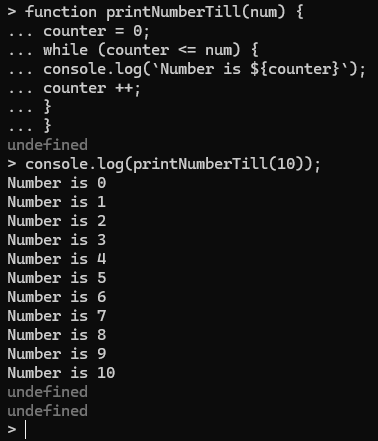
**INDEX**

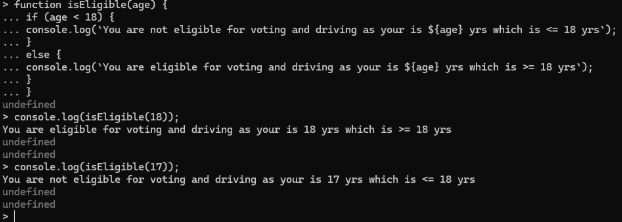
| **Practical No.** | **Practical Name** |
| --- | --- |
| 1) | Perform the REPL in Node.js |
| 2) | Using modules, perform the Arithmetic Operations |
| 3) | Using modules, find the Area of a Circle, Rectangle, Square |
| 4) | Write a program to print the Prime Numbers from 1 to 50 |
| 5) | Write a program to find the reverse of a four-digit number |
| 6) | Write a program to find if the number is odd or even |
| 7) | Write a program to check if the entered number is Armstrong or not |
| 8) | Write a program to take the marks of four subjects from the user and check if the student has passed the examination or not, calculate percentage and grade |
| 9) | Write a program to print the Fibonacci series |
| 10) | Write a program to convert the temperature entered by the user |
| 11) | Write a program to demonstrate the factorial of a number using Anonymous Functions |
| 12) | Write a program to demonstrate the Pattern using Anonymous Functions |
| 13) | Write a program to demonstrate the arithmetic operations using Callback Functions |
| 14) | Write a program to demonstrate the setTimeout function |
| 15) | Write a program to place the order for a pizza using EventsEmitter |
| 16) | Write a program to demonstrate EventEmitters functions |
| 17) | Write a program to calculate the salary using EventEmitter class |
| 18) | Write a program to create an EventEmitters to print the sum of odd and even numbers from an array |
| 19) | Write a program to demonstrate File handling in Node.js |
| 20) | Write a Node.js code to display the Employee Job Registration Form saved in an HTML file in response to the client’s access request to the server |
| 21) | Write a program to handle request URLs between various HTML pages using HTTP Server |
| 22) | Write a program to implement the database in node.js |

**1) Perform the REPL in Node.js**

**a.** Print numbers using a while loop.



**b.** Using a conditional statement estimated the eligibility of one for voting and driving.



**2) Using modules, perform the Arithmetic Operations**

**ArithmeticOperation.js**

function add (a,b){

return a+b;

}

function sub (a,b){

return a-b;

}

function mul (a,b){

return a\*b;

}

function div (a,b){

return a/b;

}

exports.add=add;

exports.sub=sub;

exports.mul=mul;

exports.div=div;

**Demo.js**

const req1 = require("./ArithOperations")

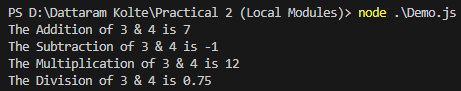
console.log(`The Addition of 3 & 4 is ${req1.add(3,4)}`)

console.log(`The Subtraction of 3 & 4 is ${req1.sub(3,4)}`)

console.log(`The Multiplication of 3 & 4 is ${req1.mul(3,4)}`)

console.log(`The Division of 3 & 4 is ${req1.div(3,4)}`)

**OUTPUT:**

****

**3) Using modules find the Area of a circle, rectangle, square.**

**Area.js**

function circleArea(r){

return 3.142\*(r\*\*2);

}

exports.circleArea=circleArea;

function squareArea(s){

return s\*\*2;

}

exports.squareArea=squareArea;

function recArea(l,b){

return l\*b;

}

exports.recArea=recArea;

**Demo.js**

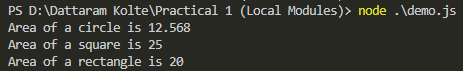
var req1 = require("./Area");

console.log("Area of a circle is "+req1.circleArea(2));

console.log("Area of a square is "+req1.squareArea(5));

console.log("Area of a rectangle is "+req1.recArea(5,4));

**OUTPUT:**



**4) Write a program to print the Prime Numbers from 1 to 50.**

**PrimeNum.js**

function primeNum(){

console.log("The prime numbers from 1 to 50\n")

for(i=2; i<=50; i++){

var count=0;

for(j=2; j<=i/2; j++){

if(i%j==0){

count=1;

break;

}

}

if (count==0){

console.log(i);

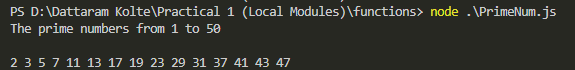
}

}

}

primeNum();

**OUTPUT:**



**5) Write a program to find the reverse of a four-digit number.**

**ReverseNum.js**

function revNum(a){

var rev=0

var temp=a;

while(a!=0){

r=a%10;

rev=(rev\*10)+r;

a=parseInt(a/10);

}

console.log("Reverse of "+temp+" is "+rev);

}

revNum(1234);

revNum(3041602);

**OUTPUT:**

****

**6) Write a program to find if the number is odd or even.**

**OddEven.js**

function oddeve(a){

if (a%2 == 0){

console.log(a+" is even.");

}

else{

console.log(a+" is odd.");

}

}

oddeve(2);

oddeve(3);

**OUTPUT:**

****

**7) Write a program to check if the entered number is Armstrong or not.**

**Armstrong.js**

const prompt = require("prompt-sync")();

const num=parseInt(prompt("Enter a number: "));

let temp1=num;

let sum=0;

while(temp1>0){

let reminder=temp1%10;

sum=sum+(reminder\*\*num.toString().length);

temp1=parseInt(temp1/10);

}

if (sum == num){

console.log("Number is Armstrong");

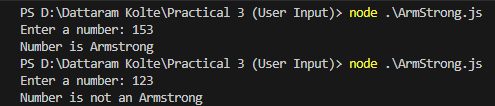
}

else{

console.log("Number is not an Armstrong");

}

**OUTPUT:**



**8) Write a program to take the marks of four subjects from user and check is the student has passed the examination or not, if passed then calculate the percentage and grade of the student.**

**UserInput.js**

const prompt = require("prompt-sync")();

const sub1 = parseInt(prompt('Enter marks for sub1: '));

const sub2 = parseInt(prompt('Enter marks for sub2: '));

const sub3 = parseInt(prompt('Enter marks for sub3: '));

const sub4 = parseInt(prompt('Enter marks for sub4: '));

let obt = sub1+sub2+sub3+sub4

let percentage = (obt/400)\*100

if (sub1>=45 && sub2>=45 && sub3>=45 && sub4>=45)

{

console.log("Your percentage is "+percentage.toFixed(2)+"%")

if(percentage>85)

{

console.log("Your Grade is O")

}

else if(percentage>70)

{

console.log("Grade is A")

}

else if(percentage>60)

{

console.log("Grade is B")

}

else if(percentage>=45)

{

console.log("Grade is C")

}

}

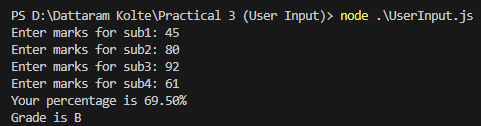
else

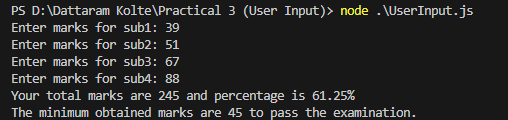
{

console.log("Your total marks are "+obt+" and percentage is "+percentage.toFixed(2)+"%\nThe minimum obtained marks are 45 to pass the examination.")

}

**OUTPUT:**





**9) Write a program to print the Fibonacci series.**

**Fibonacci.js**

const promtp = require("prompt-sync")()

const num = parseInt(promtp("Enter a number: "))

let a = 0

let b = 1

process.stdout.write(`The ${num} numbers of fibonacci series: ${a} ${b}`)

for (i = 1; i <= num; i++){

let c = a+b

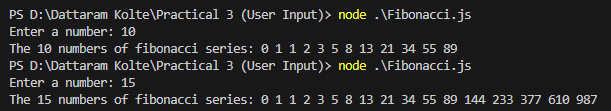
process.stdout.write(` ${c}`)

a = b

b = c

}

**OUTPUT:**

****

**10) Write a program to convert the temperature entered by the user**

**Temperature.js**

const prompt = require("prompt-sync")()

function c2f(temp){

return ((9/5)\*temp)+32

}

function f2c(temp){

return ((5/9)\*(temp-32))

}

const sel = (prompt("Select Temperature (Celsius-'C' or Fahrenheit-'F') : ")).charAt(0)

if (sel == "C" || sel == "c"){

let temp = parseFloat(prompt("Enter the temperature in Celsius: "))

console.log(`The temperature in Fahrenheit is ${c2f(temp).toFixed(2)} °F`)

}

else if (sel == "F" || sel == "f"){

let temp = parseFloat(prompt("Enter the temperature in Fahrenheit: "))

console.log(`The temperature in Celsius is ${f2c(temp).toFixed(2)} °C`)

}

else{

console.log("You have selected the wrong input")

}

**11) Write a program to demonstrate the factorial of number using the Anonymous Functions**

**Factorial.js**

//Creating a Anonymous Function

const factorial = function (num) {

let sum=1;

for(i=2; i<=num; i++){

sum \*= i;

}

return sum;

}

console.log("The factorial of 6 is "+factorial(6))

**OUTPUT:**

****

**12) Write a program to demonstrate the Pattern using the Anonymous Functions.**

**Pattern.js**

//Creating a Arrow Function

const pattrn = (num) => {

for( i = 1; i <= num; i++){

for (j = i; j < num; j++){

process.stdout.write(" ");

}

for( k = 1; k <= i-1; k++){

process.stdout.write("\* ");

}

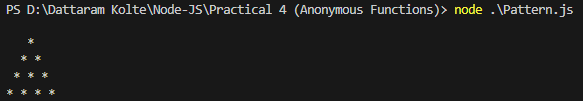
console.log();

}

}

pattrn(5)

**OUTPUT:**

****

**13) Write a program to demonstrate the arithmetic operations using the Callback Functions.**

**Callback.js**

//Creating a callback function for Addition

function add (a,b) {

return a+b;

}

//Creating an anonymous callback function for Multiplication

const mul = function (a,b) {

return a\*b

}

//Creating a function for calling callback function

function func (callback) {

console.log(callback(5,2))

}

func(add)

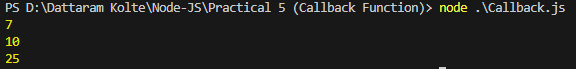
func(mul)

func( (num, p) => {

return num\*\*p

})

**OUTPUT:**

****

**14) Write a program to demonstrate the setTimeout function**

**SetTimeout.js**

const greet = function() {

process.stdout.write("Hello, Dattaram")

}

console.log("Welcome to the Node JS")

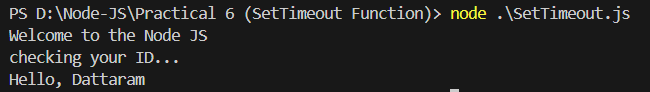
setTimeout(() => {

console.log("checking your ID...")

},2000)

setTimeout(greet,3000)

**OUTPUT:**

****

**15) Write a program to place the order for a pizza using the EventsEmitter class**

**Event1.js**

const EventEmitter = require("node:events")

const emitter = new EventEmitter()

emitter.on("order-pizza", (size,toppings) => {

console.log(`Order recieved ! Baking the ${size} pizza with ${toppings} topppings.`)

})

emitter.on("order-ready",() => {

console.log("Order is ready !!!")

})

emitter.emit("order-pizza","Large","Onion")

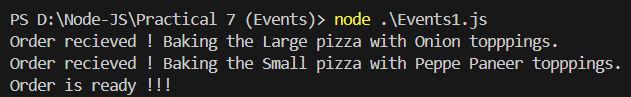
emitter.emit("order-pizza","Small","Peppe Paneer")

setTimeout(() => {

emitter.emit("order-ready")

},4000)

**OUTPUT:**

****

**16) Write a program to demonstrate Events by the same name**

**Event2.js**

const events = require("node:events")

const eventEmitter = new events.EventEmitter()

function listner1(){

console.log("Event received by Listner 1")

}

function listner2(){

console.log("Event received by Listner2");

}

eventEmitter.addListener("Write",listner1);

eventEmitter.on("Write",listner2);

eventEmitter.emit("Write")

console.log(eventEmitter.listenerCount("Write"))

eventEmitter.removeListener("Write",listner1);

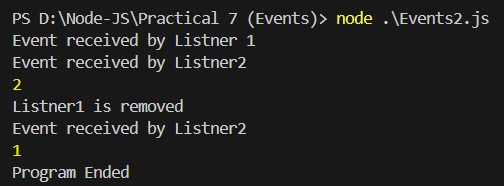
console.log("Listner1 is removed")

eventEmitter.emit("Write")

console.log(eventEmitter.listenerCount("Write"))

console.log("Program Ended")

**OUTPUT:**

****

**17) Write a program to calculate the salary using the EventEmitter class**

**EventEmitter.js**

const EventEmitter = require("node:events") //EventEmitter is a class

//Extending the event emitter class to another class

class SalaryCalculator extends EventEmitter{

//Method to calculate the salary

calculateSalary(basic, ta){

const hra = 0.2 \* basic //20% of the basic

const da = basic // 100% of the basic

const inc\_tax = 0.3 \* basic //30% of the basic

const prof\_tax = 200 // Professional tax is 200

const salary = basic + hra + da + ta - inc\_tax - prof\_tax

this.emit("salary\_disp",salary)

}

}

//Creating an object of class SalaryCalculator

const obj1 = new SalaryCalculator()

//Creating an event of the class SalaryCalculator

obj1.on('salary\_disp',(salary) => {

console.log(`The calculated salary is ${salary} Rs`)

})

//Calling the method of the class SalaryCalculator

obj1.calculateSalary(8000,1000)

**OUTPUT:**

****

**18) Write a program to create an event to print the sum of odd and even numbers from an array**

**ArrayEmitter.js**

const EventEmitter = require("node:events")

const emitter = new EventEmitter()

//Creating an event to print the sum of the even numbers

emitter.on("even\_disp",(num) => {

console.log(`The sum of the even numbers in array is ${num}`)

})

//Creating an event to print the sum of the odd numbers

emitter.on("odd\_disp",(num) => {

console.log(`The sum of the odd numbers in array is ${num}`)

})

//Point of Execution

const arr = [1,2,3,4,5,6,7,8,9,10,22,21,20]

let even = 0

let odd = 0

for (i=0; i<arr.length; i++){

if (arr[i]%2 == 0){

even+=arr[i]

}

else{

odd+=arr[i]

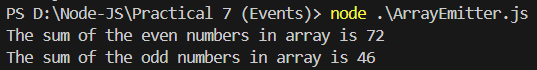
}

}

emitter.emit('even\_disp',even)

emitter.emit('odd\_disp',odd)

**OUTPUT:**

****

**19) Write a program to demonstrate File handling in Node.js**

**FileHandle.js**

const fs = require("node:fs")

//Creating and writing into a file

fs.writeFile("Datta.txt","Hello, Dattaram!",function (err, data) {

console.log("Writing a file...")

