Institute of Distance and Open Learning,

Vidyanagari, Kalina, Santacruz (E) -400098

Institute of Distance and Open Learning (IDOL)
University of Mumbai.

CERTIFICATE

This is to certify that

Mr. Deepesh Mangesh Mhatre
(Application No : 81389)

Has satisfactorily completed the practical of subject

Web Technology

Towards the partial fulfillment of the
MASTER OF COMPUTER APPLICATION (MCA)
As laid by the University of Mumbai.

Principal	**External Examiner**	**Internal Guide**

INDEX

S. No.	Name of the Practical	Signature
	Node JS Practical Index	
1.	Write a program to pass a message “Hello Node JS” using Node JS.	
2.	Write a program to demonstrate Node.js Functions.	
3.	Write a program to demonstrate the Call-back function - Anonymous function using Node JS.	
4.	Write a program to demonstrate Node.js Modules.	
5.	Write a program to demonstrate various Node.js Events.	
6.	Create an HTTP Server and perform operations on it.	
7.	Write a program to demonstrate routing through an HTTP server.	
8.	Write a program to demonstrate custom events using Node JS.	
9.	Using File Handling demonstrate all basic file operations (Create, write, read, delete & buffer)	
10.	Create an application to establish a connection with the MySQL database and perform basic database operations on it.	

MCAL14: Web Technologies Lab

S. No.	Name of the Practical	Signature
	Angular JS Practical Index	
1.	Write a program to perform Arithmetic operations (Addition, Subtraction, Multiplication, Division) using Angular JS.	
2.	Write a program to demonstrate Angular JS Expressions (Using String).	
3.	Write a program to demonstrate Angular JS Expressions (Using Numbers & String).	
4.	Write a program to demonstrate Objects as Expressions using Angular JS.	
5.	Write a program to perform Validations in form using Angular JS.	
6.	Write a program to perform Validations in form using Angular JS. (Validation True/False)	
7.	Write a program to demonstrate Angular JS Filters (Uppercase & Lowercase).	
8.	Write a program to demonstrate Currency Filter using Angular JS.	
9.	Write a code to Demonstrate Student Info form using Angular JS.	
10	Write a code to Demonstrate Student Info form using Bootstrap in Angular JS.	
11.	Write a code to Design a form in Bootstrap.	
12.	Write an Angular JS script to demonstrate $digest () function.	
13.	Write an Angular JS script to demonstrate $apply () function.	
14.	Create a webpage to change the background color (bgcolor) of the table dynamically using Angular JS.	
15.	Write an AngularJS script to print details of a bank name, MICR code, IFC code, address, etc. in tabular form. (use ng-repeat)	
16.	Write a script to design a table using controller ng-repeat and different HTML tags in Angular JS.	
17.	Write a web page to process student marks using Angular JS controller.	
18.	Create a web page to fetch multiple images and change images dynamically using AngularJS.	
19.	Create a web page to process product details using Angular JS controller organize product data using an array of objects.	
20.	Create a web page to update the previous example to display the product details in table format.	
21.	Create a webpage to demonstrate the wege of filters in AngularJS.	
22.	Create a webpage to demonstrate usage of order by the filter in AngularJS.	
23.	Create a webpage to sort the student details using order by the filter in Angular JS.	
24.	Create a webpage to sort the student detail based on the Selected column using order by the filter in Angular JS.	
25.	Create a webpage to apply searching on student details by using the filter option of Angular JS.	

Name: DeepeshMangeshMhatre

App No : 81389

MCAL14: Web Technologies Lab

S. No.	Name of the Practical	Signature
	Angular JS Practical Index	
26.	Create a webpage to organize product data so that it is always the user to search & sort based on the given scenarios.	
27.	Create a web page to perform Key Event Directives “ng-key down” in Angular JS.	
28.	Create a webpage to perform Mouse Event Directives “ng- mouse down” in Angular JS.	
29.	Write a webpage to perform Mouse Event Directives “ng- mouseenter” in Angular JS.	
30.	Create a webpage to perform Mouse Event Directives “ng- mouseenter” in Angular JS. (Colour Change)	
31.	Write a script to create a webpage to implement a login function using Angular JS event.	
32.	Write a script to create a math operation using the Angular JS event.	
33.	Write a script using built-in object $scope in Angular JS, which contains application data and method.	
34.	Write a script to demonstrate $rootscope in Angular JS.	
35.	Create a webpage to demonstrate Single Page Application (SPA) using Angular JS.	
36.	Create a webpage to demonstrate Advance Single Page Application (SPA) using Angular JS.	
37.	Create a webpage to demonstrate Single Page Application (SPA) for student login and form-filling using Angular JS.	

Node JS Practical

Practical No. 1

Aim: Write a program to pass a message “Hello Node JS” using Node JS.

Line of code:

1)

```
Node.js command prompt
Your environment has been set up for using Node.js 16.14.0 (ia32) and npm.

C:\Users\Dell>f:

F:\>cd FY-MCA

F:\FY-MCA>cd nodejs
The system cannot find the path specified.

F:\FY-MCA>cd node js

F:\FY-MCA\node js>node
Welcome to Node.js v16.14.0.
Type ".help" for more information.
> console.log("Hello world");
Hello world
undefined
> .save helloworld.js
Session saved to: helloworld.js
>
(To exit, press Ctrl+C again or Ctrl+D or type .exit)
>

F:\FY-MCA\node js>node helloworld.js
Hello world
```

2)

```
console.log("Hello nodejs");
```

Execute Screen:

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS F:\FY-MCA\node js> node hello.js
Hello nodejs
PS F:\FY-MCA\node js>
```

Name: DeepeshMangeshMhatre

App No : 81389

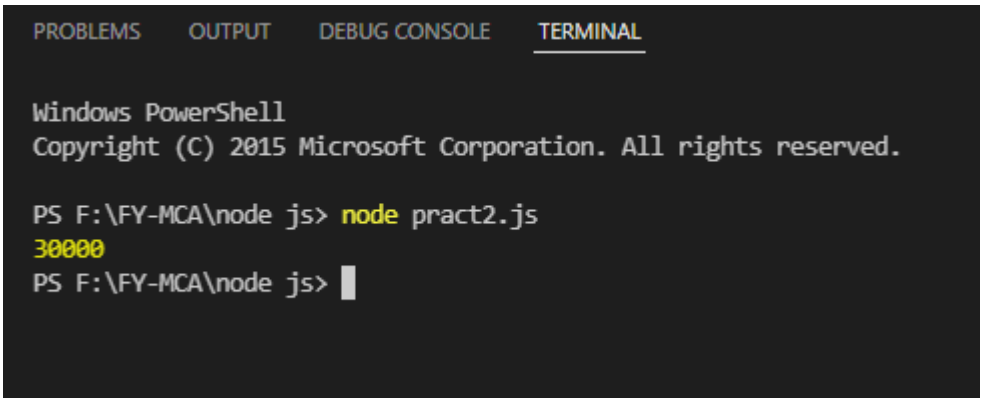
Practical No. 2

Aim: Write a program to demonstrate Node.js Functions.

Line of Code:

```
//nodejs function

function multi(x,y)

{

   return x*y;

}

var result = multi(100,300);

console.log(result);
```

Execution Screen:



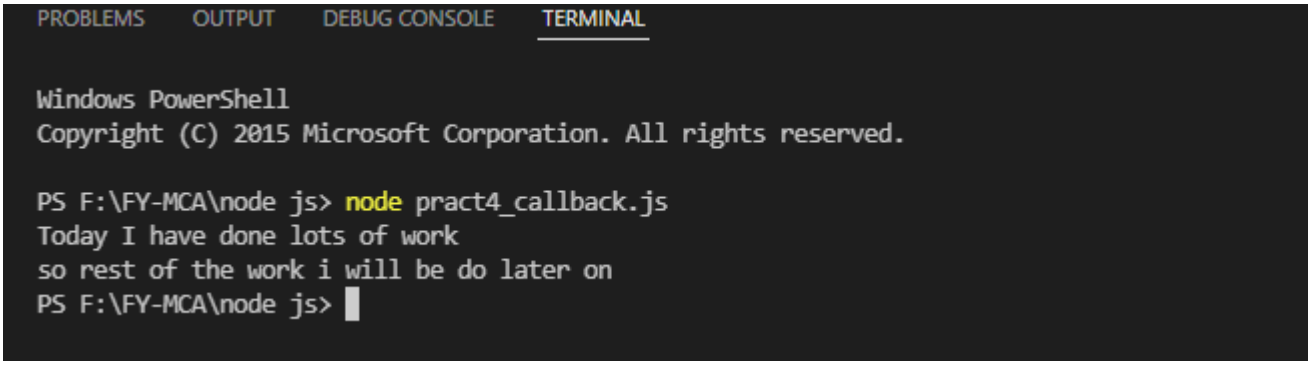
Practical No. 3

Aim: Write a program to demonstrate the Call-back function - Anonymous function using Node JS.

Line of Code:

```
/* What is Call back function
A callback is a function passed as an argument to another function.


/*Standard  function definition
function <name_of_function> ()
{
   return statement;
}
*/
//Callback function - Anonymous function
const message = function ()
{
   console.log("Today I have done lots of work");
}
setTimeout(message,3000);
//callback function as an arrow function
setTimeout(() =>
{
   console.log("so rest of the work i will be do later on");
}, 3000);
```

Execution Screen:



Practical No. 4

Aim: Write a program to demonstrate Node.js Modules.

Line of code:

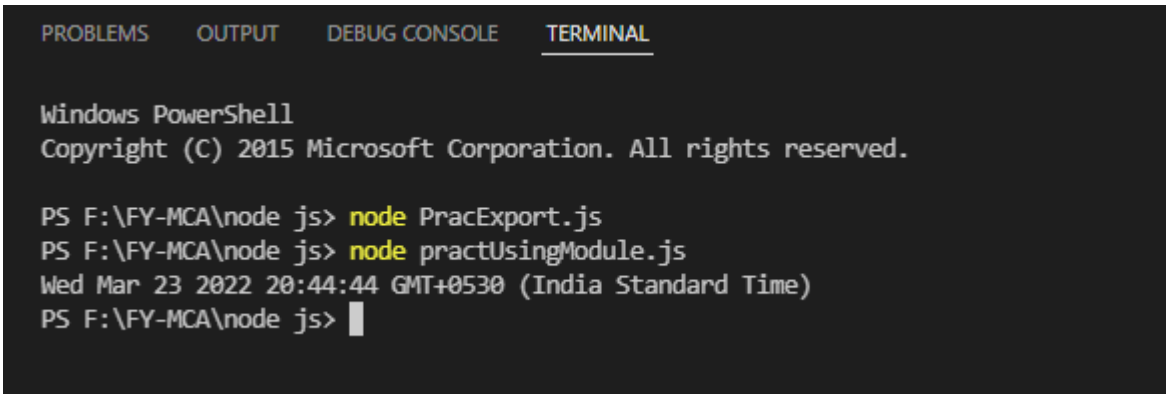
1) pracExport.js

```
//creating module and export it

exports.myDateFun=function()

{

 return Date();

}
```

2) practUsingModule.js

```
//understand how to call the modules

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

console.log(dt.myDateFun());
```

Execution Screen:



Practical No. 5

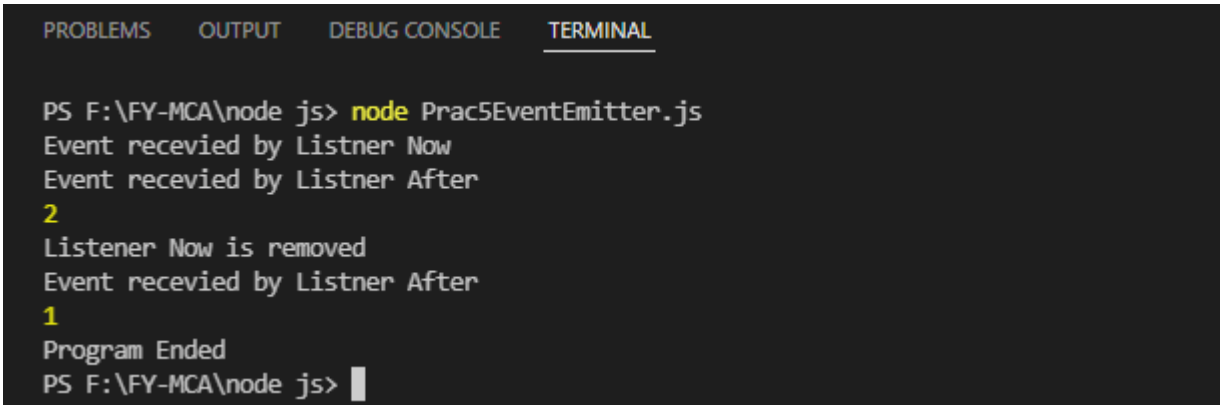
Aim: Write a program to demonstrate various Node.js Events.

Line of code:

```
//pract 5 events
// step 1 importing event
const events = require("events");
// step 2 creating an Event emitter object
const eventEmitter = new events.EventEmitter();
//write a function of event 1
function listnerNow() {
 console.log("Event recevied by Listner Now");
}
//write a function of event 2
function listnerAfter() {
 console.log("Event recevied by Listner After");
}
// step 3 adding listener through addlistener or on
eventEmitter.addListener("write", listnerNow);
eventEmitter.on("write", listnerAfter);
// step 4 emiting event
eventEmitter.emit("write");
console.log(eventEmitter.listenerCount("write"));
// step 5 removing listener
eventEmitter.removeListener("write", listnerNow);
console.log("Listener Now is removed");
eventEmitter.emit("write");
```

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

console.log("Program Ended")
```

Execution Screen:



Practical No. 6

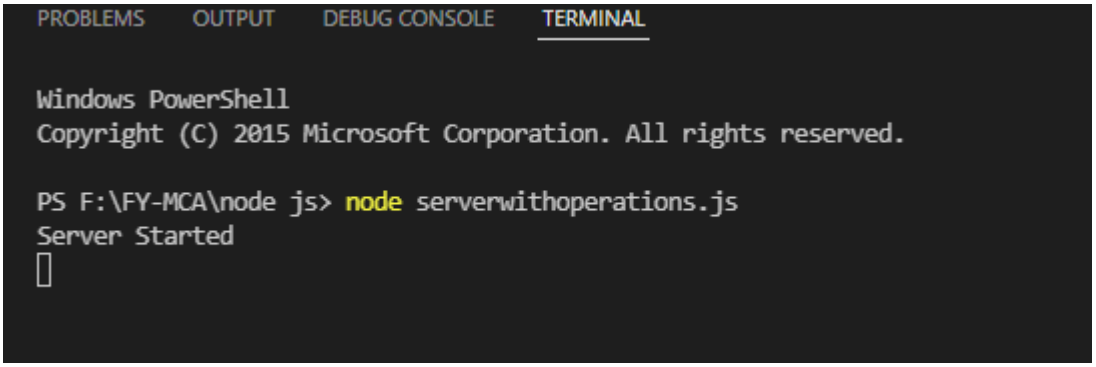
Aim: Create an HTTP Server and perform operations on it.

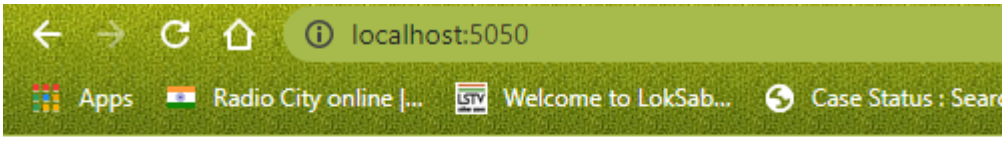
Line of code:

1)

```
//1.
var http = require('http');
//2.
var server = http.createServer(function (req, resp) {
    //3.
    resp.writeHead(200, { 'Content-Type': 'text/html' });
    //4.
    resp.end("Hello, this is the test web server created");
});
//5.
server.listen(5050);
console.log('Server Started');
```

Execution Screen:





Hello, this is the test web server created

2)

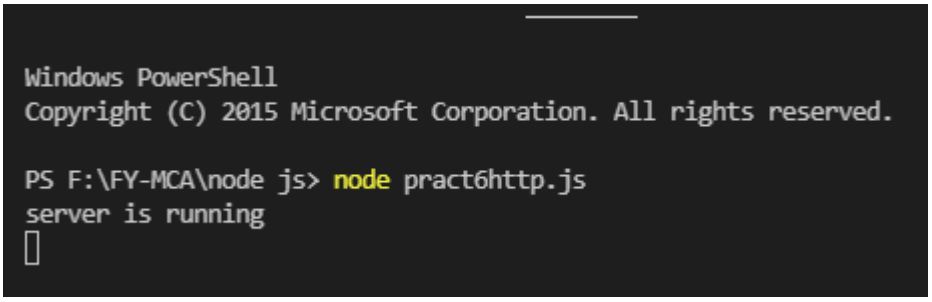
myfirstmodule.js

```
//creating module and export it
exports.myDateFun=function()
{
 return Date();
}
```

pract6http.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 is currently: " + dt.myDateFun());
  res.end();
}).listen(8080);
console.log("server is running");
```

Execution Screen:



on localhost: 8080

The date and time is currently: Wed Mar 23 2022 21:02:35 GMT+0530 (India Standard Time)

Practical No. 7

Aim: Write a program to demonstrate routing through an HTTP server.

Line of Code:

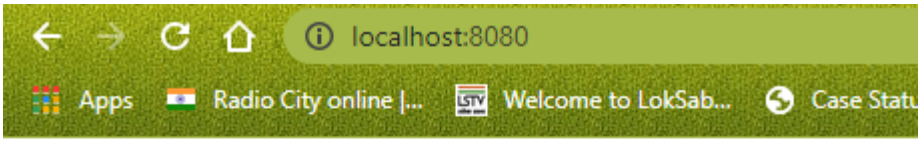
```
//understand routing through http server//pract7
var http=require('http');
var server=http.createServer(function(req,res){
   if(req.url=='/')
{
   res.writeHead(200,{'content-Type':'text/html'})
   res.write("<h1>Home Page</h1><br><br>");
   res.write("<h4>Wellcom to Collage Website</h4>")
   res.end();
}
else if (req.url=='/admin')
{
   res.writeHead(200,{'content-Type':'text/html'})
   res.write("<h1>Admin Page</h1>");
   res.write("Admin Login Page");
   res.end();
}
else if (req.url=='/student')
{
   res.writeHead(200,{'content-Type':'text/html'})
   res.write("<h1>Student Detail Page</h1>");
   res.write("Wellcome to Student");
}
else if (req.url=='/teacher')
{
   res.writeHead(200,{'content-Type':'text/html'})
   res.write("<h1>Teacher Detail Page</h1>");
   res.write("Wellcome to teacher");
   res.end();
}
```

```
else if (req.url=='/staff')
{
    res.writeHead(200,{'content-Type':'text/html'})
    res.write("<h1>Staff Detail Page</h1>");
    res.write("Wellcome to Staff");
    res.end();
}
else
{
    res.write("<h1>Invalid Page</h1>");
    res.end();
}
});
server.listen(8080);
console.log("server is running");
```

Execution Screen:

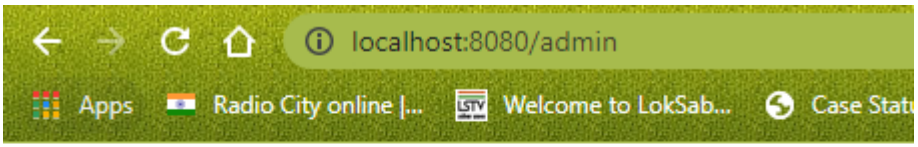
```
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS F:\FY-MCA\node js> node pract7routing.js
server is running
```

on localhost:8080



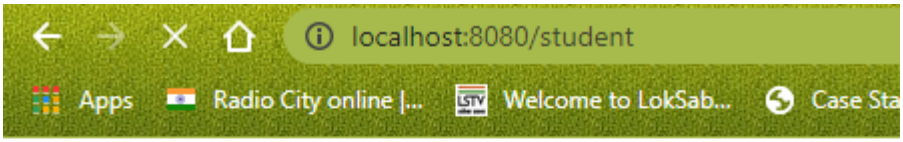
Home Page

Wellcom to Collage Website



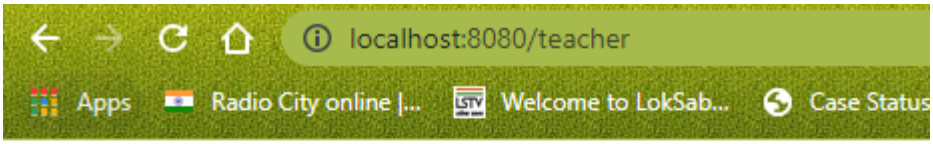
Admin Page

Admin Login Page



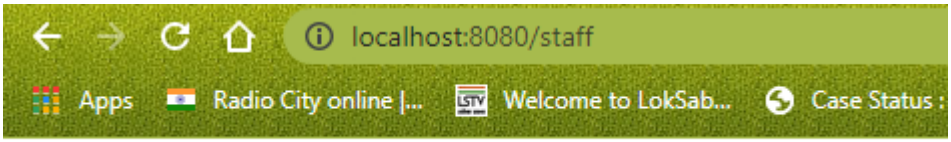
Student Detail Page

Wellcome to Student



Teacher Detail Page

Wellcome to teacher



Staff Detail Page

Wellcome to Staff

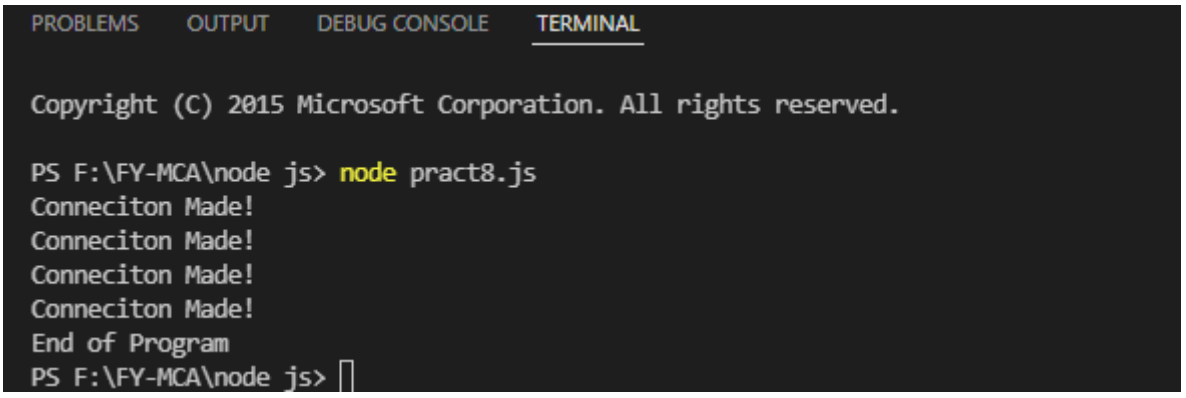
Practical No. 8

Aim: Write a program to demonstrate custom events using Node JS.

Line of Code:

```
//custom event//pract8


const events = require("events");
const eventEmitter = new events.EventEmitter();
eventEmitter.on("connection", handleConnectionEvent);
eventEmitter.emit("connection");
eventEmitter.emit("connection");
eventEmitter.emit("connection");
eventEmitter.emit("connection");
function handleConnectionEvent() {
    console.log("Conneciton Made!");
}
console.log("End of Program");
```

Execution Screen:



Practical No. 9

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

Line of Code:

1) Fileoperation.txt(Create file)

Hello I am from input text file

This is my second line

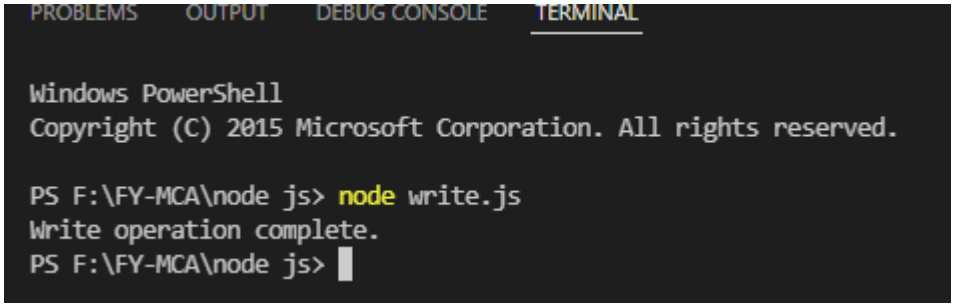
This is my third line

This will be the final line

2) Write operation:

```
//writing file
var fs=require('fs');
fs.writeFile('Myfile.txt','This is my file @viva Institute of Technology!!',function(err){
   if (err)
   console.log(err);
   else
   console.log('Write operation complete.');
});
```

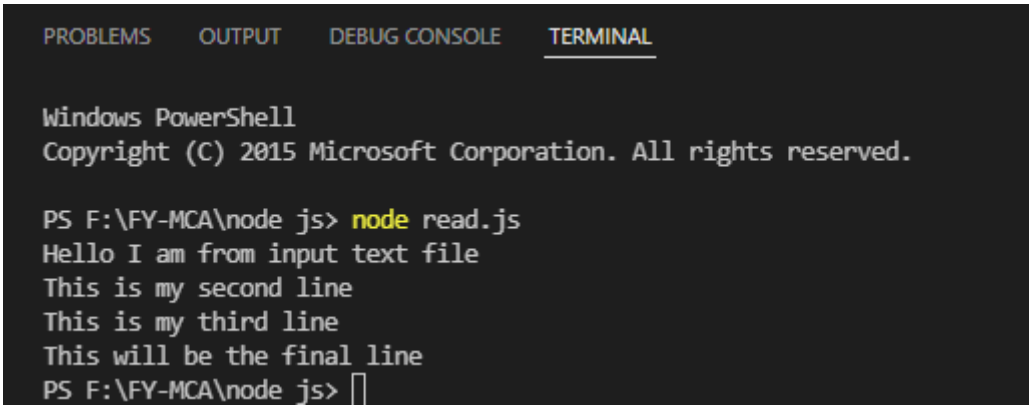
Output:



3) reading file operation

```
//node js file system
//reading file

var fs = require('fs');

fs.readFile('Fileoperation.txt', function (err, data) {
                if (err) throw err;
   console.log(data.toString());
});
```

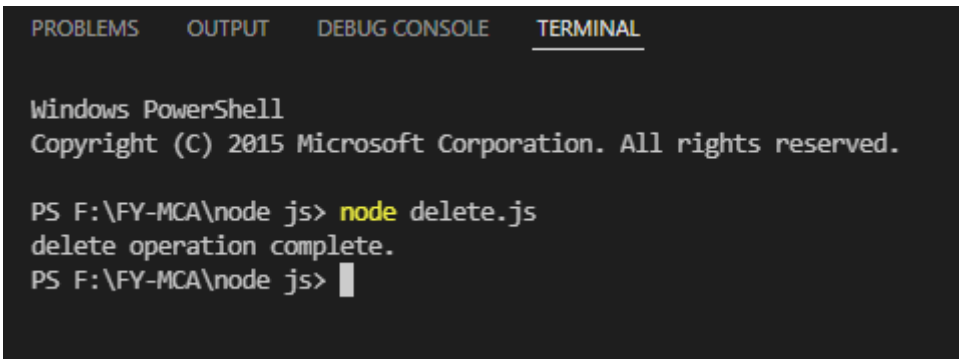
Output:



4) Delete file operation

```
//delete the file
var fs = require('fs');
fs.unlink('delete.txt',function(){
   console.log('delete operation complete.');
});
```

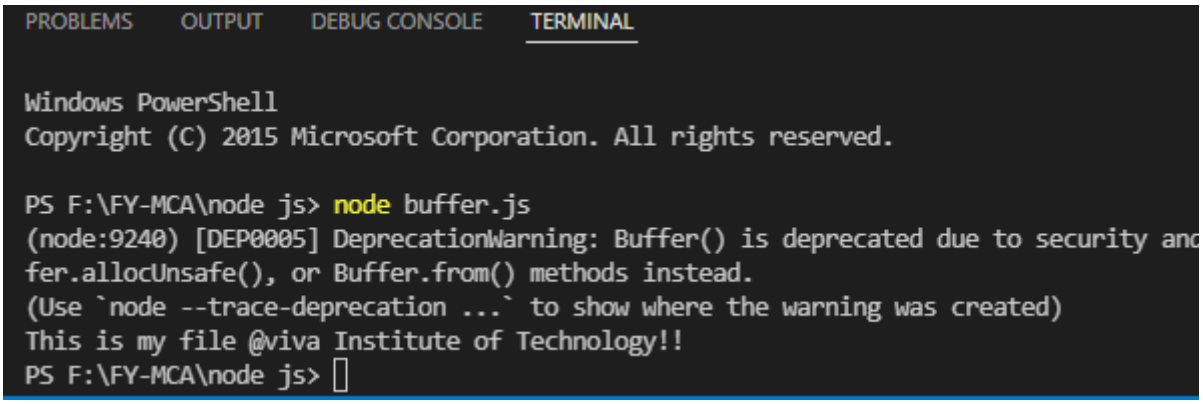
Output:



5) Buffer operation

```
var fs = require('fs');
fs.open('Myfile.txt', 'r', function (err, fd) {
    if (err) {
        return console.error(err);
  }
    var buffr = new Buffer(10240);
    fs.read(fd, buffr, 0, buffr.length, 0, function (err, bytes) {
    if (err) throw err;

   // Print only read bytes to avoid junk.
        if (bytes > 0) {
        console.log(buffr.slice(0, bytes).toString());
    }

 // Close the opened file.
    fs.close(fd, function (err) {
        if (err) throw err;
    });
  });
});
```

Output:



Practical No. 10

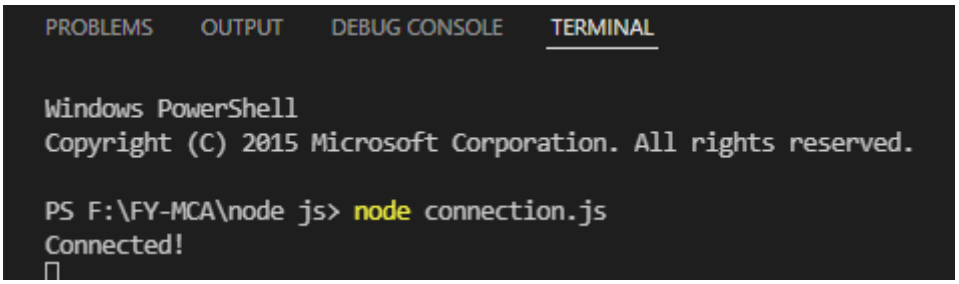
Aim: Create an application to establish a connection with the MySQL database and perform basic database operations on it.

Line of Code:

1) Creating Connection

```
//checking database connection
var mysql = require('mysql2');


var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "Viva@123",
  port:3306
});


con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
});
```

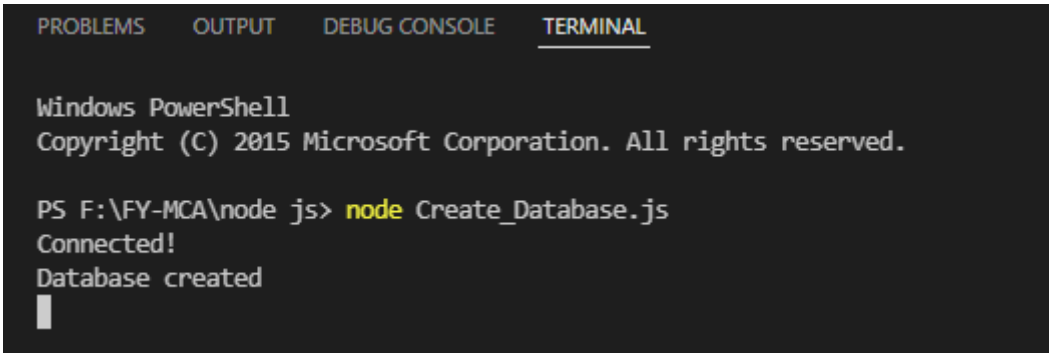
Output:



2) Creating Database:

```
//creating database
var mysql = require('mysql2');


var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "Viva@123",
  port: 3306
});


con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  con.query("CREATE DATABASE Fymca2", function (err, result) {
    if (err) throw err;
    console.log("Database created");
  });
});
```

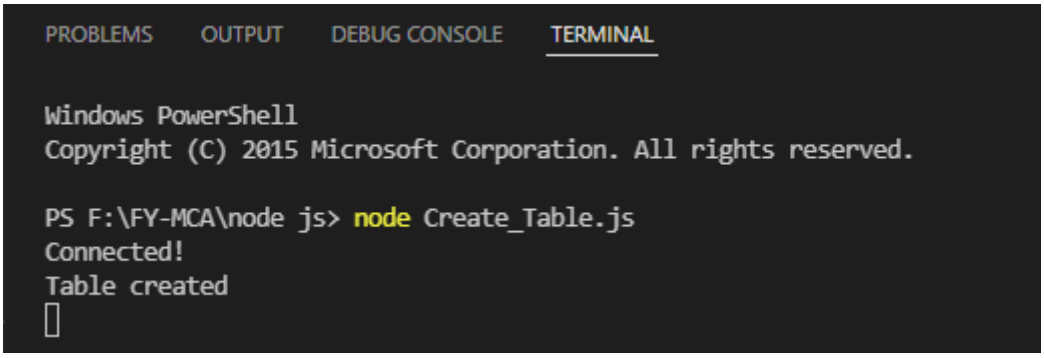
Output:



3) Creating Table

```
//creating table in database
var mysql = require('mysql2');

var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "Viva@123",
  database: "Fymca1"
});

con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  var sql = "CREATE TABLE customers2(name VARCHAR(255), address VARCHAR(255))";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("Table created");
  });
});
```

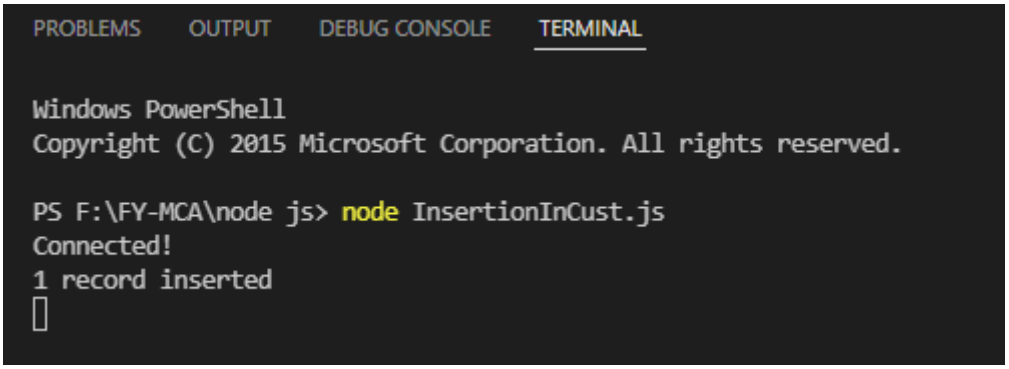
Output:



4) Insert Record

```
//inserting record inside table
var mysql = require('mysql2');

var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "Viva@123",
  database: "Fymca1"
});

con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  var sql = "INSERT INTO customers2 (name, address) VALUES ('Company Inc', 'Highway 37')";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("1 record inserted");
  });
});
```

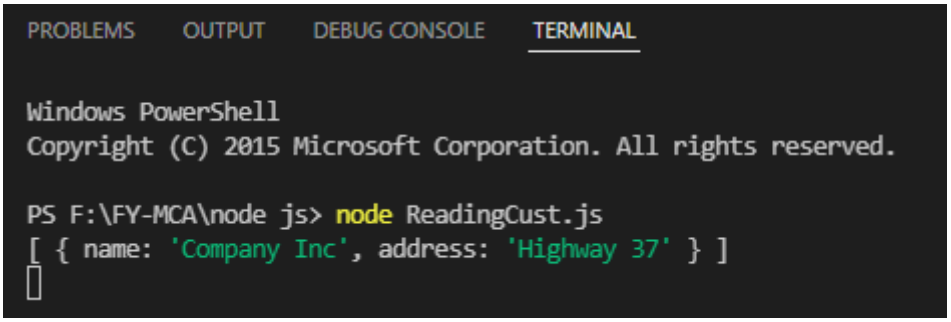
Output:



5) Reading Record

```
var mysql = require('mysql2');


var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "Viva@123",
  database: "Fymca1"
});


con.connect(function(err) {
  if (err) throw err;
  con.query("SELECT * FROM customers2", function (err, result, fields) {
    if (err) throw err;
    console.log(result);
  });
});
```

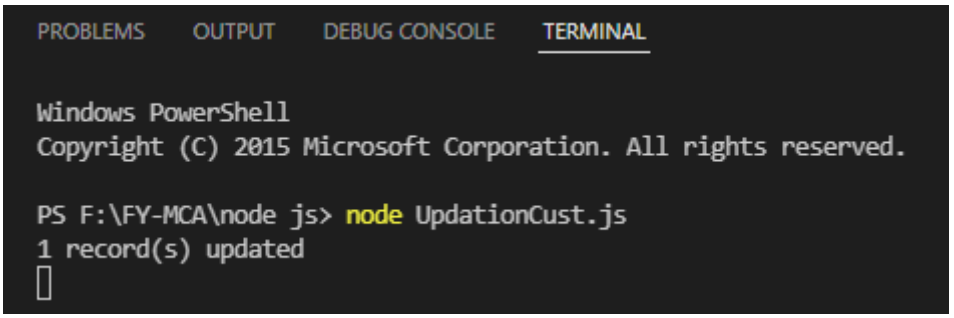
Output:



6) Update Record

```
var mysql = require('mysql2');


var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "Viva@123",
  database: "Fymca1"
});


con.connect(function(err) {
  if (err) throw err;
  var sql = "UPDATE customers2 SET address = 'Canyon 123' WHERE address = 'Highway 37'";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log(result.affectedRows + " record(s) updated");
  });
});
```

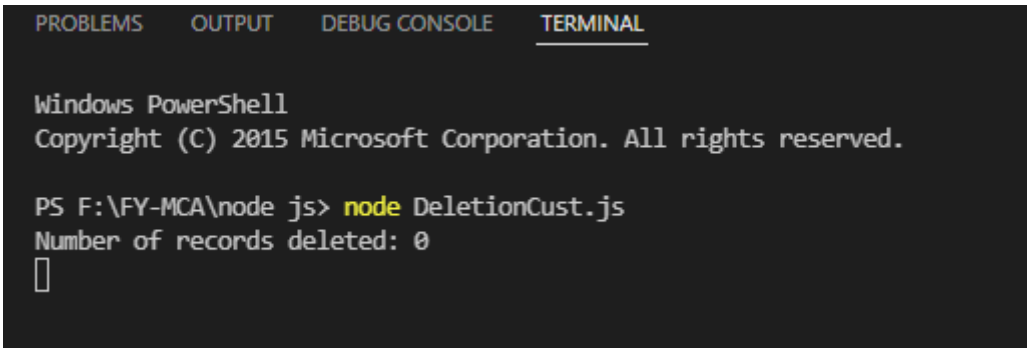
Output:



7) Delete Record

```
var mysql = require('mysql2');


var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "Viva@123",
  database: "Fymca1"
});


con.connect(function(err) {
  if (err) throw err;
  var sql = "DELETE FROM customers2 WHERE address = 'Highway 37'";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("Number of records deleted: " + result.affectedRows);
  });
});
```

Output:



ANGULAR JS PRACTICAL

Practical No. 1

Aim: Write a code to demonstrate Arithmetic Operators using Angular JS.

Line of code:

```
<!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="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-app="">
    <h1>The addition is 2+2={{2+2}}</h1> <br>
    <h1>The substraction is 2-2={{2-2}}</h1> <br>
    <h1>The multiplication is 2*2={{2*2}}</h1> <br>
    <h1>The division is 2/2={{2/2}}</h1> <br>
</body>
</html>
```

Execute Screen:

The addition is 2+2=4

The substraction is 2-2=0

The multiplication is 2*2=4

The division is 2/2=1

Practical No. 2

Aim: Write a code to demonstrate controllers using Angular JS through an application.

Line of code:

```
<!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">
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
    <title>Document</title>
</head>
<body ng-app="" ng-init="fname='Deepali';lname='Shivde'">
    {{"Deepali Shivde"}} <br>
    <p>
        My First Name is <span ng-bind=fname></span> <br>
        <b>First Name</b> {{fname}}
    </p>
    <p>
        My Last Name is <span ng-bind=lname></span> <br>
        <b>Last Name</b> {{lname}} <br>
    </p>
    <p>
        My Full Name is {{fname+' '+lname}}
    </p>
</body>
</html>
```

Execute Screen:

Deepali Shivde

My First Name is Deepali
First Name Deepali

My Last Name is Shivde
Last Name Shivde

My Full Name is Deepali Shivde

Practical No. 3

Aim: Create an application to demonstrate Arithmetic Operators and controllers using Angular JS.

Line of code:

```
<!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="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-app="">
   <div>
      2+2={{2+2}} <br>
   </div>

   <div ng-init="fname='Deepali';lname='Shivde'">
      2-2={{2-2}} <br>
      2*2={{2*2}} <br>
      2/2={{2/2}} <br>

      {{"Deepali Shivde"}} <br>
         <p>
            My First Name is <span ng-bind=fname></span> <br>
            <b>First Name</b> {{fname}}
         </p>
         <p>
            My Last Name is <span ng-bind=lname></span> <br>
            <b>Last Name</b> {{lname}} <br>
         </p>


         <p>
            My Full Name is {{fname+' '+lname}}
         </p>
   </div>
</body>
</html>
```

MCAL14: Web Technologies Lab

Execute Screen:-

2+2=4
2-2=0
2*2=4
2/2=1
Deepali Shivde

My First Name is Deepali
First Name Deepali

My Last Name is Shivde
Last Name Shivde

My Full Name is Deepali Shivde

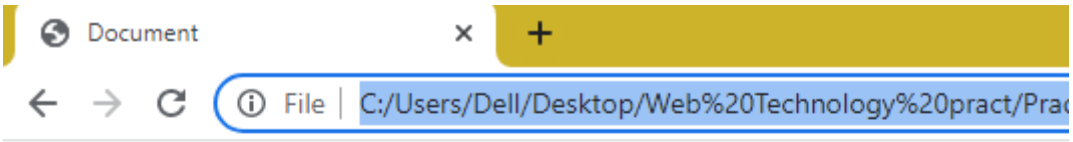
Practical No. 4

Aim: Write a code to demonstrate controllers using Angular JS through an application.

Line of code:

```
<!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="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-app="" ng-init="College={fname:'VIVA', mname: 'Technical', lname: 'Campus'}">
{{College.fname}} {{College.mname}} {{College.lname}} <br>
<p>
   My College First Name is <span ng-bind='College.fname'></span><br>
   <b>First Name</b> {{College.fname}}
</p>
<p>
   My College Last Name is <span ng-bind='College.lname'></span><br>
   <b>Last Name</b> {{College.lname}}
</p>
<p>
   My College Middle Name is <span ng-bind='College.mname'></span><br>
   <b>Middle Name</b> {{College.mname}}
</p>
</body>
</html>
```

Execute Screen:



VIVA Technical Campus

My College First Name is VIVA
First Name VIVA

My College Last Name is Campus
Last Name Campus

My College Middle Name is Technical
Middle Name Technical

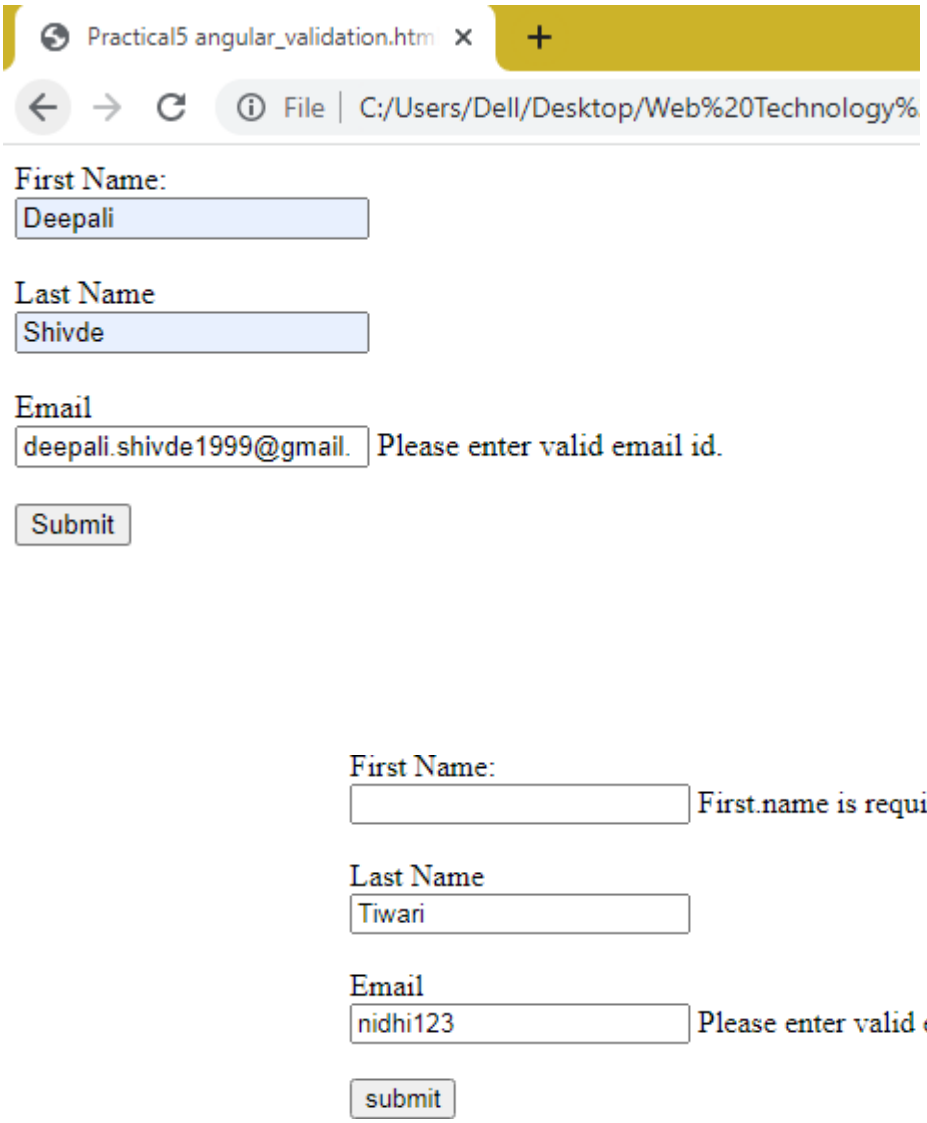
Practical No. 5

Aim: Create an application to demonstrate validation using Angular JS.

Line of code:

```
<!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="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/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-minlenght="3" ng-model="student.lastname"/>
        <span ng-show="studentForm.firstname.$touched && studentForm.firstname.$error.minlength">min 3
chars.</span>
        <span ng-show="studentForm.firstname.$touched && studentForm.firstname.$error.minlength">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>
```

Execute Screen:


Practical5 angular_validation.htm
File | C:/Users/Dell/Desktop/Web%20Technology%
First Name:
Deepali
Last Name
Shivde
Email
deepali.shivde1999@gmail.
Please enter valid email id.
Submit
First Name:
First.name is required
Last Name
Tiwari
Email
nidhi123
Please enter valid email id
submit

Practical No. 6

Aim: Create an application to demonstrate form validation using Angular JS.

Line of code:

```
<!DOCTYPE html>
<html>
<head>
      <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-app>
       <form name="studentForm" novalidate>
<p>
                  First Name Status: <br />
                  Pristine: {{studentForm.firstName.$pristine}} <br /> Touched:
                  {{studentForm.firstName.$touched}}<br /> Untouched:
                  {{studentForm.firstName.$untouched}}<br />Valid:
                  {{studentForm.firstName.$valid}} <br /> Invalid:
                  {{studentForm.firstName.$invalid}} <br /> Dirty:
                  {{studentForm.firstName.$dirty}}   <br   /> Error:
                  {{studentForm.firstName.$error}} <br />

            </p>
<label for="firstName">First Name: </label> <br />
                  <input type="text" name="firstName" ng-model="student.firstName" ng-required="true" />
                  <span ng-show="studentForm.firstName.$touched && studentForm.firstName.$error.required">First
name is required.</span><br /><br />
                  <label for="lastName">Last Name</label><br /><input type="text" name="lastName" ng-minlength="3"
ng-maxlength="10" ng-model="student.lastName" /> <br />
                  <span ng-show="studentForm.lastName.$error.minlength">min 3 chars.</span>
                  <span ng-show="studentForm.lastName.$error.maxlength">Max 10 chars.</span>
<br/>

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

Execute Screen:

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
[n]
min 3 chars.
[Save]

MCAL14: Web Technologies Lab

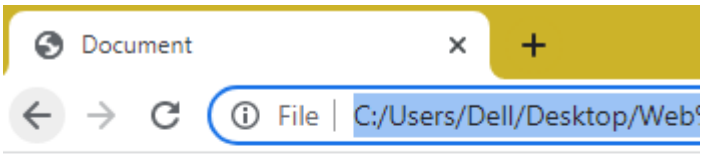
Practical No. 7

Aim: Create an application to demonstrate form validation using Angular JS.

Line of code:

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

</head>
<body>
    <div ng-app="myapp" ng-controller="uppercaseCtrl">
        <p>The name is {{firstname+' '+lastname|uppercase }}</p>
        <p>The name is {{firstname+' '+lastname|lowercase }}</p>
    </div>
    <script>
        angular.module('myapp',[]).controller('uppercaseCtrl',function($scope){
            $scope.firstname='Nidhi',
            $scope.lastname='Tiwari'
        });
        </script>

</body>
</html>
```

Execute Screen:



The name is DEEPALI SHIVDE

The name is deepali shivde

Practical No. 8

Aim: Write a code to demonstrate currency filters using Angular JS.

Line of code:

```
<!DOCTYPE html>
<html lang="en">
<head>
	<meta charset="UTF-8">
	<meta name="viewport" content="width=device-width, initial-scale=1.0">
	<title>Document</title>
	<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-app="myApp" ng-controller="currencyCtrl">
	<p>Price: {{price|currency}}</p>

</body>
<script> angular.module("myApp",[]).controller("currencyCtrl",function($scope){
		$scope.price=100;

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

Execute Screen:

Price: $100.00

Practical No. 9

Aim: Write a code to Demonstrate the Student Info form using Angular JS.

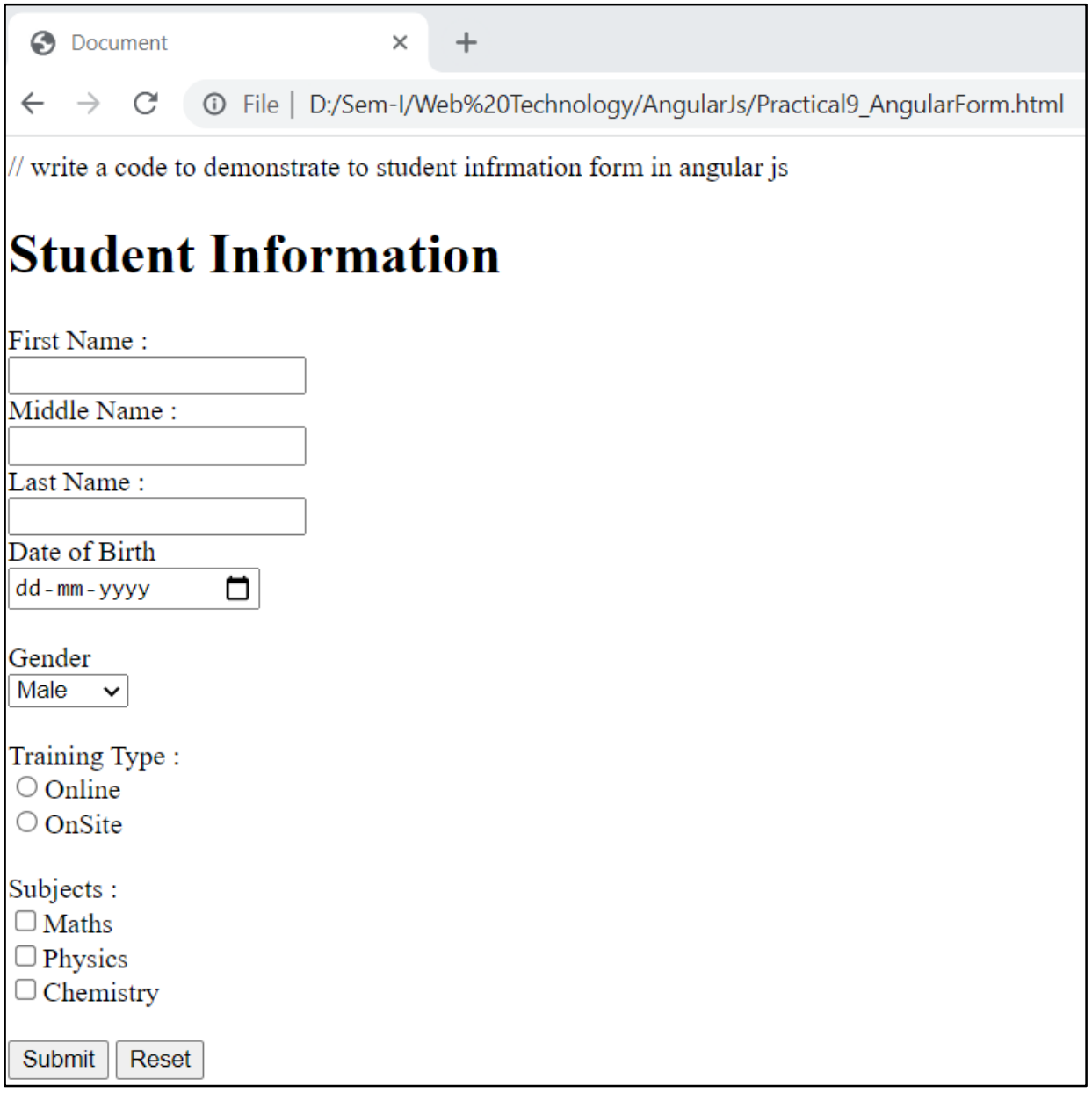
Line of code:

```
// Write a code to demonstrate to student infrmation form using angular js
<!DOCTYPE html>
<html ng-app="studentApp">
<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>
</head>
<body ng-controller="studentController">
    <h1>Student Information</h1>
    <form ng-submit="submitStudentForm()">
        <label for="firstName">First Name :</label><br>
        <input type="text" id="firstName" ng-model="student.firstName"><br>

        <label for="MiddleName">Middle Name :</label><br>
        <input type="text" id="middleName" ng-model="student.MiddleName"/><br>

        <label for="lastName">Last Name :</label><br>
        <input type="text" id="lastName" ng-model="student.MiddleName"/><br>


        <label for="dob">Date of Birth</label><br>

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

        <label for="gender">Gender</label><br>
        <select id="gender" ng-model="student.gender"><br><br>
<option value="male">Male</option>
<option value="female">Female</option>
        </select><br><br>

        <span>Training Type : </span><br>
        <label><input type="radio" value="online" name="training" ng-
model="studenttrainingType">Online</label><br>
        <label><input type="radio" value="onsite" name="training" ng-
model="studenttrainingType">OnSite</label><br><br>

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

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

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

    <script>
// creating module

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

        // creating controller

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

            $scope.originalStudent = {
                firstName: 'James',
                middleName: 'Carter',
                lastName: 'Bond',
                DoB: new Date('01/31/1980'),
                gender: 'male',
                trainingType: 'online',
                maths: false,
                physics: true,
                chemistry: true
            };

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

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

            // 5.  creating submitStudentForm() function this will call when
user submits form
            $scope.submitStudentForm = function() {

                var onSuccess = function(data, status, headers, config){
                    alert('Student saved successfully');
                };

                var onError = function (data, status, headers, config) {
                    alert('Error Occurred.');
                }

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

```
                    .error(onError);
            };

            $scope.resetForm = function() {
                $scope.student = angular.copy($scope.originalStudent);
            };
        });
    </script>

</body>
</html>
```

Execute Screen:



Practical No. 10

Aim: Write a code to Demonstrate the Student Info form using Bootstrap in Angular JS.

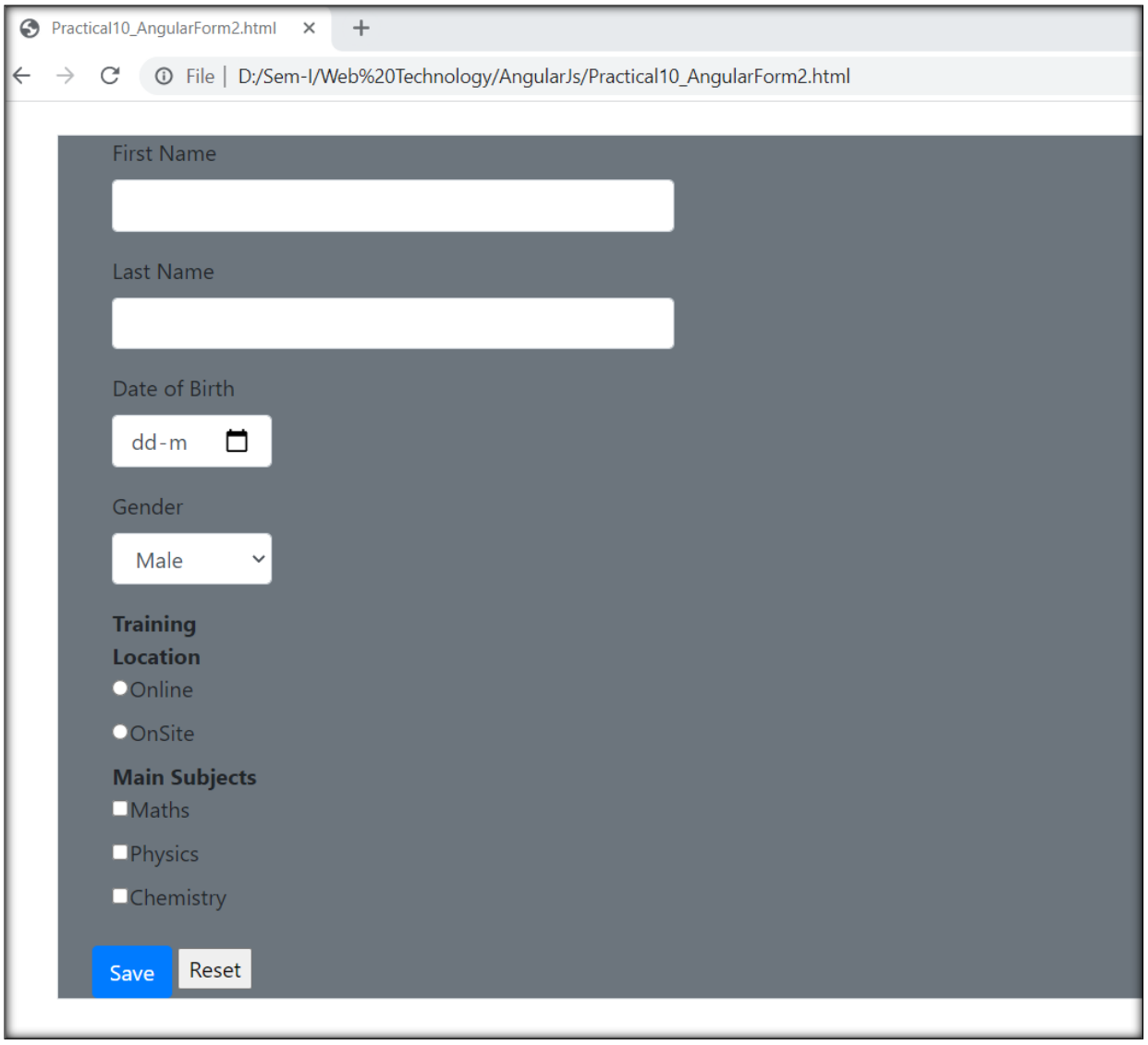
Line of code:

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
    <link
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet" />
    <script src="angular-min.js"></script>
</head>
<body ng-controller="studentController" class="container" > <br />
    <div class="container px-4 border bg-secondary wrap_text">
        <form class="form-horizontal" ng-submit="submitStudnetForm()"
role="form">
        <div class="form-group">
            <label for="firstName" class="col-sm-3 control-label">First
Name</label>
            <div class="col-sm-6">
                <input type="text" id="firstName" class="form-control" ng-
model="student.firstName" />
            </div>
            <div class="col-sm-3"></div>

        </div>
        <div class="form-group">
            <label for="lastName" class="col-sm-3 control-label">Last
Name</label>
            <div class="col-sm-6">
                <input type="text" id="lastName" class="form-control" ng-
model="student.lastName" />
            </div>
            <div class="col-sm-3"></div>
        </div>

        <div class="form-group">
            <label for="dob" class="col-sm-3 control-label">Date of
Birth</label>
            <div class="col-sm-2">
                <input type="date" id="dob" class="form-control" ng-
model="student.DoB" />
            </div>
            <div class="col-sm-7"></div>
        </div>

        <div class="form-group">
            <label for="gender" class="col-sm-3 control-label">Gender</label>
```

```
            <div class="col-sm-2">
                <select id="gender" class="form-control" ng-
model="student.gender">
                    <option value="male">Male</option>
                    <option value="female">Female</option>
                </select>
            </div>
            <div class="col-sm-7"></div>
        </div>

        <div class="form-group">
            <div class="col-sm-3"></div>
            <div class="col-sm-2">
                <span><b>Training Location</b></span>
                <div class="radio">
                    <label><input value="online" type="radio" name="training"
ng-model="student.trainingType" />Online</label>
                </div>
                <div class="radio">
                    <label><input value="onsite" type="radio" name="training"
ng-model="student.trainingType" />OnSite</label>
                </div>
            </div>
            <div class="col-sm-7">
                <span><b>Main Subjects</b></span>
                <div class="checkbox">
                    <label><input type="checkbox" ng-model="student.maths"
/>Maths</label>
                </div>
                <div class="checkbox">
                    <label><input type="checkbox" ng-model="student.physics"
/>Physics</label>
                </div>
                <div class="checkbox">
                    <label><input type="checkbox"  ng-model="student.chemistry"
/>Chemistry</label>
                </div>
            </div>
        </div>

        <input type="submit" value="Save" class="btn btn-primary col-sm-offset-
3" />
        <input type="reset" value="Reset" ng-click="resetForm()"
    </form>
</div>
    <script>
     //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',
                maths: false,
                physics: true,
                chemistry: true
            };

            //4. copy originalStudent to student. student will be bind to a
form
            $scope.student = angular.copy($scope.originalStudent);

            //5. create submitStudentForm() function. This will be called when
user submits the form
            $scope.submitStudnetForm = function () {

                var onSuccess = function (data, status, headers, config) {
                    alert('Student saved successfully.');
                };

                var onError = function (data, status, headers, config) {
                    alert('Error occured.');
                }

                $http.post('/student/submitData', { student:$scope.student })
                    .success(onSuccess)
                    .error(onError);

            };

            //6. create resetForm() function. This will be called on Reset
button click.
            $scope.resetForm = function () {
                $scope.student = angular.copy($scope.OriginalStudent);
            };
    });
    </script>
</body>
</html>
```

Execute Screen:



Practical No. 11

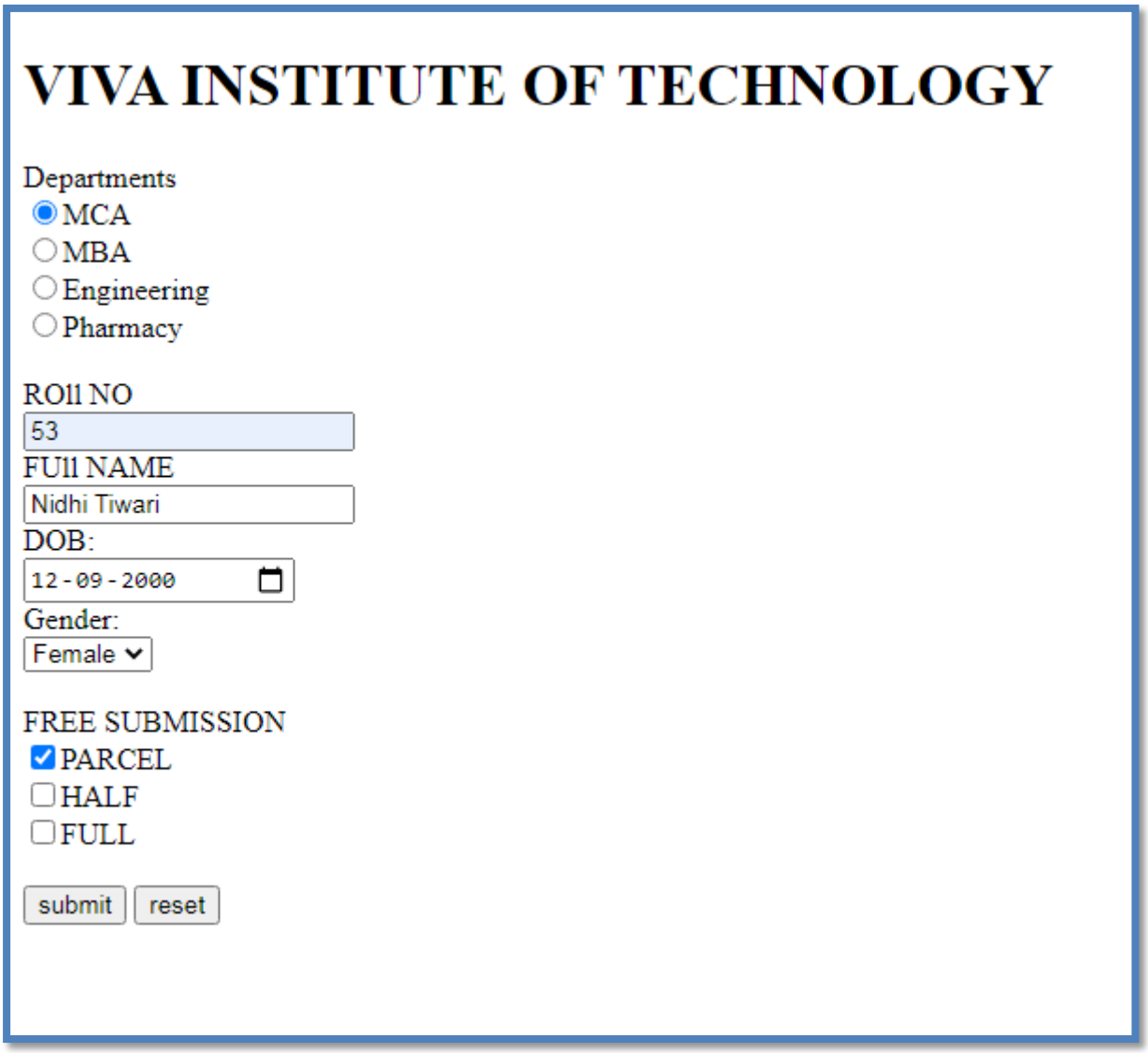
Aim: Write a code to design a form in bootstrap.

Code:

```
<!-- Write a code to design a form in bootstrap-->
<!DOCTYPE html>
<html lang="en">
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body  ng-controller="studentController">
<h1> VIVA INSTITUTE OF TECHNOLOGY </h1>
<form  ng-submit="submitStudentForm()">

<span> Departments </span><br />
<label><input value="MCA" type="radio" name="training" ng- model="student.trainingType"
/>MCA</label><br />
<label><input value="MBA"  type="radio" name="training" ng- model="student.trainingType"
/>MBA</label><br />
<label><input value="Engineering"  type="radio" name="training"  ng- model="student.trainingType"
/>Engineering</label><br />
<label><input value="Pharmacy" type="radio"  name="training" ng- model="student.trainingType"
/>Pharmacy</label><br /><br />

<label for="lastname"> ROll NO </label><br />
<input  type="lastname"  id="lastname"  ng-model="student.lastname"  /><br/>

<label for="lastname"> FUll NAME </label><br />
<input  type="lastname"  id="lastname"  ng-model="student.lastname"  /><br/>

<label for="dob">DOB: </label><br/>
<input  type="date"  id="dob"  ng-model="student.DOB"  /><br  />

<label for="gender">Gender: </label><br />
<select  id="gender"  ng-model="student.gender">
<option  value="male">Male  </option>
<option  value="female">Female  </option>
</select><br /><br />
<span>FREE  SUBMISSION</span><br  />
<label><input  type="checkbox"  ng-model="student.maths"
/>PARCEL</label><br  />
<label><input  type="checkbox"  ng-model="student.chemistry"
/>HALF</label><br  />
<label><input  type="checkbox"  ng-model="student.physics"
/>FULL</label><br /><br />
```

```
<input type="submit" value="submit" />
<input type="reset" ng-click="resetForm()" value="reset" />
</form>
<script>
//1.create a module
var studentApp = angular.module('studentApp',[]);

//2.create controller studentApp.controller("stdentController",function($scope,$http) {

//3.attach originalStudent model object
$scope.originalStudent = { firstname:'gitesh', middlename:'satish', lastname:'lad', DOB:'new
Date('03/12/2000'),
gender:'male', trainingType:'online', maths:false, chemistry:true, physics:true
};

//4.copy originalStudent to student. Student will be bind to the form
$scope.student = angular.copy($scope.originalStudent);

//5.create submitStudentForm() function. This will be called when student submits the form
$scope.submitStudentForm = function () {

var onSuccess = function (date, status, header, config) { alert('Student savedsuccessfully.');
};

var onError = function (date, status, header, 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 buttonclick.
$scope.resetForm = function () {
$scope.student = angular.copy($scope.originalStudent);
};

});

</script>
</body>
</html>
```

Output:


VIVA INSTITUTE OF TECHNOLOGY
Departments
MCA
MBA
Engineering
Pharmacy
ROll NO
53
FUll NAME
Nidhi Tiwari
DOB:
12-09-2000
Gender:
Female
FREE SUBMISSION
PARCEL
HALF
FULL
submit
reset

Practical No. 12

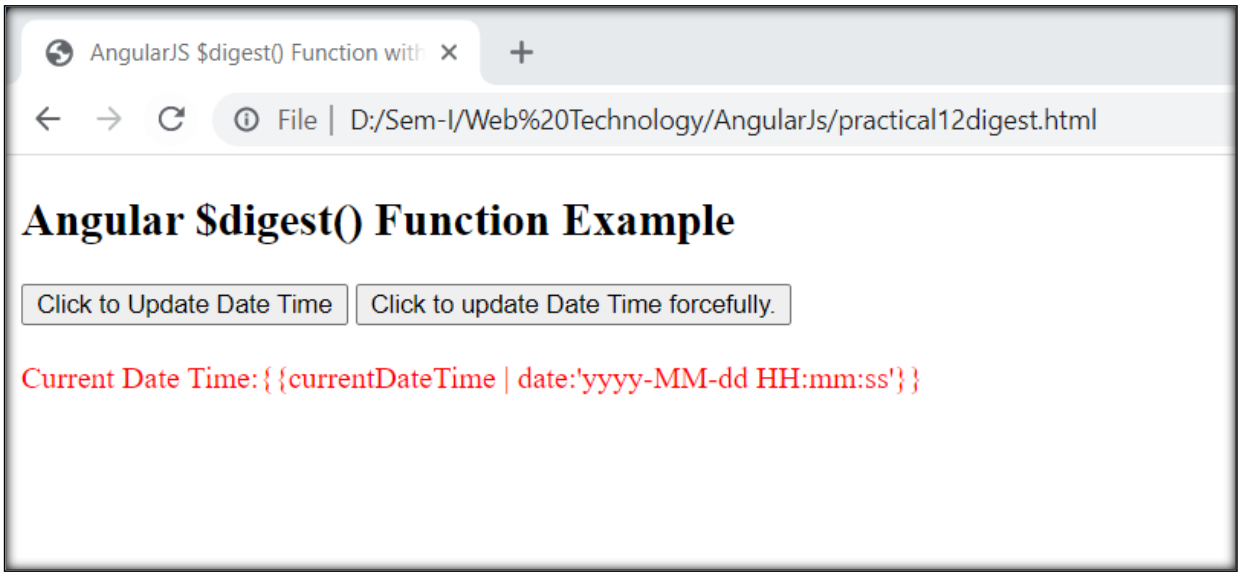
Aim: Write an Angular JS script to demonstrate $digest () function.

Line of code:

```
<!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>AngularJS $digest() Function with Example</title>
    <script src="angular.min.js"></script>

        <script type="text/javascript">
        var app = angular.module('digestApp', []);
        app.controller('digestCtrl', function($scope)
        {
        $scope.currentDateTime = new Date();
        $scope.updatedtime = function()
        {
            $scope.currentDateTime = new Date()
        }

        // added an event listener
        var event = document.getElementById("btndigest");
        event.addEventListener('click', function()
        {
            // get dateTime
            $scope.currentDateTime = new Date()

            // the digest method is use to update date time forcefully.
            $scope.$digest();
        });

        });

</script>

</head>
<body>
    <div ng-app="digestApp" ng-controller="digestCtrl">

        <h2>Angular $digest() Function Example</h2>
        <input type="button" value="Click to Update Date Time" ng-
click="updatedtime()">
        <input type="button" value="Click to update Date Time forcefully."
id="btndigest">
```

```
        <br><br>
        <span style="color: Red">Current Date Time:{{currentDateTime |
date:'yyyy-MM-dd HH:mm:ss'}}</span>
    </div>
</body>
</html>
```

Execute Screen:



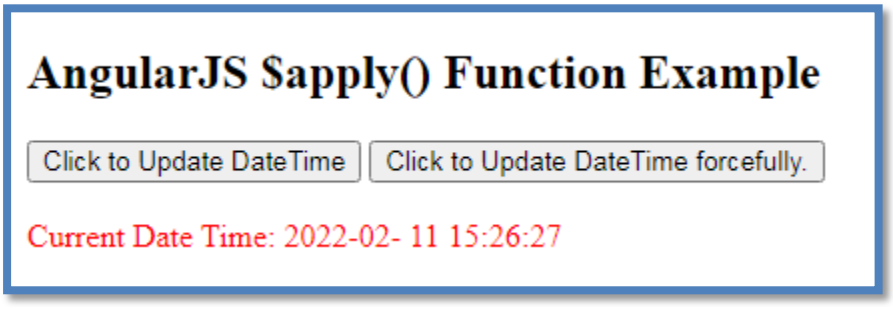
Practical No. 13

Aim: Write a code to demonstrate Arithmetic Operators using Angular JS.

Code:

```
<!DOCTYPE html>
<html>
<head>
<title>AngularJs $apply() Function with Example</title>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"
></script>
<script type="text/javascript">
var app = angular.module('applyApp', []); app.controller('applyCtrl', function ($scope) {
$scope.currentDateTime = new Date();
$scope.updatedtime = function () {
$scope.currentDateTime = new Date();
}
var event = document.getElementById("btnapply"); event.addEventListener('click',function () {
$scope.$apply(function () {
$scope.currentDateTime = new Date();
});
});
});
</script>
</head>
<body>
<div ng-app="applyApp" ng-controller="applyCtrl">
<h2>AngularJS $apply() Function Example</h2>
<input type="button" value="Click to Update DateTime" ng-click="updatedtime()"
/>
<input type="button" value="Click to Update DateTime forcefully." id="btnapply" />
<br /><br />
<span style="color:Red">Current Date Time: {{currentDateTime | date:'yyyy-MM- dd
HH:mm:ss'}}</span>
</div>
</body>
</html>
```

Output:



Name: Deepesh Mangesh Mhatre

App No: 81389

Practical No. 14

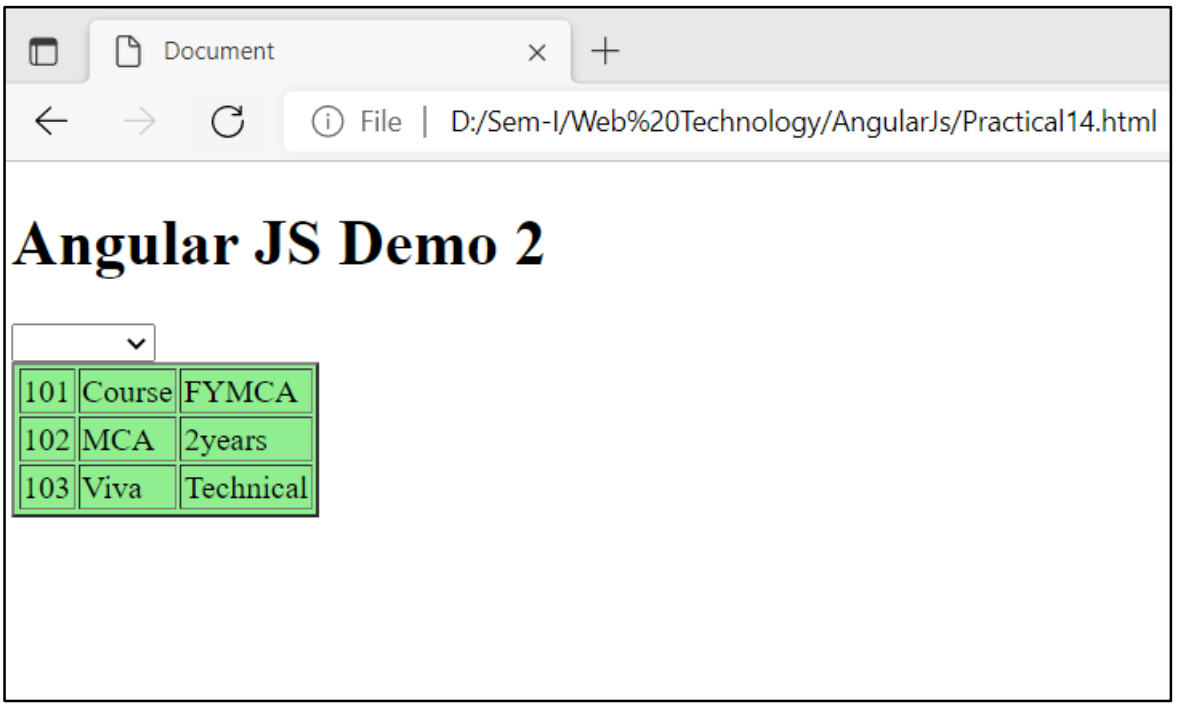
Aim: Create a webpage to change the background color (bgcolor) of the table dynamically using Angular JS.

Line of code:

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

</head>
<body>
    <h1>Angular JS Demo 2</h1>
    <div ng-app="" ng-init="x='lightgreen'" >
        <select color name ng-model="x">
            <option>Pink</option>
            <option>light red</option>
            <option>Red</option>
            <option>Aqua</option>

        </select>
        <br>
        <table bgcolor="{{x}}" width ="80xp" Border="2">
            <tr>
                <td>101</td>
                <td>Course</td>
                <td>FYMCA</td>
            </tr>
            <tr>
                <td>102</td>
                <td>MCA</td>
                <td>2years</td>
            </tr>
            <tr>
                <td>103</td>
                <td>Viva</td>
                <td>Technical</td>
            </tr>

        </table>
```

```
    </div>
</body>
</html>
```

Execute Screen:



Practical No. 15

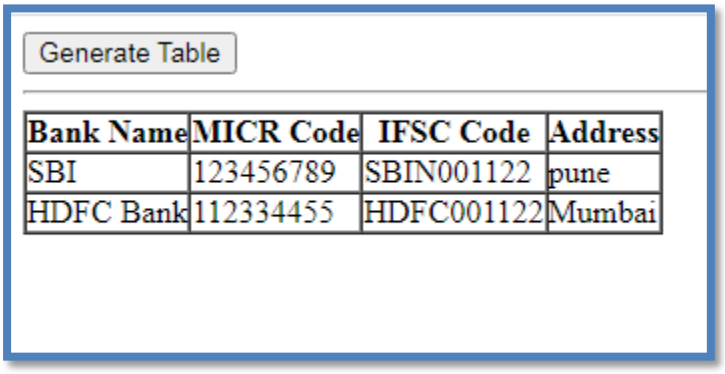
Aim: Create a web page Demonstration of NG repeat on Angular JS.

Code:

```
<!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>
</head>
<body>
<script text="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
<script text="text/javascript">
var app = angular.module('MyApp',[]) app.controller('MyController', function
($scope)
{
$scope.IsVisible = false;
$scope.GenerateTable = function()
{
$scope.bankdetail = [
{Bankname: "SBI", MICRcode: 123456789, IFSCCode:"SBIN001122", address: "pune" },
{Bankname: "HDFC Bank", MICRcode: 112334455, IFSCCode:"HDFC001122", address:"Mumbai" }
];
$scope.IsVisible = true;
};
});
</script>
<div ng-app="MyApp" ng-controller="MyController">
```

```
<input type="button" value="Generate Table" ng-click="GenerateTable()"
/>
<hr />
<table cellpadding="0" cellspacing="0" ng-show="IsVisible" border = 1
>
<tr>
<th>Bank Name</th>
<th>MICR Code</th>
<th>IFSC Code</th>
<th>Address</th>
</tr>
<tbody ng-repeat="m in bankdetail">
<tr>
<td>{{m.Bankname}}</td>
<td>{{m.MICRcode}}</td>
<td>{{m.IFSCCode}}</td>
<td>{{m.address}}</td>
</tr>
</tbody>
</table>
</div>
</body>
</html>
```

Output:



Practical No. 16

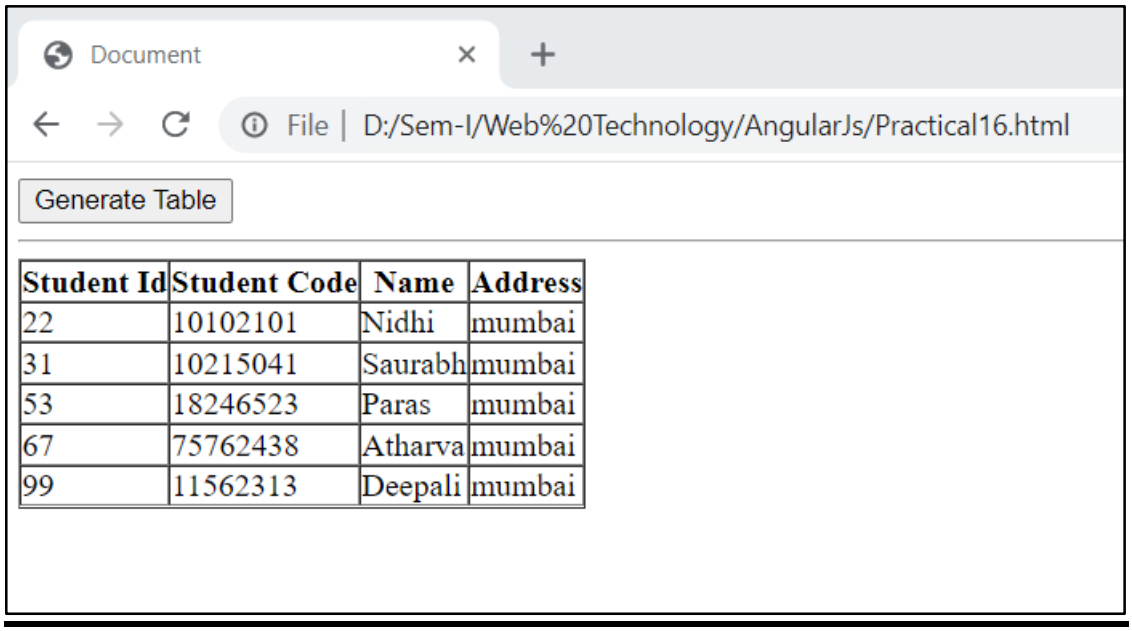
Aim: Write a script to design a table using controller ng-repeat and different HTML tags in Angular JS.

Line of code:

```
<!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>
</head>
<body>
    <script text="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></sc
ript>
    <script text="text/javascript">
        var app = angular.module('MyApp',[])
        app.controller('MyController', function ($scope)
        {
            $scope.IsVisible = false;
            $scope.GenerateTable = function()
            {
                $scope.Studentdetail = [
                    {StudentId: 22, StudentCode: "10102101", Name:"Nidhi",
Address: "mumbai" },
                    {StudentId: 31, StudentCode: "10215041", Name:"Saurabh",
Address: "mumbai" },
                    {StudentId: 53, StudentCode: "18246523", Name:"Paras",
Address: "mumbai" },
                    {StudentId: 67, StudentCode: "75762438", Name:"Atharva",
Address: "mumbai" },
                    {StudentId: 99, StudentCode: "11562313", Name:"Deepali",
Address: "mumbai" },
                ];
            $scope.IsVisible = true;
            };
        });
    </script>
    <div ng-app="MyApp" ng-controller="MyController">
        <input type="button" value="Generate Table" ng-click="GenerateTable()"
/>
        <hr />
        <table cellpadding="0" cellspacing="0" ng-show="IsVisible" border = 1>
            <tr>
                <th>Student Id </th>
                <th>Student Code </th>
```

```
                <th>Name </th>
                <th>Address</th>
            </tr>
            <tbody ng-repeat="s in Studentdetail">
                <tr>
                    <td>{{s.StudentId}}</td>
                    <td>{{s.StudentCode}}</td>
                    <td>{{s.Name}}</td>
                    <td>{{s.Address}}</td>
                </tr>
            </tbody>
        </table>
    </div>
</body>
</html>
```

Execute Screen:



Student Id	Student Code	Name	Address
22	10102101	Nidhi	mumbai
31	10215041	Saurabh	mumbai
53	18246523	Paras	mumbai
67	75762438	Atharva	mumbai
99	11562313	Deepali	mumbai

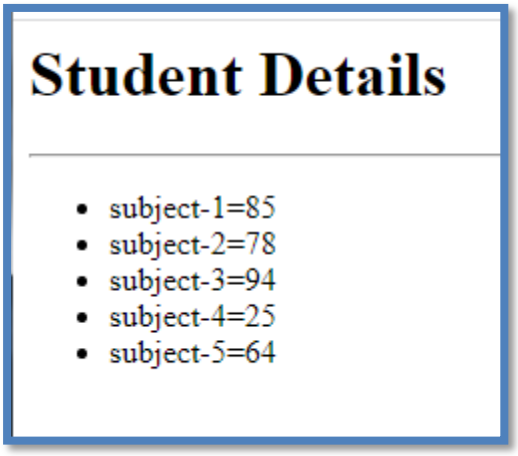
Practical No. 17

Aim: Write a webpage to process student marks using angular.js controller

Code:

```
<html>
<head>
<script     src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"
></script>
<script>
var obj=angular.module("myApp",[]); obj.controller("studentcontroller",function($scope)
{
$scope.marks=[85,78,94,25,64];
});
</script>
</head>
<body ng-app="myApp">
<h1>Student  Details</h1>
<hr>
<div  ng-controller="studentcontroller">
<ul>
<li  ng-repeat=  "item  in  marks"> subject-{{$index+1}}={{item}}
</li>
</ul>
</div>
</body>
</html>
```

Output:



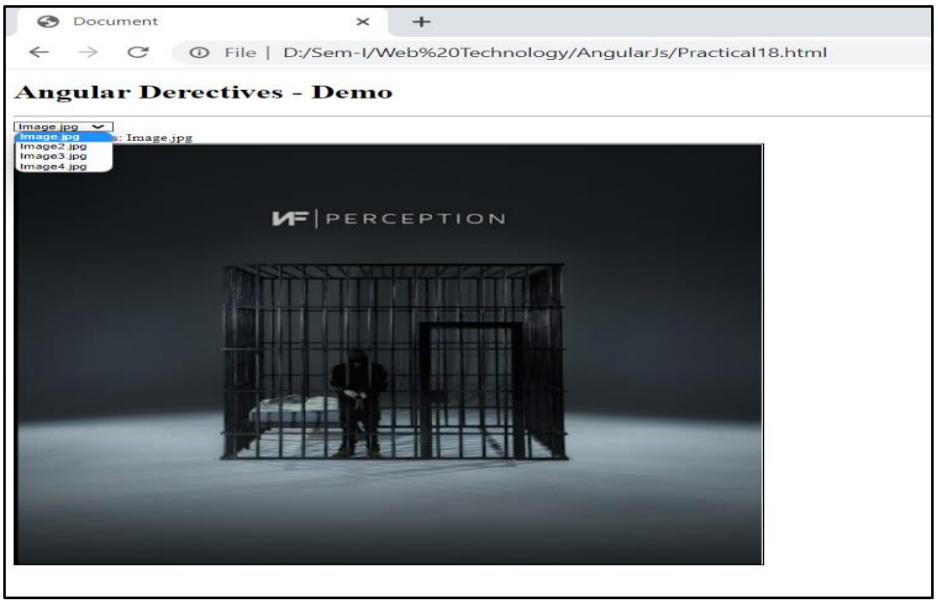
Practical No. 18

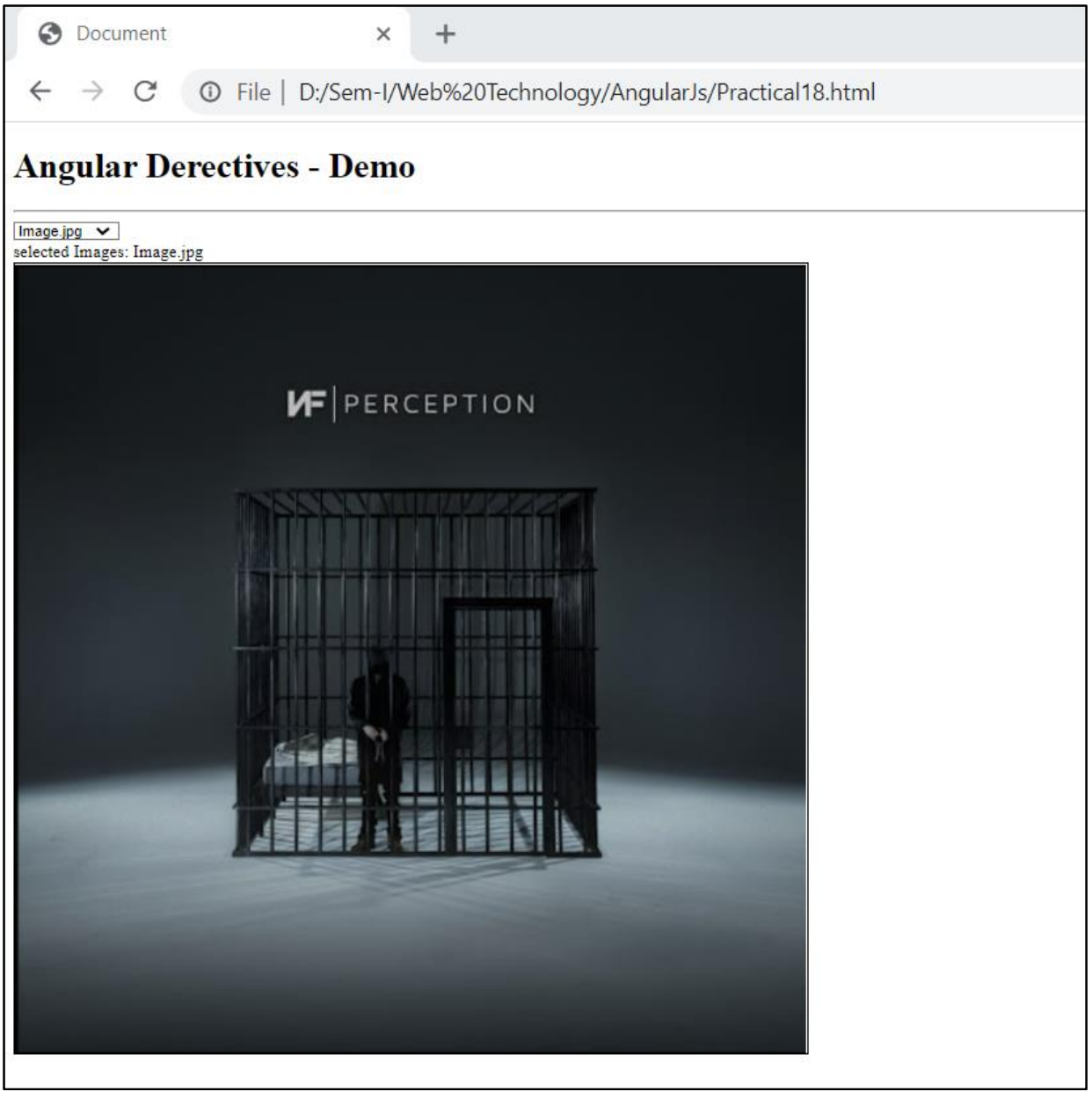
Aim: Create a web page to fetch multiple images and change images dynamically using AngularJS.

Line of code:

```
<!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="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"
></script>
</head>
<body ng-app="" ng-init="fname='Image.jpg'">
    <h1>Angular Derectives - Demo</h1>
    <hr></hr>
    <select ng-model="fname">
        <option>Image.jpg</option>
        <option>Image2.jpg</option>
        <option>Image3.jpg</option>
        <option>Image4.jpg</option>
    </select> </br>
    <span>selected Images: {{fname}} </span>
    <br>
    <img border="2" height="723px" width="726px" src ="{{fname}}" />
</body>
</html>
```

Execute Screen:





Name: Deepesh Mangesh Mhatre

App No: 81389

Practical No. 19

Aim: Create a webpage to process product details using angular JS controller organizedproducts data using an array of objects.

Code:

```
<!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="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
      <script>
         var obj=angular.module("myApp",[]); obj.controller("Productcontroller",function($scope)
         {
            $scope.product=[
               {pid:14589,pname:"Nicon  Camera",uprice:18200,  image:"1.jpg"},

               {pid:14589,pname:"Canon  Camera",uprice:18200,  image:"2.jpg"},

               {pid:14589,pname:"Samsung  Camera",uprice:18200,  image:"3.jpg"},

               {pid:14589,pname:"Sony  Camera",uprice:18200,  image:"4.jpg"
               }];
         });
      </script>
   </head>
   <body ng-app="myApp">
      <h1>Product  Details</h1>
      <hr>
      <div  ng-controller="Productcontroller">
        <div style="text-align: center; float: left; padding: 3px; margin: 5px; border: 2pxsolid blue;
height: 150px; width: 200px;" ng-repeat="item in product">
      <span><u>{{item.pname}}</u></span><br>
      <img  border="1"  height="90"  width="150"  src="{{item.image}}">
      <br>
      Product ID: {{item.pid}} <br> Price:<b>INR{{item.uprice}}.00</b>
         </div>
        </div>
      </body>
</html>
```

Output:


Product Details
Nicon Camera
Product ID: 14589
Price:INR18200.00
Canon Camera
Love Nature
Product ID: 14589
Price:INR18200.00
Samsung Camera
Product ID: 14589
Price:INR18200.00
Sony Camera
Product ID: 14589
Price:INR18200.00

MCAL14: Web Technologies Lab

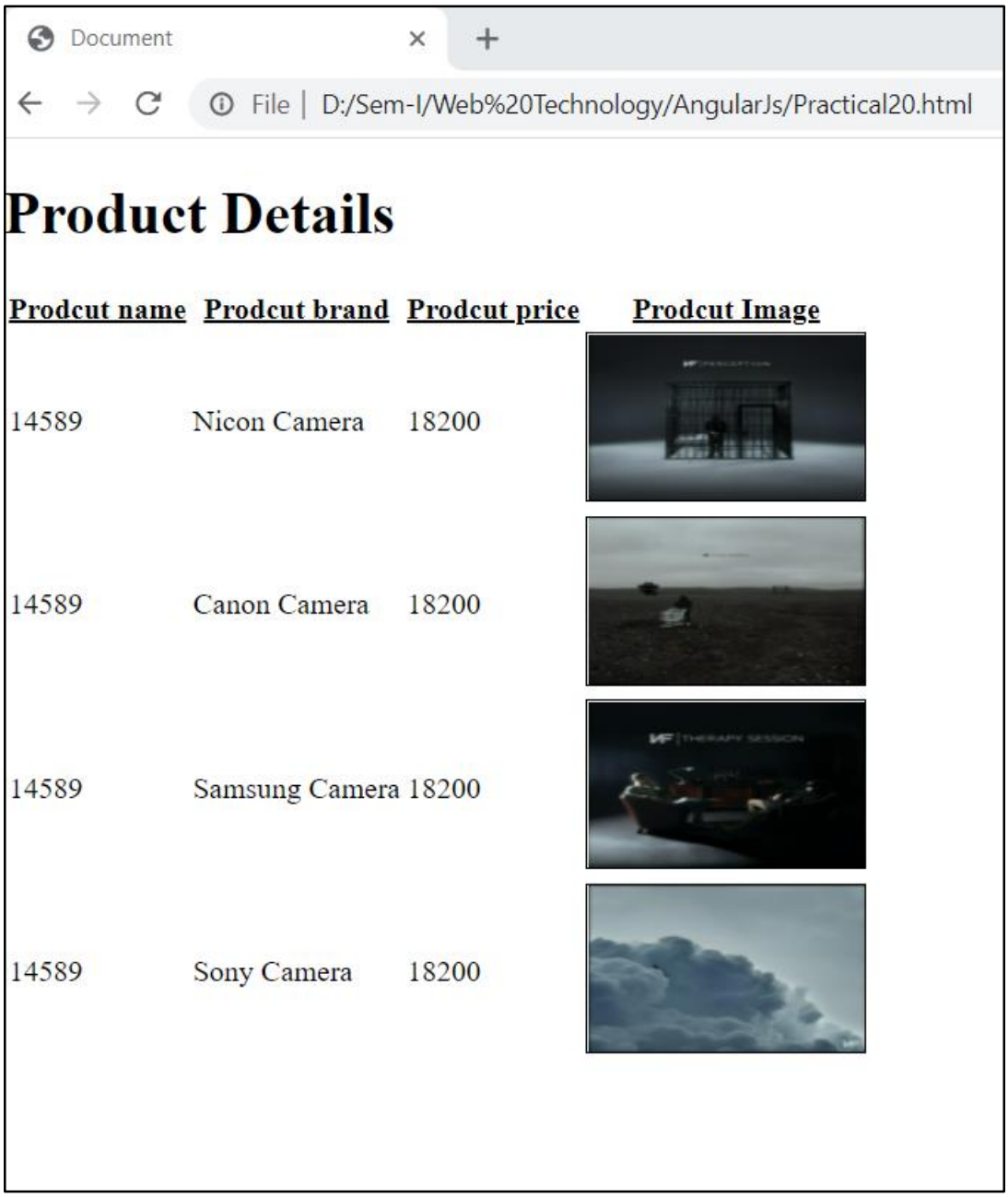
Practical No. 20

Aim: Create a web page to update the previous example to display the product details in table format.

Line of code:

```
<!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="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
        <script>
            var obj=angular.module("myApp",[]);
            obj.controller("Productcontroller",function($scope)
            {
                $scope.product=[
                    {pid:14589,pname:"Nicon Camera",uprice:18200, image:"image.jpg"},

                    {pid:14589,pname:"Canon Camera",uprice:18200, image:"image2.jpg"},

                    {pid:14589,pname:"Samsung Camera",uprice:18200, image:"image3.jpg"},

                    {pid:14589,pname:"Sony Camera",uprice:18200, image:"image4.jpg"
                    }];
            });
        </script>
    </head>
    <body ng-app="myApp">
        <h1>Product Details</h1>
        <table ng-controller="Productcontroller">
            <tr>
                <th><u>Prodcut name</u></th>
                <th><u>Prodcut brand</u></th>
                <th><u>Prodcut price</u></th>
                <th><u>Prodcut Image</u></th>
            </tr>
            <tr ng-repeat="item in product">
                <td>{{item.pid}}</td>
                <td>{{item.pname}}</td>
                <td>{{item.uprice}}</td>
                <td><img border="1" height="90" width="150" src="{{item.image}}"></td>

            </tr>

        </table>
    </body>
</html>
```

Execute Screen:



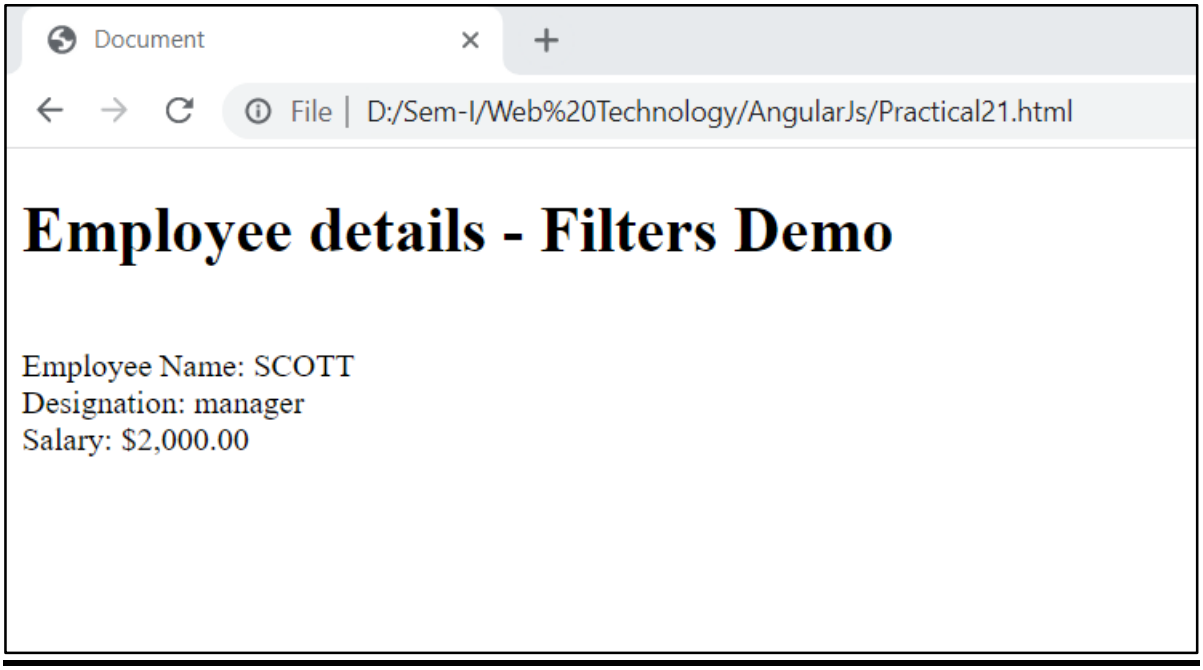
Practical No. 21

Aim: Create a webpage to demonstrate the wege of filters in AngularJS.

Line of code:

```
<!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="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
    <script>
        var obj = angular.module("myApp",[]);
        obj.controller("EmpController",function($scope){
            $scope.ename = "scott";
            $scope.job = "Manager";
            $scope.sal = 2000;
        });
    </script>
</head>
<body ng-app="myApp">
    <h1>Employee details - Filters Demo</h1>
    </br>
    <div ng-controller = "EmpController">
        <span>
            Employee Name: {{ename|uppercase}}<br>
            Designation: {{job|lowercase}}<br>
            Salary: {{sal|currency}}
        </span>
    </div>
</body>
</html>
```

Execute Screen:



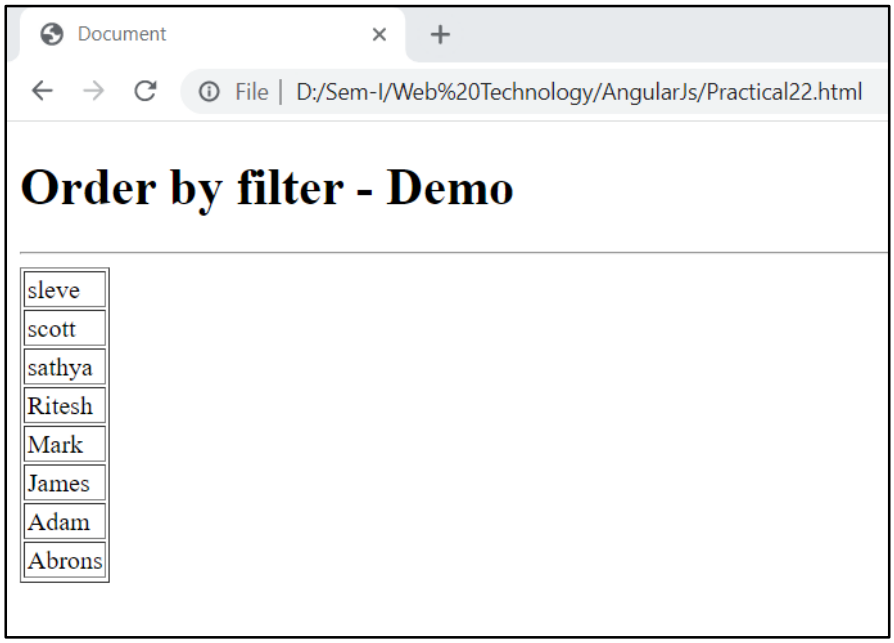
Practical No. 22

Aim: Create a webpage to demonstrate usage of order by the filter in AngularJS.

Line of code:

```
<!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 text="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
    <script>
        var obj = angular.module('myApp',[]);
        obj.controller('OrderbyController', function($scope){
            $scope.names =
["scott","sleve","sathya","Abrons","Mark","James","Ritesh","Adam"];
        });
    </script>
</head>
<body ng-app="myApp">
    <h1>Order by filter - Demo</h1>
    <hr>
    <table border="width=100%" ng-controller="OrderbyController">
        <tr ng-repeat="item in names | orderBy:item:true">
            <td>{{ item }}</td>
        </tr>
    </table>
</body>
</html>
```

Execute Screen:



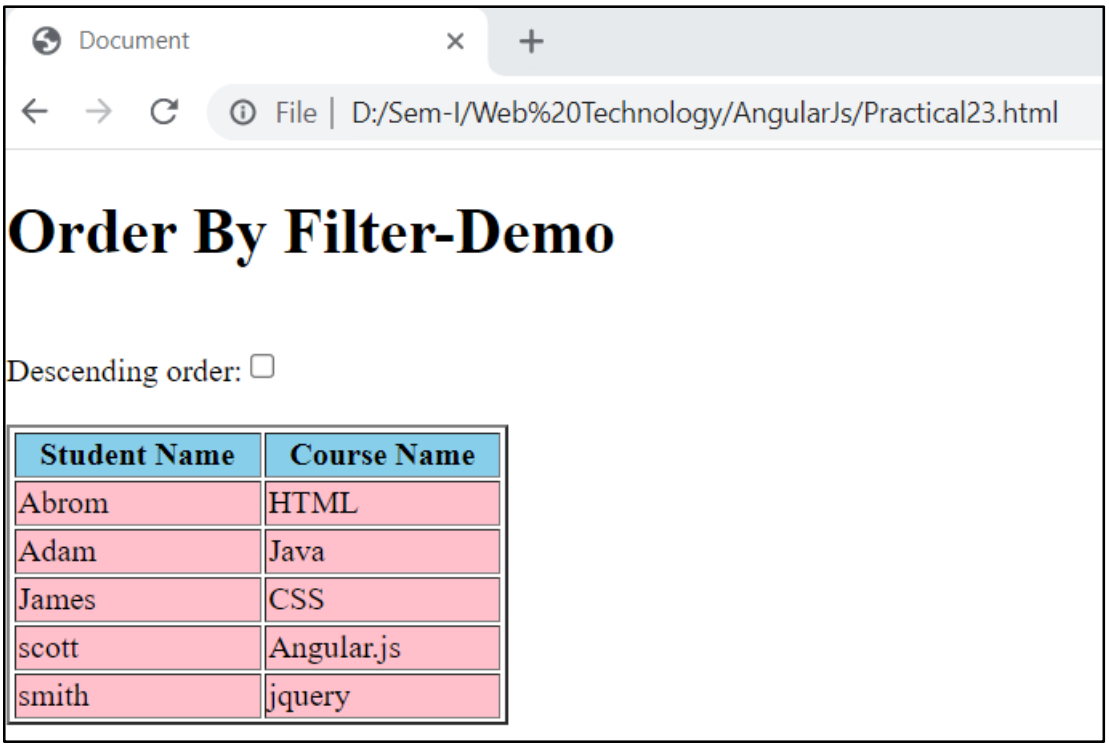
Practical No. 23

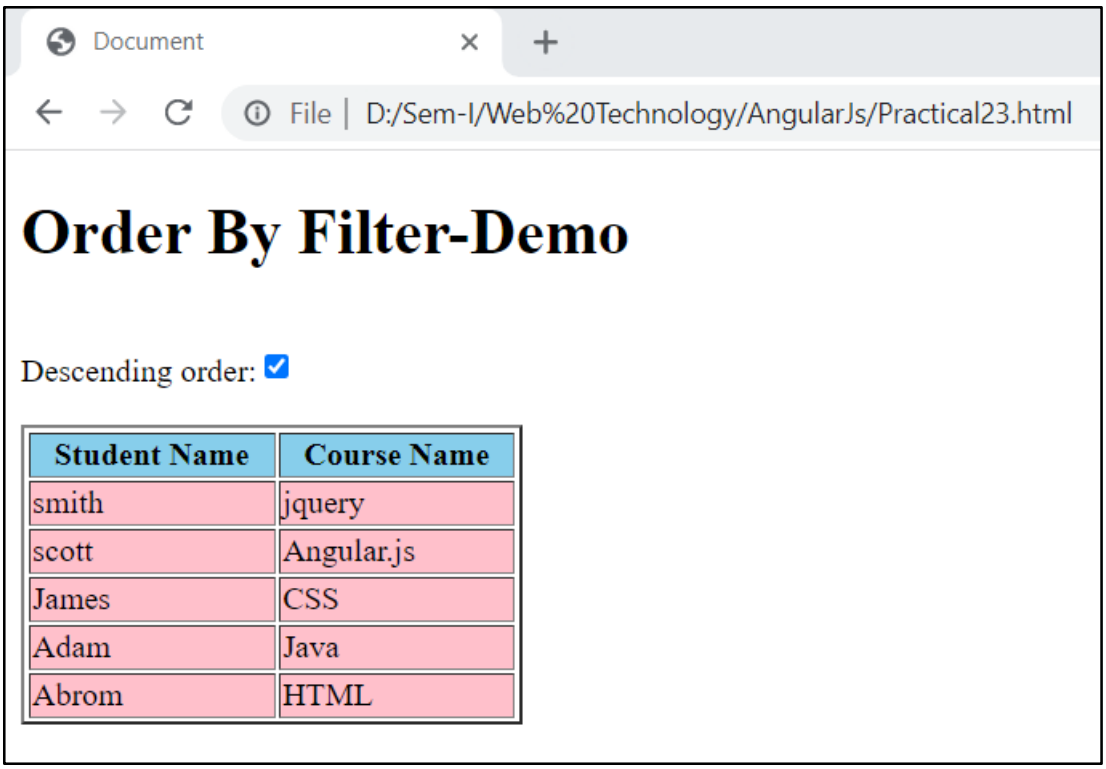
Aim: Create a webpage to sort the student details using order by the filter in Angular JS.

Line of code:

```
<!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="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
    <script>
        var obj = angular.module("myApp",[]);
        obj.controller("Democontroller",function($scope){
            $scope.student=[
                {sname: "scott",course:"Angular.js"},
                {sname: "Abrom",course:"HTML"},
                {sname: "smith",course:"jquery"},
                {sname: "James",course:"CSS"},
                {sname: "Adam",course:"Java"}
            ];
            $scope.x=false;
        });
    </script>
</head>
<body ng-app="myApp">
    <h1>Order By Filter-Demo</h1><br>
    <div ng-controller="Democontroller">
        Descending order:<input type="checkBox" ng-model="x"/><br></br>
        <table border="2" width="250px">
            <tr bgcolor="skyBlue">
                <th>Student Name</th>
                <th>Course Name</th>
            </tr>
            <tr bgcolor="pink" ng-repeat="item in student |orderBy:'sname':x">
                <td>{{item.sname}}</td>
                <td>{{item.course}}</td>
            </tr>
        </table>
    </div>

</body>
</html>
```

Execute Screen:





MCAL14: Web Technologies Lab

Practical No. 24

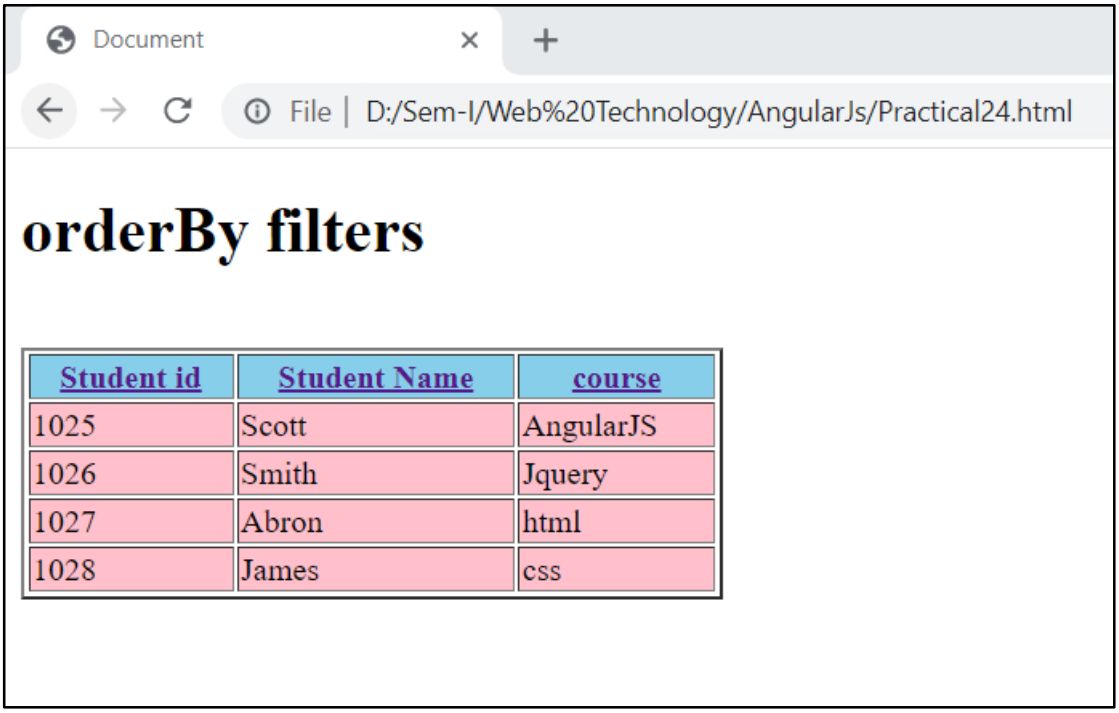
Aim: Create a webpage to sort the student detail based on the Selected column using order by the filter in Angular JS.

Line of code:

```
<!-- Create a webpage to sort the student details based on selected column using orderBy
filter -->
<!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="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
    <script>
        var obj=angular.module("myApp",[]);
            obj.controller("Democontroller",function($scope)
            {
                $scope.student=[
                    {sid:1025,sname:"Scott",course:"AngularJS"},
                    {sid:1026,sname:"Smith",course:"Jquery"},
                    {sid:1027,sname:"Abron",course:"html"},
                    {sid:1028,sname:"James",course:"css"}];
                    $scope.x=false;
                    $scope.y="sid";

            });
    </script>
</head>
<body ng-app="myApp">
    <h1>orderBy filters</h1></br>
    <div ng-controller="Democontroller">
        <table border="2" width="350px">
            <tr bgcolor="skyblue">
                <th><a href="" ng-click="x=|x:y='sid'">Student id</a></th>
                <th><a href="" ng-click="x=|x:y='sname'">Student Name</a></th>
                <th><a href="" ng-click="x=|x:y='course'">course</a></th>
            </tr>
            <tr bgcolor="pink" ng-repeat="item in student | orderBy:y:x">
                <td>{{item.sid}}</td>
                <td>{{item.sname}}</td>
                <td>{{item.course}}</td>
            </tr>
        </table>
    </div>

</body>
```

```
</html>
```

Execute Screen:



Practical No. 25

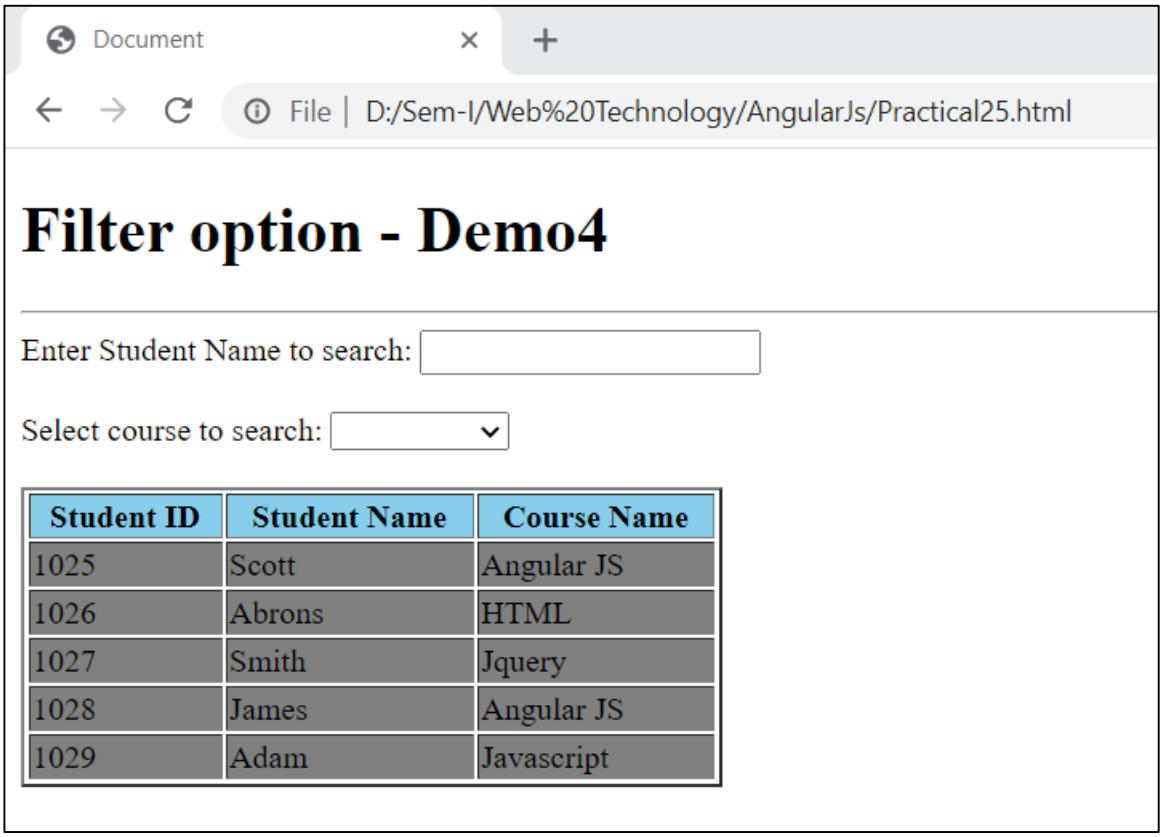
Aim: Create a webpage to apply searching on student details by using the filter option of Angular JS.

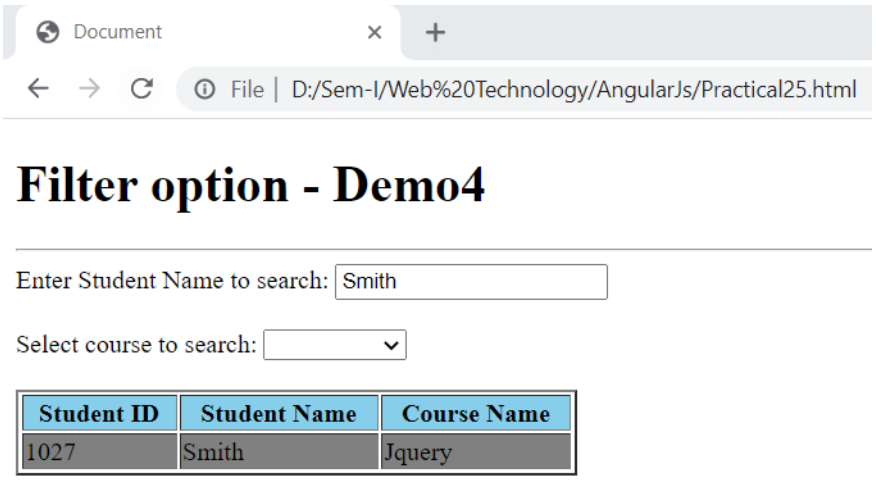
Line of code:

```
<!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="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"
></script>
    <script>
        var obj=angular.module("myApp",[]);
        obj.controller("Democontroller",function($scope)
        {
            $scope.students=[
            {sid:1025,sname:"Scott",course:"Angular JS"},
            {sid:1026,sname:"Abrons",course:"HTML"},
            {sid:1027,sname:"Smith",course:"Jquery"},
            {sid:1028,sname:"James",course:"Angular JS"},
            {sid:1029,sname:"Adam",course:"Javascript"}
            ];
            $scope.s1="";
            $scope.s2="";
        });
    </script>
</head>
<body ng-app="myApp">
    <h1>Filter option - Demo4 </h1>
    <hr>
    <div ng-controller="Democontroller">
        Enter Student Name to search:
        <input type="text" ng-model="s1"/><br><br>
        Select course to search:
        <select ng-model="s2">
            <option>Angular JS</option>
            <option>Jquery</option>
            <option>HTML</option>
            <option>Javascript</option>
            <option value=" ">All courses</option>
        </select><br><br>
        <table border="2" width="350px">
            <tr bgcolor="SkyBlue">
                <th>Student ID</th>
                <th>Student Name</th>
                <th>Course Name</th>
            </tr>
```

```
            <tr bgcolor="Grey" ng-repeat="item in students| filter :
{'sname':s1,'course':s2}">
                <td>{{item.sid}}</td>
                <td>{{item.sname}}</td>
                <td>{{item.course}}</td>
            </tr>
        </table>

    </div>

</body>
</html>
```

Execute Screen:





Practical No. 26

Aim: Create a webpage to organize product data so that it is always the user to search & sort based on the given scenarios.

Line of code:

```
<!--update the previous details to display in the following format-->

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

      <script>

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

         obj.controller("Productcontroller",function($scope)

         {

            $scope.product=[

            {pname:"Dell Laptop",brand:"Dell",uprice:"85000", image:"Image.jpg"},


            {pname:"Sony Laptop",brand:"Sony",uprice:"56000", image:"Image1.jpg"},


            {pname:"Lenevo Laptop",brand:"Lenevo",uprice:"75000", image:"Image2.jpg"}];

         });

      </script>

   </head>
```

```
<body ng-app="myApp">
    <h1>Product Details</h1>
    <table ng-controller="Productcontroller">
        <tr>
            <th><u>Prodcut name</u></th>
            <th><u>Brand</u></th>
            <th><u>Unit price</u></th>
            <th><u>Picture Image</u></th>
        </tr>
        <tr ng-repeat="item in product">
            <td>{{item.pname}}</td>
            <td>{{item.brand}}</td>
            <td>{{item.uprice}}</td>
            <td><img border="1" height="90" width="150" src="{{item.image}}"></td>
        </tr>

    </table>
</body>
</html>
```

Execute Screen:

Product Details

Prodcut name	Brand	Unit price	Picture Image
Dell Laptop	Dell	85000	
Sony Laptop	Sony	56000	
Lenevo Laptop	Lenevo	75000	

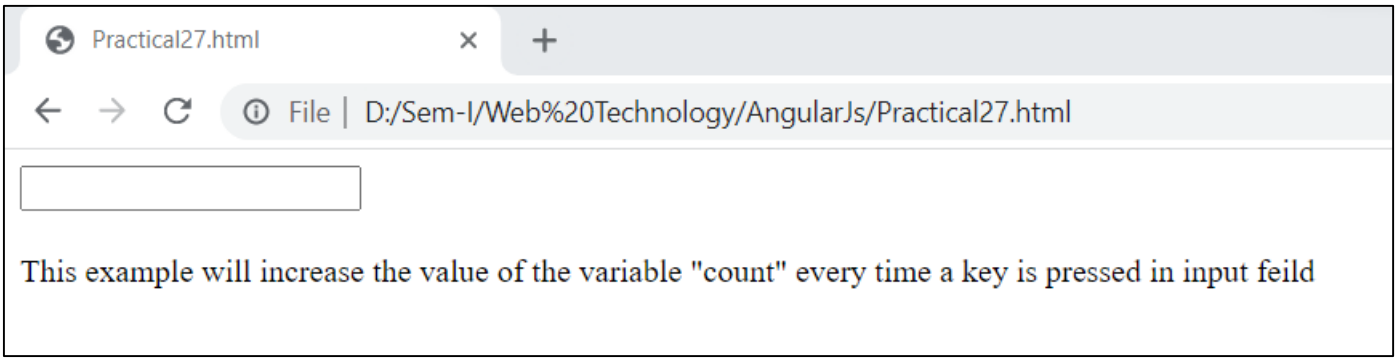
Practical No. 27.

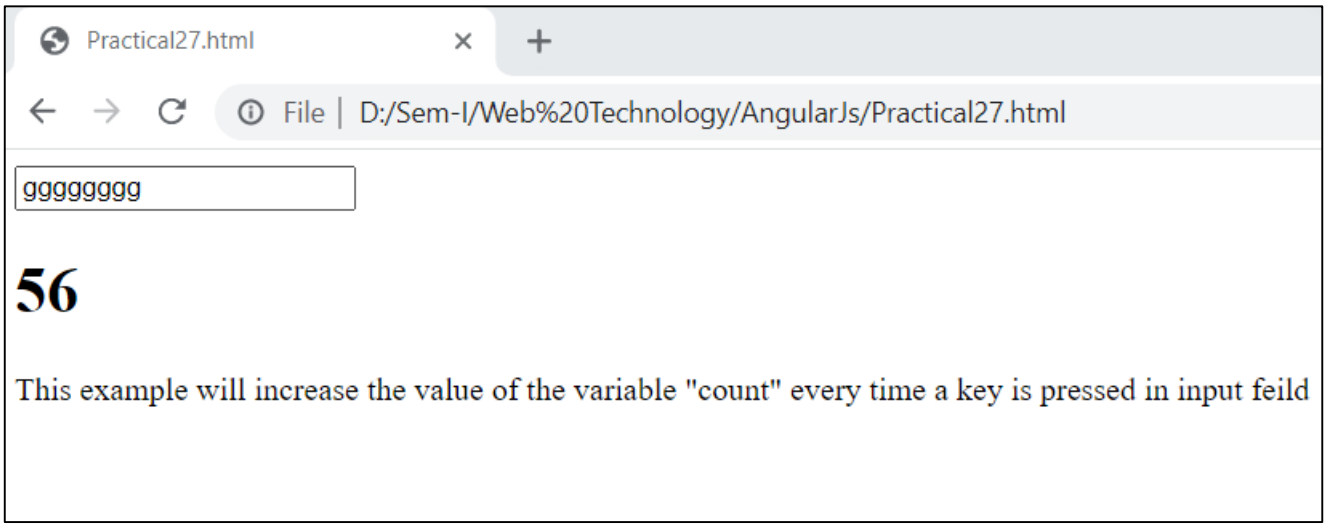
Aim: Create a web page to perform Key Event Directives “ng-key down” in Angular JS.

Line of code:

```
<!DOCTYPE html>
<html>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"
></script>
    <body ng-app="">
        <input ng-keydown="count = count + 1" ng-init="count-0" />
        <h1>{{count}}</h1>
        <p>This example will increase the value of the variable "count" every time a key is
pressed in input feild</p>
</body>
</html>
```

Execute Screen:





MCAL14: Web Technologies Lab

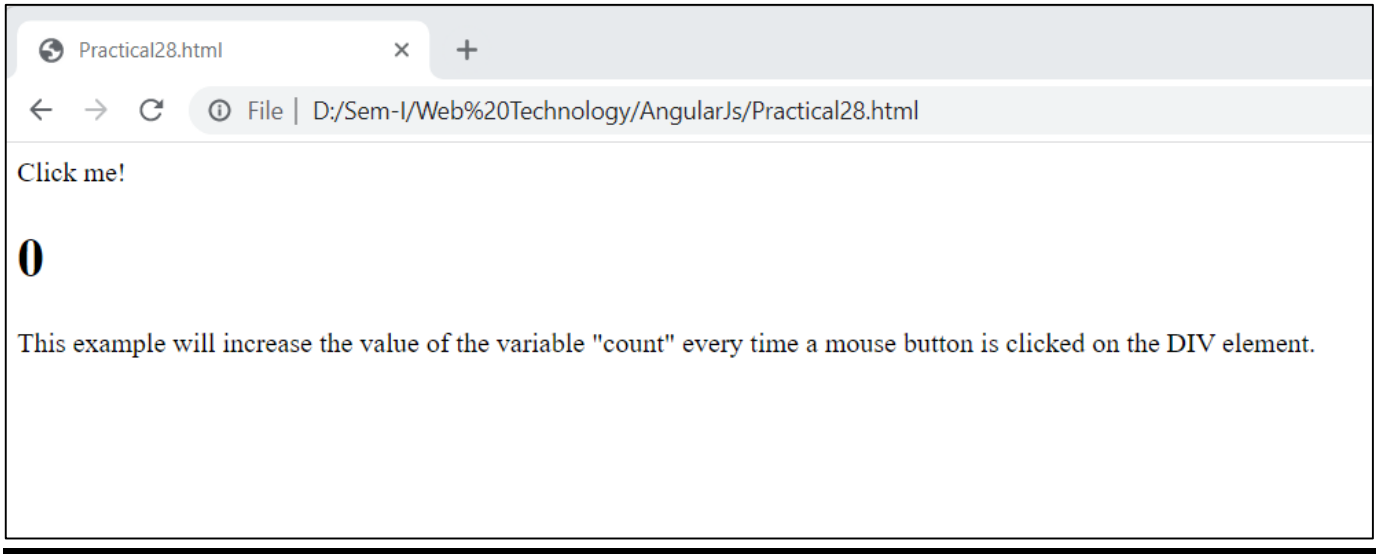
Practical No. 28

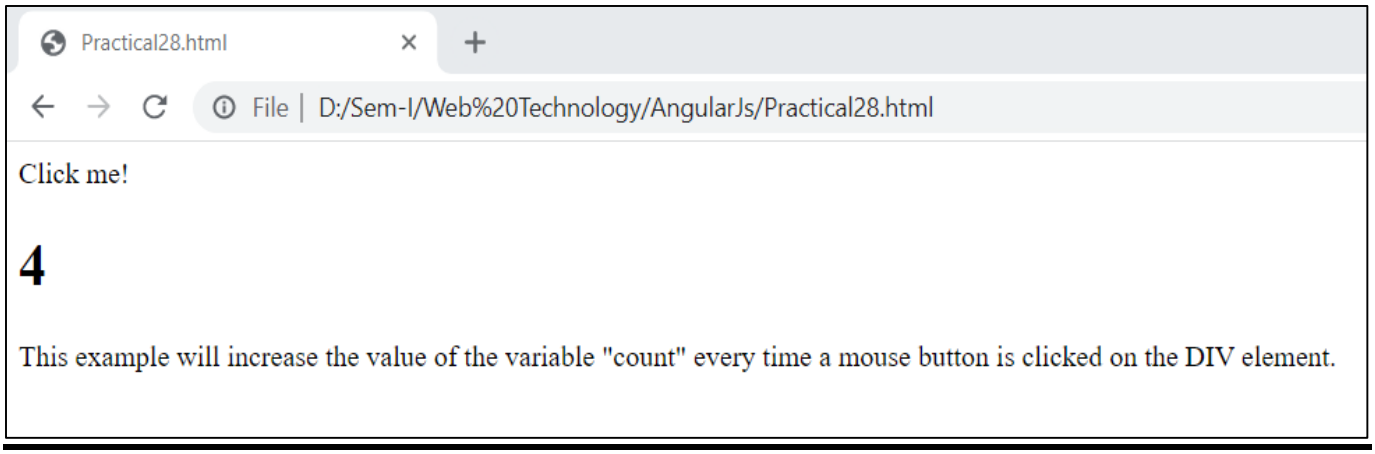
Aim: Create a webpage to perform Mouse Event Directives “ng- mouse down” in Angular JS.

Line of code:

```
<!DOCTYPE html>
<html>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<body ng-app="">

<div ng-mousedown="count = count + 1" ng-init="count=0">Click me!</div>

<h1>{{count}}</h1>

<p>This example will increase the value of the variable "count" every time a mouse button is clicked on the DIV element.</p>

</body>
</html>
```

Execute Screen:





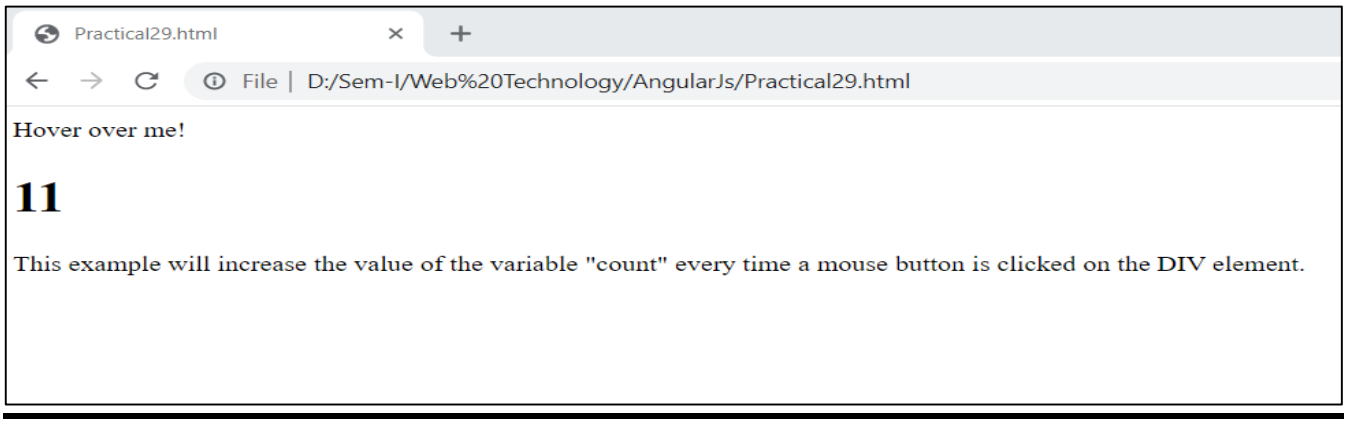
Practical No. 29

Aim: Write a webpage to perform Mouse Event Directives “ng- mouseenter” in Angular JS.

Line of code:

```
<!DOCTYPE html>
<html>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<body ng-app="">

<div ng-mouseenter="count = count + 1" ng-init="count=0">Hover over me!</div>

<h1>{{count}}</h1>

<p>This example will increase the value of the variable "count" every time a mouse button is
clicked on the DIV element.</p>

</body>
</html>
```

Execute Screen:



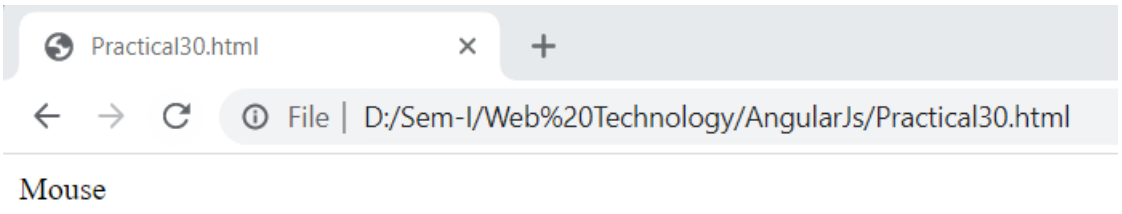
Practical No. 30

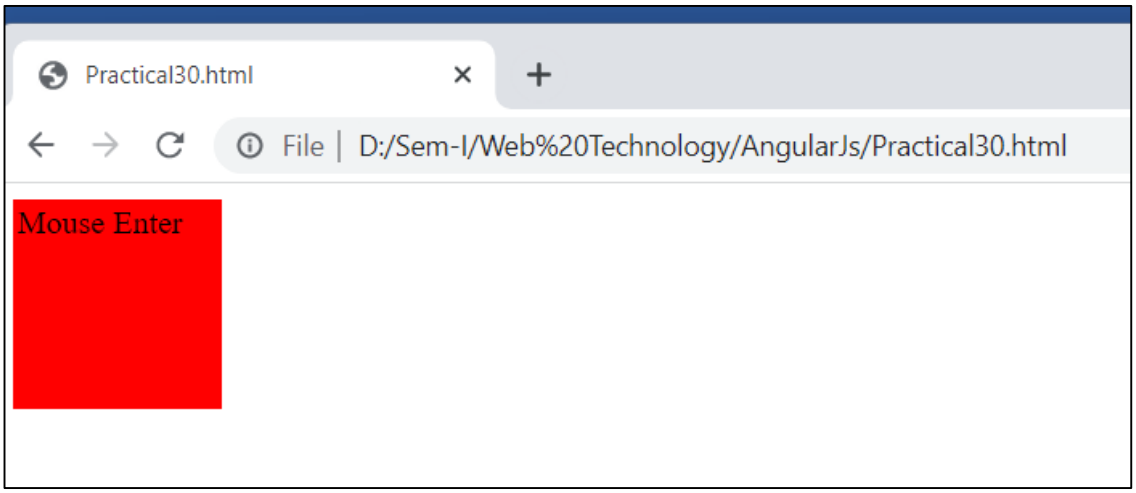
Aim: Create a webpage to perform Mouse Event Directives “ng- mouseenter” in Angular JS. (Colour Change)

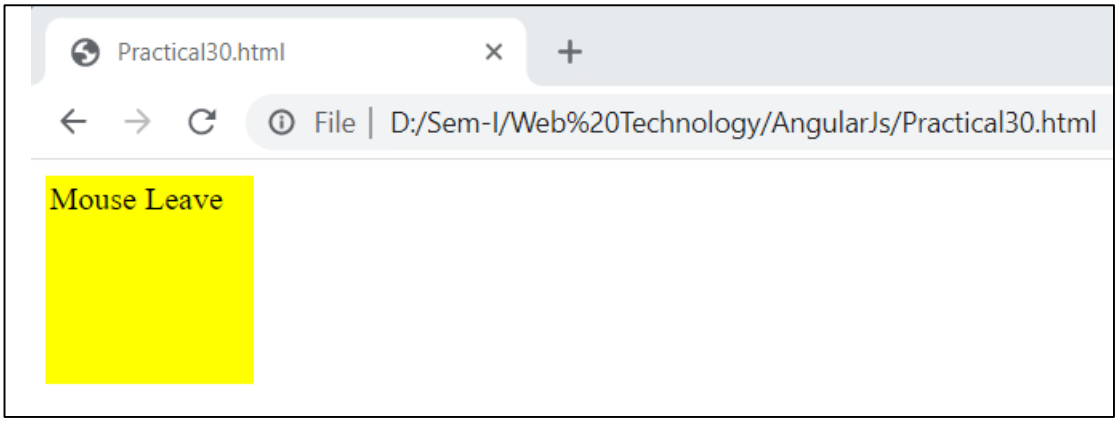
Line of code:

```
<!--Mouse Event-->
<!DOCTYPE html>
<html>
<head>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
    <style>
        .redDiv {
            width: 100px;
            height: 100px;
            background-color: red;
            padding:2px 2px 2px 2px;
        }

        .yellowDiv {
            width: 100px;
            height: 100px;
            background-color: yellow;
            padding:2px 2px 2px 2px;
        }
    </style>
</head>
<body ng-app>
        <div ng-class="{redDiv: enter, yellowDiv: leave}" ng-
mouseenter="enter=true;leave=false;" ng-mouseleave="leave=true;enter=false">
            Mouse <span ng-show="enter">Enter</span> <span ng-show="leave">Leave</span>
        </div>
</body>
</html>
```

Execute Screen:




Practical30.html
File | D:/Sem-I/Web%20Technology/AngularJs/Practical30.html
Mouse Enter


Practical30.html
File | D:/Sem-I/Web%20Technology/AngularJs/Practical30.html
Mouse Leave

MCAL14: Web Technologies Lab

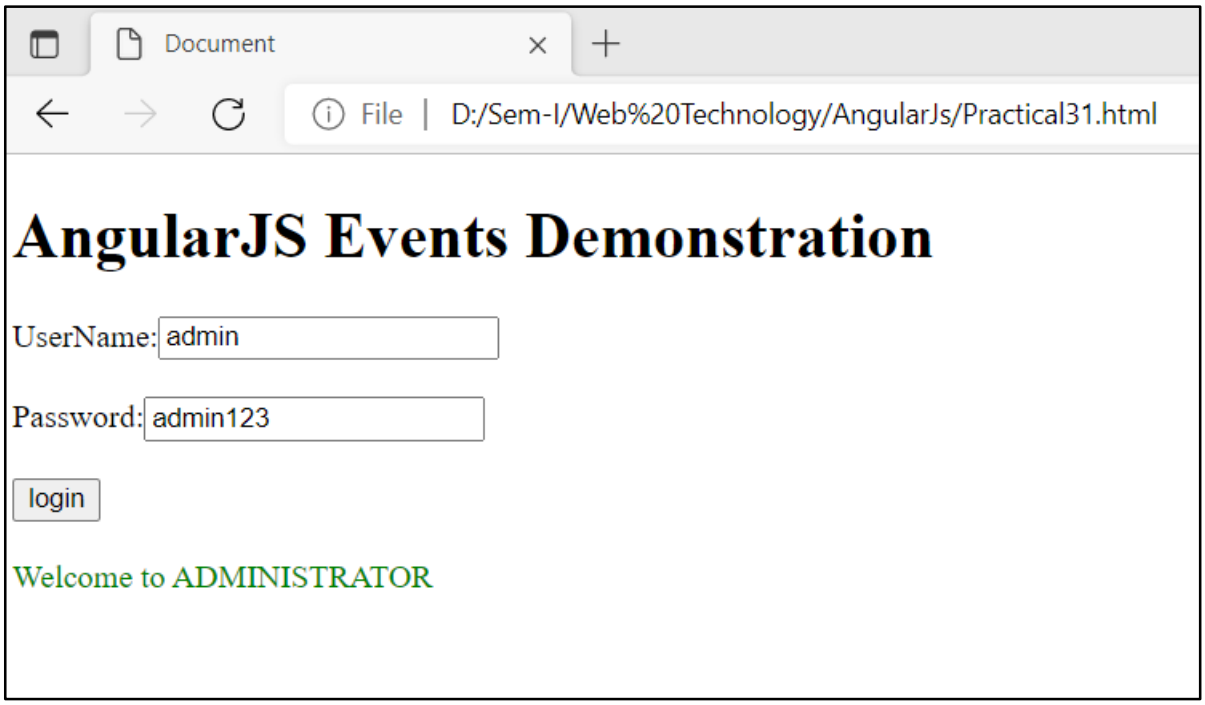
Practical No. 31

Aim: Write a script to create a webpage to implement a login function using Angular JS event.

Line of code:

```
<!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="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"
></script>
    <script>
        var obj=angular.module("myApp",[]);
        obj.controller("Democontroller",function($scope)
        {
            $scope.uid="";
            $scope.pwd="";
            $scope.message="";
            $scope.str="Black";
            $scope.login=function()
            {
                if($scope.uid=="admin" && $scope.pwd=="admin123")
                {
                    $scope.message="Welcome to ADMINISTRATOR";
                    $scope.str="Green"
                }
                else
                {
                    $scope.message="Invalid UserID or Password";
                    $scope.str="Red"
                }
            }
        });
    </script>
</head>
<body ng-app="myApp">
    <h1>AngularJS Events Demonstration</h1>
    <div ng-controller="Democontroller">
    UserName:<input type="text" ng-model="uid"/><br><br>
    Password:<input type="text" ng-model="pwd"/><br><br>
    <input type="button" ng-click="login()" value="login"/><br><br>
    <span style="color:{{str}}">{{message}}</span>
</div>
</body>
</html>
```

Execute Screen:



MCAL14: Web Technologies Lab

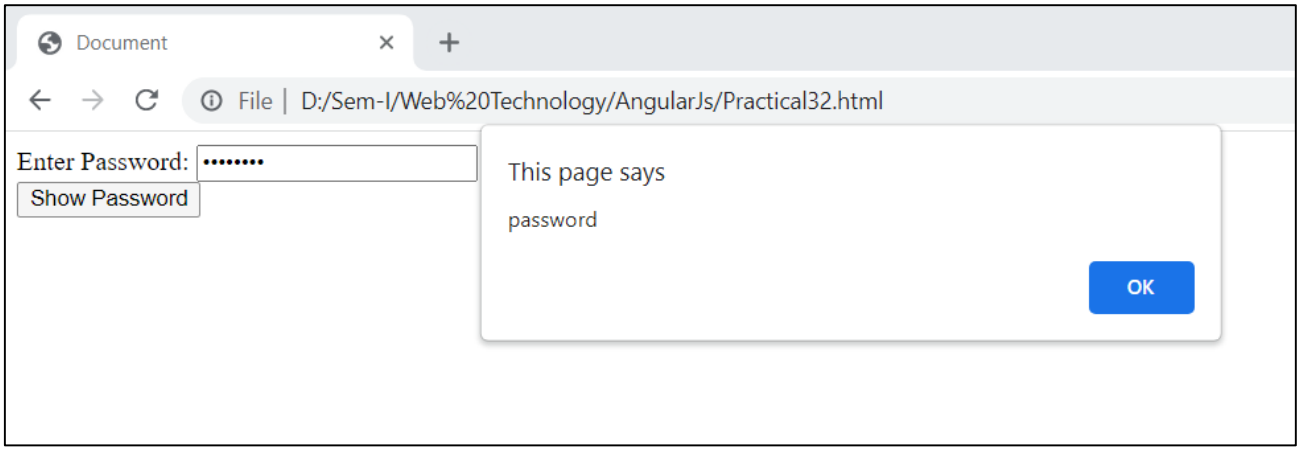
Practical No. 32

Aim: Write a script to create a math operation using the Angular JS event.

Line of code:

```
<!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="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-app="myApp">
     <div ng-controller="myController">
         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>
```

Execute Screen:



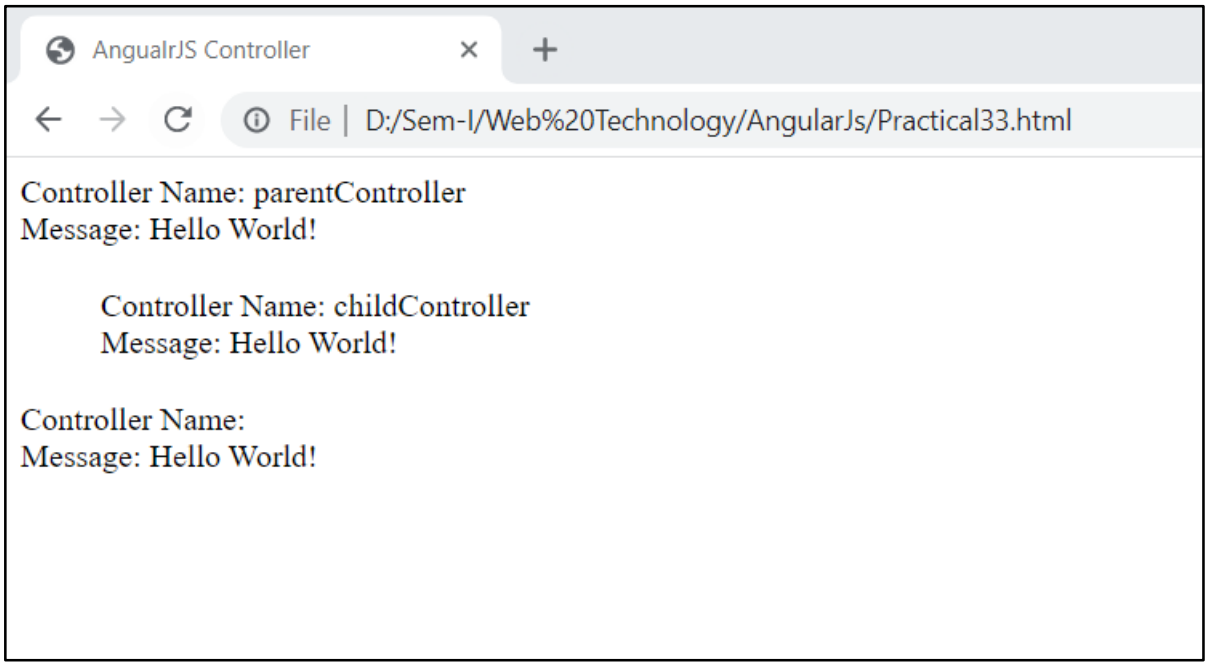
Practical No. 33

Aim: Write a script using built-in object $scope in Angular JS, which contains application data and method.

Line of code:

```
<!DOCTYPE html>
<html>
<head>
 <title>AngualrJS Controller</title>
 <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-app="myNgApp">
 <div ng-controller="parentController">
 Controller Name: {{controllerName}} <br />
 Message: {{message}} <br />
 <div style="margin:20px 0 20px 40px;" ng-controller="childController">
 Controller Name: {{controllerName}} <br />
 Message: {{message}} <br />
 </div>
 </div>
 <div ng-controller="siblingController">
 Controller Name: {{controllerName}} <br />
 Message: {{message}} <br />
 </div>
 <script>
 var ngApp = angular.module('myNgApp', []);
 ngApp.controller('parentController', function ($scope, $rootScope) {
 $scope.controllerName = "parentController";
 $rootScope.message = "Hello World!";
 });
 ngApp.controller('childController', function ($scope) {
 $scope.controllerName = "childController";
 });
 ngApp.controller('siblingController', function ($scope) {

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

Execute Screen:



Practical No. 34

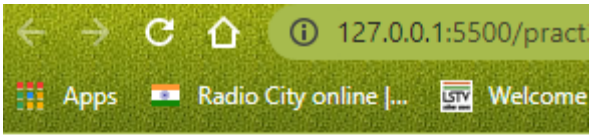
Aim: Write a script to demonstrate $rootscope in Angular JS.

Line of code:

```
<!--Rootscope in Angular JS -->
<!DOCTYPE html>
<html>
<head>
   <title>AngualrJS Controller</title>
   <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-app="myNgApp">
   <div ng-controller="parentController">
      Controller Name: {{controllerName}} <br />
      Message: {{message}} <br />
      <div style="margin:20px 0 20px 40px;" ng-controller="childController">
         Controller Name: {{controllerName}} <br />
         Message: {{message}} <br />
      </div>
   </div>
   <div  ng-controller="siblingController">
      Controller Name: {{controllerName}} <br />
      Message: {{message}} <br />
   </div>
   <script>
      var ngApp = angular.module('myNgApp', []);

      ngApp.controller('parentController', function ($scope, $rootScope) {
```

```
        $scope.controllerName = "parentController";
        $rootScope.message = "Hello World!";
    });

    ngApp.controller('childController', function ($scope) {
        $scope.controllerName = "childController";
    });

    ngApp.controller('siblingController', function ($scope) {

    });
  </script>

</body>
</html>
```

Execute Screen:



Controller Name: parentController
Message: Hello World!

Controller Name: childController
Message: Hello World!

Controller Name:
Message: Hello World!

Practical No. 35

Aim: Create a webpage to demonstrate Single Page Application (SPA) using Angular JS.

Line of code:

1) about.html

```
<div class="jumbotron text-center">
    <h1>About Page</h1>

    <p>{{ message }}</p>
</div>
```

2)Contact.html

```
<div class="jumbotron text-center">
    <h1>Contact Page</h1>
    <p>{{ message }}</p>
</div>
```

3) Home.html

```
<div class="jumbotron text-center">
    <h1>Home Page</h1>
    <p>{{ message }}</p>
</div>
```

4)Index.html

```
<!DOCTYPE html>
    <html ng-app="myApp">
    <head>
      <!-- SCROLLS -->
      <!-- load bootstrap and fontawesome via CDN -->
      <link rel="stylesheet" href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css" />
      <link rel="stylesheet" href="//netdna.bootstrapcdn.com/font-awesome/4.0.0/css/font-awesome.css" />

      <!-- SPELLS -->
      <!-- load angular and angular route via CDN -->
      <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.2.25/angular.min.js"></script>
        <script src="//ajax.googleapis.com/ajax/libs/angularjs/1.2.25/angular-route.js"></script>
      <script src="script.js"></script>
    </head>
    <body ng-controller="mainController">
```

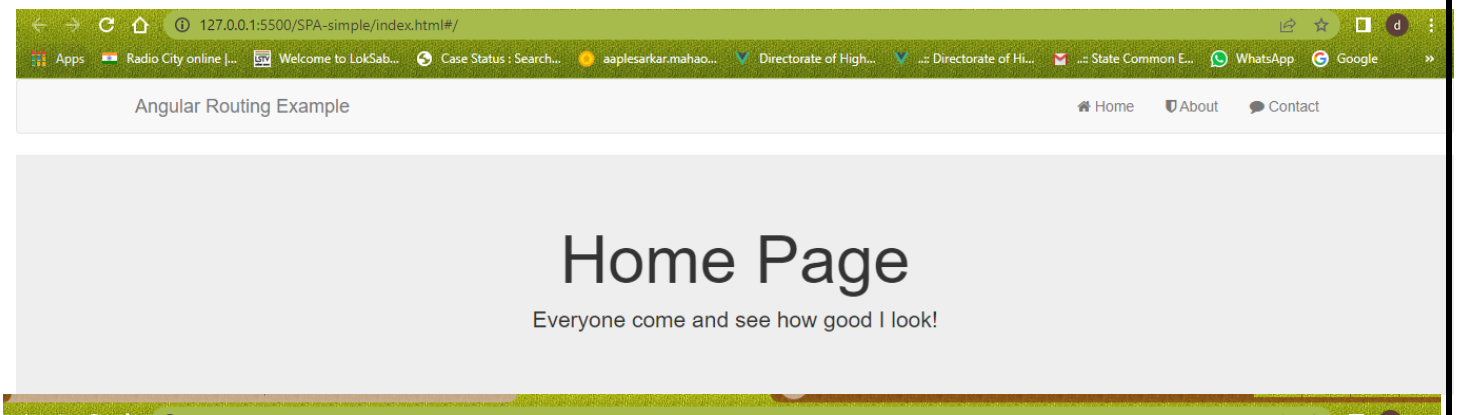
```
<!-- HEADER AND NAVBAR -->
<header>
  <nav class="navbar navbar-default">
  <div class="container">
    <div class="navbar-header">
      <a class="navbar-brand" href="/">Angular Routing Example</a>
    </div>

    <ul class="nav navbar-nav navbar-right">
      <li><a href="#"><i class="fa fa-home"></i> Home</a></li>
      <li><a href="#about"><i class="fa fa-shield"></i> About</a></li>
      <li><a href="#contact"><i class="fa fa-comment"></i> Contact</a></li>
    </ul>
  </div>
  </nav> </header>

<!-- MAIN CONTENT AND INJECTED VIEWS -->
<div id="main">
      <!-- angular templating -->
  <!-- this is where content will be injected -->
  <div ng-view></div>
</div>

</body>
</html>
```

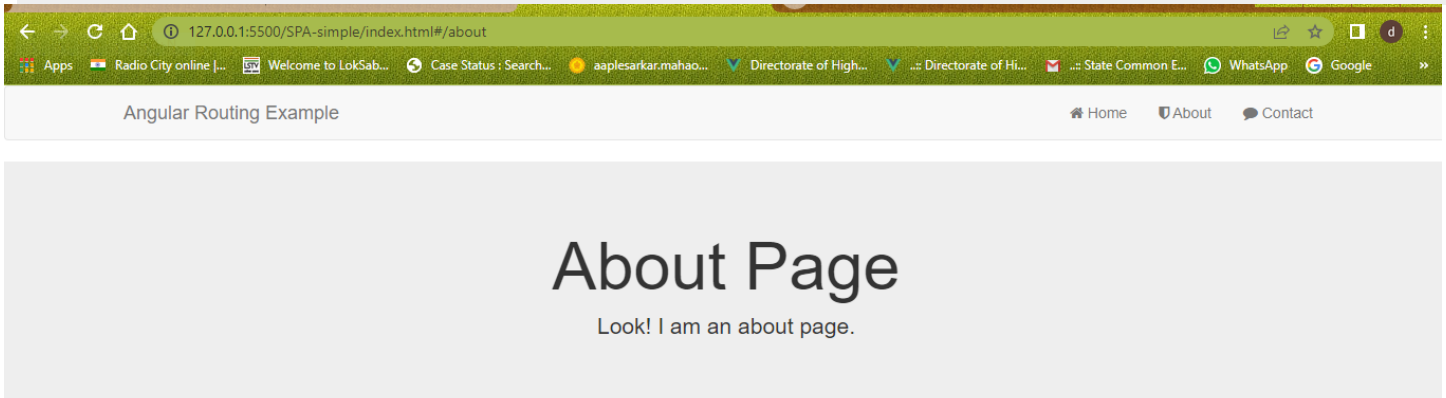
5) Script.js

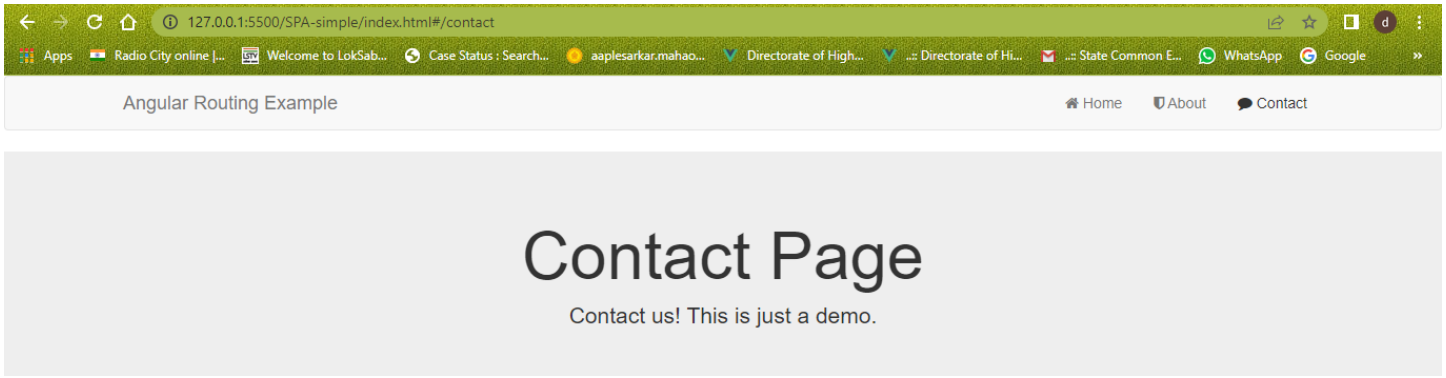
```
// script.js

// create the module and name it scotchApp
  // also include ngRoute for all our routing needs
  var myApp = angular.module('myApp', ['ngRoute']);

  // configure our routes
   myApp.config(function($routeProvider) {
    $routeProvider

      // route for the home page
      .when('/', {
        templateUrl : 'home.html',
        controller  : 'mainController'
      })

 // route for the about page
      .when('/about', {
        templateUrl : 'about.html',
        controller  : 'aboutController'
```

```
        // route for the contact page
        .when('/contact', {
            templateUrl : 'contact.html',
            controller  : 'contactController'
        });
});

// create the controller and inject Angular's $scope
myApp.controller('mainController', function($scope) {
    // create a message to display in our view
    $scope.message = 'Everyone come and see how good I look!';
});

myApp.controller('aboutController', function($scope) {
    $scope.message = 'Look! I am an about page.';
});

myApp.controller('contactController', function($scope) {
    $scope.message = 'Contact us! This is just a demo.';
});
```

Execute Screen:







Name: Deepesh Mangesh Mhatre

App No: 81389

Practical No. 36

Aim: Create a webpage to demonstrate Advance Single Page Application (SPA) using Angular JS.

Line of code:

1) app.js

```
//creating module object
var myApp = angular.module("myApp", ["ngRoute", "ngResource"]);

//create configuration with module object it provides route details
myApp.config(function ($routeProvider) {
  $routeProvider.when("/Route1", {
    templateUrl: "Students.html",
    controller: "StudentsController",
  });
  $routeProvider.when("/Route2", {
    templateUrl: "Courses.html",
    controller: "CoursesController",
  });
});

//Controller for Students.html
myApp.controller("StudentsController", function ($scope) {
  $scope.Sname = "Scott";
  $scope.Course = "AngularJS";
});

//Controller for Courses.html
myApp.controller("CoursesController", function ($scope) {
  $scope.Courses = ["AngularJS", "PHP", "jQuery", "JSP", "ASP"];
});
```

2) Courses.html

```
<h3>Course Details</h3>
<hr />
<ul>
    <li ng-repeat="item in Courses">{{item}}</li>
</ul>
```

3) index.html

```
<html>

<head>
    <style>
        #div1 {
            margin: 15px;
            padding: 15px;
            border: 2px Solid green;
            background-color: lightyellow;
        }
    </style>
```

```
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.2.7/angular.js"></script>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.2.7/angular-route.js"></script>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.2.7/angular-resource.js"></script>
    <script src="./app.js"></script>

    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet" />

 </head>

 <body ng-app="myApp">
    <h1>SPA with AJS Routing Demo</h1>
    <hr />
    <a href="#Route1"> Student Details</a>
    <a href="#Route2"> Course Details</a>
    <hr />
    <div id="div1" ng-view></div>
    <hr />
 </body>

</html>
```

4) student.html

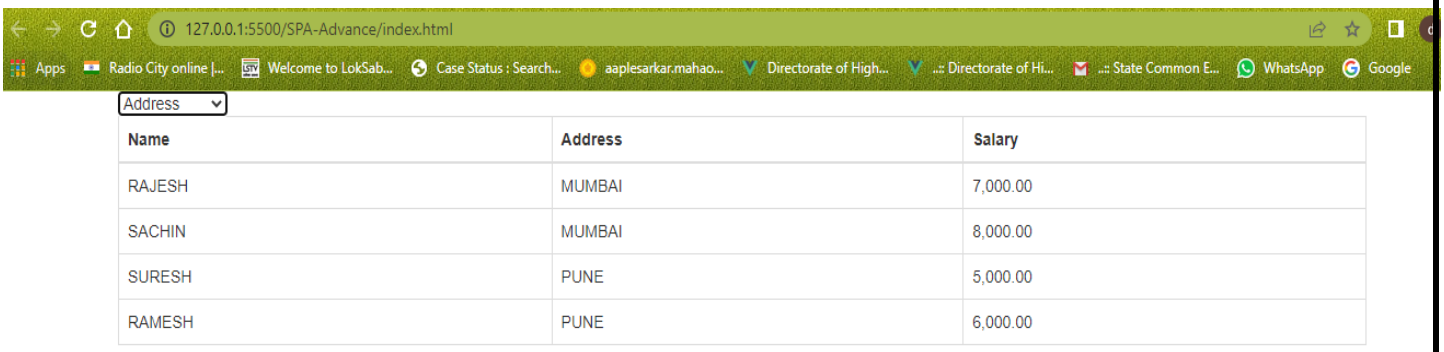
```
 <h3>Student Details</h3><hr/>
 <span>Student Name:{{Sname}}</span><br/>
<span>Course Name:{{Course}}</span>
```

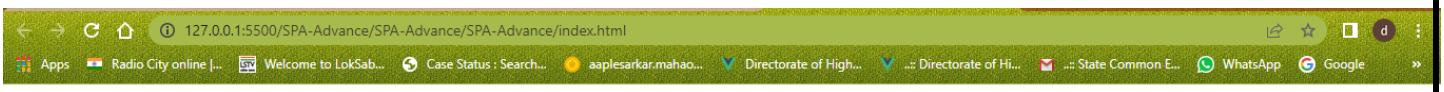
5) index.html

```
 <!doctype html>
 <html>
 <head>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
    <!-- Latest compiled and minified CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
       integrity="sha384-
 BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
 crossorigin="anonymous">

 </head>

 <body ng-app="myangularjsapp">
    <div class="container" ng-controller="myController">
       <select ng-model="SortColumn">
          <option value="name">Name Asc</option>
          <option value="Address">Address</option>
          <option value="Salary">Salary Asc</option>
```

```
<option value="-Salary">Salary Desc</option>
</select>
      <table class="table table-bordered">
        <thead>
          <tr>
            <th>Name</th>
            <th>Address</th>
            <th>Salary</th>
          </tr>
        </thead>
        <tr ng-repeat="employee in employees | orderBy: SortColumn">
          <td>{{employee.name | uppercase }}</td>

          <td>{{employee.Address | uppercase }}</td>
          <td>{{employee.Salary | number :2 }}</td>

        </tr>
        </tbody>
      </table>
  </div>
  <script>

    var app = angular.module("myangularjsapp", []).controller("myController", function ($scope) {

      var employees = [{
        name: "Suresh",
        Address: "Pune",
        Salary: "5000.00"
      }, {
        name: "Ramesh",
        Address: "Pune",
        Salary: "6000.00"
      }, {
        name: "Rajesh",
        Address: "Mumbai",
        Salary: "7000.00"
      }, {
        name: "Sachin",
        Address: "Mumbai",
        Salary: "8000.00"
      }];
      $scope.employees = employees;
    });
  </script>

</body>
</html>
```

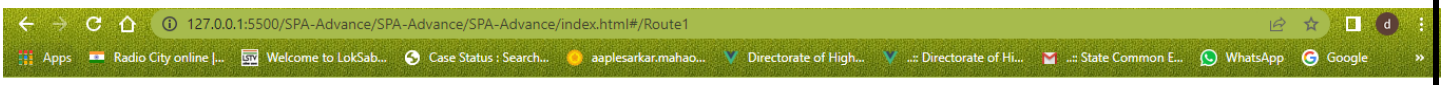
Execute Screen:





SPA with AJS Routing Demo

Student Details Course Details

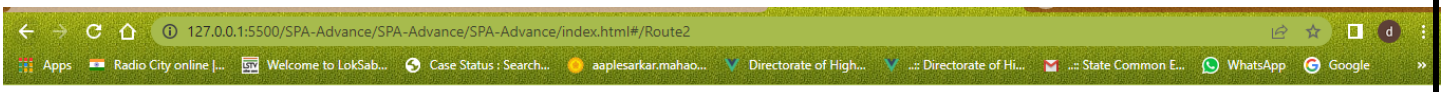


SPA with AJS Routing Demo

Student Details Course Details

Student Details

Student Name:Scott
Course Name:AngularJS



SPA with AJS Routing Demo

Student Details Course Details

Course Details

- AngularJS
- PHP
- jQuery
- JSP
- ASP

Name: Deepesh Mangesh Mhatre

App No: 81389

MCAL14: Web Technologies Lab

Practical No. 37

Aim: Create a webpage to demonstrate Single Page Application (SPA) for student login and form-filling using Angular JS.

Line of code:

1) Login.html

```
<form class="form-horizontal" role="form" name="loginForm" novalidate>
  <div class="form-group">
    <div class="col-sm-3">
    </div>
    <div class="col-sm-6">
      <input type="text" id="userName" name="userName" placeholder="User Name" class="form-control"
        ng-model="userName" required />
      <span class="help-block" ng-show="loginForm.userName.$touched && loginForm.userName.$invalid">Please enter
        User Name.</span>
    </div>
    <div class="col-sm-3">
    </div>

  </div>
  <div class="form-group">
    <div class="col-sm-3">
    </div>
    <div class="col-sm-6">
      <input type="password" id="password" name="password" placeholder="Password" class="form-control"
        ng-model="password" required />
      <span ng-show="loginForm.password.$touched && loginForm.password.$error.required">Please enter
        Password.</span>
    </div>
    <div class="col-sm-3">
    </div>
  </div>

  <input type="submit" value="Login" class="btn btn-primary col-sm-offset-3" ng-click="authenticate(userName)" />
</form>
```

2) index.html

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
  <title></title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
  <script src="./angular-route.js"></script>
  <script src="https://code.angularjs.org/1.8.2/angular-resource.min.js"></script>
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet" />
```

```
</head>

<body ng-app="ngRoutingDemo">
   <h1>Angular Routing Demo</h1>

   <div class="container">

      <div ng-view></div>
   </div>
   <script>


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

      app.config(function ($routeProvider) {

         $routeProvider.when('/', {
            templateUrl: "/login.html",
            controller: "loginController"
         }).when('/student/:username', {
            templateUrl: "/student.html",
            controller: "studentController"
         }).otherwise({
            redirectTo: "/"
         });



      });
      app.controller("loginController", function ($scope, $location) {

         $scope.authenticate = function (username) {
            // write authentication code here..

            $location.path('/student/' + username)
         };

      });

      app.controller("studentController", function ($scope, $routeParams) {
         $scope.username = $routeParams.username;
      });
```

```
 </script>
</body>

</html>
```

3) LoginService.js

```
//loginService.js
var app = angular.module("myApp");

app.factory("LoginService", function () {
  var admin = "admin";
  var pass = "password";
  var isAuthenticated = false;

  return {
    login: function (username, password) {
      isAuthenticated = username === admin && password === pass;
      return isAuthenticated;
    },
    isAuthenticated: function () {
      return isAuthenticated;
    },
  };
});
```

4) student.html

```
<div>
   <p>Welcome {{username}}</p>
   <a href="/">Log out</a>
</div>

<form class="form-horizontal" ng-submit="submitStudnetForm()" role="form">
   <div class="form-group">
      <label for="firstName" class="col-sm-3 control-label">First Name</label>
      <div class="col-sm-6">
         <input type="text" id="firstName" class="form-control" ng-model="student.firstName" />
      </div>
      <div class="col-sm-3"></div>

   </div>
   <div class="form-group">
      <label for="lastName" class="col-sm-3 control-label">Last Name</label>
      <div class="col-sm-6">
         <input type="text" id="lastName" class="form-control" ng-model="student.lastName" />
      </div>
```

```
    <div class="col-sm-3"></div>
  </div>

  <div class="form-group">
    <label for="dob" class="col-sm-3 control-label">DoB</label>
    <div class="col-sm-2">
      <input type="date" id="dob" class="form-control" ng-model="student.DoB" />
    </div>
    <div class="col-sm-7"></div>
  </div>

  <div class="form-group">
    <label for="gender" class="col-sm-3 control-label">Gender</label>
    <div class="col-sm-2">
      <select id="gender" class="form-control" ng-model="student.gender">
        <option value="male">Male</option>
        <option value="female">Female</option>
      </select>
    </div>
    <div class="col-sm-7"></div>
  </div>

  <div class="form-group">
    <div class="col-sm-3"></div>
    <div class="col-sm-2">
      <span><b>Training Location</b></span>
      <div class="radio">
        <label><input value="online" type="radio" name="training"
            ng-model="student.trainingType" />Online</label>
      </div>
      <div class="radio">
        <label><input value="onsite" type="radio" name="training"
            ng-model="student.trainingType" />OnSite</label>
      </div>
    </div>
    <div class="col-sm-7">
      <span><b>Main Subjects</b></span>
      <div class="checkbox">
        <label><input type="checkbox" ng-model="student.maths" />Maths</label>
      </div>
      <div class="checkbox">
        <label><input type="checkbox" ng-model="student.physics" />Physics</label>
      </div>
      <div class="checkbox">
        <label><input type="checkbox" ng-model="student.chemistry" />Chemistry</label>
      </div>
    </div>
  </div>

  <input type="submit" value="Save" class="btn btn-primary col-sm-offset-3" />
 <input type="reset" value="Reset" ng-click="resetForm()" </form>
```

Execute Screen:

Angular Routing Demo