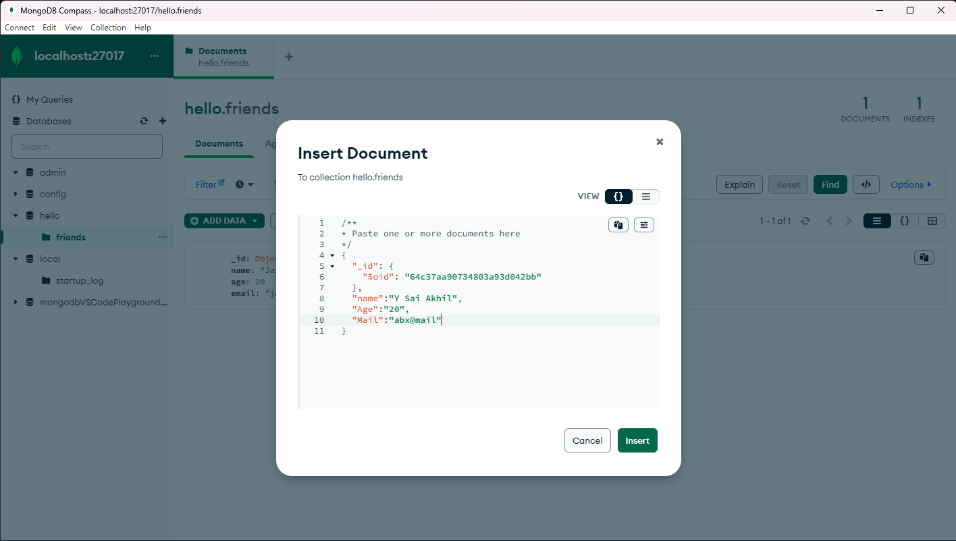
**Source Code:**

**Creating Database and Collection**

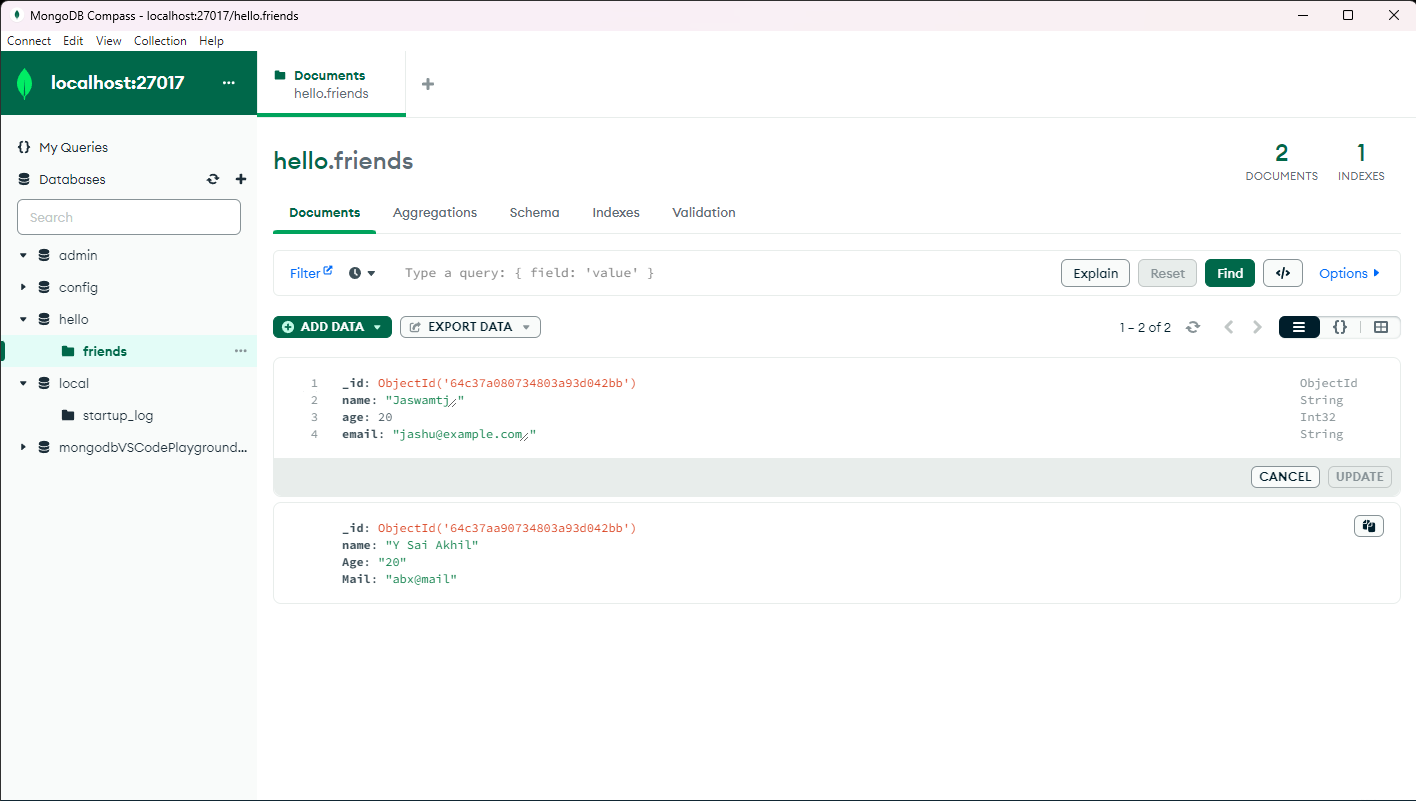
**After**

****

**INSERT DOCUMENTS**

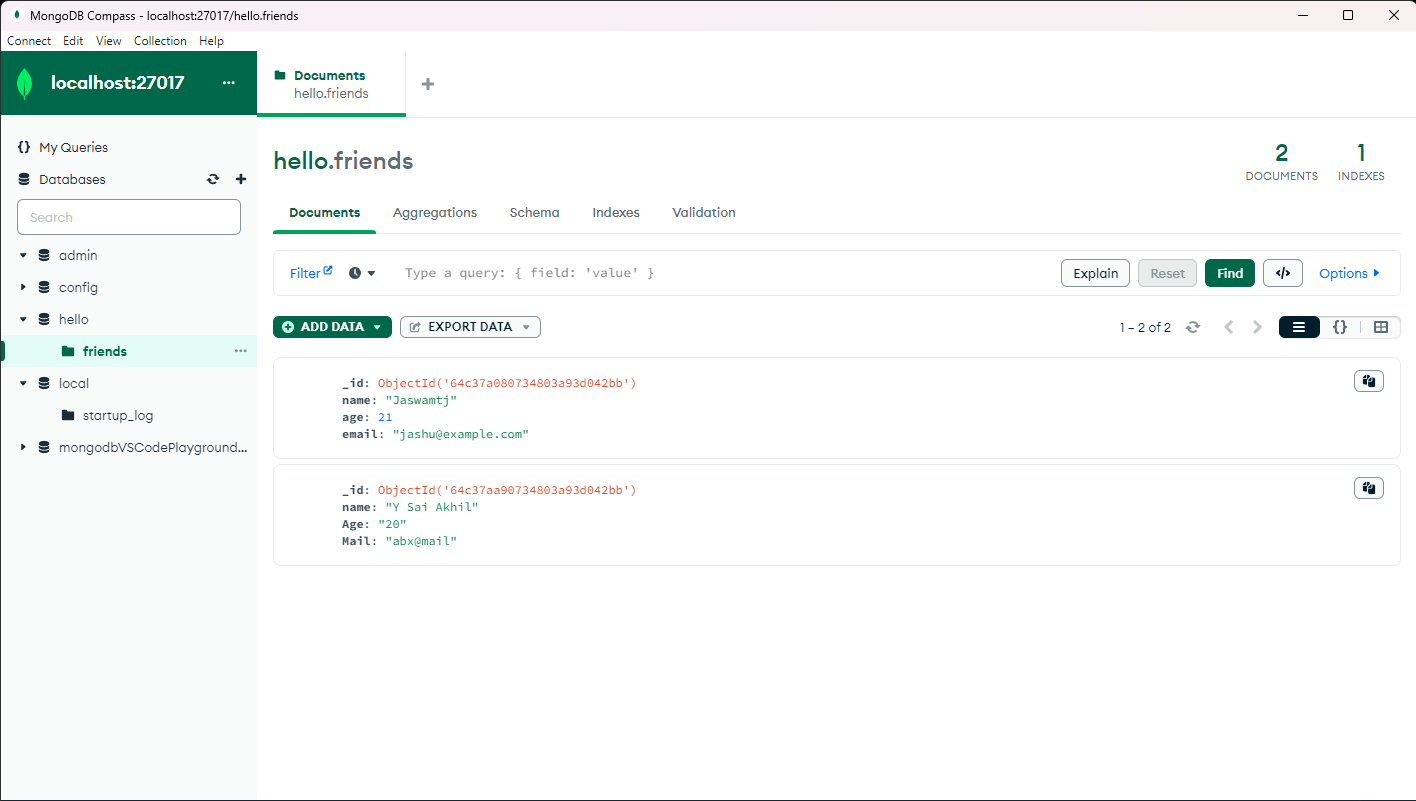
****

**UPDATION**

****

**A screenshot of a computer

Description automatically generated**

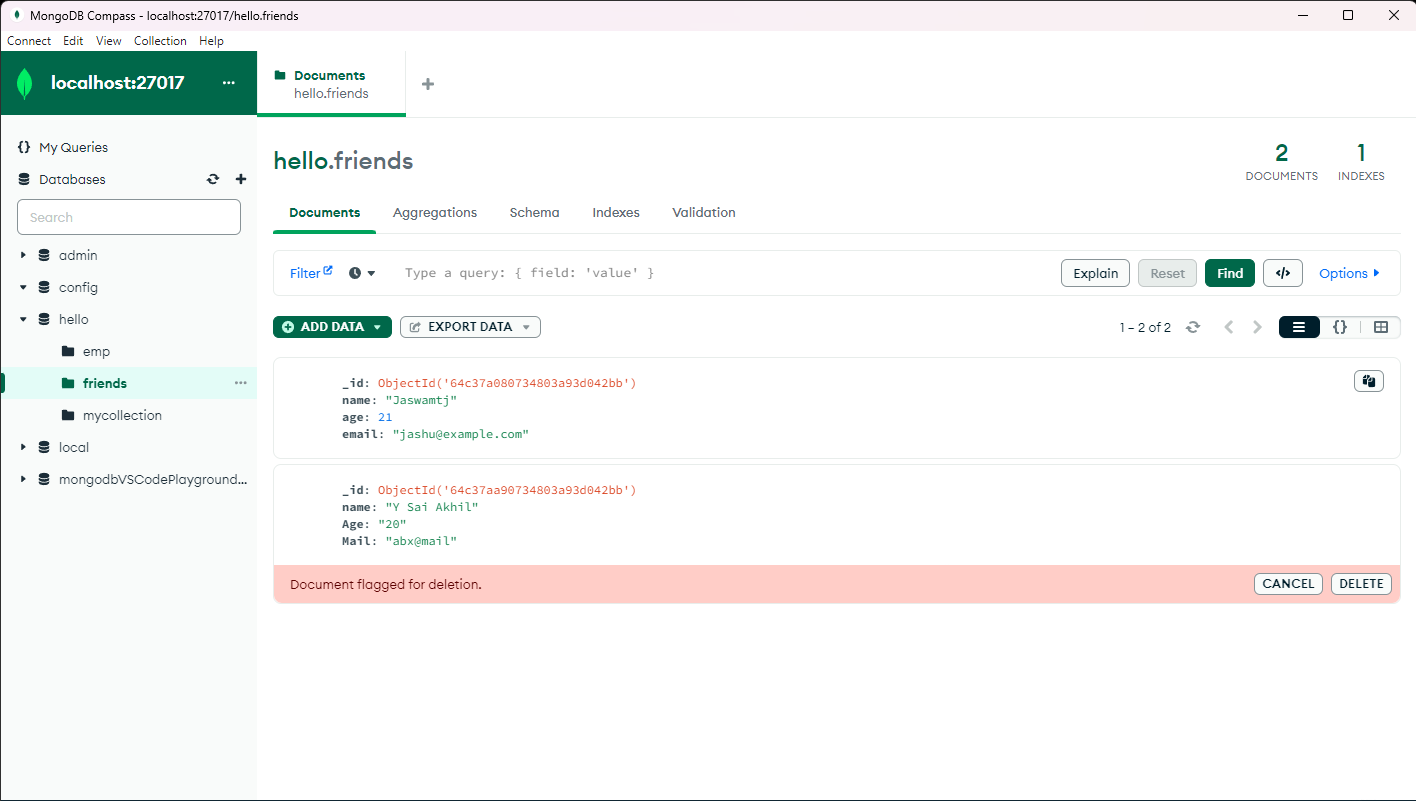
****

**Fetch**

**A screenshot of a computer

Description automatically generated**

**Deletetion**

****

**A screenshot of a computer

Description automatically generated**

