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

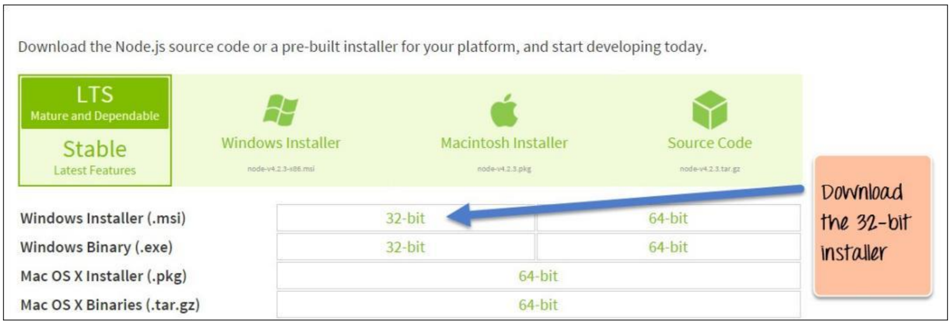
Sr. No.	Practical Name	Date
1	Create an application in angular.js to demonstrate arithmetic operations and list.	06/09/2023
2	Create an application in angular.js to calculate registration fees if the number of people and registration amount is given by the user	06/09/2023
3	Create an application in angular.js 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-1

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.



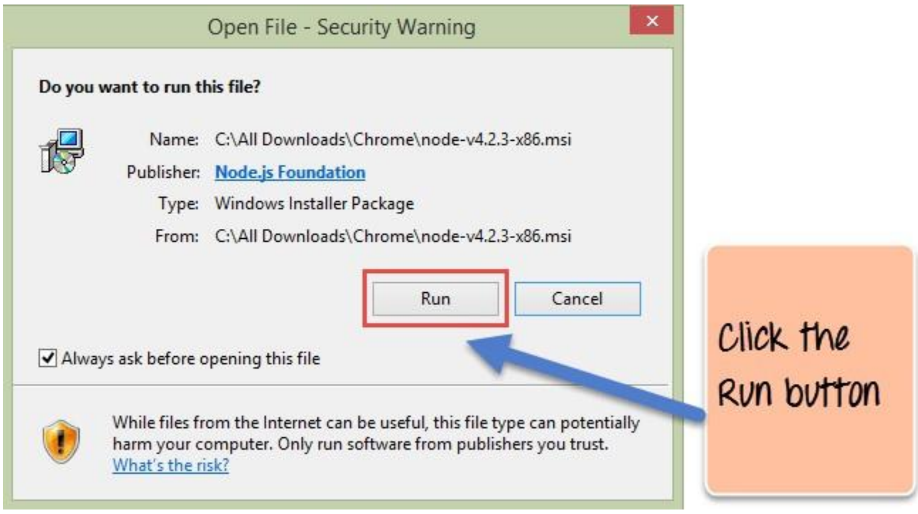
Download the Node.js source code or a pre-built installer for your platform, and start developing today.

Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.exe)	32-bit	64-bit
Mac OS X Installer (.pkg)	64-bit	
Mac OS X Binaries (.tar.gz)	64-bit	

Download the 32-bit installer

- 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.



Open File - Security Warning

Do you want to run this file?

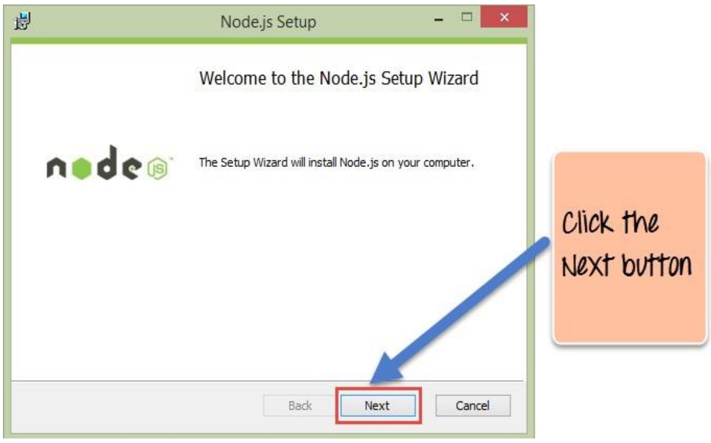
Name: C:\All Downloads\Chrome\node-v4.2.3-x86.msi
Publisher: Node.js Foundation
Type: Windows Installer Package
From: C:\All Downloads\Chrome\node-v4.2.3-x86.msi

☑ Always ask before opening this file

While files from the Internet can be useful, this file type can potentially harm your computer. Only run software from publishers you trust. What's the risk?

Click the Run button

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



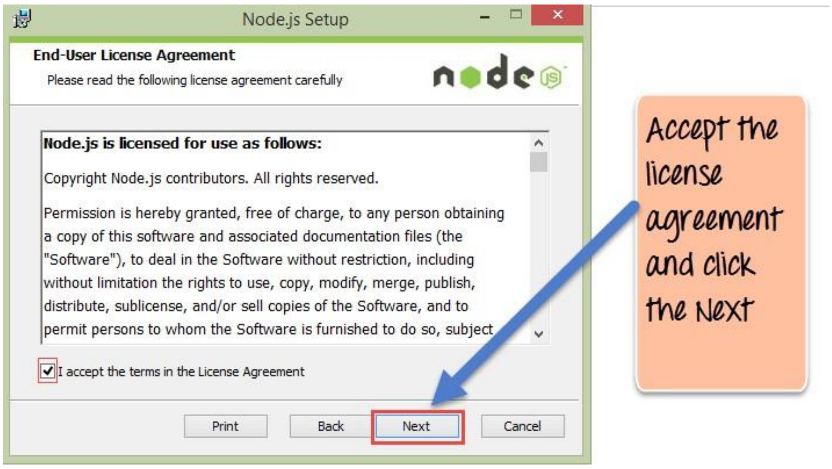
Node.js Setup

Welcome to the Node.js Setup Wizard

The Setup Wizard will install Node.js on your computer.

Click the Next button

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



Node.js Setup

End-User License Agreement

Please read the following license agreement carefully

Node.js is licensed for use as follows:

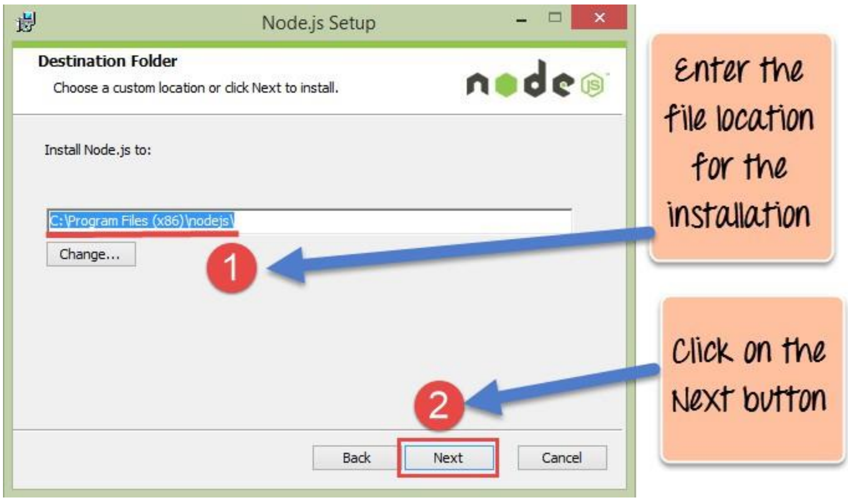
Copyright Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject

☑ I accept the terms in the License Agreement

Accept the license agreement and click the Next

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.



Node.js Setup

Destination Folder

Choose a custom location or click Next to install.

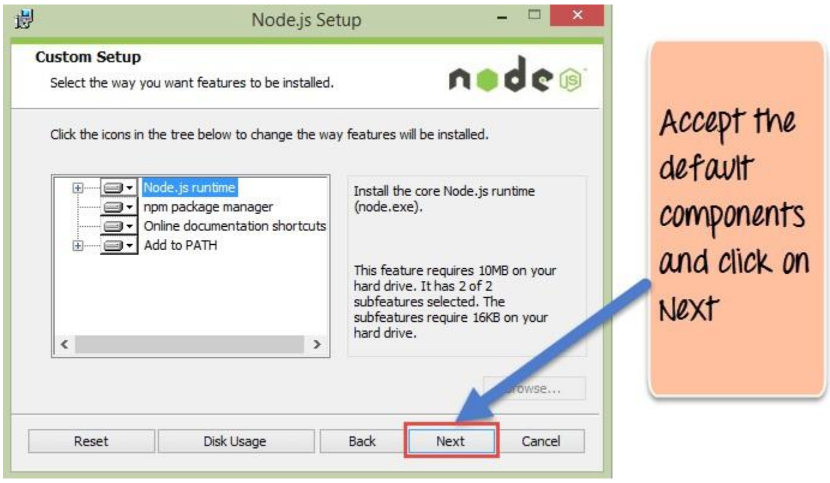
Install Node.js to:

C:\Program Files (x86)\nodejs\

Enter the file location for the installation

Click on the Next button

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



Node.js Setup

Custom Setup

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

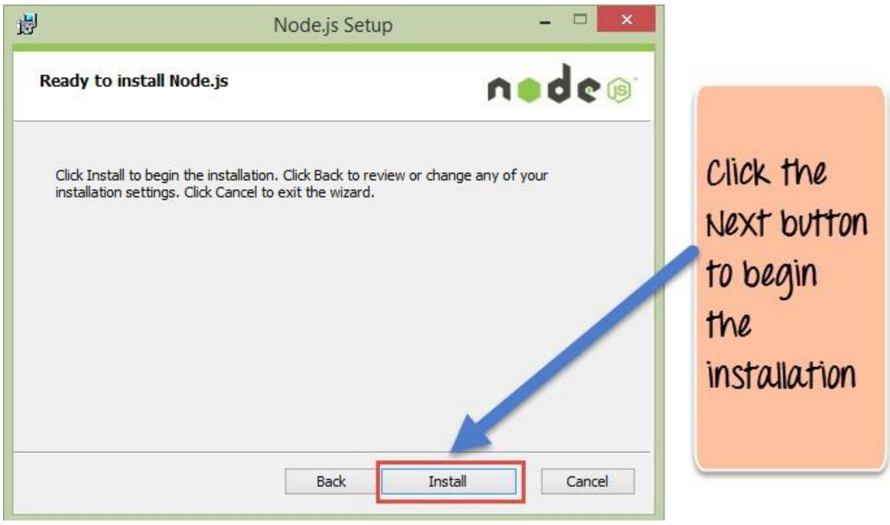
- Node.js runtime
- npm package manager
- Online documentation shortcuts
- Add to PATH

Install the core Node.js runtime (node.exe).

This feature requires 10MB on your hard drive. It has 2 of 2 subfeatures selected. The subfeatures require 16KB on your hard drive.

Accept the default components and click on Next

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



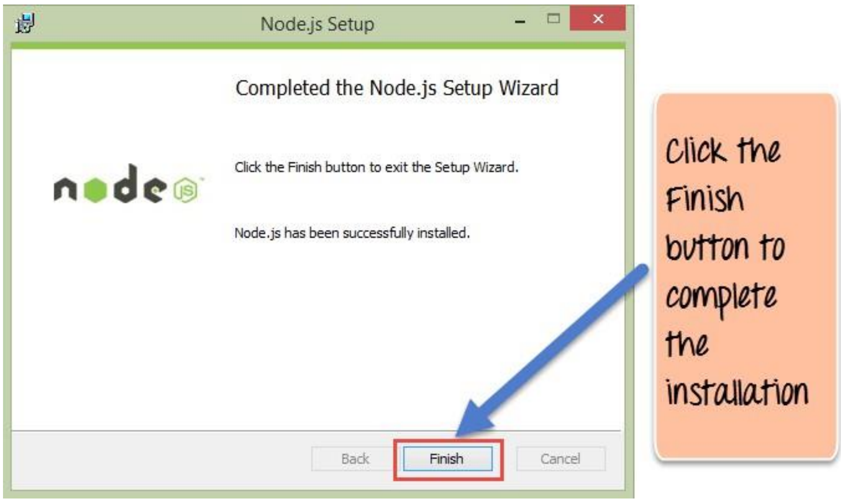
Node.js Setup

Ready to install Node.js

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

Click the Next button to begin the installation

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



Node.js Setup

Completed the Node.js Setup Wizard

Click the Finish button to exit the Setup Wizard.