})

//Appedning the text into a file

fs.appendFile("Datta.txt","\nThis is the appended statements.",function (err, data) {

console.log("Appending a file...")

})

//Reading a file

fs.readFile("Datta.txt","utf8",function (err, data) {

console.log("Reading a file...")

console.log(data)

})

//Deleting file

fs.unlink("Datta.txt",function(err, data) {

console.log("Deleting a file...")

console.log("File deleted successfully.")

})

//OpenSync and WriteSync

const fd = fs.openSync("Datta.txt","r+")

const text = "John Doe"

const position = 1

const obs = fs.writeSync(fd, text, position, "utf8")

console.log(obs)

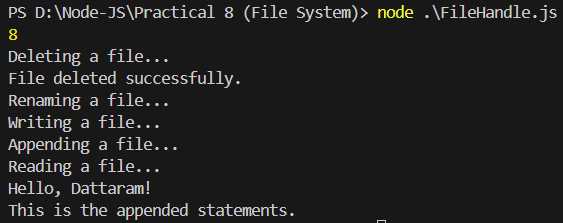
// Renaming a file

fs.rename("Datta.txt","TextDocument.txt",function (err, data) {

console.log("Renaming a file...")

})

**OUTPUT:**

****

**20) Write a Node.js code to display the Employee Job Registration Form saved in an HTML file in response to the client’s access request to the server**

**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<title>Dattaram Kolte</title>

</head>

<body>

<h1>Employee Registeration Form</h1>

<form action="formFirst" method="post" style="text-size-adjust: 24px;">

<table cellpadding="5" cellspacing="0">

<tr>

<td><label>First Name:</label></td>

<td><input type="text" name="fname"></td>

</tr>

<tr>

<td><label>Last Name:</label></td>

<td><input type="text"></td>

</tr>

<tr>

<td><label>Date of Birth:</label></td>

<td><input type="date"></td>

</tr>

<tr>

<td><label>Gender:</label></td>

<td>

<select >

<option value="0">select your gender</option>

<option value="1">Male</option>

<option value="2">Female</option>

</select>

</td>

</tr>

<tr>

<td>Phone Number:</td>

<td><input type="text"></td>

</tr>

<tr>

<td>Email ID:</td>

<td><input type="text"></td>

</tr>

<tr>

<td>Department:</td>

<td>

<select>

<option>select your department</option>

<option>HR</option>

<option>Sales</option>

<option>Management</option>

<option>IT</option>

</select>

</td>

</tr>

<tr>

<td colspan="2" style="text-align: center;">

<button>submit</button>

</td>

</tr>

</table>

</form>

</body>

</html>

**HttpServer2.js**

const http = require("node:http")

const fs = require("node:fs")

const server = http.createServer((req,res) => {

fs.readFile("index.html", (err, data) => {

if (data) {

res.writeHead(200, {'Content-Type':"text/html"})

res.end(data)

}

})

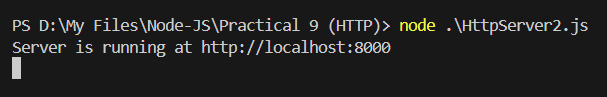
})

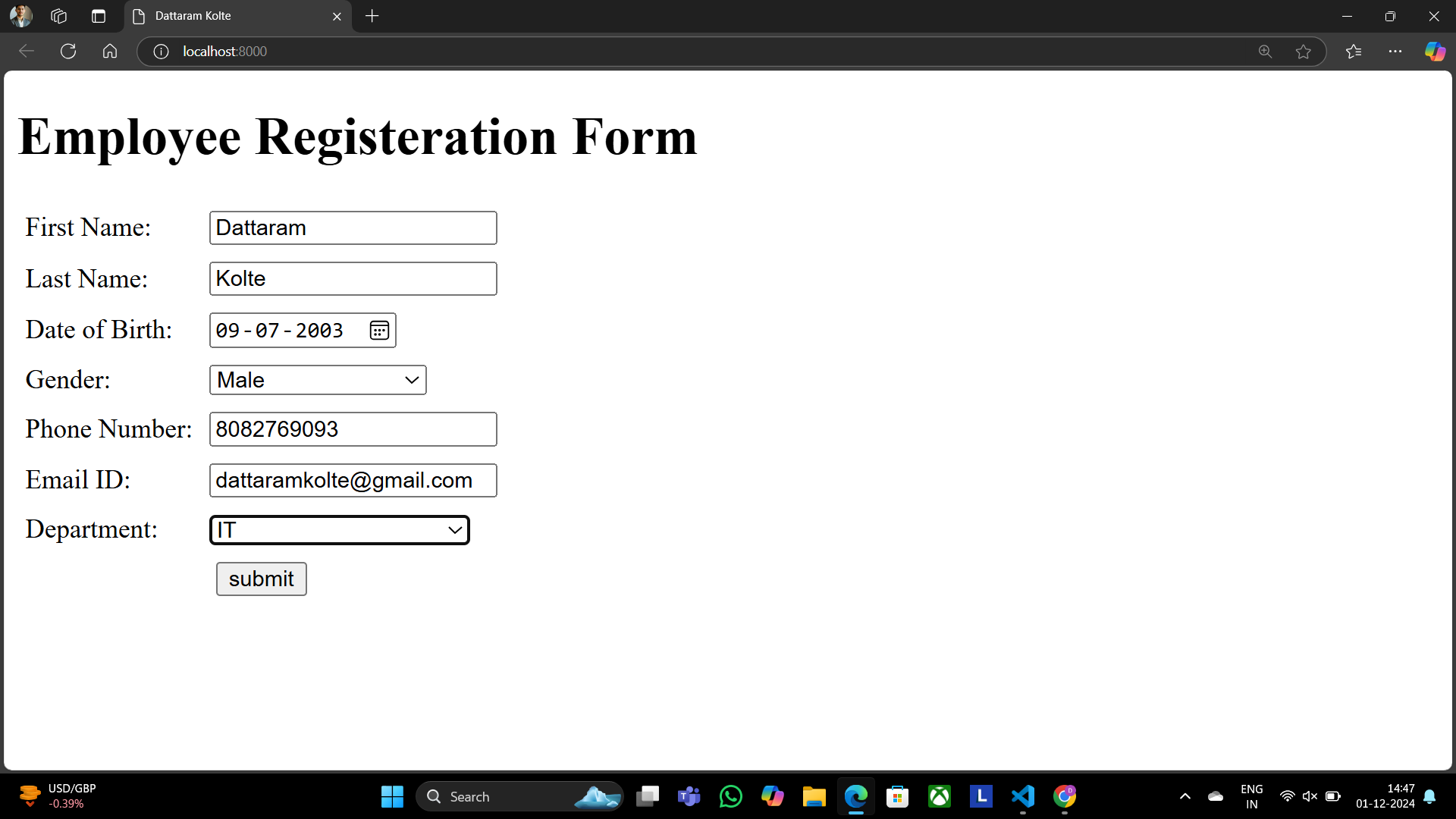
server.listen(8000, () => {

console.log("Server is running at http://localhost:8000")

})

**OUTPUT:**

****

****

**21) Write a program to handle request URLs between various HTML pages using HTTP Server**

**HttpServer1.js**

var http = require('http');

var server = http.createServer(function(req, res) {

if (req.url == '/') {

res.writeHead(200, {

'content-type': 'text/html'

});

res.write('<html></head><body>');

res.write('<style> ul li{display: inline-block; float: right; height: 40px;} ul li a{padding: 20px; background:orange; color: white;}</style>');

res.write('<div><h1>First WebPage using http Server</h1></div><div><ul><li><a href="/admin">Contact Admin</a></li><li><a href="/student">Student</a></li><li><a href="/home">Home</a></li></ul></div></div>');

res.write('<div style="background: white; padding: 20px;"><h2>Start Page</h2><p>This is my first webpage!</p><p>Hi everyone</p></div></body></html>');

res.end();

} else if (req.url == '/home') {

res.writeHead(200, {

'content-type': 'text/html'

});

res.write('<html><head><style>body{padding-left: 43px; padding-right:43px; background-color:lightyellow;} </style></head><body><p><h1>This is home page</h1></p><h1>Dattaram Kolte</h1><h3>This page is a brief insight to who I am.</h3>');

res.write('<nav style="background-color:white; text- align:center;"><ul><li><a href="/">Start Page</a></li><li><a href="/student">Student</a></li><li><a href="/admin">Admin</a></li></ul></nav></body></html>');

res.end();

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

res.writeHead(200, {

'content-type': 'text/html'

});

res.write('<div style="display: inline-block; float: right; height: 40px; padding: 20px;"><ul><li><a href="/home">Home</a></li><li><a href="/">Start Page</a></li> <li><a href="/admin">Contact Admin</a></li></ul></div>');

res.write('<html><head><style>body{background- color:pink;}</style><title>Form</title></head><body bgcolor="White" ><h1 align="center">Student Page Form</h1>');

res.write('<form action="url" method="post"><fieldset><legend>Personal Imformation</legend>');

res.write('<lable><Strong>Student Name</strong></lable><br/><input type="text" name="Student Name" placeholder="Enter Your Name" /><br/>');

res.write('<lable><Strong>Email</strong></lable><br/><input type="email" name="eamil" placeholder="Enter Your Email Address" /></br>');

res.write('<lable><Strong>Password</strong></lable><br/>');

res.write('<input type="password" name="Password" placeholder="Enter Your Password" /></br><lable><Strong>Gender</strong></lable><br/>');

res.write('<input type="Radio" name="Gender" value="Male" />Male<input type="Radio" name="Gender" value="FeMale" />FeMale<br/>');

res.write('<lable><Strong>Hobbies</strong></lable><br/>');

res.write('<input type="checkbox" name="Hobbies" value="Playing Sports" />Playing Sports<br/>');

res.write('<input type="checkbox" name="Hobbies" value="Listening Music" />Listening Music<br/>');

res.write('<input type="checkbox" name="Hobbies" value="Traveling"/>Traveling<br/>');

res.write('<input type="checkbox" name="Hobbies" value="Reading Books"/>Reading Books<br/>');

res.write('<lable><Strong>Select Your City</strong></lable><select name="City">');

res.write('<option value="Mumbai">Mumbai</option><option value="Gujrat">Gujrat</option><option value="Pune">Pune</option>');

res.write(' <option value="Thane">Thane</option></select></br><input type="submit" onclick=alert("Thanks!") name="submit" value="Submit"/></form>');

res.end();

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

res.writeHead(200, {

'content-type': 'text/html'

});

res.write('<style>ul li{display: inline-block; float: right; height:40px;} ul li a{padding: 20px; background:orange; color: white;}</style>');

res.write('<div><ul><li><a href="/admin">Contact Admin</a></li><li><a href="/student">Student</a></li><li><a href="/home">Home</a></li></ul></div></div><br><br>');

res.write('<html><head><style>legend{text-align:center;} body{background-color:faf89a;border: 5px solid darkred;} form{display: inline- block; float: center; padding: 20px;} ');

res.write('border-radius:4px; padding:40px 5px; max- width:100%;}</style></head>');

res.write('<legend><h1><u>Admin Login</u></h1></legend>');

res.write('<form action="#" method="POST" autocomplete="off">');

res.write('<div class="input\_field"><h3>Username</h3></div><div class="input\_field"><input type="text" ');

res.write('name="userid" placeholder="Username" required/></div>');

res.write('<div class="input\_field"><h3>Password</h3></div><div class="input\_field"><input type="Password"');

res.write('name="pword" placeholder="Password" required/></div><p>');

res.write('<style>button{border:none; border-radius:5px; text-align:center; padding:15px 15px; background- color:lavender;<div></div></style>');

res.write('<button onclick=alert("SUCESS")>LOGIN NOW</button></form>');

res.end();

} else {

res.end('Invalid request');

