NODE JS PRACTICALS

Practical No. 1

Aim: Write a program to pass a message “Hello Node JS” using Node JS.

Source Code:

```
console.log("Hello Node JS");
```

Output Screen:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
PS C:\NodeJs> node Helloworld.js
Hello node js
PS C:\NodeJs>
```

Practical No. 2

Aim: Write a program to demonstrate Node.js Functions

Source Code:

```
//understand the javascript function
//standard function

function myfun(num1,num2)
{
   console.log(num1+num2);
   console.log(num1-num2);
   console.log(num1*num2);
   console.log(num1/num2);

}
myfun(12,3);
```

Output Screen:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
PS C:\NodeJs> node Function
15
9
36
4
PS C:\NodeJs>
```

Practical No. 3

a) **Aim:** Write a program to demonstrate Call-back function - Anonymous function using NodeJS.

Source Code:

```
//callback function - Anonymous Function
const message=function(){
   console.log("Welcome to node js");

}
setTimeout(message,3000);
//callback back as an Arrow function

setTimeout(()=>{
   console.log("Calling from arrow funciton");
},3000);
```

Output Screen:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\NodeJs> node Callback_Fn.js
Welcome to node js
Calling from arrow funciton
PS C:\NodeJs>
```

b) **Aim:** Write a program to demonstrate Call-back function using NodeJS.

Source Code:

```
//Javascript callback function example
function displayresult(some)
{
   console.log(some);
}
function calculate(x,y,mycallback)
{
   let sum=x+y;
   mycallback(sum);
}
calculate(5,10,displayresult);
```

Output Screen:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
PS C:\NodeJs> node Callback_Fn2.js
15
PS C:\NodeJs>
```

Practical No. 4

Aim: Write a program to demonstrate Node.js Modules

Source Code:

```
//Implementing own modules
var dt=require('./myfirstmodule');
console.log(dt.myDateTime());
```

Using myfirstmodule.js

```
//Creating own modules
exports.myDateTime=function()
{
   return Date();
}
```

Output Screen:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
PS C:\NodeJs> node modules.js
Thu Oct 10 2024 13:48:59 GMT+0530 (India Standard Time)
PS C:\NodeJs>
```

Practical No. 5

Aim: Write a program to demonstrate various Node.js Events

Source Code:

```
// step 1 importing event
const events = require("events");
// step 2 creating an Event emitter object
const eventEmitter = new events.EventEmitter();
//write a function of event 1
function listner1() {
   console.log("Event recevied by Listner 1");
}
//write a function of event 2
function listner2() {
   console.log("Event recevied by Listner 2");
}
// step 3 adding listener through addlistener or on
eventEmitter.addListener("write", listner1);
eventEmitter.on("write", listner2);
// step 4 emiting event
eventEmitter.emit("write");
console.log(eventEmitter.listenerCount("write"));
// step 5  removing listener
eventEmitter.removeListener("write", listner1);
console.log("Listener 1 is removed");
eventEmitter.emit("write");

console.log(eventEmitter.listenerCount("write"));
console.log("Program Ended");
```

Output Screen:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
PS C:\NodeJs> node Events.js
Event recevied by Listner 1
Event recevied by Listner 2
2
Listener 1 is removed
Event recevied by Listner 2
1
Program Ended
PS C:\NodeJs>
```

Practical No. 6

Aim: Create an HTTP Server. Write a program to demonstrate routing through http server.

Source Code:

```
// create Server


var http=require('http');
var server=http.createServer(function(req,res)
{  res.write("This is the http server");
});
server.listen(2000);
console.log("server is running");
```

Output Screen:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
PS C:\NodeJs> node create_server.js
server is running
```

Practical No. 7

Aim: Write a program to demonstrate routing through http server.

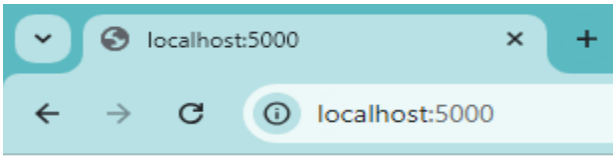
Source Code:

```
//understand routing in http module
var http = require('http'); // Import Node.js core module

var server = http.createServer(function (req, res) {   //create web server
   if (req.url == '/') { //check the URL of the current request

      // set response header
      res.writeHead(200, { 'Content-Type': 'text/html' });

      // set response content
      res.write('<html><body><p>This is home Page.</p></body></html>');
      res.end();

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

      res.writeHead(200, { 'Content-Type': 'text/html' });
      res.write('<html><body><p>This is student Page.</p></body></html>');
      res.end();

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

      res.writeHead(200, { 'Content-Type': 'text/html' });
      res.write('<html><body><p>This is admin Page.</p></body></html>');
      res.end();

   }
   else
      res.end('Invalid Request!');

});

server.listen(5000); //6 - listen for any incoming requests

console.log('Node.js web server at port 5000 is running..')
```

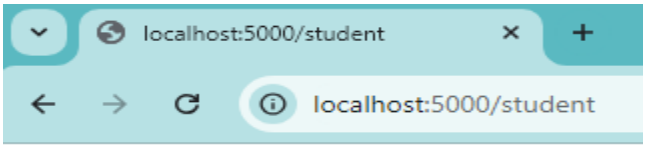
Output Screen:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

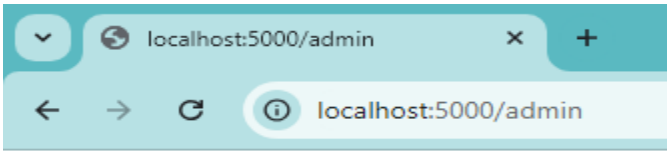
```
Node.js v18.14.2
PS E:\Web Technology\NodeJs-Day1> node Routing.js
Node.js web server at port 5000 is running..
```



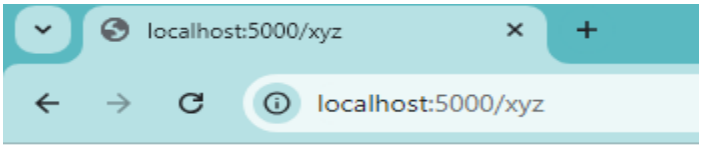
This is home Page.



This is student Page.



This is admin Page.



Invalid Request!

Practical No. 8

Aim: Write a program to demonstrate custom event using Node JS.

Source Code:

