

**JOURNAL**

**ADVANCED WEB PROGRAMMING**

**SUBMITTED BY**

**DEVDA BHUMI**

**SYDS**

**SDDS004A**

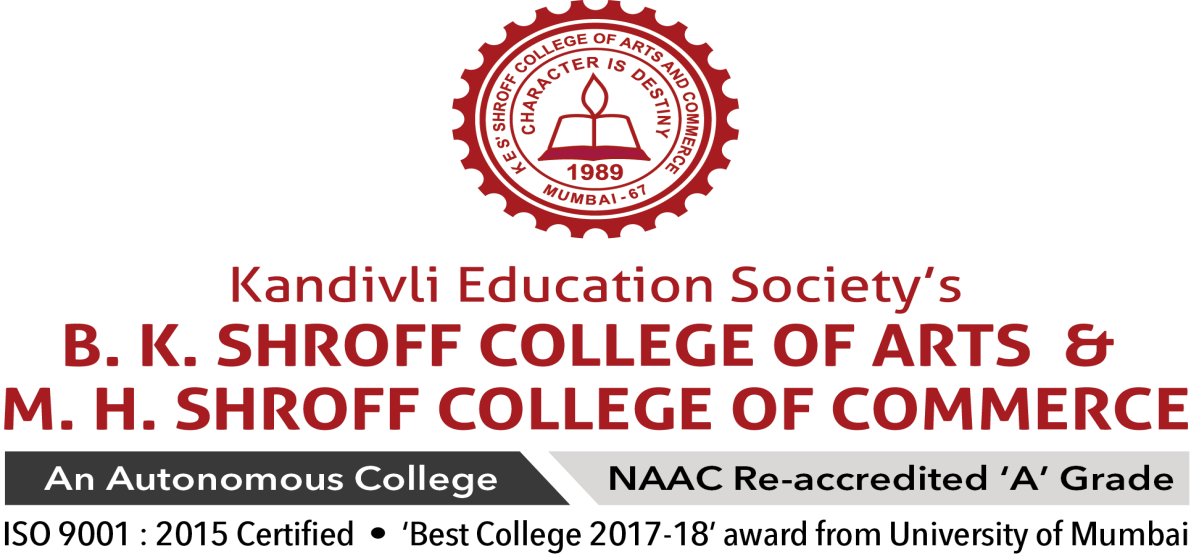
**SEMESTER IV**

**UNDER THE GUIDANCE OF**

**ASST. PROF. HIRAL SACHALA**

**ACADEMIC YEAR**

**2023 - 2024**



# CERTIFICATE

This is to certify that Ms. Devda Bhumi of Second Year BSc. Data Science Div A, roll number SDDS004A, Semester IV (2023 - 2024) has successfully completed the Journal for the course Advanced web programming as per the guidelines of KES’ Shroff College of Arts and Commerce, Kandivali (W), Mumbai 400067.

**Teacher In-charge**   **Principal**

**ASST. PROF. Hiral Sachala** **Dr. L. Bhushan**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Exp No.** | **Experiment** | **Page Nos.** |
| **1.** | Create an application to demonstrate Node.js Modules | 4, 5 |
| **2.** | Create an application to demonstrate various Node.js Events | 6, 7 |
| **3.** | Create an application to demonstrate Node.js Functions | 8-10 |
| **4.** | Using File Handling demonstrate all basic file operations (Create, write, read, delete) | 11, 12 |
| **5.** | Create an HTTP Server and perform operations on it | 13 |
| **6.** | Create an application to establish a connection with the MySQL database and perform basic database operations on it 7 Create an application using | 14-16 |
| **7.** | Create an application using Filters | 17-21 |
| **8.** | Create an application to demonstrate directives | 22-24 |
| **9.** | Demonstrate controllers in Angular.js through an application | 25-27 |
| **10.** | Demonstrate features of React.js forms with a program | 28, 29 |
| **11.** | Create a Login Page using React.js | 30-34 |

**Practical 1**

**Aim:** **Create an application to demonstrate Node.js Modules**

Code:

//Create an application to demonstrate Node.js Modules

buf = new Buffer(26);

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

buf[i] = i+97;

}

console.log(buf.toString('ascii'));

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

console.log(buf.toString('utf8',0,5));

console.log(buf.toString(undefined,0,5));

//Implementing concat buffer

var buffer1 = new Buffer('TutorialsPoint');

var buffer2 = new Buffer('Simple Easy Learning');

var buffer5 = Buffer.concat([buffer1,buffer2]);

var buffer6 = buffer1.compare(buffer2);

console.log("buffer5 content: " + buffer5.toString());

//implement Compare buffer

var buffer3 = new Buffer('ABC');

var buffer4 = new Buffer('ABCD');

var result = buffer3.compare(buffer4);

if(result < 0) {

console.log(buffer3 + " comes before " + buffer4);

}else if(result == 0) {

console.log(buffer3 + " is same as " + buffer4);

}else {

console.log(buffer3 + " comes after " + buffer4);

}

//Practical 1

//Create an application to demonstrate Node.js Modules

buf = new Buffer(26);

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

    buf[i] = i+97;

}

console.log(buf.toString('ascii'));

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

console.log(buf.toString('utf8',0,5));

console.log(buf.toString(undefined,0,5));

//Implementing concat buffer

var buffer1 = new Buffer('TutorialsPoint');

var buffer2 = new Buffer('Simple Easy Learning');

var buffer5 = Buffer.concat([buffer1,buffer2]);

var buffer6 = buffer1.compare(buffer2);

console.log("buffer5 content: " + buffer5.toString());

//implement Compare buffer

var buffer3 = new Buffer('ABC');

var buffer4 = new Buffer('ABCD');

var result = buffer3.compare(buffer4);

if(result < 0) {

    console.log(buffer3 + " comes before " + buffer4);

}else if(result == 0) {

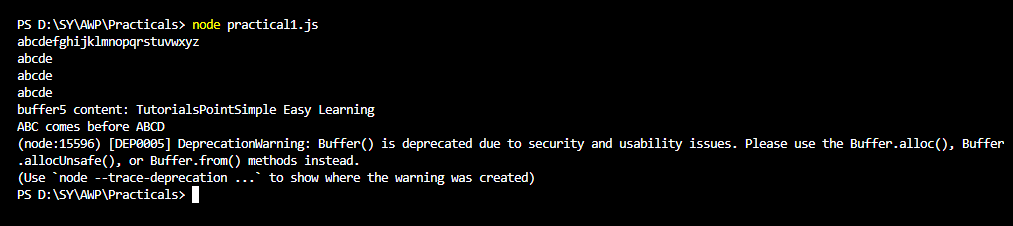
    console.log(buffer3 + " is same as " + buffer4);

}else {

    console.log(buffer3 + " comes after " + buffer4);

}

**Output**



**Practical 2**

**Aim:** **Create an application to demonstrate various Node.js Events.**

Code:

//EVENT EMITTER

var events = require('events');

var eventEmitter = new events.EventEmitter();

//listener #1

var listner1 = function listner1() {

console.log('listener1 executed.');

}

//listener #2

var listner2 = function listner2() {

console.log('listener2 executed.');

}

//Bind the connection event with the listner1 function

eventEmitter.addListener('connection',listner1);

//Bind the connection event with the listner2 function

eventEmitter.on('connection',listner2);

var eventListeners = require('events').EventEmitter.listenerCount(eventEmitter,'connection');

console.log(eventListeners + "Listner(s) listening to connection event");

//Fire the connection event

eventEmitter.emit('connection');

eventListeners = require('events').EventEmitter.listenerCount(eventEmitter,'connection');

console.log(eventListeners + "Listner(s) listening to connection event");

console.log("Program Ended.");

//Practical 2

//EVENT EMITTER

var events = require('events');

var eventEmitter = new events.EventEmitter();

//listener #1

var listner1 = function listner1() {

    console.log('listener1 executed.');

}

//listener #2

var listner2 = function listner2() {

    console.log('listener2 executed.');

}

//Bind the connection event with the listner1 function

eventEmitter.addListener('connection',listner1);