**Experiment - 11**

**Aim: Demonstrate Working with MongoDB using MongoDB Shell.**

**Source Code:**

**shell:**

db.getCollection('sales').insertMany([

{ 'item': 'abc', 'price': 10, 'quantity': 2, 'date': new Date('2014-03-01T08:00:00Z') },

{ 'item': 'jkl', 'price': 20, 'quantity': 1, 'date': new Date('2014-03-01T09:00:00Z') },

{ 'item': 'xyz', 'price': 5, 'quantity': 10, 'date': new Date('2014-03-15T09:00:00Z') },

{ 'item': 'xyz', 'price': 5, 'quantity': 20, 'date': new Date('2014-04-04T11:21:39.736Z') },

{ 'item': 'abc', 'price': 10, 'quantity': 10, 'date': new Date('2014-04-04T21:23:13.331Z') },

{ 'item': 'def', 'price': 7.5, 'quantity': 5, 'date': new Date('2015-06-04T05:08:13Z') },

{ 'item': 'def', 'price': 7.5, 'quantity': 10, 'date': new Date('2015-09-10T08:43:00Z') },

{ 'item': 'abc', 'price': 10, 'quantity': 5, 'date': new Date('2016-02-06T20:20:13Z') },

]);

// Run a find command to view items sold on April 4th, 2014.

const salesOnApril4th = db.getCollection('sales').find({

date: { $gte: new Date('2014-04-04'), $lt: new Date('2014-04-05') }

}).count();

// Print a message to the output window.

console.log(`${salesOnApril4th} sales occurred in 2014.`);

// Here we run an aggregation and open a cursor to the results.

// Use '.toArray()' to exhaust the cursor to return the whole result set.

// You can use '.hasNext()/.next()' to iterate through the cursor page by page.

db.getCollection('sales').aggregate([

// Find all of the sales that occurred in 2014.