```
//custom event

const events = require("events");
const eventEmitter = new events.EventEmitter();
eventEmitter.on("connection", handleConnectionEvent);
eventEmitter.emit("connection");
eventEmitter.emit("connection");
eventEmitter.emit("connection");
eventEmitter.emit("connection");
function handleConnectionEvent() {
   console.log("Conneciton Made!");
}
console.log("End of Program");
```

Output Screen:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Technology\NodeJs-Day1> node Custom_Events
Conneciton Made!
Conneciton Made!
Conneciton Made!
Conneciton Made!
End of Program
PS E:\Web Technology\NodeJs-Day1>
```

Practical No. 9

Aim: Using File Handling demonstrate all basic file operations (Create, write, read, delete & buffer)

Source Code:

ReadTextFile.js

```
var fs = require('fs');

fs.readFile('FileOperation.txt', function
                    (err, data) {if (err) throw err;
    console.log(data.toString());
});
```

Output Screen:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\arsha\Documents\WEB TECH\nodeLearning> node ReadTextFile.js
hello I m from he text file
this program is based on getting the file input
through node js
PS C:\Users\arsha\Documents\WEB TECH\nodeLearning>
```

WriteTextFile.js

```
var fs=require('fs');
fs.writeFile('NewTextFile.txt' ,'Hi Welcome to the txt file',
    function(err){if(err) throw err;
    else {
        console.log("Writing complete");
    }
});
```

Output Screen:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\arsha\Documents\WEB TECH\nodeLearning> node WriteTextFile.js
Writing complete
PS C:\Users\arsha\Documents\WEB TECH\nodeLearning>
```

BufferRead.js

```
//Write a program to read a file using buffer byte

by bytevar fs = require('fs');

fs.open('NewTextFile.txt', 'r', function (err, fd) {

    if (err) {

        return console.error(err); }

    var buffr = new Buffer.alloc(10240);

    fs.read(fd, buffr, 0, buffr.length, 0, function (err,
        bytes) {if (err) throw err;

    // Print only read bytes to avoid junk.

      if (bytes>0) {

            console.log(buffr.slice(0, bytes).toString());  }

        // Close the opened file.
        fs.close(fd, function (err) {

if (err) throw err; });   });    });
```

Output Screen:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\arsha\Documents\WEB TECH\nodeLearning> node BufferRead.js
Hi Welcome to the txt file
PS C:\Users\arsha\Documents\WEB TECH\nodeLearning>
```

DeletionFile.js

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

fs.unlink('Starkexpo.txt',

function(){

    console.log('Delete Operation completed');


});
```

Output Screen:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\arsha\Documents\WEB TECH\nodeLearning> node DeletionFile.js
Delete Operation completed
PS C:\Users\arsha\Documents\WEB TECH\nodeLearning>
```

Practical No. 10

Aim: Create an application to establish a connection with the MySQL database and perform basicdatabase operations on it.

Line of Code:

//Write a program for Database connection in nodejs

```
//checking database connection
var mysql = require('mysql');

var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: ""
});

con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
});
```

Output Screen:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\arsha\Documents\WEB TECH\nodeLearning> node Connection.js
Connected!
```

```
//creating database
var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: ""
});
con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  con.query("CREATE DATABASE mydb", function (err, result) {
    if (err) throw err;
    console.log("Database created");
  });     });
```

Output Screen:

```
PS C:\Users\arsha\Documents\WEB TECH\nodeLearning> node '.\Database creation.js'
Connected!
Database created
```

```
//creating table in database
var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "",
  database: "mydb"     });
```

```
con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  var sql = "CREATE TABLE customers (name VARCHAR(255), address VARCHAR(255))";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("Table created");
  });   });
```

Output Screen:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\arsha\Documents\WEB TECH\nodeLearning> node '.\Table Creation.js'
Connected!
Table created
```

```
//inserting record inside table
var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "",
  database: "mydb"   });
con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
```

```
 var sql = "INSERT INTO customers (id, name, address) VALUES (‘1’ 'Tony Stark',
'Miami')";
 con.query(sql, function (err, result) {
  if (err) throw err;
  console.log("1 record inserted");
 });   });
```

Output Screen:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\arsha\Documents\WEB TECH\nodeLearning> node '.\Table Insertion.js'
Connected!
1 record inserted
```

```
 //Table Reading.js
var mysql = require('mysql');
var con = mysql.createConnection({
 host: "localhost",
 user: "root",
 password: "",
 database: "mydb"   });
con.connect(function(err) {
 if (err) throw err;
 con.query("SELECT * FROM customers", function (err, result, fields) {
  if (err) throw err;
  console.log(result);   });   });
```

Output Screen:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\arsha\Documents\WEB TECH\nodeLearning> node '.\Table Reading.js'
Connected!
[ { id: 1, name: 'Tony Stark', address: 'Miami' } ]
```

Table Updation.js

```
//Write a program to perfrom Updation of rows on table using nodejs

var mysql = require('mysql');

var con = mysql.createConnection({

  host: "localhost",

  user: "root",

  password: "",

  database: "mydb"   });

  con.connect(function (err)