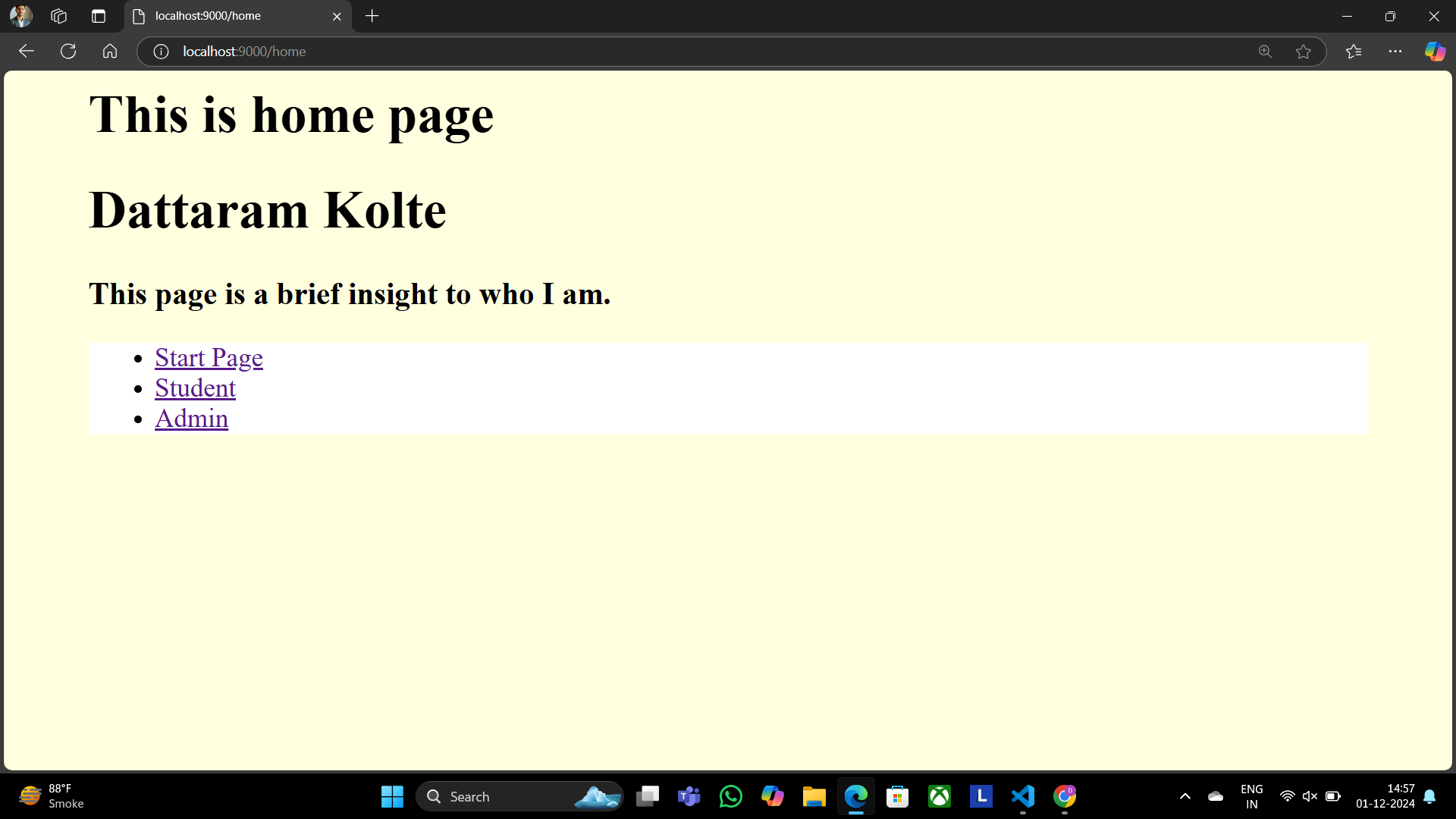
}

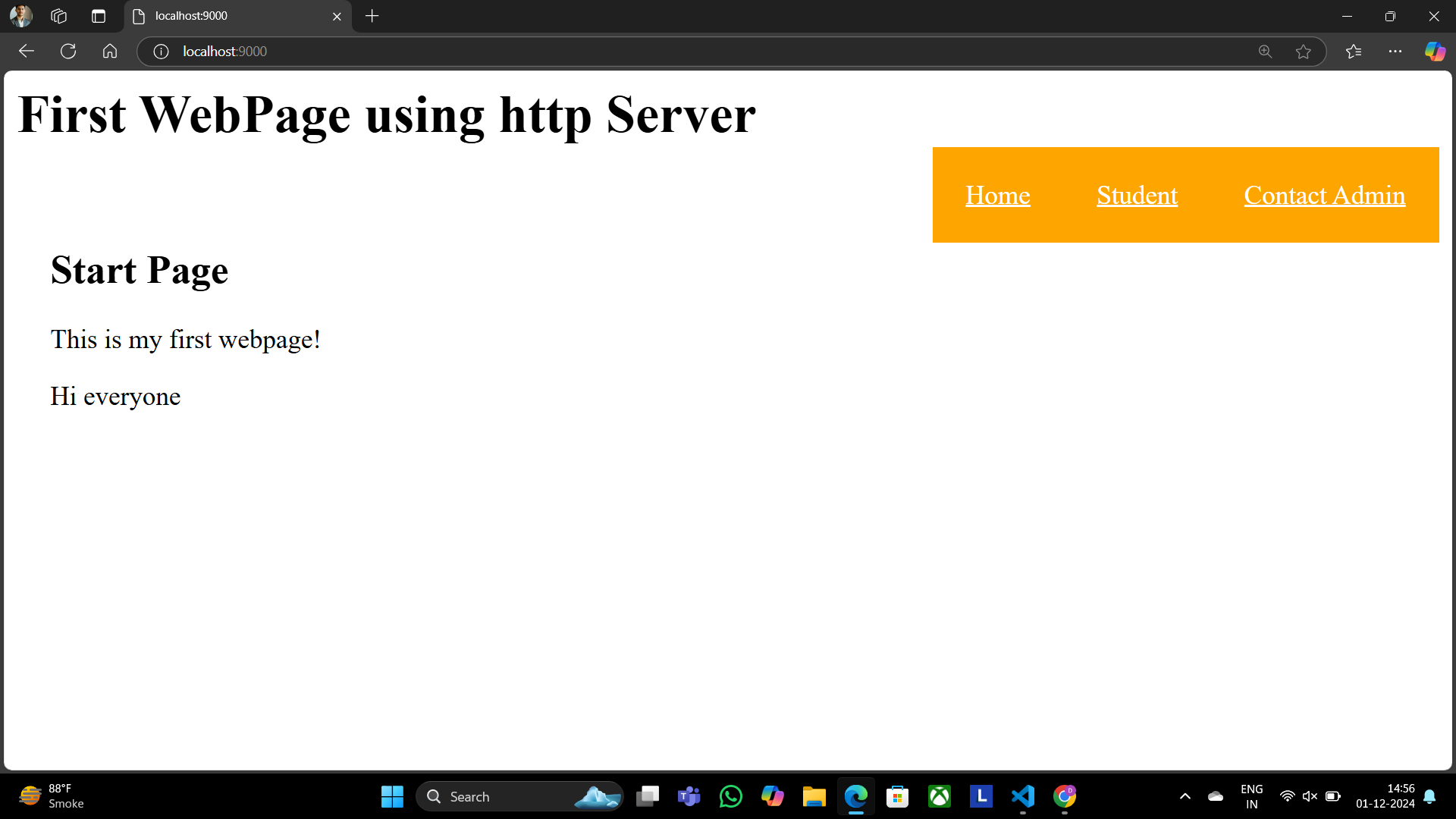
});

server.listen(9000);

console.log('Server is running at <http://localhost:9000>');

**OUTPUT:**

****

****

**22) Write a program to implement the database in node.js**

**Database1.js**

const mysql = require("mysql")

const con = mysql.createConnection({

host:"localhost",

user:"root",

password:"",

})

con.connect(function(err) {

if(err) throw err;

console.log("connected\nhttp://localhost/phpmyadmin/index.php?route=/database/structure&db=conference")

//Creating the Database

var query = "CREATE DATABASE IF NOT EXISTS CONFERENCE;"

con.query(query, function(err, result){

if (err) throw err;

console.log("Database Created.")

})

//Using the Database

query = "USE CONFERENCE;"

con.query(query, function(err, result){

console.log("Using the CONFERENCE.")

})

//Creating the Table

query = `CREATE TABLE IF NOT EXISTS conf(

id int auto\_increment primary key,

name varchar(20),

prof varchar(20),

qual varchar(20),

title varchar(50),

org varchar(50)

);`

con.query(query, function(err, result){

if (err) throw err;

console.log("Table created.")

})

// //Inserting the values

const records = [

['Alice', 'Professor', 'PhD', 'AI Research', 'University A'],

['Bob', 'Associate Professor', 'MSc', 'Machine Learning', 'University B'],

['Charlie', 'Lecturer', 'MSc', 'Data Science Innovations', 'University C'],

['David', 'Professor', 'PhD', 'Cybersecurity Research', 'University A'],

['Eva', 'Senior Researcher', 'PhD', 'Robotics and AI', 'University B']

];

query = 'INSERT INTO conf (name, prof, qual, title, org) VALUES ?'

con.query(query, [records], function (err, result) {

if (err) throw err

console.log(result.affectedRows + " records inserted.")

})

//Retrieving the data

query = "SELECT \* from conf;"

con.query(query, function(err, result){

if(err) throw err;

console.log(result)

})

//Updating a row

query = `UPDATE conf set name = "Dattaram" where id = 4;`

con.query(query, function(err, result){

if(err) throw err;

console.log("Values updated.")

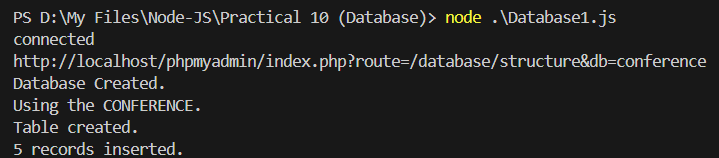
})

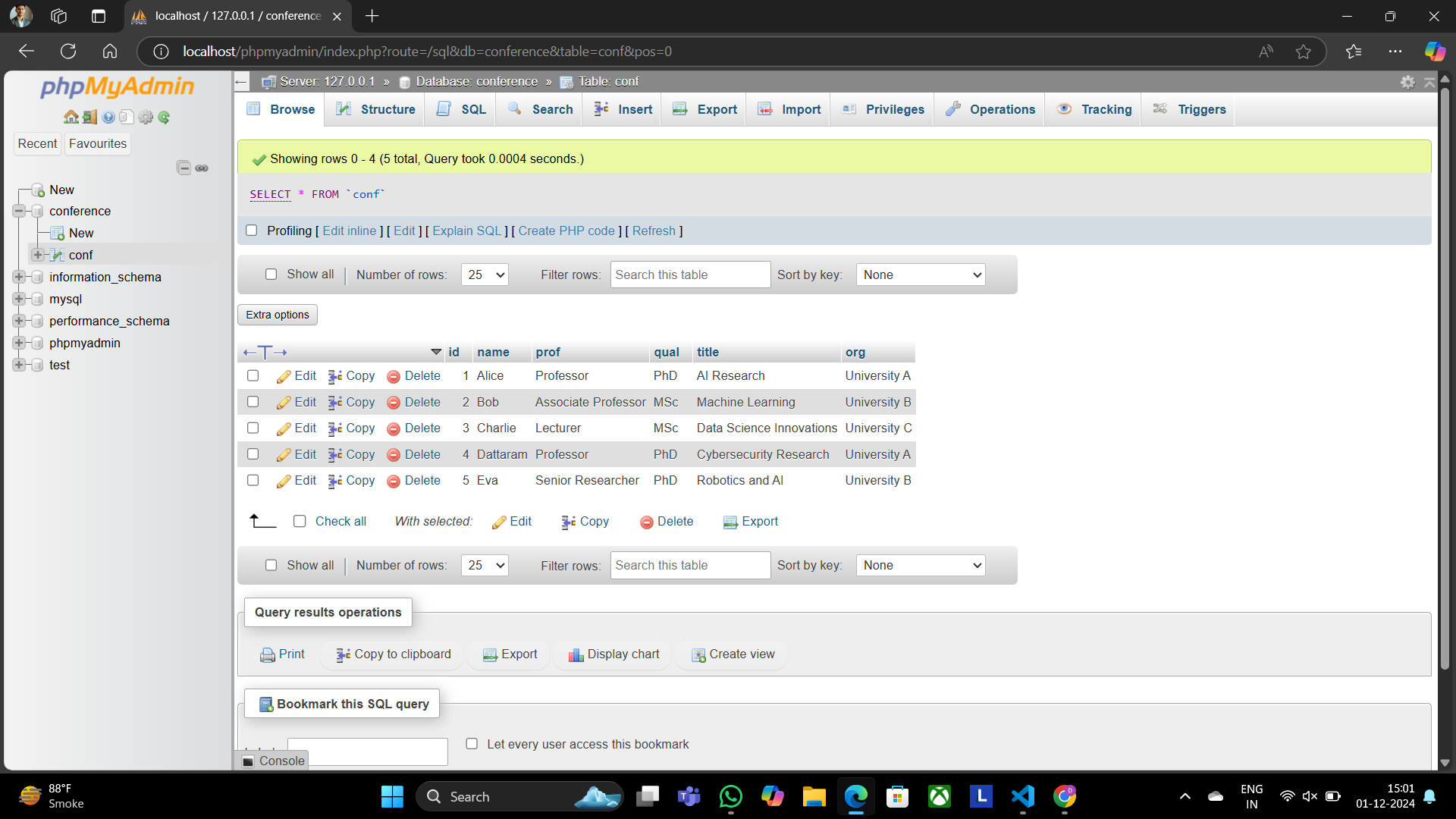
//Closing the connection

con.end

})

