Node JS Practical

Practical 1

AIM:- Write a program to pass a message " Hello ,I am Aradhya" using Node JS.

Line of Code :

```
console.log("Hello ,I am Aradhya");
```

Output Screen :

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS      powershell  + ∨  ⬚  🗑  …  ✕

PS D:\Aradhya_WT> node Practical1.js
Hello ,I am Aradhya
PS D:\Aradhya_WT>
```

Practical 2

AIM:- Write a program to demonstrate Node.js Functions

Line of Code :

```javascript
function multiply(x,y){
    return x*y;
}
let result = multiply(34,56);
console.log(result);
```

Output Screen :

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS        powershell  + ∨   □  🗑  ⋯  ✕

PS D:\Aradhya_WT> node Practical2.js
1904
PS D:\Aradhya_WT>
```
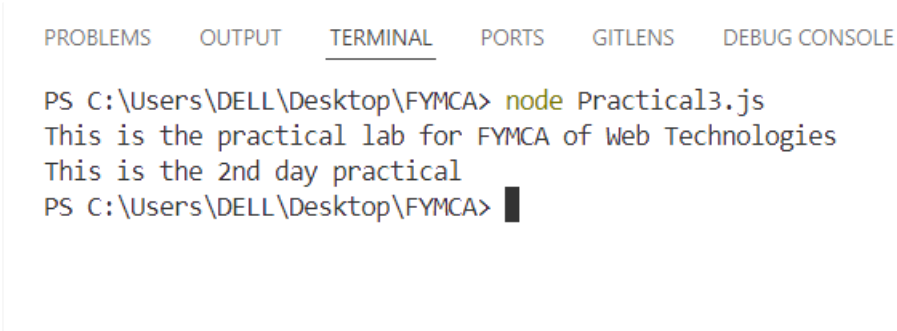
Practical 3

AIM:- Write a program to demonstrate Call-Back function – Anonymous function using Node JS

Line of Code :

```
const message=function()
{
    console.log("This is the practical lab for FYMCA of Web Technologies");
}
setTimeout(message,10000)
    //callback function as an arrow function
    setTimeout(() => {
        console.log("This is the 2nd day practical");
    },3000);
```

Output Screen:

```
PROBLEMS    OUTPUT    TERMINAL    PORTS    GITLENS    DEBUG CONSOLE

PS C:\Users\DELL\Desktop\FYMCA> node Practical3.js
This is the practical lab for FYMCA of Web Technologies
This is the 2nd day practical
PS C:\Users\DELL\Desktop\FYMCA> 
```

Name: Arif Shaikh   **Roll No. 39**

Practical 4

AIM:- Write a program to demonstrate Node.js Modules.
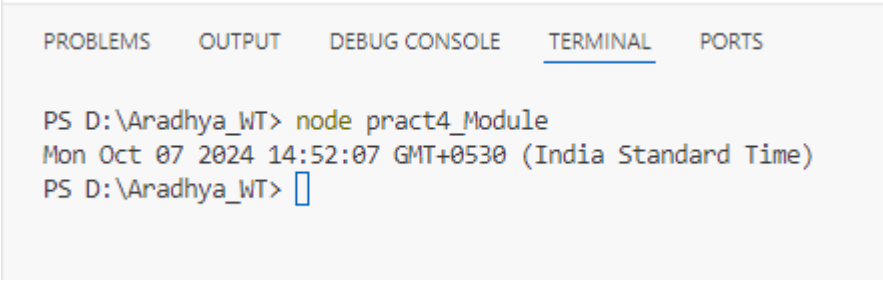
Line of Code :

File : Practt4.js :

```javascript
exports.myDateFun = function() {
    return Date();
}
```

File : pract4_Module.js :

```javascript
//understand how to call the modules

const dt = require('./Practt4');
console.log(dt.myDateFun());
```

Output Screen :

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS


PS D:\Aradhya_WT> node pract4_Module
Mon Oct 07 2024 14:52:07 GMT+0530 (India Standard Time)
PS D:\Aradhya_WT> []
```
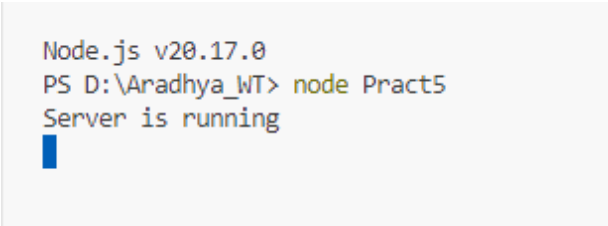
Name: Arif Shaikh   **Roll No. 39**

Practical 5

AIM:- **Create an HTTP Server and perform operations on it.**
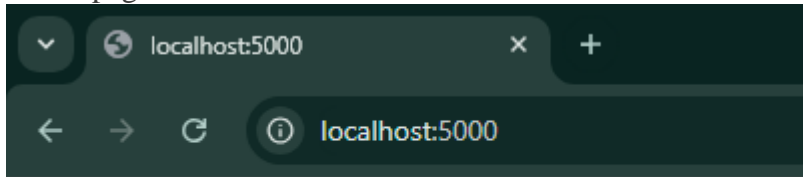
Line of Code:

```
//understand routing through http server
var http = require('http');
var server = http.createServer(function(req,res){
    if(req.url=='/')
    {
        res.writeHead(200,{'content-Type':'text/html'})
        res.write("<h1>Home page</h1>");
        res.end();
    }
    else if(req.url=='/student')
    {
        res.writeHead(200,{'content-Type':'text/html'})
        res.write("<h1>Master of Computer Applications</h1>");
        res.end();
    }
    else if(req.url=='/admin')
    {
        res.writeHead(200,{'content-Type':'text/html'})
        res.write("<h1>Your fee structure will be displayed on the notice Board</h1>");
        res.end();
    }
    else
    {
        res.write("<h1>Invalid Page</h1>");
        res.end();
    }
});
server.listen(5000);
console.log("Server is running");
```
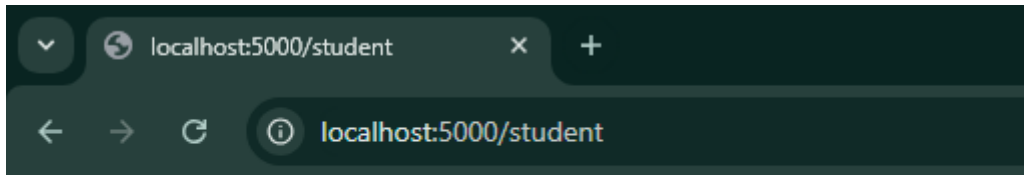
Output Screen:

```
Node.js v20.17.0
PS D:\Aradhya_WT> node Pract5
Server is running
```

Name: Arif Shaikh   **Roll No. 39**

Home page:

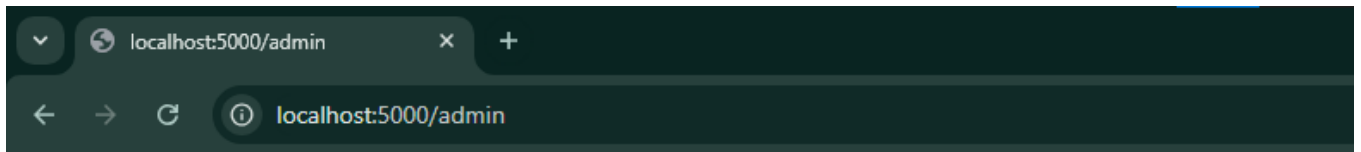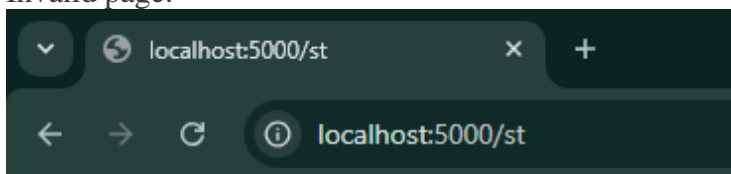# Home page

Student:

Master of Computer Applications

Admin:

Your fee structure will be displayed on the notice Board

Invalid page:

Invalid Page

Practical 6

AIM:- Write a program to demonstrate various Node.js Events.

Line of Code:

```javascript
//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 listener1(){
    console.log("Event received by Listener 1");
}
//write a function of event 2
function listener2(){
    console.log("Event received by Listener 2");
}
//step 3 adding listener through addlistener or on
eventEmitter.addListener("write",listener1);
eventEmitter.on("write",listener2);

//step 4 emiting event
eventEmitter.emit("write");
console.log(eventEmitter.listenerCount("write"));

//step 5 removing listener
eventEmitter.removeListener("write",listener1);
console.log("Listener1 is removed");
eventEmitter.emit("write");

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

Output Screen:

```
PROBLEMS    OUTPUT    TERMINAL    PORTS    GITLENS    DEBUG CONSOLE

PS C:\Users\DELL\Desktop\FYMCA> node Prac6_EventEmitter.js
Event recevied by Listener 1
Event received by Listener 2
2
Listener 1 is removed
Event received by Listener 2
1
Program Ended
PS C:\Users\DELL\Desktop\FYMCA>
```

Name: Arif Shaikh   **Roll No. 39**

Practical 7

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

Line of Code:

```
//Import events module
const events = require('events');

//create an event emitter object
const eventEmitter = new events.EventEmitter();

//
eventEmitter.on("Connection",handleConnectionEvent);

function handleConnectionEvent()
{
    console.log("Connection Made!");
}
eventEmitter.emit("Connection");
eventEmitter.emit("Connection");
eventEmitter.emit("Connection");
eventEmitter.emit("Connection");
console.log("End of Program");
```

Output Screen:

```
PROBLEMS    OUTPUT    TERMINAL    PORTS    GITLENS    DEBUG CONSOLE

PS C:\Users\DELL\Desktop\FYMCA> node Practical8.js
Connection Made!
Connection Made!
Connection Made!
Connection Made!
End of Program
PS C:\Users\DELL\Desktop\FYMCA>
```

Name: Arif Shaikh   **Roll No. 39**

Practical 8

AIM:- Using File Handling demonstrate all basic file operations (Create, write, read, delete)

Line of Code:

Read_file:

```
var fs = require('fs');
fs.readFile('asp.txt', function(err,data){if(err) throw err;
console.log(data.toString());
});
```
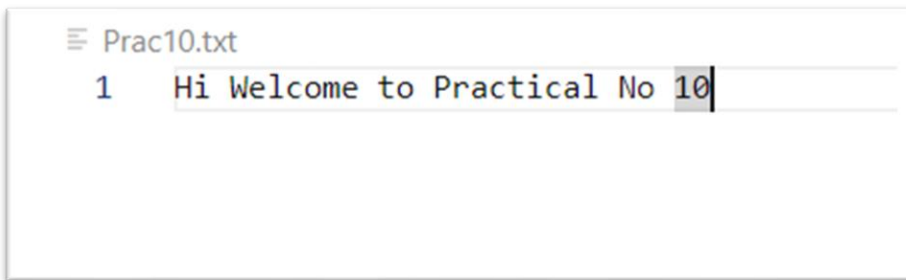
Output Screen:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Users\ADMIN\Desktop\practwt> node 9_ReadTextFile.js
This is Practical no 09
I am student of Viva Technology.
There is no one who loves pain itself,
who seeks after it and wants to have it,
simply because it is pain..
PS C:\Users\ADMIN\Desktop\practwt>
```

Write_file:

```
var fs=require('fs');
fs.writeFile('write.txt','A papal decree of 1493 had assigned all land in the New World
west of 50 degrees W longitude to Spain and all the land east of that line to Portugal.
Magellan offered to prove that the East Indies fell under Spanish authority. ',
function(err){if(err) throw err;
    else{
        console.log("Department of MCA");
    }
});
```

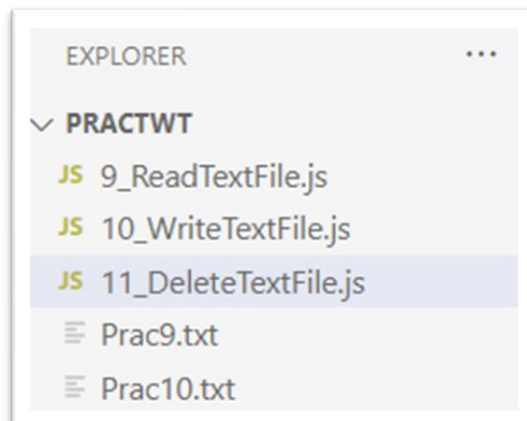**Name: Arif Shaikh**   **Roll No. 39**

Output Screen:

≡ Prac10.txt

```
1    Hi Welcome to Practical No 10
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
PS C:\Users\ADMIN\Desktop\practwt> node 10_WriteTextFile.js
I am student of Viva Technology.
PS C:\Users\ADMIN\Desktop\practwt>
```

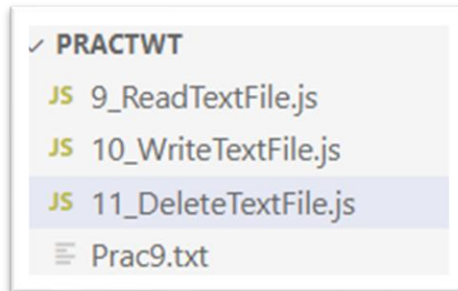EXPLORER   ...

∨ **PRACTWT**
   JS 9_ReadTextFile.js
   JS 10_WriteTextFile.js
   JS 11_DeleteTextFile.js
   ≡ Prac9.txt
   ≡ Prac10.txt

Name: Arif Shaikh   **Roll No. 39**

delete_file:

```
const fs = require('fs');
fs.unlink('delete_demo.txt',function(){
    console.log('Delete Operation completed');
});
```

✓ **PRACTWT**

JS 9_ReadTextFile.js
JS 10_WriteTextFile.js
JS 11_DeleteTextFile.js
≡ Prac9.txt

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\ADMIN\Desktop\practwt> node 11_DeleteTextFile.js
Delete Operation Completed.
PS C:\Users\ADMIN\Desktop\practwt> █

Practical 9

AIM:- Create an application to establish a connection with the MySQL database and perfrom basic database operations on it.
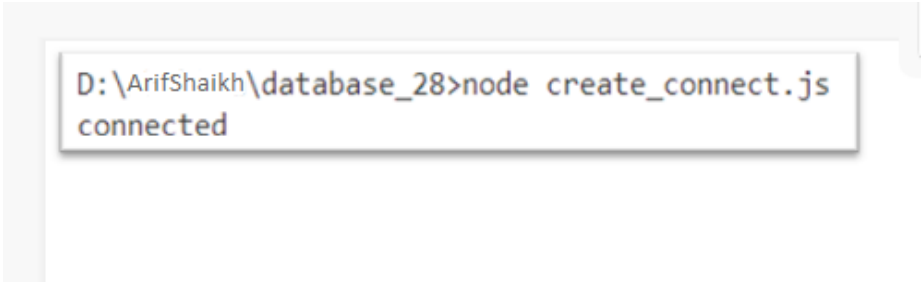
creating_connection.js:

Line of Code:

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

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

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

```
D:\ArifShaikh\database_28>node create_connect.js
connected
```

Output Screen :

creating_database.js:

Line of Code:

```
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 Asp34", function(err,result){
```
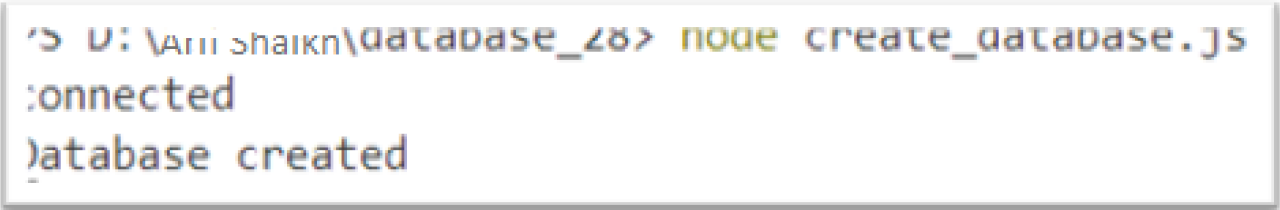
Name: Arif Shaikh   **Roll No. 39**

```
        if(err) throw err;
        console.log("Database created!");

    }) });
```

Output Screen:

```
'S D:\Arif Shaikh\database_28> node create_database.js
connected
Database created
```
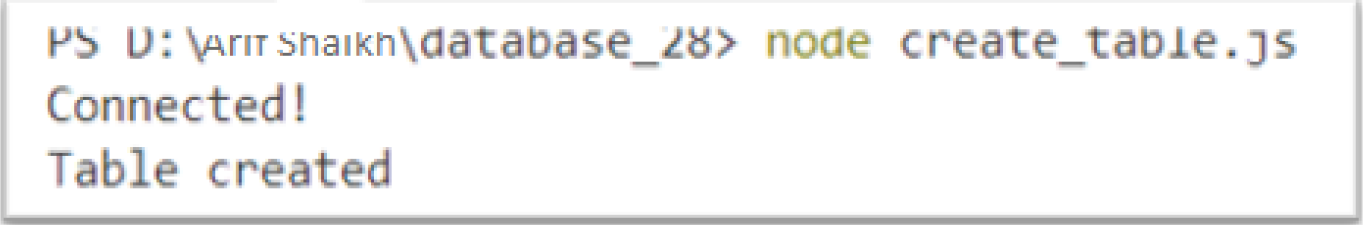
creating_table.js:

Line of Code:

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

var con = mysql.createConnection({
    host:"localhost",
    user:"root",
    password:"",
    database:"Asp34"
});

con.connect(function(err){
    if(err) throw err;
    console.log("Connected!");
    var sql = "CREATE TABLE employee(id INT,name VARCHAR(20),department VARCHAR(20))";
    con.query(sql,function(err,result){
        if(err)throw err;
        console.log("Table created");
    });
});
```

Output Screen:

```
PS D:\Arif Shaikh\database_28> node create_table.js
Connected!
Table created
```

Inserting_records.js:
Line of Code:

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

var con = mysql.createConnection({
    host:"localhost",
    user:"root",
    password:"",
    database:"Asp34"
});
con.connect(function(err){
    if(err) throw err;
    console.log("Connected!");
    var sql = "INSERT INTO employee (id,name,department) VALUES
(3,'Neena','Sales'),(4,'Lex','HR'),(5,'Raj','Accounts'),(6,'Aradhya','IT'),(7,'Yash','IT')";
    con.query(sql,function(err,result){
        if(err)throw err;
        console.log("1 record inserted");
    });
});
```

Output Screen:

```
PS D:\Arif Shaikh\database 28> node insert_record.js
Connected
5 record inserted
```
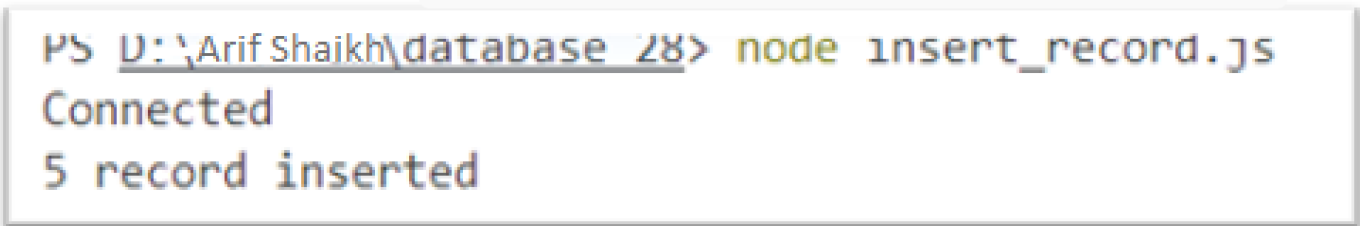
selecting_records.js:
Line of Code:

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

var con = mysql.createConnection({
    host:"localhost",
    user:"root",
    password:"",
    database:"Asp34"
});

con.connect(function(err){
    if(err) throw err;
    con.query( "SELECT * FROM employee", function(err,result,fields){
        if(err)throw err;
        console.log(result);
```

Name: Arif Shaikh   **Roll No. 39**

```
    });
});
```

Output Screen:

```
PS D:\Arif Shaikh\database_28> node selecting_record.js
[
  RowDataPacket { id: 1, name: 'company INC', address: 'Highway 32' },
  RowDataPacket { id: 2, name: 'highway ltd', address: 'sdd camp' },
  RowDataPacket { id: 3, name: 'Man ltd', address: 'edc camp' },
  RowDataPacket { id: 4, name: 'SBI ltd', address: 'uid camp' },
  RowDataPacket { id: 5, name: 'camp ltd', address: 'vir camp' }
]
```

updating_records.js:
Line of Code:

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

var con = mysql.createConnection({
    host:"localhost",
    user:"root",
    password:"",
    database:"Asp34"
});

con.connect(function(err){
    if(err) throw err;
    var sql = "UPDATE employee SET id = 8 WHERE name = 'Aradhya'";
    con.query( sql, function(err,result,fields){
        if(err)throw err;
        console.log(result.affectedRows + "record(s) updated");
    });
});
```
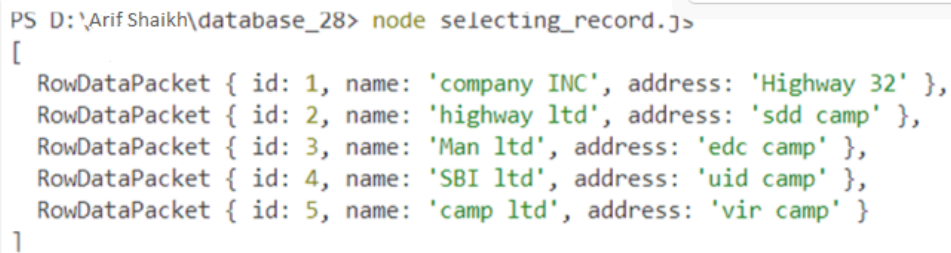
Output Screen:

```
PS D:\Arif Shaikh\database_28> node updating_record.js
1 Record(s) updated
```

Name: Arif Shaikh   **Roll No. 39**

delete_record.js:
Line of Code:

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

var con = mysql.createConnection({
    host:"localhost",
    user:"root",
    password:"",
    database:"Asp34"
});

con.connect(function(err){
    if(err) throw err;
    var sql = "DELETE from employee WHERE id = 3";
    con.query( sql, function(err,result){
        if(err)throw err;
        console.log( "Number of records deleted: " + result.affectedRows);
    });
});
```
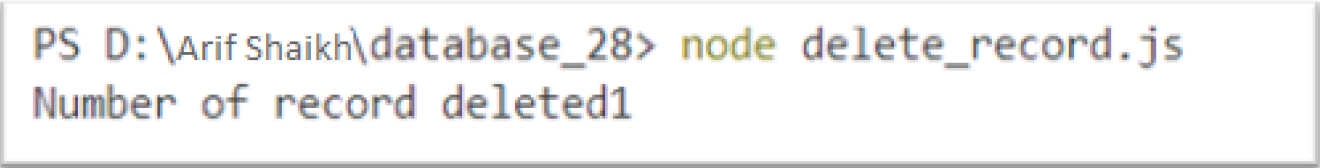
Output Screen:

```
PS D:\Arif Shaikh\database_28> node delete_record.js
Number of record deleted1
```

Practical 10

Aim: Creating application in react JS to implement the component life cycle.

Extensions used while doing react js:
1. thunder client
2. ES7 React/Redux/GraphQL/React-Native snippets
3. Bracket Pair Color DLW
4. Auto Rename Tag
5. Live Server
6. Prettier - Code formatter

React Docs :
https://create-react-app.dev/

Line of Code:

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

const LifeCycleDemo = () => {
  const [counter, setCounter] = useState(0);

  useEffect(() => {
    console.log('Component Did Mount: Component has been mounted in the DOM');

    return () => {
      console.log('Component Will Unmount: Component is about to be removed');
    };
  }, []);
  useEffect(() => {
    console.log('Component Did Update: State has changed, re-rendered');
  }, [counter]);
  const increaseCounter = () => {
    setCounter(prevCounter => prevCounter + 1);
  };

  console.log('Render: Rendering the component');

  return (
    <div>
      <h1>React Component Life Cycle</h1>
      <p>Counter: {counter}</p>
      <button onClick={increaseCounter}>Increase Counter</button>
    </div>
  );
};
```
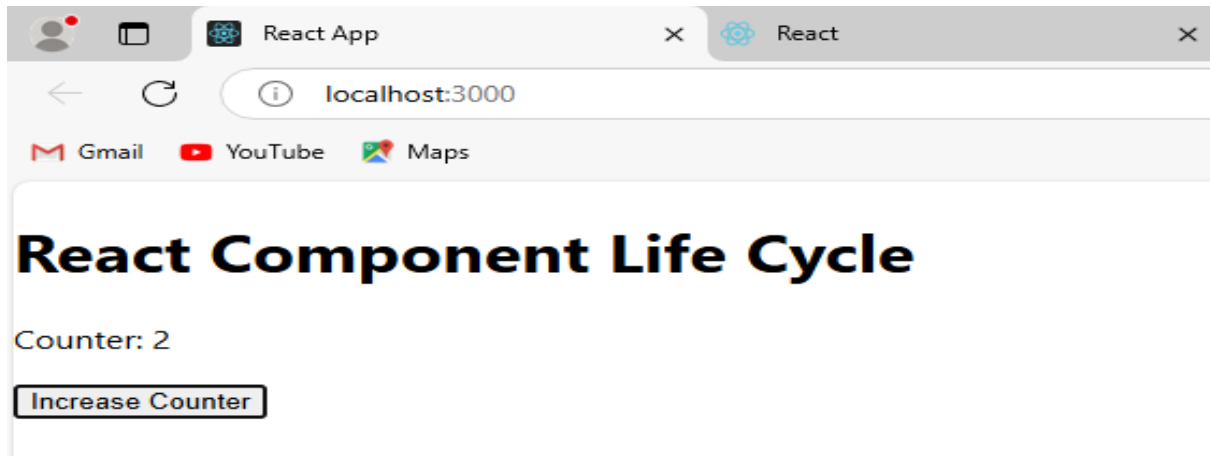
Name: Arif Shaikh   **Roll No. 39**

```
export default LifeCycleDemo;
```

Output Screen:

Practical 11

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

Difference between Class Components and Functional Components. We'll create a simple React app where both types of components are implemented and displayed on the screen.

Step 1: Setting Up the React App

1. Create React App:

    npx create-react-app react-component-demo

    cd react-component-demo
    npm start

Step 2: Creating Class and Functional Components

We'll create two separate components, one using the class syntax and one using functional syntax.

2.1 Class Component

In src/ClassComponent.js, create a class component:

```
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>
 );
```

Name: Arif Shaikh   **Roll No. 39**

```
 }
}
export default ClassComponent;
```
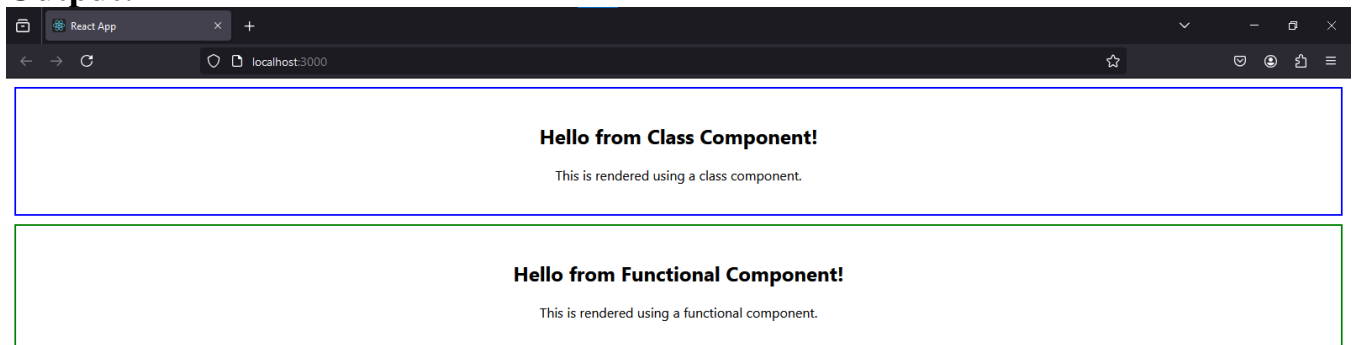
2.2 Functional Component

In src/FunctionalComponent.js, create a functional component:

```
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;
```

App.js :
```
import logo from './logo.svg';
import './App.css';
import ClassComponent from './ClassComponent';
import FunctionalComponent from './FunctionalComponent';
function App() {
 return (
   <div className="App">
    <ClassComponent/>
    <FunctionalComponent/>
   </div>
 );}
export default App;
```

Output:

Practical 12 :

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

Step 1: Setting Up the React App

npx create-react-app react-import-export-demo

cd react-import-export-demo

npm start

Step 2: Create Multiple Components

We'll create three components, each in separate files, and import them into the main App.js file.

2.1 Header Component

Create a file Header.js inside the src folder. This will be a simple functional component:

```jsx
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;
```

2.2 Footer Component

```jsx
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 © 2024</p>
</footer>
```

Name: Arif Shaikh   **Roll No. 39**

```
);
}

export default Footer;
```
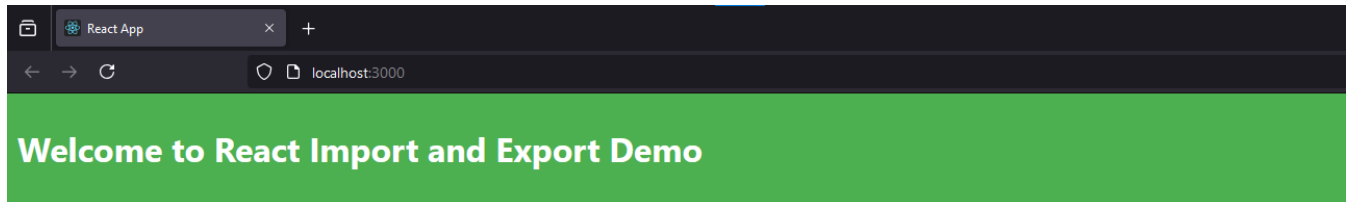
2.3 MainContent Component

```
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;
```

Open src/App.js and modify it as follows:

```
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:

Welcome to React Import and Export Demo

Main Content Area

This is the main content section where you can place any additional information or components.

React Import and Export Demo © 2024

MCAL14: Web Technologies Lab

Practical 13:

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

Step 1: Setting Up the React App

npx create-react-app react-state-props-demo
cd react-state-props-demo
npm start

ParentComponent.js:
```jsx
import React, { useState } from 'react';
import ChildComponent from './ChildComponent';
function ParentComponent() {
 // Define state
 const [message, setMessage] = useState('Click For React info');
 const updateMessage = () => {
 setMessage('React allows developers to create user interfaces by combining components,
which are self-contained pieces of code that describe a portion of the user interface.
React abstracts away much of the rendering work, allowing developers to focus on the UI
design. ');
 };
 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;
```

ChildComponent.js:
```jsx
import React from 'react';
function ChildComponent({ message, updateMessage }) {
 return (
 <div style={{ border: '2px solid green', padding: '20px', margin: '10px'
}}>  <h2>Child Component</h2>
 <p>React info: {message}</p>
 <button onClick={updateMessage}>Click For React info</button>  </div>
 );
}
export default ChildComponent;
```
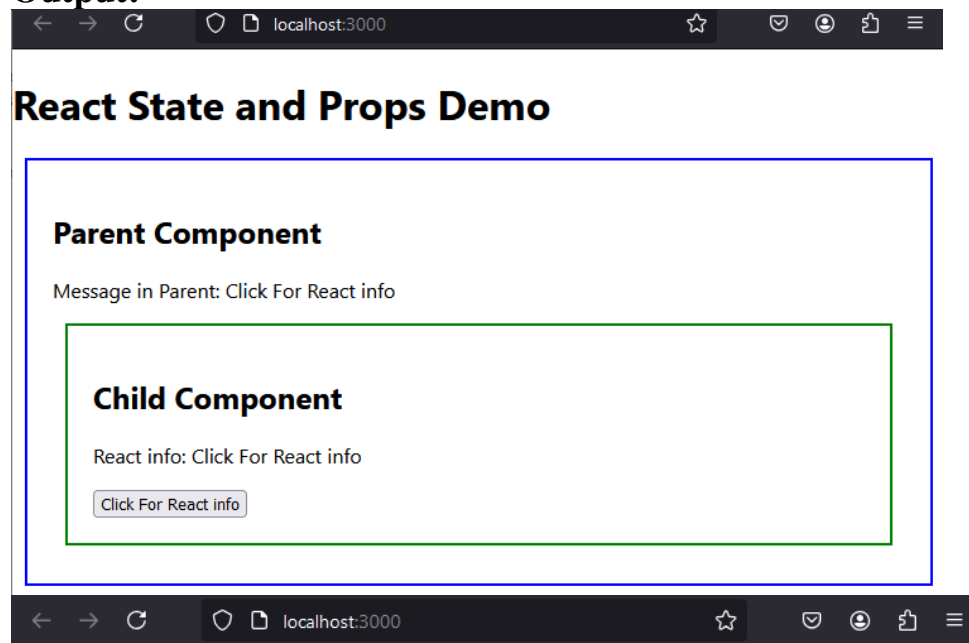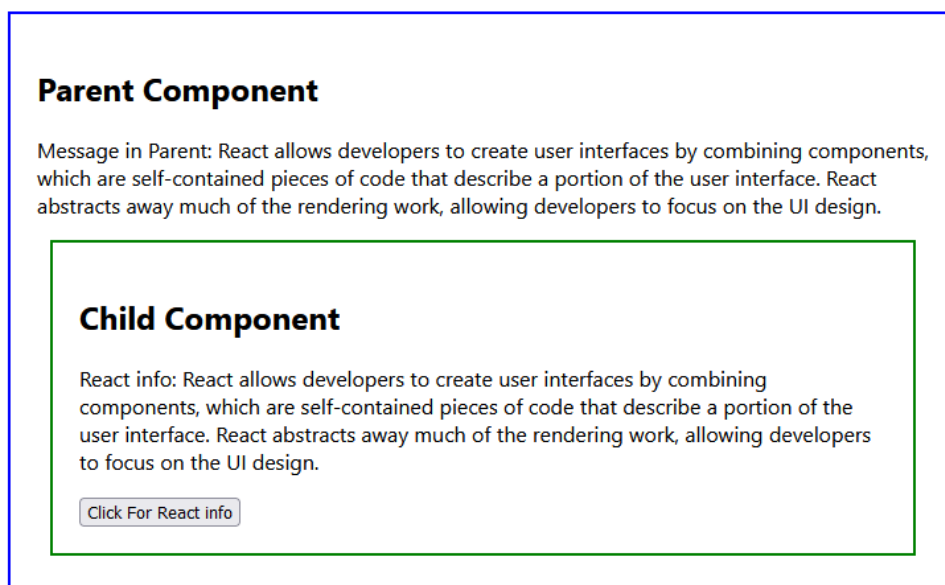
Name: Arif Shaikh   **Roll No. 39**

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:

Name: Arif Shaikh

Roll No. 39

Practical 14:

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

DOMEventsComponent.js:

```jsx
import React, { useState } from 'react';
function DOMEventsComponent() {
 const [inputValue, setInputValue] = useState('');
 const [hovered, setHovered] = useState(false);
 // Handle button click
 const handleClick = () => {
 alert('Order Submitted!');
 };
 // 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>Menu</h2>
 {/* onClick Event */}
 <button onClick={handleClick} style={{ padding: '10px', fontSize: '16px' }}> Submit
Order
 </button>
 {/* onChange Event */}
 <div style={{ margin: '20px 0' }}>
 <input
 type="text"
 placeholder="Type a Dish. ... "
 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',
```

Name: Arif Shaikh   **Roll No. 39**

```
cursor: 'pointer',
}}
>
{hovered ? 'Free, Butter Chicken for you!' : 'Hover for surprise'}
</div>
</div>
);
}
export default DOMEventsComponent;
```

App.js :

```
import React from 'react';
import DOMEventsComponent from './DOMEventsComponent';
function App() {
 return (
 <div className="App">
 <h1>Scan&Dine Demo</h1>
 <DOMEventsComponent />
 </div>
 );
}
export default App;
```

Output:

Scan&Dine Demo

Menu

Submit Order

Gulab Jamun|

You typed: Gulab Jamun

Scan&Dine Demo

Menu

Submit Order

Gulab Jamun

You typed: Gulab Jamun

⊕ localhost:3000

Order Submitted!

OK

Scan&Dine Demo

Menu

Submit Order

Gulab Jamun

You typed: Gulab Jamun

Hover for surprise

← → C ○ □ localhost:3000 ☆ ☑ ☺ ☰

Scan&Dine Demo

Menu

Submit Order

Gulab Jamun

You typed: Gulab Jamun

Free, Butter Chicken for you!

Name: Arif Shaikh   **Roll No. 39**
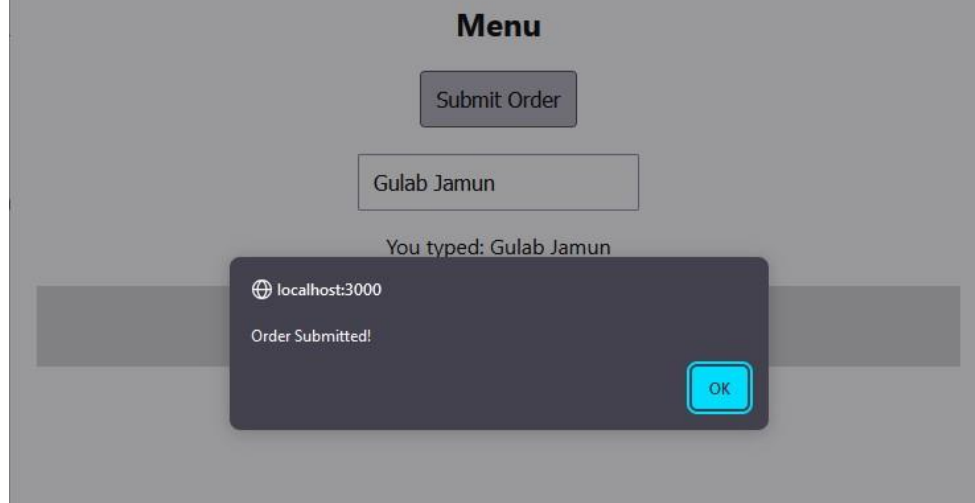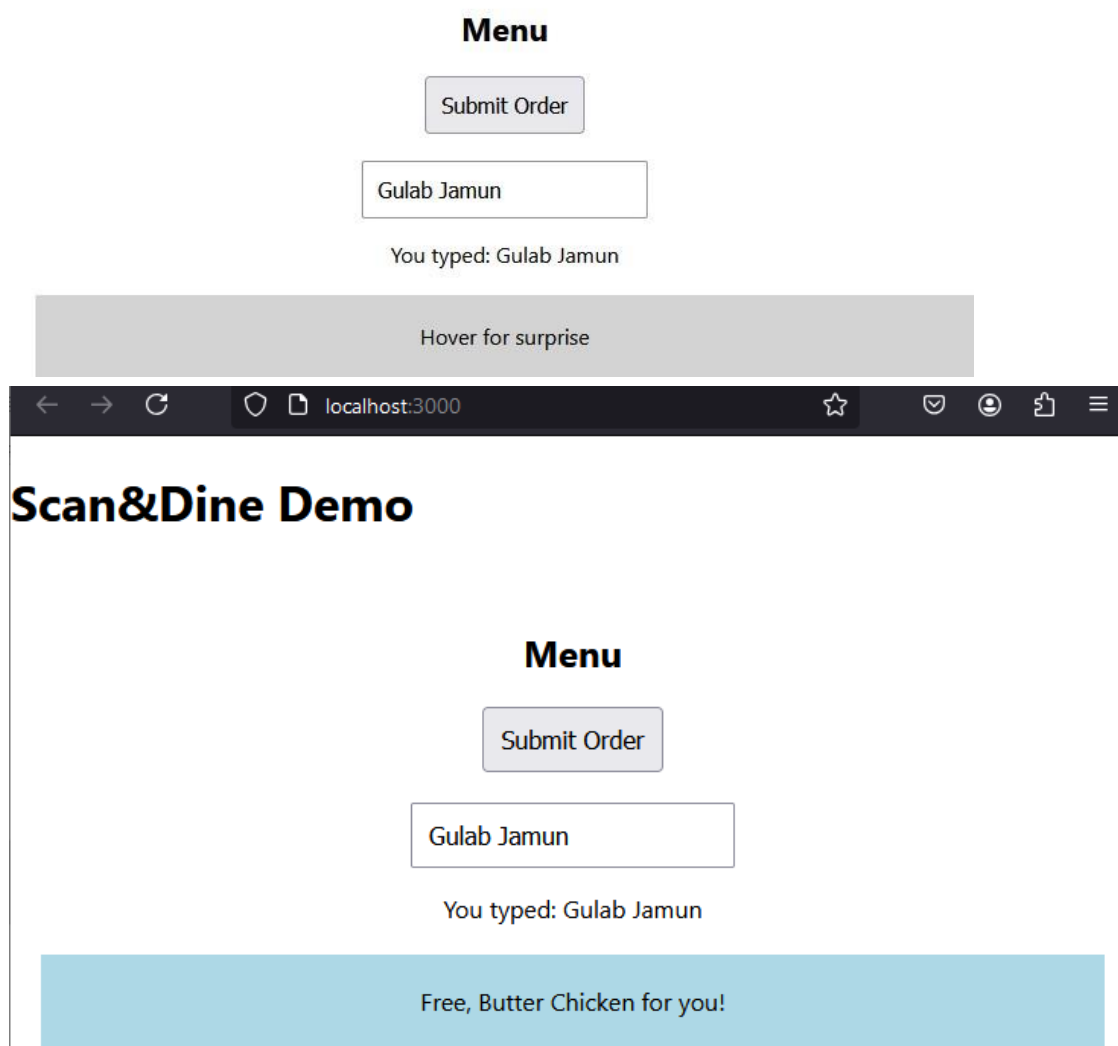
Practical 15

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

2.1 Create a RegistrationForm.js

```javascript
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
```

Name: Arif Shaikh   **Roll No. 39**

```jsx
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>Please Login Here</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' }}
```

Name: Arif Shaikh   **Roll No. 39**

```
/>
{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;
```
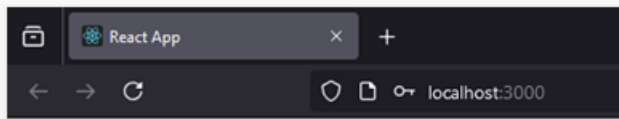
Step 3: Modify App Component
3.1 Update App.js

```
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;
```

Step 4: Running the App

npm start

Output:

React Form with Validation

Please Login Here

Form submitted successfully!

Practical 16

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

2.1 Create the Components

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 (
```

Name: Arif Shaikh   **Roll No. 39**

```
<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

3.1 Update 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:

React Routing Demo

Home   About   Contact

Welcome to the Home Page

This is the main page of our application.

React Routing Demo

Home   About   Contact

About Us

This is the about page where we describe our app.

React Routing Demo

Home   About   Contact

Contact Us

This is the contact page for inquiries.

Practical No. 17

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

DataFetching.js:

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

ThemeContext.js

```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>
);
};
```

ThemedComponent.js
```js
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
```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>
);
```

Name: Arif Shaikh   **Roll No. 39**

```
}
export default App;
```

Output:

React Hooks

Counter

Current Count: 0

[Increase] [Decrease]

Data Fetching

sunt aut facere repellat provident occaecati excepturi optio reprehenderit
qui est esse
ea molestias quasi exercitationem repellat qui ipsa sit aut
eum et est occaecati
nesciunt quas odio

-
-
-
-
-

Theme Toggle

Current Theme: Light

[Toggle Theme]