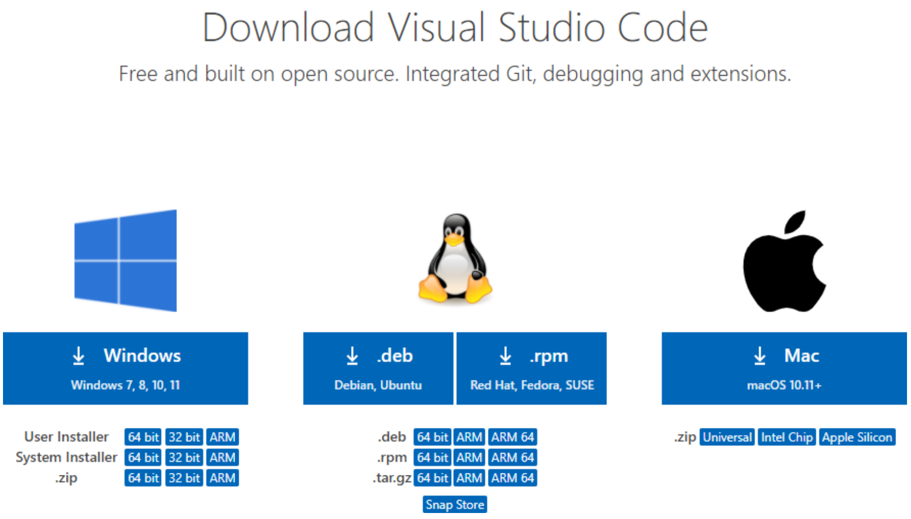
Node.js has been successfully installed.

Click the Finish button to complete the installation

PRACTICAL-2

Steps to download visual studio

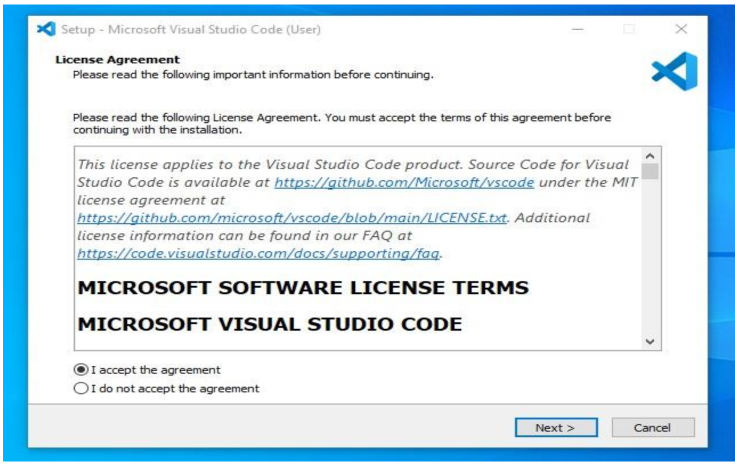
Step 1: Visit the official website of the Visual Studio Code using any web browser like Google Chrome, Microsoft Edge, etc. and Press the “Download for Windows” button



Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

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 click the button.



Setup - Microsoft Visual Studio Code (User)

License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

This license applies to the Visual Studio Code product. Source Code for Visual Studio Code is available at https://github.com/Microsoft/vscode under the MIT license agreement at https://github.com/microsoft/vscode/blob/main/LICENSE.txt. Additional license information can be found in our FAQ at https://code.visualstudio.com/docs/supporting/faq.

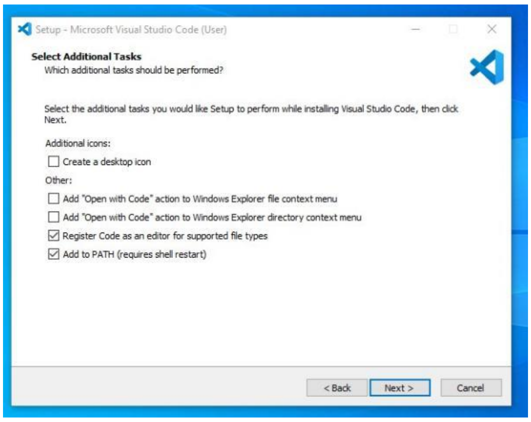
MICROSOFT SOFTWARE LICENSE TERMS

MICROSOFT VISUAL STUDIO CODE

● I accept the agreement

○ I do not accept the agreement

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.



Setup - Microsoft Visual Studio Code (User)

Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.

Additional icons:

□ Create a desktop icon

Other:

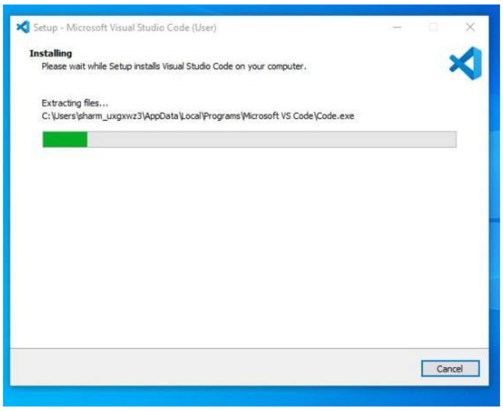
□ Add "Open with Code" action to Windows Explorer file context menu

□ Add "Open with Code" action to Windows Explorer directory context menu

☑ Register Code as an editor for supported file types

☑ Add to PATH (requires shell restart)

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.



Setup - Microsoft Visual Studio Code (User)

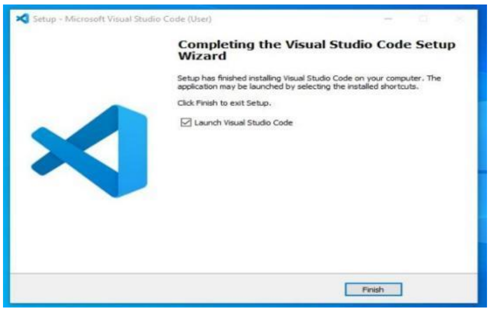
Installing

Please wait while Setup installs Visual Studio Code on your computer.

Extracting files...

C:\Users\sharm_uxgxwz3\AppData\Local\Programs\Microsoft VS Code\Code.exe

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.



Setup - Microsoft Visual Studio Code (User)

Completing the Visual Studio Code Setup Wizard

Setup has finished installing Visual Studio Code on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

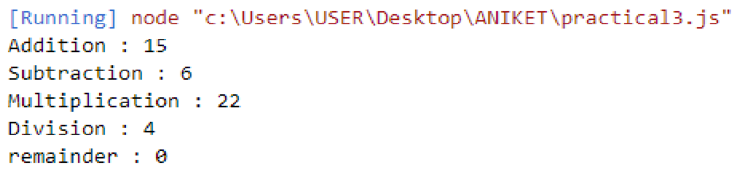
☑ Launch Visual Studio Code

PRACTICAL-3

Aim: Write an application to demonstrate Arithmetic Operation(+,-,*,/,%)

```
function sum(a,b)
{
    return a+b;
}
p=sum(10,5);
console.log("Addition : "+p);

//subtraction
function subtract(c,d)
{
    return c-d;
}
q=subtract(11,5);
console.log("Subtraction : "+q);

//Multiplication
function multiply(e,f)
{
    return e*f;
}
r=multiply(11,2);
console.log("Multiplication : "+r);

//division
function divi(g,h)
{
    return g/h;
}
t=divi(8,2);
console.log("Division : "+t);

//remainder
function rema(i,j)
{
    return i%j;
}
u=rema(8,2);
console.log("remainder : "+u);
```

Output:



```
[Running] node "c:\Users\USER\Desktop\ANIKET\practical3.js"
Addition : 15
Subtraction : 6
Multiplication : 22
Division : 4
remainder : 0
```

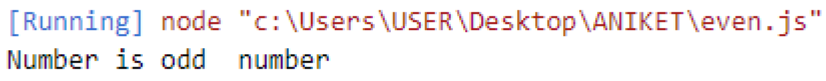
PRACTICAL-4

Aim: Write an application to check number is odd or even:

Code:

```
// to check the number is Even or Odd number
var number=3;
if(number%2==0)
{
    console.log("Number is even number");
}
else
{
    console.log("Number is odd  number");
}
```

Output:



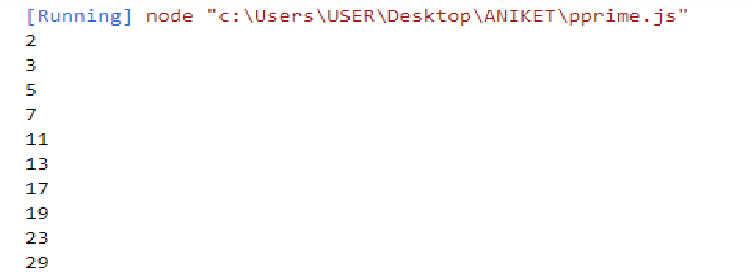
```
[Running] node "c:\Users\USER\Desktop\ANIKET\even.js"
Number is odd  number
```

PRACTICAL-5

Aim: Write program to check the number is prime or not:

```
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:



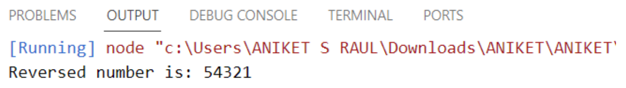
```
[Running] node "c:\Users\USER\Desktop\ANIKET\pprime.js"
2
3
5
7
11
13
17
19
23
29
```

PRACTICAL-6

Aim: Create an application in node.js to reverse the given number

```
var number = 12345;

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

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

Output:



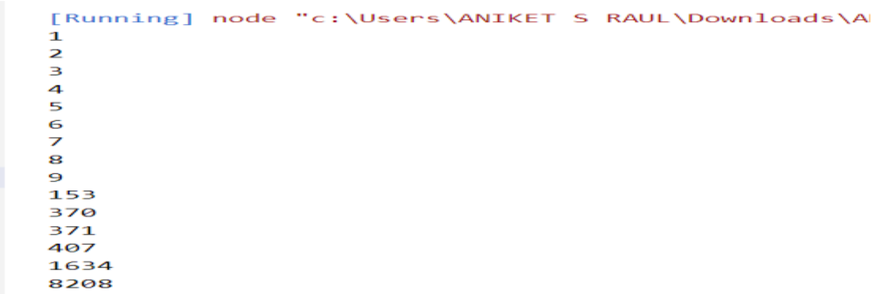
```
[Running] node "c:\Users\ANIKET S RAUL\Downloads\ANIKET\ANIKET
Reversed number is: 54321
```

PRACTICAL-7

Aim: Create an application in node.js to display armstrong number

```
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:



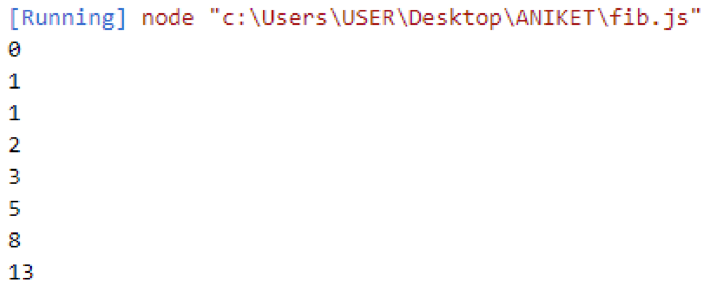
```
[Running] node "c:\Users\ANIKET S RAUL\Downloads\A
1
2
3
4
5
6
7
8
9
153
370
371
407
1634
8208
```

PRACTICAL-8

Aim: Create an application in node.js to print Fibonacci series

```
//Program to print fibonacci series
var x=0;
var y=1;
console.log(x);
console.log(y);
for(i=0;i<=5;i++)
{
    var z=x+y;
    console.log(z);
    x=y;
    y=z;
}
```

Output:



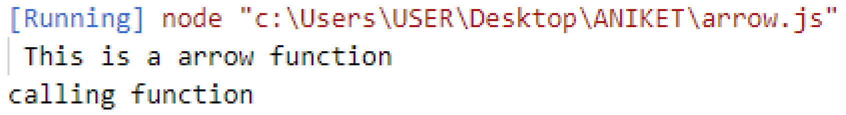
```
[Running] node "c:\Users\USER\Desktop\ANIKET\fib.js"
0
1
1
2
3
5
8
13
```

PRACTICAL-9

Aim : Create an application in node.js to demonstrate use of set-timeout & arrow function:

```
const message=function()
{
    console.log(" This is a arrow function");
}
setTimeout(message,5000);

setTimeout(()=>
{
 console.log("calling function ");
},8000
);
```

Output:



```
[Running] node "c:\Users\USER\Desktop\ANIKET\arrow.js"
 This is a arrow function
calling function
```

PRACTICAL-10

Aim: To demonstrate module exports in node.js

Code:

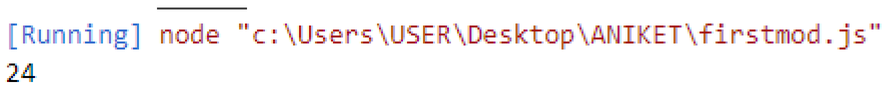
- create the file first.js

```
//demonstrate the module

function add(a,b)
{
    return a+b;
}
exports.add=add;
```

- create firstmod.js

```
var req=require('./first.js');
var res=req.add(15,9);
console.log(res)
```

Output:



```
[Running] node "c:\Users\USER\Desktop\ANIKET\firstmod.js"
24
```

PRACTICAL-11

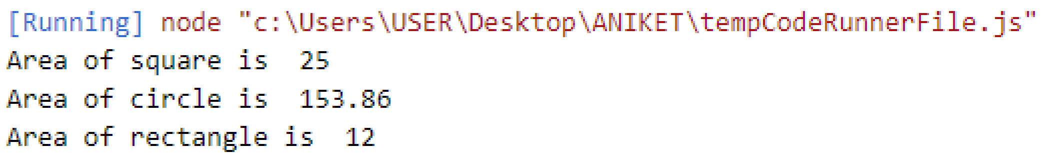
Aim: Write an application to find the area of square circle rectangle using module in node.js :

Code:

```
// SQUARE
 function square(s)
 {
  return s*s;
 }

 // circle
 function circle(r)
 {
  return 3.14*r*r;
 }

 //rectangle

 function rectangle(l,b)
 {
  return l*b;
 }
  exports.square=square;
 exports.circle=circle;
 exports.rectangle=rectangle;
```

Step2: Create file “result.js” and write following code:

```
var req=require('./area.js');
var sqa=req.square(5);
var cir=req.circle(7);
var rec=req.rectangle(4,3);
console.log("Area of square is ",sqa);
console.log("Area of circle is ",cir);
console.log("Area of rectangle is ",rec);
```



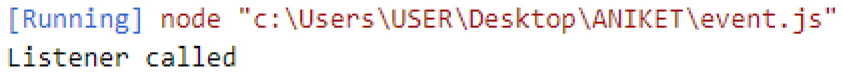
```
[Running] node "c:\Users\USER\Desktop\ANIKET\tempCodeRunnerFile.js"
Area of square is  25
Area of circle is  153.86
Area of rectangle is  12
```

PRACTICAL-12

Aim: Create an application in node.js to demonstrate events module

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

const emitter=new eventi();

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

    console.log('Listener called')

});

emitter.emit('messageLogged');
```

Output:



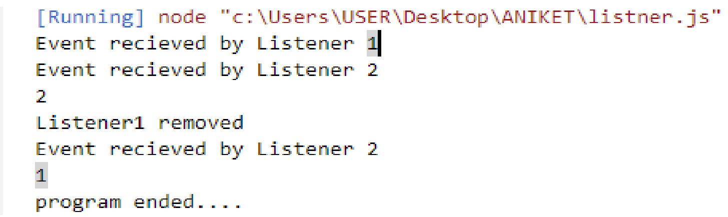
```
[Running] node "c:\Users\USER\Desktop\ANIKET\event.js"
Listener called
```

PRACTICAL-13

Aim: write an application to demonstrate function(removeListner,listnerCount)

```
const events =require("events");

const evente = new events.EventEmitter();

function listner1()

{

    console.log("Event recieved by Listener 1");

}

function listner2()

{

    console.log("Event recieved by Listener 2");

}

evente.addListener("write",listner1);

evente.on("write",listner2);

evente.emit("write",listner2);

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

evente.removeListener("write",listner1);

console.log("Listener1 removed");

evente.emit("write");

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

console.log("program ended....");
```

Output:



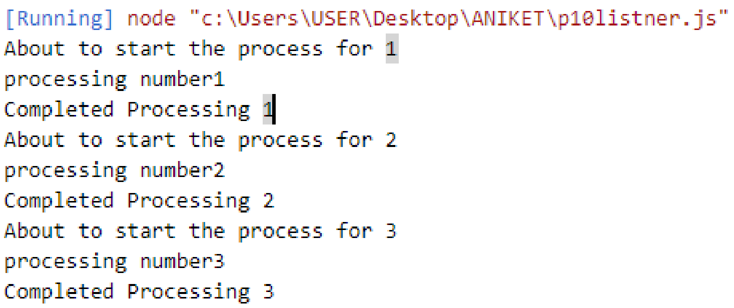
```
[Running] node "c:\Users\USER\Desktop\ANIKET\listner.js"
Event recieved by Listener 1
Event recieved by Listener 2
2
Listener1 removed
Event recieved by Listener 2
1
program ended....
```

PRACTICAL-14

AIM:Create an application in nodejs to create Return Event Emitter.

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

function loop(num)

{

    var e= new emitter();

    setTimeout(function()

    {

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

         {

            e.emit('BeforeProcess' ,i );

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

            e.emit('AfterProcess',i);

         }

    },2000

    )

    return e;

}

 var lp= loop(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:



```
[Running] node "c:\Users\USER\Desktop\ANIKET\p10listner.js"
About to start the process for 1
processing number1
Completed Processing 1
About to start the process for 2
processing number2
Completed Processing 2
About to start the process for 3
processing number3
Completed Processing 3
```

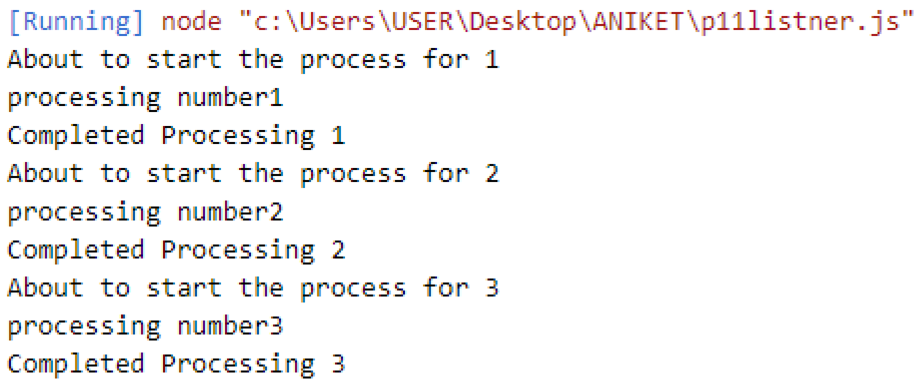
PRACTICAL-15

AIM: Create an application in nodejs to create Extent Event Emitter.

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

var util =require('util');

function loop(num)

{

    var e=this;

    setTimeout(function()

    {

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

         {

            e.emit('BeforeProcess' ,i );

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

            e.emit('AfterProcess',i);
```

```
        }

    },2000

    )

    return this;

}

util.inherits(loop,emitter)

 var lp= new loop(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:



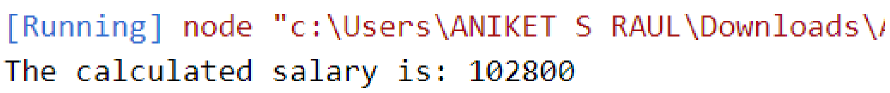
```
[Running] node "c:\Users\USER\Desktop\ANIKET\p11listner.js"
About to start the process for 1
processing number1
Completed Processing 1
About to start the process for 2
processing number2
Completed Processing 2
About to start the process for 3
processing number3
Completed Processing 3
```

PRACTICAL-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.

```
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:



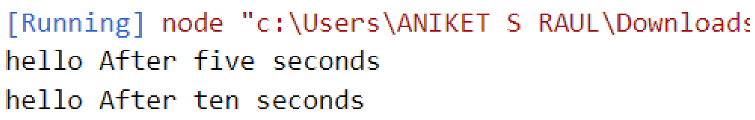
```
[Running] node "c:\Users\ANIKET S RAUL\Downloads\
The calculated salary is: 102800
```

PRACTICAL-17

AIM: Create an application in node.js to display message after 5 second & 10 second.

```
const myfun =delay =>{

   console.log('hello After ' + delay + ' seconds');
}
};

setTimeout(myfun,5000,'five');

setTimeout(myfun,10000,'ten');
```

Output:



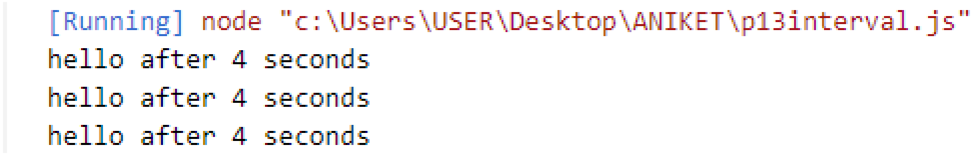
```
[Running] node "c:\Users\ANIKET S RAUL\Downloads
hello After five seconds
hello After ten seconds
```

PRACTICAL-18

AIM: Create an application in node.js to demonstrate set Interval Function.

```
setInterval

(

    () =>console.log("hello after 4 seconds"),4000

);
```

Output:



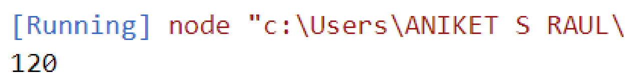
```
[Running] node "c:\Users\USER\Desktop\ANIKET\p13interval.js"
hello after 4 seconds
hello after 4 seconds
hello after 4 seconds
```

PRACTICAL-19

AIM: Create an application in node.js to demonstrate Factorial of number

```
function factorial(n) {

    let i = n;
    let res = 1; while (i > +1) {

        res = res * i; i--

    }

    return res;

}


const num = 5;

const result = factorial(num); console.log(result);
```

Output:



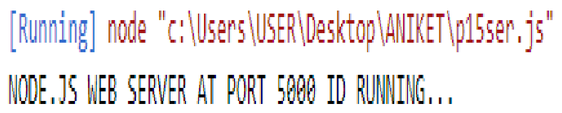
```
[Running] node "c:\Users\ANIKET S RAUL\
120
```

PRACTICAL-20

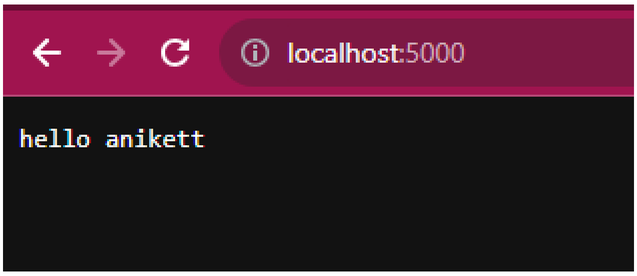
AIM: Write as application to create http Server and Display message.

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

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

    res.write("hello anikett");

    res.end();

});

server.listen(5000);

console.log("NODE.JS WEB SERVER AT PORT 5000 ID RUNNING...");
```

Output:



```
[Running] node "c:\Users\USER\Desktop\ANIKET\p15ser.js"
NODE.JS WEB SERVER AT PORT 5000 ID RUNNING...
```



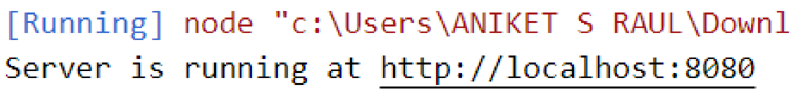
```
hello anikett
```

PRACTICAL-21

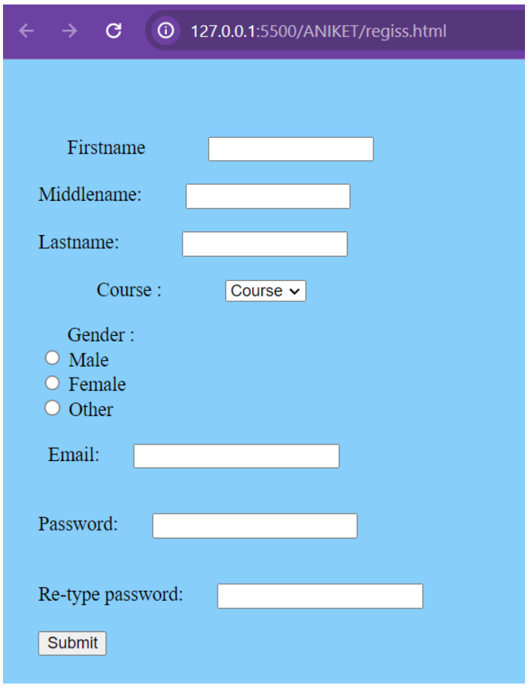
AIM: Write an node.js code to display Employee job registration form saved in HTML file in resonse to clients access request to the server.

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


http.createServer((req, res) => {
    fs.readFile('regiss.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:



```
[Running] node "c:\Users\ANIKET S RAUL\Downl
Server is running at http://localhost:8080
```



127.0.0.1:5500/ANIKET/regiss.html

Firstname ▭

Middlename: ▭

Lastname: ▭

Course : Course ⌄

Gender :
○ Male
○ Female
○ Other

Email: ▭

Password: ▭

Re-type password: ▭

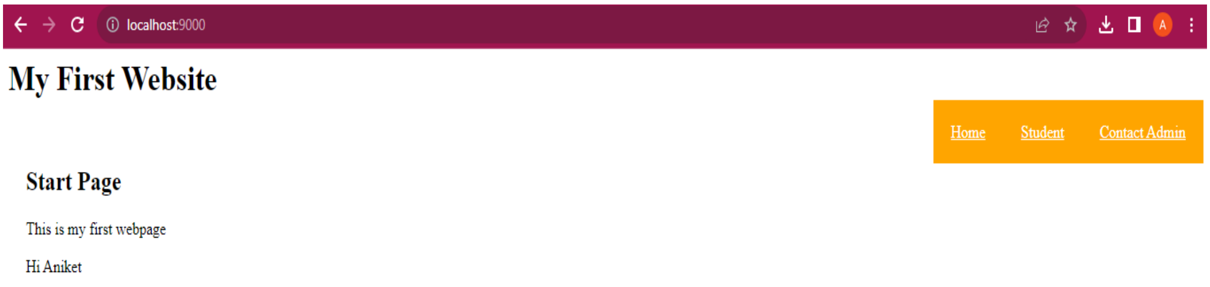
Submit

PRACTICAL-22

AIM: Write as application to create Home page, Admin page and Student page using http server.

```
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 </p><p>Hi Aniket</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>Aniket Raul</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:cyan;}</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:



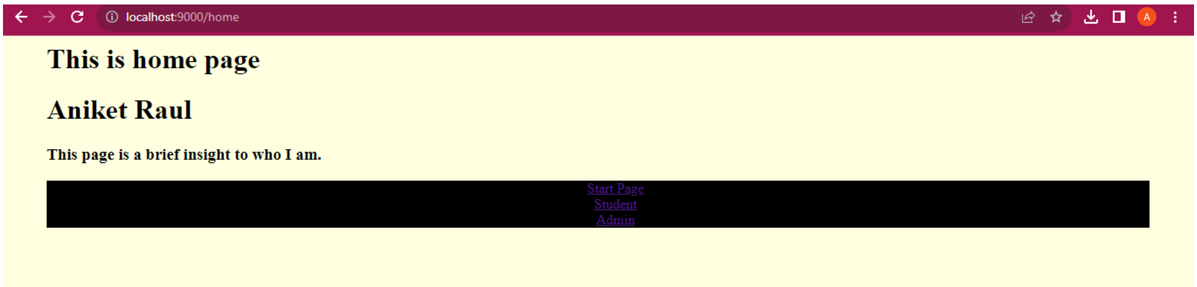
My First Website

- Home
- Student
- Contact Admin

Start Page

This is my first webpage

Hi Aniket

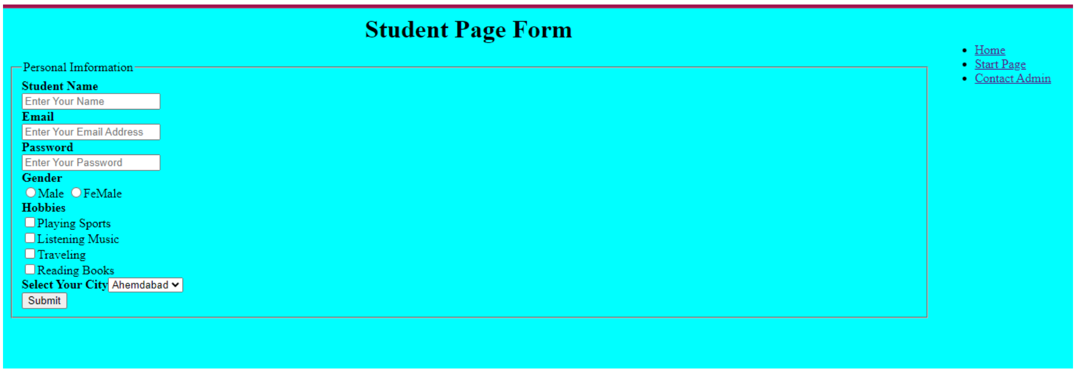


This is home page

Aniket Raul

This page is a brief insight to who I am.

Start Page
Student
Admin

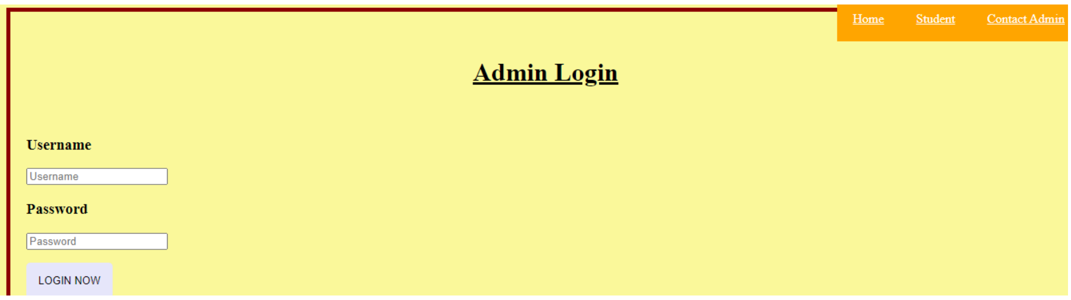


Student Page Form

- Home
- Start Page
- Contact Admin

Personal Imformation

Student Name
Enter Your Name
Email
Enter Your Email Address
Password
Enter Your Password
Gender
○ Male ○ FeMale
Hobbies
□ Playing Sports
□ Listening Music
□ Traveling
□ Reading Books
Select Your City Ahemdabad
Submit



Home Student Contact Admin

Admin Login

Username

Username

Password

Password

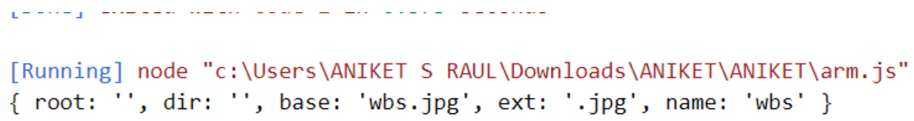
LOGIN NOW

PRACTICAL-23

AIM: Write an application to display details of current path

```
const location = require("path");

const localobj = location.parse("wbs.jpg");

console.log(localobj);
```

Output:

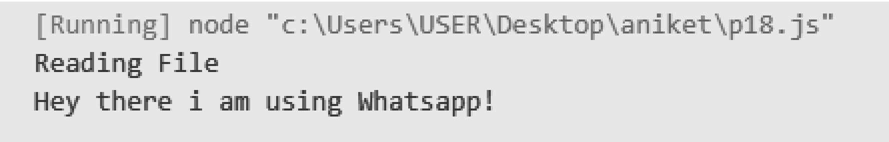


```
[Running] node "c:\Users\ANIKET S RAUL\Downloads\ANIKET\ANIKET\arm.js"
{ root: '', dir: '', base: 'wbs.jpg', ext: '.jpg', name: 'wbs' }
```

PRACTICAL-24

Aim:write node.js program to read a file

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

 fs.readFile("ani.js",'utf8',function(err,data)

 {

     console.log("Reading File");

     console.log(data);

 });
```



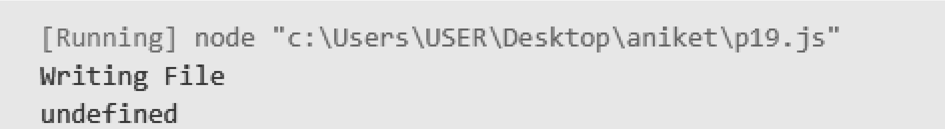
```
[Running] node "c:\Users\USER\Desktop\aniket\p18.js"
Reading File
Hey there i am using Whatsapp!
```

PRACTICAL-25

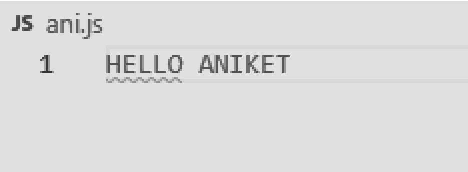
Aim: write node js program to write a file

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

 fs.writeFile("ani.js",'HELLO ANIKET',function(err,data)

 {

    console.log("Writing File");

    console.log(data);

 });
```

Output:



```
[Running] node "c:\Users\USER\Desktop\aniket\p19.js"
Writing File
undefined
```



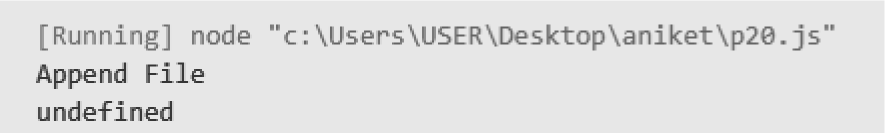
```
JS ani.js
  1    HELLO ANIKET
```

PRACTICAL-26

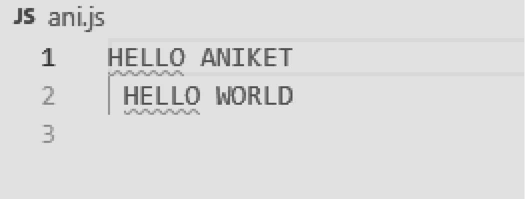
Aim: write node js program to Append a file

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

 fs.appendFile("ani.js","\n HELLO WORLD\n",function(err,data)

 {

    console.log("Append File");

    console.log(data);

 });
```

Output:



```
[Running] node "c:\Users\USER\Desktop\aniket\p20.js"
Append File
undefined
```



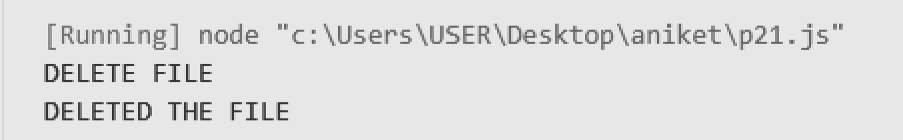
```
HELLO ANIKET
 HELLO WORLD

```

PRACTICAL-27

Aim: write node js program to delete a file

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

 fs.unlink("ani.js",function(err,data)

 {

    console.log("DELETE FILE");

    console.log("DELETED THE FILE");

 });
```

Output:



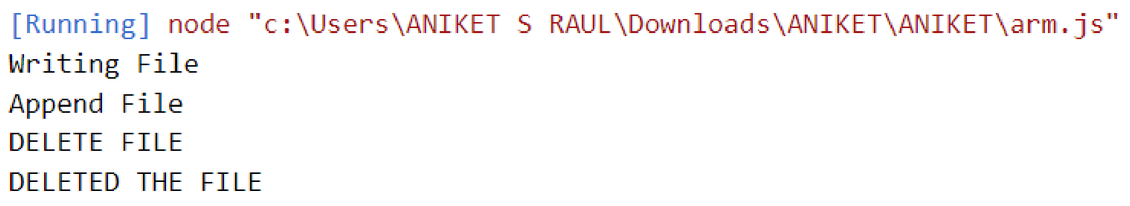
```
[Running] node "c:\Users\USER\Desktop\aniket\p21.js"
DELETE FILE
DELETED THE FILE
```

PRACTICAL-28

Aim: Combine Read, Write, Append, Delete file in one program in Node.js

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


fs.writeFile("ani.js", 'HELLO ANIKET', function (err, data) {

    console.log("Writing File");

    console.log(data);

});


fs.appendFile("ani.js", "\n HELLO WORLD\n", function (err, data) {
    console.log("Append File");

    console.log(data)

});


fs.unlink("ani.js", function (err, data) {

    console.log("DELETE FILE");

    console.log("DELETED THE FILE");

});
```

Output:



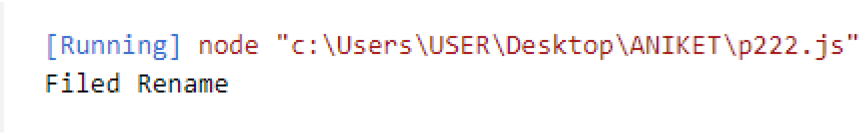
```
[Running] node "c:\Users\ANIKET S RAUL\Downloads\ANIKET\ANIKET\arm.js"
Writing File
Append File
DELETE FILE
DELETED THE FILE
```

PRACTICAL-29

Aim: write node js program to rename a file

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

fs.rename('aaaaa.txt','hello.txt',function(err){

    if(err) throw err;

    console.log('Filed Rename')

});
```

Output:



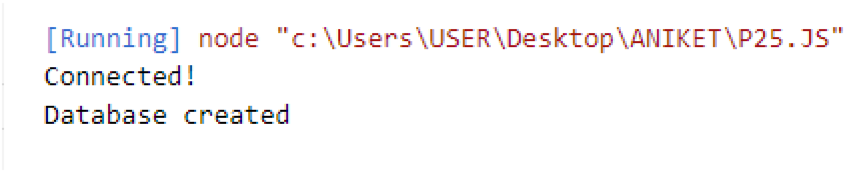
```
[Running] node "c:\Users\USER\Desktop\ANIKET\p222.js"
Filed Rename
```

PRACTICAL-30

Aim: Write a node js program to create a database

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

var con = mysql.createConnection({

  host: "localhost",

  user: "root",

  password: "12345",


});

con.connect(function(err) {

  if (err) {throw err;}

 console.log("Connected!");

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

    if (err) {throw err;}

    console.log("Database created");

  });

});
```

Output:



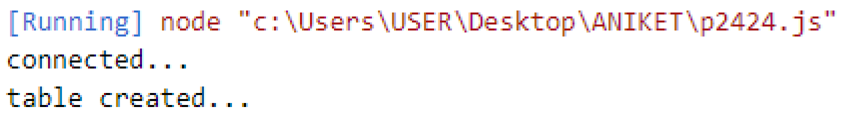
```
[Running] node "c:\Users\USER\Desktop\ANIKET\P25.JS"
Connected!
Database created
```

PRACTICAL-31

Aim: Write a node js program to create a table

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

var con=mysql.createConnection

(

    {

        host:'localhost',
        user:'root',

        password:'12345',

        database:'aniani'

    }

);

con.connect(function(err)

{if(err) throw err;

    console.log("connected...");

    var sql = "CREATE TABLE student1(id INT(10) PRIMARY KEY AUTO_INCREMENT,name
VARCHAR(255), address VARCHAR(255),course VARCHAR(20), contact INT(15))";

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

    {

        if(err) throw err;

        console.log("table created...");

    });

});
```

Output:



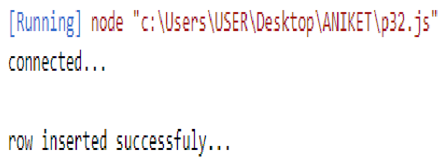
```
[Running] node "c:\Users\USER\Desktop\ANIKET\p2424.js"
connected...
table created...
```

PRACTICAL-32

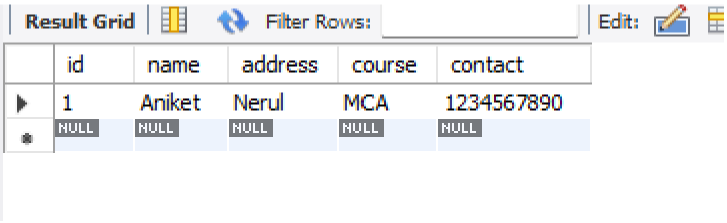
Aim: Write a node js program to insert value in a table

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

var con=mysql.createConnection

({

        host:'localhost',

        user:'root',
        password:'12345',

        database:'aniani'

    }

);

con.connect(function(err)

{ if(err) throw err;

    console.log("connected...");

    var sql1 = "INSERT INTO student1(id ,name , address,course , contact)
VALUES('1','Aniket','Nerul','MCA','1234567890')";

//var sql1="select * from student1";

    con.query(sql1,function(err,result)

    {

        if(err) throw err;

        console.log()

        console.log("row inserted successfuly...");

    });

});
```

Output:



```
[Running] node "c:\Users\USER\Desktop\ANIKET\p32.js"
connected...


row inserted successfuly...
```



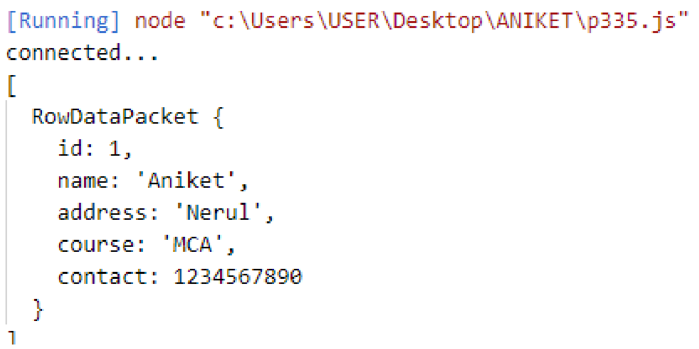
	id	name	address	course	contact
▶	1	Aniket	Nerul	MCA	1234567890
*	NULL	NULL	NULL	NULL	NULL

PRACTICAL-33

Aim: Write a node js program to select values from table

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

var con=mysql.createConnection

(

    {

        host:'localhost',

        user:'root',
        password:'12345',
        database:'aniani'
    }
);
con.connect(function(err)
{
    if(err) throw err;
    console.log("connected...");
    var sql2="select * from student1";
    con.query(sql2,function(err,result)
    {
        if(err) throw err;
        console.log(result);
    });
});
```

Output:



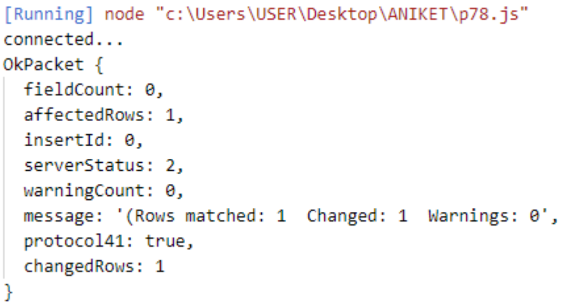
```
[Running] node "c:\Users\USER\Desktop\ANIKET\p335.js"
connected...
[
  RowDataPacket {
    id: 1,
    name: 'Aniket',
    address: 'Nerul',
    course: 'MCA',
    contact: 1234567890
  }
]
```

PRACTICAL-34

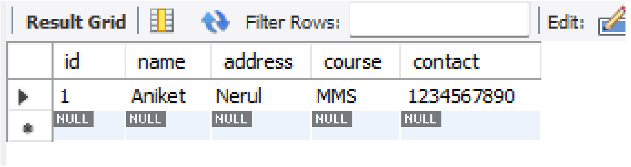
Aim: Write a node js program to update values from table

```
var mysql=require('mysql');
var con=mysql.createConnection
(
    {
        host:'localhost',
        user:'root',
        password:'12345',

        database:'aniani'

    }

);

con.connect(function(err)

{

    if(err) throw err;

    console.log("connected...");

    var sql2= "UPDATE student1 SET course ='MMS' WHERE ID='1'";

    con.query(sql2,function(err,result)

    {

        if(err) throw err;

        console.log(result);

    });

});
```

Output:



```
[Running] node "c:\Users\USER\Desktop\ANIKET\p78.js"
connected...
OkPacket {
  fieldCount: 0,
  affectedRows: 1,
  insertId: 0,
  serverStatus: 2,
  warningCount: 0,
  message: '(Rows matched: 1  Changed: 1  Warnings: 0',
  protocol41: true,
  changedRows: 1
}
```



	id	name	address	course	contact
▶	1	Aniket	Nerul	MMS	1234567890
✱	NULL	NULL	NULL	NULL	NULL

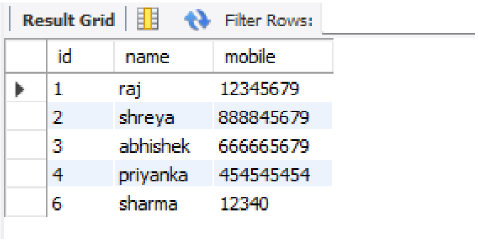
PRACTICAL-35

Aim: 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

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

var con = mysql.createConnection

({

 host: "localhost",

user: "root", password: "12345", database:"aniani"

});
```

```
con.connect(function(err) { if (err) throw err;

console.log("Connected successfully to server");

var sql = "SELECT * FROM participants WHERE name = 'Sharma'"; con.query(sql, function(err, result) {

if (err) throw err; console.log("Participant found: ", result);



var newMobileNumber = '1234567890';

var updateSql = `UPDATE participants SET mobile = '${newMobileNumber}' WHERE name = 'sharma'`;



con.query(updateSql, function(err, result) { if (err) throw err;

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

});

});

});
```

Output:



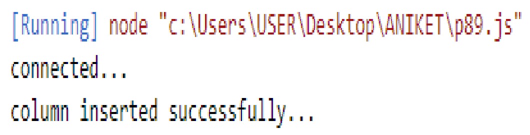
id	name	mobile
1	raj	12345679
2	shreya	888845679
3	abhishek	666665679
4	priyanka	454545454
6	sharma	12340

PRACTICAL-36

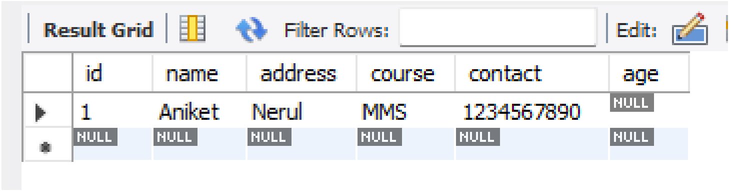
Aim: Write a node js program to add new column in table

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

var con=mysql.createConnection

(

    {

        host:'localhost',

        user:'root',
        password:'12345',

        database:'aniani'

    }

);

con.connect(function(err)

{

    if(err) throw err;

    console.log("connected...");

    var sql = "ALTER TABLE student1 ADD age INT(5)";

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

    {

        if(err) throw err;

        console.log("column inserted successfully...");

    });

});
```

Output:



```
[Running] node "c:\Users\USER\Desktop\ANIKET\p89.js"
connected...
column inserted successfully...
```



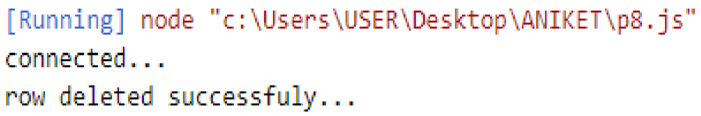
	id	name	address	course	contact	age
▶	1	Aniket	Nerul	MMS	1234567890	NULL
✱	NULL	NULL	NULL	NULL	NULL	NULL

PRACTICAL-37

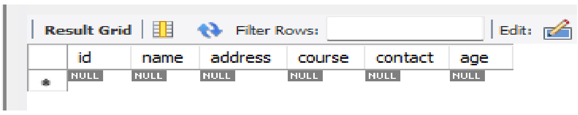
Aim: Write a node js program to delete the row in table

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

var con=mysql.createConnection

(

    {

        host:'localhost',

        user:'root',

        password:'12345',
        database:'aniani'

    }

);

con.connect(function(err)

{

    if(err) throw err;

    console.log("connected...");

    var sql= "DELETE FROM student1 WHERE ID='1'";

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

    {

        if(err) throw err;

        console.log("row deleted successfuly...");

    });

});
```

Output:



```
[Running] node "c:\Users\USER\Desktop\ANIKET\p8.js"
connected...
row deleted successfuly...
```



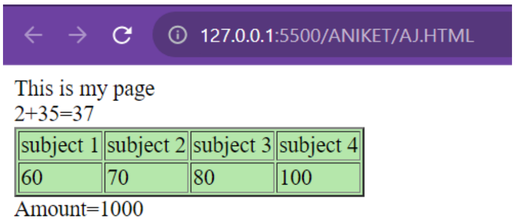
	id	name	address	course	contact	age
*	NULL	NULL	NULL	NULL	NULL	NULL

PRACTICAL-1

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

```
<!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="">
        This is my page
        <div ng-init="marks=[60,70,80,100]">
                2+35={{2+35}}
                <table border="2" style=" background-color:rgb(181, 231, 171)">
                        <tr>
                                <td>subject 1 </td>
                                <td>subject 2 </td>
                                <td>subject 3 </td>
                                <td>subject 4 </td>
                            </tr>
                        <tr>
                                <td>{{marks[0]}}</td>
                                <td>{{marks[1]}}</td>
                                <td>{{marks[2]}}</td>
                                <td>{{marks[3]}}</td>

                            </tr>
                    </table>
            </div>
        <div ng-init="people=5 ; reg=200">
                Amount={{people * reg}}
    </div>
       </body>
</html>
```

Output:



This is my page
2+35=37

subject 1	subject 2	subject 3	subject 4
60	70	80	100

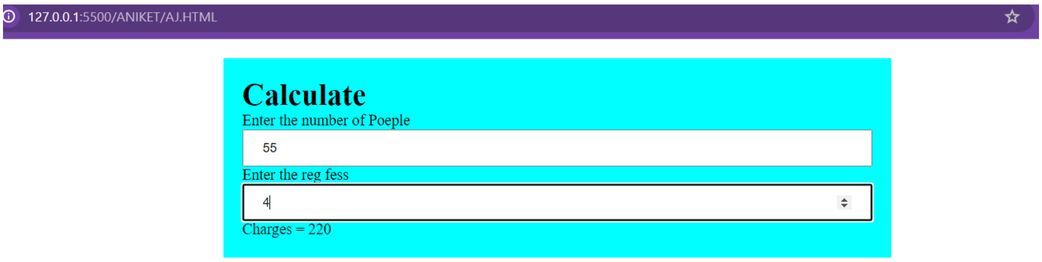
Amount=1000

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.

```
<!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;

                    }

 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 :



Calculate

Enter the number of Poeple

55

Enter the reg fess

4

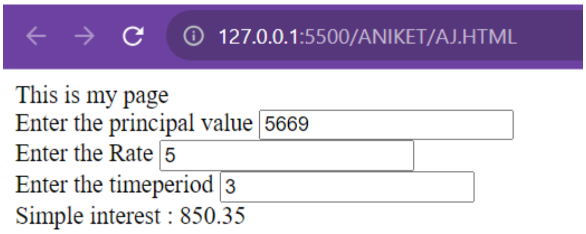
Charges = 220

PRACTICAL-3

Aim: **Create an application in angular.js to calculate simple interest take appropriate input from the user**

```
<!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="">

    This is my page

<div>

        Enter the principal value <input type="number" ng-model="prin"> <br>

        Enter the Rate <input type="number" ng-model="rate"> <br>

        Enter the timeperiod <input type="number" ng-model="time"> <br>

          Simple interest : {{(prin*rate*time)/100}}

    </div>

</body>


</html>
```

Output :


127.0.0.1:5500/ANIKET/AJ.HTML

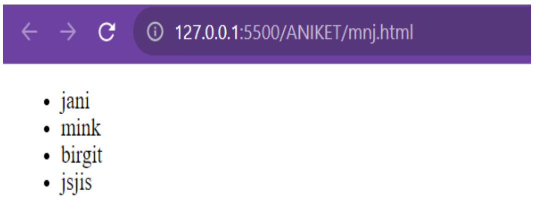
This is my page
Enter the principal value 5669
Enter the Rate 5
Enter the timeperiod 3
Simple interest : 850.35

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**

```
<!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>
   <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=["jani","carl","mink","hedge","joe","gustav","birgit","mary","kay","jsjis"];


    });

    </script>

 </body>

 </html>
```

Output :



127.0.0.1:5500/ANIKET/mnj.html

- jani
- mink
- birgit
- jsjis

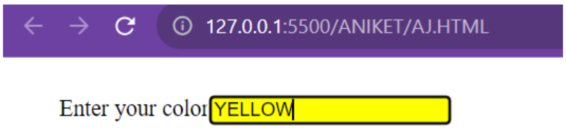
PRACTICAL-5

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

```
<!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>

 <div ng-app="">

        Enter your color<input type="text" ng-model="T" style="background-color:{{T}}"><br>

    </div>
```

```
</body>

</body>

</html>
```

Output :



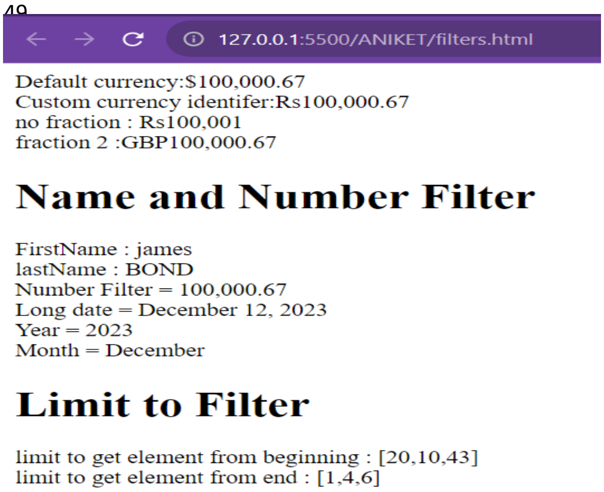
Enter your color YELLOW

PRACTICAL-6

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

```
<!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="myApp" ng-controller="myController">

    <div>

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

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

        no fraction : {{person.salary|currency: 'Rs':0}} <br>
        fraction 2 :<span ng-bind="person.salary|currency:'GBP':2"></span>
    </div>
    <div>
        <h1>Name and Number Filter</h1>
        FirstName : {{person.firstname|lowercase}}<br>
        lastName : {{person.lastname|uppercase}} <br>
        Number Filter = {{person.salary|number:2}} <br>
        Long date = {{DOB|date:'longDate'}} <br>
        Year = {{DOB|date:"yyyy"}} <br>
        Month = {{DOB|date:"MMMM"}} <br>
    </div>
    <div>
        <h1>Limit to Filter</h1>
        limit to get element from beginning : {{limitarr| limitTo :3}} <br>
        limit to get element from end : {{limitarr| limitTo :-3}}

    </div>
    <script>
        var myApp = angular.module('myApp', []);
        myApp.controller("myController", function ($scope) {
            $scope.person = { firstname: 'James', lastname: 'Bond ', salary: 100000.67 }
            $scope.DOB = new Date();
            $scope.limitarr = [20, 10, 43, 5, 1, 4, 6]

        });
    </script>
</body>
</html>
```

Output :



```
Default currency:$100,000.67
Custom currency identifer:Rs100,000.67
no fraction : Rs100,001
fraction 2 :GBP100,000.67
```

Name and Number Filter

```
FirstName : james
lastName : BOND
Number Filter = 100,000.67
Long date = December 12, 2023
Year = 2023
Month = December
```

Limit to Filter

```
limit to get element from beginning : [20,10,43]
limit to get element from end : [1,4,6]
```

PRACTICAL-7

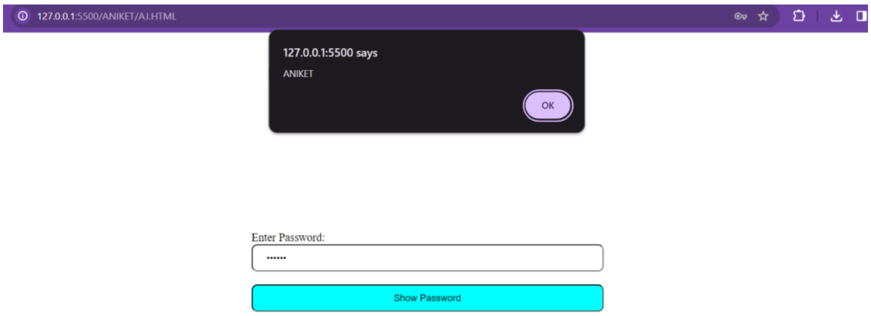
AIM: Create an application in angular.js to demonstrate to display text in alert box

```
<!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 :



127.0.0.1:5500/ANIKET/AJ.HTML

127.0.0.1:5500 says

ANIKET

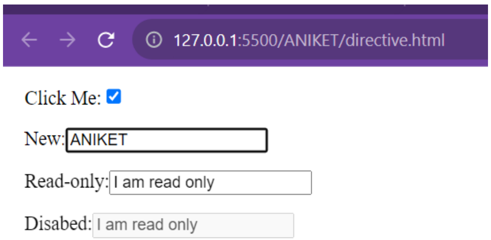
Enter Password:

PRACTICAL-8

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

```
<!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: 20px;

            display: block;

            margin: 15px 0 0 10px;

        }

    </style>

</head>



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

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

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

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

    <div>Disabed:<input ng-disabled="checked" type="text" value="I am read only " /></div>

</body>



</html>
```

Output :



Click Me: ☑

New: ANIKET

Read-only: I am read only

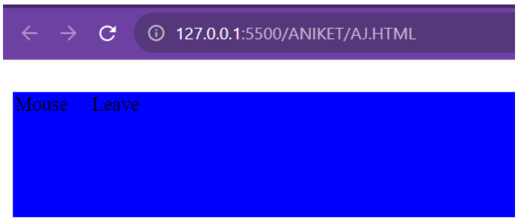
Disabed: I am read only

PRACTICAL-9

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

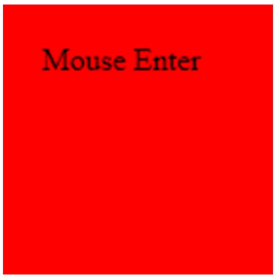
```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Document</title>
     <script src="angular.min.js"></script>
        <style>
        .redDiv {
            width: 100 px;
            height: 100px;
            background-color: red;
            padding: 2px 2px 2px 2px;
        }
         .blueDiv {
            width: 100 px;
            height: 100px;
            background-color: blue;
            padding: 2px 2px 2px 2px;
        }
    </style>
</head>
<body ng-app>
        <div ng-class="{redDiv:enter,blueDiv: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 :



127.0.0.1:5500/ANIKET/AJ.HTML

Mouse Leave



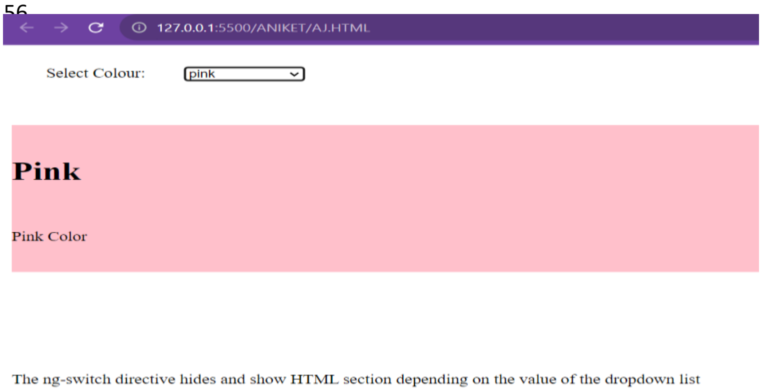
Mouse Enter

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

```
<!DOCTYPE html>

<html lang="en"><head>

    <meta charset="UTF-8">

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

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <title>Document</title>

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

</head>


<body ng-app="">

    <div>

        <form>

            Select Colour:

            <select ng-model="myVar">

                <option value="pink">pink

  <option value="ablu">Sky Blue

<option value="lav">lavender

 </select>

        </form>

    </div>
    <div ng-switch="myVar">

        <div ng-switch-when="">

        </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="ablu" style="background-color: skyblue;">

                <h1>Sky Blue</h1>

                <p>Sky blue color</p>

            </div>

        </div>

        <div ng-switch="myVar">

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

                <h1>Lavender</h1>

                <p>lavender color</p>

            </div>

        </div>

        <p>The ng-switch directive hides and show HTML section depending on the value of the dropdown list</p>

</body>

</html>
```

Output :



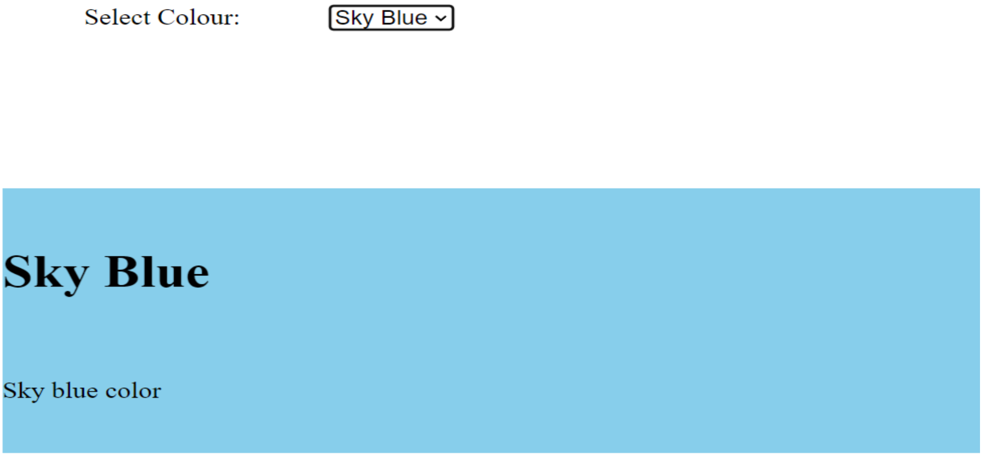
127.0.0.1:5500/ANIKET/AJ.HTML

Select Colour: pink

Pink

Pink Color

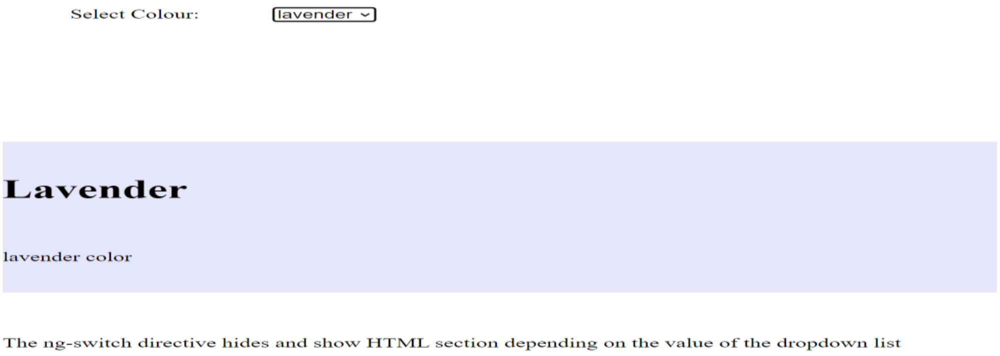
The ng-switch directive hides and show HTML section depending on the value of the dropdown list



Select Colour: Sky Blue

Sky Blue

Sky blue color



Select Colour: lavender

Lavender

lavender color

The ng-switch directive hides and show HTML section depending on the value of the dropdown list

PRACTICAL-11

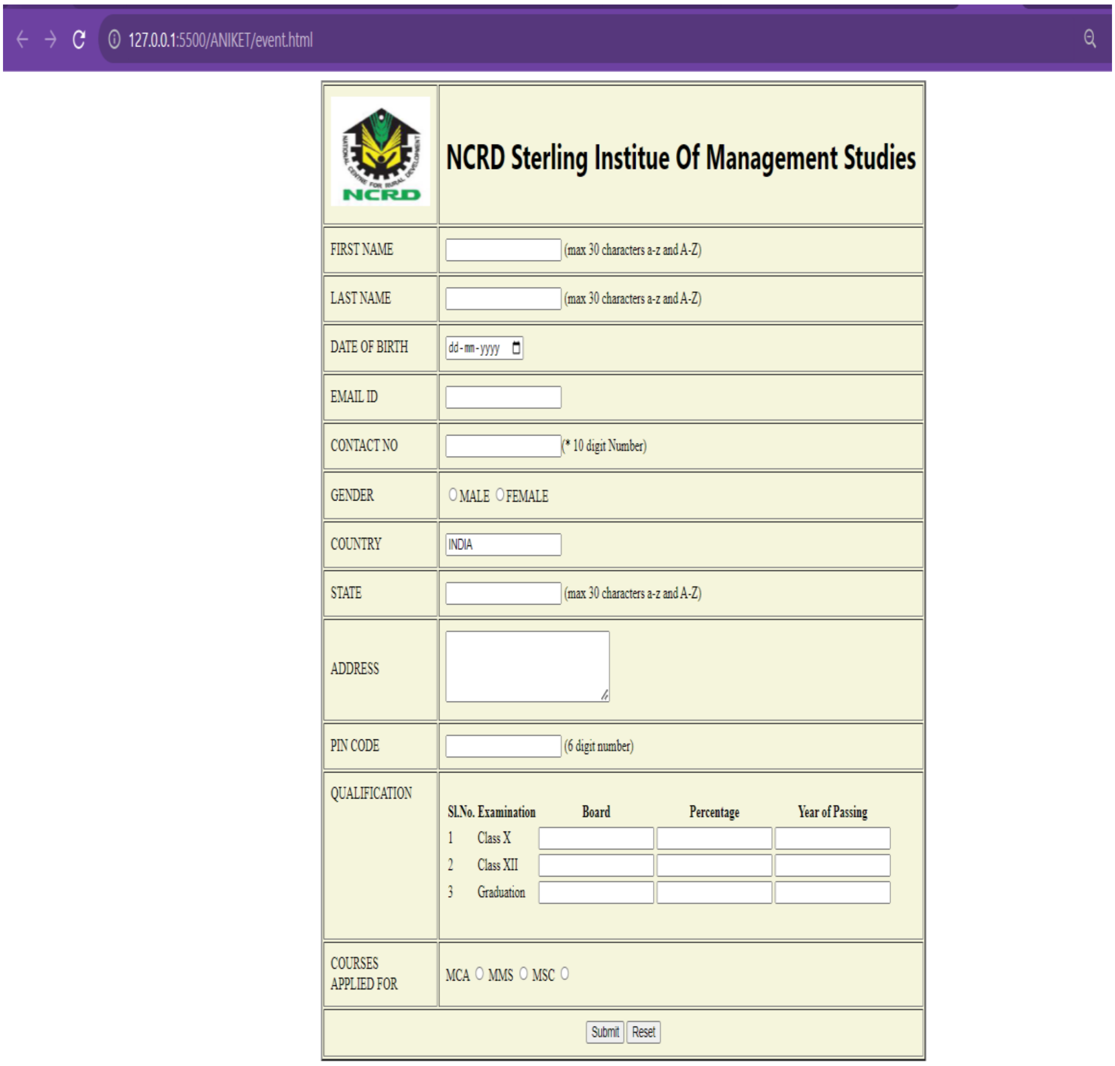
Aim:-**Write an Angular JS code to display a Registration form for Student applying for a new Course. Display all the values entered by the students.**

```
<!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-controller="stdctrl">
    <form ng-submit="ngsubmitform()">
        <table align="center" cellpadding="10" border="2" style="background-color:beige">
            <tr>
                <td><img src="aaa.jpg" width="150px;" height="100px"></td>
                <td align="center">
                    <h1 style="font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;"> NCRD Sterling Institue Of
                        Management Studies</h1>
                </td>
            </tr>


            <!----- First Name ---------------------------------------------------------->
            <tr>
                <td>FIRST NAME</td>
                <td><input type="text" name="First_Name" maxlength="30" ng-model="student.fname" />
                    (max 30 characters a-z and A-Z)
                </td>
            </tr>


            <!----- Last Name ----------------------------------------------------------->
            <tr>
                <td>LAST NAME</td>
                <td><input type="text" name="Last_Name" maxlength="30" ng-model="student.lname" />
                    (max 30 characters a-z and A-Z)
                </td>
            </tr>


            <tr>
                <td>DATE OF BIRTH</td>
                <td><input type="date" NAME="" ng-model="student.dt"></td>
            </tr>


            <tr>
                <td>EMAIL ID</td>
                <td><input type="email " ng-model="student.mail"></td>
            </tr>


            <tr>
                <td>CONTACT NO</td>
                <td><input type="text" maxlength="10" ng-model="student.nu">(* 10 digit Number)</td>
            </tr>


            <tr>
                <td>GENDER</td>
                <td><input type="radio" name="gen" ng-model="student.gen">MALE
                    <input type="radio" name="gen" ng-model="student.gen">FEMALE
                </td>
            </tr>
            <tr>
                <td>COUNTRY</td>
                <td><input type="text" name="Country" value="INDIA" readonly="readonly"
ng-model="student.coun" /></td>
            </tr>
            <tr>
                <td>STATE</td>
                <td><input type="text" name="State" maxlength="30" ng-model="student.sta"
/>
                    (max 30 characters a-z and A-Z)
                </td>
            </tr>
            <tr>
                <td>ADDRESS</td>
                <td><textarea rows="4" cols="30" ng-model="student.add"></textarea></td>
            </tr>

            <tr>
                <td>PIN CODE</td>
                <td><input type="text" name="Pin_Code" maxlength="6" ng-
model="student.pin" />
                    (6 digit number)
                </td>
            </tr>
            <tr>
                <td>QUALIFICATION <br /><br /><br /><br /><br /><br /><br /></td>

                <td>
                    <table>

                        <tr>
```

```
                            <td align="center"><b>Sl.No.</b></td>
                            <td align="center"><b>Examination</b></td>
                            <td align="center"><b>Board</b></td>
                            <td align="center"><b>Percentage</b></td>
                            <td align="center"><b>Year of Passing</b></td>
                        </tr>

                        <tr>
                            <td>1</td>
                            <td>Class X</td>
                            <td><input type="text" name="ClassX_Board" maxlength="30" ng-model="student.xbo" /></td>
                            <td><input type="text" name="ClassX_Percentage" maxlength="30" ng-model="student.xper" />
                            </td>
                            <td><input type="text" name="ClassX_YrOfPassing" maxlength="30" ng-model="student.xpa" />
                            </td>
                        </tr>

                        <tr>
                            <td>2</td>
                            <td>Class XII</td>
                            <td><input type="text" name="ClassXII_Board" maxlength="30" ng-model="student.x2bo" /></td>
                            <td><input type="text" name="ClassXII_Percentage" maxlength="30" ng-model="student.x2per" />
                            </td>
                            <td><input type="text" name="ClassXII_YrOfPassing" maxlength="30" ng-model="student.x2pa" />
                            </td>
                        </tr>

                        <tr>
```

```
                            <td>3</td>
                            <td>Graduation</td>
                            <td><input type="text" name="Graduation_Board" maxlength="30"
ng-model="student.grbo" />
                            </td>
                            <td><input type="text" name="Graduation_Percentage"
maxlength="30"
                                    ng-model="student.grper" /></td>
                            <td><input type="text" name="Graduation_YrOfPassing"
maxlength="30"
                                    ng-model="student.grpa" /></td>
                        </tr>


                        <tr>
                            <td></td>
                            <td></td>
                            <td align="center"></td>
                            <td align="center"></td>
                        </tr>
                    </table>

                </td>
            </tr>

            <!----- Course ----------------------------------------------------------->
            <tr>
                <td>COURSES<br />APPLIED FOR</td>
                <td>
                    MCA
                    <input type="radio" name="Course" value="BCA" ng-model="student.ccc">
                    MMS
                    <input type="radio" name="Course" value="B.Com" ng-model="student.ccc">
                    MSC
                    <input type="radio" name="Course" value="B.Sc" ng-model="student.ccc">

                </td>
            </tr>
            <tr>
                <td colspan="2" align="center">
                    <input type="submit" value="Submit" onclick="submitForm()">
                    <input type="reset" value="Reset" ng-click="resetForm()" value="Reset">
                </td>
            </tr>
        </table>
    </form>

    </form>
    <script>
        function submitForm() {
            alert("Form submitted ");
        }

        //1. create app module
        var studentApp = angular.module('studentApp', []);
        //2. create controller
        studentApp.controller("studentController", function ($scope, $http) {
            //3. attach originalStudent model object
            $scope.originalStudent = {
                firstName: 'James',
                lastName: 'Bond',
                DoB: new Date('01/31/1980'),
                gender: 'male',

                trainingType: 'online',

                MCA: false,

                MBA: true,

                Pharmacy: true,

                Email: 'rohitmahajan3@gmail.com'

            };

            //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.submitStudentForm = 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);

            };

        });

    </script>

</body>


</html>
```

Output :



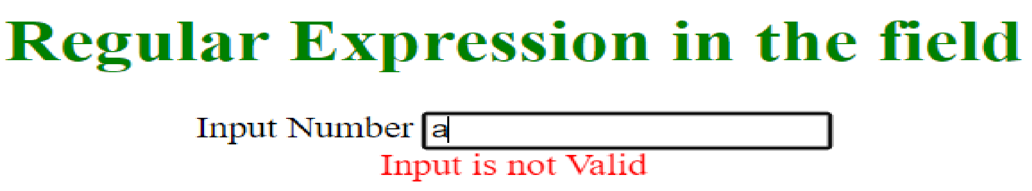
	NCRD Sterling Institue Of Management Studies
FIRST NAME	(max 30 characters a-z and A-Z)
LAST NAME	(max 30 characters a-z and A-Z)
DATE OF BIRTH	dd-mm-yyyy
EMAIL ID	
CONTACT NO	(* 10 digit Number)
GENDER	○MALE ○FEMALE
COUNTRY	INDIA
STATE	(max 30 characters a-z and A-Z)
ADDRESS	
PIN CODE	(6 digit number)
QUALIFICATION	Sl.No. Examination Board Percentage Year of Passing 1 Class X 2 Class XII 3 Graduation
COURSES APPLIED FOR	MCA ○ MMS ○ MSC ○

PRACTICAL-12

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

```
<!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="app" style="text-align:center">
    <h1 style="color: green;">Regular Expression in the 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.$error.pattern" style="color: red">
                Input is not Valid
            </span>
        </ng-form>
    </div>
    <script>
        var app = angular.module("app", []);
        app.controller('reg', ['$scope', function ($scope) {
            $scope.re = /^[0-9]{1,6}$/;
        }]);
    </script>
</body>
</html>
```

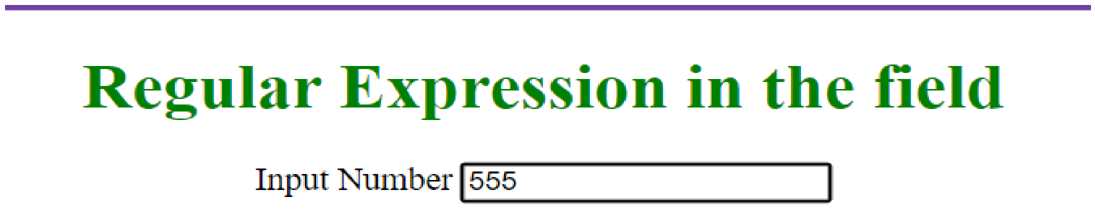
Output :



Regular Expression in the field

Input Number a

Input is not Valid



Regular Expression in the field

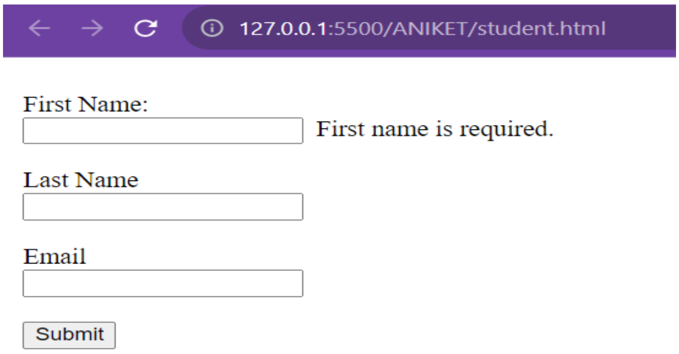
Input Number 555

PRACTICAL-13

Aim: Angular program to demonstrate validations directives

```
<!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 :



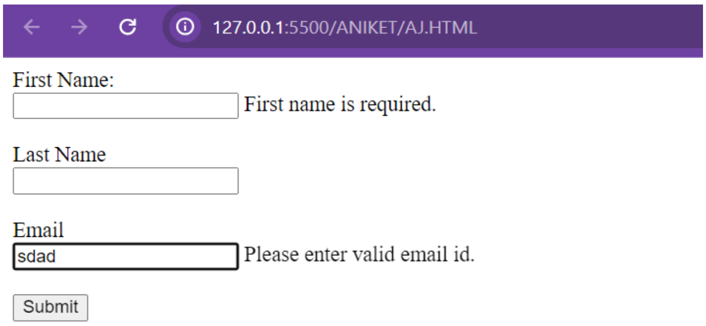
127.0.0.1:5500/ANIKET/student.html

First Name:

First name is required.

Last Name

Email



127.0.0.1:5500/ANIKET/AJ.HTML

First Name:

First name is required.

Last Name

Email

sdad

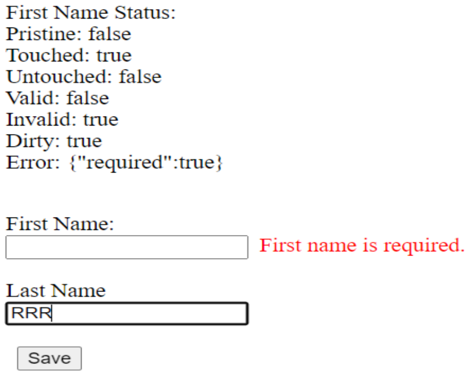
Please enter valid email id.

PRACTICAL-14

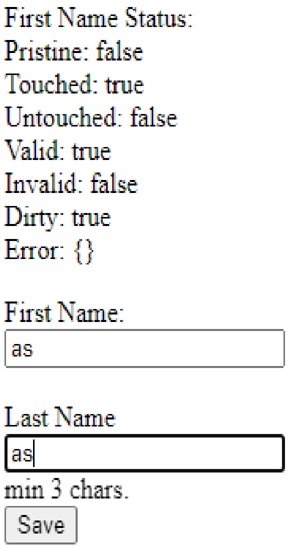
Aim: Angular program to demonstrate the state properties of form fields

```
<!DOCTYPE html>
<html>

<head>

    <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"
            style="color: red;">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" style="color: red;">min 3
chars.</span>  <span

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



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

         </form>

</body>



</html>
```

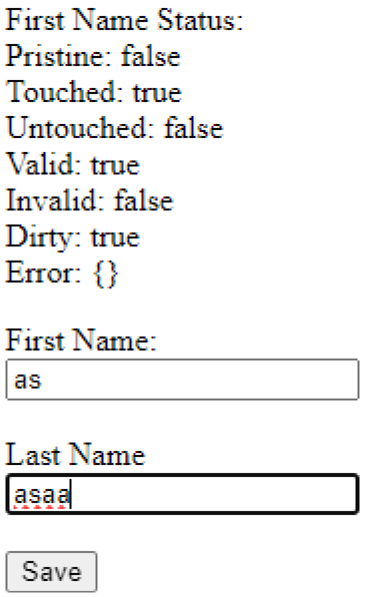
Output :



```
First Name Status:
Pristine: false
Touched: true
Untouched: false
Valid: false
Invalid: true
Dirty: true
Error: {"required":true}

First Name:
[          ] First name is required.

Last Name
[RRR       ]

Save
```



```
First Name Status:
Pristine: false
Touched: true
Untouched: false
Valid: true
Invalid: false
Dirty: true
Error: {}

First Name:
[as        ]

Last Name
[as        ]
min 3 chars.
Save
```



```
First Name Status:
Pristine: false
Touched: true
Untouched: false
Valid: true
Invalid: false
Dirty: true
Error: {}

First Name:
[as        ]

Last Name
[asaa      ]

Save
```

PRACTICAL-15

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

```
   <!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", {
               templateUrl: 'regiss.html'
            })
            .when("/login", {
               templateUrl: 'loginn.html'
            })
      });
   </script>
</body>
</html>
```

Output :

Angular Routing Demo

Registration Login

PRACTICAL-16

*Aim:*Create an application with Login page and Registration Page using Single Page Application(SPA)

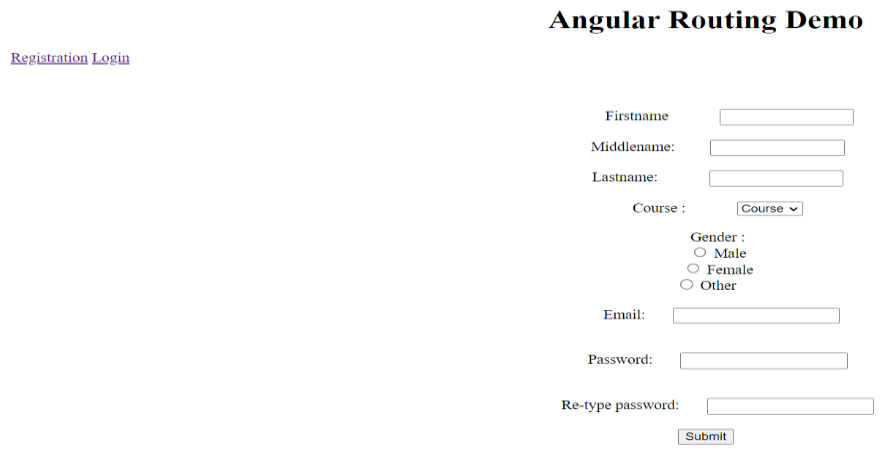
regiss.html

```
<Html>
     <head>
     <title>
     Registration Page
     </title>
     </head>
     <body bgcolor="Lightskyblue">
     <br>
     <br>
     <form>
```

```
<label> Firstname </label>
<input type="text" name="firstname" size="15"/> <br> <br>
<label> Middlename: </label>
<input type="text" name="middlename" size="15"/> <br> <br>
<label> Lastname: </label>
<input type="text" name="lastname" size="15"/> <br> <br>
<label>
Course :
</label>
<select>
<option value="Course">Course</option>
<option value="BCA">BCA</option>
<option value="BBA">BBA</option>
<option value="B.Tech">B.Tech</option>
<option value="MBA">MBA</option>
<option value="MCA">MCA</option>
<option value="M.Tech">M.Tech</option>
</select>
<br>
<br>
<label>
Gender :
</label><br>
<input type="radio" name="male"/> Male <br>
<input type="radio" name="female"/> Female <br>
<input type="radio" name="other"/> Other
<br>
<br>
```

```
     Email:

     <input type="email" id="email" name="email"/> <br>

     <br> <br>

     Password:

     <input type="Password" id="pass" name="pass"> <br>

     <br> <br>

     Re-type password:

     <input type="Password" id="repass" name="repass"> <br> <br>

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

     </form>

     </body>

     </html>
```

Output :

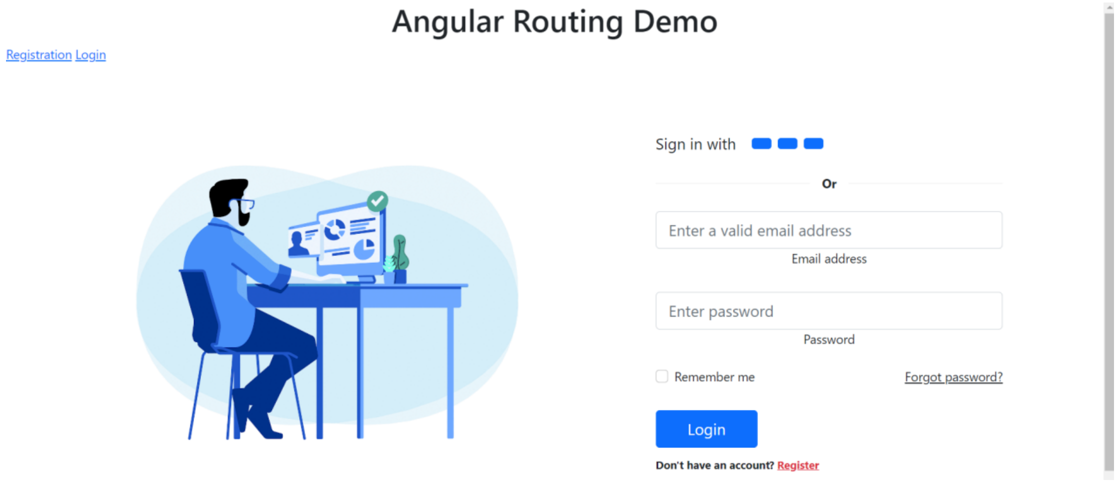


loginn.html

```
     <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

    <title>Document</title>

    <style>

.divider:after,

.divider:before {

content: "";

flex: 1;

height: 1px;

background: #eee;

}

.h-custom {

height: calc(100% - 73px);

}

@media (max-width: 450px) {

.h-custom {

height: 100%;

}

}

    </style>

</head>

<body>

    <section class="vh-100">

        <div class="container-fluid h-custom">

          <div class="row d-flex justify-content-center align-items-center h-100">

            <div class="col-md-9 col-lg-6 col-xl-5">

              <img src="b.gif"

                class="img-fluid" alt="Sample image">

            </div>

            <div class="col-md-8 col-lg-6 col-xl-4 offset-xl-1">

              <form>
        <div class="d-flex flex-row align-items-center justify-content-center
justify-content-lg-start">
         <p class="lead fw-normal mb-0 me-3">Sign in with</p>
         <button type="button" class="btn btn-primary btn-floating mx-1">
          <i class="fab fa-facebook-f"></i>
         </button>

         <button type="button" class="btn btn-primary btn-floating mx-1">
          <i class="fab fa-twitter"></i>
         </button>

         <button type="button" class="btn btn-primary btn-floating mx-1">
          <i class="fab fa-linkedin-in"></i>
         </button>
        </div>

        <div class="divider d-flex align-items-center my-4">
         <p class="text-center fw-bold mx-3 mb-0">Or</p>
        </div>

        <!-- Email input -->
        <div class="form-outline mb-4">
         <input type="email" id="form3Example3" class="form-control form-control-
lg"
          placeholder="Enter a valid email address" />
         <label class="form-label" for="form3Example3">Email address</label>
        </div>
  <!-- Password input -->
        <div class="form-outline mb-3">
         <input type="password" id="form3Example4" class="form-control form-
control-lg"
          placeholder="Enter password" />
         <label class="form-label" for="form3Example4">Password</label>
        </div>


        <div class="d-flex justify-content-between align-items-center">
         <!-- Checkbox -->
         <div class="form-check mb-0">
          <input class="form-check-input me-2" type="checkbox" value=""
id="form2Example3" />
          <label class="form-check-label" for="form2Example3">
           Remember me
          </label>
         </div>
         <a href="#!" class="text-body">Forgot password?</a>
        </div>


        <div class="text-center text-lg-start mt-4 pt-2">
         <button type="button" class="btn btn-primary btn-lg"
          style="padding-left: 2.5rem; padding-right: 2.5rem;">Login</button>
         <p class="small fw-bold mt-2 pt-1 mb-0">Don't have an account? <a
href="#!"
           class="link-danger">Register</a></p>
        </div>
   </form>
      </div>
     </div>
    </div>
    <div
     class="d-flex flex-column flex-md-row text-center text-md-start justify-content-
between py-4 px-4 px-xl-5 bg-primary">
    <!-- Right -->
     <div>
      <a href="#!" class="text-white me-4">
       <i class="fab fa-facebook-f"></i>
      </a>
```

```
            <a href="#!" class="text-white me-4">
              <i class="fab fa-twitter"></i>
            </a>
            <a href="#!" class="text-white me-4">
              <i class="fab fa-google"></i>
            </a>
            <a href="#!" class="text-white">
              <i class="fab fa-linkedin-in"></i>
            </a>
          </div>
          <!-- Right -->
        </div>
      </section>
</body>
</html>
```

Output :



Angular Routing Demo

Don't have an account? Register