**OUTPUT:**

****

****

**23)** Write a program to Display Hello World using ReactJS

**App.js**

import './App.css';

function App() {

return (

<div className="App">

<p>

<h1>Hello Dattaram, Welcome to your First React Page</h1>

</p>

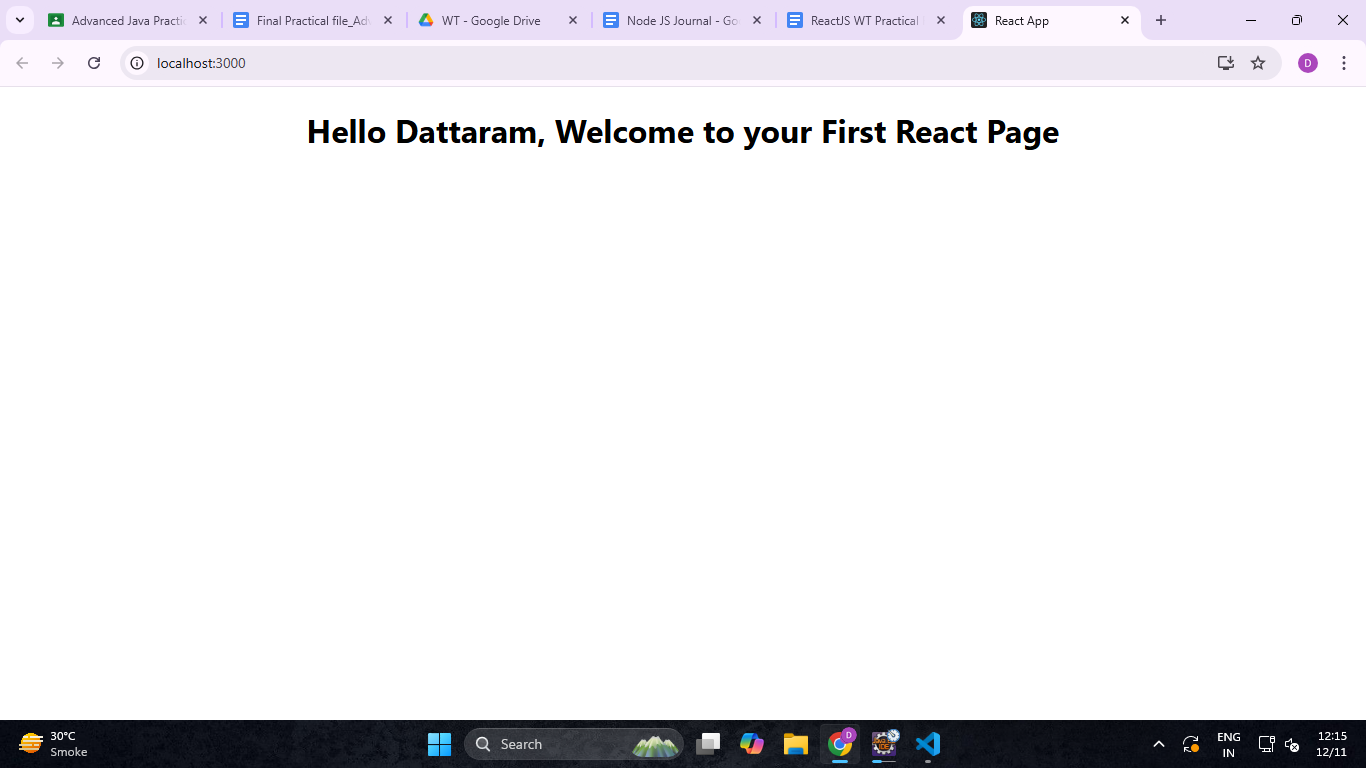
</div>

);

}

export default App;

**OUTPUT:**

****

**24)** Create an application in ReactJS to implement component life cycle

**ComponentLifeCycle.js**

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

const LifecycleComponent = () => {

const [count, setCount] = useState(0);

const [message, setMessage] = useState('Hello, World!');

// Equivalent to componentDidMount, componentDidUpdate, componentWillUnmount

useEffect(() => {

// This function will run once when the component mounts (initial render)

console.log('Component mounted!');

// This return function acts like componentWillUnmount

return () => {

console.log('Component will unmount!');

};

}, []); // Empty dependency array means this runs only once on mount

useEffect(() => {

// This function will run every time the count changes (update phase)

console.log(`Count updated to: ${count}`);

}, [count]); // This runs only when `count` changes

const handleClick = () => {

setCount(count + 1); // Increment count

};

const handleMessageChange = () => {

setMessage('Message has been changed!');

};

return (

<div>

<h1>React Component Lifecycle Example</h1>

<p>Message: {message}</p>

<p>Count: {count}</p>

<button onClick={handleClick}>Increment Count</button>

<button onClick={handleMessageChange}>Change Message</button>

</div>

);

};

export default LifecycleComponent;

**App.js**

import './App.css';

import LifecycleComponent from './ComponentLifeCycle';

function App() {

return (

<div>

<LifecycleComponent />

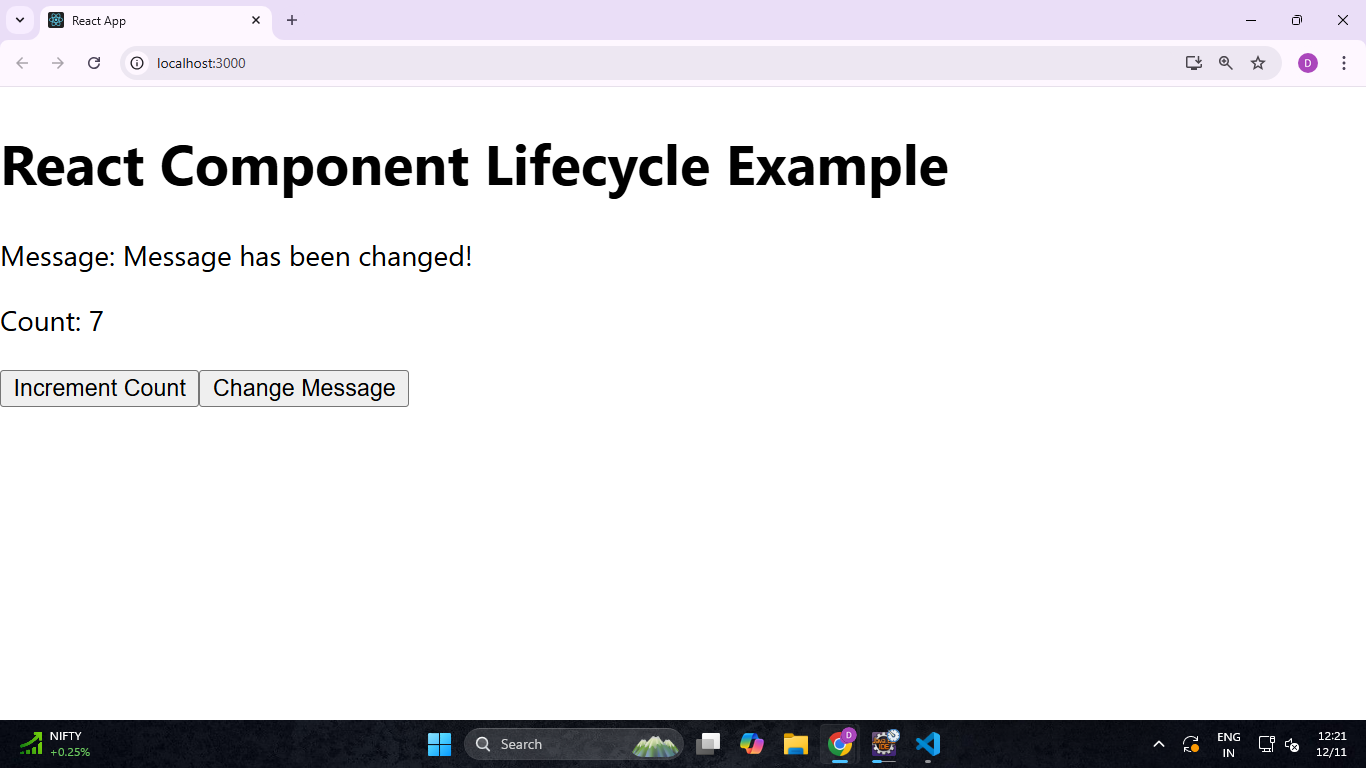
</div>

);

}

export default App;

**OUTPUT:**

****

**25)** Create an application to implement class and functional component in ReactJS

**MyClassComponent.js**

import React, {Component} from 'react';

class MyClassComponent extends Component {

constructor(props){

super(props);

this.state={

message :'Hello , Welcome to React Class Component ',

counter: 0,

};

}

incrementCounter=()=> {

this.setState((prevState)=> ({

counter:prevState.counter+1,

}));

};

render(){

return(

<div style={{textAlign:'center' ,marginTop:'50px'}}>

<h1>{this.state.message}</h1>

<p>Counter :{this.state.counter}</p>

<button onClick={this.incrementCounter} style={{padding:'10 px 20 px', fontSize:'16px'}}>

IncrementCounter

</button>

</div>

);

}

}

export default MyClassComponent;

**App.js**

import './App.css';

import MyClassComponent from './MyClassComponent';

function App() {

return (

<div>

<MyClassComponent />

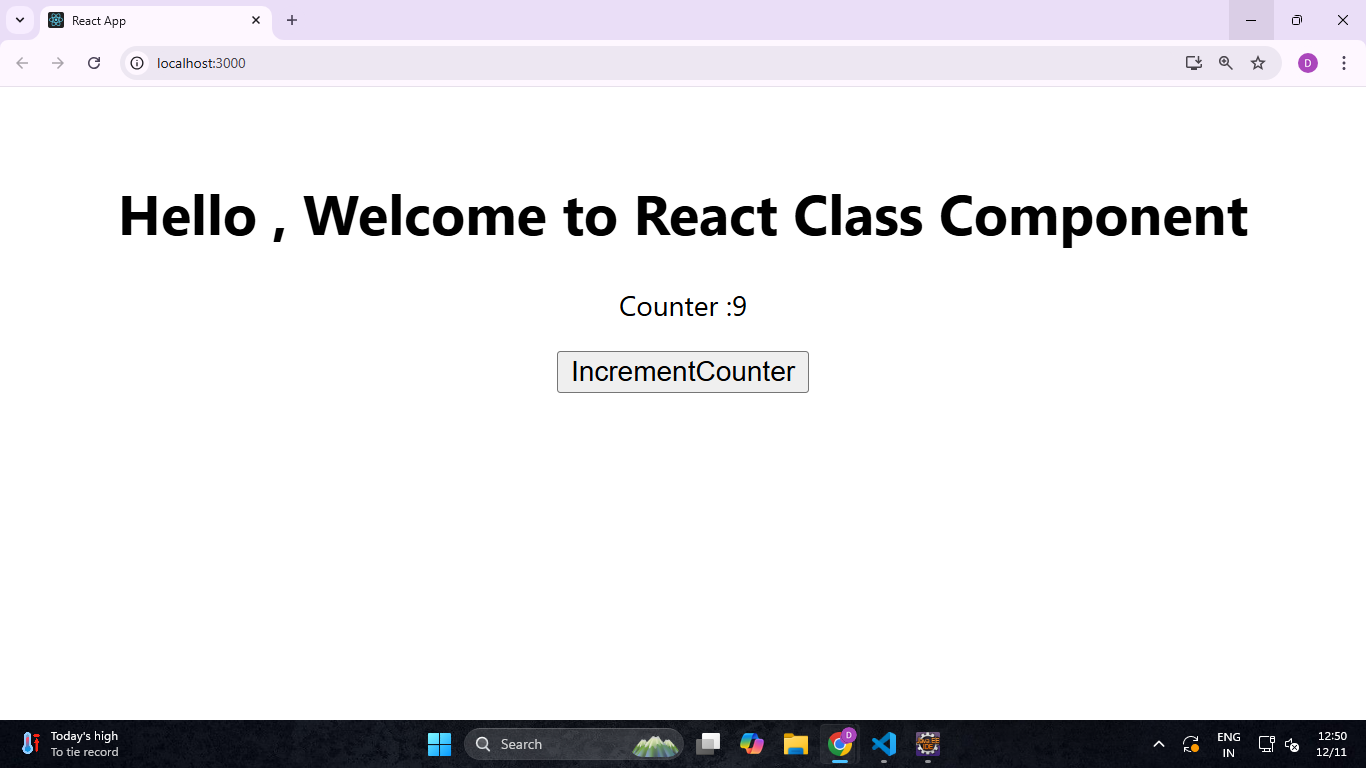
</div>

);

}

export default App;

**OUTPUT:**

****

**26)** Create an application to implement functional component in ReactJS

**FunctionalComponent.js**

import React, {useState} from 'react';

const AddTwoNumbers=()=> {

const [num1, setNum1] = useState('');

const [num2, setNum2] = useState('');

const [sum, setSum] = useState(null);

const handleAddition= () =>{

const result =parseFloat(num1) +parseFloat(num2);

setSum(result);

