ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO: 1

Aim: Perform the REPL in Node.js

Code:

To perform the REPL (Read-Eval-Print Loop) in Node.js

a) Open Command Prompt

b)Write command ‘node’:To launch the Node.js REPL , where you can write Javascript code and execute it interactively.

c) Use the REPL by typing JS expression or functions.

```
C:\Users\rikiy\OneDrive\Documents\WT Practicals> > console.log("Hello World")
Hello World
undefined
> function multiply(x,y,z){return x*y*z;}
undefined
> multiply(4,5,6)
120
> function powerVal(a,b) {
return a**b;
}
undefined
> powerval(3,4)
81
> powerval(2,3)
8
> name=function(nm){
return "Hello "+nm;
[Function: name]
> name("bob")
'Hello bob'
```

ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO : 2

Aim: Using modules, perform the Arithmetic Operations

Code:

```
var req = require('./abc');
var addRes = req.add(15, 9); console.log('Addition

Result: ' + addRes);var subRes = req.sub(15, 9);

console.log('Subtraction Result: ' + subRes);

var multiRes = req.multi(15, 9);

console.log('Multiplication Result: ' + multiRes);


var divRes = req.div(15, 9);

console.log('Division Result: ' + divRes);
```

Output:

```
PS C:\Users\Lenovo\Documents\100(An
Addition Result: 24
Subtraction Result: 6
Multiplication Result: 135
Division Result: 1.6666666666666667
```

ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO: 3

Aim: Using modules, find the Area of a Circle, Rectangle, Square

Code:

```
function areaOfSquare(side) {
   return side * side;
}
const sideLength = 5;
const squareArea = areaOfSquare(sideLength);
console.log(`The area of the square is: ${squareArea}`);
const pi = 3.141592653589793;
function findArea(r) {
   return pi * r * r;
}
let r = 5;
let circleArea = findArea(r);
console.log("Area of Circle is: " + circleArea);
function areaRectangle(a, b) {
   return a * b;
}
let a = 5;
let b = 6;
console.log("Area of Rectangle = " + areaRectangle(a, b));
```

Output:

```
PS C:\Users\Lenovo\Documents\100(Ani
The area of the square is: 25
Area of Circle is: 78.53981633974483
Area of Rectangle = 30
```

ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

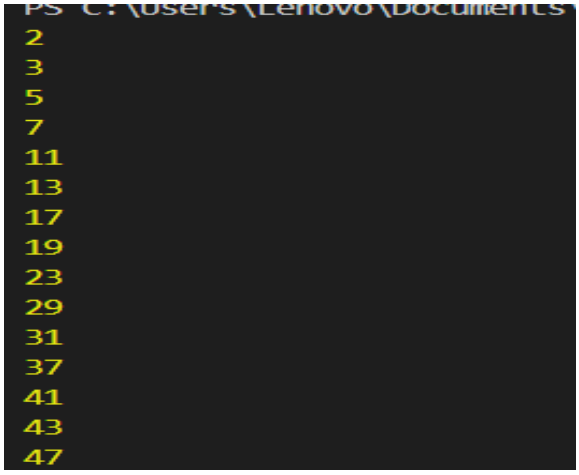
PRACTICAL NO : 4

Aim: Write a program to print the Prime Numbers from 1 to 50

Code :

```
for (let j = 2; j <= 50; j++) {
   let count = 0; // Reset count for each number
   for (let i = 1; i <= j; i++) {
      if (j % i == 0) count++;
   }
   if (count == 2) {
      console.log(j); // Prime numbers
   }
}
```

Output:



ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

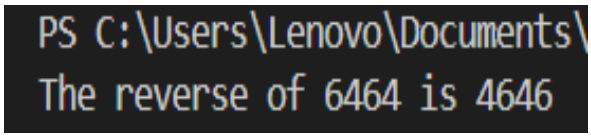
PRACTICAL NO : 5

Aim: Write a program to ind the reverse of a four-digit number

Code :

```
function reverseFourDigitNumber(num) {
    if (num < 1000 || num > 9999) {
        return "Please enter a four-digit number";
    }
    const reverseNum = num.toString().split('').reverse().join('');
    return parseInt(reverseNum, 10)
}
const number = 6464;
reversed = reverseFourDigitNumber(number);
console.log(`The reverse of ${number} is ${reversed}`);
```

Output:



ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

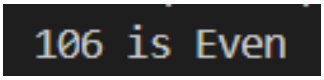
PRACTICAL NO : 6

Aim: Write a program to find if the number is odd or even

Code:

```
function checkEvenOdd(n) {
    if (n % 2 === 0) {
        console.log(n + " is Even");
    } else {
        console.log(n + " is Odd");

    }
}
const n = 106; checkEvenOdd(n);
```

Output:



ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO : 7

Aim: Write a program to check if the entered number is Armstrong or not

Code :

```
function isArmstrong(number) {
   let temp = number;
   let o = order(temp) let sum = 0;
   is greater than 0 while (temp) {
      remainder = temp % 10;
      temp = Math.floor(temp / 10);
      sum = sum + Math.pow(remainder, o);
   }
   if (sum === number) {
      console.log(number + " is an Armstrong Number");
   }
   else {
      console.log(number + " is Not an Armstrong Number");
   }
}
function order(number) {
   let n = 0;
   while (number > 0) {
      n++;
      number = Math.floor(number / 10);
   }
   return n;
}
isArmstrong(6); isArmstrong(520);
```

Output:

```
PS C:\Users\Lenovo\Documents\1
6 is an Armstrong Number
520 is Not an Armstrong Number
```

ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO : 8

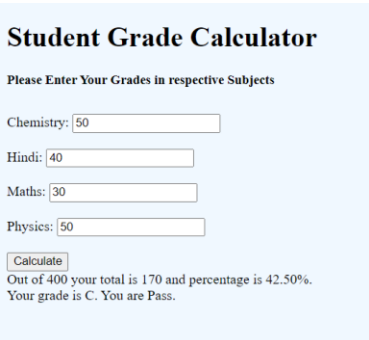
Aim: Write a program to take the marks of four subjects from user and check if the student has passed the examination or not, calculate percentage and grade.

Code :

```
<!DOCTYPE html>
<html lang="en">
<body style="background-color:aliceblue;" ></body>
<head>
<style>
  h2 { font-size: 15px; }
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Student Grade Calculator</title>
</style>
</head>
 <h1>Student Grade Calculator</h1>
 <h2>Please Enter Your Grades in respective Subjects</h2><br>

 <!-- Input fields for marks -->
 <label for="chemistry">Chemistry:</label>
 <input type="number" id="chemistry" placeholder="Enter Chemistry marks"><br><br>
<label for="hindi">Hindi:</label>
<input type="number" id="hindi" placeholder="Enter Hindi marks"><br><br>
<label for="maths">Maths:</label>
 <input type="number" id="maths" placeholder="Enter Maths marks"><br><br>
<label for="phy">Physics:</label>
 <input type="number" id="phy" placeholder="Enter Physics marks"><br><br>
<!-- Button to calculate grades -->
 <button onclick="calculate()">Calculate</button>
<!-- Area to display results -->
```

```
 <div id="showdata"></div>
 <script>

  const calculate = () => {

    const subjects = ["chemistry", "hindi", "maths", "phy"];

    const scores = subjects.map(subject => parseFloat(document.querySelector(`#${subject}`).value) ||
0);

    if (scores.some(score => score === 0)) { document.querySelector("#showdata").innerHTML =

     "Please enter all the fields";return;

    }
    const totalGrades = scores.reduce((sum, score) => sum + score, 0);const percentage = (totalGrades /
    400) * 100;

    const grades = percentage >= 80 ? "A" :

              percentage >= 60 ? "B" :

              percentage >= 40 ? "C" : "F";

    const result = percentage >= 39.5 ? "Pass" : "Fail";

    document.querySelector("#showdata").innerHTML  = `

     Out of 400 your total is ${totalGrades} and percentage is ${percentage.toFixed(2)}%.
<br>

     Your grade is ${grades}. You are ${result}.

  };

 </script>

</body>

</html>
```

Output:



ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO: 9

Aim: Write a program to print the Fibonacci series.

Code:

```
function printFibonacci(n) {
   let fibSeries = [];
   let a = 0, b = 1;
   for (let i = 0; i < n; i++) {
      fibSeries.push(a); // Add the current Fibonacci number to the series
      let next = a + b;  // Calculate the next Fibonacci number
      a = b;             // Update `a` to the current `b`
      b = next;          // Update `b` to the next Fibonacci number
   }
   console.log(`Fibonacci series up to ${n} terms:`, fibSeries.join(', '));
}
printFibonacci(10);
```

Output:

```
Fibonacci series up to 10 terms: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34
```

ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO: 10

Aim: Write a program to convert the temperature entered by the user

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Celsius to Fahrenheit Converter</title>
</head>
<body style="background-color:greenyellow;">
  <h2>Temperature Converter</h2>
  <p>Enter Celsius to convert to Fahrenheit:</p>
  <input id="inputCelsius" type="number" placeholder="Celsius"
  oninput="convertToFahrenheit()">
  <p>Fahrenheit: <span id="outputFahrenheit"></span></p>
  <script>
    function convertToFahrenheit() {
      const celsius = parseFloat(document.getElementById("inputCelsius").value) || 0;

      document.getElementById("outputFahrenheit").textContent = (celsius * 1.8 + 32).toFixed(2);

    }
  </scri
  >
</bod
  y>
</htm
  l>
```

Output:

ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO: 11

Aim: Write a program to demonstrate the factorial of a number using Anonymous Functions

Code:

```
const calculateFactorial = (n) => {
    let res = 1;
    for (let i = 2; i <= n; i++) {
        res *= i; // Multiply `res` by each number from 2 to `n`
    }
    console.log("Factorial of " + n + " is " + res);
    return res; // Return the factorial
};
calculateFactorial(8);
```

Output:

```
Factorial of 8 is 40320
```

ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

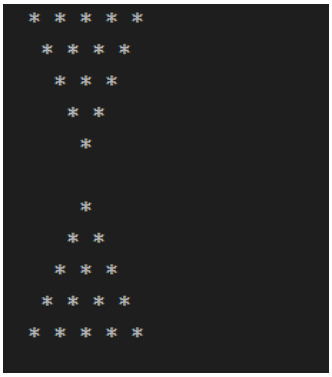
PRACTICAL NO: 12

Aim: Write a program to demonstrate the Pattern using Anonymous Functions.

Code:

```
const generatePatterns = (n) => {
   let string = "";
   for (let i = n; i > 0; i--) {
      for (let j = n; j > i; j--) {
         string += " "; // Add spaces before stars
      }
      for (let k = 0; k < i; k++) {
         string += "* "; // Add stars
      }
      string += "\n"; // Move to the next line
   }
   let string2 = "";
   for (let i = 1; i <= n; i++) {
      for (let j = n; j > i; j--) {
         string2 += " "; // Add spaces before stars
      }
      for (let k = 0; k < i; k++) {
         string2 += "* "; // Add stars
      }
      string2 += "\n"; // Move to the next line
   }
   console.log("Pattern 1: Inverted Triangle");
   console.log(string);
   console.log("Pattern 2: Regular Triangle");
   console.log(string2);
};
generatePatterns(5);
```

Output:



ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

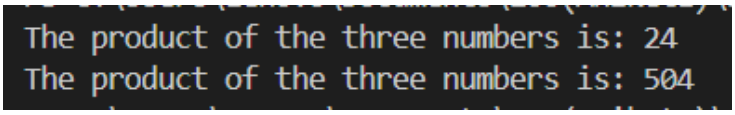
PRACTICAL NO: 13

Aim: Write a program to demonstrate the arithmetic operations using Callback Functions

Code:

```
function multiplyThreeNumbers(num1, num2, num3, callback) {
   const result = num1 * num2 * num3;
   callback(result); // Invoke the callback with the product
}
multiplyThreeNumbers(2, 3, 4, function(product) {
   console.log("The product of the three numbers is: " + product);
});
function displayResult(product) {
   console.log("The product of the three numbers is: " + product);
}
multiplyThreeNumbers(7, 8, 9, displayResult);
```

Output:



ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO: 14

Aim: Write a program to demonstrate the setTimeout function

Code:

```
function message() {
   console.log("Hello NodeJs Welcome!");
}
setTimeout(message, 5000);
setTimeout(() => {
   console.log("Calling from Arrow function");
}, 8000);
```

Output:

```
Hello NodeJs Welcome!
Calling from Arrow fucntion
```

ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO: 15

Aim: Write a program to place the order for a pizza using Events

Code:

```
const EventEmitter = require('node:events');

const emitter = new EventEmitter();

emitter.on('order-pizza', () => {
   console.log('Order received. Baking a pizza...');
});
emitter.emit('order-pizza');
```

Output:

```
PS C:\Users\Lenovo\Documents\1
Order received Baking a pizza
```

ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO: 16

Aim: Write a program to demonstrate Events by the same name

Code:

```
const events = require("events");
const eventEmitter = new events.EventEmitter();
function listener1() {
   console.log("Event received by Listener 1");
}
function listener2() {
   console.log("Event received by Listener 2");
}
eventEmitter.addListener("write", listener1);
eventEmitter.on("write", listener2);
eventEmitter.emit("write");
console.log(eventEmitter.listenerCount("write"));
eventEmitter.removeListener("write", listener1);
console.log("Listener 1 is removed");
eventEmitter.emit("write");
console.log(eventEmitter.listenerCount("write"));
console.log("Program Ended");
```

Output:

```
Event received by Listner 1
Event received by Listner2
0
Listner1 is removed
0
program Ended ....
```

ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO: 17

Aim: Write a program to calculate the salary using Events

Code:

```
const EventEmitter = require('events');

class SalaryCalculator extends EventEmitter {
calculateSalary(basic, ta) {
const hra = 0.2 * basic; // HRA is 20% of basic
const da = basic; // DA is 100% of Basic
const incomeTax = 0.3 * basic; //Income Tax is 30% of Basic
const professionalTax = 200; //Professional Tax is 200
 const salary = basic + hra + da + ta - incomeTax - professionalTax;
this.emit('calculateSalary', salary);

}
}

const salaryCalculator = new SalaryCalculator();
 salaryCalculator.on('calculateSalary', (salary) => {
  console.log(`The calculated salary is: ${salary}`);
});

salaryCalculator.calculateSalary(50000,8000); //Basic Salary is 50000 and TA is 8000
```

Output:

ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO: 18

Aim: Write a program to create an event to print the sum of odd and even numbers from an array.

Code:

```
const EventEmitter = require('events');
const sumEmitter = new EventEmitter();
function calculateSum(numbers) {
    let sumOdd = 0;
    let sumEven = 0;
    numbers.forEach(number => {
        if (number % 2 === 0) {
            sumEven += number;
        } else {
            sumOdd += number;
        }
    });
    sumEmitter.emit('sumCalculated', sumOdd, sumEven);
}

sumEmitter.on('sumCalculated', (sumOdd, sumEven) => {
    console.log(`Sum of Odd Numbers: ${sumOdd}`);
    console.log(`Sum of Even Numbers: ${sumEven}`);
});

const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
calculateSum(numbers); // This will trigger the 'sumCalculated' event
```

Output:

```
2 is even.
3 is odd.
5 is odd.
8 is even.
10 is even.
11 is odd.
13 is odd.
15 is odd.
```

ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO: 19

Aim: Write a program to demonstrate File handling in Node.js

Code:

```
const fs = require("fs");
fs.writeFile("_com.txt", 'Hello world', function (err) {
   if (err) {
      console.error("Error writing file:", err);
      return;
   }
   console.log("Writing File");
   fs.appendFile("_com.txt", "\nHello Everyone \nGive ThumbsUp", function (err) {
      if (err) {
         console.error("Error appending file:", err);
         return;
      }
      console.log("Appending File");
      fs.readFile("_com.txt", 'utf8', function (err, data) {
         if (err) {
            console.error("Error reading file:", err);
            return;
         }
         console.log("Reading File");
         console.log(data);
         fs.unlink("_com.txt", function (err) {
            if (err) {
               console.error("Error deleting file:", err);
               return;
            }
            console.log("Deleting File");
            console.log("File Deleted Successfully");
         });
      });
   });
});
```

Output:

```
Writing File
Appending File
Reading File
Hello world
Hello Everyone
Give ThumbsUp
Deleting File
File Deleted Successfully
```

ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO: 20

Aim: Write a Node.js code to display Employee Job Registration Form saved in an HTML ile in response to the client's access request to the server

Code:

Form.js:

```
const http = require("http");
const fs = require("fs");
http.createServer((req, res) => {
   fs.readFile('register.html', (err, data) => {
      if (err) {
         res.writeHead(500, { 'Content-Type': 'text/html' });
         res.end("Error reading the file");
         console.error("Error reading file:", err);
      } else {
         res.writeHead(200, { 'Content-Type': 'text/html' });
         res.end(data);
      }
   });
}).listen(8000, () => {
   console.log("Server is running at http://localhost:8000");
});
```

Register.html :

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Employee Registration Form</title>
   <style>
      body {
         background-color: #000000;
         color: white;
         font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;
      }
      h1 {
         text-align: center;
      }
      form {
         width: 300px;
         margin: 0 auto;
         padding: 20px;
```

```
        border: 2px solid white;
        border-radius: 10px;
      }
      input[type="text"],
      input[type="number"] {
        width: 100%;
        padding: 8px;
        margin: 5px 0;
        border-radius: 5px;
        border: 1px solid #ccc;
      }
    </style>
</head>
<body>
    <h1>Employee Registration Form</h1>
    <form>
      <label for="name">Employee Name:</label>
      <input type="text" id="name" name="name" placeholder="Enter Employee Name"
required/><br><br>

      <label for="id">Employee ID:</label>
      <input type="number" id="id" name="id" placeholder="Enter Employee ID"
required/><br><br>

      <label for="salary">Salary:</label>
      <input type="number" id="salary" name="salary" placeholder="Enter Salary"
required/><br><br>

      <label for="location">Location:</label>
      <input type="text" id="location" name="location" placeholder="Enter Location"
required/><br><br>

      <input type="submit" value="Submit"/>
    </form>
</body>
</html>
```

Output:

ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO: 21

Aim: Write a program to handle request url between various HTML pages.

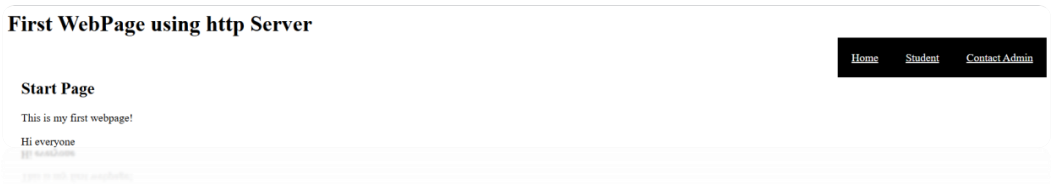
Code:

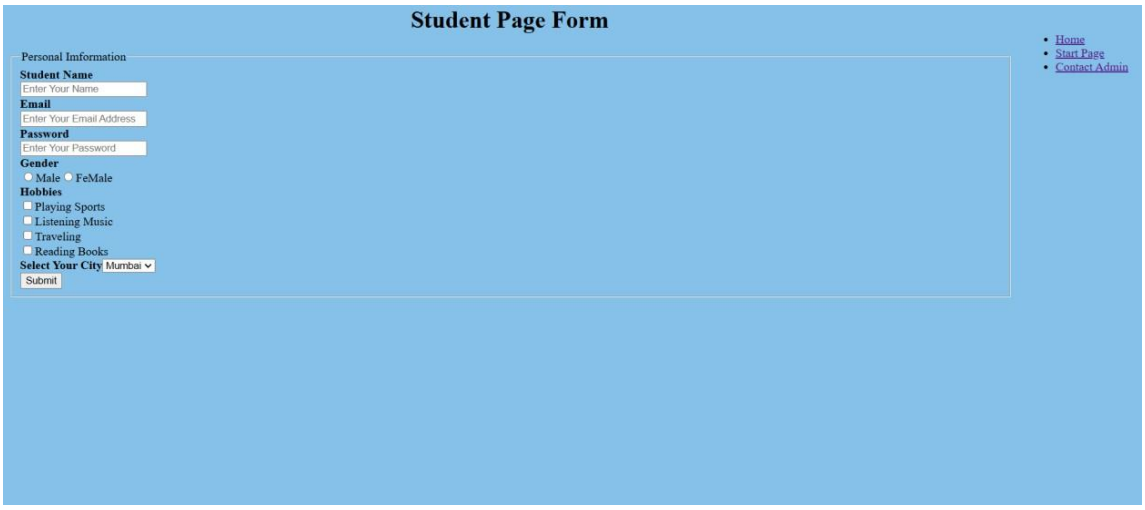
```
var http = require('http');

var server = http.createServer(function(req, res) {
   if (req.url == '/') {
      res.writeHead(200, {'content-type': 'text/html'});
      res.write('<html><head><style> ul li {display: inline-block; float: right; height: 40px;} ul li a
{padding: 20px; background: black; color: white;}</style></head><body>');
      res.write('<div><h1>First WebPage using http Server</h1></div>');
      res.write('<div><ul><li><a href="/admin">Contact Admin</a></li><li><a
href="/student">Student</a></li><li><a href="/home">Home</a></li></ul></div>');
      res.write('<div style="background: white; padding: 20px;"><h2>Start Page</h2><p>This is my
first webpage!</p><p>Hi everyone</p></div>');
      res.write('</body></html>');
      res.end();
   }
   else if (req.url == '/home') {
      res.writeHead(200, {'content-type': 'text/html'});
      res.write('<html><head><style>body {padding-left: 43px; padding-right: 43px; background-color:
#7dcea0;}</style></head><body>');
      res.write('<p><h1>This is home page</h1></p><h1>Aniket Lad</h1><h3>This page is a brief
insight to who I am.</h3>');
      res.write('<nav style="background-color: white; text-align: center;"><ul><li><a href="/">Start
Page</a></li><li><a href="/student">Student</a></li><li><a
href="/admin">Admin</a></li></ul></nav>');
      res.write('</body></html>');
      res.end();
   }
   else if (req.url == '/student') {
      res.writeHead(200, {'content-type': 'text/html'});
      res.write('<html><head><style>body {background-color: pink;} ul {display: inline-block; float:
right; height: 40px;} li {padding: 20px; background: black; color: white;}</style></head><body>');
      res.write('<h1 align="center">Student Page Form</h1>');
      res.write('<form action="url" method="post"><fieldset><legend>Personal
Information</legend>');
      res.write('<label><strong>Student Name</strong></label><br/><input type="text"
name="Student Name" placeholder="Enter Your Name" /><br/>');
      res.write('<label><strong>Email</strong></label><br/><input type="email" name="email"
placeholder="Enter Your Email Address" /><br/>');
      res.write('<label><strong>Password</strong></label><br/><input type="password"
name="Password" placeholder="Enter Your Password" /><br/>');
      res.write('<label><strong>Gender</strong></label><br/>');
      res.write('<input type="radio" name="Gender" value="Male" />Male');
```

```
        res.write('<input type="radio" name="Gender" value="Female" />Female<br/>');
        res.write('<label><strong>Hobbies</strong></label><br/>');
        res.write('<input type="checkbox" name="Hobbies" value="Playing Sports" />Playing
Sports<br/>');
        res.write('<input type="checkbox" name="Hobbies" value="Listening Music" />Listening
Music<br/>');
        res.write('<input type="checkbox" name="Hobbies" value="Traveling" />Traveling<br/>');
        res.write('<input type="checkbox" name="Hobbies" value="Reading Books" />Reading
Books<br/>');
        res.write('<label><strong>Select Your City</strong></label>');
        res.write('<select name="City"><option value="Mumbai">Mumbai</option><option
value="Gujarat">Gujarat</option><option value="Pune">Pune</option><option
value="Thane">Thane</option></select><br/>');
        res.write('<input type="submit" onclick="alert(\'Thanks!\')" name="submit"
value="Submit"/></form>');
        res.write('</body></html>');
        res.end();
    }
    else if (req.url == '/admin') {
        res.writeHead(200, {'content-type': 'text/html'});
        res.write('<html><head><style>ul li {display: inline-block; float: right; height: 40px;} ul li a
{padding: 20px; background: black; color: white;}</style></head><body>');
        res.write('<div><ul><li><a href="/admin">Contact Admin</a></li><li><a
href="/student">Student</a></li><li><a href="/home">Home</a></li></ul></div><br><br>');
        res.write('<legend><h1><u>Admin Login</u></h1></legend>');
        res.write('<form action="#" method="POST" autocomplete="off">');
        res.write('<div class="input_field"><h3>Username</h3></div><div class="input_field"><input
type="text" name="userid" placeholder="Username" required/></div>');
        res.write('<div class="input_field"><h3>Password</h3></div><div class="input_field"><input
type="Password" name="pword" placeholder="Password" required/></div>');
        res.write('<button onclick="alert(\'Success\')">LOGIN NOW</button></form>');
        res.write('</body></html>');
        res.end();
    }
    else {
        res.end('Invalid request');
    }
});

server.listen(9000, () => {
    console.log('Node.js web server at port 9000 is running');
});
```

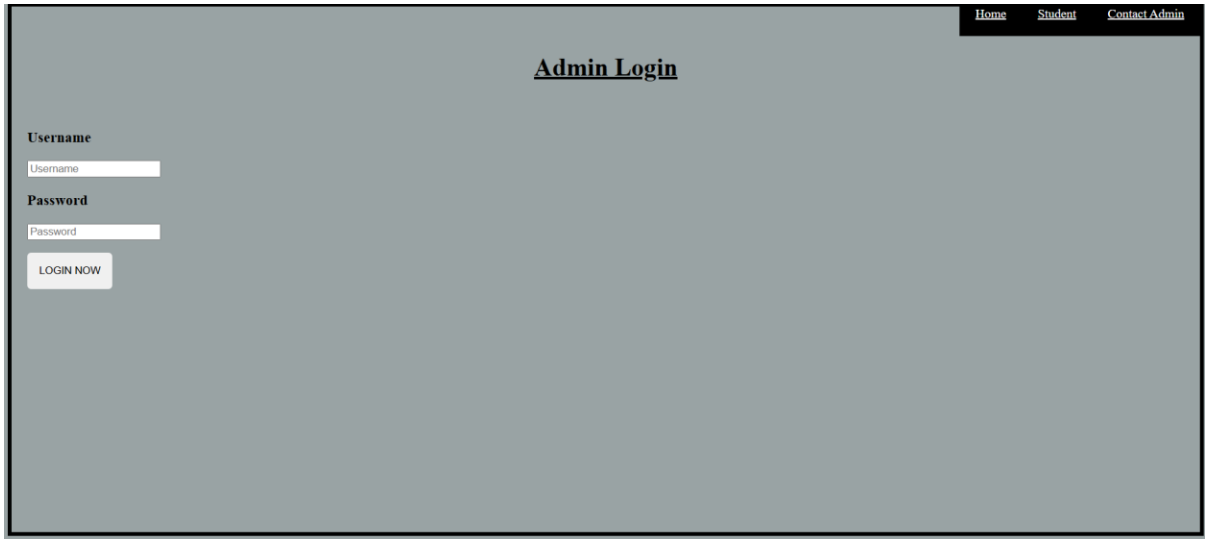
ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

Output:





ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B



PRACTICAL NO : 22

Aim: Write a program to implement the database in node.js using Xaamp.

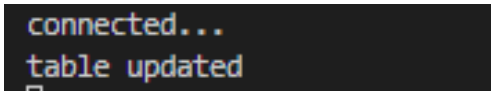
Code :

22.1) Create an Application to insert rows into Student Table in Node.js

CODE:

```
var mysql=require('mysql')
var con=mysql.createConnection({
   host:"localhost",
   user:"root",
   password:"",
   database:"Student_s4"
});
con.connect(function(err)
{
   if(err) throw err;
   console.log("connected...");
   var sql2 = "INSERT INTO student(id ,name , address,course , contact)
VALUES('2','Ramesh','Panvel','MCA','1234567891'),('3','Raju','Kalyan','MCA','12345678
92'),('4','Roshan','Dombivali','MCA','1234567893'),('5','Dinesh','Ambernath','MCA','1234
567894'),('6','Jitesh','Dombivali','MCA','1234567895')";
   con.query(sql2,function(err,result)
{
   if(err) throw err;
   console.log("table updated");
});
});
```

OUTPUT :



	id	name	address	course	contact	age
Edit Copy Delete	1	raj	thane	MCA	1234567890	NULL
Edit Copy Delete	2	Ramesh	Panvel	MCA	1234567891	NULL
Edit Copy Delete	3	Raju	Kalyan	MCA	1234567892	NULL
Edit Copy Delete	4	Roshan	Dombivali	MCA	1234567898	NULL
Edit Copy Delete	5	Dinesh	Ambernath	MCA	1234567894	NULL
Edit Copy Delete	6	Jitesh	Dombivali	MCA	1234567895	NULL

22.2) Write a Node.js application to retrieve and update the record related to the entries received for the conference participation. Update the mobile number of participant whose name is Roshan,

CODE:

```
var mysql=require('mysql')
var con=mysql.createConnection({
   host:"localhost",
   user:"root",
   password:"",
   database:"Student_s4"
});
con.connect(function(err) {
   if (err) throw err;
   console.log("Connected successfully to server");
   var sql = "SELECT * FROM student WHERE name = 'Roshan'";

   con.query(sql, function(err, result) {
      if (err) throw err;
      console.log("Participant found: ", result);
      var newMobileNumber = '1234567898';
      var updateSql = `UPDATE student SET contact = '${newMobileNumber}'
WHERE name = 'Roshan'`;
   con.query(updateSql, function(err, result) {
   if (err) throw err;
   console.log("Number of records updated: " + result.affectedRows);
   });
});
});
```

OUTPUT:

```
PS C:\Sakib T-131\WT PRACTICAL> node database4.js
Connected successfully to server
Participant found:  [
  RowDataPacket {
    id: 4,
    name: 'Roshan',
    address: 'Dombivali',
    course: 'MCA',
    contact: 1234567893
  }
]
Number of records updated: 1
```

22.3)Create an Application to add column to Student table in Node.js

CODE:

```
var mysql=require('mysql')
var con=mysql.createConnection({
   host:"localhost",
   user:"root",
   password:"",
   database:"Student_s4"
});
con.connect(function(err)
{
if(err) throw err;
console.log("connected...");
var sql = "ALTER TABLE student ADD age INT(5)";
con.query(sql,function(err,result)
{
   if(err) throw err;
   console.log("table altered");
});
});
```

OUTPUT:

```
PS C:\Sakib T-131\WT PRACTICAL> node database5.js
connected...
table altered
```

ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO : 23

Aim : Write a program to Display Hello World using ReactJS :

Code :

App.js

```
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Hello World ! My name is Sakib.
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
}

export default App;
```

Output :

ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO : 24

Aim : Create an application in ReactJS to implement component life cycle

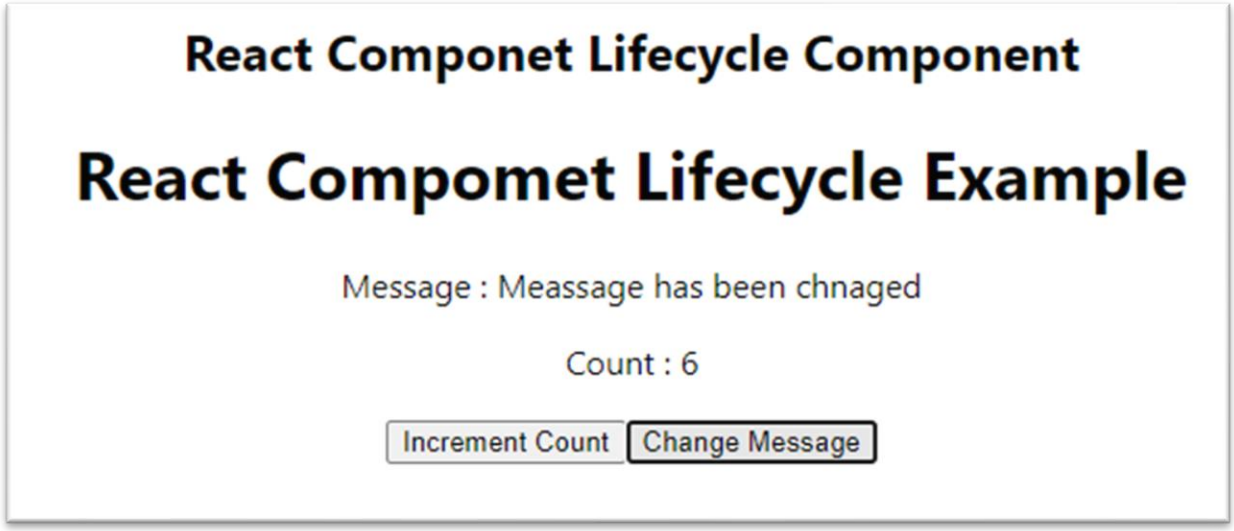
Code :