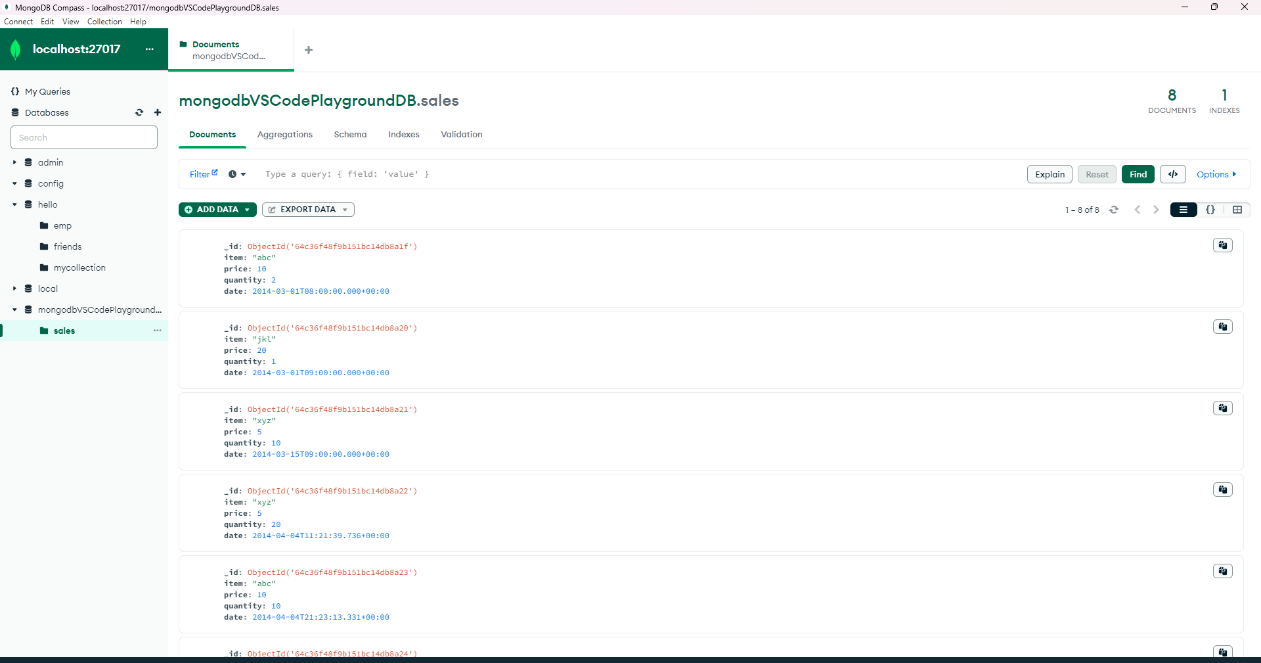
{ $match: { date: { $gte: new Date('2014-01-01'), $lt: new Date('2015-01-01') } } },

// Group the total sales for each product.

{ $group: { \_id: '$item', totalSaleAmount: { $sum: { $multiply: [ '$price', '$quantity' ] } } } }

]);

**Output:**

****

**Experiment - 12**

**Aim: Demonstrate Accessing MongoDB from Node.js**

**Source Code:**

**App.js:**

**// app.js**

const dbModule = require('./db');

async function main() {

const db = await dbModule.connectToDatabase();

const collection = db.collection('mycollection');

try {

// Insert a document

const insertResult = await collection.insertOne({

name: 'John Doe',

age: 30,

email: 'john@example.com'

});

console.log('Inserted document:', insertResult.insertedId);

// Find documents

const findResult = await collection.find({ age: { $gt: 25 } }).toArray();

console.log('Found documents:', findResult);

// Update a document

const updateResult = await collection.updateOne({ name: 'John Doe' }, { $set: { age: 31 } });

console.log('Updated document:', updateResult.modifiedCount);

// Delete a document

const deleteResult = await collection.deleteOne({ name: 'John Doe' });

console.log('Deleted document:', deleteResult.deletedCount);

} catch (error) {

console.error('Error performing operations:', error);

} finally {

dbModule.getDatabase().close();

console.log('Disconnected from MongoDB');

}

}

main();

**db.js:**

const { MongoClient } = require('mongodb');

const uri = 'mongodb://localhost:27017/emp';

let db = null;

async function connectToDatabase() {

    if (db) return db;

    const client = new MongoClient(uri, { useNewUrlParser: true, useUnifiedTopology: true });

    try {

        await client.connect();

        db = client.db();

        console.log('Connected to MongoDB');

        return db;

    } catch (error) {

        console.error('Error connecting to MongoDB:');

        throw error; // Rethrow the error to be handled by the caller

    }

}

function getDatabase() {

    if (!db)

        throw new Error('Database not connected');

    return db;

}

module.exports = { connectToDatabase, getDatabase };

**OUTPUT:**

**A screenshot of a computer

Description automatically generated**

**import React, { Component } from 'react';**

**class ClassCounter extends Component {**

**constructor() {**

**super();**

**this.state = {**

**count: 0,**

**};**

**}**

**render() {**

**return (**

**<div>**

**<h2>Class Component Counter</h2>**

**<p>Count: {this.state.count}</p>**

**<button onClick={()=>this.setState({count:this.state.count+1})}>Increment</button>**

**<button onClick={()=>this.setState({count:this.state.count-1})}>Decrement</button>**

**</div>**

**);**

**}**

**}**

**export default ClassCounter;**

**import React from 'react';**

**import ReactDOM from 'react-dom/client';**

**import './index.css';**

**import Function from './Function.js';**

**import ClassCounter from './Classcomponent';**

**import reportWebVitals from './reportWebVitals';**

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

**root.render(**

**<React.StrictMode>**

**<Function />**

**<ClassCounter/>**

**</React.StrictMode>**

**);**

**// If you want to start measuring performance in your app, pass a function**

**// to log results (for example: reportWebVitals(console.log))**

**// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals**

**reportWebVitals();**

**import React from "react";**

**import { useState } from "react";**

**function Function(){**

**const [count,setcount]=useState(0)**

**return(**

**<div>**

**<h1> count:{count}</h1>**

**<button onClick={(()=>setcount(count+1))}>increase</button>**

**<button onClick={()=>setcount(count-1)}>decrease</button>**

**</div>**

**);**

**}**

**export default Function;**

**link:** [**https://chat.openai.com/share/80bac3f0-f96a-40df-9e0b-edbd78686df8**](https://chat.openai.com/share/80bac3f0-f96a-40df-9e0b-edbd78686df8)

**props:**

**// src/App.js**

**import React from 'react';**

**import './App.css';**

**// ChildComponent receives a prop and displays it**

**const ChildComponent = (props) => {**

**return <p>Message from parent: {props.message}</p>;**

**};**

**// ParentComponent renders ChildComponent and passes a prop**

**const ParentComponent = () => {**

**return (**

**<div>**

**<h1>React Props Example</h1>**

**<ChildComponent message="Hello from parent!" />**

**</div>**

**);**

**};**

**function App() {**

**return (**

**<div className="App">**

**<ParentComponent />**

**</div>**

**);**

**}**

**export default App;**

[**https://chat.openai.com/share/78a8b5d6-c484-4861-8d3e-9670f405ade5**](https://chat.openai.com/share/78a8b5d6-c484-4861-8d3e-9670f405ade5)