  { if (err) throw err;
      console.log("Connected!");

      var sql = "UPDATE CUSTOMERS SET address='Miami' where
      address='Texas'";con.query(sql, function (err, result) {

    if (err) throw err;

   console.log(result.affectedRows + " record(s) updated");   });  })
```

Output Screen:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\arsha\Documents\WEB TECH\nodeLearning> node '.\Table Updation.js'
Connected!
1 record(s) updated
```

Table Deletion.js

```
var mysql = require('mysql');

var con = mysql.createConnection({

  host: "localhost",

  user: "root",

  password: "",

  database: "mydb"   });

  con.connect(fun
      ction (err) { if
      (err) throw
      err;
      console.log("
      Connected!");

      var sql = "DELETE FROM CUSTOMERS WHERE
      address='Brooklyn'";con.query(sql, function (err,
      result) {

          if (err) throw err;

          console.log("Number of records deleted: " + result.affectedRows);

      });    })
```

Output Screen:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\arsha\Documents\WEB TECH\nodeLearning> node '.\Table Deletion.js'
Connected!
Number of records deleted: 1
```

REACT JS PRACTICALS

Practical No. 11

Aim: Create an application to implement class and functional components in ReactJS.

Line of Code:

Step 1: Setting Up the React App

Create React App: npx create-react-app react-component-demo
cd react-component-demo
npm start

Step 2: Creating Class and Functional Components

//ClassComponent.js

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

class ClassComponent extends Component {

constructor(props) {

super(props);

this.state = {

message: 'Hello from Class Component!',

};  }

render() {

return (

<div style={{ border: '2px solid blue', padding: '20px', margin: '10px' }}>

<h2>{this.state.message}</h2>

<p>This is rendered using a class component.</p>

</div>

); }  }

export default ClassComponent;
```

//FunctionalComponent.js

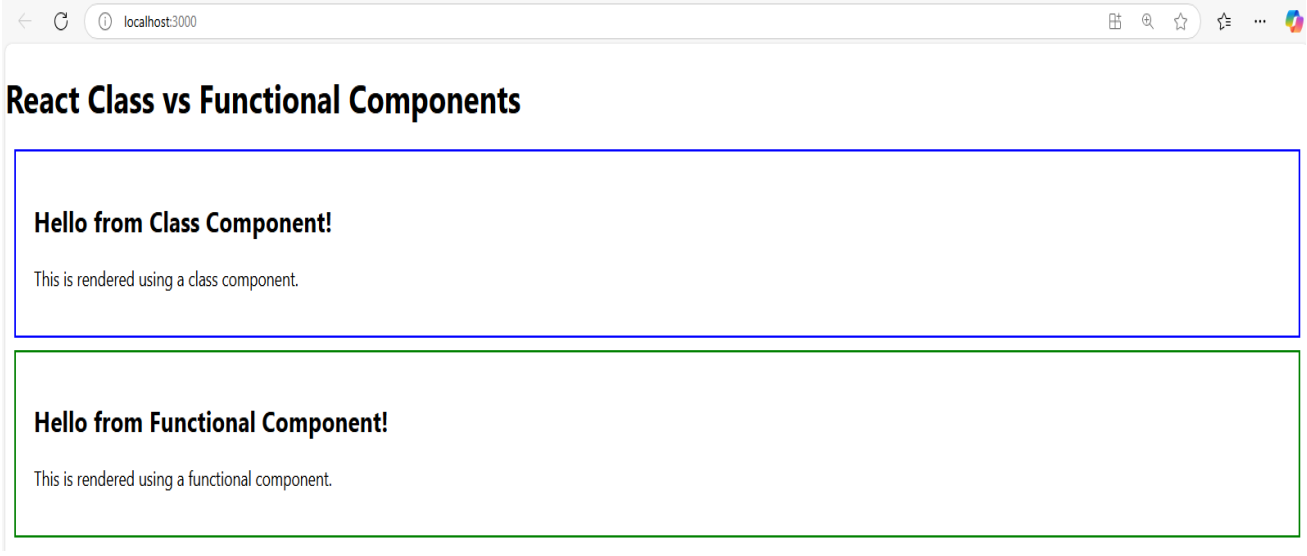
```
import React, { useState } from 'react';

function FunctionalComponent() {

const [message] = useState('Hello from Functional Component!');

return (

<div style={{ border: '2px solid green', padding: '20px', margin: '10px' }}>

<h2>{message}</h2>

<p>This is rendered using a functional component.</p>

</div>

);   }

export default FunctionalComponent;
```

Step 3: Import Components into App

//App.js

```
import React from 'react';
import ClassComponent from './ClassComponent';
import FunctionalComponent from './FunctionalComponent';

function App() {
  return (
    <div className="App">
      <h1>React Class vs Functional Components</h1>
      <ClassComponent />
      <FunctionalComponent />
    </div>
  );
}
export default App;
```

Output Screen:



Practical No. 12

Aim: Create an application in ReactJS to import and export components.

Line of Code:

Step 1: Setting Up the React App

Create React App: npx create-react-app react-component-demo
cd react-component-demo
npm start

Step 2: Create Multiple Components

//Header.js

```
import React from 'react';
function Header() {
return (
<header style={{ backgroundColor: '#4CAF50', color: 'white', padding: '10px' }}>
<h1>Welcome to React Import and Export Demo</h1>
</header>
);
}
export default Header;
```

//Footer.js

```
import React from 'react';

function Footer() {

return (

<footer style={{ backgroundColor: '#333', color: 'white', padding: '10px',

position: 'fixed', bottom: 0, width: '100%' }}>

<p>React Import and Export Demo</p>

</footer>

);  }

export default Footer;
```

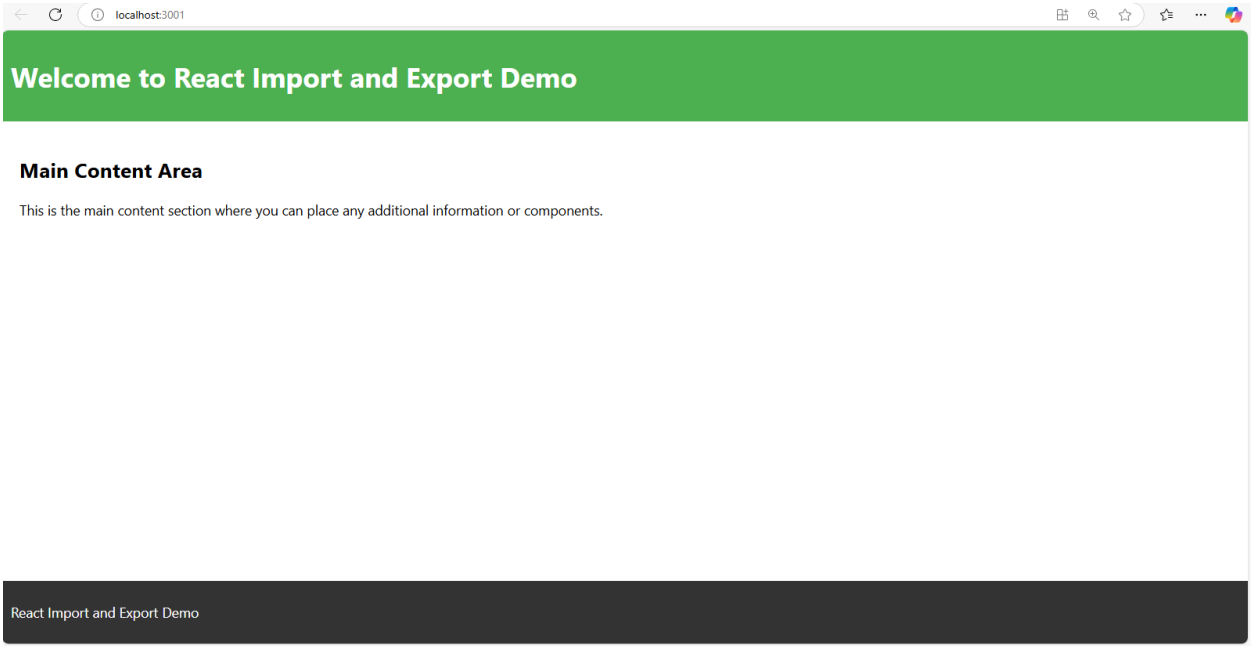
// MainContent Component

```
import React from 'react';
function MainContent() {
return (
<div style={{ padding: '20px' }}>
<h2>Main Content Area</h2>
<p>This is the main content section where you can place any additional
information or components.</p>
</div>
);  }
export default MainContent;
```

Step 3: Import Components into App

```
import React from 'react';
import Header from './Header'; // Importing Header component
import MainContent from './MainContent'; // Importing MainContent component
import Footer from './Footer'; // Importing Footer component
function App() {
return (
<div>
<Header />   <MainContent />    <Footer />
</div>
); }
export default App;
```

Output Screen:



Practical No. 13

Aim: Create an application to implement state and props in ReactJS.

Line of Code:

Step 1: Setting Up the React App

Create React App: npx create-react-app react-component-demo
cd react-component-demo
npm start

Step 2: Create a Parent Component

//ParentComponent.js

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

import ChildComponent from './ChildComponent';

function ParentComponent() {

// Define state

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

const updateMessage = () => {

setMessage('Message Updated from Parent!');

};

return (

<div style={{ border: '2px solid blue', padding: '20px', margin: '10px' }}>

<h2>Parent Component</h2>

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

{/* Passing message and function as props to ChildComponent */}

<ChildComponent message={message} updateMessage={updateMessage} />

</div>

); }

export default ParentComponent;
```

Step 3: Create a Child Component

//ChildComponent.js

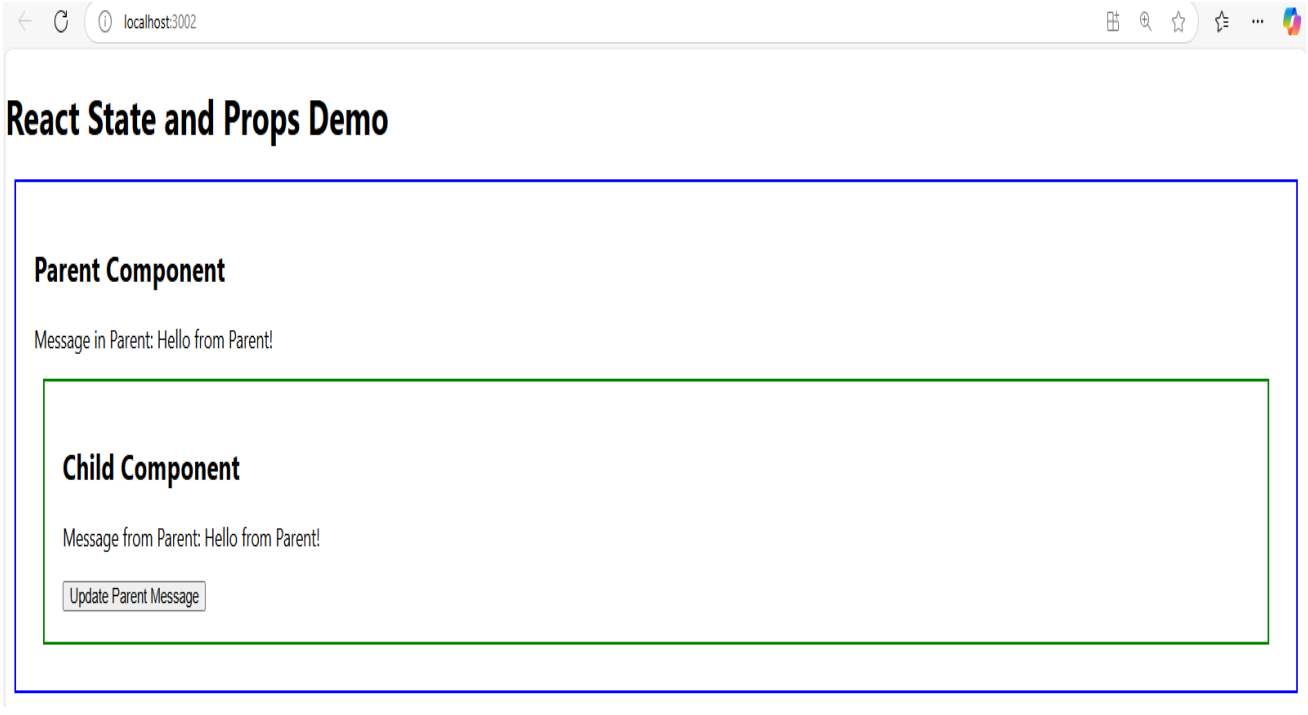
```
import React from 'react';
function ChildComponent({ message, updateMessage }) {
return (
<div style={{ border: '2px solid green', padding: '20px', margin: '10px' }}>
<h2>Child Component</h2>
<p>Message from Parent: {message}</p>
<button onClick={updateMessage}>Update Parent Message</button>
</div>
);  }
export default ChildComponent;
```

Step 4: Modify App.js

//App.js

