**1)Write a node js program to create student database in MySQL, create table studentinfo, add 3 records and display it on terminal**

var mysql=require('mysql');

var con=mysql.createConnection({

host:"localhost",

user:"root",

password:"",

database:"student\_info"

});

con.connect(function(error)

{

if(error) throw error;

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

{

if(error) throw error;

console.log(result);

})

});

**2) Create an application to demonstrate angular filters like currency, uppercase, lowercase, filter, orderby**

orderby

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.16/angular.min.js"></script>

</head>

<body ng-app="myApp" >

<h1>AngularJS orderBy Filter Demo: </h1>

<div ng-controller="myController">

<select ng-model="SortOrder">

<option value="+name">Name (asc)</option>

<option value="-name">Name (dec)</option>

<option value="+phone">Phone (asc)</option>

<option value="-phone">Phone (dec)</option>

</select>

<ul ng-repeat="person in persons | orderBy:SortOrder">

<li>{{person.name}} - {{person.phone}}</li>

</ul>

</div>

<script>

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

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

$scope.persons = [{ name: 'Aishwarya', phone: '512-455-1276' },

{ name: 'Vaishnavi', phone: '899-333-3345' },

{ name: 'Payal', phone: '511-444-4321' },

{ name: 'Janhavi', phone: '145-788-5678' },

{ name: 'Aditi', phone: '433-444-8765' },

{ name: 'Ashwini', phone: '218-345-5678' }]

$scope.SortOrder = '+name';

});

</script>

</body>

</html>

**3)Write a node js program Using File Handling to demonstrate all basic file operations (Create, write, read, delete)**

var http=require('http');

var fs=require('fs');

http.createServer(function(req,res)

{

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

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

{

if(err)throw err;

console.log('File Saved');

});

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

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

{

if(err)throw err;

fs.close(fd,function(err)

{

if(err) throw err;

});

});

})

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

{

if(err) throw err;

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

res.write(data);

res.end();

})

}).listen(8086);

1. **Create an angular application to demonstrate ng-model directive**

<!DOCTYPE html>

<html>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

<body>

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

Name: <input ng-model="name">

<h1>You entered: {{name}}</h1>

</div>

<script>

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

app.controller('myCtrl', function($scope) {

$scope.name = "VAISHNAVI";

});

</script>

</body>

</html>

1. **Create an node js application to demonstrate event emitter**

demo\_file.js

var fs = require('fs');

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

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

console.log('The file is open');

});

event\_emitter.js

var events = require('events');

var eventEmitter = new events.EventEmitter();

var myEventHandler = function () {

console.log('I hear a scream!');

}

eventEmitter.on('scream', myEventHandler);

eventEmitter.emit('scream');

1. **Create an node js application to demonstrate file handling (any two operations)**

var http=require('http');

var fs=require('fs');

http.createServer(function(req,res)

{

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

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

{

if(err)throw err;

console.log('File Saved');

});

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

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

{

if(err)throw err;

fs.close(fd,function(err)

{

if(err) throw err;

});

});

})

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

{

if(err) throw err;

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

res.write(data);

res.end();

})

}).listen(8086);

**Write a node js program to create own module**

**Myfirstmodule.js**

exports.myDateTime = function () {

return Date();

};

**Demo\_module.js**

var http = require('http');

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

http.createServer(function (req, res) {

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

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

res.end();

}).listen(8080);

1. **Write a node js program to demonstrate http module**

**httpWbServer.js**

var http = require('http');

http.createServer(function (req, res) {

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

res.write('Node.js says hello!');

res.end();

}).listen(8087);

1. Write a node js program to demonstrate timer module

**setImmediate()**

setImmediate(function A() {

setImmediate(function B() {

console.log(1);

setImmediate(function D() {

console.log(2);

});

});

setImmediate(function C() {

console.log(3);

setImmediate(function E() {

console.log(4);

});

});

});

console.log('Started...');

**setInterval()**

// Executed after every 1000 milliseconds

// from the start of the program

setInterval(function A() {

return console.log('Hello World!');

}, 1000);

// Executed right away

console.log('Executed before A...');

**setTimeout()**

// Executed after 3000 milliseconds

// from the start of the program

setTimeout(function A() {

return console.log('Hello World!');

}, 3000);

// executed right away

console.log('Executed before A...');

**clearImmediate()**

var si = setImmediate(function A() {

console.log(1);

});

// clears setInterval si

clearImmediate(si);

console.log(2);)

**clearInterval()**

**var si = setInterval(function A() {**

**return console.log("Hello World!");**

**}, 500);**

**setTimeout(function() {**

**clearInterval(si);**

**}, 2000);  
 clearTimeout()**

// si1 is cleared by clearTimeout()

var si1 = setTimeout(function A() {

return console.log("Hello World!");

}, 3000);

// only si2 is executed

var si2 = setTimeout(function B() {

return console.log("Hello Geeks!");

}, 3000);

clearTimeout(si1);

1]Write a angular program to print Fibonacci Series Using Controllers in AngularJS.

<html>

  <head>

     <title>Fibonacci Series Using Controllers in AngularJS</title>

     <script type="text/javascript" src="angular.min.js"></script>

      <script>

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

        myApp.controller("Fibonacci\_Series",function($scope) {

            $scope.Fibonacci\_Generate=function()

            {

       $scope.result = "Fibonacci Series"

      var arr = [];

      let n1 = 0, n2 = 1, nextTerm;

    for (let i = 1; i <= $scope.a; i++) {

        arr.push(n1);

        document.getElementById("outputDiv").innerHTML = arr;

        nextTerm = n1 + n2;

        n1 = n2;

        n2 = nextTerm;

    }

  }

      $scope.Clear\_All=function()

          {

          $scope.a = "";

-------------------------------------------------------------------------------

          document.getElementById("outputDiv").innerHTML = "";

          $scope.result = "";

      }

     });

     </script>

 </head>

 <style>

body {

    font-family: arial;

    font-size: 25px;

    font-weight: bold;

}

</style>

 <body ng-app="myModule" ng-controller="Fibonacci\_Series">

     <h3>Fibonacci Series Using Controllers in AngularJS

     </h3>

 <div>

    <table border="0">

         <tr>

            <td>

            Give a Positive Number

         </td>

         <td>

           <input type="number" ng-model="a"/>

         </td>

        <tr>

      <tr>

         <td colspan="10">

             <br>

             <input type="button" ng-click="Fibonacci\_Generate();"

             value="Generate Fibonacci Series"/>

           <input type="button" ng-click="Clear\_All();"

           value="Clear"/>

         </td>

      </tr>

      </table>

     </div>

     <p> {{result}} </p>

    <textarea id="outputDiv" name="outputDiv" rows="4" cols="50">

</textarea>

  </body>

</html>

2] Write a angular program to demonstrate filters.

<html>

    <head>

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

    </head>

    <body>

        <div ng-app="">

            <input type="text" ng-model="name1"placeholder="Enter your name"><br>

            <h1>Output:{{name1 |lowercase}}</h1>

            <input type="text" ng-model="name2"placeholder="Enter your name"><br>

            <h1>Output:{{name2 |uppercase}}</h1>

            <input type="text" ng-model="name3"placeholder="Enter currency"><br>

            <h1>Output:{{name3 |currency}}</h1>

        </div>

    </body>

</html>

3] Create an angular application to demonstrate $scope object as an argument.

<html>

    <head>

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

    </head>

    <body>

        <div ng-app="firstapp" ng-controller="ctr11">

        <ul>

        <li ng-repeat="x in students |filter:'a'">

        {{x}}

        </li>

        </ul>

        </div>

        <script>

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

        demoapp.controller("ctr11",function($scope){

            $scope.students=[

            'aditi','aishwarya','Vaishnavi','Payal','Janhavi'

        ];

        });

        </script>

    </body>

</html>

4] Create an angular application to demonstrate ng-model directive.

<html>

    <head>

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

    </head>

    <body>

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

        <input type="text" ng-model="firstname"><br>

        <input type="text" ng-model="lastname"><br>

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

        </div>

        <script>

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

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

            $scope.firstname="Aditi";

            $scope.lastname="Shinde";

        });

        </script>

    </body>

</html>

ii.)Write a angular program to create array and display its elements with the help of ng-init

<html>

<head>

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

</head>

<body>

<div ng-app="" ng-init="students=['Aditi','Aishwarya','Vaishnavi']">

<h1>Total:{{students[1] }}</h1>

<h1 ng-repeat="x in students">Name : {{x}} </h1>

</div>

</body>

</html>

ii.) to demonstrate uppercase, lowercase, orderby filter

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.16/angular.min.js"></script>

</head>

<body ng-app="myApp" >

<body>

<div ng-app="">

<input type="text" ng-model="name1"placeholder="Enter your name"><br>

<h1>Output:{{name1 |lowercase}}</h1>

<input type="text" ng-model="name2"placeholder="Enter your name"><br>

<h1>Output:{{name2 |uppercase}}</h1>

<input type="text" ng-model="name3"placeholder="Enter currency"><br>

<h1>Output:{{name3 |currency}}</h1>

</div>

</body>

<h1>Student List: </h1>

<div ng-controller="myController">

<select ng-model="SortOrder">

<option value="+name">Name (asc)</option>

<option value="-name">Name (dec)</option>

<option value="+phone">Phone (asc)</option>

<option value="-phone">Phone (dec)</option>

</select>

<ul ng-repeat="person in persons | orderBy:SortOrder">

<li>{{person.name}} - {{person.phone}}</li>

</ul>

</div>

<script>

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

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

$scope.persons = [{ name: 'Aishwarya', phone: '512-455-1276' },

{ name: 'Vaishnavi', phone: '899-333-3345' },

{ name: 'Payal', phone: '511-444-4321' },

{ name: 'Janhavi', phone: '145-788-5678' },

{ name: 'Aditi', phone: '433-444-8765' },

{ name: 'Ashwini', phone: '218-345-5678' }]

$scope.SortOrder = '+name';

});

</script>

</body>

</html>

1. to demonstrate Filter an Array Based on User Input

<html lang="en">

<head>

<meta charset="utf-8" />

<title>GeeksforGeeks</title>

<script src=

"https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js">

</script>

</head>

<body>

<div style="text-align: center">

<h1 style="color: green">GeeksforGeeks</h1>

<h3>Filtering an Array in AngularJS</h3>

</div>

<div>

<div ng-app="gfgApp"

ng-controller="langsCtrl">

<p>Search any word in input field:</p>

<p>

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

</p>

<ul>

<li ng-repeat=

"word in lang | filter:test">

{{ word }}

</li>

</ul>

</div>

<script>

angular.module("gfgApp", []).controller("langsCtrl",

function ($scope) {

$scope.lang = ["Java","C++","Python",

"Go","C","SQL","JavaScript",

];

});

</script>

</div>

</body>

</html>

i.)to demonstrate ng-repeat directive

<html>

<head>

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

</head>

<body>

<div ng-app="" ng-init="students=['Aditi','Aishwarya','Vaishnavi']">

<h1>Total:{{students[1] }}</h1>

<h1 ng-repeat="x in students">Name : {{x}} </h1>

</div>

</body>

</html>

ii.)to demonstrate ng-app directive

<html>

<head>

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

</head>

<body>

<div ng-app="" ng-init="products=3;rate=24">

<h1>Total:{{products\*rate}}</h1>

</div>

</body>

</html>

**OR**

<html>

<head>

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

</head>

<body>

<div ng-app="" ng-init="products=3;rate=24">

<h1>Total:{{products\*rate}}</h1>

</div>

</body>

</html>

iii.)to demonstrate ng-model directive

<html>

<head>

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

</head>

<body>

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

<input type="text" ng-model="firstname"><br>

<input type="text" ng-model="lastname"><br>

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

</div>

<script>

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

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

$scope.firstname="Aditi";

$scope.lastname="Shinde";

});

</script>

</body>

</html>

1. Create your own module in node js to do arithmetic operations

Function.js

exports.function\_name = function(arg1, arg2, ....argN) {

    // function body

};

Calc.js

exports.add = function (x, y) {

    return x + y;

};

exports.sub = function (x, y) {

    return x - y;

};

exports.mult = function (x, y) {

    return x \* y;

};

exports.div = function (x, y) {

    return x / y;

};

App.js

var calculator = require('./calc');

var x = 50, y = 20;

console.log("Addition of 50 and 20 is "

                   + calculator.add(x, y));

console.log("Subtraction of 50 and 20 is "

                   + calculator.sub(x, y));

console.log("Multiplication of 50 and 20 is "

                   + calculator.mult(x, y));

console.log("Division of 50 and 20 is "

                   + calculator.div(x, y));

1. **Write a angular program to create object student with the help of ng-init directive and display its property with help of angular js expression**

<!DOCTYPE html>

<html ng-app="">

<head>

<meta charset="UTF-8">

<title>AngularJS ng-init Example</title>

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

</head>

<body>

<div ng-init="student = {name: 'John', age: 21, gender: 'male'}">

<p>Name: {{student.name}}</p>

<p>Age: {{student.age}}</p>

<p>Gender: {{student.gender}}</p>

</div>

</body>

</html>

1. **Write a angular program to take input from user for the choice of background color of input field as well as background color of page and change color of both body and input field as per user choice with the help of ng-model directive.**

<!DOCTYPE html>

<html ng-app="">

<head>

<meta charset="UTF-8">

<title>AngularJS ng-model Example</title>

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

<style>

body {

background-color: {{bgColor}};

}

input[type=text] {

background-color: {{inputColor}};

}

</style>

</head>

<body ng-init="bgColor = 'white'; inputColor = 'white'">

<div>

<label for="bgColorInput">Choose a background color for the page:</label>

<input type="text" id="bgColorInput" ng-model="bgColor">

</div>

<div>

<label for="inputColorInput">Choose a background color for the input field:</label>

<input type="text" id="inputColorInput" ng-model="inputColor">

</div>

</body>

</html>