App.js

```
import logo from './logo.svg';
import './App.css';
import React, {useState, useEffect} from 'react';

const LifecycleCompenent= () => {

  const[count, setCount]= useState(0);
  const[message, setMessage]= useState('Hello World !');

  // Equivalent to componentDidMount ,componentDidUpdate , componentWillUnmount
  useEffect(() => {
    //This function will run once when the component mounts(initial render)

    console.log('Component mounted !');

    //This function will run once when the componentWillUnmount
    return () => {
      console.log('Component will unmount !');
    };
}, []); //Empty dependency array means this run only once on mount

useEffect(() => {
  //This function will run every time the count changes (update phase)
  console.log('Count updated to ${count}');
} ,[count]); //This run only when 'count' changes

const  handleClick= () => {
  setCount(count + 1); //Increment Count
};
const handleMessageChange= () => {
  setMessage('Message has been changed'); //Update Message
};

return (
  <div>
    <h1> React Component Lifecycle Example</h1>
    <p> Message  : {message} </p>
    <p> Count : {count}</p>
    <button onClick={handleClick}> Increment Count</button>
    <button onClick={handleMessageChange}> Change Message</button>
  </div>
```

```
);
};
const App=() => {
  return (
    <div className='App'>
      <h2>React Component Lifecycle Component</h2>
      <LifecycleCompenent/>
    </div>
  );
};
export default App;
```

Output :



ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO : 25

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

Code :

Class Component

MyClassComponet.js

```
import React, {Component} from 'react';
class MyClassComponent extends Component {
    constructor(props){
    super(props);
    this.state={
      message :'Hello , Welcome to React Class Component ',
      counter: 0,
    };
}

incrementCounter=()=> {
   this.setState((prevState)=> ({
      counter:prevState.counter+1,
   }));
};

render(){
return(
   <div style={{textAlign:'center' ,marginTop:'50px'}}>
   <h1>{this.state.message}</h1>
   <p>Counter :{this.state.counter}</p>
   <button onClick={this.incrementCounter} style={{padding:'10 px 20 px', fontSize:'16px'}}>
   IncrementCounter
   </button>
   </div>
);

}
}
export default MyClassComponent;
```

App.js

```
import logo from './logo.svg';
import './App.css';
import MyClassComponent from './MyClassComponent';

function App() {
  return (
```

```
    <div>
      <MyClassComponent/ >
    </div>
  );
}

export default App;
```

Output :

ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO : 26

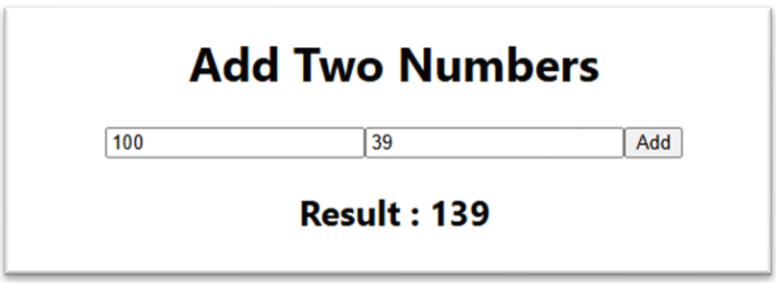
Aim: Create an application to implement functional component in ReactJS

CODE :

app.js

```
import './App.css';
import  React, {useState} from 'react';
const AddTwoNumbers=()=> {
  const [num1, setNum1] = useState('');
  const [num2, setNum2] = useState('');
  const [sum, setSum] = useState(null);
  const handleAddition= () =>{
    const result =parseFloat(num1) +parseFloat(num2);
    setSum(result);
  };
  return (
    <div style={{textAlign:'center',marginTop:'50px'}}>
     <h1> Add Two Numbers</h1>
     <div style ={{ marginBottom:'20px'}}>
      <input type="number" placeholder='Enter first number'
      value={num1}
      onChange={(e) => setNum1(e.target.value)}
      style={{marginRight:'10 px' ,padding:'5 px'}} />
       <input type="number" placeholder='Enter second number'
      value={num2}
      onChange={(e) => setNum2(e.target.value)}
      style={{marginRight:'10 px' ,padding:'5 px'}} />
      <button onClick={handleAddition} style={{padding: '5px 10 px'}}>Add</button>
      {sum!==null && <h2> Result : {sum}</h2>}
      </div>
      </div>
  );
};
export default AddTwoNumbers;
```

Output :



PRACTICAL NO : 27

Aim :Create an application in ReactJS import and export the files (components)

Code:

FileUploader.js

```
import React, { useState } from "react";
const FileUploader = () => {
  const [fileContent, setFileContent] = useState(""); // Declare useState correctly
  const handleFileUpload = (e) => {
    const file = e.target.files[0];
    const reader = new FileReader();
    reader.onload = (event) => {
      setFileContent(event.target.result); // Correctly set file content
    };
    if (file) reader.readAsText(file); // Read file as text
  };
  return (
    <div>
      <h3>Upload a File</h3>
      <input type="file" onChange={handleFileUpload} />
      {fileContent && ( // Conditionally render file content
        <div>
          <h4>File Content:</h4>
          <textarea value={fileContent} readOnly rows="10" cols="50" />
        </div>
      )}
    </div>
  );
};
export default FileUploader;
```

FiledDownlaoder.js

```
import React from "react";
const FileDownloader = () => {
  const handleDownload = () => {
    const content = "This is some sample text for the file.";
    const blob = new Blob([content], { type: "text/plain" });
    const url = URL.createObjectURL(blob);
    const link = document.createElement("a");
    link.href = url;
    link.download = "sample.txt";
    link.click();
    URL.revokeObjectURL(url);
```

```
  };

  return (
    <div>
      <h3>Download a File </h3>
      <button onClick={handleDownload}>Download</button>
    </div>
  );
};
export default FileDownloader;
```

App.js

```
import React from "react";
import FileDownloader from "./components/FileDownloader.js";
import FileUploader from "./components/FileUploader.js";
function App() {
  return (
    <div style={{ textAlign: "center", margin: "20px" }}>
      <h1>React File Import/Export</h1>
      <FileUploader />
      <FileDownloader />
    </div>
  );
}
export default App;
```

Output :

ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO : 28

Aim :Create an application to increment and decrement counter using state.

Code:

App.js

```
import './App.css';
import React, { useState } from 'react';
const Counter = () => {
const [count, setCount] = useState(0);
return (
<div>
<h1>Count: {count}</h1>
<button onClick={() => setCount(count + 1)}>Increment</button>
<button onClick={() => setCount(count - 1)}>Decrement</button>
</div>
);
};
export default Counter;
```

Output :

ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO : 29

Aim : Create an application to display your name using prop.

Code :

App.js

```
const Greeting = ({ name }) => {
  return <h1>Hello, {name}!</h1>;
  };
  const App = () => {
  return (
  <div>
  <Greeting name="Alice" />
  <Greeting name="Bob" />
  </div>
  );
  };
  export default App;
```

Output :

ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO: 30

Aim: Create an application to implement To-Do task.

Code:

App.js

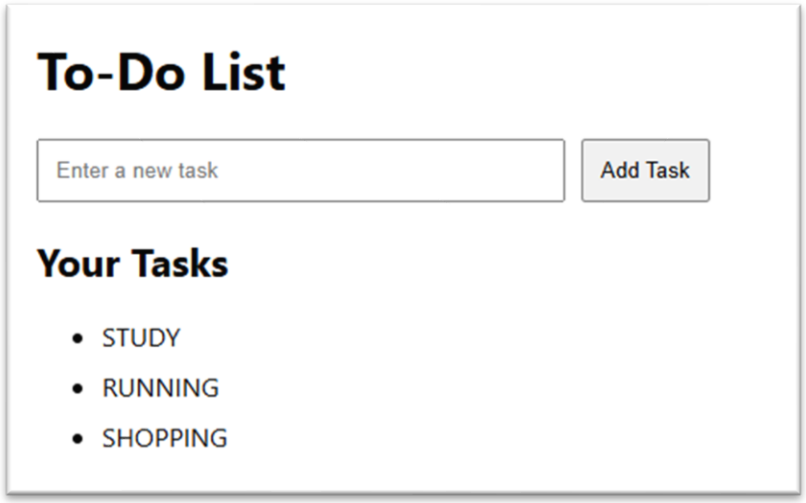
```
import React, { useState } from 'react';
import TaskList from './TaskList';
const App = () => {
const [tasks, setTasks] = useState([]); // State to manage tasks
const [taskInput, setTaskInput] = useState(''); // State for input field
const handleAddTask = () => {
if (taskInput.trim() !== '') {
setTasks([...tasks, taskInput]); // Add new task to the list
setTaskInput(''); // Clear input field
} };
return (
<div style={{ padding: '20px' }}>
<h1>To-Do List</h1>
<div>
<input
type="text"
value={taskInput}
onChange={(e) => setTaskInput(e.target.value)}
placeholder="Enter a new task"
style={{ padding: '10px', width: '300px', marginRight: '10px' }}
/>
<button onClick={handleAddTask} style={{ padding: '10px' }}>
Add Task
</button>
</div>
<TaskList tasks={tasks} />
</div>
);
};
export default App;
```

Tasklist.js - Child Component

```
import React from 'react';
const TaskList = ({ tasks }) => {
return (
<div style={{ marginTop: '20px' }}>
<h2>Your Tasks</h2>
{tasks.length === 0 ? (
```

```
<p>No tasks added yet.</p>
) : (
<ul>
{tasks.map((task, index) => (
<li key={index} style={{ marginBottom: '10px' }}>
{task}
</li>
))}
</ul>
)}
</div>
);
};
export default TaskList;
```

Output:



ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO: 31

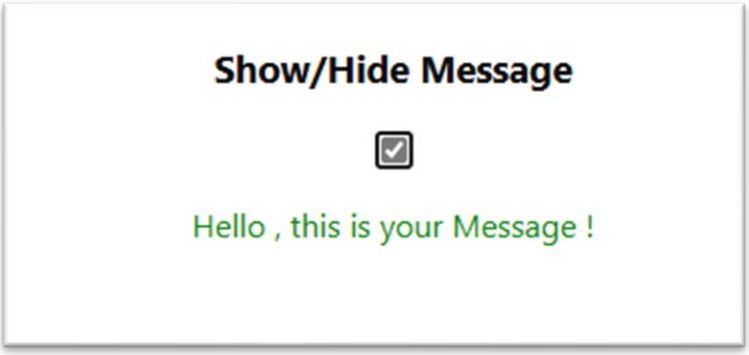
Aim: Create an application in ReactJS to use DOM events-onChange.

Code:

App.js

```
import React , {useState} from "react";
function ToggleMessage() {
  const[isChecked , setIsChecked]= useState(false); // State to track checkbox toggle

  const handleCheckboxChange= (event) => {
    setIsChecked(event.target.checked); // Update state when checkbox is toggled
  };
  return (
    <div style={{margin:"20 px", textAlign:"center"}}>
     <h3> Show/Hide Message</h3>
     <label>
      <input  type="checkbox" onChange={handleCheckboxChange} // Event handler for
checkbox
      style={{marginRight:"10 px"}}
     />
     </label>
     <div style={{marginTop:"20 px"}}>
      {isChecked && <p style={{color:"green"}}> Hello , this is your Message !</p>}
     </div>
    </div>
  )
}
export default ToggleMessage;
```

Output



ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO: 32

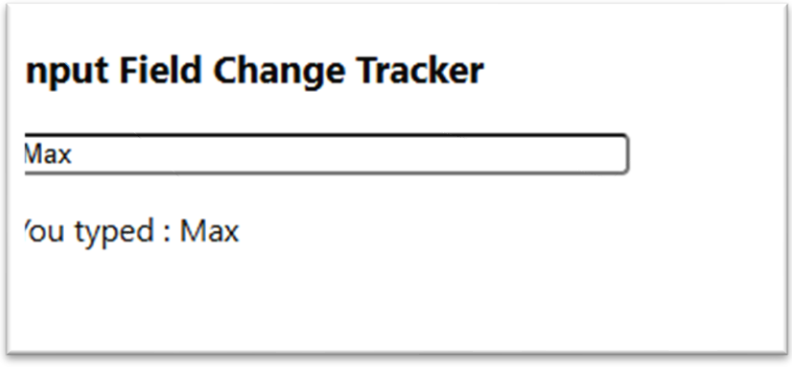
Aim: Write a program that tracks the changes in an input field and displays the entered text in real-time using onChange DOM event.

Code:

App.js

```
import React, {useState} from "react";
function InputTracker() {
   const[text, setText]=useState(""); // State to store the input value
   const handleChange=(event) => {
      setText(event.target.value); // Update the state with input value
   };
   return (
<div style={{margin:"20 px"}}>
<h3>Input Field Change Tracker</h3>
<input type="text" placeholder="Type something here.." value={text}
onChange={handleChange} // Event handler for onChange
style={{ padding :"8 px", border:"1 px solid #ccc" , borderRadius:"4px" ,width:"300px",}} />
<p style={{marginTop:"10 px"}} >You typed : {text} </p>
</div>
);
}
export default InputTracker;
```

Output :



ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO: 33

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

Code:

App.js

```
import React, {useState} from "react";
function KeyCodeDisplay() {
  const [keyCode, setKeyCode] = useState("");

  const handleKeyUp=(e) => {
    setKeyCode(`Key Code : ${e.keyCode}`);
  };
  return (
    <div>
    <input type="text" onKeyUp={handleKeyUp} placeholder="Press a key.." />
    <p>{keyCode}</p>

    </div>
  )
}
export default KeyCodeDisplay;
```

Output :

ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO: 34

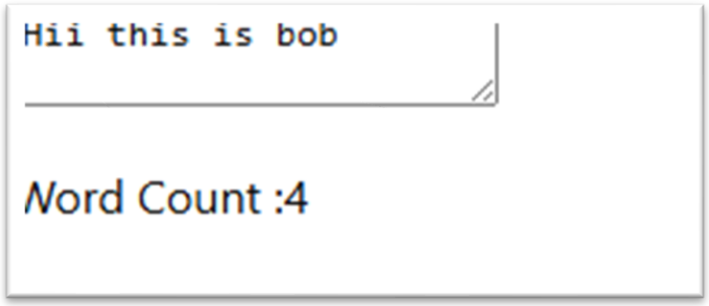
Aim: Write a Program to Counts words as they are typed using onKeyUp event.

Code:

App.js

```
import React, {useState} from "react";
function WordCount() {
  const [wordCount, setWordCount] = useState(0);
  const handleKeyUp=(e) => {
    const words=e.target.value.trim().split(/\s+/);
    setWordCount(words[0]===""?0: words.length);
  };
  return (
    <div>
    <textarea onKeyUp={handleKeyUp}
    placeholder="Enter text here" />
    <p> Word Count :{wordCount}</p>
    </div>
  )
}
export default WordCount;
```

Output :



ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO: 35

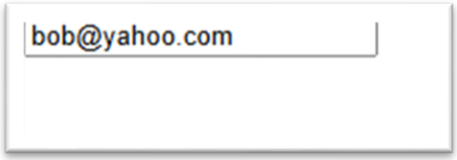
Aim: Write a Program to implement validation logic for an email field using onBlur event.

Code:

App.js

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

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

  const handleBlur=(e) => {

    const email=e.target.value;
    if(!email.includes("@")) {
      setError("Invalid email");
    } else {
      setError("");

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

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

export default ValidateOnBlur;
```

Output:



ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO: 36

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

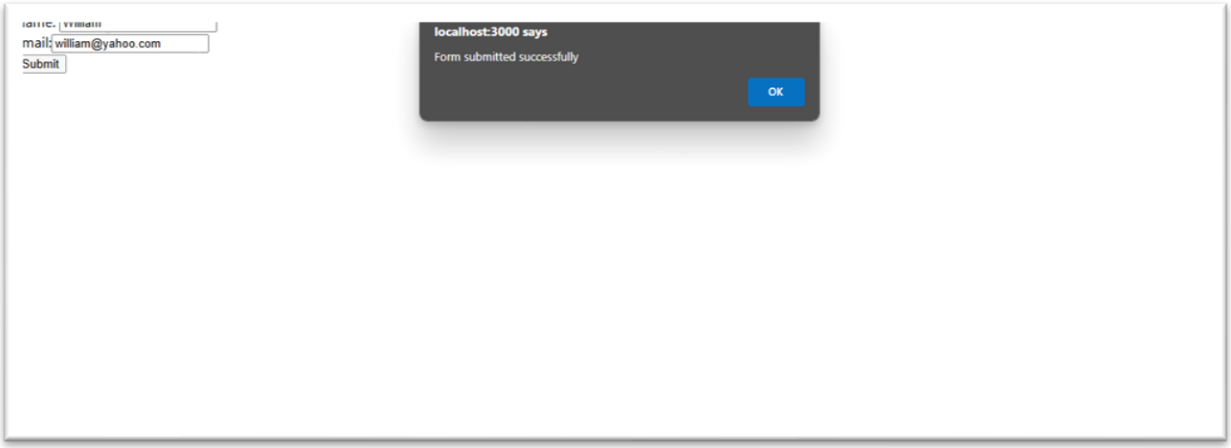
Code:

App.js

```
import React ,{useState} from "react";
function BasicFormValidation () {
 const[formData, setFormData]=useState({name:"", email:""});
 const[errors,setErrors]=useState({});

 const handleChange=(e) => {
  const {name, value} =e.target;
  setFormData({...formData,[name]:value});
 };

 const validate=() => {
  const newErrors={} ;
  if(!formData.name)newErrors.name="Name is required";
  if(!formData.email)newErrors.email="Email is required";
  else if(!/\S+@\S+\.\S+/.test(formData.email))
   newErrors.email="Email is invalid";

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

 };

 const handleSubmit=(e) => {
  e.preventDefault();
  if(validate()) {
   alert("Form submitted successfully");
  }
 };

 return(
  <form onSubmit={handleSubmit}>
   <div>
    <label> Name: </label>
    <input type="text"
    name="name"
    value={formData.name}
    onChange={handleChange}
    />
    {errors.name && <p style={{color:"red"}}>{errors.name}</p> }
```

```
    </div>

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

      {errors.email && <p style={{color:"red"}}>{errors.email} </p> }
    </div>
    <button type="submit" >Submit</button>
  </form>
 );
}

export default BasicFormValidation;
```

Output :



ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO: 37

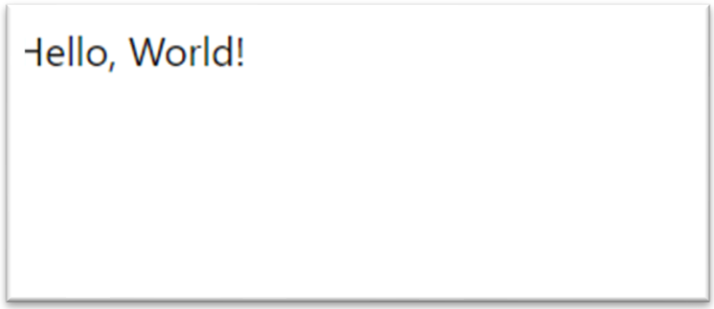
Aim: Write a Program to implement useEffect hook.

Code:

App.js

```
import React, { useEffect } from 'react';
function SimpleComponent() {
useEffect(() => {
console.log('Component mounted!');
}, []); // Empty dependency array ensures this runs only once on mount
return <div>Hello, World!</div>;
}
export default SimpleComponent;
```

Output :



ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

PRACTICAL NO: 38

Aim: Create SPA using react router.

Code:

Home.js

```
import React from 'react';
const Home = () => {
  return (
    <div>
      <h1>Home Page</h1>
      <p>Welcome to the Home Page!</p>
    </div>
  );
};
export default Home;
```

About.js

```
import React from 'react';
const About = () => {
  return (
    <div>
      <h1>About Page</h1>
      <p>This is the About Page.</p>
    </div>
  );
};
export default About;
```

Contact.js

```
import React from 'react';
const Contact = () => {
  return (
    <div>
      <h1>Contact Page</h1>
      <p>This is the Contact Page.</p>
    </div>
  );
};
export default Contact;
```

ANSARI AFTAB SALEEM
ROLL NO: 70
FYMCA B

App.js

```
import React from 'react';
import { BrowserRouter as Router, Route, Switch, Link } from 'react-router-dom';
import Home from './components/Home';
import About from './components/About';
import Contact from './components/Contact';
const App = () => {
  return (
    <Router>
      <div>
        {/* Navigation links */}
        <nav>
          <ul>
            <li><Link to="/">Home</Link></li>
            <li><Link to="/about">About</Link></li>
            <li><Link to="/contact">Contact</Link></li>
          </ul>
        </nav>
        {/* Define Routes */}
        <Routes>
          <Route path="/" exact component={Home} />
          <Route path="/about" component={About} />
          <Route path="/contact" component={Contact} />
        </ Routes >
      </div>
    </Router>
  );
};
export default App;
```

Output: