Name: Jagtap Kalpana Santosh
Div : B 13

Bharati Vidyapeeth's

Institute of Management & Information Technology

C.B.D. Belapur, Navi Mumbai 400614

Vision:
Providing high quality, innovative and value-based education in information technology to build competent professionals. **Mission**
M1. Technical Skills:-To provide solid technical foundation theoretically as well as practically capable of providing quality services to industry.
M2. Development: -Department caters to the needs of students through comprehensive educational programs and promotes lifelong learning in the field of computer Applications.
M3. Ethical leadership:-Department develops ethical leadership insight in the students to succeed in industry, government and academia.

CERTIFICATE

This is to certify that the journal is the work of Ms.Kalpana Santosh Jagtap Roll No. 13 of MCA (Sem- _1 Div:______B_) for the academic year 2020 - 2021
Subject Code:__MCAL14 Subject Name: Web Technology

Subject-in-charge Principal
Date: ____________

External Examiner Date:____________

Name: Jagtap Kalpana Santosh
Div : B 13

Bharati Vidyapeeth's Institute of Management & Information Technology
MCA Semester I AY 2020-21

MCAL14: Web Technologies

INDEX

Name : Kalpana Santosh Jagtap MCA Sem 1 Div : B 13

Name: Jagtap Kalpana Santosh
Div : B 13

- ## Nodejs Module

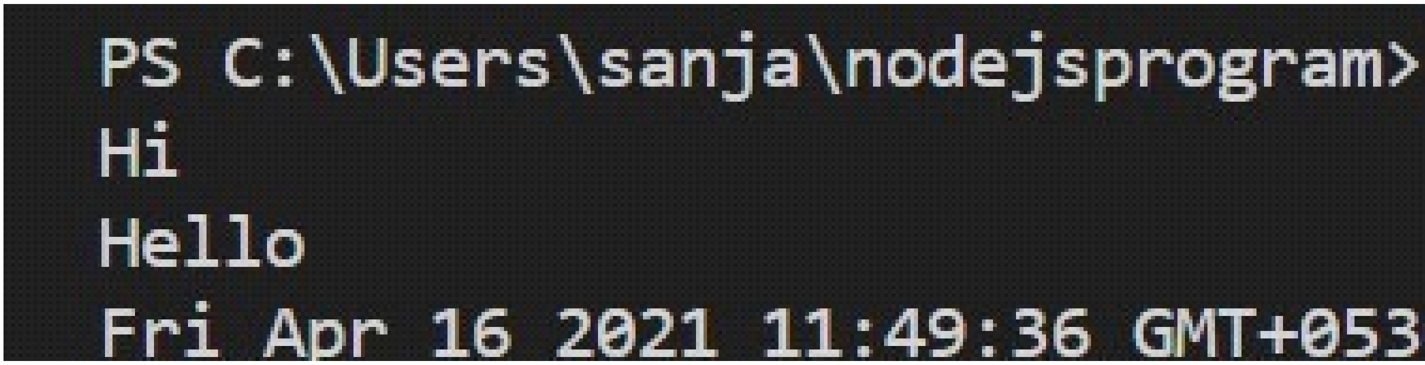
1.1 Create an application to demonstrate Node.js Modules.
Solution:
mymodule.js

```
exports.testFunc=function(){ console.log("Hello")
}
exports.curDate=function(){ return Date()
}
```

date1.js

```
const mymodule=require('./mymodule')
console.log("Hi")
mymodule.testFunc() var
today=mymodule.curDate();
console.log(today)
```

OUTPUT:



```
PS C:\Users\sanja\nodejsprogram>
Hi
Hello
Fri Apr 16 2021 11:49:36 GMT+053
```

- ## Events

- **Create an application to demonstrate various Node.js**

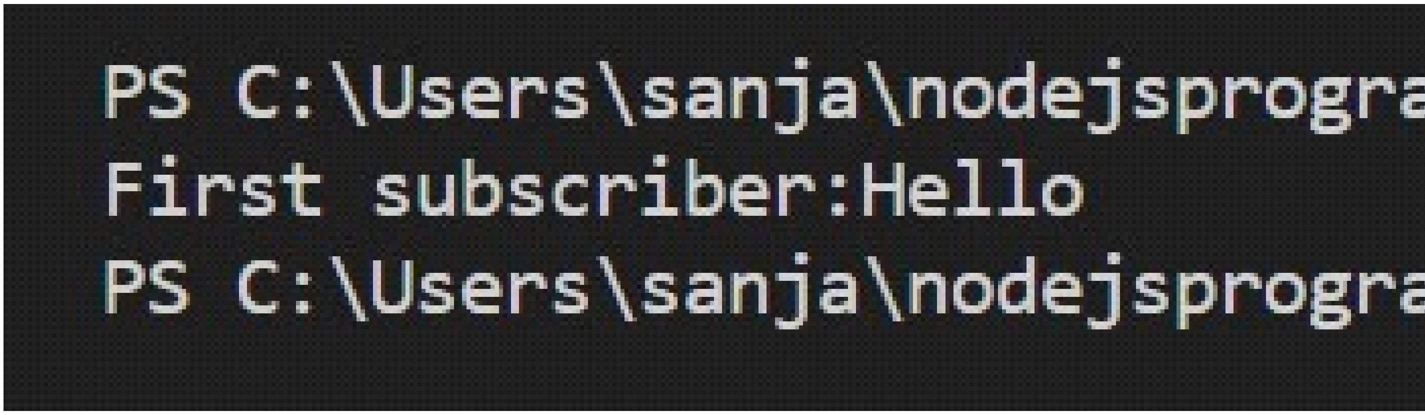
Events. Solution: even.js

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

Name: Jagtap Kalpana Santosh
Div : B 13

```
var      em=new      events.EventEmitter();
em.on('FirstEvent',function(data){
console.log('First subscriber:'+data);

});
em.emit('FirstEvent','Hello');
```

OUTPUT:



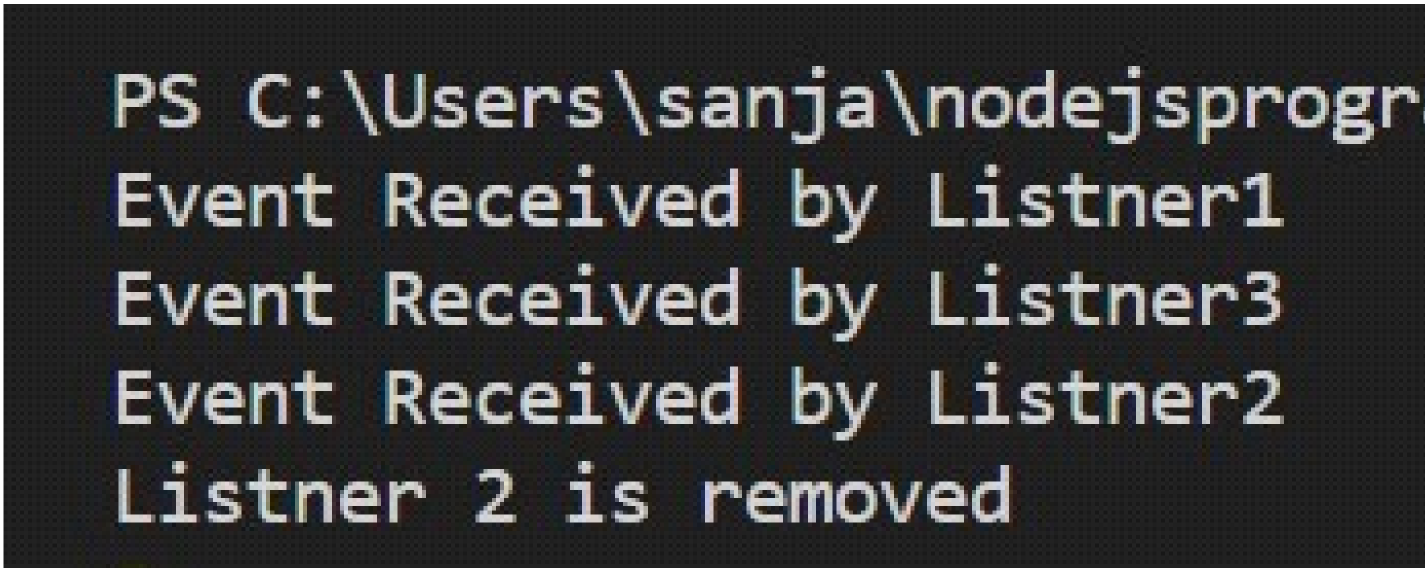
```
PS C:\Users\sanja\nodejsprogra
First subscriber:Hello
PS C:\Users\sanja\nodejsprogra
```

- **Implement all Methods of EventEmitter class. Solution:**

event2.js

```
const     events=require('events');     const
eventEmitter=new       events.EventEmitter();
function listner1()
{ console.log('Event Received by
 Listner1');
} function
listner2()
{ console.log('Event Received by
 Listner2');
} function
listner3()
{ console.log('Event Received by
 Listner3');
}
eventEmitter.addListener('write',listner1);
eventEmitter.addListener('write',listner3);
eventEmitter.on('write',listner2);
eventEmitter.emit('write');
eventEmitter.removeAllListeners('write',listner2); console.log('Listner 2
is removed'); console.log(eventEmitter.listenerCount('write'));
eventEmitter.emit('write');
console.log(eventEmitter.listenerCount('write')); console.log('Program
Ended');
```

OUTPUT:



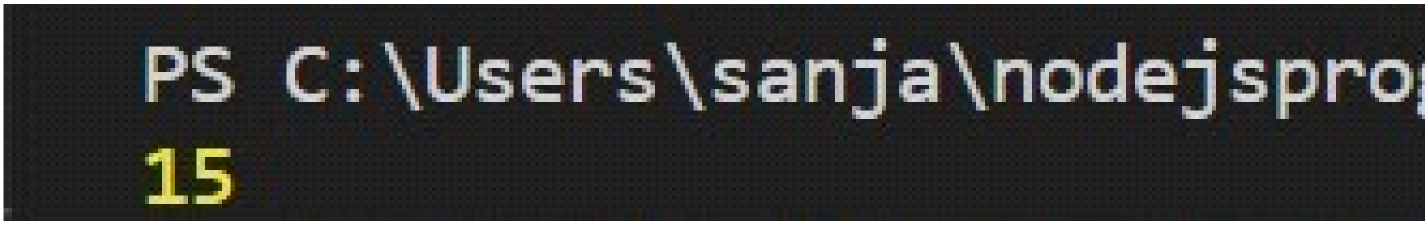
```
PS C:\Users\sanja\nodejsprogr
Event Received by Listner1
Event Received by Listner3
Event Received by Listner2
Listner 2 is removed
```

- **Create an application to demonstrate Node.js Functions**

Solution:

```
function add(x,y)
{ return x+y;
} result=add(5,10);
console.log(result);
```

OUTPUT:



```
PS C:\Users\sanja\nodejsprog
15
```

Name: Jagtap Kalpana Santosh
Div : B 13

3. File System and HTTP Server

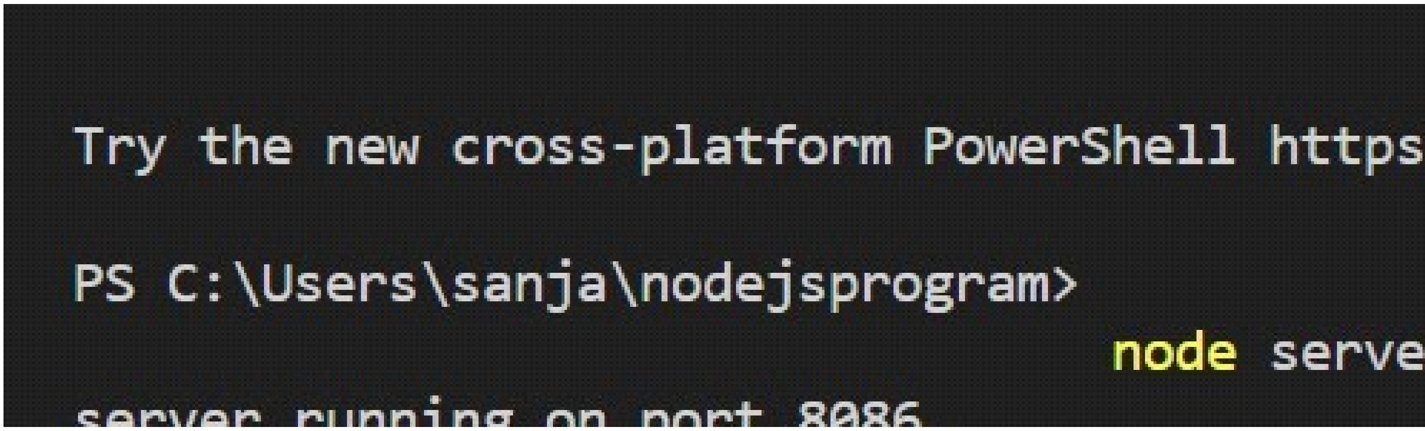
3.1 Create an HTTP Server and perform operations on it.

Solution:

server.js

```
var http=require('http'); const
server=http.createServer(function(req,res)
{ res.write('Hello World!!');
    res.end();

}) server.listen(8086);
console.log("server running on port
8086");
```

OUTPUT:



```
Try the new cross-platform PowerShell https

PS C:\Users\sanja\nodejsprogram>
                                  node serve
server running on port 8086
```

3.2 Using File Handling demonstrate all basic file operations (Create, write, read, delete)

Solution:

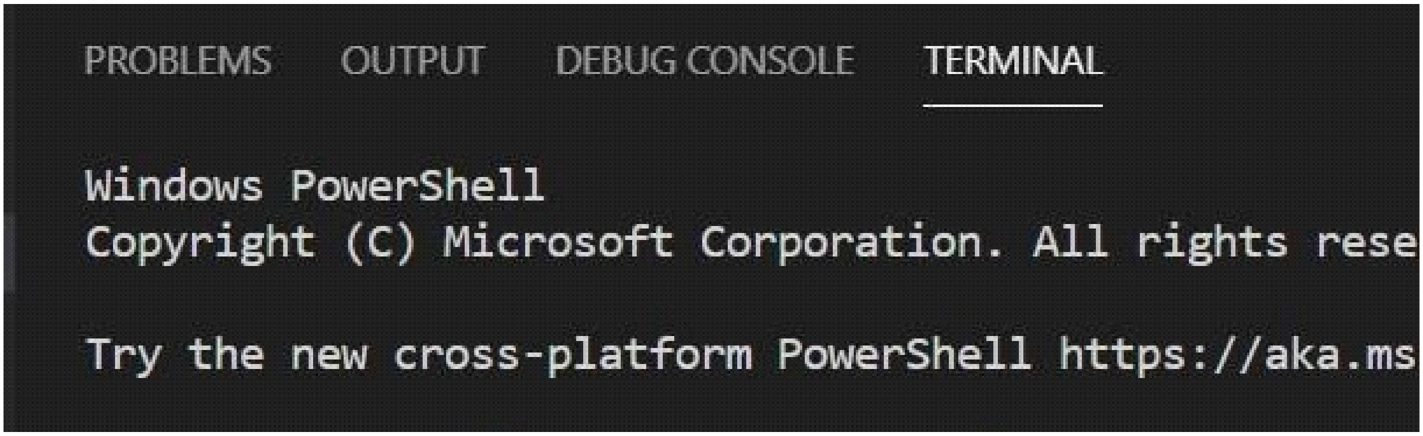
Creating file:

create.js

```
var fs=require('fs');
fs.writeFile('mynewfile.txt','Keep   Learning',function(err){   if(err)
throw err; console.log('Saved!');
});
```

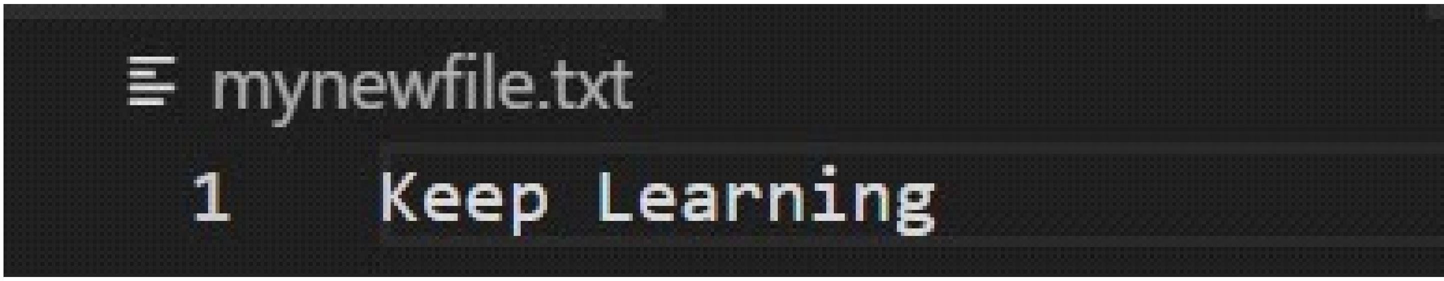
OUTPUT:

Name: Jagtap Kalpana Santosh
Div : B 13



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

Try the new cross-platform PowerShell https://aka.ms
```

File Created

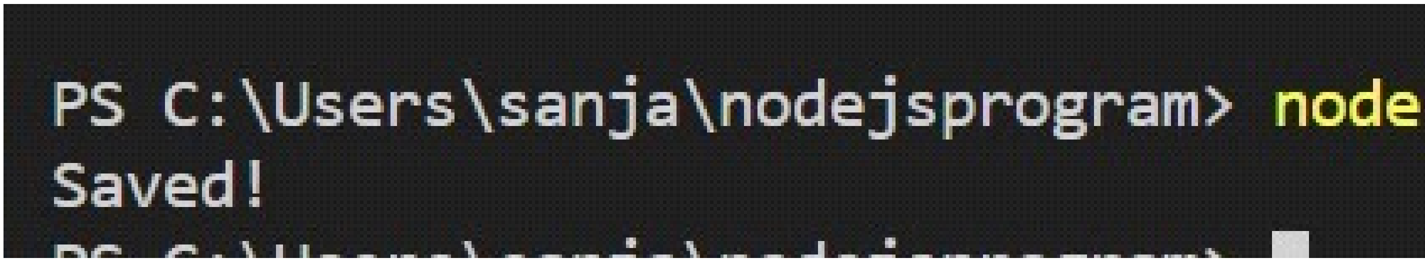


```
mynewfile.txt
1    Keep Learning
```

Writing File:
append.js

```
var    fs=require('fs');    fs.appendFile('mynewfile.txt','Hello
Everyone',function(err){        if(err)        throw        err;
console.log('Saved!'); });
```

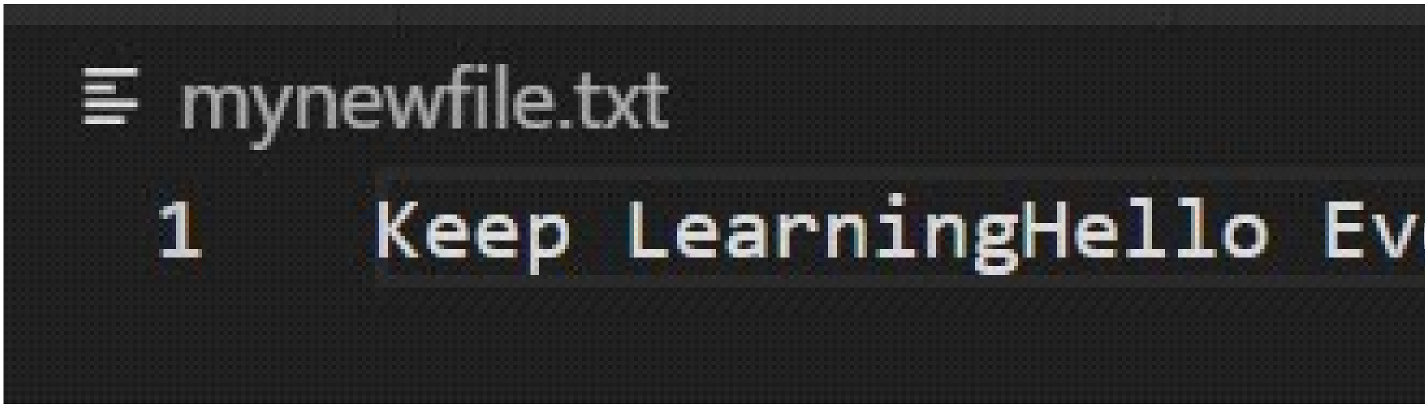
OUTPUT:



```
PS C:\Users\sanja\nodejsprogram> node
Saved!
PS C:\Users\sanja\nodejsprogram>
```

File appended

Name: Jagtap Kalpana Santosh
Div : B 13



mynewfile.txt

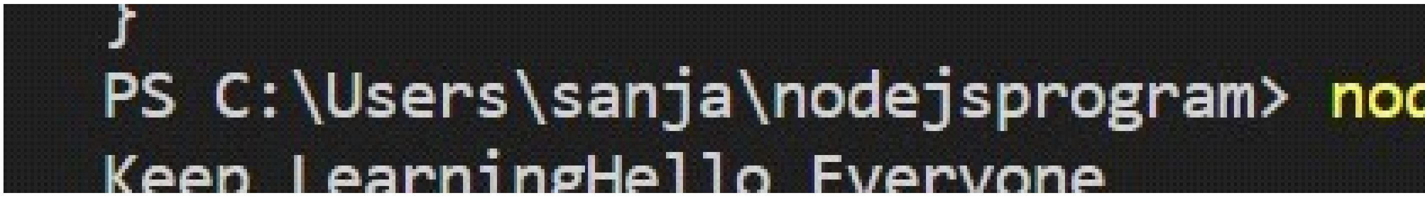
```
Keep LearningHello Ev
```

Reading File:

read.js

```
var fs=require('fs');
fs.readFile('mynewfile.txt','utf8'  ,function(err,data){  if(err)  throw
err; console.log(data);
});
```

OUTPUT:



```
}
PS C:\Users\sanja\nodejsprogram> noc
Keep LearningHello Everyone
```

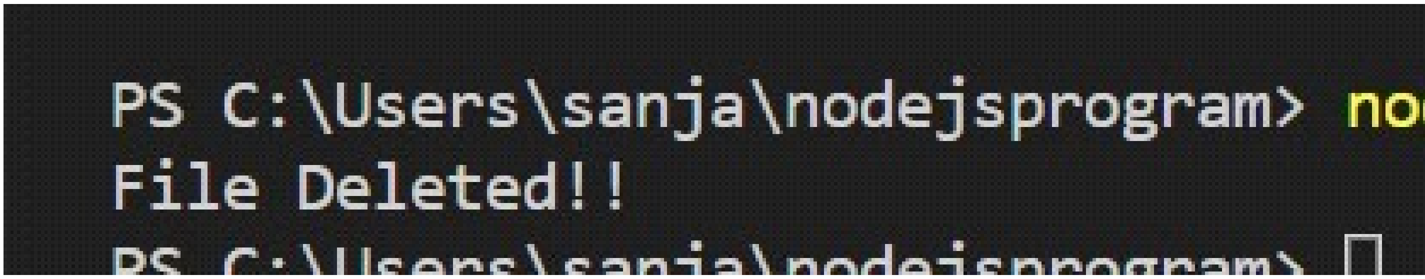
Deleting File:

delete.js

```
var                                        fs=require('fs');
fs.unlink('mynewfile.txt',function(err){  if(err)  throw  err;
console.log('File Deleted!!');
});
```

OUTPUT:

Name: Jagtap Kalpana Santosh
Div : B 13



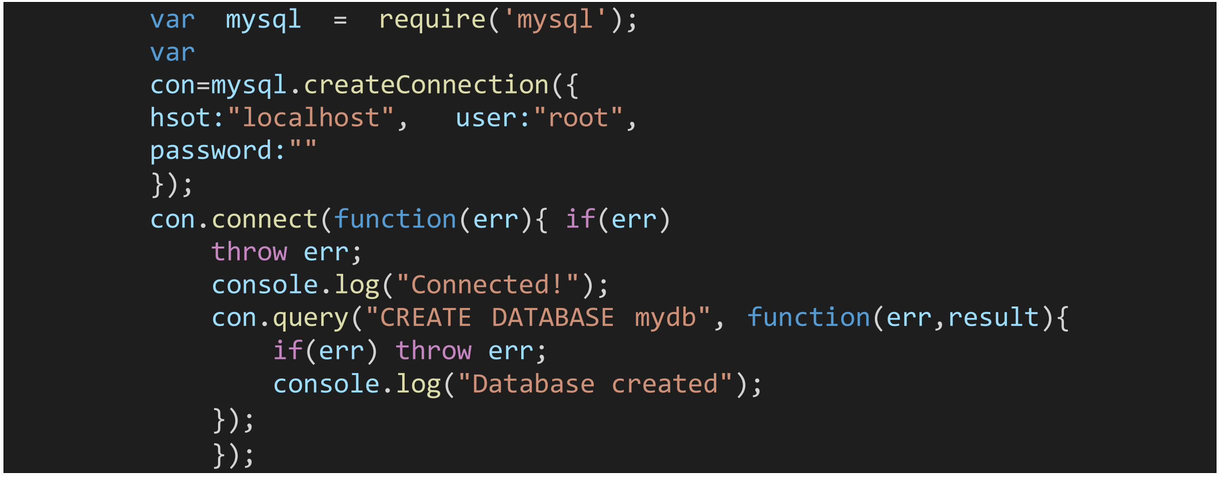
```
PS C:\Users\sanja\nodejsprogram> no
File Deleted!!
PS C:\Users\sanja\nodejsprogram> 
```

- <u>**MySQL database connectivity.**</u>

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

Solution:

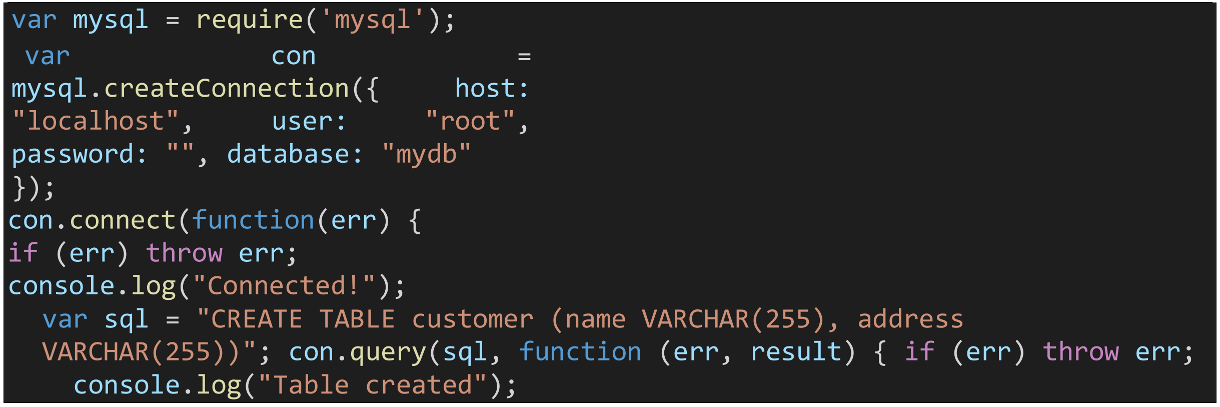
Create Database mydb <u>democonn.js</u>



```
var  mysql  =  require('mysql');
var
con=mysql.createConnection({
hsot:"localhost",   user:"root",
password:""
});
con.connect(function(err){ if(err)
    throw err;
    console.log("Connected!");
    con.query("CREATE DATABASE mydb", function(err,result){
        if(err) throw err;
        console.log("Database created");
    });
    });
```

OUTPUT:

Create Table customer <u>createtable.js</u>



```
var mysql = require('mysql');
 var            con            =
mysql.createConnection({    host:
"localhost",     user:     "root",
password: "", database: "mydb"
});
con.connect(function(err) {
if (err) throw err;
console.log("Connected!");
  var sql = "CREATE TABLE customer (name VARCHAR(255), address
  VARCHAR(255))"; con.query(sql, function (err, result) { if (err) throw err;
    console.log("Table created");
  });
});
```

OUTPUT:

Insert Data into Table insertin.js

```
var  mysql=require('mysql');  var
con=mysql.createConnection({
host:     "localhost",     user:
"root", password: "", database:
"mydb"
});
con.connect(function(err){ if
    (err) throw err;
    console.log("Connected!");
    var  sql="INSERT  INTO  customer  (name,address)  VALUES  ?";
    var values = [
        ['Uttkarsha', 'Sanpada 19'],
        ['Asmita', 'Koperkhairane 20'],
        ['Rutvik', 'Vashi 21'],
        ['Saanvi', 'Nerul 22'],
        ['Shanaya', 'Panvel 23'],
        ['Swara', 'Seawoods 24']
    ];
    con.query(sql,[values],function (err, result){ if
        (err) throw err;
        console.log("Number of records Inserted:" +result.affectedRows);
    }); });
```

OUTPUT:

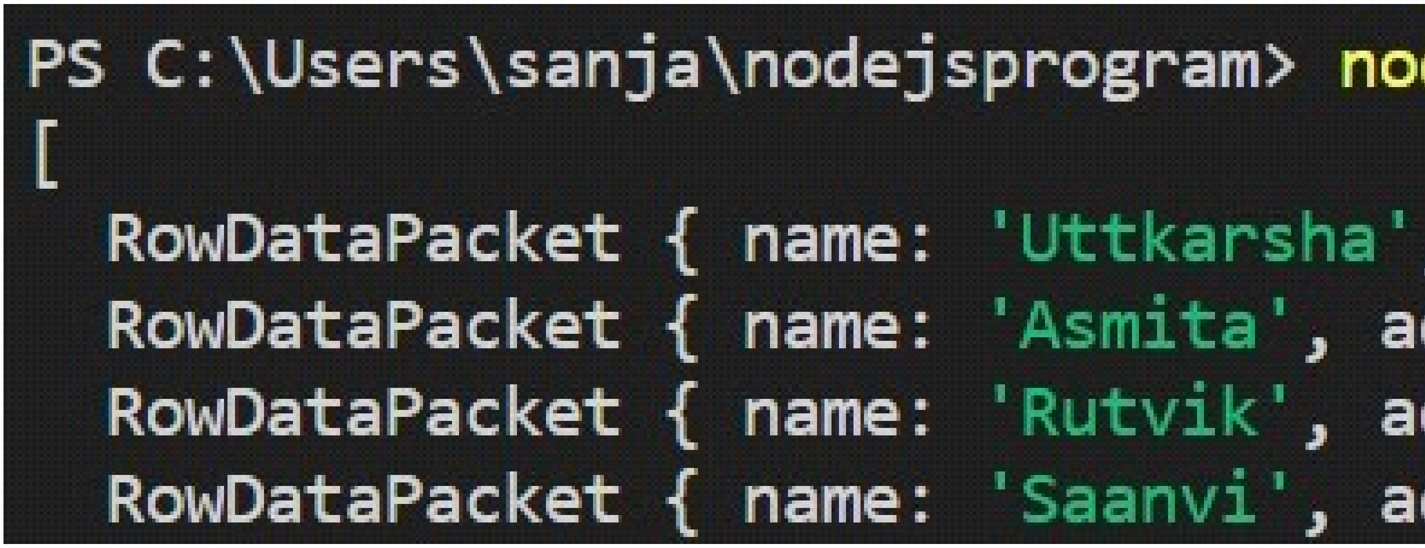
Show Data From Table

select.js