};

return (

<div style={{textAlign:'center',marginTop:'50px'}}>

<h1> Add Two Numbers</h1>

<div style ={{ marginBottom:'20px'}}>

<input type="number" placeholder='Enter first number'

value={num1}

onChange={(e) => setNum1(e.target.value)}

style={{marginRight:'10 px' ,padding:'5 px'}} />

<input type="number" placeholder='Enter second number'

value={num2}

onChange={(e) => setNum2(e.target.value)}

style={{marginRight:'10 px' ,padding:'5 px'}} />

<button onClick={handleAddition} style={{padding: '5px 10 px'}}>Add</button>

{sum!==null && <h2> Result : {sum}</h2>}

</div>

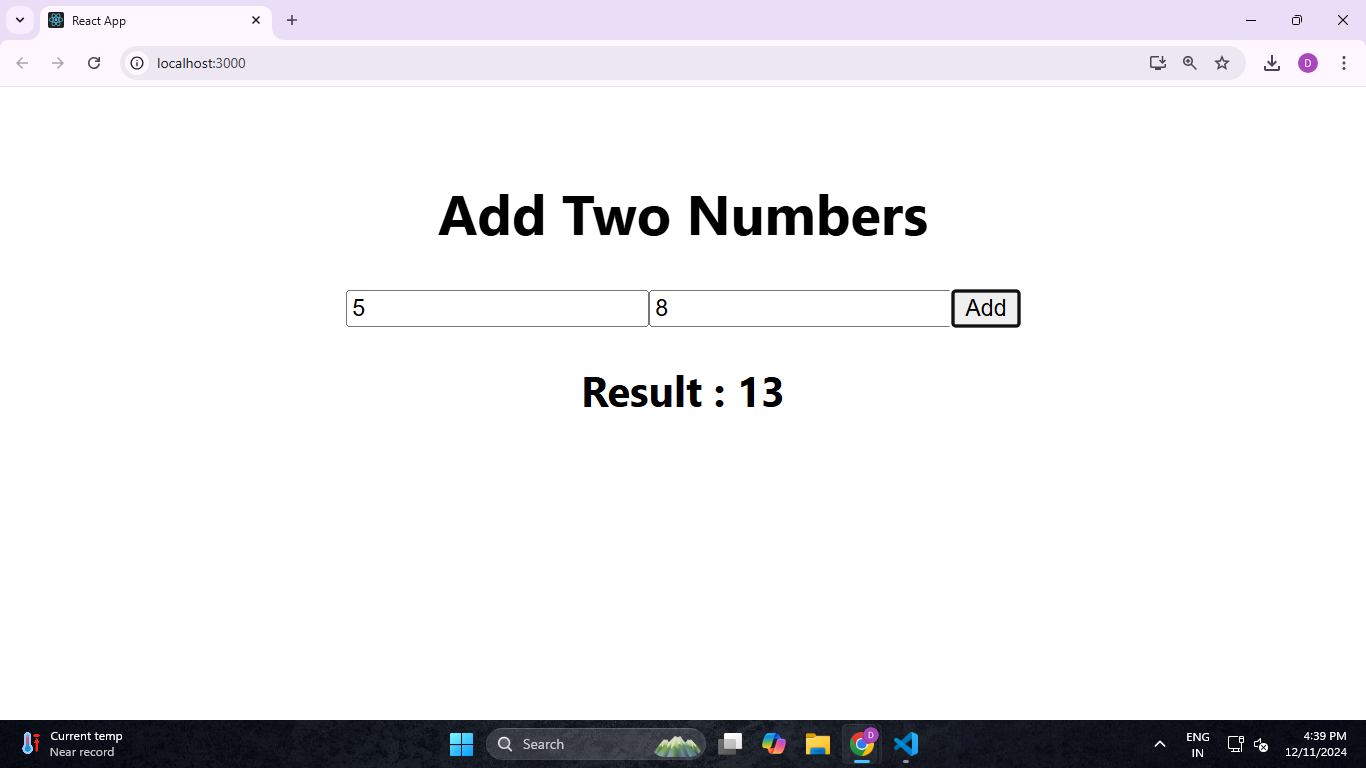
</div>

);

};

export default AddTwoNumbers;

**OUTPUT:**

****

**27)** Create an application in ReactJS import and export the files (components)

**FileUploader.js**

import React, { useState } from "react";

const FileUploader = () => {

const [fileContent, setFileContent] = useState(""); // Declare useState correctly

const handleFileUpload = (e) => {

const file = e.target.files[0];

const reader = new FileReader();

reader.onload = (event) => {

setFileContent(event.target.result); // Correctly set file content

};

if (file) reader.readAsText(file); // Read file as text

};

return (

<div>

<h3>Upload a File</h3>

<input type="file" onChange={handleFileUpload} />

{fileContent && ( // Conditionally render file content

<div>

<h4>File Content:</h4>

<textarea value={fileContent} readOnly rows="10" cols="50" />

</div>

)}

</div>

);

};

export default FileUploader;

**FileDownloader.js**

import React from "react";

const FileDownloader = () => {

const handleDownload = () => {

const content = "This is some sample text for the file.";

const blob = new Blob([content], { type: "text/plain" });

const url = URL.createObjectURL(blob);

const link = document.createElement("a");

link.href = url;

link.download = "sample.txt";

link.click();

URL.revokeObjectURL(url);

};

return (

<div>

<h3>Download a File </h3>

<button onClick={handleDownload}>Download</button>

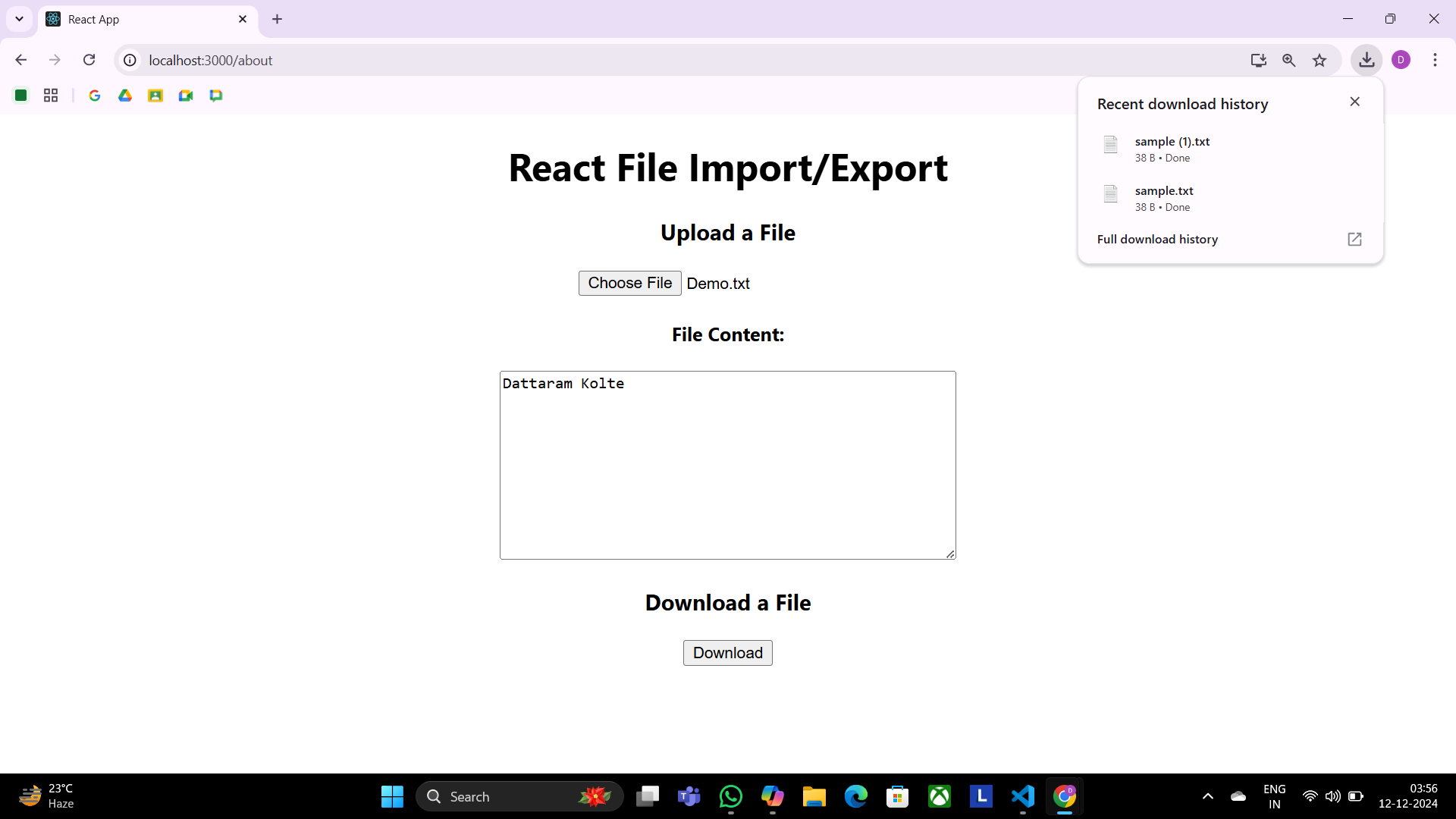
</div>

);

};

export default FileDownloader;

**OUTPUT:**

****

**28)** Create an application to increment and decrement counter using state

**Conuter.js**

import './App.css';

import React, { useState } from 'react';

const Counter = () => {

const [count, setCount] = useState(0);

return (

<div style={{textAlign:"center"}}>

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

<button onClick={() => setCount(count + 1)}>Increment</button>

<button onClick={() => setCount(count - 1)}>Decrement</button>

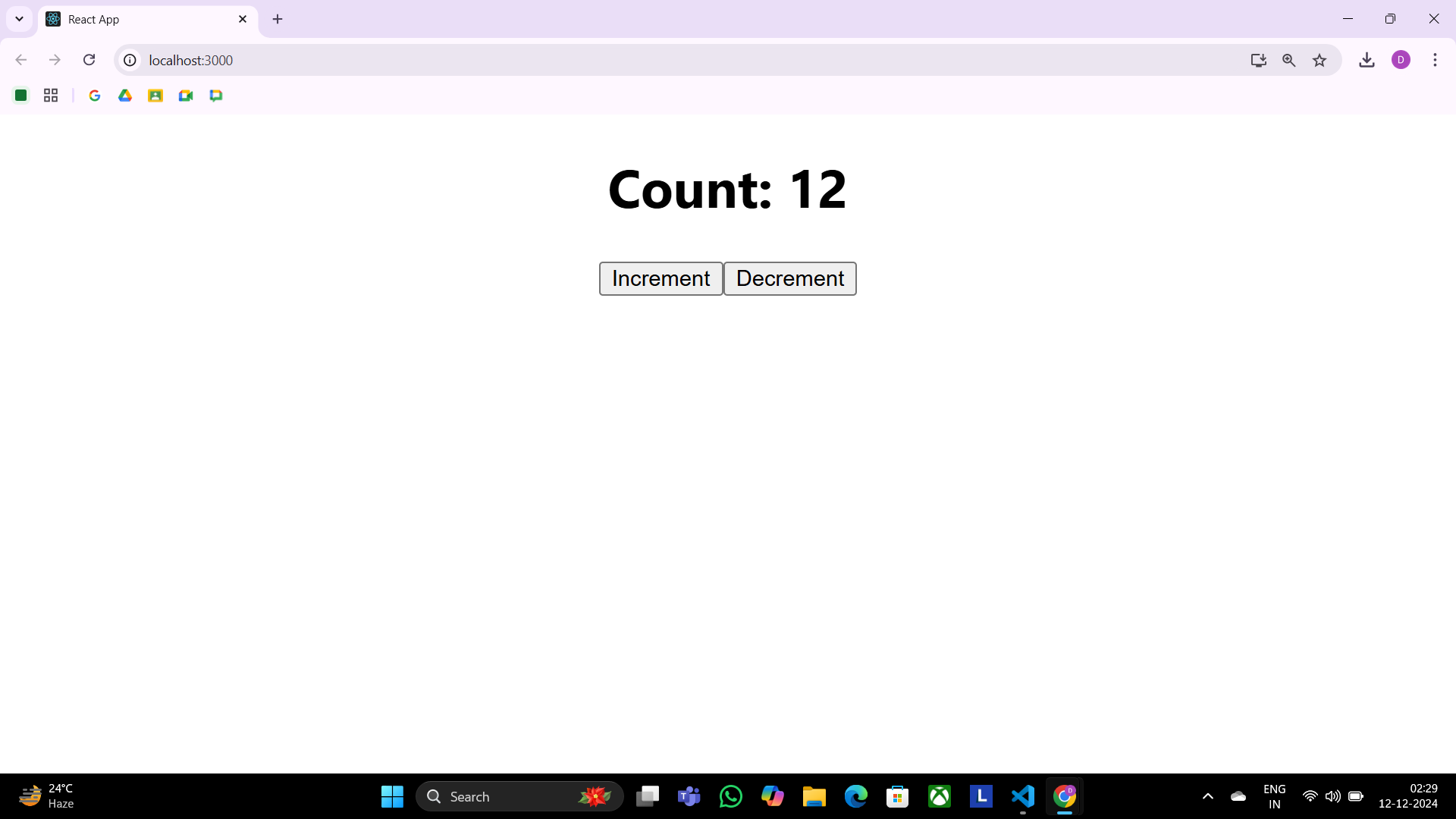
</div>

);

};

export default Counter;

**OUTPUT:**

****

**29)** Create an application to display your name using prop

**App.js**

const Greeting = ({ name }) => {

return <h1>Hello, {name}!</h1>;

};

const App = () => {

return (

<div>

<Greeting name="Dattaram" />

<Greeting name="Kolte" />

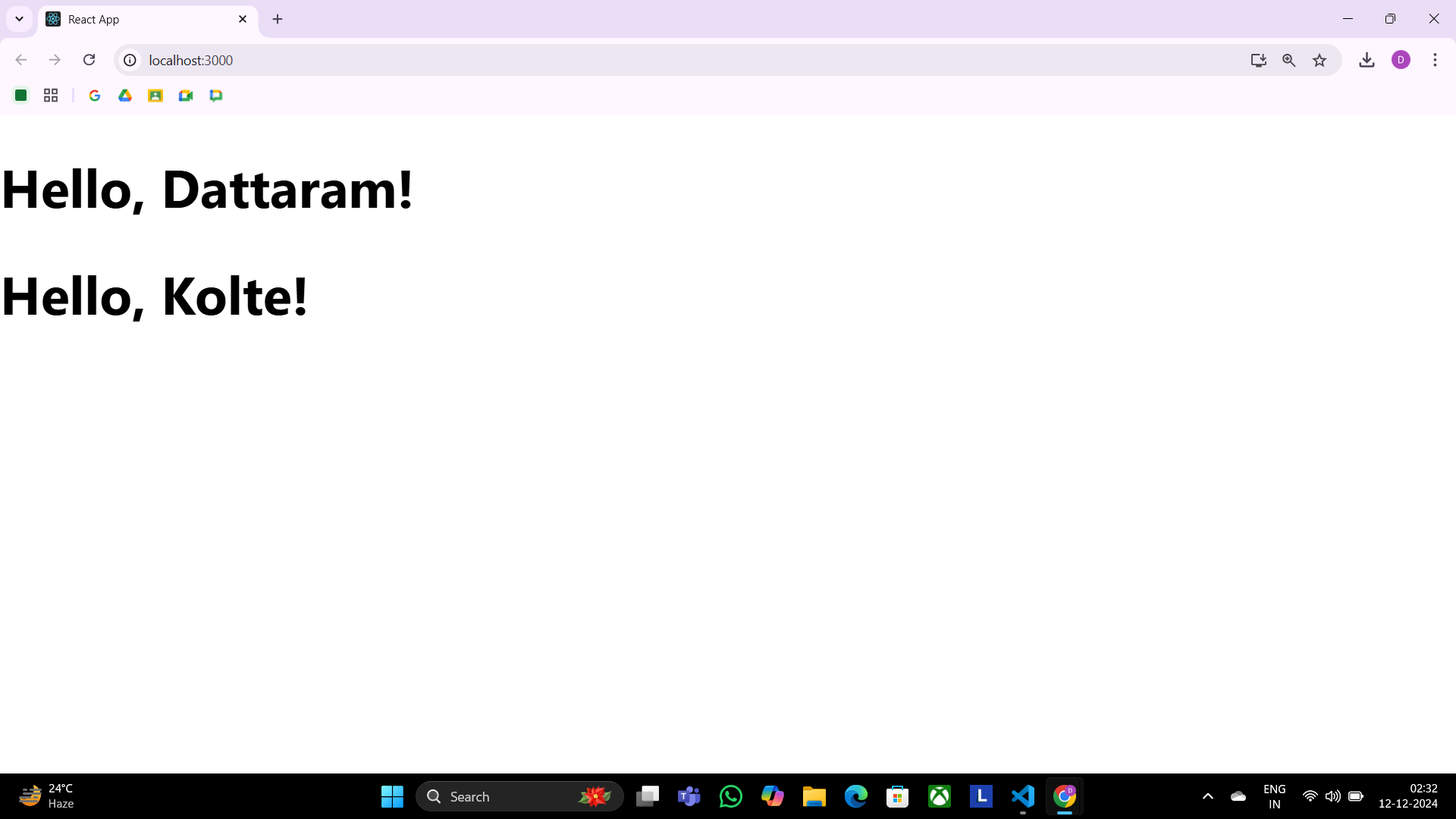
</div>

);

