**Practical No.1**

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

**Exercise –**

Create node js application and use built in module as well as create own modules

Implementation:

**Program:**

1)myfirstmodule.js

exports.myDateTime = function () {

    return Date();

  };

2)demo\_module.js

var http = require('http');

var dt = require('./myfirstmodule');

http.createServer(function (req, res) {

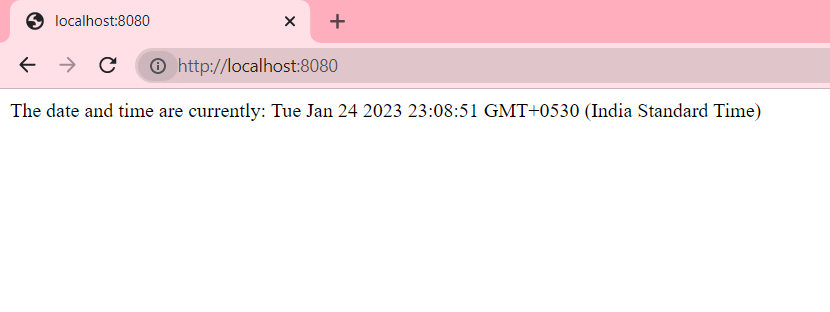
  res.writeHead(200, {'Content-Type': 'text/html'});

  res.write("The date and time are currently: " + dt.myDateTime());

  res.end();

}).listen(8080);

**Output:**



**Practical No.2**

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

**Exercise –**

Create an application to demonstrate various Node.js Events

Implementation:

**Program:**

1)demo\_file.js

var fs = require('fs');

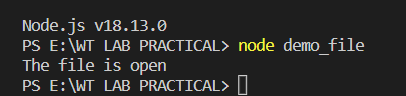
var rs = fs.createReadStream('./demofile.txt');

rs.on('open', function () {

  console.log('The file is open');

});

**Output:**



2)event\_emitter.js

var events = require('events');

var eventEmitter = new events.EventEmitter();

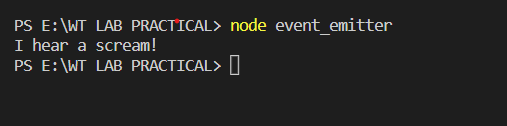
var myEventHandler = function () {

  console.log('I hear a scream!');

}

eventEmitter.on('scream', myEventHandler);

eventEmitter.emit('scream');

**Output:**

**Practical No. 3**

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

**Exercise –**

Create an application to demonstrate Node.js Functions

Implementation:

**Program:**

1)demofile.txt

REPUBLIC DAY

26 January was chosen as the date for Republic Day as it was on that day in 1930 when the Declaration of

Indian Independence was proclaimed by the Indian National Congress.

The original text of the Preamble to the Constitution of India.

The Constitution of India came into force on 26 January 1950.

2)practical3.js

var http =require('http');

var fs=require('fs');

http.createServer(function(req,res)

{

    var c="";

    var reader1=fs.createReadStream('demofile.txt');

    reader1.setEncoding('UTF-8');

    reader1.on('error',function(err)

    {

        console.log(err);

    }

    ).on('data',function(chunk)

    {

        c +=chunk;

    }).on('end',function()

    {

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

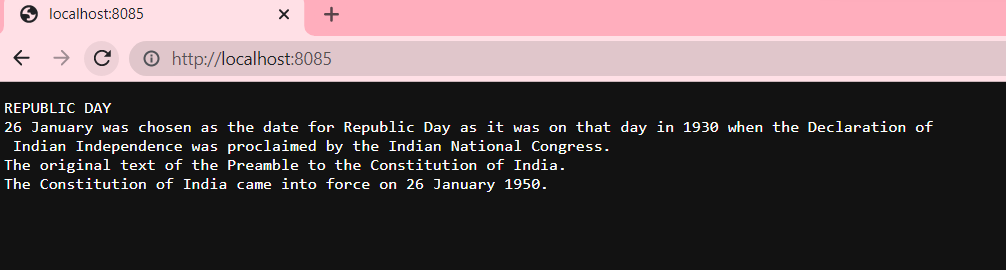
        res.write(c);

        res.end();

    })

}).listen(8085);

**Output:**



**Practical No. 4**

**Title: Using File Handling demonstrate all basic file operations**

**(Create, write, read, delete)**

**Exercise –**

Create an application to demonstrate file operations (Create, write, read,

delete)

Implementation:

**Program:**

practical4.js

var http=require('http');

var fs=require('fs');

http.createServer(function(req,res)

{

    var txt1="<h2>Hello this is file handling practical</h2>";

    fs.writeFile('index.html',txt1,function(err)

    {

        if(err)throw err;

        console.log('File Saved');

    });

    fs.open('index.html','a',function(err,fd){

        fs.appendFile(fd,'This is appended text',function(err)

        {

            if(err)throw err;

            fs.close(fd,function(err)

            {

                if(err) throw err;

            });

        });

    })

    fs.readFile('index.html',function(err,data)

    {

        if(err) throw err;

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

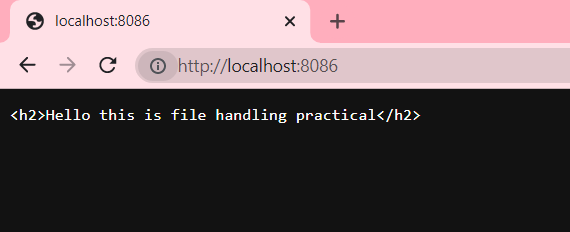
        res.write(data);

        res.end();

    })

}).listen(8086);

**Output:**



**Practical No. 5**

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

**Exercise –**

Create an HTTP Server and perform operations on it

Implementation:

**Program:**

httpWbServer.js

var http = require('http');

http.createServer(function (req, res) {

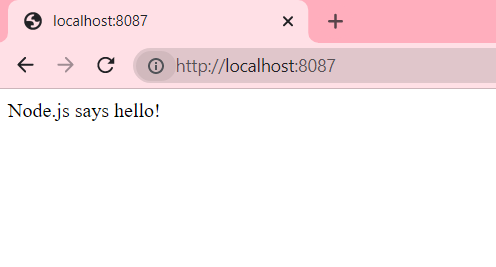
    res.writeHead(200, {'Content-Type': 'text/html'});

    res.write('Node.js says hello!');

    res.end();

}).listen(8087);

**Output:**



**Practical No.6**

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

**Exercise –**

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

**Implementation:**

**Program:**

practical6.js

var mysql=require('mysql');

var con=mysql.createConnection({

    host:"localhost",

    user:"root",

    password:"",

    database:"student\_info"

});

con.connect(function(error)

{

    if(error) throw error;

    con.query("select\*from studentInfo",function(error,result)

    {

        if(error) throw error;

        console.log(result);

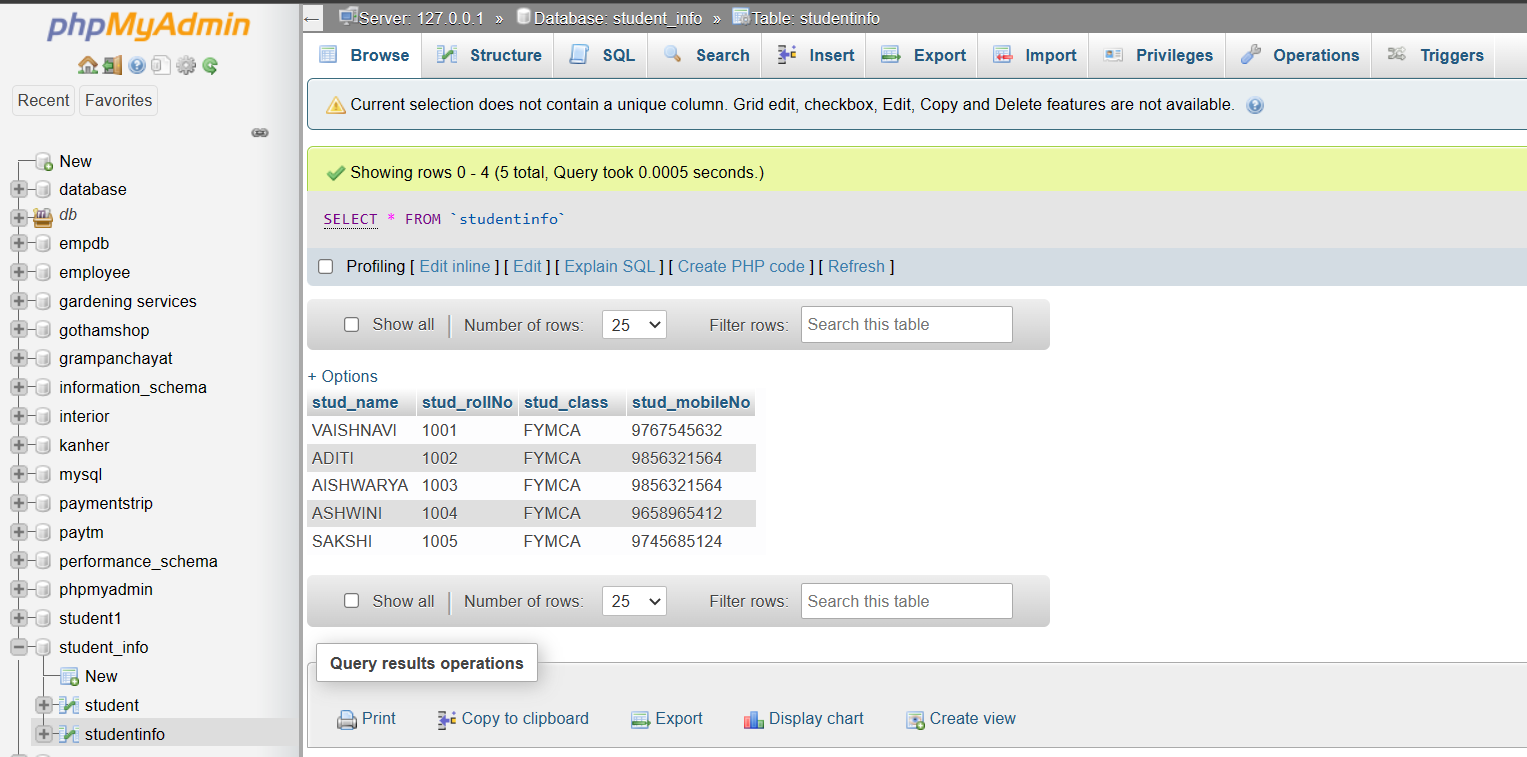
    })

});

**Output:**

****

**Database(student\_info) -Table name(studentInfo):**

****

**Practical No.7**

**Title: Create an application using Filters**

**Exercise –**

Create an application using Filters

**Implementation:**

**Program:**

1)practical7.html

<!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>

    <script src="./myscript1.js"></script>

</head>

<body>

    <div ng-app="firstapp" ng-controller="firstctrl">

        <input type="text" ng-model="test">

        <li ng-repeat="x in colors | filter:test">

           {{x}}

        </li>

    </div>

</body>

</html>

2)myscript1.js

angular.module('firstapp',[]).controller('firstctrl',($scope)=>

{

    $scope.colors=[

    'red',

    'blue',

    'powderblue',

    'tomato',

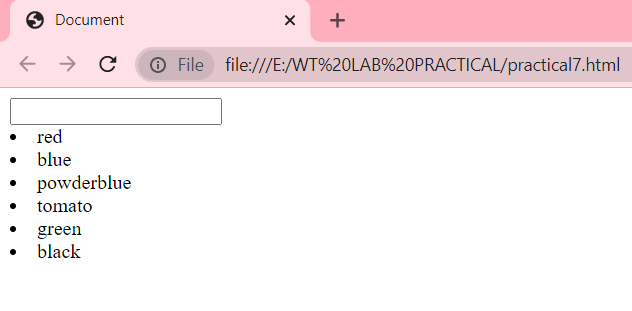
    'green',

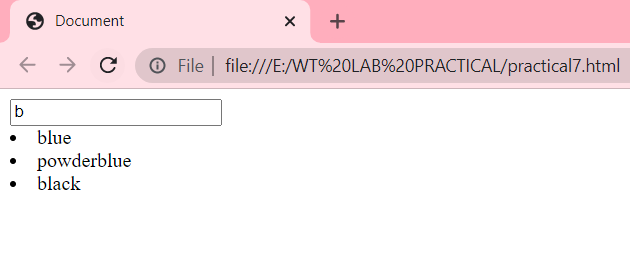
    'black'

    ];

});

**Output:**

****



**Practical No.8**

**Title: Create an application to demonstrate directives**

**Exercise –**

Create an application to demonstrate directives

**Implementation:**

**Program:**

1)NEW.html

<!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="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js">

</script>

</head>

<body>

    <div ng-app="firstapp" >

                    <p>Enter your First Name: <input type="text" ng-model="fname"></p>

                    <p>Enter your Last Name: <input type="text" ng-model="lname"></p>

                    <p ng-bind="fname"></p>

                    <p ng-bind="lname"></p>

                    <p ng-bind="fname+ ' '+lname"></p>

                    <p>{{fname +' '+ lname}}</p>

                    </div>

<script >angular.module('firstapp',[]).controller('firstctrl',($scope)=>

    {

       $scope.fname="";

       $scope.lname="";

    });

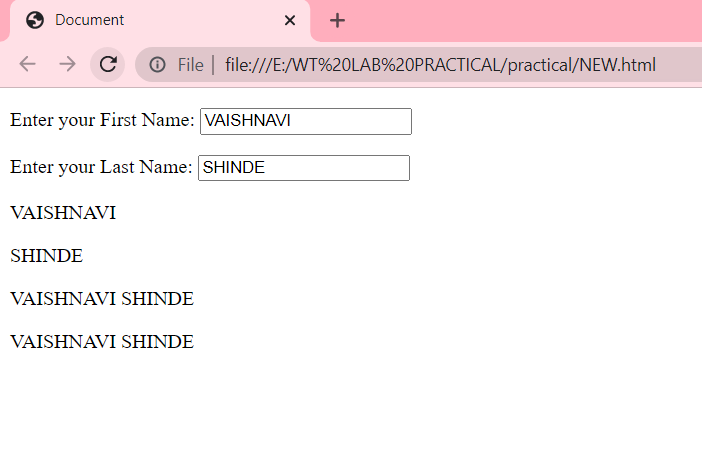
    </script>

    </div>

</body>

</html>

**Output:**

****

**Practical No.9**

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

**Exercise –**

Create an application to demonstrate directives

**Implementation:**

**Program:**

<html>

    <head>

        <script src="https://code.angularjs.org/1.6.9/angular.min.js"></script>

    </head>

    <body>

        <div ng-app="demo" ng-controller="democtr">

           <label>Enter First Name:<input type="text" ng-model="firstname"></label><br>

           <label>Enter Last Name:<input type="text" ng-model="lastname"></label><br>

           Full Name:{{firstname+" "+lastname}}

        </div>

        <script>

            var app=angular.module('demo',[]);//new module

            app.controller('democtr',function($scope){ //define controller

                $scope.firstname="Vaishnavi";

                $scope.lastname="Shinde";

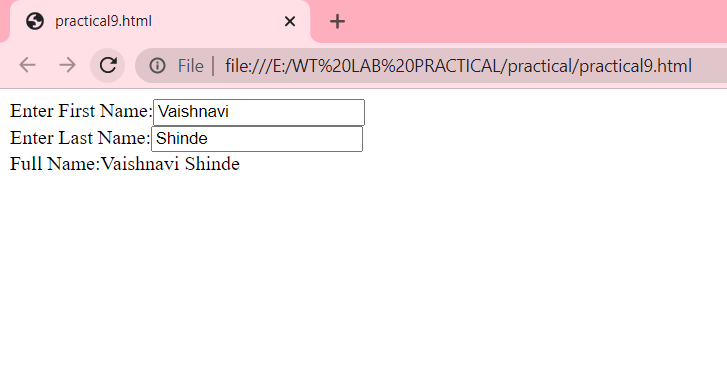
            });

        </script>

    </body>

    </html>

**Output:**

****

**Practical No. 10**

**Title: Create a SPA (Single Page Application)**

**Aim: To Create a SPA (Single Page Application)**

**Implementation:**

**Program:**

**Index\_stud.js**

const express=require('express');

var mysql=require('mysql');

var app=express();//create express app

var bodyparser=require('body-parser');//import body parser

app.use(bodyparser.json());

app.use(bodyparser.urlencoded({extended:true}));

app.get('/student\_info',(req,res)=>{

res.sendFile(\_\_dirname+'/student\_info.html');

});

//database connection

var con=mysql.createConnection({

host:'localhost',

user:'root',

password:'',

database:'student\_info'

});

app.post('/formsubmit',(req,res)=>{

fname=req.body.fname;

mname=req.body.mname;

lname=req.body.lname;

sdiv=req.body.sdiv;

con.connect(function(error){

if(error) throw error;

console.log('connected');

con.query("INSERT INTO student(fname,mname,lname,sdiv) VALUES('"+fname+"','"+mname+"','"+lname+"','"+sdiv+"' )",

function(error,result){

if(error) throw error;

console.log('Record Inserted');

})

})

});

app.listen(8088);

**student\_info.html**

<html>

<head>

<title>Registration</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

\*{

margin: 0;

padding: 0;

font-family: sans-serif;

}

.container{

width: 100%;

height: 100vh;

background: linear-gradient(rgba(0,0,0,0.75),rgba(0,0,0,0.75)),url(bg1.jpg);

background-size: cover;

font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode';

color: #a6a0a0;

display: flex;

align-items: center;

justify-content: center;

}

.card{

width: 450px;

height: 550px;

box-shadow: 0 0 40px rgba(0,0,0,0.80);

perspective: 1000px;

background-image: url("https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.shutterstock.com%2Fsearch%2Fstudent-login&psig=AOvVaw0gMd7kn-ci7rFVbssIDYxF&ust=1676478592859000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCJDVtM23lf0CFQAAAAAdAAAAABAD.jpg");

}

.inner-box{

position: relative;

width: 100%;

height: 100%;

transform-style: preserve-3d;

transition: transform 1s;

}

.card-front, .card-back{

position: absolute;

width: 100%;

height: 100%;

background-position: center;

background-image: linear-gradient(rgba(0,0,10,0.60),rgba(0,0,10,0.60));

padding: 55px;

box-sizing: border-box;

backface-visibility: hidden;

}

.card-back{

transform: rotateY(-180deg);

}

.card h2{

font-weight: normal;

font-size: 24px;

text-align: center;

margin-bottom: 20px;

}

.input-box{

width: 100%;

background: transparent;

border: 1px solid #fff;

margin: 6px;

height: 32px;

border-radius: 20px;

padding: 0 10px;

box-sizing: border-box;

outline: inherit;

text-align: center;

color: #fff;

}

::placeholder{

color: #fff;

font-size: 12px;

}

button{

width: 50%;

background: transparent;

border: 1px solid #fff;

margin: 35px 0 10px;

height: 32px;

font-size: 12px;

border-radius: 20px;

padding: 0 10px;

box-sizing: border-box;

outline: none;

color: #fff;

cursor: pointer;

}

.submit-btn::after{

color: #333;

line-height: 32px;

font-size: 17px;

height: 32px;

width: 32px;

border-radius: 50%;

background: #fff;

position: absolute;

right: -1px;

top: -1px;

}

span{

font-size: 13px;

margin-left: 70px;

}

.card .btn{

margin-top: 70px;

}

.card a{

color: #fff;

text-decoration: none;

display: block;

text-align: center;

font-size: 13px;

margin-top: 8px;

}

</style>

</head>

<body>

<div class="container">

<div class="card">

<div class="inner-box" id="card">

<div class="card-front">

<h2> Student Login</h2>

<form action="formsubmit" method="POST">

<center >

<input type="text" class="input-box" placeholder="Enter First Name" name="fname" id="fname">

<br>

<input type="text"class="input-box" placeholder="Enter Middle Name" name="mname" id="mname">

<br>

<input type="text" class="input-box" placeholder="Enter Last Name" name="lname" id="lname">

<br>

<input type="text" class="input-box" placeholder="Enter Division" name="sdiv" id="sdiv">

<br>

<br>

<input type="submit">

</center>

</div>

</div>

</div>

</div>

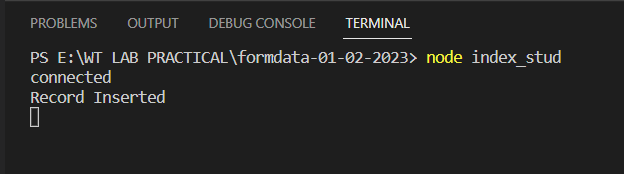
</form>

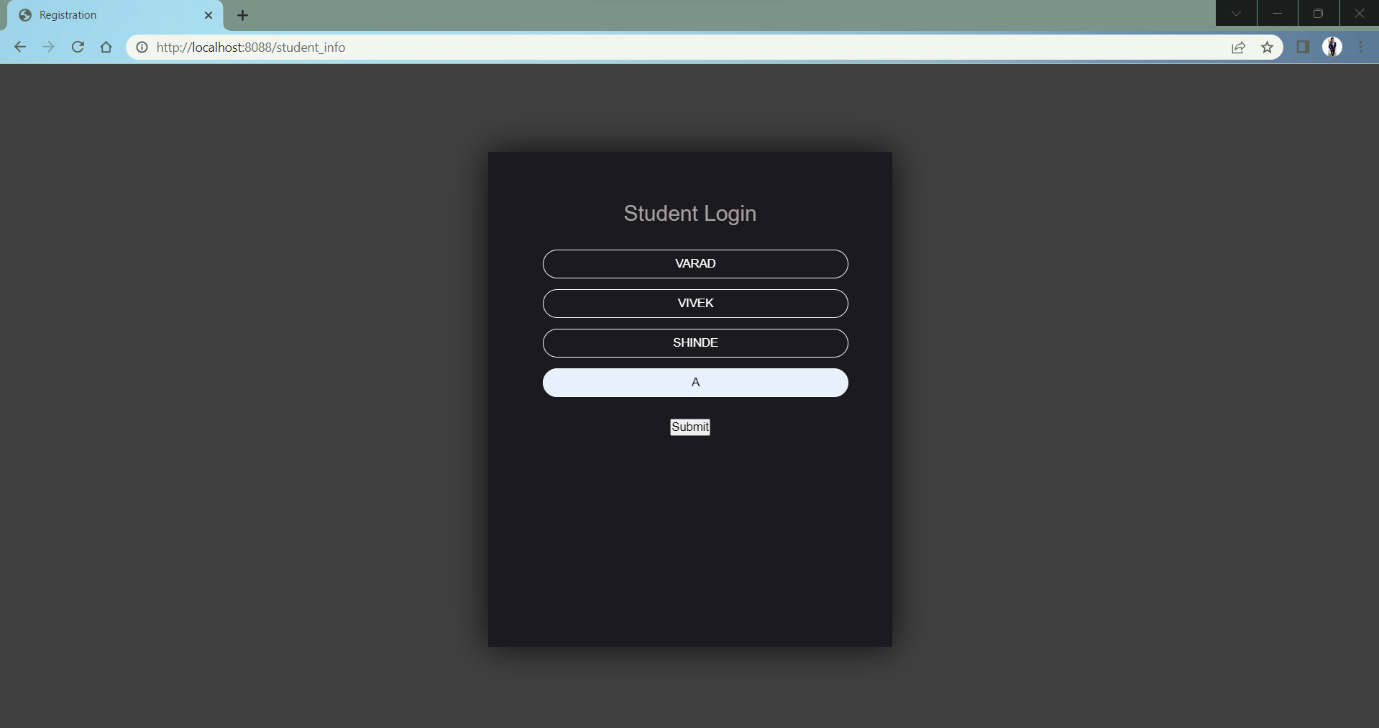
</body>

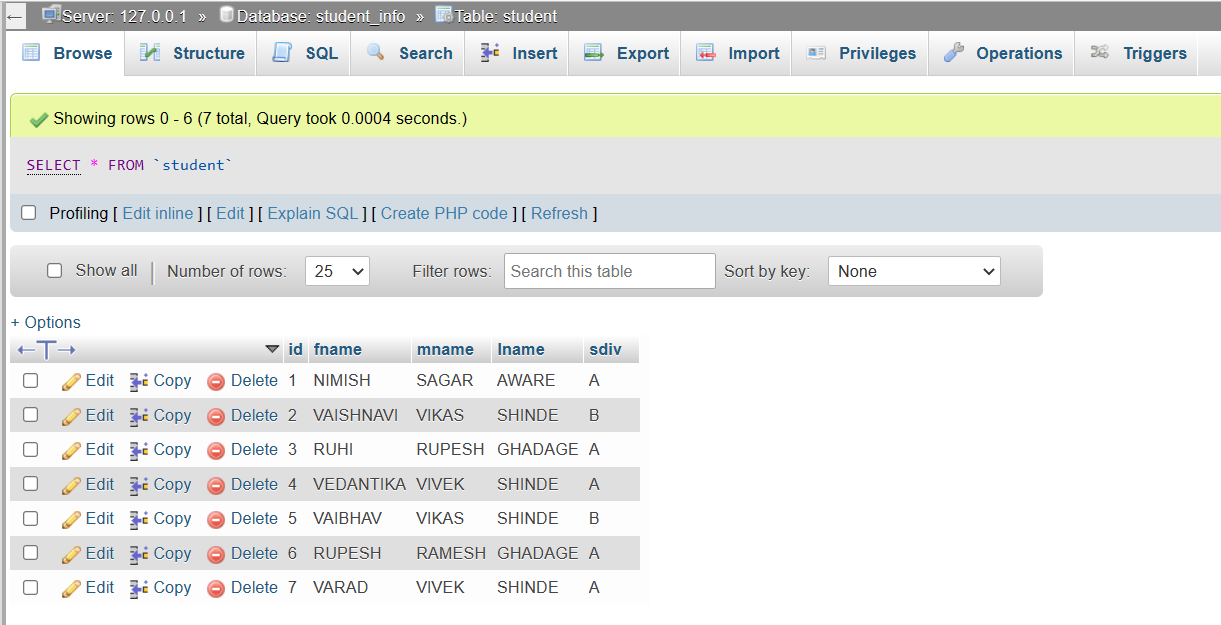
</html>

**Output:**

**1)**

****

**2)**

**3)**