```
import React from 'react';
import ParentComponent from './ParentComponent';
function App() {
return (
<div className="App">
<h1>React State and Props Demo</h1>
<ParentComponent />
</div>
); }
export default App;
```

Output Screen:



Practical No. 14

Aim: Create an application in ReactJS to use DOM events.

Line of Code:

Step 1: Setting Up the React App

Create React App: npx create-react-app react-component-demo
cd react-component-demo
npm start

Step 2: Handling Different DOM Events

//DOMEventsComponent.js

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

function DOMEventsComponent() {

const [inputValue, setInputValue] = useState('');

const [hovered, setHovered] = useState(false);

// Handle button click

const handleClick = () => {

alert('Button clicked!');

};

// Handle input change

const handleChange = (event) => {

setInputValue(event.target.value);

};

// Handle mouse hover

const handleMouseOver = () => {

setHovered(true);

};

const handleMouseOut = () => {

setHovered(false);

};
```

```
return (
<div style={{ padding: '20px', textAlign: 'center' }}>
<h2>React DOM Events Demo</h2>
{/* onClick Event */}
<button onClick={handleClick} style={{ padding: '10px', fontSize: '16px' }}>
Click Me
</button>
{/* onChange Event */}
<div style={{ margin: '20px 0' }}>
<input
type="text"
placeholder="Type something..."
value={inputValue}
onChange={handleChange}
style={{ padding: '10px', fontSize: '16px' }}
/>
<p>You typed: {inputValue}</p>
</div>
{/* onMouseOver Event */}
<div
onMouseOver={handleMouseOver}
onMouseOut={handleMouseOut}
style={{
backgroundColor: hovered ? 'lightblue' : 'lightgray',
padding: '20px',
cursor: 'pointer',
}}
```

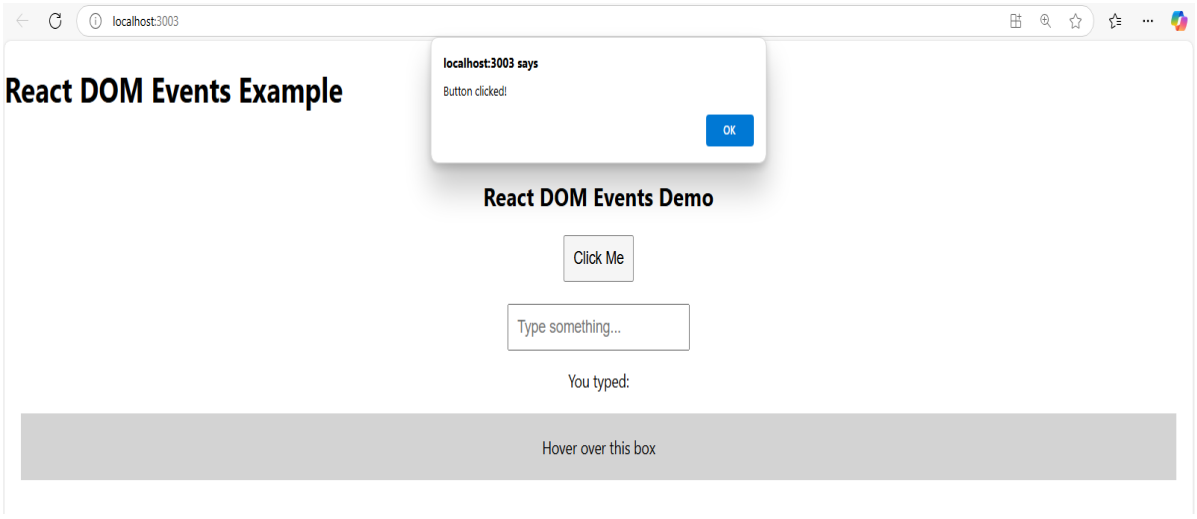
```
>
{hovered ? 'Mouse is Over' : 'Hover over this box'}
</div>
</div>
);  }
export default DOMEventsComponent;
```

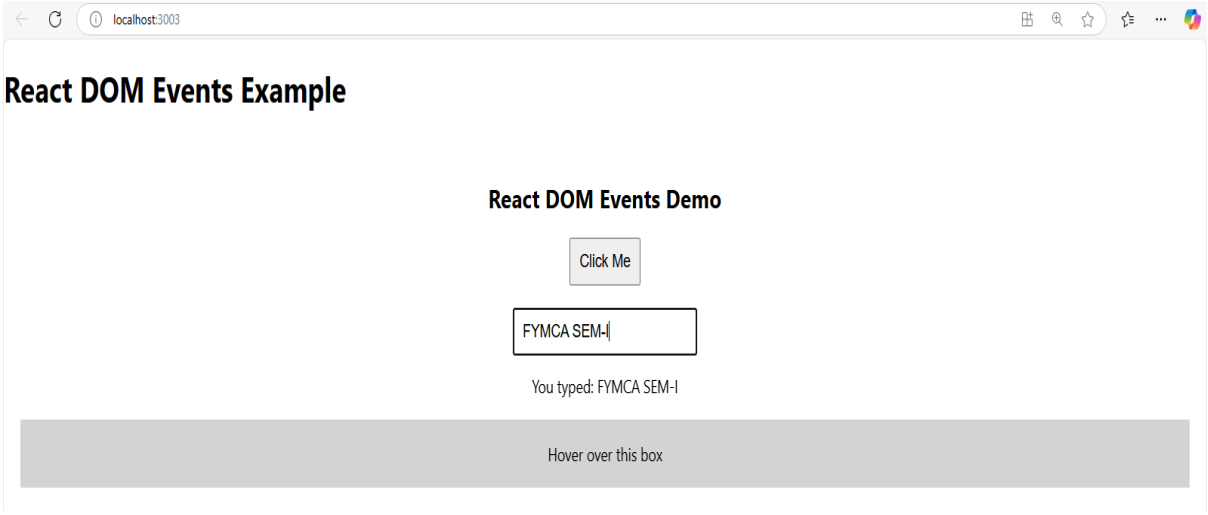
Step 3: Modify App Component

//App.js

```
import React from 'react';
import DOMEventsComponent from './DOMEventsComponent';
function App() {
return (
<div className="App">
<h1>React DOM Events Example</h1>
<DOMEventsComponent />
</div>
); }
export default App;
```

Output Screen:





Practical No. 15

Aim: Create an application in ReactJS form and add client and server-side validation.

Line of Code:

Step 1: Setting Up the React App

```
npx create-react-app react-form-validation-demo
cd react-form-validation-demo
npm start
```

Step 2: Create the Form Component

//RegistrationForm.js

```
 import React, { useState } from 'react';
function RegistrationForm() {
const [formData, setFormData] = useState({
name: '',
email: '',
password: ''
});
const [formErrors, setFormErrors] = useState({});
const [isSubmitted, setIsSubmitted] = useState(false);
// Handle form input changes
const handleChange = (e) => {
const { name, value } = e.target;
setFormData({
...formData,
[name]: value
});
};
```

```
// Validate form data
const validate = () => {
let errors = {};
if (!formData.name) errors.name = 'Name is required';
if (!formData.email) {
errors.email = 'Email is required';
} else if (!/\S+@\S+\.\S+/.test(formData.email)) {
errors.email = 'Email address is invalid';
}
if (!formData.password) {
errors.password = 'Password is required';
} else if (formData.password.length < 6) {
errors.password = 'Password must be at least 6 characters';
}
return errors;
};
// Handle form submission
const handleSubmit = (e) => {
e.preventDefault();
const errors = validate();
setFormErrors(errors);
if (Object.keys(errors).length === 0) {
// Mock server request
setIsSubmitted(true);
console.log('Form data submitted:', formData);
// In real implementation, send data to the server here
} else{
```

```
setIsSubmitted(false);
}
};
return (
<div className="form-container" style={{ textAlign: 'left', padding: '20px' }}>
<h2>Registration Form</h2>
{isSubmitted && <p style={{ color: 'green' }}>Form submitted successfully!</p>}
<form onSubmit={handleSubmit}>
<div>
<label>Name:</label>
<input
type="text"
name="name"
value={formData.name}
onChange={handleChange}
style={{ display: 'block', marginBottom: '10px', padding: '5px' }}
/>
{formErrors.name && <p style={{ color: 'red' }}>{formErrors.name}</p>}
</div>
<div>
<label>Email:</label>
<input
type="email"
name="email"
value={formData.email}
onChange={handleChange}
style={{ display: 'block', marginBottom: '10px', padding: '5px' }}
```

```
/>
{formErrors.email && <p style={{ color: 'red' }}>{formErrors.email}</p>}
</div>
<div>
<label>Password:</label>
<input
type="password"
name="password"
value={formData.password}
onChange={handleChange}
style={{ display: 'block', marginBottom: '10px', padding: '5px' }}
/>
{formErrors.password && <p style={{ color: 'red' }}>{formErrors.password}</p>}
</div>
<button type="submit" style={{ padding: '10px 20px', marginTop: '10px' }}>
Submit
</button>
</form>
</div>
);
}
export default RegistrationForm;
```