};

export default App;

**OUTPUT:**

****

**30)** Create an application to implement To-Do task

**TaskList.js**

import React from 'react';

const TaskList = ({ tasks }) => {

return (

<div style={{ marginTop: '20px' }}>

<h2>Your Tasks</h2>

{tasks.length === 0 ? (

<p>No tasks added yet.</p>

) : (

<ul>

{tasks.map((task, index) => (

<li key={index} style={{ marginBottom: '10px' }}>

{task}

</li>

))}

</ul>

)}

</div>

);

};

export default TaskList;

**App.js**

import React, { useState } from 'react';

import TaskList from './TaskList';

const App = () => {

const [tasks, setTasks] = useState([]); // State to manage tasks

const [taskInput, setTaskInput] = useState(''); // State for input field

const handleAddTask = () => {

if (taskInput.trim() !== '') {

setTasks([...tasks, taskInput]); // Add new task to the list

setTaskInput(''); // Clear input field

}

};

return (

<div style={{ padding: '20px' }}>

<h1>To-Do List</h1>

<div>

<input

type="text"

value={taskInput}

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

placeholder="Enter a new task"

style={{ padding: '10px', width: '300px', marginRight: '10px' }}

/>

<button onClick={handleAddTask} style={{ padding: '10px' }}>

Add Task

</button>

</div>

<TaskList tasks={tasks} />

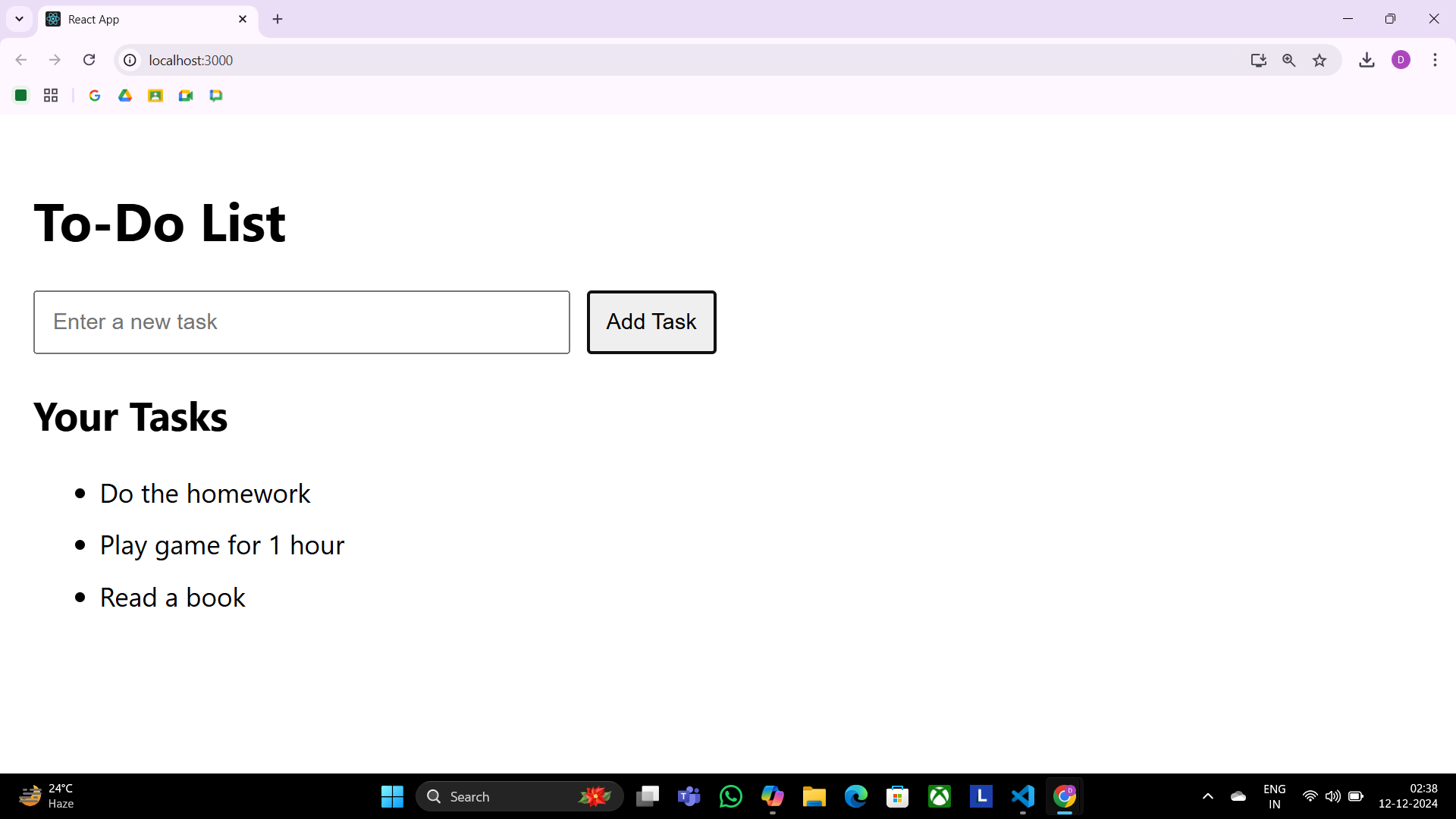
</div>

);

};

export default App

**OUTPUT:**

****

**31)** Create an application in ReactJS to use DOM events- onChange

**OnChangeEvent.js**

import React , {useState} from "react";

function ToggleMessage() {

const[isChecked , setIsChecked]= useState(false); // State to track checkbox toggle

const handleCheckboxChange= (event) => {

setIsChecked(event.target.checked); // Update state when checkbox is toggled

};

return (

<div style={{margin:"20 px", textAlign:"center"}}>

<h3> Show/Hide Message</h3>

<label>

<input type="checkbox" onChange={handleCheckboxChange} // Event handler for checkbox style={{marginRight:"10 px"}} /> </label>

<div style={{marginTop:"20 px"}}>

{isChecked && <p style={{color:"green"}}> Hello , this is Dattaram Kolte !</p>}

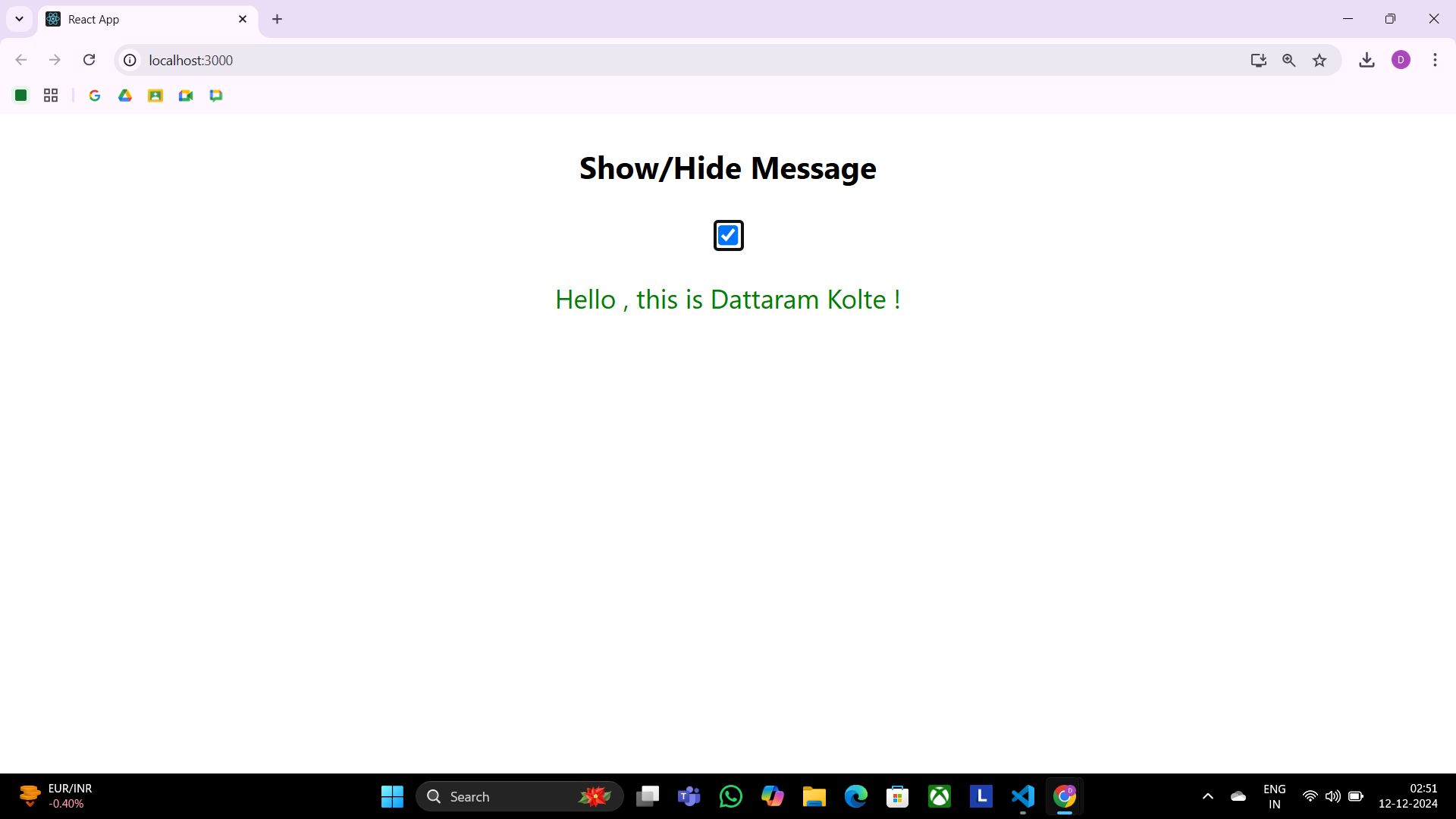
</div> </div>

)

}

export default ToggleMessage;

**OUTPUT:**

****

**32)** Write a program that tracks the changes in an input field and displays the entered text in real-time using onChange DOM event

**InputTracker.js**

import React, { useState } from "react”;

function InputTracker() {

const [text, setText] = useState("");

const handleChange = (event) => {

setText(event.target.value);

};

return (

<div style={{ margin: "20px" }}>

<h3>Input Field Change Tracker</h3>

<input type="text" placeholder="Type something here.." value={text} onChange={handleChange} style={{ padding: "8px", border: "1px solid #ccc", borderRadius: "4px", width: "300px", }} />

<p style={{ marginTop: "10px" }}>You typed: {text}</p>

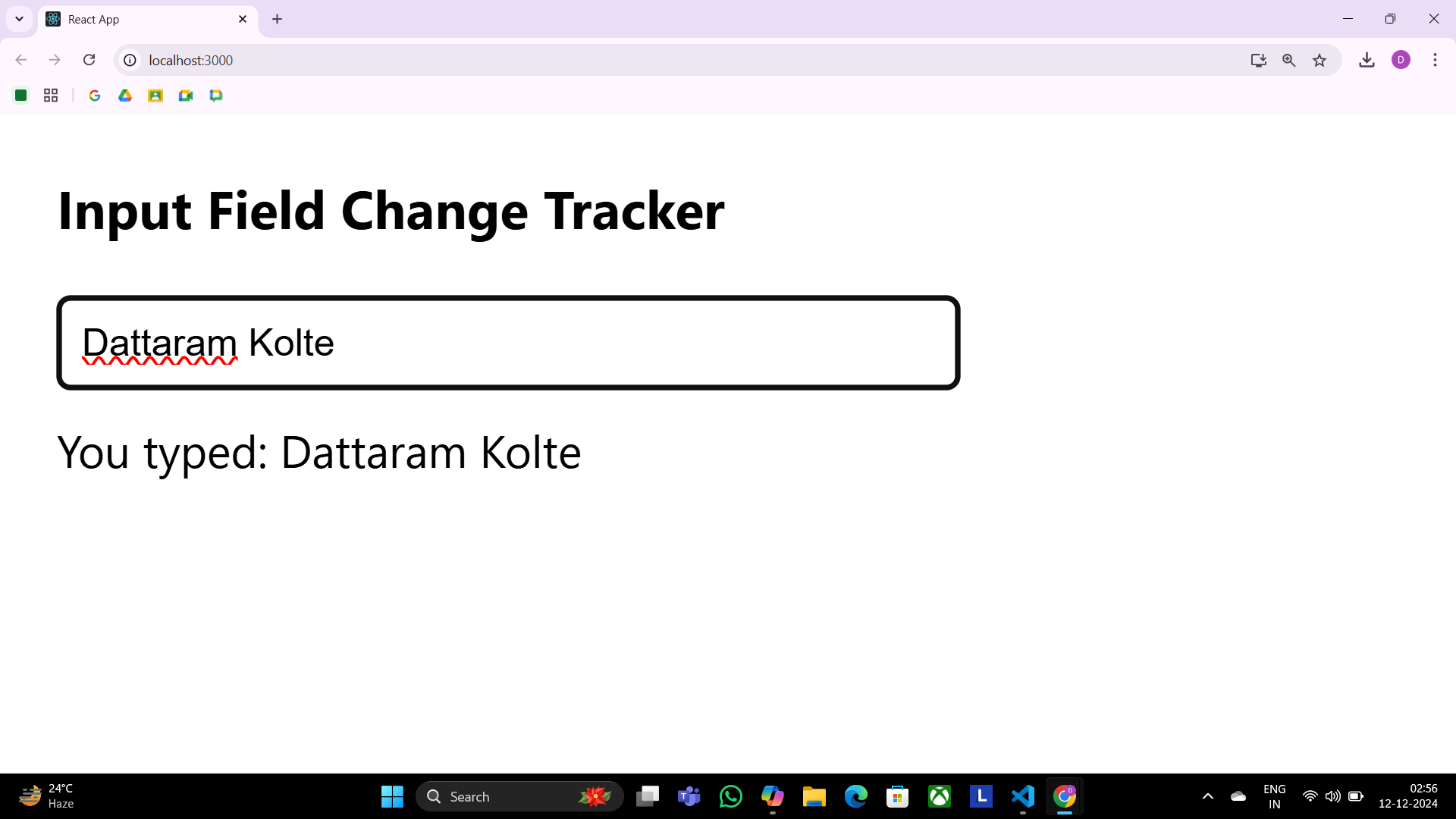
</div>

);

}

export default InputTracker;

**OUTPUT:**

****

**33)** Create an application in ReactJS to use DOM events- onKeyUp

**KeyUp.js**

import React, { useState } from "react";

function KeyCodeDisplay() {

const [keyCode, setKeyCode] = useState("");

const handleKeyUp = (e) => {

setKeyCode(`Key Code: ${e.keyCode}`);

};

return (

<div>

<input type="text" onKeyUp={handleKeyUp} placeholder="Press a key.." />

<p>{keyCode}</p>

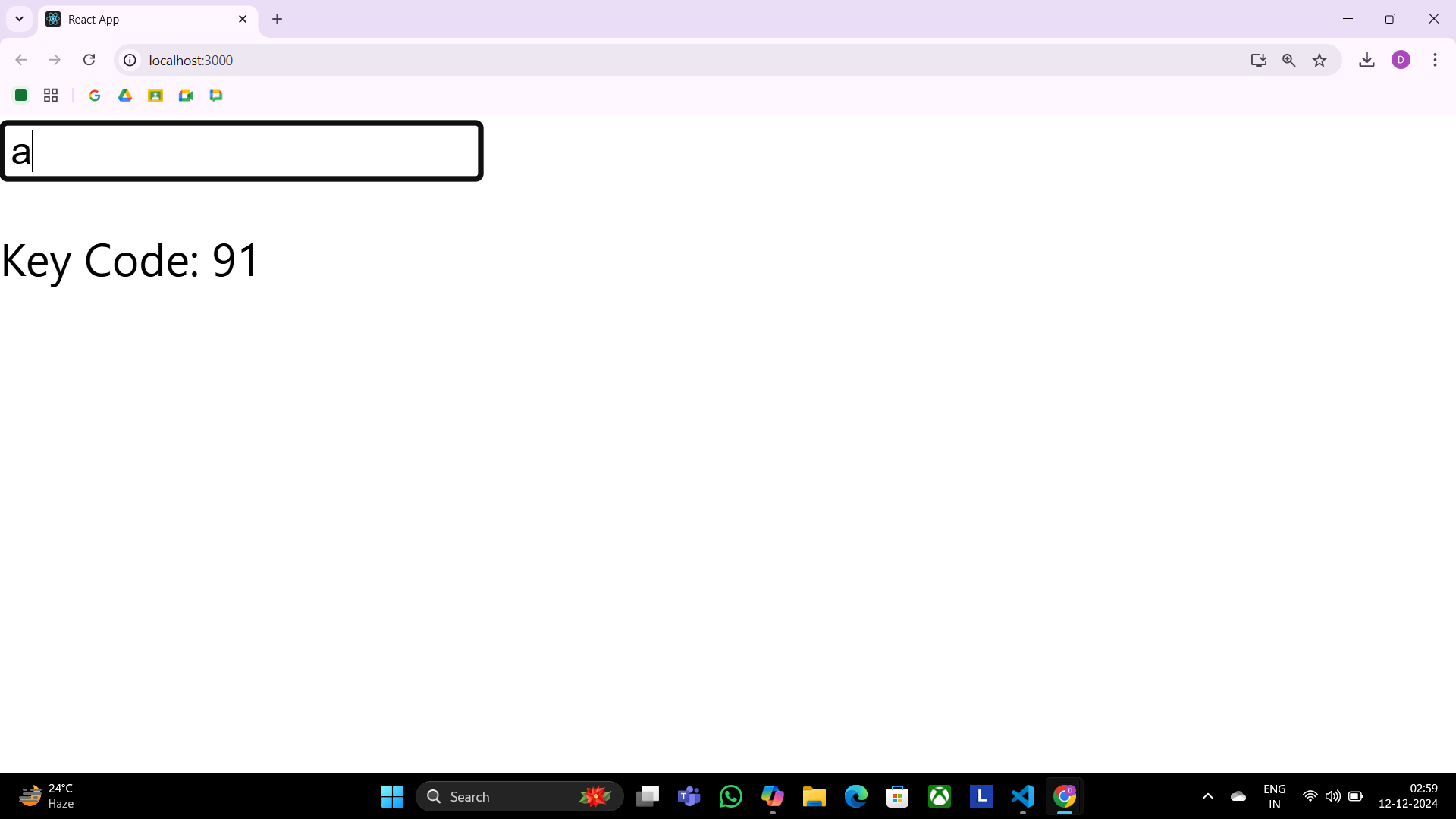
</div>

);

}

export default KeyCodeDisplay;

**OUTPUT:**

****

**34)** Write a Program to Counts words as they are typed using onKeyUp event

**CountOnKeyUp.js**

import './App.css';

import WordCount from './Dom Event/CountOnKeyUp';

function App() {

return (

<div>

<WordCount/ >

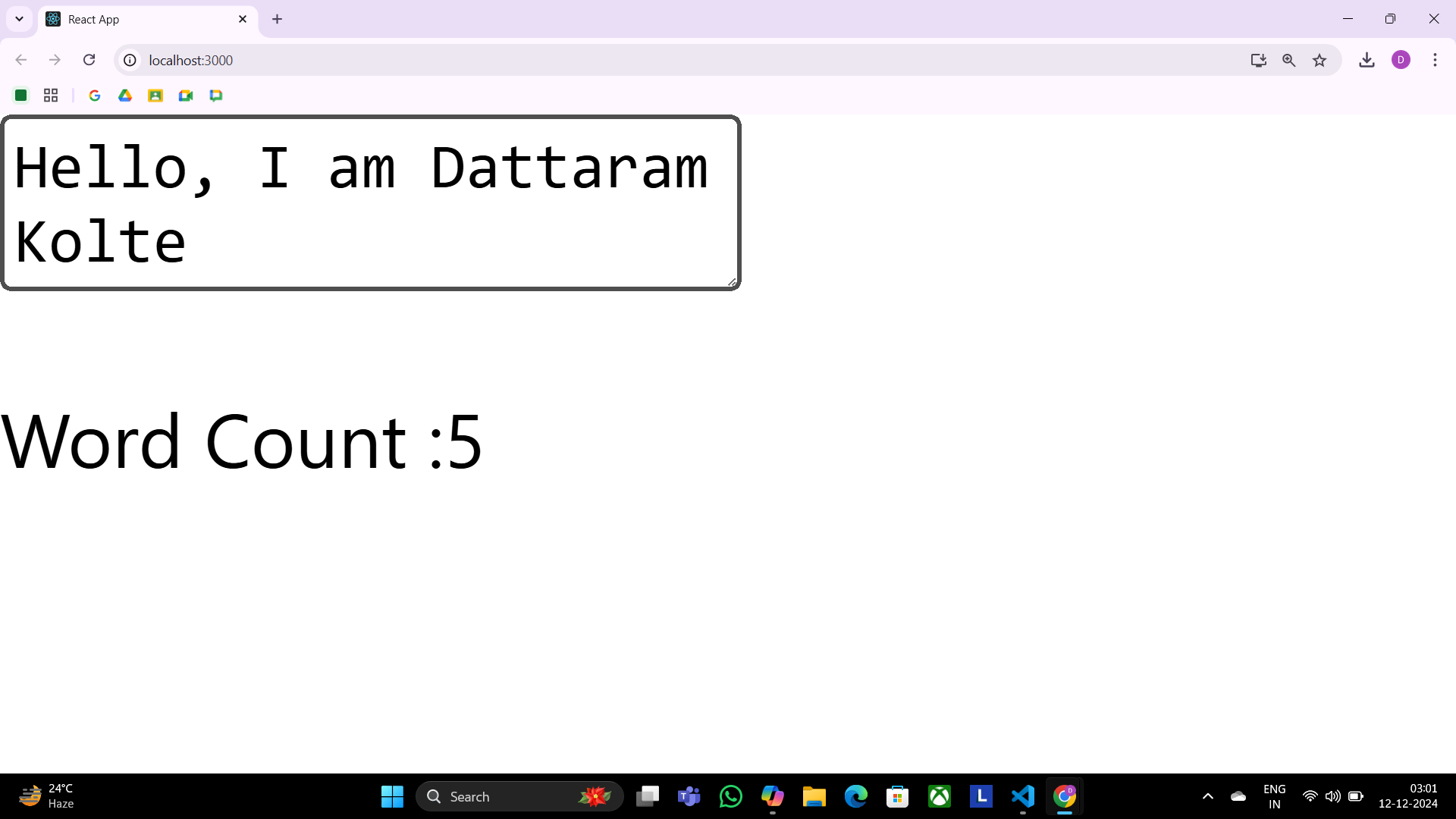
</div>

);

}

export default App;

**OUTPUT:**

****

**35)** Write a Program to implement validation logic for an email field using onBlur event

**OnBlur.js**

import React, { useState } from "react";

function ValidateOnBlur() {

const [error, setError] = useState("");

const handleBlur = (e) => {

const email = e.target.value;

if (!email.includes("@")) {

setError("Invalid email");

} else {

setError("");

}

};

return (

<div>

<input type="text" onBlur={handleBlur} placeholder="Enter your email" />

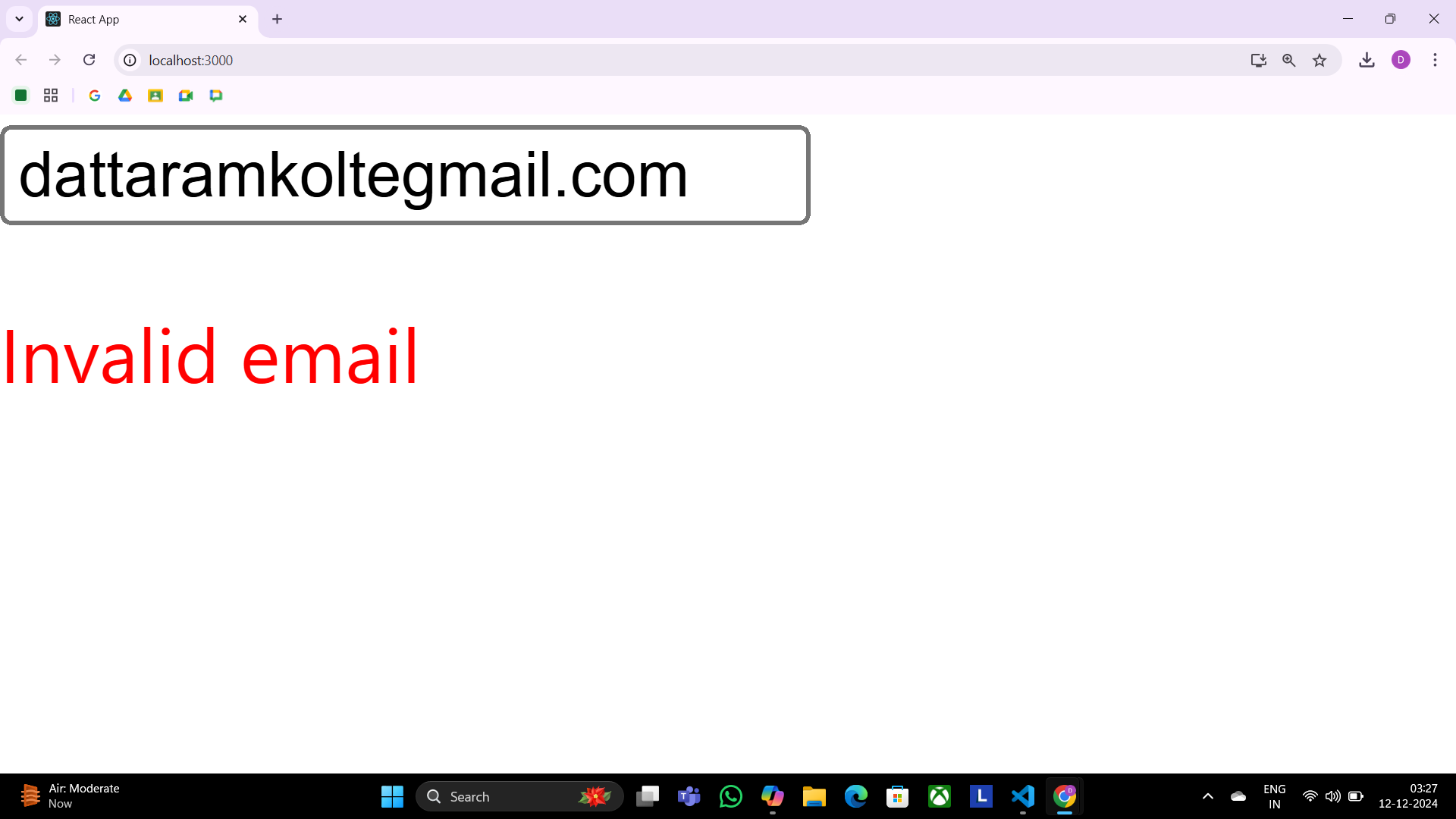
{error && <p style={{ color: "red" }}>{error}</p>} </div>

);

}

export default ValidateOnBlur;

**OUTPUT:**

****

**36)** Create an application in ReactJS form and add client validation

**FormValidation.js**

import React ,{useState} from "react";

function BasicFormValidation () {

const[formData, setFormData]=useState({name:"", email:""});

const[errors,setErrors]=useState({});

const handleChange=(e) => {

const {name, value} =e.target;

setFormData({...formData,[name]:value});

};

const validate=() => {

const newErrors={} ;

if(!formData.name)newErrors.name="Name is required";

if(!formData.email)newErrors.email="Email is required";

else if(!/\S+@\S+\.\S+/.test(formData.email))

newErrors.email="Email is invalid";

setErrors(newErrors);

return Object.keys(newErrors).length===0;

};

const handleSubmit=(e) => {

e.preventDefault();

if(validate()) {

alert("Form submitted successfully");

}

};

return(

<form onSubmit={handleSubmit}>

<div> <label> Name: </label>

<input type="text" name="name" value={formData.name} onChange={handleChange}/>

{errors.name && <p style={{color:"red"}}>{errors.name}</p> }

</div>

<div>

<label>Email:</label>

<input type="text" name="email" value={formData.email} onChange={handleChange} />

{errors.email && <p style={{color:"red"}}>{errors.email} </p> }

</div>

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

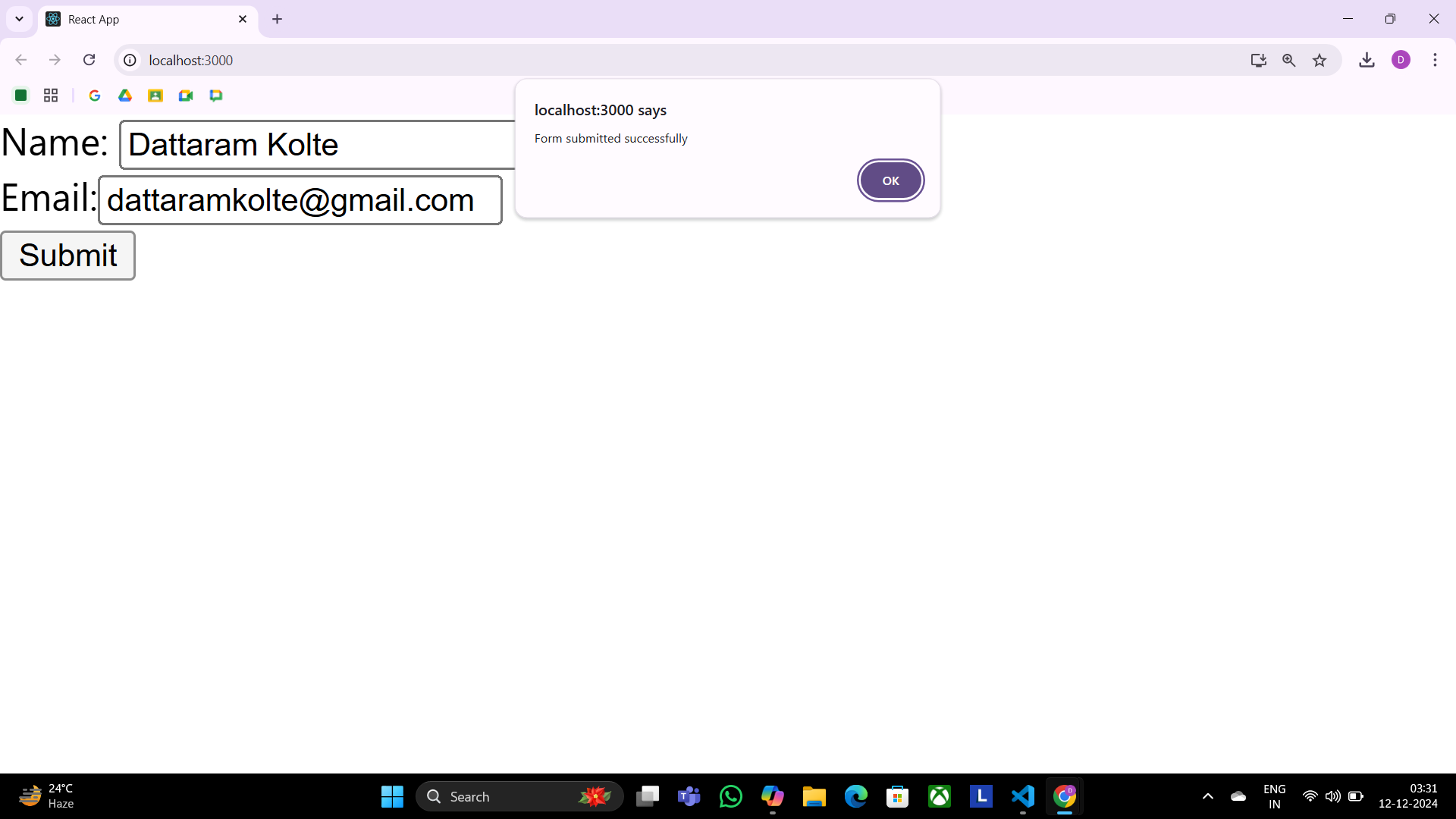
</form>

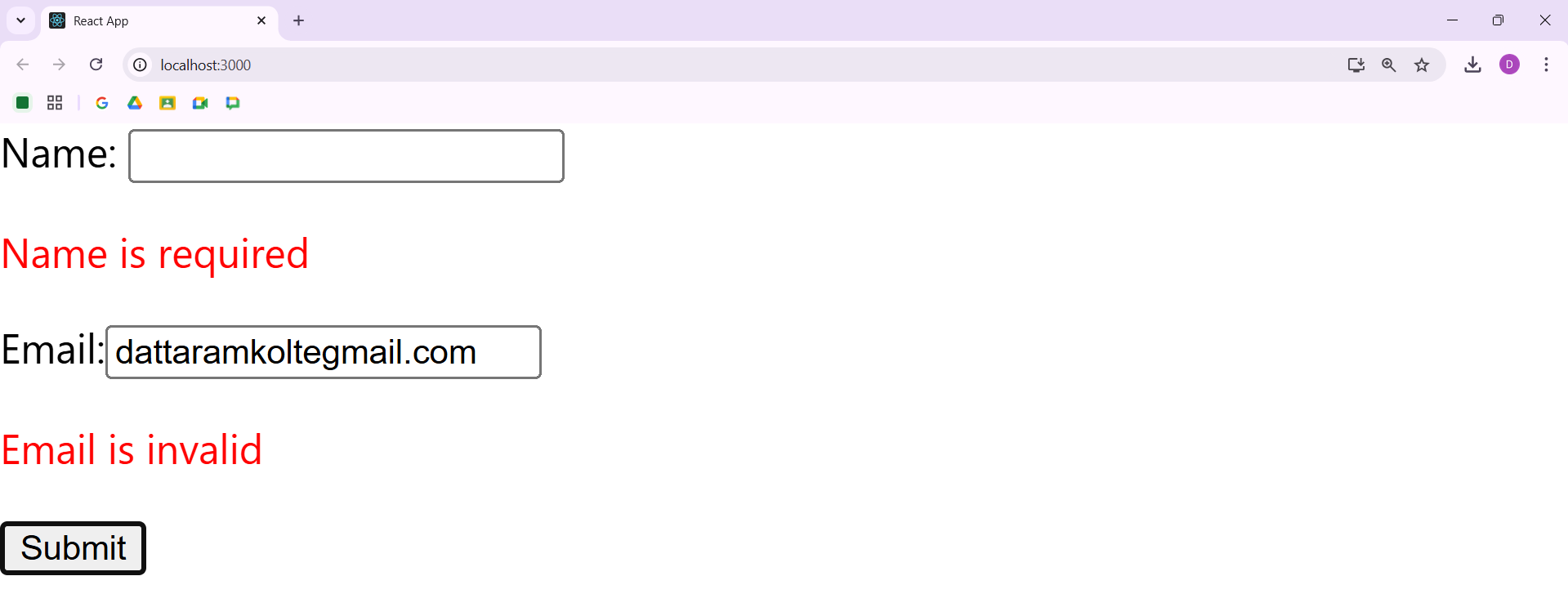
);

}

export default BasicFormValidation;

**OUTPUT:**

****



**37)** Write a Program to implement useEffect hook

**App.js**

import React, { useEffect } from 'react';

function SimpleComponent() {

useEffect(() => {

console.log('Component mounted!');

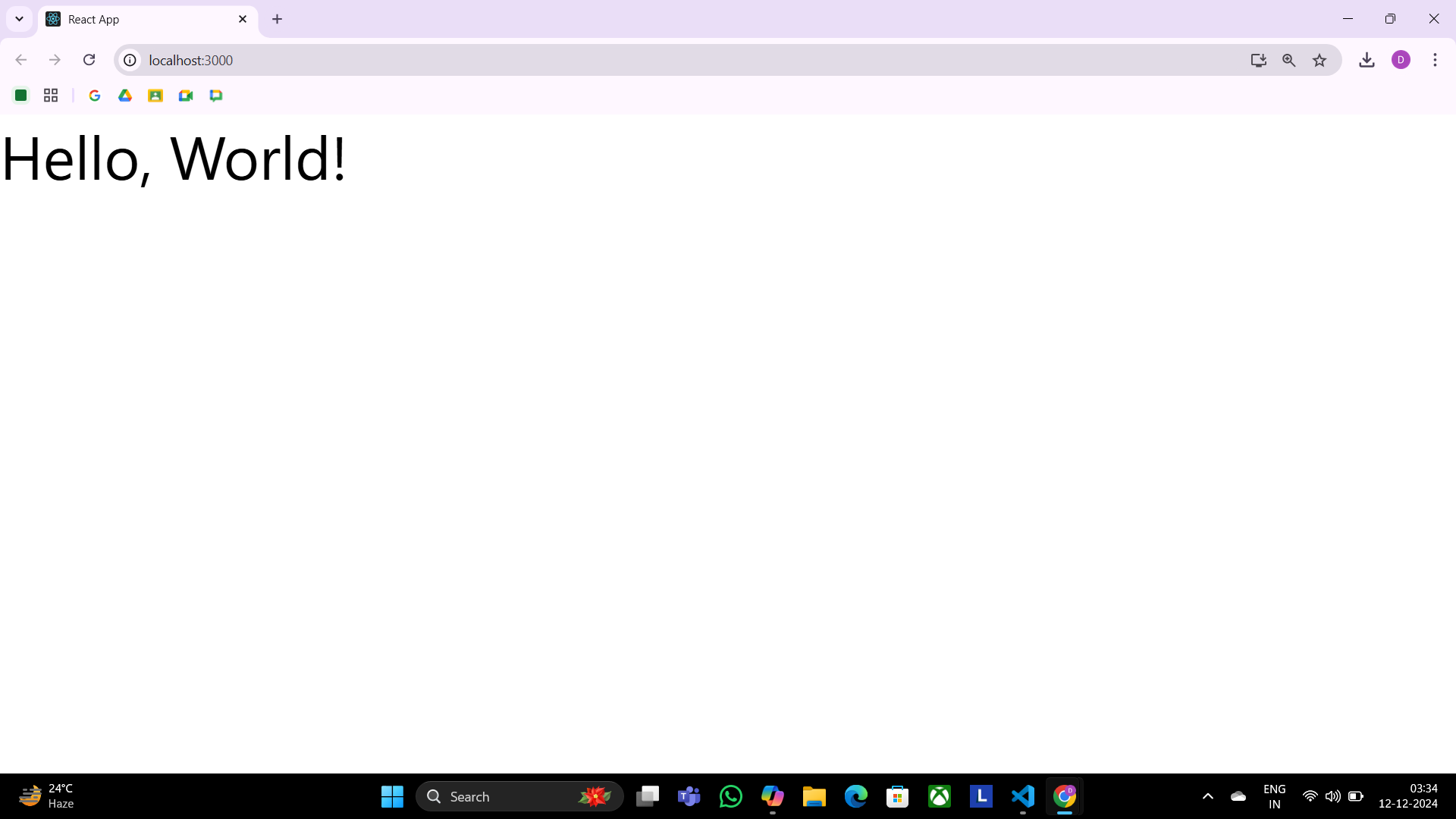
}, []); // Empty dependency array ensures this runs only once on mount

return <div>Hello, World!</div>;

}

export default SimpleComponent;

**OUTPUT:**

****

**38)** Create SPA using React Router

**App.js**

import React from "react";

import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";

import Home from "./components/Home";

import AboutUs from "./components/AboutUs";

import Contact from "./components/ContactUs";

const App = () => {

return (

<Router>

<nav>

<ul>

<li>

<Link to="/">Home</Link>

</li>

<li>

<Link to="/about">About</Link>

</li>

<li>

<Link to="/contact">Contact</Link>

</li>

</ul>

</nav>

<Routes>

<Route path="/" element={<Home />} />

<Route path="/about" element={<AboutUs />} />

<Route path="/contact" element={<Contact />} />

</Routes>

</Router>

);

};

export default App;

**Home.js**

import React from "react";

const Home = () => {

return (<div><h1>Home</h1></div>) }

export default Home;

**ContactUs.js**

import React from "react";

const Contact= () => {

return (<div><h1>Contact Us</h1></div>) }

export default Contact;

**AboutUs.js**

import React from "react";

const Contact= () => {

return (<div><h1>AboutUs</h1></div>) }

export default AboutUs;

**OUTPUT:**