//Bind the connection event with the listner2 function

eventEmitter.on('connection',listner2);

var eventListeners = require('events').EventEmitter.listenerCount(eventEmitter,'connection');

console.log(eventListeners + "Listner(s) listening to connection event");

//Fire the connection event

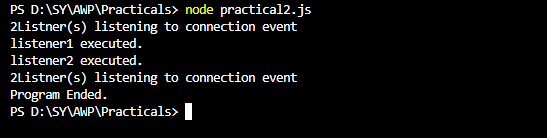
eventEmitter.emit('connection');

eventListeners = require('events').EventEmitter.listenerCount(eventEmitter,'connection');

console.log(eventListeners + "Listner(s) listening to connection event");

console.log("Program Ended.");

**Output**



**Practical 3**

**Aim:** **Create an application to demonstrate Node.js Functions**

Code:

//Import required modules

const readline = require('readline');

//Create a readline interface to take user input

const rl = readline.createInterface({

input: process.stdin,

output: process.stdout

});

//Function to add two numbers

function add(a,b) {

return a + b;

}

//Function to subtract two numbers

function subtract(a,b) {

return a - b;

}

//Function to multiply two numbers

function multiply(a,b) {

return a \* b;

}

//Function to divide two numbers

function divide(a,b) {

if (b !== 0){

return a / b;

}else {

return 'Error: Cannot divide by zero';

}

}

//Function to perform calculator operations based on user input

function calculator() {

rl.question('Enter first number:', (num1) => {

rl.question('Enter second number:', (num2) => {

num1 = parseFloat(num1);

num2 = parseFloat(num2);

console.log('\nResults:');

console.log('-Sum:', add(num1,num2));

console.log('-Difference:', subtract(num1,num2));

console.log('-Product:', multiply(num1,num2));

console.log('-Quotient:', divide(num1,num2));

rl.close();

});

});

}

//Call the calculator function to start the program

calculator();

//Practical 3

//Import required modules

const readline = require('readline');

//Create a readline interface to take user input

const rl = readline.createInterface({

    input: process.stdin,

    output: process.stdout

});

//Function to add two numbers

function add(a,b) {

    return a + b;

}

//Function to subtract two numbers

function subtract(a,b) {

    return a - b;

}

//Function to multiply two numbers

function multiply(a,b) {

    return a \* b;

}

//Function to divide two numbers

function divide(a,b) {

    if (b !== 0){

        return a / b;

    }else {

        return 'Error: Cannot divide by zero';

    }

}

//Function to perform calculator operations based on user input

function calculator() {

    rl.question('Enter first number:', (num1) => {

        rl.question('Enter second number:', (num2) => {

            num1 = parseFloat(num1);

            num2 = parseFloat(num2);

            console.log('\nResults:');

            console.log('-Sum:', add(num1,num2));

            console.log('-Difference:', subtract(num1,num2));

            console.log('-Product:', multiply(num1,num2));

            console.log('-Quotient:', divide(num1,num2));

            rl.close();

        });

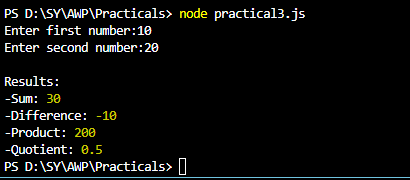
    });

}

//Call the calculator function to start the program

calculator();

**Output**



**Practical 4**

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

Code:

//Create an application to demonstrate various Node.js Events

//READ AND WRITE FILE

var http = require('http');

var fs = require('fs');

http.createServer(function(req,res) {

//Open a file on the server and return its content:

fs.readFile('demofile1.html', function(err, data) {

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

res.write(data);

return res.end();

});

});

//CREATE FILE

var fs = require('fs');

fs.appendFile('mynewfile1.txt','Hello content', function(err) {

if (err) throw err;

console.log('Saved');

});

var fs = require('fs');

fs.open('mynewfile2.txt','w', function(err, file) {

if (err) throw err;

console.log('Saved');

});

var fs = require('fs');

fs.writeFile('mynewfile3.txt','Hello content',function(err) {

if (err) throw err;

console.log('Saved');

});

//DELETE FILE

var fs = require('fs');

fs.unlink('mynewfile2.txt',function (err) {

if (err) throw err;

console.log('File deleted');

});

console.log('Server running at http://127.0.0.1:8081/');

//Practical 4

//Create an application to demonstrate various Node.js Events

//READ AND WRITE FILE

var http = require('http');

var fs = require('fs');

http.createServer(function(req,res) {

    //Open a file on the server and return its content:

    fs.readFile('demofile1.html', function(err, data) {

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

        res.write(data);

        return res.end();

    });

});

//CREATE FILE

var fs = require('fs');

fs.appendFile('mynewfile1.txt','Hello content', function(err) {

    if (err) throw err;

    console.log('Saved');

});

var fs = require('fs');

fs.open('mynewfile2.txt','w', function(err, file) {

    if (err) throw err;

    console.log('Saved');

});

var fs = require('fs');

fs.writeFile('mynewfile3.txt','Hello content',function(err) {

    if (err) throw err;

    console.log('Saved');

});

//DELETE FILE

var fs = require('fs');

fs.unlink('mynewfile2.txt',function (err) {

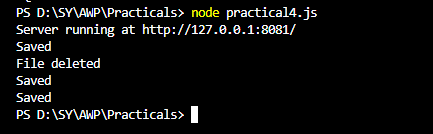
    if (err) throw err;

    console.log('File deleted');

});

console.log('Server running at http://127.0.0.1:8081/');

**Output**



**Practical 5**

**Aim:** **Create an HTTP Server and perform operations on it**

Code:

var http = require("http");

http.createServer(function (request, response) {

//Send the http header

//http status: 200 : OK

//Content Type: text/plain

response.writeHead(200, {'Content-Type': 'text/plain'});

//Send the response body as "Hello World"

response.end('Hello World\n');

}).listen(8081);

//Console will print the message

console.log('Server running at http://127.0.0.1:8081/');

//Practical 5

var http = require("http");

http.createServer(function (request, response) {

    //Send the http header

    //http status: 200 : OK

    //Content Type: text/plain

    response.writeHead(200, {'Content-Type': 'text/plain'});

    //Send the response body as "Hello World"

    response.end('Hello World\n');

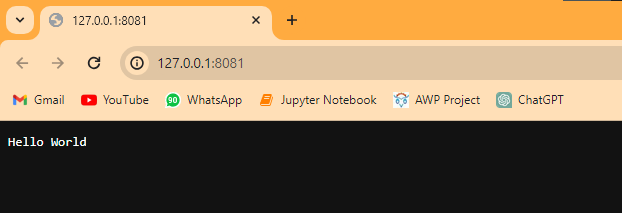
}).listen(8081);

//Console will print the message

console.log('Server running at http://127.0.0.1:8081/');

**Output**





**Practical 6**

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

Code:

var mysql = require("mysql");

var con = mysql.createConnection({

host: "localhost",

user: "root",

password: "student", //your mysql passwords

database: "SYDS",

});

//Create database

con.connect(function(err) {

if (err) throw err;

console.log("Connected!");

con.query("CREATE DATABASE SYDS", function (err, result) {

if (err) throw err;

console.log("Database created");

});

});

//Create table

con.connect(function(err) {

if (err) throw err;

console.log("Connected!");

var sql = "CREATE TABLE Student ( rollno int,name VARCHAR(255), address VARCHAR(255))";

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

if (err) throw err;

console.log("Table created");

});

});

//Insert data

con.connect(function(err) {

if (err) throw err;

console.log("Connected!");

var sql = "INSERT INTO Student ( rollno ,name, address) VALUES ?";

var values = [

[1,'John', 'Borivali'],

[2,'Peter', 'Kandivali 4'],

[3,'Amy', 'Dadar'],

];

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

if (err) throw err;

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

});

});

var mysql = require("mysql");

var con = mysql.createConnection({

  host: "localhost",

  user: "root",

  password: "student", //your mysql passwords

  database: "SYDS",

});

//Create database

con.connect(function(err) {

  if (err) throw err;

  console.log("Connected!");

  con.query("CREATE DATABASE SYDS", function (err, result) {

    if (err) throw err;

    console.log("Database created");

  });

});

//Create table

con.connect(function(err) {

  if (err) throw err;

  console.log("Connected!");

  var sql = "CREATE TABLE Student ( rollno int,name VARCHAR(255), address VARCHAR(255))";

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

    if (err) throw err;

    console.log("Table created");

  });

});

//Insert data

con.connect(function(err) {

  if (err) throw err;

  console.log("Connected!");

  var sql = "INSERT INTO Student ( rollno ,name, address) VALUES ?";

  var values = [

    [1,'John', 'Borivali'],

    [2,'Peter', 'Kandivali 4'],

    [3,'Amy', 'Dadar'],

  ];

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

    if (err) throw err;

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

  });

});

//Select From

con.connect(function(err) {

  if (err) throw err;

  con.query("SELECT \* FROM Student", function (err, result, fields) {

    if (err) throw err;

    console.log(result);

  });

});

**Output**

****

**Practical 7**

**Aim:** **Create an application using Filters.**

Code:

<html>

<head>

<title>Angular JS Filters</title>

<script src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js">

</script>

</head>

<body>

<h2>AngularJS Sample Application</h2>

<div ng-app = "mainApp" ng-controller = "studentController">

<table border = "0">

<tr>

<td>Enter first name:</td>

<td><input type = "text" ng-model = "student.firstName"></td>

</tr>

<tr>

<td>Enter last name: </td>

<td><input type = "text" ng-model = "student.lastName"></td>

</tr>

<tr>

<td>Enter fees: </td>

<td><input type = "text" ng-model = "student.fees"></td>

</tr>

<tr>

<td>Enter subject: </td>

<td><input type = "text" ng-model = "subjectName"></td>

</tr>

</table>

<br/>

<table border = "0">

<tr>

<td>Name in Upper Case: </td><td>{{student.fullName() | uppercase}}</td>

</tr>

<tr>

<td>Name in Lower Case: </td><td>{{student.fullName() | lowercase}}</td>

</tr>

<tr>

<td>fees: </td><td>{{student.fees | currency}}

</td>

</tr>

<tr>

<td>Subject:</td>

<td>

<ul>

<li ng-repeat = "subject in student.subjects | filter: subjectName |orderBy:'marks'">

{{ subject.name + ', marks:' + subject.marks }}

</li>

</ul>

</td>

</tr>

</table>

</div>

<script>

var mainApp = angular.module("mainApp", []);

mainApp.controller('studentController', function($scope) {

$scope.student = {

firstName: "Mahesh",

lastName: "Parashar",

fees:500,

subjects:[

{name:'Physics',marks:70},

{name:'Chemistry',marks:80},

{name:'Math',marks:65}

],

fullName: function() {

var studentObject;

studentObject = $scope.student;

return studentObject.firstName + " " + studentObject.lastName;

}

};

});

</script>

</body>

</html>

//Select From

con.connect(function(err) {

if (err) throw err;

con.query("SELECT \* FROM Student", function (err, result, fields) {

if (err) throw err;

console.log(result);

});

});

<!--Practical 7-->

<html>

   <head>

      <title>Angular JS Filters</title>

      <script src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js">

      </script>

   </head>

   <body>

      <h2>AngularJS Sample Application</h2>

      <div ng-app = "mainApp" ng-controller = "studentController">

         <table border = "0">

            <tr>

               <td>Enter first name:</td>

               <td><input type = "text" ng-model = "student.firstName"></td>

            </tr>

            <tr>

               <td>Enter last name: </td>

               <td><input type = "text" ng-model = "student.lastName"></td>

            </tr>

            <tr>

               <td>Enter fees: </td>

               <td><input type = "text" ng-model = "student.fees"></td>

            </tr>

            <tr>

               <td>Enter subject: </td>

               <td><input type = "text" ng-model = "subjectName"></td>

            </tr>

         </table>

         <br/>

         <table border = "0">

            <tr>

               <td>Name in Upper Case: </td><td>{{student.fullName() | uppercase}}</td>

            </tr>

            <tr>

               <td>Name in Lower Case: </td><td>{{student.fullName() | lowercase}}</td>

            </tr>

            <tr>

               <td>fees: </td><td>{{student.fees | currency}}

               </td>

            </tr>

            <tr>

               <td>Subject:</td>

               <td>

                  <ul>

                     <li ng-repeat = "subject in student.subjects | filter: subjectName |orderBy:'marks'">

                        {{ subject.name + ', marks:' + subject.marks }}

                     </li>

                  </ul>

               </td>

            </tr>

         </table>

      </div>

      <script>

         var mainApp = angular.module("mainApp", []);

         mainApp.controller('studentController', function($scope) {

            $scope.student = {

               firstName: "Mahesh",

               lastName: "Parashar",

               fees:500,

               subjects:[

                  {name:'Physics',marks:70},

                  {name:'Chemistry',marks:80},

                  {name:'Math',marks:65}

               ],

               fullName: function() {

                  var studentObject;

                  studentObject = $scope.student;

                  return studentObject.firstName + " " + studentObject.lastName;

               }

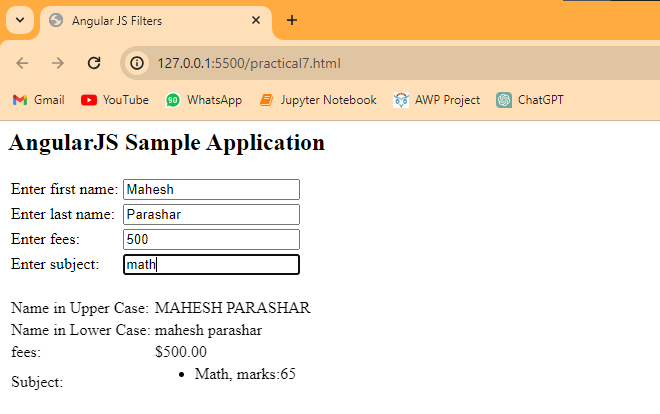
            };

         });

      </script>

   </body>

</html>

**Output:**

**Practical 8**

**Aim:** **Create an application to demonstrate directives.**

Code:

<html>

<head>

<title>Angular JS Custom Directives</title>

</head>

<body>

<h2>AngularJS Sample Application</h2>

<div ng-app = "mainApp" ng-controller = "StudentController">

<student name = "Mahesh"></student><br/>

<student name = "Piyush"></student>

</div>

<script src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js">

</script>

<script>

var mainApp = angular.module("mainApp", []);

mainApp.directive('student', function() {

var directive = {};

directive.restrict = 'E';

directive.template = "Student: <b>{{student.name}}</b> , Roll No: <b>{{student.rollno}}</b>";

directive.scope = {

student : "=name"

}

directive.compile = function(element, attributes) {

element.css("border", "1px solid #cccccc");

var linkFunction = function($scope, element, attributes) {

element.html("Student: <b>"+$scope.student.name +"</b> ,Roll No: <b>"+$scope.student.rollno+"</b><br/>");

element.css("background-color", "#ff00ff");

}

return linkFunction;

}

return directive;

});

mainApp.controller('StudentController', function($scope) {

$scope.Mahesh = {};

$scope.Mahesh.name = "Mahesh Parashar";

$scope.Mahesh.rollno = 1;

$scope.Piyush = {};

$scope.Piyush.name = "Piyush Parashar";

$scope.Piyush.rollno = 2;

});

</script>

</body>

</html>

<html>

 <head>

 <title>Angular JS Custom Directives</title>

 </head>

 <body>

 <h2>AngularJS Sample Application</h2>

 <div ng-app = "mainApp" ng-controller = "StudentController">

 <student name = "Mahesh"></student><br/>

 <student name = "Piyush"></student>

 </div>

 <script src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js">

 </script>

 <script>

 var mainApp = angular.module("mainApp", []);

 mainApp.directive('student', function() {

 var directive = {};

 directive.restrict = 'E';

 directive.template = "Student: <b>{{student.name}}</b> , Roll No: <b>{{student.rollno}}</b>";

 directive.scope = {

 student : "=name"

 }

 directive.compile = function(element, attributes) {

 element.css("border", "1px solid #cccccc");

 var linkFunction = function($scope, element, attributes) {

 element.html("Student: <b>"+$scope.student.name +"</b> ,Roll No: <b>"+$scope.student.rollno+"</b><br/>");

 element.css("background-color", "#ff00ff");

 }

 return linkFunction;

 }

 return directive;

 });

 mainApp.controller('StudentController', function($scope) {

 $scope.Mahesh = {};

 $scope.Mahesh.name = "Mahesh Parashar";

 $scope.Mahesh.rollno = 1;

 $scope.Piyush = {};

 $scope.Piyush.name = "Piyush Parashar";

 $scope.Piyush.rollno = 2;

 });

 </script>

 </body>

</html>

**Output**

****

**Practical 9**

**Aim:** **Demonstrate controllers in Angular.js through an application.**

Code:

<html>

<head>

<title>Angular JS Controller</title>

<script src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js">

</script>

</head>

<body>

<h2>AngularJS Sample Application</h2>

<div ng-app = "mainApp" ng-controller = "studentController">

Enter first name: <input type = "text" ng-model = "student.firstName"><br>

<br>

Enter last name: <input type = "text" ng-model = "student.lastName"><br>

<br>

You are entering: {{student.fullName()}}

</div>

<script>

var mainApp = angular.module("mainApp", []);

mainApp.controller('studentController', function($scope) {

$scope.student = {

firstName: "Mahesh",

lastName: "Parashar",

fullName: function() {

var studentObject;

studentObject = $scope.student;

return studentObject.firstName + " " + studentObject.lastName;

}

};

});

</script>

</body>

</html>

<!--Practical 9-->

<html>

   <head>

      <title>Angular JS Controller</title>

      <script src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js">

      </script>

   </head>

   <body>

      <h2>AngularJS Sample Application</h2>

      <div ng-app = "mainApp" ng-controller = "studentController">

         Enter first name: <input type = "text" ng-model = "student.firstName"><br>

         <br>

         Enter last name: <input type = "text" ng-model = "student.lastName"><br>

         <br>

         You are entering: {{student.fullName()}}

      </div>

      <script>

         var mainApp = angular.module("mainApp", []);

         mainApp.controller('studentController', function($scope) {

            $scope.student = {

               firstName: "Mahesh",

               lastName: "Parashar",

               fullName: function() {

                  var studentObject;

                  studentObject = $scope.student;

                  return studentObject.firstName + " " + studentObject.lastName;

               }

            };

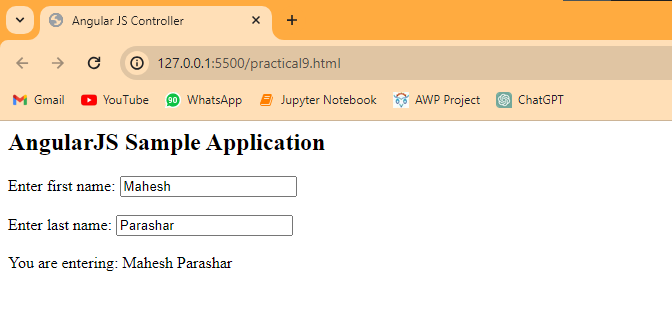
         });

      </script>

   </body>

</html>

**Output**



**Practical 10**

**Aim:** **Demonstrate features of React.js forms with a program.**

Code:

import React, { Component } from 'react';

class App extends React.Component {

constructor(props) {

super(props);

this.state = {

companyName: ''

};

}

changeText(event) {

this.setState({

companyName: event.target.value

});

}

render() {

return (

<div>

<h2>Simple Event Example</h2>

<label htmlFor="name">Enter company name: </label>

<input type="text" id="companyName" onChange={this.changeText.bind(this)}/>

<h4>You entered: { this.state.companyName }</h4>

</div>

);

}

}

export default App;

//Practical 10

import React, { Component } from 'react';

class App extends React.Component {

    constructor(props) {

        super(props);

        this.state = {

            companyName: ''

        };

    }

    changeText(event) {

        this.setState({

            companyName: event.target.value

        });

    }

    render() {

        return (

        <div>

        <h2>Simple Event Example</h2>

        <label htmlFor="name">Enter company name: </label>

        <input type="text" id="companyName" onChange={this.changeText.bind(this)}/>

        <h4>You entered: { this.state.companyName }</h4>

        </div>

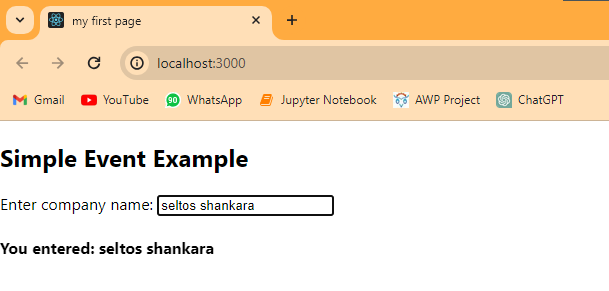
        );

    }

}

export default App;

**Output**



**Practical 11**

**Aim:** **Create a login page using React.js**

Code:

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

import "./App.css";

import Home from "./components/Home/Home.jsx";

import Login from "./components/Login/Login.jsx";

import Navbar from "./components/Navbar/Navbar.jsx";

function App() {

return (

<div className="App">

<BrowserRouter>

<Navbar />

<Routes>

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

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

</Routes>

</BrowserRouter>

</div>

);

}

export default App;

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

import "./App.css";

import Home from "./components/Home/Home.jsx";

import Login from "./components/Login/Login.jsx";

import Navbar from "./components/Navbar/Navbar.jsx";

function App() {

  return (

    <div className="App">

      <BrowserRouter>

        <Navbar />

        <Routes>

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

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

        </Routes>

      </BrowserRouter>

    </div>

  );

}

export default App;

Home.jsx

import React from 'react'

const Home = () => {

return (

<div className='home-background'>

<h3>Home Page</h3>

</div >

)

}

export default Home;

import React from 'react'

const Home = () => {

  return (

    <div className='home-background'>

      <h3>Home Page</h3>

    </div >

  )

}

export default Home;

Login.jsx

import "./Login.css"

import React from 'react';

const Login = () => {

return (

<div className="header">

<h2>Login Page</h2>

<form className="loginform">

<div className="useranme">

<label>Username </label>

<input type="text" placeholder="Enter a username" />

</div>

<br />

<div className="pass">

<label>Passwords </label>

<input type="Password" placeholder="Enter a Password" />

</div>

<br />

<div className="buttom">

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

</div>

</form>

</div>

);

}

export default Login

import "./Login.css"

import React from 'react';

const Login = () => {

  return (

    <div className="header">

      <h2>Login Page</h2>

      <form className="loginform">

        <div className="useranme">

          <label>Username </label>

          <input type="text" placeholder="Enter a username" />

        </div>

        <br />

        <div className="pass">

          <label>Passwords </label>

          <input type="Password" placeholder="Enter a Password" />

        </div>

        <br />

        <div className="buttom">

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

        </div>

      </form>

    </div>

  );

}

export default Login

Navbar.jsx

import React from 'react'

import "./Navbar.css"

const Navbar = () => {

return (

<div className='navbar'>

<div className='nav-content'>

<ul>

<a href='/'><li className='nav-item'>Home</li></a>

<a href='/about'><li className='nav-item'>Login</li></a>

<a href='/about'><li className='nav-item'>Register</li></a>

</ul>

</div>

</div>

)

}

export default Navbar

import React from 'react'

import "./Navbar.css"

const Navbar = () => {

  return (

    <div className='navbar'>

      <div className='nav-content'>

        <ul>

          <a href='/'><li className='nav-item'>Home</li></a>

          <a href='/about'><li className='nav-item'>Login</li></a>

          <a href='/about'><li className='nav-item'>Register</li></a>

        </ul>

      </div>

    </div>

  )

}

export default Navbar

Index.jsx

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

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

root.render(

<React.StrictMode>

<App />

</React.StrictMode>

);

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

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

root.render(

  <React.StrictMode>

    <App />

  </React.StrictMode>

);

**Output**