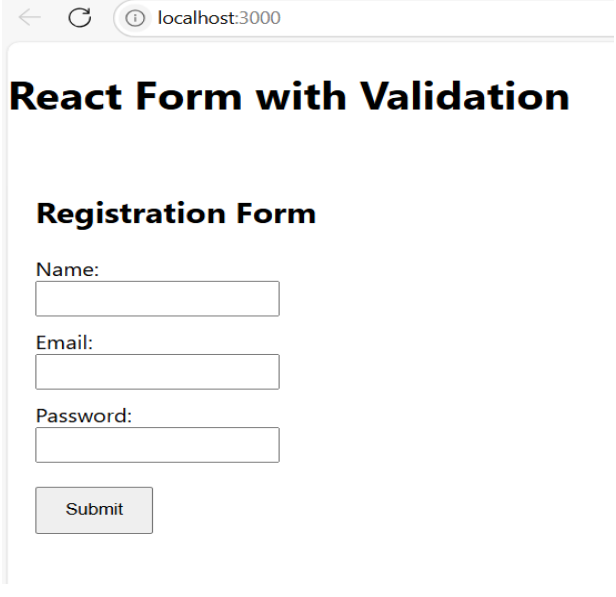
Step 3: Modify App Component

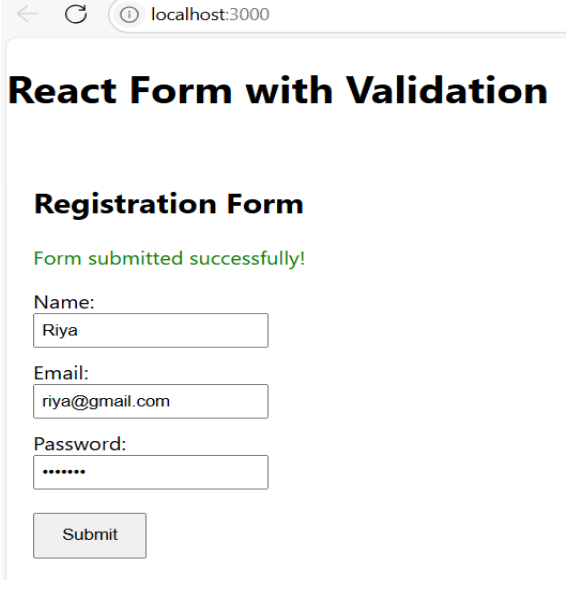
//App.js

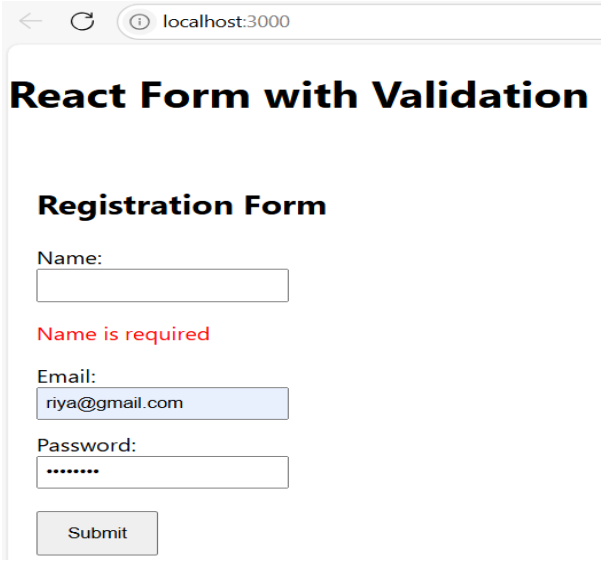
```
import React from 'react';
import RegistrationForm from './RegistrationForm';
function App() {
```

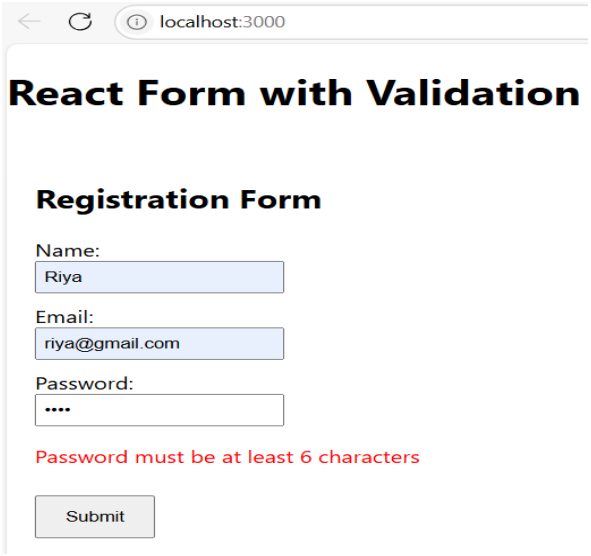
```
return (
<div className="App">
<h1>React Form with Validation</h1>
<RegistrationForm />
</div>
);
}
export default App;
```

Output Screen:






localhost:3000
React Form with Validation
Registration Form
Name:
Name is required
Email:
riya@gmail.com
Password:
Submit


localhost:3000
React Form with Validation
Registration Form
Name:
Riya
Email:
riya@gmail.com
Password:
Password must be at least 6 characters
Submit

Practical No. 16

Aim: Create an application in ReactJS that uses routing for navigation.

Line of Code:

Step 1: Setting Up the React App

```
npx create-react-app react-routing-demo
cd react-routing-demo
npm install react-router-dom
npm start
```

Step 2: Setting Up React Router

//Home.js

```
import React from 'react';
function Home() {
return (
<div style={{ textAlign: 'center', padding: '20px' }}>
<h2>Welcome to the Home Page</h2>
<p>This is the main page of our application.</p>
</div>
);
}
export default Home;
```

//About.js

```
import React from 'react';
function About() {
return (
<div style={{ textAlign: 'center', padding: '20px' }}>
```

```
<h2>About Us</h2>
<p>This is the about page where we describe our app.</p>
</div>
);
}
export default About;
```

//Contact.js

```
import React from 'react';
function Contact() {
return (
<div style={{ textAlign: 'center', padding: '20px' }}>
<h2>Contact Us</h2>
<p>This is the contact page for inquiries.</p>
</div>
);
}
export default Contact;
```