**calculator:**

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

**import './App.css';**

**function App() {**

**const [result,setResult] = useState('');**

**const handleClick=(e)=>{**

**setResult(result.concat(e.target.name));**

**};**

**const calculate=(e)=>{**

**try{**

**setResult(eval(result).toString());**

**}**

**catch(error){**

**setResult('Error');**

**}**

**};**

**const backSpace=(e)=>{**

**setResult(result.slice(0,-1));**

**};**

**const clear=(e)=>{**

**setResult('');**

**};**

**return (**

**<div className="App">**

**<div className="calculator">**

**<input type="text" value={result}/>**

**<div className="keypad">**

**<button type="button" onClick={clear}>**

**Clear**

**</button>**

**<button type="button" onClick={backSpace}>**

**C**

**</button>**

**<button type="button" onClick={handleClick} name="/">**

**/**

**</button>**

**<button type="button" onClick={handleClick} name="\*">**

**\***

**</button>**

**<button type="button" onClick={handleClick} name="-">**

**-**

**</button>**

**<button type="button" onClick={handleClick} name="+">**

**+**

**</button>**

**<button type="button" onClick={handleClick} name="9">**

**9**

**</button>**

**<button type="button" onClick={handleClick} name="8">**

**8**

**</button>**

**<button type="button" onClick={handleClick} name="7">**

**7**

**</button>**

**<button type="button" onClick={handleClick} name="6">**

**6**

**</button>**

**<button type="button" onClick={handleClick} name="5">**

**5**

**</button>**

**<button type="button" onClick={handleClick} name="4">**

**4**

**</button>**

**<button type="button" onClick={handleClick} name="3">**

**3**

**</button>**

**<button type="button" onClick={handleClick} name="2">**

**2**

**</button>**

**<button type="button" onClick={handleClick} name="1">**

**1**

**</button>**

**<button type="button" onClick={handleClick} name="0">**

**0**

**</button>**

**<button type="button" onClick={calculate} >**

**=**

**</button>**

**</div>**

**</div>**

**</div>**

**);**

**}**

**export default App;**

**css**

**.App {**

**text-align: center;**

**}**

**.calculator{**

**border: 2px solid gray;**

**margin:100px auto;**

**width: 300px;**

**padding: 10px;**

**background-color: orange;**

**}**

**input[type="text"]{**

**width:90%;**

**padding: 2%;**

**border-radius: 5px;**

**margin-bottom: 1%;**

**}**

**button{**

**padding: 5px;**

**border: none;**

**border-radius: 5px;margin:5px;**

**}**

**.keypad{**

**display: grid;**

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

**}**

**.App-logo {**

**height: 40vmin;**

**pointer-events: none;**

**}**

**@media (prefers-reduced-motion: no-preference) {**

**.App-logo {**

**animation: App-logo-spin infinite 40s linear;**

**}**

**}**

**.App-header {**

**background-color: #282c34;**

**min-height: 100vh;**

**display: flex;**

**flex-direction: column;**

**align-items: center;**

**justify-content: center;**

**font-size: calc(10px + 2vmin);**

**color: white;**

**}**

**.App-link {**

**color: #61dafb;**

**}**

**@keyframes App-logo-spin {**

**from {**

**transform: rotate(0deg);**

**}**

**to {**

**transform: rotate(360deg);**

**}**

**}**

**form:**

**import { useState } from 'react'**

**import React from 'react'**

**import './App.css';**

**function App() {**

**const [age, setage] = useState();**

**const [name, setname] = useState('');**

**const [email,setemail]=useState('');**

**const[ssc\_marks,setsscmarks]=useState();**

**const[inter\_marks,setintermarks]=useState();**

**const[phone,setphone]=useState();**

**const[course1,setcourse1]=useState(false);**

**const[course2,setcourse2]=useState(false);**

**const[course3,setcourse3]=useState(false);**

**const [gender,setgender] = useState("");**

**const [submitted,setsubmitted] = useState(false);**

**const handle\_submit = (e) => {**

**e.preventDefault();**

**setsubmitted(true);**

**}**

**return(**

**<>**

**{**

**submitted ? (**

**<div className='result'>**

**Name: {name}<br/><br/>**

**Age: {age}<br/><br/>**

**Email: {email} <br/><br/>**

**SSC marks: {ssc\_marks}<br/><br/>**

**Inter marks: {inter\_marks}<br/><br/>**

**phone no: {phone}<br/><br/>**

**Gender: {gender}<br/><br/>**

**Courses: {course1 ? ("Java, "):("")}{course2 ? ("C, "):("")}{course3 ? ("Python"):("")}**

**</div>**

**): (**

**<center>**

**<form onSubmit={handle\_submit} className='forms'>**

**Name: <input type="text" value={name} onChange={(e) => setname(e.target.value)}/><br/><br/>**

**Age: <input type="number" value={age} onChange={(e) => setage(e.target.value)}/><br/><br/>**

**Email: <input type="email" value={email} onChange={(e) => setemail(e.target.value)}/> <br/><br/>**

**Password: <input type="password"/> <br/><br/>**

**SSC marks:<input type="number" value={ssc\_marks} onChange={(e) => setsscmarks(e.target.value)}/><br/><br/>**

**Intermediate marks:<input type="number" value={inter\_marks} onChange={(e) => setintermarks(e.target.value)}/><br/><br/>**

**Phone no:<input type="number" value={phone} onChange={(e) => setphone(e.target.value)}/><br/><br/>**

**Select Courses: <input type="checkbox" name="java" checked={course1} onChange={(e) => setcourse1(e.target.checked)}/>Java**

**<input type="checkbox" name="c" checked={course2} onChange={(e) => setcourse2(e.target.checked)}/>C**

**<input type="checkbox" name="python" checked={course3} onChange={(e) => setcourse3(e.target.checked)}/>Python<br/><br/>**

**<input type="radio" value="Male" name="Gender" onChange={(e) => setgender(e.target.value)}/>Male**

**<input type="radio" value="FeMale" name="Gender" onChange={(e) => setgender(e.target.value)}/>Female<br/><br/>**

**<button> submit</button>**

**</form>**

**</center>**

**)}**

**</>**

**)}**

**export default App;**