```
var  mysql=require('mysql');  var
con=mysql.createConnection({
host:     "localhost",     user:
"root", password: "", database:
"mydb"
});
con.connect(function(err){
    if (err) throw err;
    con.query("SELECT    *    FROM    customer",    function    (err,
        result,fields){ if (err) throw err; console.log(result);
    }); });
```

Name: Jagtap Kalpana Santosh
Div : B 13

OUTPUT:



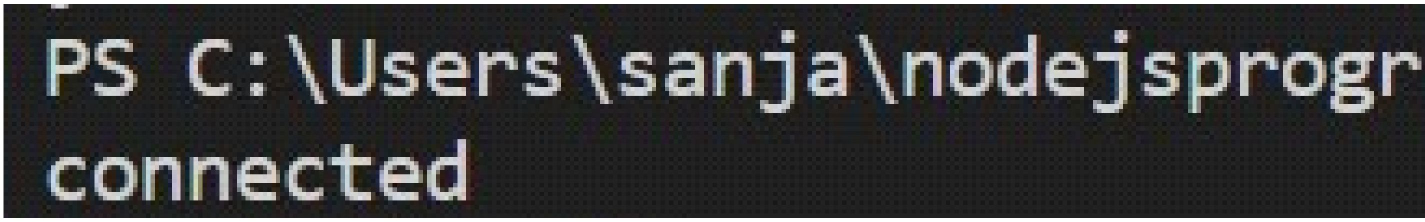
```
PS C:\Users\sanja\nodejsprogram> no
[
   RowDataPacket { name: 'Uttkarsha'
   RowDataPacket { name: 'Asmita', a
   RowDataPacket { name: 'Rutvik', a
   RowDataPacket { name: 'Saanvi', a
```

Alter Data alter.js

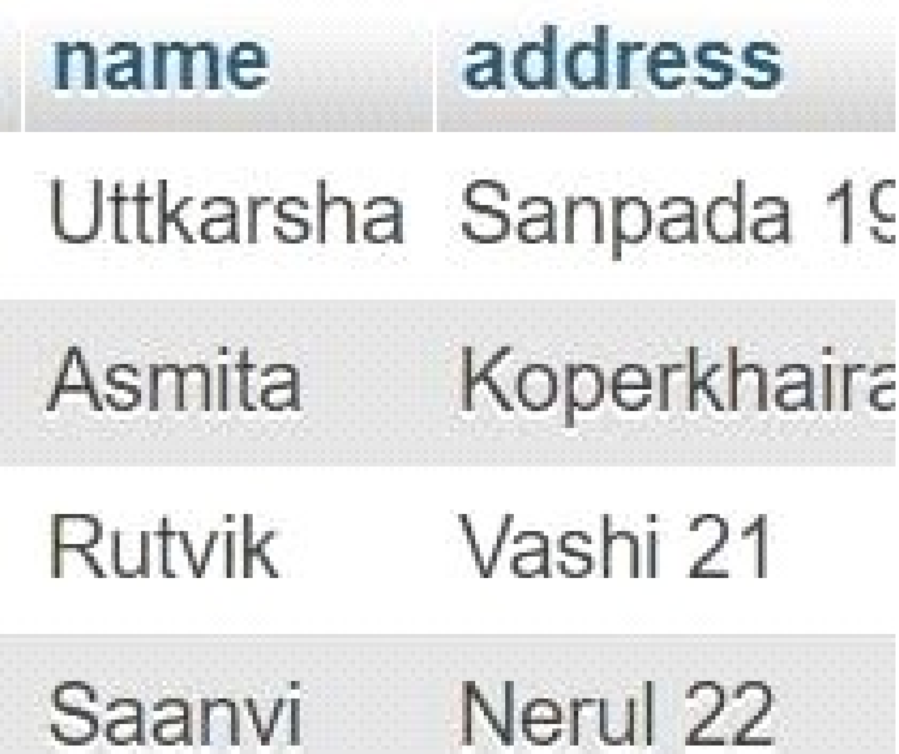
```
var mysql =require('mysql'); var
con = mysql.createConnection({
host:"localhost", user:"root",
password:"", database:"mydb"
});
con.connect(function(err){ if(err)
    throw err;
    console.log("connected");
     var sql="ALTER TABLE customer ADD COLUMN id INT AUTO_INCREMENT PRIMARY KE
Y";  con.query(sql,  function(err,result){
     if(err) throw err;
        console.log("Table
Altered"); }); });
```

OUTPUT:

Name: Jagtap Kalpana Santosh
Div : B 13

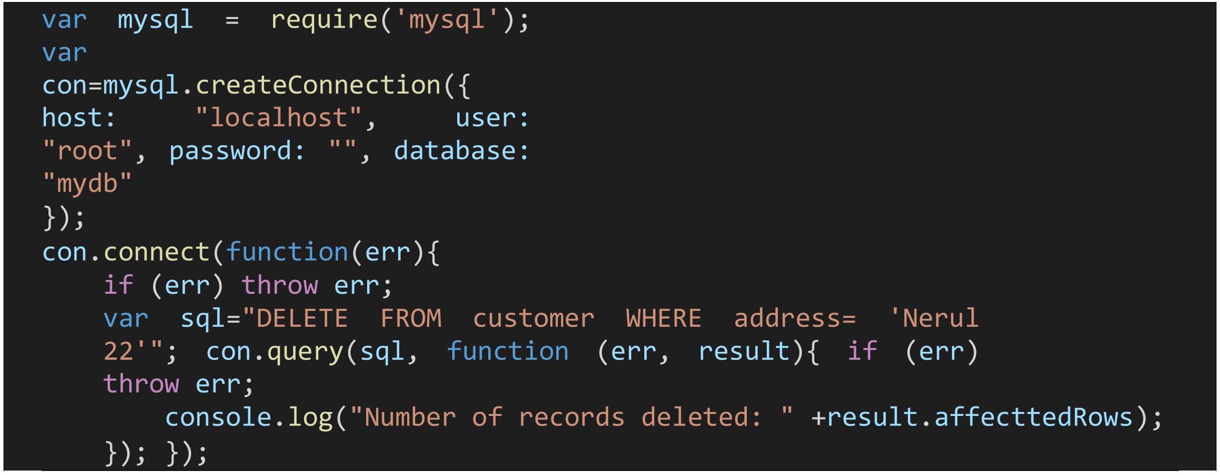


```
PS C:\Users\sanja\nodejsprogr
connected
```



name	address
Uttkarsha	Sanpada 1
Asmita	Koperkhair
Rutvik	Vashi 21
Saanvi	Nerul 22

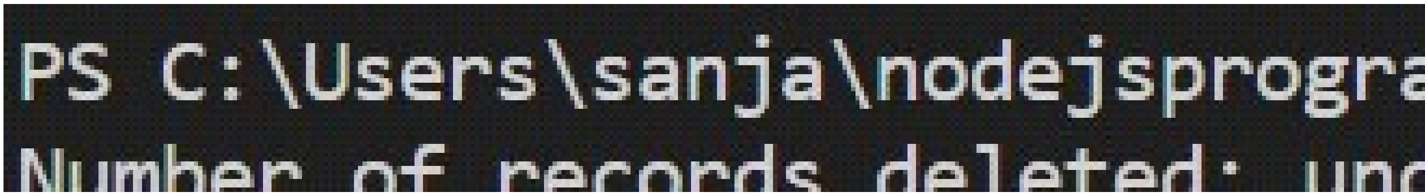
Delete Records From Table **deleterecord.js**



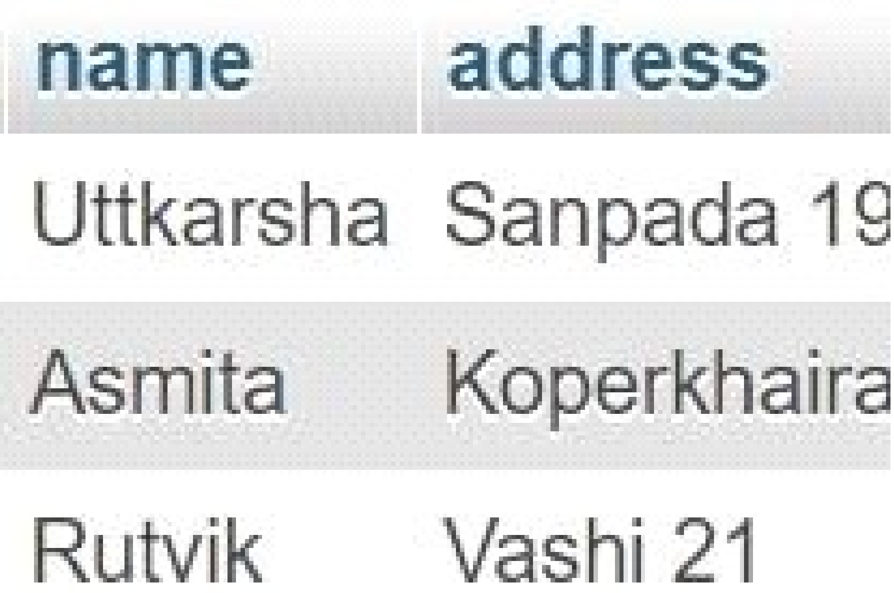
```
var  mysql  =  require('mysql');
var
con=mysql.createConnection({
host:     "localhost",     user:
"root", password: "", database:
"mydb"
});
con.connect(function(err){
    if (err) throw err;
    var  sql="DELETE  FROM  customer  WHERE  address=  'Nerul
    22'"; con.query(sql, function  (err,  result){  if  (err)
    throw err;
        console.log("Number of records deleted: " +result.affecttedRows);
    }); });
```

OUTPUT:

Name: Jagtap Kalpana Santosh
Div : B 13



```
PS C:\Users\sanja\nodejsprogra
Number of records deleted: und
```



name	address
Uttkarsha	Sanpada 19
Asmita	Koperkhaira
Rutvik	Vashi 21

Update data **update.js**

```
var  mysql=require('mysql');  var
con=mysql.createConnection({
host:     "localhost",     user:
"root", password: "", database:
"mydb"
});
con.connect(function(err){
    if (err) throw err;
    var sql="UPDATE customer SET address= 'Kharghar 25' WHERE address='Vashi 2
1'"; con.query(sql, function (err,
    result){ if (err) throw err;
        console.log(result.affectedRows + "record(s) updated");
    }); });
```

OUTPUT:

Table:-newcustomer

Name: Jagtap Kalpana Santosh
Div : B 13

PrimaryKey1.js

```
var  mysql=require('mysql')  var
con=mysql.createConnection({
host:     "localhost",     user:
"root", password: "mydb"
});
con.connect(function(err){ if
    (err) throw err;
    console.log("Connected!");
    var sql= "CREATE TABLE customer (id INT AUTO_INCREMENT PRIMARY KEY, name V
ARCHAR(255),   address   VARCHAR(255))";
    con.query(sql, function(err,result){
    if(err) throw err;
        console.log("Table created");
}); });
```

OUTPUT:

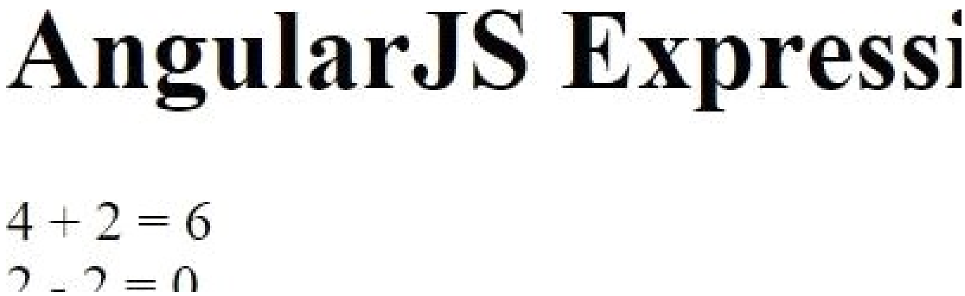
5. Angularjs

5.1 Write a program in angularJs of expression for operators and variables Solution:

```
<!DOCTYPE html>

<html >

<head>

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

</head>

<body >

  <h1>AngularJS Expression Demo:</h1>

  <div ng-app>

    4 + 2 = {{4 + 2}} <br />

    2 - 2 = {{2 - 2}} <br />

    2 * 2 = {{2 * 2}} <br />

    2 / 2 = {{2 / 2}}

  </div>

</body>
```

```
</html>
```

OUTPUT:



AngularJS Expressi

4 + 2 = 6
2 - 2 = 0

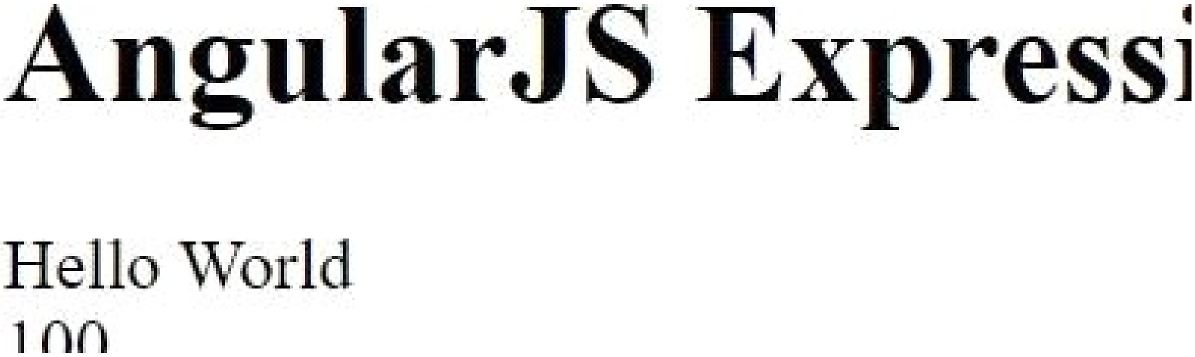
Write a program in angularJs of expression contains any two data type.

Solution:

```
<html >
<head>
   <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body >
   <h1>AngularJS Expression Demo:</h1>
   <div ng-app>
      {{"Hello World"}}<br />
      {{100}}<br /> {{true}}<br />
      {{10.2}}
   </div>
</body>
</html>
```

OUTPUT:

Name: Jagtap Kalpana Santosh
Div : B 13



AngularJS Expressi

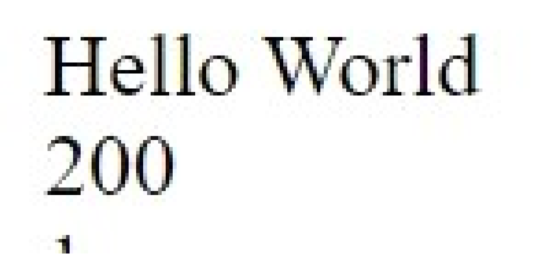
Hello World
100

of expression for arithmetic operators which will produce the result based on the type of operands.

Solution:

```
<!DOCTYPE html>
<html >
<head>
   <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body >
   <div ng-app>
      {{"Hello" + " World"}}<br />
      {{100 + 100 }}<br />
      {{true + false}}<br />
      {{10.2 + 10.2}}<br />
   </div>
</body>
</html>
```

OUTPUT:



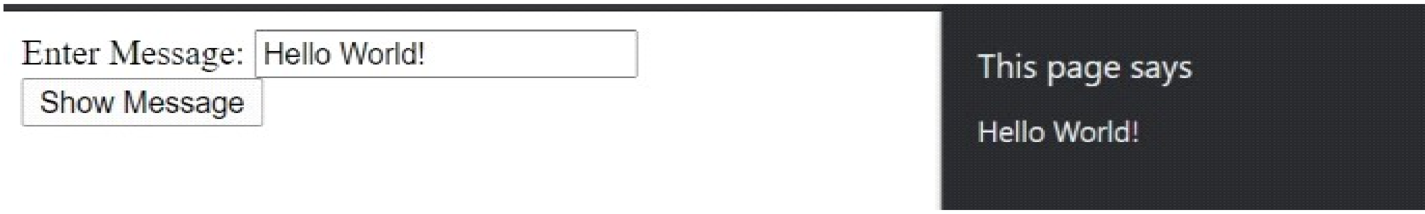
```
Hello World
200
1
```

Write a program in angularJs which demonstrates handling click event of a button

Solution:

```
<!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="myController">
    Enter Message: <input type="text" ng-model="message" /> <br />
    <button ng-click="showMsg(message)">Show Message</button>
  </div>
  <script>   var   ngApp   =   angular.module('myNgApp',   []);
    ngApp.controller('myController', function ($scope) {
      $scope.message = "Hello World!";
      $scope.showMsg = function (msg) { alert(msg);
      };
    });
  </script></body>
</html>
```

OUTPUT:



Enter Message: Hello World!

This page says

Hello World!

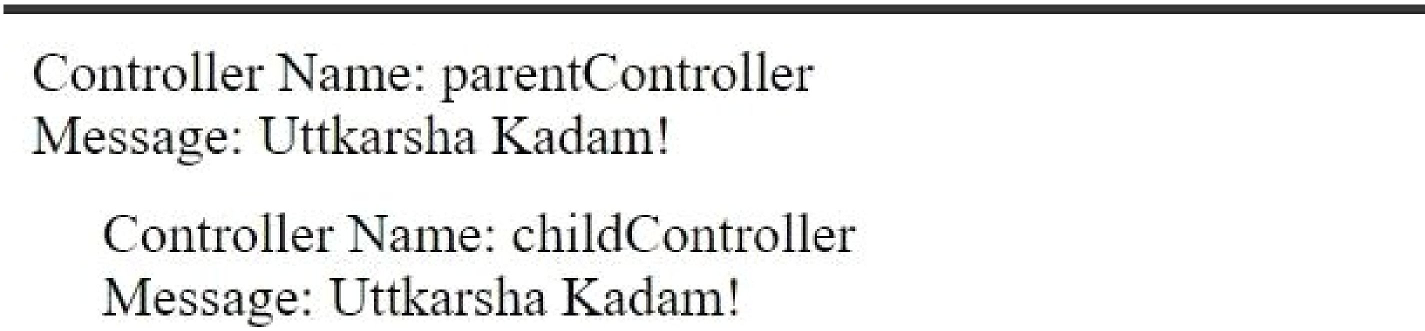
for scope object where controller available to the HTML elements and its child elements.

Solution:

```
<!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:10px 0 10px 20px;" 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 = "Uttkarsha Kadam!";
```

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


        });

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

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

OUTPUT:



```
Controller Name: parentController
Message: Uttkarsha Kadam!

    Controller Name: childController
    Message: Uttkarsha Kadam!
```

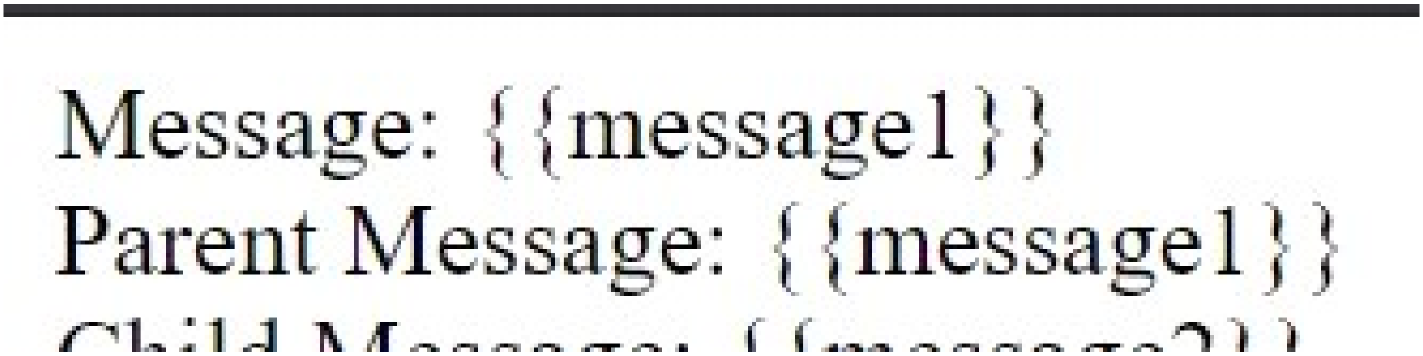
5.6 demonstrates multiple controllers.

Solution:

```
<!DOCTYPE html>
<html>
<head>
    <title>AngualrJS Controller</title>
    <script src="~/Scripts/angular.js"></script>
</head>
<body ng-app="myNgApp">
    <div ng-controller="parentController"> Message: {{message1}}
```

```
        <div ng-controller="childController">
           Parent Message: {{message1}} </br>
           Child Message: {{message2}} </div>
        Child Message: {{message2}}
    </div>
    <script> var ngApp = angular.module('myNgApp', []);
ngApp.controller('parentController', function ($scope) { $scope.message1 = "This is
        parentController";
        });
        ngApp.controller('childController', function ($scope) { $scope.message2 =
        "This is childController"; });
    </script>
</body>
</html>
```

OUTPUT:



```
Message: {{message1}}
Parent Message: {{message1}}
Child Message: {{message2}}
```

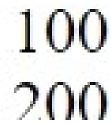
5.7 Write a program in angularJs to demonstrates ng-init directive for string, number, array, and object.

Solution: **<u>ng-init</u>**

```
<!DOCTYPE html>
<html>
<head>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>
<div ng-app ng-init="greet='Hello World!'; amount= 100; myArr = [100, 200]; person = {
firstName:'Uttkarsha', lastName :'Kadam'}"> {{amount}}            <br />
      {{myArr[1]}}    <br />
      {{person.firstName}}
   </div>
</body>
</html>
```

OUTPUT:



```
100
200
```

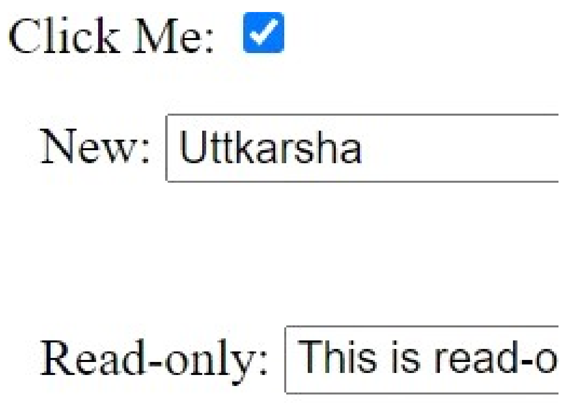
5.8 Write a program in angularJs to demonstrates ng-if, ng-readonly, and ng-disabled directives. Solution:

```
<!DOCTYPE html>
<html>
<head>
   <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
   <style> div {
         width: 100%; height: 50px;
         display:   block;   margin:
         15px 0 0 10px;

      }
   </style>
</head>
<body ng-app ng-init="checked=true" >
```

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

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

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

OUTPUT:

When click me is Checked:

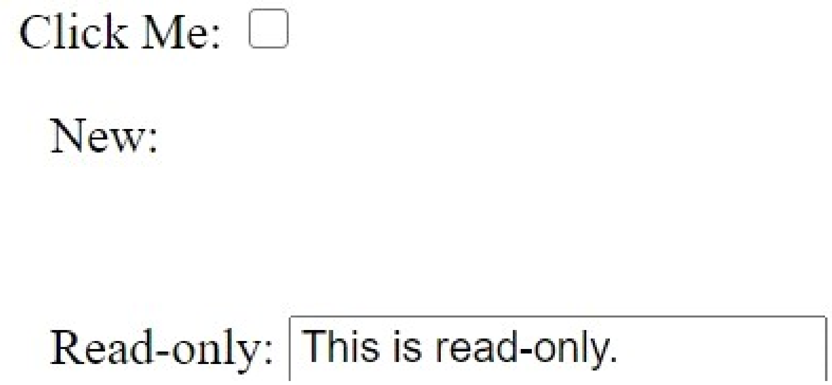


Click Me: ☑

New: Uttkarsha

Read-only: This is read-o

When Click me is not checked:



Click Me: □

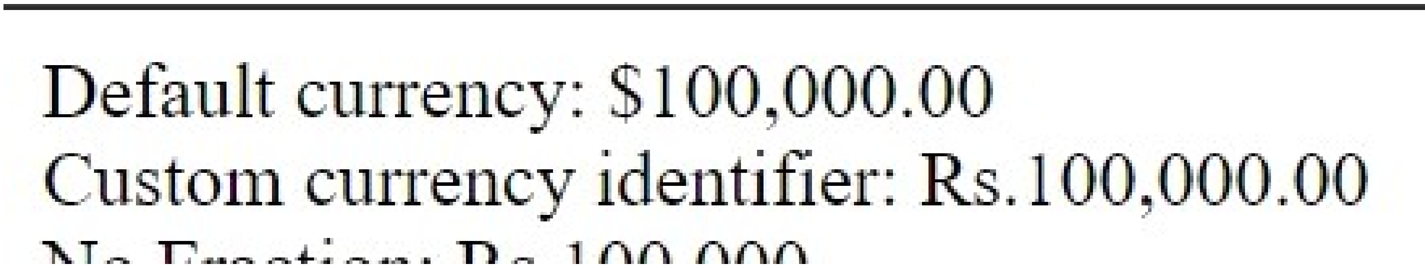
New:

Read-only: This is read-only.

Write a program in angularJs for currency filter to person salary. Solution:

```
<!DOCTYPE html>
<html >
<head>
  <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">
    Default currency: {{person.salary | currency}} <br />
    Custom currency identifier: {{person.salary | currency:'Rs.'}} <br />
    No Fraction: {{person.salary | currency:'Rs.':0}} <br />
    Fraction 2: <span ng-bind="person.salary| currency:'GBP':2"></span>
  </div><script> var myApp = angular.module('myApp', []);
    myApp.controller("myController", function ($scope) {
        $scope.person = { firstName: 'Uttkarsha', lastName: 'Kadam', salary: 100000}
    });
  </script>
</body>
</html>
```

OUTPUT:

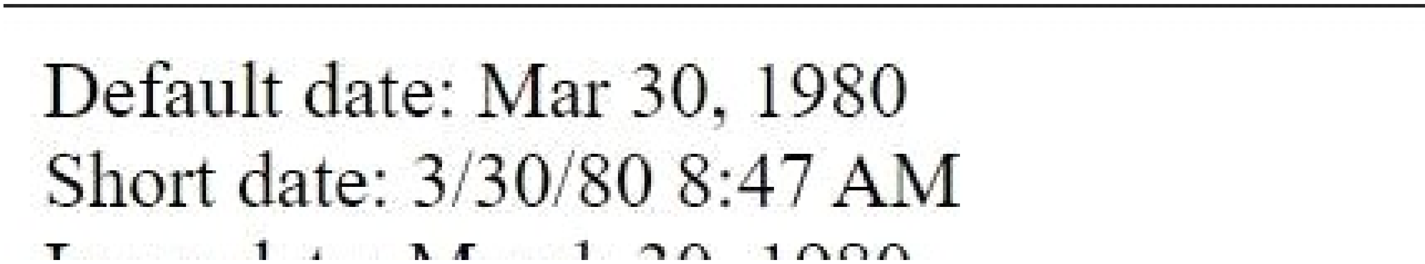


```
Default currency: $100,000.00
Custom currency identifier: Rs.100,000.00
No Fraction: Rs.100,000
```

Write a program in angularJs demonstrates Date filter Solution:

```
<!DOCTYPE html>
<html >
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-app>
  <div ng-init="person.DOB = 323234234898">
    Default date: {{person.DOB| date}} <br />
    Short date: {{person.DOB| date:'short'}} <br />
    Long date: {{person.DOB | date:'longDate'}} <br />
    Year: {{person.DOB | date:'yyyy'}} <br />
  </div>
</body>
</html>
```

OUTPUT:



```
Default date: Mar 30, 1980
Short date: 3/30/80 8:47 AM
Long date: March 30, 1980
```

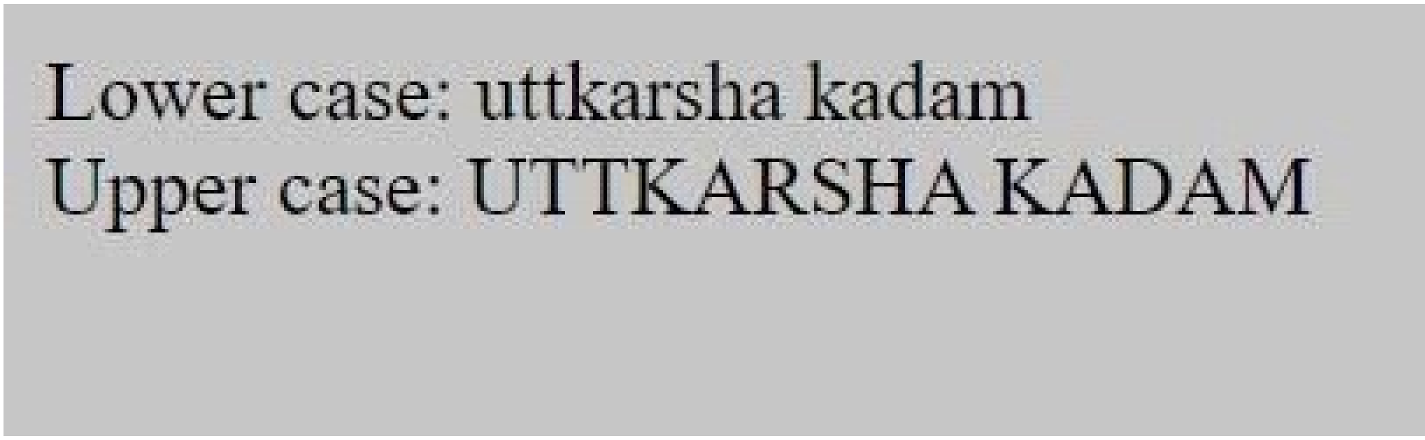
Write a program in angularJs upper case and lowercase filter.

Solution:

```
<!DOCTYPE html>
```

```
<html >
<head>
   <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-app>
<body style="background-color:silver">
   <div ng-init="person.firstName='Uttkarsha';person.lastName='Kadam'">
      Lower case: {{person.firstName + ' ' + person.lastName | lowercase}} <br />
      Upper case: {{person.firstName + ' ' + person.lastName | uppercase}} </div>
</body>
</html>
```

OUTPUT:



```
Lower case: uttkarsha kadam
Upper case: UTTKARSHA KADAM
```

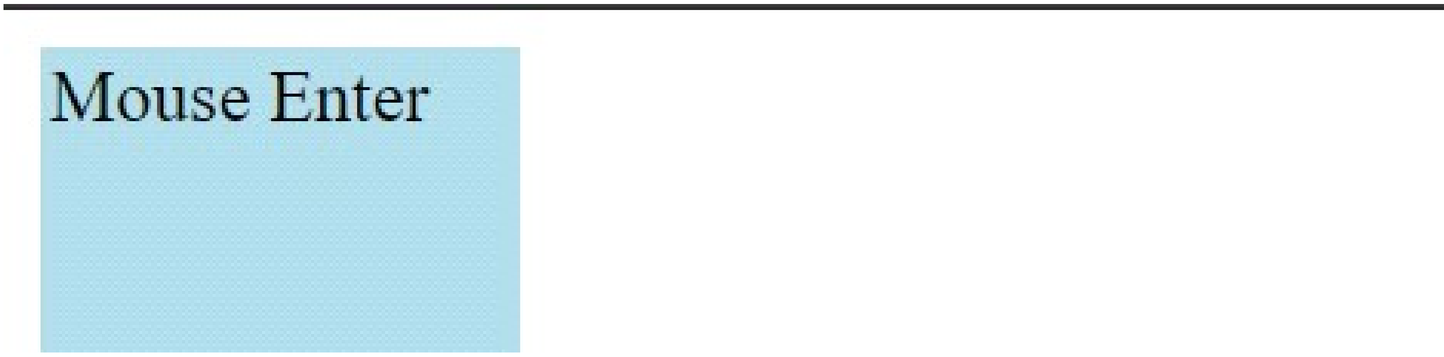
Write a program in angularJs to demonstrates mouse event.

Solution:

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

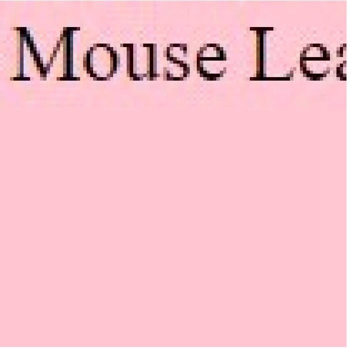
```
        .lightblueDiv   {   width:   100px;
           height:   100px;   background-
           color:  lightblue;  padding:2px
           2px 2px 2px;
        }
        .pinkDiv    {    width:    100px;
           height:  100px;  background-
           color:    pink;    padding:2px
         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

OUTPUT:



```
Mouse Enter
```

Name: Jagtap Kalpana Santosh
Div : B 13



Mouse Lea