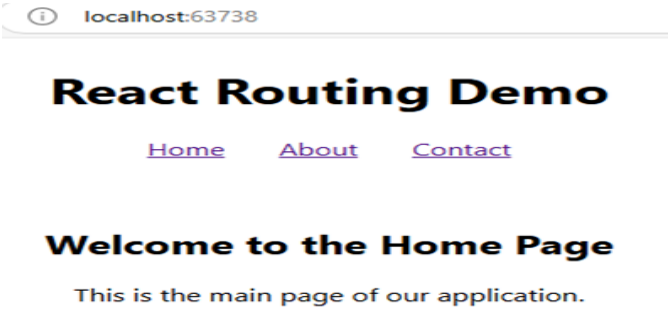
Step 3: Setting Up the Router

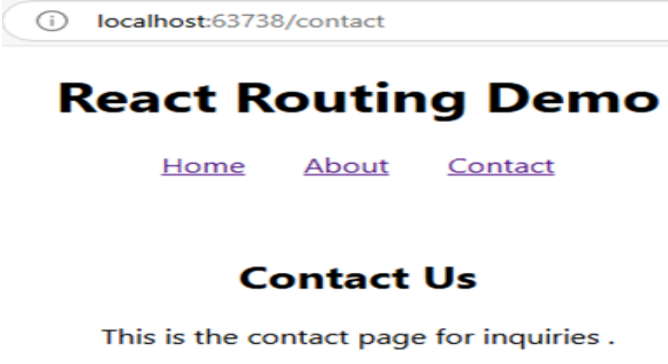
//App.js

```
import React from 'react';
import { BrowserRouter as Router, Route, Routes, Link } from 'react-router-dom';
import Home from './Home';
import About from './About';
import Contact from './Contact';
function App() {
```

```
return (
<Router>
<div style={{ textAlign: 'center', padding: '20px' }}>
<h1>React Routing Demo</h1>
{/* Navigation Links */}
<nav style={{ marginBottom: '20px' }}>
<Link to="/" style={{ margin: '0 15px' }}>Home</Link>
<Link to="/about" style={{ margin: '0 15px' }}>About</Link>
<Link to="/contact" style={{ margin: '0 15px' }}>Contact</Link>
</nav>
{/* Route Definitions */}
<Routes>
<Route path="/" element={<Home />} />
<Route path="/about" element={<About />} />
<Route path="/contact" element={<Contact />} />
</Routes>
</div>
</Router>
);
}
export default App;
```

Output Screen:





Practical No. 17

Aim: Create a simple ReactJS application with Hooks (useState, useEffect, useContext).

Line of Code:

Step 1: Setting Up the React App
Continue in the same project or create a new one if needed:
npx create-react-app react-hooks-demo
cd react-hooks-demo
npm start

Step 2: Using useState for Counter Functionality

2.1 Create a Counter Component

// Counter.js

```
import React, { useState } from 'react';
function Counter() {
const [count, setCount] = useState(0);
return (
<div style={{ textAlign: 'center', padding: '20px' }}>
<h2>Counter</h2>
<p>Current Count: {count}</p>
<button onClick={() => setCount(count + 1)}>Increase</button>
<button onClick={() => setCount(count - 1)} style={{ marginLeft: '10px'
}}>Decrease</button>
</div>
);
}
export default Counter;
```

Step 3: Using useEffect for Data Fetching

// DataFetching.js

```
import React, { useState, useEffect } from 'react';
function DataFetching() {
const [data, setData] = useState([]);
const [loading, setLoading] = useState(true);
useEffect(() => { s
fetch('https://jsonplaceholder.typicode.com/posts') l
```

```
.then((response) => response.json())
.then((data) => {
setData(data.slice(0, 5)); // Displaying only the first 5 items
setLoading(false);
})
.catch((error) => console.error(error));
}, []); // Empty array ensures this runs once on mount

return (
<div style={{ textAlign: 'center', padding: '20px' }}>
<h2>Data Fetching</h2>
{loading ? (
<p>Loading...</p>
) : (
<ul>
{data.map((item) => (
<li key={item.id}>{item.title}</li>
))}
</ul>
)}
</div>
);
}
export default DataFetching;
```

Step 4: Using useContext for Theme Management

//ThemeContext.js

```
import React, { createContext, useState } from 'react';
export const ThemeContext = createContext();
export const ThemeProvider = ({ children }) => {
const [isDarkTheme, setIsDarkTheme] = useState(false);
const toggleTheme = () => {
setIsDarkTheme((prevTheme) => !prevTheme);
};
return (
<ThemeContext.Provider value={{ isDarkTheme, toggleTheme }}>
{children}
</ThemeContext.Provider>
); }     export default ThemeContext;
```

//ThemedComponent.js

Create a ThemedComponent.js file:

```
import React, { useContext } from 'react';
import { ThemeContext } from './ThemeContext';
function ThemedComponent() {
const { isDarkTheme, toggleTheme } = useContext(ThemeContext);
return (
<div
style={{
textAlign: 'center',
padding: '20px',
backgroundColor: isDarkTheme ? '#333' : '#fff',
color: isDarkTheme ? '#fff' : '#000',
}}
>
<h2>Theme Toggle</h2>
<p>Current Theme: {isDarkTheme ? 'Dark' : 'Light'}</p>
<button onClick={toggleTheme}>Toggle Theme</button>
</div>
);  }
export default ThemedComponent;
```

// App.js

```
import React from 'react';
import Counter from './Counter';
import DataFetching from './DataFetching';
import ThemedComponent from './ThemedComponent';
import { ThemeProvider } from './ThemeContext';
function App() {
return (
<ThemeProvider>
<div className="App" style={{ textAlign: 'center', padding: '20px' }}>
<h1>React Hooks Demo</h1>
<Counter />
<DataFetching />
<ThemedComponent />
</div>
</ThemeProvider>  ); }    export default App;
```

Output Screen: