**Node.js Practical**

|  |  |  |
| --- | --- | --- |
| Sr. No. | Practical Name | Date |
| 1 | Steps to download node.js | 28/08/2023 |
| 2 | Steps to download visual studio | 28/08/2023 |
| 3 | Demonstrate the basic arithmetic operations in Node.js | 28/08/2023 |
| 4 | To determine whether a given number is even or odd in Node.js | 28/08/2023 |
| 5 | To print all prime numbers up to a given number in Node.js | 04/09/2023 |
| 6 | Create an application in Node.JS to reverse the given number and  display it | 04/09/2023 |
| 7 | Create an application in Node.js to display Armstrong number 15 | 04/09/2023 |
| 8 | To generate the first 10 numbers in the Fibonacci sequence in Node.js | 04/09/2023 |
| 9 | To demonstrate the use of setTimeout and arrow functions in Node.js | 11/09/2023 |
| 10 | To demonstrate module exports in Node.js | 11/09/2023 |
| 11 | write an application to find area of circle, square, rectangle using module in Node.js | 11/09/2023 |
| 12 | Write an application to demonstrate events module in Node.js | 11/09/2023 |
| 13 | write an application to demonstrate function (removeListner, listnerCount) in Node.js | 18/09/2023 |
| 14 | create an application in node.js to Return Event Emitter | 18/09/2023 |
| 15 | create an application in node.js to create Extend Event Emitter in Node.js | 18/09/2023 |
| 16 | Write an event emitter code to design an event called as “calculate Salary” which is used to calculate the salary of an employee by passing some arguments like Basic Salary, HRA (20% of Basic), DA(100% of Basic), TA, and deductions like Income Tax (30% of  Basic) and Professional Tax of 200 | 18/09/2023 |
| 17 | create an application in node.js to display message after 5 second &10 second | 09/10/2023 |
| 18 | create an application in node.js to demonstrate set interval function | 09/10/2023 |
| 19 | create an application in node.js to display factorial of a number | 09/10/2023 |
| 20 | Write as application to create http Server and Display message in Node.js | 09/10/2023 |
| 21 | Write a Node.js code to display Employee Job Registration Form saved in an HTML file in response to the client’s access request to the server. | 16/10/2023 |
| 22 | Write as application to create Home page, Admin page and Student page using http server in Node.js. | 16/10/2023 |
| 23 | Write in application to display details of the current file path in Node.js. | 16/10/2023 |
| 24 | Write an application to read file in Node.js | 16/10/2023 |
| 25 | Write an application to write in file in Node.js. | 23/10/2023 |
| 26 | Write an application to add data in file in Node.js. | 23/10/2023 |

|  |  |  |
| --- | --- | --- |
| 27 | Write an application to delete a file in Node.js | 23/10/2023 |
| 28 | Combine Read, Write, Append, Delete file in one program in Node.js | 23/10/2023 |
| 29 | Write and application to rename a file in Node.js | 20/11/2023 |
| 30 | Create an Application to create Database in Node.js | 20/11/2023 |
| 31 | Create an Application to create Student table with columns as id, name, address, course, contact in Node.js | 20/11/2023 |
| 32 | Create an Application to insert rows into Student table in Node.js | 20/11/2023 |
| 33 | Create an Application to display rows into Student table in Node.js | 04/12/2023 |
| 34 | Create an Application to Update rows in Student table in Node.js | 04/12/2023 |
| 35 | 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 “Sharma | 11/12/2023 |
| 36 | Create an Application to add column to Student table in Node.js | 11/12/2023 |
| 37 | Create an Application to delete records in Student table in Node.js | 11/12/2023 |

Angular Practical

|  |  |  |
| --- | --- | --- |
| Sr. No. | Practical Name | Date |
| 1 | Create an application in angular.js to demonstrate arithmetic operations and list. | 06/09/2023 |
| 2 | Create an application in angular.js to calculate registration fees if the number of people and registration amount is given by the user | 06/09/2023 |
| 3 | Create an application in angular.js to calculate simple interest take appropriate input from the user | 12/09/2023 |
| 4 | Write an application in angular.js to create an array of names and display all the names which has letter “i” using controller | 26/09/2023 |
| 5 | Create an application in angular.js to demonstrate the use of filters | 26/09/2023 |
| 6 | Create an application in angular.js to change the background color as the user changes input in the textbox | 03/10/2023 |
| 7 | Create an application in angular.js to demonstrate to display text in alert box | 17/10/2023 |
| 8 | Create an application in angular.js to demonstrate the use of ng-if, ng-disabled and ng-readonly | 17/10/2023 |
| 9 | Create an application in angular.js to demonstrate use of mouse enter and mouse-leave even | 31/10/2023 |
| 10 | Write an application in angular js to display options using select tag as user chooses the color option the respective color and content should change | 31/10/2023 |
| 11 | Write an Angular JS code to display a Registration form for Student applying for a new Course. Display all the values entered by the students. | 21/11/2023 |
| 12 | To demonstrate the use of regular expressions for validating input fields in a form | 21/11/2023 |
| 13 | To demonstrate use of validation directives. | 05/12/2023 |
| 14 | To demonstrate the state properties of form fields | 05/12/2023 |
| 15 | To demonstrate the use of a Single Page Application (SPA) | 12/12/2023 |
| 16 | Create an application with Login page and Registration Page using Single Page Application(SPA) | 12/12/2023 |

**PRACTICAL NO: 01**

**Aim:-Steps to download node.js**

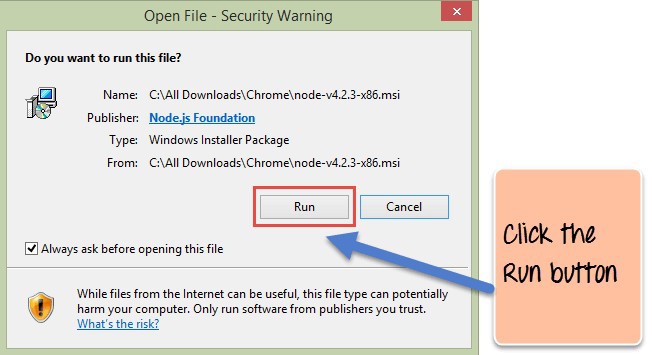
Step 1) Download Node.js Installer for Windows

Go to the site https://nodejs.org/en/download/ and download the necessary binary files.

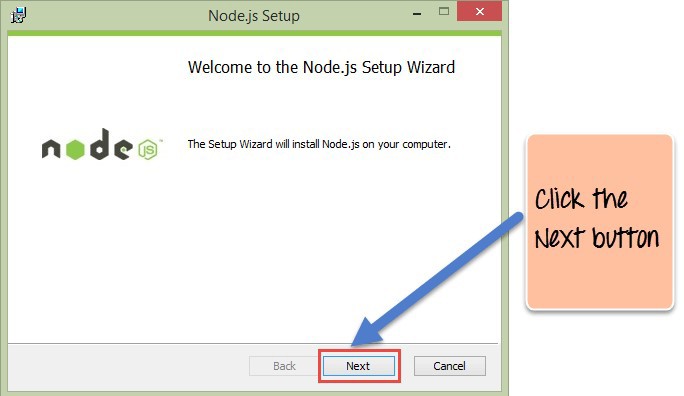


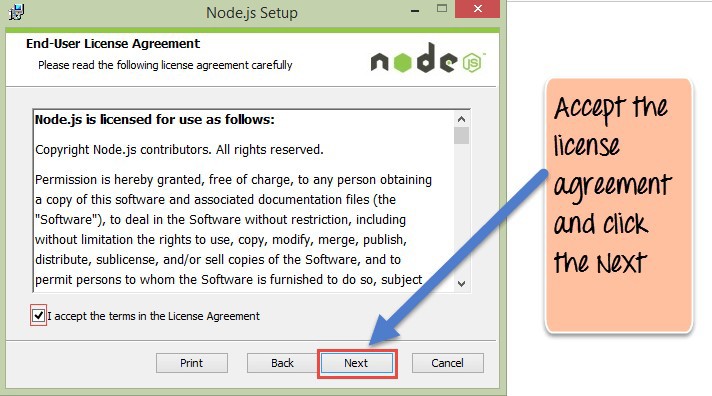
**Step 2)** Run the installation Double click on the downloaded .msi file to start the installation.

Click the Run button on the first screen to begin the installation.

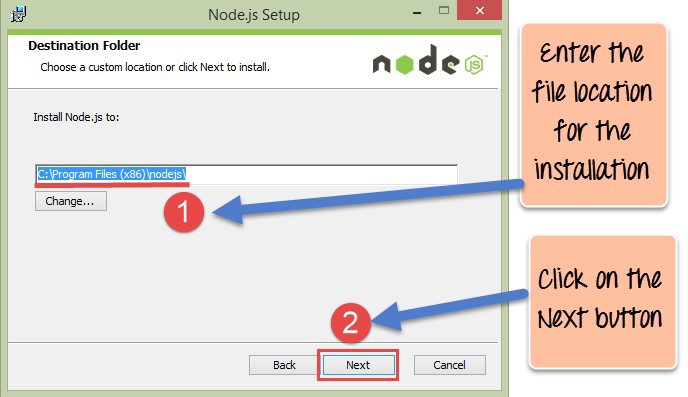


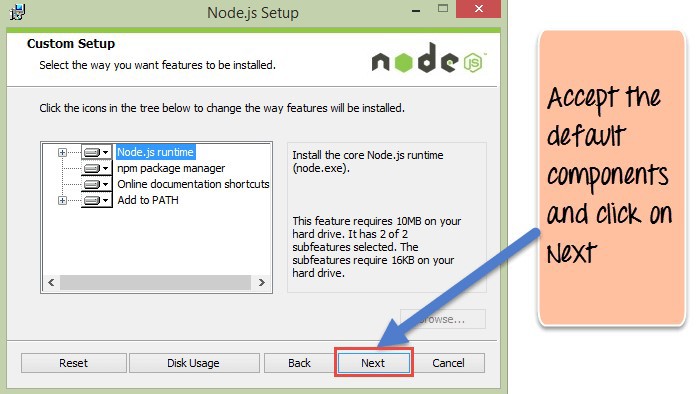
**Step 3)** Continue with the installation steps In the next screen, click the “Next” button to continue with the Node.js download and installation



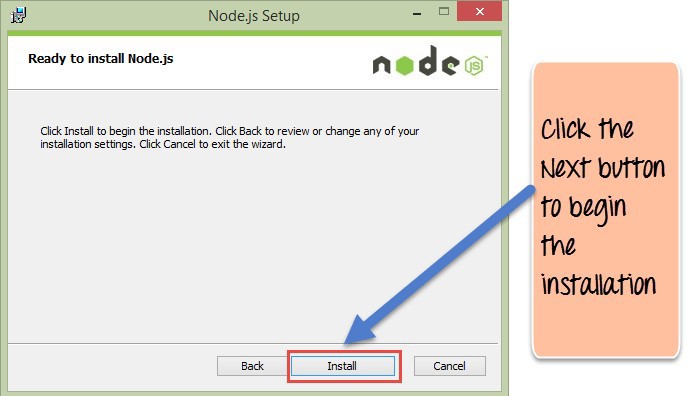
**Step 4)** Accept the terms and conditions In the next screen, Accept the license agreement and click on the Next button

**Step 5)** Set up the path In the next screen, choose the location where Node.js needs to be installed and then click on the Next button.



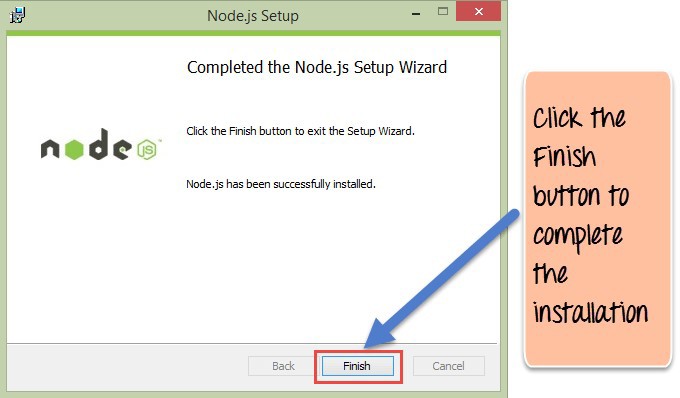
**Step 6**) Select the default components to be installed Accept the default components and click on the Next button.

**Step 7)** Start the installation In the next screen, click the Node.js install button to start installing on Windows



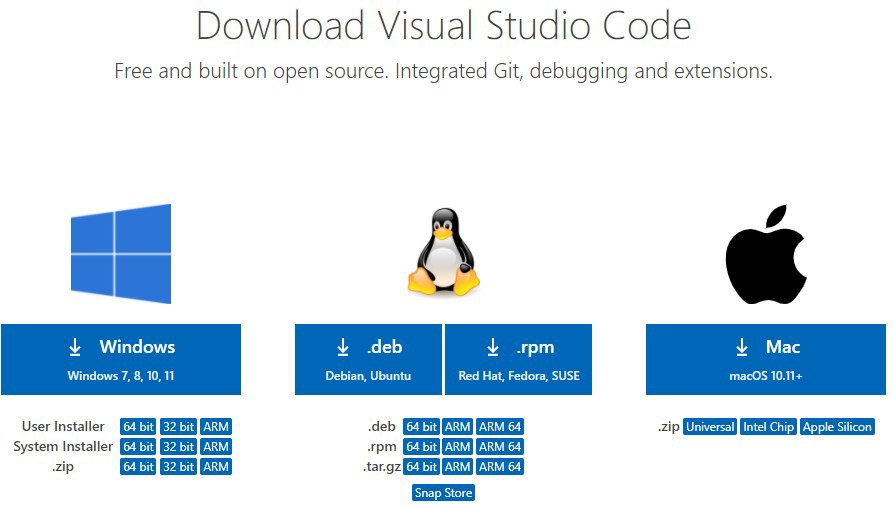
**Step 8**) Complete the installation Click the Finish button to complete the installation.

Complete the installation Click the Finish button to complete the installation.

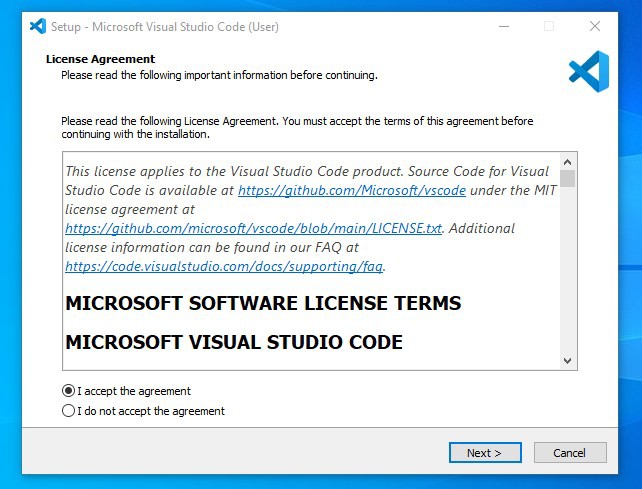


**PRACTICAL NO: 02**

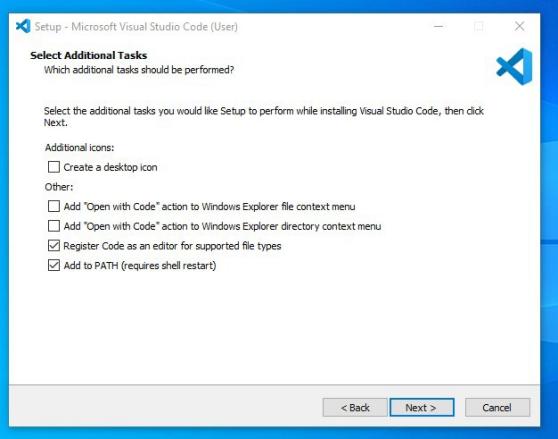
**Aim:-Steps to download visual studio**

Step 1: **Visit the official website of the** Visual Studio Code **using any web browser like Google Chrome, Microsoft Edge, etc. and Press the “**Download for Windows**” button**

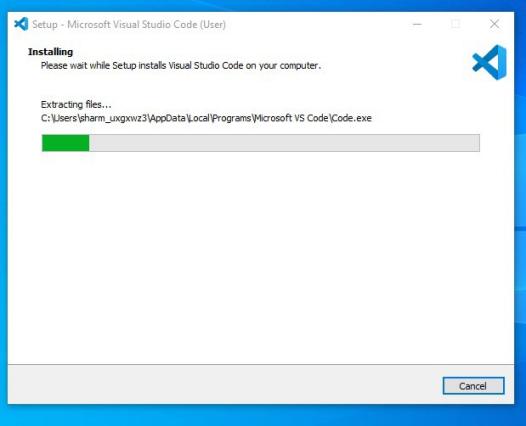
Step 2:When the download finishes, then the Visual Studio Code icon appears in the downloads folder. Click on the installer icon to start the installation process of the Visual Studio Code. After the Installer opens, it will ask you for accepting the terms and conditions of the Visual Studio Code. Click on and then clickthe button.



Step 3: **Choose the location data for running the Visual Studio Code. It will then ask you for browsing the location. Then click on** Next **button.**



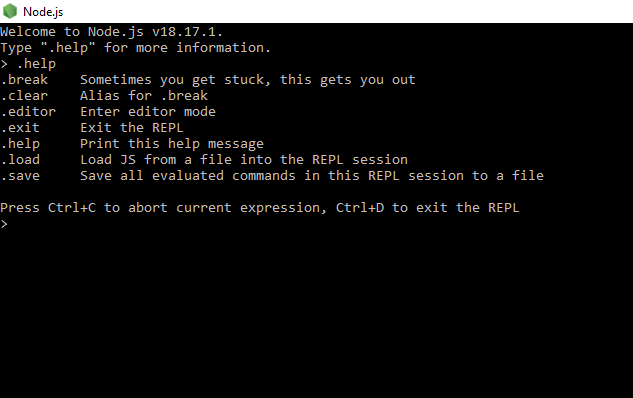
Step 4: **Then it will ask for beginning the installing setup. Click on the** Install **button. After clicking on Install, it will take about 1 minute to install the Visual Studio Code on your device.**

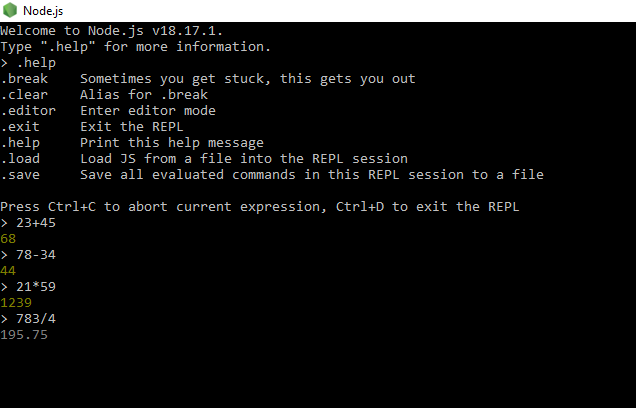


Step 5: **After the Installation setup for Visual Studio Code is finished, it will show a window like this below. Tick the “**Launch Visual Studio Code**” checkbox and then click** Next**.**

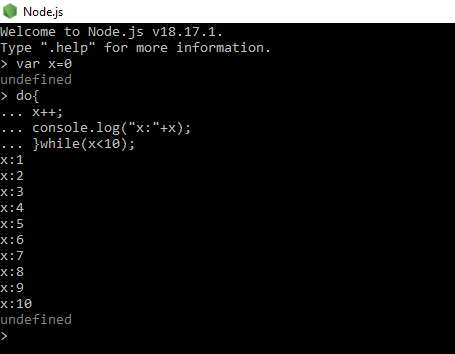


Step2: -

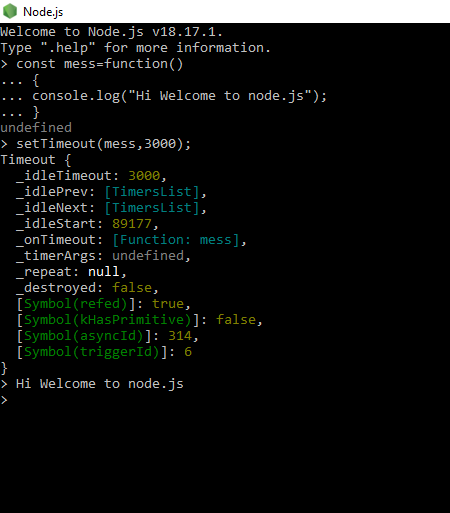




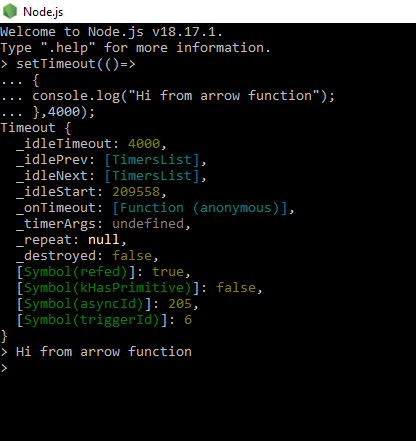
* Console program using do while program (Multiline Expression):



* Call back function:



* Arrow Call back function:



**PRACTICAL NO: 03**

**Aim:- Write an application to perform arithmetic operation.**

**Code:-**

//Addition

function sum(a,d)

{

return a+d;

}

s=sum(10,5);

console.log(s);

//Subtraction

function sub(b,f)

{

return b-f;

}

su=sub(65,54);

console.log(su);

//Multiplication

function mul(c,e)

{

return c\*e;

}

m=mul(15,16);

console.log(m);

//Division

function div(g,h)

{

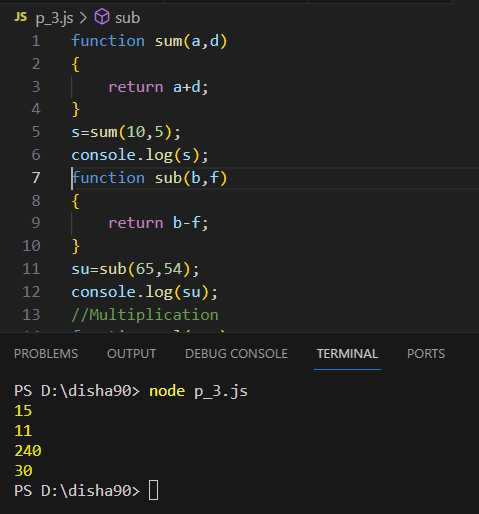
return g/h;

}

d=div(150,5);

console.log(d);

**Output:-**



**PRACTICAL NO: 04**

**Aim:- To determine whether a given number is even or odd in Node.js**

**Code:-**

function displayresult(a) {

console.log(a);

}

function check(num) {

let sum = num;

if (num % 2 == 0) {

console.log("Number is Even")

} else {

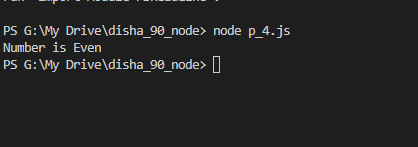
console.log("Number is odd")

}

}

check(18, displayresult)

**Output:-**



**PRACTICAL NO: 05**

**Aim:-To print all prime numbers up to a given number in Node.js**

**Code:-**

function isPrime(n)

{if(n==1||n==0) return false;

for(var i=2;i<n;i++){

if(n%i==0) return false;

} return true;

}

var num =30;

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

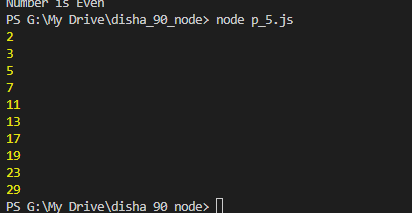
if(isPrime(i)){

console.log(i);

}

}

**Output:-**



**PRACTICAL NO: 06**

**Aim:-Create an application in NodeJS to reverse the given number and display it (Note: 5 digit number)**

**Code:-**

var number=456789;

var reversedNumber = number.toString().split('').reverse().join('');

console.log('Reversed number is: ' + reversedNumber);

**Output:-**



**PRACTICAL NO: 07**

**Aim:- Create an application in Node.js to display Armstrong number 15**

**Code:-**

function isArmstrongNumber(num) {

let sum = 0;

const strNum = String(num);

const len = strNum.length;

for (let i = 0; i < len; i++) {

sum += Math.pow(Number(strNum[i]), len);

}

return sum === num;

}

function printFirstNArmstrongNumbers(n) {

let count = 0;

let num = 1;

while (count < n) {

if (isArmstrongNumber(num)) {

console.log(num);

count++;

}

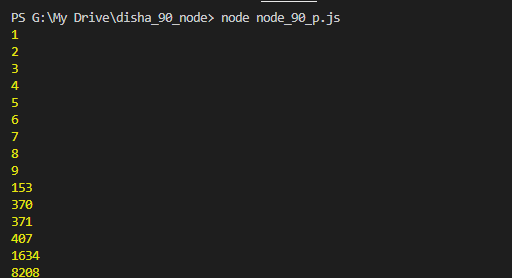
num++;

}

}

printFirstNArmstrongNumbers(15);

**Output:-**



**PRACTICAL NO: 08**

**Aim:-To generate the first 10 numbers in the Fibonacci sequence in Node.js**

**Code:-**

var a=0;

var b=1;

var c;

console.log(a);

console.log(b);

for(i=0;i<8;i++)

{

c=a+b;

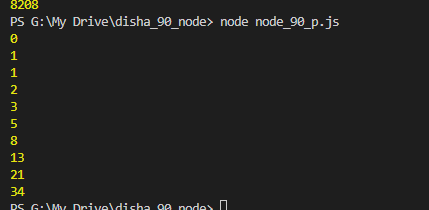
console.log(c);

a=b;

b=c;

}

**Output:-**

****

**PRACTICAL NO: 09**

**Aim:-To demonstrate the use of setTimeout and arrow functions in Node.js**

**Code :-**

const message = function(){

console.log("Hi I am Node Js, Welcome");

}

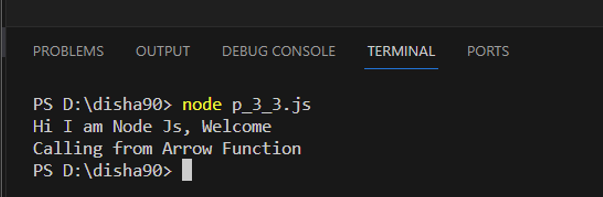
setTimeout(message, 5000);

setTimeout(() => {

console.log("Calling from Arrow Function");

},8000);

**Output:-**



**PRACTICAL NO: 10**

**Aim:-To demonstrate module exports in Node.js**

**Code:-**

**Step 1 :-** Create a file “P6first.js” and write following code

//Addition

function add(a,b){

return a+b;

}

exports.add=add;

//Subtraction

function sub(a,b){

return a-b;

}

exports.sub=sub;

//Multiplication

function mul(a,b){

return a\*b;

}

exports.mul=mul;

//Division

function div(a,b){

return a/b;

}

exports.div=div;

Step 2:- Create a file “p6firstmode.js” and write following code.

var req = require('./p6fist');

var res =req.add(26,3);

console.log("Addition:",res);

var res =req.sub(48,9);

console.log("Subtraction:",res);

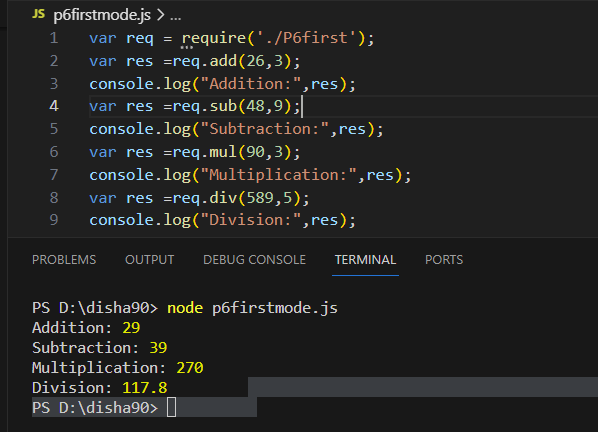
var res =req.mul(90,3);

console.log("Multiplication:",res);

var res =req.div(589,5);

console.log("Division:",res);

**Output:-**



**PRACTICAL NO: 11**

**Aim:-write an application to find area of circle, square, rectangle using module in Node.js**

**Code:-**

**Step 1:-** Create a file “p7area.js” and write following code.

//SQUARE

function square(s){

return s\*s;

}

//RECTANGLE

function rectangle(l,b){

return l\*b;

}

//CIRCLE

function circle(r){

return 3.14\*r\*r;

}

//EXPORT

exports.square=square; //SQUARE

exports.rectangle=rectangle; //RECTANGLE

exports.circle=circle; //CIRCLE

**Step 2:-** Create a file “p7result.js” and write following code.

//IMPORT FILE

var req = require('./p7area.js');

var sRes, rRes, cRes;

//IMPORT MODULE

sRes=req.square(8);

rRes=req.rectangle(6,8);

cRes=req.circle(7);

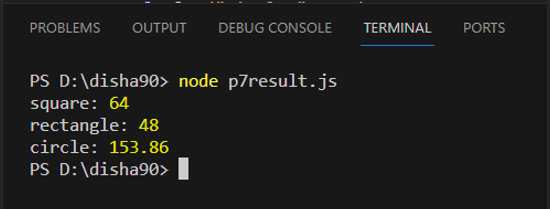
//DISPLAY RESULT

console.log("square:",sRes);

console.log("rectangle:",rRes);

console.log("circle:",cRes);

**Output :-**

****

**PRACTICAL NO: 12**

**Aim:-Write an application to demonstrate events module in Node.js**

**Code:-**

const EventEmitter = require('events');

const emitter = new EventEmitter();

//Register

emitter.on('messageLogged',function () {

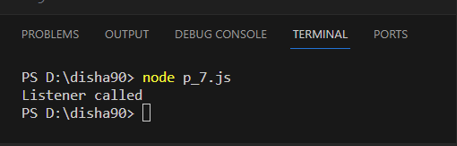
console.log('Listener called');

});

//Raise

emitter.emit('messageLogged');

**Output:**



**PRACTICAL NO: 13**

**Aim:-write an application to demonstrate function (removeListner, listnerCount) in Node.js**

**Code:-**

const events = require("events");

const eventEmitter = new events.EventEmitter();

function listner1(){

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

}

function listner2(){

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

}

eventEmitter.addListener("Write",listner1);

eventEmitter.on("Write",listner2);

eventEmitter.emit("Write");

console.log(eventEmitter.listenerCount("write"));

eventEmitter.removeListener("write",listner1);

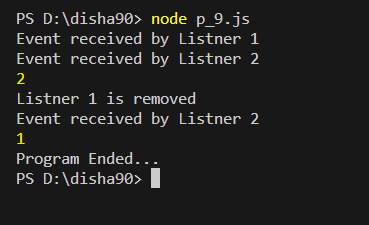
console.log("Listner1 is removed");

eventEmitter.emit("write");

console.log(eventEmitter.listenerCount("write"));

console.log("program Ended.....")

**Output:-**



**PRACTICAL NO: 14**

**Aim:- Create an application in nodejs to create Return Event Emitter.**

**Code:-**

var emitter=require('events').EventEmitter;

function LoopProcessor(num){

var e = new emitter();

setTimeout(function(){

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

e.emit('BeforProcess',i);

console.log('Processing number:'+i);

e.emit('AfterProcess',i);

}

} , 2000)

return e;

}

var lp = LoopProcessor(3);

lp.on('BeforeProcess',function(data){

console.log('About to start the process for'+data);

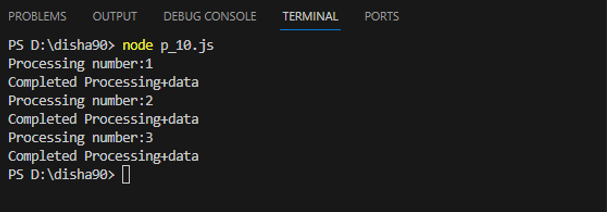
});

lp.on('AfterProcess',function(data){

console.log('Completed Processing+data');

});

**Output:-**



**PRACTICAL NO: 15**

**Aim:-create an application in node.js to create Extend Event Emitter in Node.js**

**Code:-**

var emitter=require('events').EventEmitter;

var util = require('util');

function LoopProcessor (num) {

var me = this;

setTimeout(function(){

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

me.emit ('BeforeProcess',i);

console.log('processing number: '+i);

me.emit ('After Process',i);

}

}, 2000)

return this;

}

util.inherits (LoopProcessor, emitter)

var lp = new LoopProcessor (3);

lp.on('BeforceProcess', function(data) {

console.log('About to start the process for' + data);

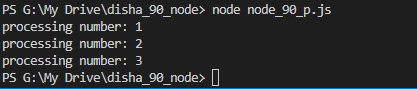
});

lp.on('AfterProcess', function(data) {

console.log('completed processing '+ data);

});

**Output:-**



**PRACTICAL NO: 16**

**Aim:- Write an event emitter code to design an event called as “calculate Salary” which is used to calculate the salary of an employee by passing some arguments like Basic Salary, HRA (20% of Basic), DA(100% of Basic), TA, and deductions like Income Tax (30% of Basic) and Professional Tax of 200.**

**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}`);

});

// Example usage:

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

8000

**Output:-**



**PRACTICAL NO: 17**

**Aim:- Create an application in nodejs to display message after 5 second & 10 second.**

**Code:-**

const myfun = delay => {

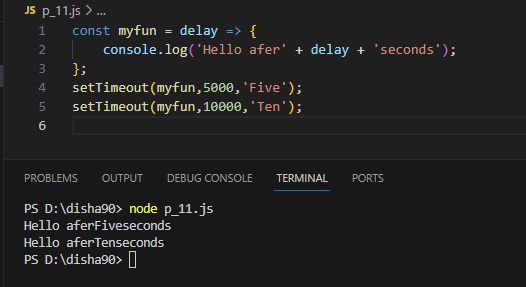
console.log('Hello afer' + delay + 'seconds');

};

setTimeout(myfun,5000,'Five');

setTimeout(myfun,10000,'Ten');

**Output:-**



**PRACTICAL NO: 18**

**Aim:- Create an application in nodejs to demonstrate set Interval Function.**

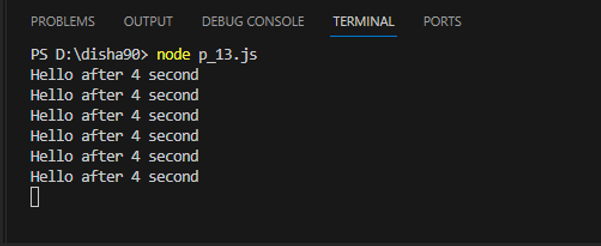
**Code:-**

setInterval(

()=> console.log('Hello after 4 second'),4000

);

**Output:-**

****

**PRACTICAL NO: 19**

**Aim:- Create an application in nodejs to display factorial of a number**

**Code:-**

function factorial(n){

let i=n;

let res =1;

while (i>+1)

{

res=res \* i;

i--

}

return res;

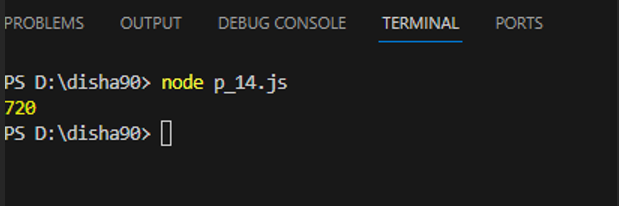
}

const num = 6;

const result = factorial(num);

console.log(result);

**Output:-**

****

**PRACTICAL NO: 20**

**Aim:- Write an application to create http server and Display message.**

**Code:-**

var http = require('http');

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

res.write("HELLO NODE JS");

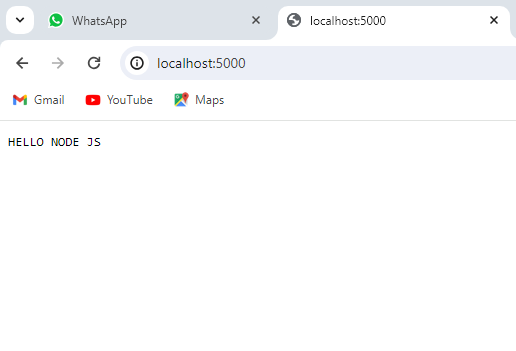
res.end();

});

server.listen(5000);

console.log('Node.js web serverat port 5000 is running...')

**Output:-**



**PRACTICAL NO: 21**

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

**Code:-**

const http = require('http');

const fs = require('fs');

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

fs.readFile('form.html', (err, data) => {

if (data) {

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

res.end(data);

}

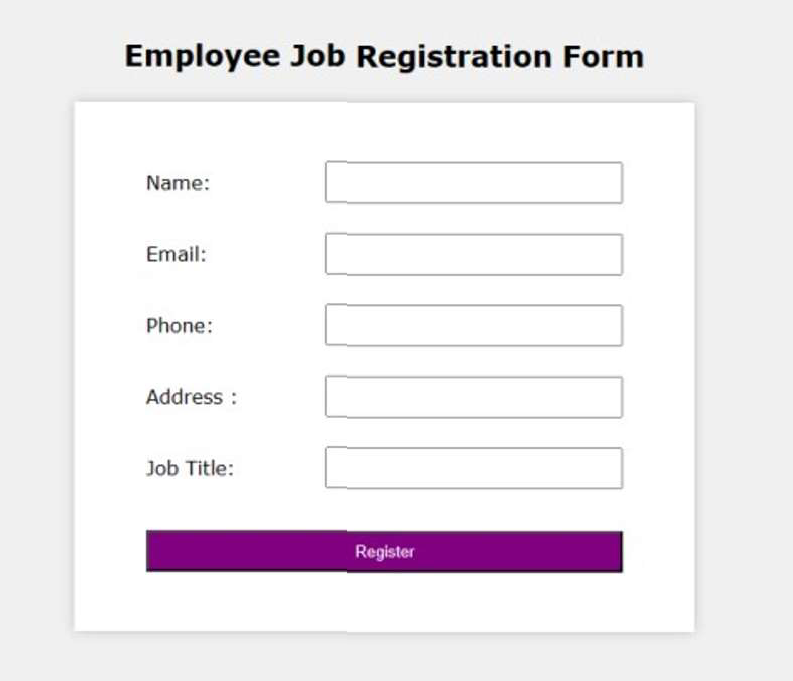
});

}).listen(8080, () => {

console.log('Server is running at http://localhost:8080');

});

**Output:-**



**PRACTICAL NO: 22**

**Aim:-Write as application to create Home page, Admin page and Student page using http server in Node.js.**

**Code:-**

var http = require('http');

const{ text } = require('stream/consumers');

var server = http.createServer(function(req,res){ if(req.url=='/'){

res.writeHead(200,{'content-type':'text/html'}); res.write('<html></head><body>');

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

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

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

res.end();

}

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

{

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

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

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

res.end();

}

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

{

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

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

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

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

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

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

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

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

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

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

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

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

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

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

res.write('<option value="Ahemdabad">Ahemdabad</option><option value="Kalol">Kalol</option><option value="Surat">Surat</option>');

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

res.end();

}

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

{

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

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

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

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

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

res.write('<legend><h1><u>Admin Login</u></h1></legend>'); res.write('<form action="#" method="POST" autocomplete="off">');

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

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

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

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

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

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

}

else{

res.end('Invalid request');

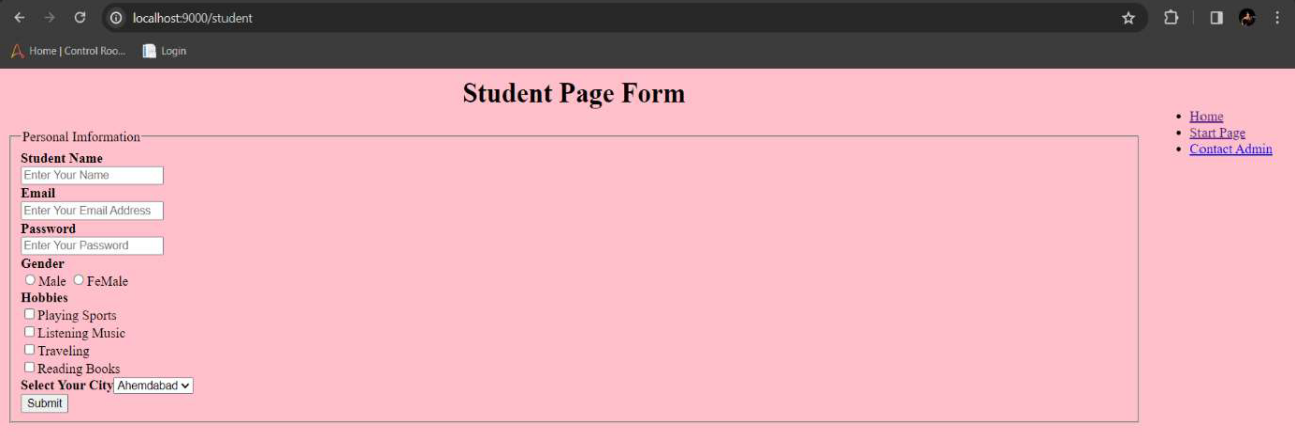
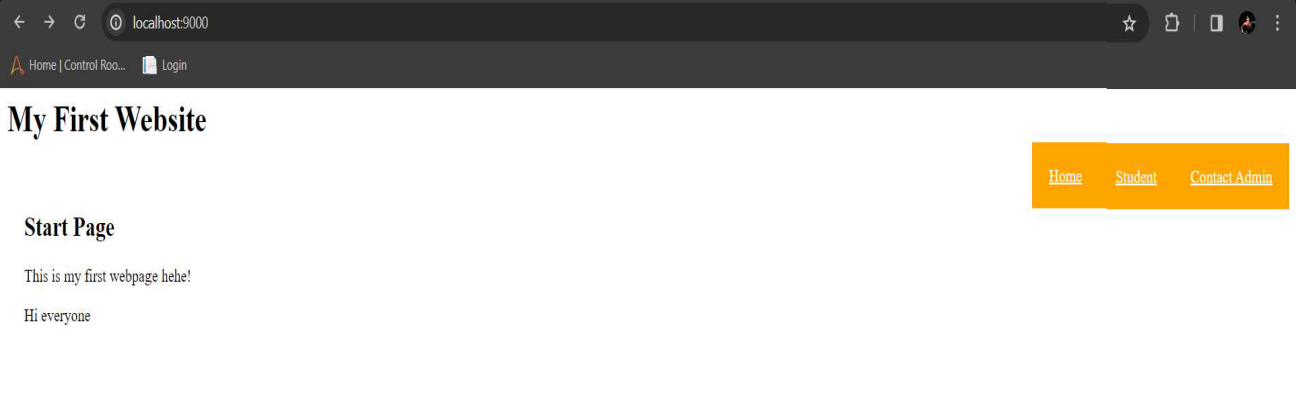
}

});

server.listen(9000);

console.log('Node.js web server at port 9000 is running');

**Output:-**



**PRACTICAL NO: 23**

**Aim:-Write in application to display details of the current file path in Node.js.**

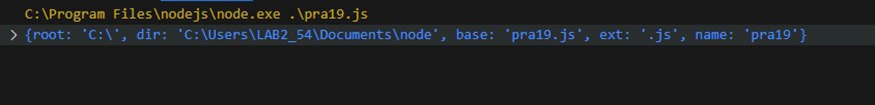
**Code:-**

const location = require("path");

const localobj = location.parse(\_\_filename);

console.log(localobj);

**Output:-**

****

**PRACTICAL NO: 24**

**Aim:-Write an application to read file in Node.js.**

**Code:-**

const fs = require('fs');

fs.readFile("\_txt.txt",'utf8',function(err,data)

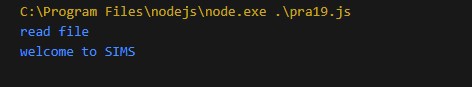
{

console.log("Reading File");

console.log(data);

});

**Output:-**



**PRACTICAL NO: 25**

**Aim:-Write an application to write in file in Node.js.**

**Code:-**

const fs = require("fs");

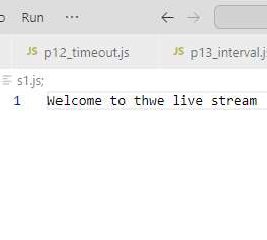
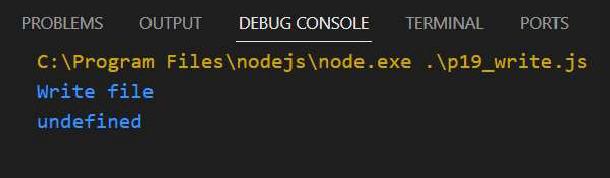
fs.writeFile("\_txt.txt",'Welcome to the live stream',function (err,data)

{

console.log("Writing File");

});

**Output:-**



**PRACTICAL NO: 26**

**Aim:- Write an application to add data in file in Node.js.**

**Code:-**

const fs = require("fs");

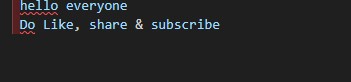
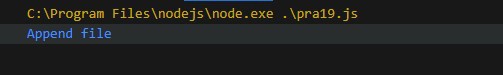
fs.appendFile("\_txt.txt","\nHello Everyone \nLet's play agian",

function (err,data){

console.log("append file");

});

**Output:-**



**PRACTICAL NO: 27**

**Aim:-Write an application to delete a file in Node.js.**

**Code:-**

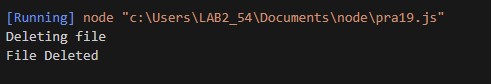
const fs=require("fs"); fs.unlink("welcome.js",function(err,data)

{

console.log("Deleting file"); console.log("File DeletdSuccecdwr");

});

**Output:-**



**PRACTICAL NO: 28**

**Aim:-Combine Read,Write,Append and Delete files in one node js program.**

**Code:-**

const fs = require("fs"); fs.writeFile("\_com.txt",'Helloworld',function (err,data)

{ console.log("Writing File");

}); fs.appendFile("\_com.txt","\nHello Everyone \nGiveThumbsUp",function (err,data) {

console.log("append file");

}); fs.readFile("\_com.txt",'utf8',function(err,data)

{ console.log("Reading File");

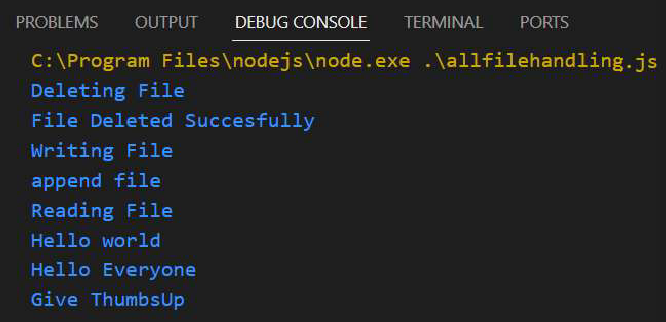
console.log(data);

}); fs.unlink("\_com.txt",function(err,data)

{ console.log("Deleting File"); console.log("File Deleted Succesfully");

});

**Output:-**



**PRACTICAL NO: 29**

**Aim:-write an application to rename a file.**

**Code:-**

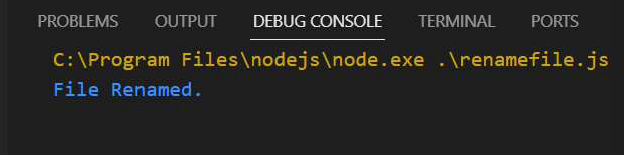
var fs = require('fs');

fs.rename('snake.js','newName.js', function (err) {

if(err) throw err; console.log('File Renamed.');

});

**Output:-**



**PRACTICAL NO: 30**

**Aim:-Create an application to create database I nodejs.**

**Code:-**

my=require("mysql"); con=my.createConnection({ host:"localhost", user:"root", password:"12345"

});

con.connect(function(err)

{

if (err)throw err;

con.query("create database fymca",function(err)

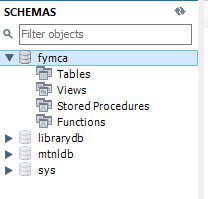
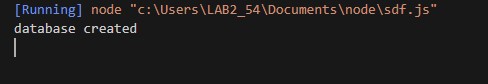
{

console.log("databse created");

});

});

**Output:-**



**PRACTICAL NO: 31**

**Aim:-Create an Application to create Student table with columns Name , roll no , class , contact in Node.js.**

**Code:-**

my=require("mysql"); con=my.createConnection({ host:"localhost", user:"root", password:"12345" database:"fymca"

});

con.connect(function(err)

{

if(err)throw err

{

con.query("create table student(name varchar(20),roll int ,class varchar(30),contact varchar(10))",function(err)

{

if (err){

console.log(err);

}

console.log(" Student

table

created in

database

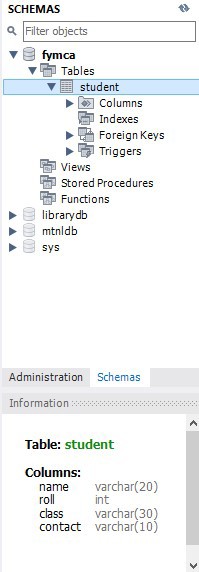
abc" );

});

}

**Output:-**





**PRACTICAL NO: 32**

**Aim:-Create an application to insert rows into student table in Nodejs.**

**Code:-**

my=require("mysql"); con=my.createConnection({ host:"localhost", user:"root", password:"12345" database:"fymca"

});

con.connect(function(err)

{

if(err)throw err

{

con.query("insert into student values('satyam',59,'mca')",function(err)

{

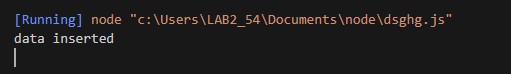
console.log("data inserted");

});

}

});

**Output:-**



**PRACTICAL NO: 33**

**Aim:-Create an application to display rows from student table in Nodejs.**

**Code:**

my=require("mysql"); con=my.createConnection({ host:"localhost", user:"root", password:"12345" database:"fymca"

});

con.connect(function(err)

{

if(err)throw err

{

con.query("select \* from student",function(err,result)

{

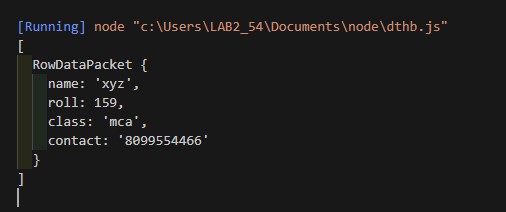
}

});

});

console.log(result);

**Output:-**



**PRACTICAL NO: 34**

**Aim:-Create an application to Update rows into student table in**

**Nodejs.**

**Code:-** my=require("mysql"); con=my.createConnection({ host:"localhost", user:"root", password:"12345" database:"fymca"

});

con.connect(function(err)

{

if(err)throw err

{

con.query("Update student set roll=59 where name='satyam'",function(err,result)

{

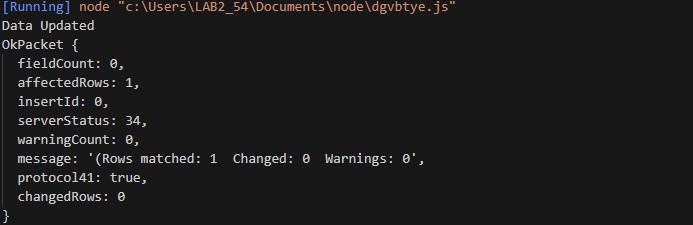
console.log("Data Updated"); console.log(result);

});

}

});

**Output:-**



**PRACTICAL NO: 35**

**Aim:-Create an application to update mobile number of student name=”xyz” into student table in Nodejs.**

**Code:-**

my=require("mysql"); con=my.createConnection({ host:"localhost", user:"root", password:"12345" database:"fymca"

});

con.connect(function(err)

{

if(err)throw err

{

con.query("Update student set contact=9321483781 where name='satyam'",function(err)

{

console.log("Data Updated");

});

con.query("select \* from student",function(err,result)

{

console.log(result);

});

}

**Output:-**

**Before update**



**After update**



**PRACTICAL NO: 36**

**Aim:-Create an application to add columns into student table in Nodejs**.

**Code:-**

my=require("mysql"); con=my.createConnection({ host:"localhost", user:"root", password:"12345" database:"fymca"

});

con.connect(function(err)

{

if(err)throw err

{

con.query("Alters table student add contact varchar(10)",function(err)

{

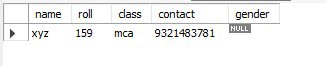
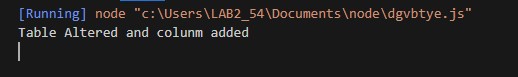
console.log("Table Altered and colunm added");

});

}

});

**Output:-**



**PRACTICAL NO: 37**

**Aim:- Create an application to Delete record from student table in Nodejs.**

**Code:-**

my=require("mysql"); con=my.createConnection({ host:"localhost", user:"root", password:"12345" database:"fymca"

});

con.connect(function(err)

{

if(err)throw err

{

con.query("Delete from student where roll=59",function(err)

{

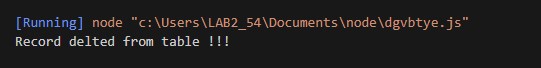
console.log("Record delted from table !!!");

});

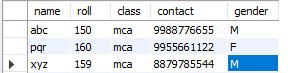
}

});

**Output:-**



**Before Deleting**



**After Deleting**



**Practical No.1**

**Aim:-Create an application in angular js to demonstrate arithmetic operations and list.**

**Code:-**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>AN 01</title>

    <script src="angular.min.js"></script>

    <style>

        \*{

            margin: 0;

            padding: 0;

            box-sizing: border-box;

        }

        body {

            padding: 20px 0;

            display: flex;

            flex-direction: column;

            align-items: center;

            justify-content: center;

            gap: 10px;

        }

        div {

            width: 700px;

            display: flex;

            flex-direction: column;

            padding: 20px;

            background-color: aqua;

        }

    </style>

</head>

<body ng-app="">

    <div>

        <h1>This is my First page </h1>

        <br> Amount = {{2+3}} <br> division = {{10/5}}

    </div>

    <div ng-init="marks=[60,70,80,90,100]">

        <h1>This is my Second page </h1>

        Subject 1 = {{marks[0]}} <br>

        Subject 2 = {{marks[1]}} <br>

        Subject 3 = {{marks[2]}} <br>

        Subject 4 = {{marks[3]}} <br>

        Subject 5 = {{marks[4]}} <br>

    </div>

    <div ng-init="people=2; reg=20">

        <h1>This is my Third page </h1>

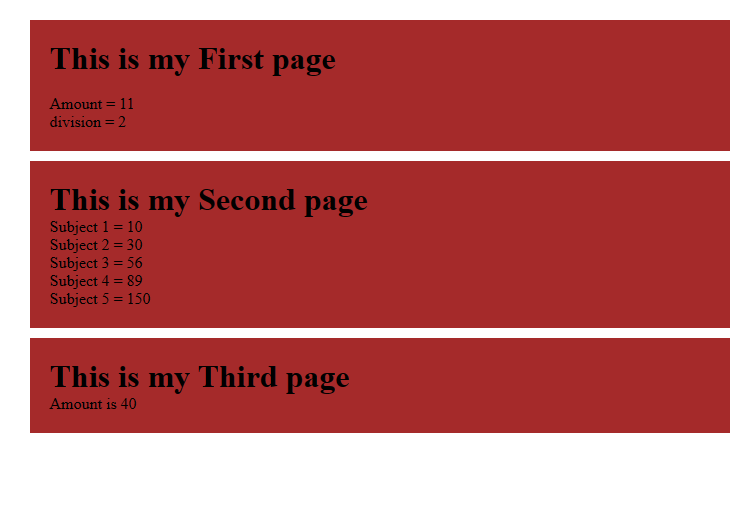
        Amount is {{people\*reg}}

    </div>

</body>

</html>

**Output :-**



**Practical 2**

**Aim:-Create an application in angular js to calculate registration fees if the number of people and registration amount is given 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>AN 02</title>

    <style>

        \*{

            margin: 0;

            padding: 0;

            box-sizing: border-box;

        }

        body {

            padding: 20px 0;

            display: flex;

            flex-direction: column;

            align-items: center;

            justify-content: center;

            gap: 10px;

        }

        div {

            width: 700px;

            display: flex;

            flex-direction: column;

            padding: 20px;

            background-color: aqua;

        }

        input {

            padding: 10px 20px;

        }

</style>

    <script src="angular.min.js"></script>

</head>

<body ng-app>

    <div>

        <h1>Calculate </h1>

        Enter the number of Poeple <input type="number" ng-model="pop">

        Enter the reg fess <input type="number" ng-model="rege">

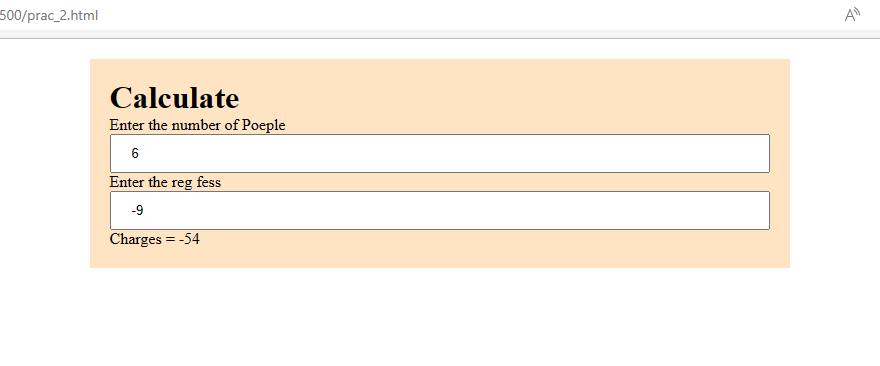
        Charges = {{pop\*rege}}

    </div>

</body>

</html>

**Output:-**



**Practical 3**

**Aim:-Create an application in angular js to calculate simple interest take appropriate input from 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>AN 03</title>

<style>

\*{

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

padding: 20px 0;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

gap: 10px;

}

div {

width: 700px;

display: flex;

flex-direction: column;

padding: 20px;

background-color: aqua;

}

input {

padding: 5px 20px;

}

</style>

<script src="angular.min.js"></script>

</head>

<body ng-app>

<div>

<h1>Calculate </h1>

<span>Principle = <input type="number" ng-model="prin"></span>

<span>Rate of Intrest = <input type="number" ng-model="roi"></span>

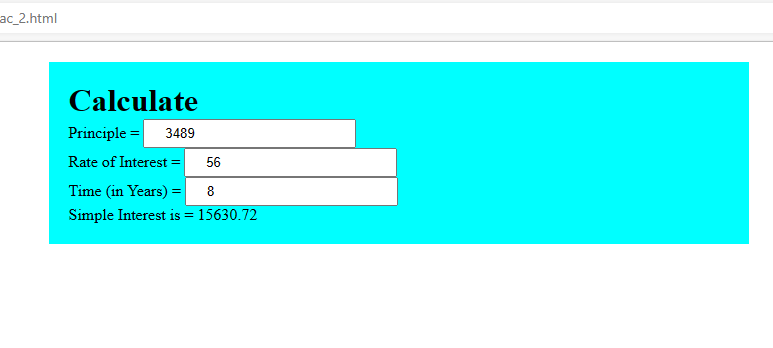
<span>Time (in Years) = <input type="number" ng-model="t"></span>

<span>Simple Inrest is = {{(prin\*roi\*t)/100}}</span>

</div>

</body>

</html>

**Output:-**

**Practical 4**

Aim:-Write an application in angular js to create an array of names and display all the names which has letter “i” using controller

Code:-

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>AN 02</title>

    <script src="angular.min.js"></script>

</head>

<body>

    <div ng-app="myApp" ng-controller="namesCtrl">

        <ul>

            <li ng-repeat="x in names|filter:'i'">{{x}}</li>

        </ul>

    </div>

    <script>

        angular.module("myApp", []).controller("namesCtrl", function ($scope) {

            $scope.names = ["Disha",

            "Chiranatan",

            "Rupali",

            "Tanji",

            "Mane",

            "Shreesha",

            "Mane"

            ];

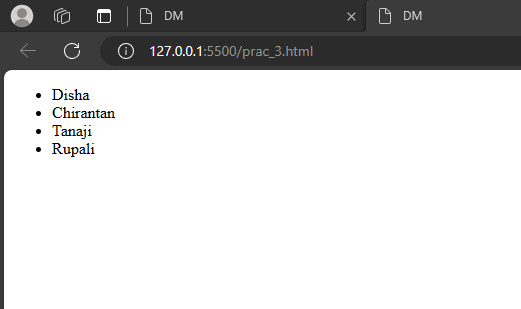
        });

    </script>

</body>

</html>

**Output:-**

****

**Practical 5**

**Aim:-Create an application in angular js to demonstrate the use of filters in angular js**

**Code:-**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>AN 04</title>

    <script src="angular.min.js"></script>

    <style>

        div {

            box-shadow: 2px 2px 2px 1px rgba(0, 0, 0, 0.2);

            padding: 16px 20px;

            background-color: aliceblue;

        }

        body {

            display: flex;

            flex-direction: column;

            gap: 20px;

            background: rgb(34, 193, 195);

            background: linear-gradient(0deg, rgba(34, 193, 195, 1) 0%, rgba(253, 187, 45, 1) 100%);

        }

    </style>

</head>

<body ng-app="myApp" ng-controller="myController">

    <div>

        <h1>Name and Number Filter </h1>

        Default Currency : {{person.salary|currency}} <br />

        Custom Currency : {{person.salary|currency:'Rs.'}} <br />

        No Fraction Currency : {{person.salary|currency:'Rs.':0}} <br />

        <!-- fraction 2 Currency : {{person.salary|currency:'GBP':2}} <br/> -->

        fraction 2 Currency : <span ng-bind="person.salary|currency:'GBP ':4"></span>

    </div>

    <div>

        <h1>Name,Number and Date Filter </h1>

        First Name : {{person.firstName|lowercase}} <br />

        Last Name : {{person.lastName|lowercase}} <br />

        Number Filter : {{person.salary|number:1}} <br />

        Long Date : {{DOB|date:'longDate'}} <br />

        Year : {{DOB|date:'yyyy'}} <br />

        Month Number : {{DOB|date:'MM'}} <br />

        Month Name: {{DOB|date:'MMMM'}} <br />

    </div>

    <div>

        <h1>Limit Filter</h1>

        Limit to get elements from Beginning :{{limitarr|limitTo:3}} <br />

        Limit to get elements from Ending :{{limitarr|limitTo:-3}} <br />

        Limit to get elements from String :{{limitarr|limitTo:4}} <br />

        <ul>

            <li ng-repeat="x in limitarr | orderBy">{{x}}</li>

        </ul>

    </div>

    <script>

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

        myApp.controller("myController", function ($scope) {

            $scope.person = { firstName: 'Raj', lastName: 'Bond', salary: 1000000.6750 };

            $scope.DOB = new Date();

            $scope.limitarr = [20, 10, 43, 5, 1, 4, 6]

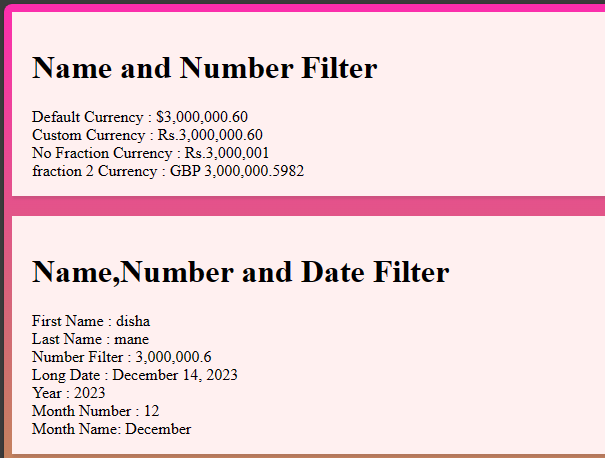
        });

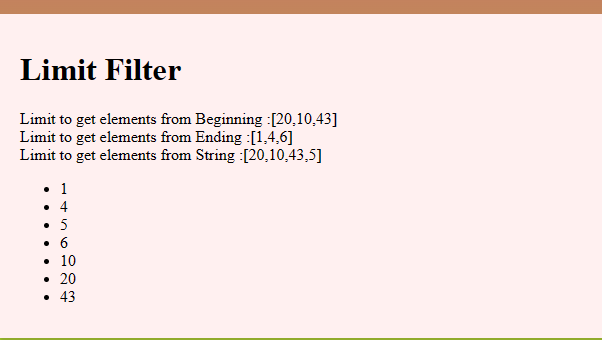
    </script>

</body>

</html>

**Output:-**





**Practical 6**

**Aim:-Create an application in angular js to change the background color as the user changes input in the text box**

**Code:-**

<!DOCTYPE html>

<html lang="en">

<head>

    <title>DM</title>

</head>

<style>

\*{

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

body{

    height: 100vh;

    display: flex;

    justify-content: center;

    align-items: center;

}

input{

    margin: 2em;

    width: 30em;

    height: 3em;

    padding: 20px;

    border: .4em solid black;

    border-radius: 20%;

    cursor: pointer;

    box-shadow: 0 0 .5em #111;

}

</style>

<body>

    <h1 class="head"> Type the color </h1>

    <!-- <input type="color" id="clr"> -->

    <input type="text" id="clr">

</body>

<script>

const bgclr = document.getElementById("clr");

const headingg = document.querySelector(".head");

bgclr.addEventListener("input", () => {

document.body.style.backgroundColor = bgclr.value;

});

</script>

</html>

**Output:-**

****

**Practical 7**

Aim:-Create an application in angular js to demonstrate to display text in alert box

Code:-

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>AN 05</title>

    <script src="angular.min.js"></script>

    <style>

        \*{

            margin: 0;

            padding: 0;

            box-sizing: border-box;

        }

        body{

            display: flex;

            align-items: center;

            justify-content: center;

            height: 100vh;

        }

        button {

            padding: 10px 20px;

            background: aqua;

            border-radius: 8px;

        }

        input{

            padding: 10px 20px;

            border-radius: 8px

        }

        div{

            display: flex;

            flex-direction: column;

            width: 500px;

        }

    </style>

</head>

<body ng-app="myApp">

    <div ng-controller="myController" class="">

        Enter Password: <input type="password" ng-model="password"/> <br />

        <button ng-click="DisplayMessage(password)">Show Password</button>

    </div>

    <script>

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

        myApp.controller("myController", function ($scope, $window) {

            $scope.DisplayMessage = function (value) {

                $window.alert(value);

            }

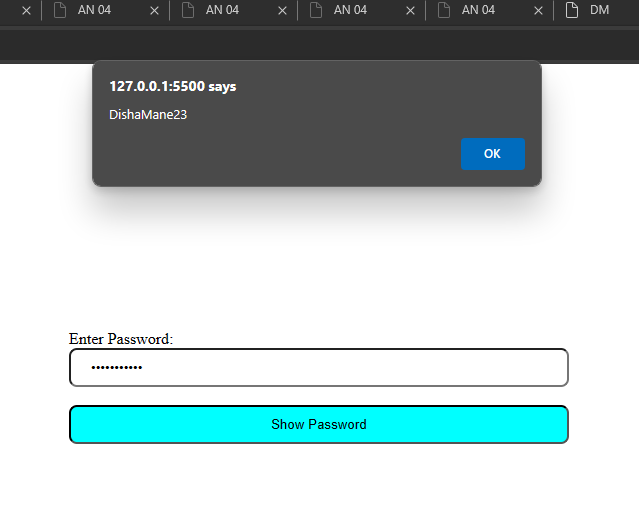
        });

    </script>

</body>

</html>

**Output:-**



**Practical 8**

**Aim:-Create an application in angular js to demonstrate the use of ng-if, ng-disabled and ng-read only**

**Code:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<script src="angular.min.js"></script>

<style>div{

width: 100%; height: 50px; display: block;

margin: 15px 0 0 10px;

}

</style>

</head>

<body ng-app ng-init="checked=true">

Click Me:<input type="checkbox" ng-model="checked"/><br/>

<div>New:<input ng-if="checked" type="text"/></div>

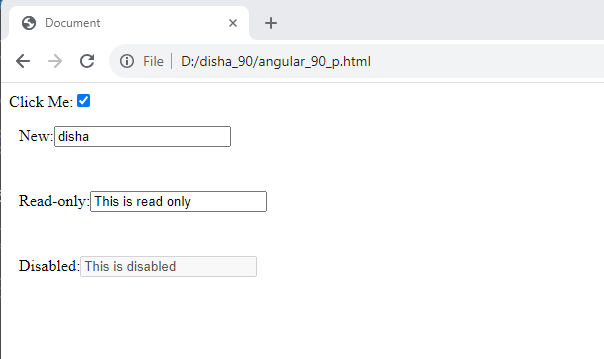
<div>Read-only:<input ng-readonly="checked" type="text" value="This is read only"/></div>

<div>Disabled:<input ng-disabled="checked" type="text" value="This is disabled"/></div>

</body>

</html>

**Output:-**



**Practical 9**

**Aim:-Create an application in angular js to demonstrate use of mouse-enter and mouse-leave event**

**Code:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<script src="angular.min.js"></script>

<style>

.redDiv{width: 100px; height: 100px; background-color: rgb(0, 255, 34); padding: 2px 2px 2px 2px;}

.yellowDiv{width: 100px; height: 100px; background-color: rgb(255, 196, 0); padding: 2px 2px 2px 2px;}

</style>

</head>

<body ng-app>

<div ng-class="{redDiv:enter,yellowDiv:leave}" ng-mouseenter="enter=true;leave=false;"

ng-mouseleave="leave=true;enter=false">Mouse <span ng-show="enter">Enter</span>

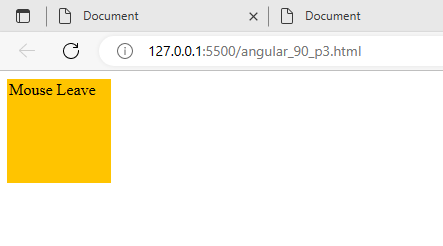
<span ng-show="leave">Leave</span>

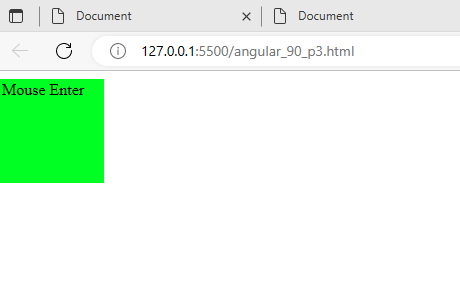
</div>

</body>

</html>

**Output:-**





**Practical 10**

**Aim:-Write an application in angular js to display options using select tag as user chooses the color option the respective color and content should change**

**Code:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>AN 08</title>

    <script src="angular.min.js"></script>

</head>

<body ng-app="">

<div><form>

    Select Color:

    <select ng-model="myVar">

        <option value="pink">Pink</option>

        <option value="blue">Sky Blue</option>

        <option value="lav">Lavender</option>

    </select>

</form>

</div>

<div ng-switch="myVar">

    <div ng-switch-when=""></div>

</div>

<div ng-switch="myVar">

<div ng-switch-when="pink" style="background-color: pink;">

<h1>Pink</h1>

<p>Pink Color</p>

</div></div>

<div ng-switch="myVar">

<div ng-switch-when="blue" style="background-color: lightblue;">

<h1>Sky Blue</h1>

<p>Sky Blue Color</p>

</div></div>

<div ng-switch="myVar">

<div ng-switch-when="lav" style="background-color: rgb(226, 184, 253);">

<h1>Lavender</h1>

<p>Lavender Color</p>

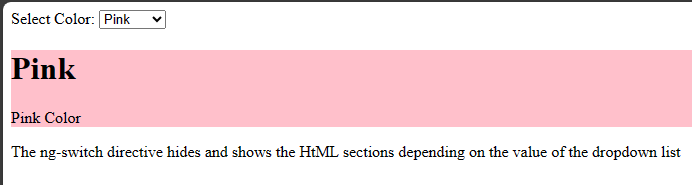
</div></div>

<p>The ng-switch directive hides and shows the HtML sections depending on the value of the dropdown list </p>

</body>

</html>

Output:-



**Practical 11**

**Aim : Write an Angular JS code to display a Registration form for Student applying for a new Course. Display all the values entered by the students.**

**Code:-**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

</head>

<body ng-controller="studentController">

<h1>Student Information:</h1>

<form ng-submit="submitStudnetForm()">

<label for="firstName">First Name: </label><br />

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

<label for="lastName">Last Name</label><br />

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

<label for="dob">DoB</label><br />

<input type="date" id="dob" ng-model="student.DoB" /> <br /><br />

<label for="gender">Gender</label> <br />

<select id="gender" ng-model="student.gender">

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

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

</select><br /> <br />

<span>Training Type:</span><br />

<label><input value="online" type="radio" name="training" ng-model="student.trainingType" />Online</label><br />

<label><input value="onsite" type="radio" name="training" ng-model="student.trainingType" />OnSite</label>

<br /><br />

<span>Subjects</span><br />

<label><input type="checkbox" ng-model="student.maths" />Maths</label> <br />

<label><input type="checkbox" ng-model="student.physics" />Physics</label> <br />

<label><input type="checkbox" ng-model="student.chemistry" />Chemistry</label><br /><br />

<input type="submit" value="Submit" ng-click="msg()" />

<input type="reset" ng-click="resetForm()" value="Reset" />

</form>

<script>

//1. create app module

var studentApp = angular.module('studentApp', []);

//2. create controller

studentApp.controller("studentController", function ($scope, $http, $window) {

//3. attach originalStudent model object

$scope.originalStudent = {

firstName: 'James',

lastName: 'Bond',

DoB: new Date('01/31/1980'),

gender: 'male',

trainingType: 'online',

maths: false,

physics: true,

chemistry: true

};

//4. copy originalStudent to student. student will be bind to a form

$scope.student = angular.copy($scope.originalStudent);

//5. create submitStudentForm() function. This will be called when user submits the form

$scope.submitStudnetForm = function () {

var onSuccess = function (data, status, headers, config) {

alert('Student saved successfully.');

};

var onError = function (data, status, headers, config) {

alert('Error occured.');

}

$http.post('/student/submitData', { student: $scope.student })

.success(onSuccess)

.error(onError);

};

//6. create resetForm() function. This will be called on Reset button click.

$scope.resetForm = function () {

$scope.student = angular.copy($scope.OriginalStudent);

};

$scope.msg = function () {

$window.alert("ho gaya karke");

}

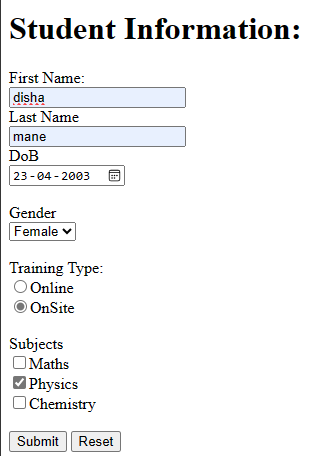
});

</script>

</body>

</html>

**Output:-**

****

**Practical 12**

**Aim:-To demonstrate the use of regular expressions for validating input fields in a form**

**Code:-**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<script src='angular.min.js'></script>

<title>Document</title>

</head>

<body ng-app="app" style="text-align:center">

<h1 style="color:green;">Regular Expression for input field</h1>

<div ng-controller="reg">

<ng-form name="num">

Input Number:

<input type="text" ng-model="number" name="number" ng-pattern="re" /><br />

<span ng-show="num.number.$error.pattern" style="color:red">

Input is not valid.

</span>

</ng-form>

</div>

</body>

<script>

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

app.controller('reg', ['$scope', function ($scope) {

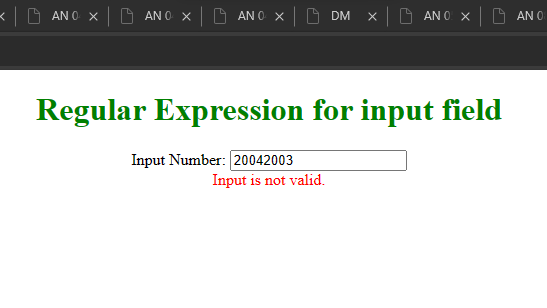
$scope.re = /^[0-9]{1,6}$/;

}]);

</script>

</html>

**Output:-**

****

**Practical 13**

**Aim : To demonstrate use of validation directives**

**Code :**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<script src='angular.min.js'></script>

</head>

<body ng-app>

<form name="studentForm" novalidate>

<label for="firstName">First Name: </label> <br />

<input type="text" name="firstName" ng-model="student.firstName" ng-required="true" />

<span ng-show="studentForm.firstName.$touched &&

studentForm.firstName.$error.required">First name is required.</span><br /><br />

<label for="lastName">Last Name</label><br />

<input type="text" name="lastName" ng-minlength="3" ng-maxlength="10" ng-model="student.lastName" />

<span ng-show="studentForm.lastName.$touched &&

studentForm.lastName.$error.minlength">min 3 chars.</span>

<span ng-show="studentForm.lastName.$touched &&

studentForm.lastName.$error.maxlength">Max 10 chars.</span><br /><br />

<label for="dob">Email</label><br />

<input type="email" id="email" ng-model="student.email" name="email" />

<span ng-show="studentForm.email.$touched && studentForm.email.$error.email">Please enter

valid email id.</span><br /><br />

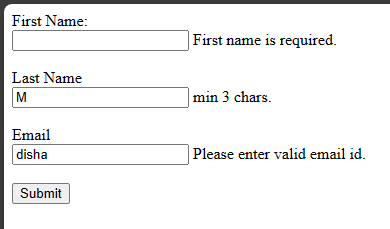
<input type="submit" value="Submit" />

</form>

</body>

</html>

**Output:-**

****

****

**Practical 14**

**Aim : To demonstrate the state properties of form fields**

**Code :**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="v iewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<script src='angular.min.js'></script>

</head>

<body ng-app>

<form name="studentForm" novalidate>

<p>

First Name Status: <br />

Pristine: {{studentForm.firstName.$pristine}} <br />

Touched: {{studentForm.firstName.$touched}}<br />

Untouched: {{studentForm.firstName.$untouched}}<br />

Valid: {{studentForm.firstName.$valid}} <br />

Invalid: {{studentForm.firstName.$invalid}} <br />

Dirty: {{studentForm.firstName.$dirty}} <br />

Error: {{studentForm.firstName.$error }} <br />

</p>

<label for="firstName">First Name: </label> <br />

<input type="text" name="firstName" ng-model="student.firstName" ng-required="true" />

<span ng-show="studentForm.firstName.$touched &&

studentForm.firstName.$error.required">First name is required.</span><br /><br />

<label for="lastName">Last Name</label><br />

<input type="text" name="lastName" ng-minlength="3" ng-maxlength="10" ng-model="student.lastName" /> <br />

<span ng-show="studentForm.lastName.$error.minlength">min 3 chars.</span>

<span ng-show="studentForm.lastName.$error.maxlength">Max 10 chars.</span> <br />

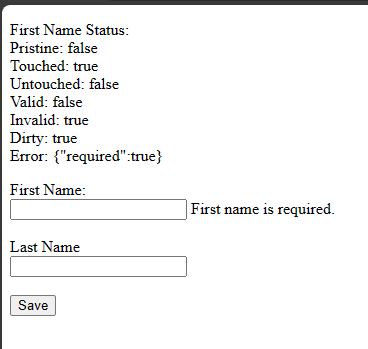
<input type="submit" value="Save" />

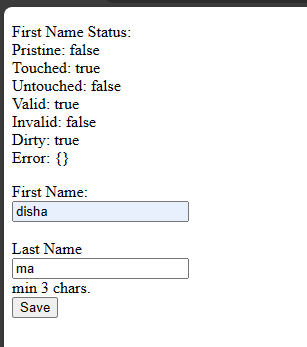
</form>

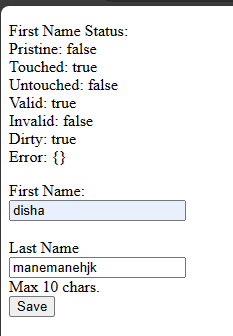
</body>

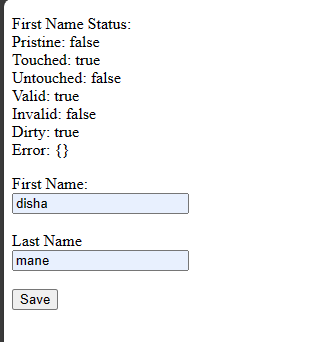
</html>

**Output:-**

****

****

****

****

**Practical 15**

**Aim : To demonstrate the use of a Single Page Application (SPA)**

**Code :**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<script src="angular.min.js"></script>

<script src="angular-route.js"></script>

</head>

<body ng-app="ngRoutingDemo">

<h1><center>Angular Routing Demo</center></h1>

<div>

<a href="#!/register">Registration</a>

<a href="#!/login">Login</a>

</div>

<div ng-view align="center"></div>

<script>

var app = angular.module('ngRoutingDemo', ['ngRoute']);

app.config(function ($routeProvider) {

$routeProvider

.when("/register",{

template:

"<h2> This is Registration Page</h2>"

})

.when("/login",{

template: "<h2> This is Login Page</h2>"

})

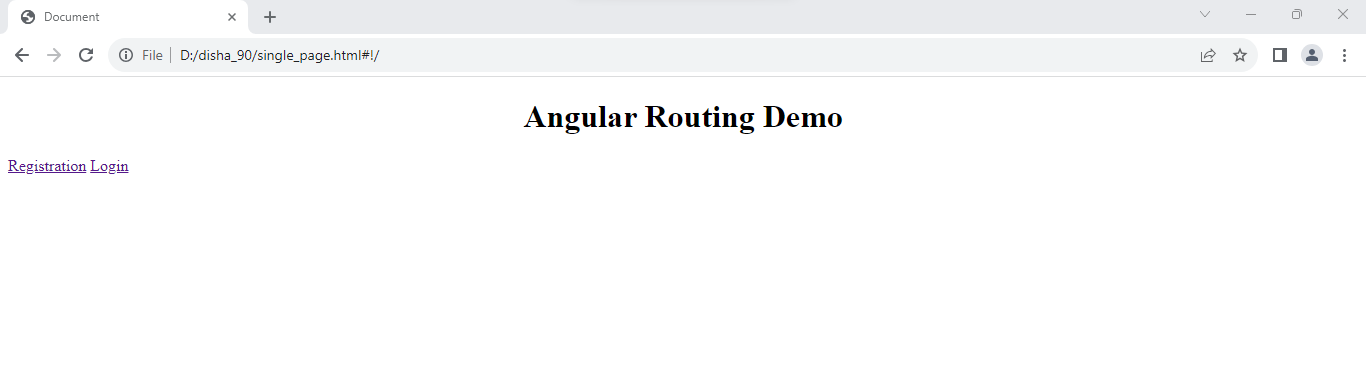
});

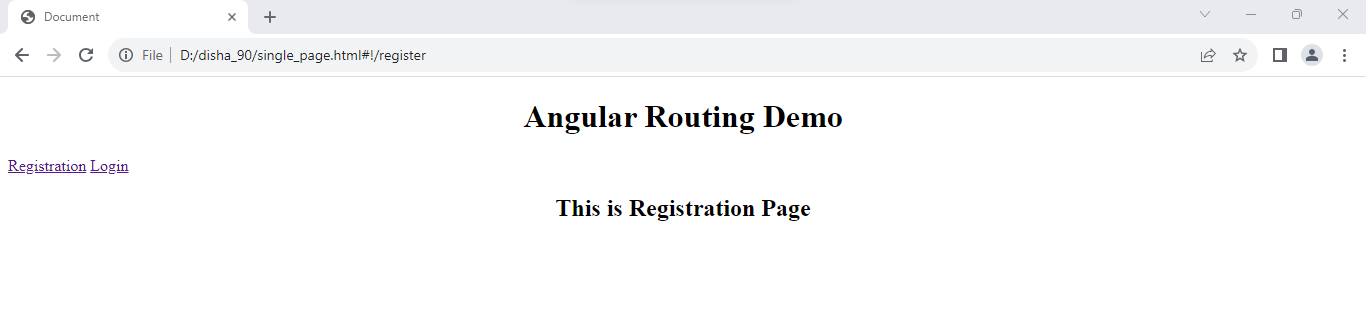
</script>

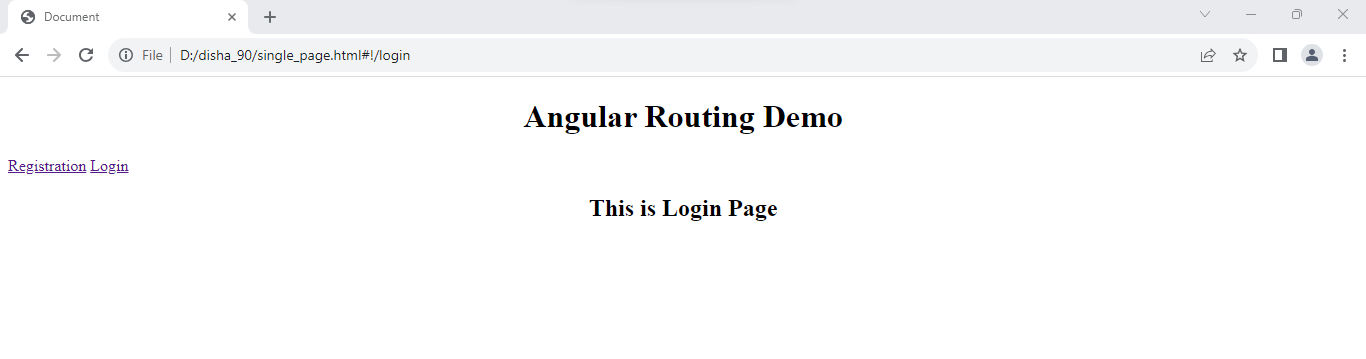
</body>

</html>

**Output:-**







**Practical 16**

**Aim :Create an application with Login page and Registration Page using Single Page**

**Application(SPA)**

**Code :**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <script src="angular.min.js"></script>

    <script src="angular.route.min.js"></script>

</head>

<body ng-app="ngRoutingDemo">

    <h1><center>Angular Routing Demo</center></h1>

    <div>

        <a href="#!/register">Registration</a>

        <a href="#!/login">Login</a>

    </div>

    <div ng-view align="center"></div>

    <script>

        var app = angular.module('ngRoutingDemo', ['ngRoute']);

        app.config(function ($routeProvider) {

        $routeProvider

        .when("/register",{

            templateUrl: "Register.html"

        })

        .when("/login",{

            templateUrl: "Login.html"

        })

        });

    </script>

</body>

</html>

Register.html Page

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

<form>

    <div>

        <h2>Registration Form</h2>

        <label for="email"><b>Email</b></label>

        <input type="text" placeholder="Enter Email" name="email" id="email" required><br><br>

        <label for="psw"><b>Password</b></label>

        <input type="password" placeholder="Enter Password" name="psw" id="psw" required><br><br>

        <label for="psw-repeat"><b>Repeat Password</b></label>

        <input type="password" placeholder="Repeat Password" name="psw-repeat" id="psw-repeat" required><br><br>

        <button type="submit" class="registerbtn">Register</button>

    </div>

</form>

</body>

</html>

Login.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <form>

        <div class="container">

            <label for="uname"><b>Username</b></label>

            <input type="text" placeholder="Enter Username" name="uname" required><br><br>

            <label for="psw"><b>Password</b></label>

            <input type="password" placeholder="Enter Password" name="psw" required><br><br>

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

          </div>

    </form>

</body>

</html>

**Output:**

