Refreshing Java Script and CSS

1. Write an HTML & CSS code to create Horizontal and Vertical Menu.

```
<!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>Menu</title>
    <style>
        h1 {
            color:rgb(49, 100, 100);
        }
        ul {
            list-style-type: none;
            margin: 0;
            padding: 0;
            overflow: hidden;
            background-color: rgb(249, 212, 0);
        }

        .nav1{
            float: left;
        }
        li:hover {
            background-color:rgb(157, 156, 87);
        }
        .hnav {
            display: inline-block;
            color: rgb(0, 0, 0);
            padding: 12px 12px;
            text-decoration: none;
        }

        .vnav {
            display: block;
            color: rgb(0, 0, 0);
            padding: 12px 12px;
            text-decoration: none;
        }

    </style>
```

```
</head>
<body>
    <div>
        <h1>Horizontal Menu</h1>
        <ul>
            <li class="nav1"><a href="#" class="hnav">Home</a></li>
            <li class="nav1"><a href="#" class="hnav">About</a></li>
            <li class="nav1"><a href="#" class="hnav">Contact</a></li>
            <li class="nav1"><a href="#" class="hnav">Login</a></li>
        </ul>
    </div>
<br>
    <div>
        <h1>Vertical Menu</h1>
        <ul>
            <li><a href="#" class="vnav">Home</a></li>
            <li><a href="#" class="vnav">About</a></li>
            <li><a href="#" class="vnav">Contact</a></li>
            <li><a href="#" class="vnav">Login</a></li>
        </ul>
    </div>
</body>
</html>
```

Output:-

2. Write JavaScript code for loop that will iterate from 0 to 15 for each iteration, it will check if the current number is odd or even, and display a message to the screen.

```
<!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 type="text/javascript">
        var i;
        for(i=0; i<=15; i++)
        {
            if(i==0)
            {
                document.writeln('<br>'+i+' '+'is Even');
            }
            else if(i%2==0)
            {
                document.writeln('<br>'+i +' '+'is Even');
            }
            else
            {
                document.writeln('<br>'+i+' '+'is Odd');
            }
        }
    </script>
</head>
<body>
</body>
</html>
```

Output:-

0 is Even
1 is Odd
2 is Even
3 is Odd
4 is Even
5 is Odd
6 is Even
7 is Odd
8 is Even
9 is Odd
10 is Even
11 is Odd
12 is Even
13 is Odd
14 is Even
15 is Odd

3. Write an HTML and JavaScript program which accepts N as input and display first N Fibonacci numbers as list.

```
<!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>Fibonacci Series</title>
</head>
<body>
    <script type="text/javascript">
        function fibo() {
            var n1=0, n2=1, nextnum, i;
            var N = document.getElementById('num').value;
            for(i=0; i<N; i++) {
                document.getElementById('result').innerHTML += (" "+n1); //for loop
overwrite innerHTML in each iteration so we use +=(compound operator)
                nextnum=n1+n2;
                n1=n2;
                n2=nextnum;
             }
        }
    </script>
    <h1>Fibonacci Series(200210116006)</h1>
    <label for="num">Enter Term:</label>
    <input type="number" id="num">
    <input type="button" value="Submit" onclick="fibo()">
    <br><br>
    <p>Fibonacci Series:<span id="result"></span></p>
</body>
</html>
```

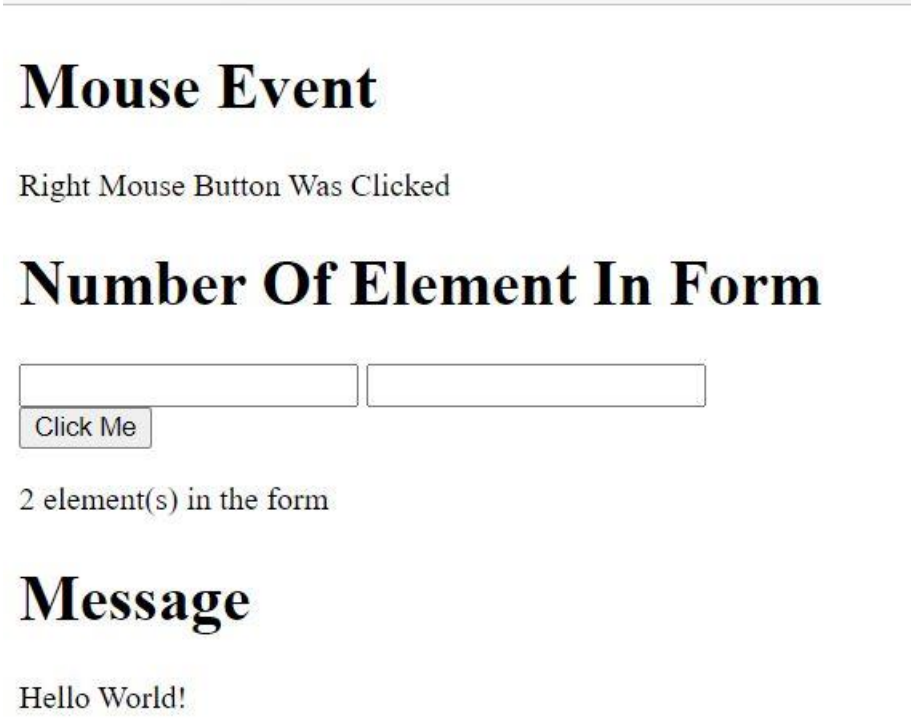
Output:-

4. Write JavaScript code to know which mouse button was clicked, number of elements in form, and write hello world on the document.

```
<!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 type="text/javascript">
        document.onmousedown = mouse;
     function mouse(event) {
         if(event.button == 0) {
             document.getElementById("result").innerHTML = "Left Mouse Button Was
Clicked.";
         }
         if(event.button == 1){
             documnet.getElemntById("result").innerHTML = "Middle Mouse Button Was
Clicked";
         }
         if(event.button == 2){
             document.getElementById("result").innerHTML = "Right Mouse Button Was
Clicked";
         }
     }

     function num() {
        var x = document.getElementById("myform").elements.length;
        document.getElementById("result2").innerHTML = x +" "+ "element(s) in the form";
     }

    //  document.getElementById("result3").innerHTML = "Hello World!";
 </script>
</head>
<body>
    <h1>Mouse Event</h1>
    <p id="result"></p>


    <h1>Number Of Element In Form</h1>
    <form id="myform" action="">
        <input type="text">
        <input type="number">
    </form>
    <button onclick="num()">Click Me</button>
```

```
    <p id="result2"></p>

    <h1>Message</h1>
    <p id="result3"></p>
    <script type="text/javascript">
             document.write("Hello World!");
    </script>
</body>
</html>
```

Output:-



5. Write JavaScript code to check mobile number (mobile no. should start with 9 or 8).

```
<!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 type="text/javascript">
        function valid() {
                var x = document.getElementById('mobile').value;
                if (x.charAt(0) == 9 || x.charAt(0) == 8) {
                    document.getElementById('error').innerHTML ="*Mobile Number should
not start with" + " " + x.charAt(0);
                }
                else {
                    document.getElementById('error').innerHTML = "Mobile Number is
valid";
                }
            }
    </script>
</head>
<body>
    <label for="mobile">Mobile Number:</label>
    <input id="mobile" type="tel" maxlength="10">
    <span id="error"></span>
    <br>
    <input type="button" value="Submit" onclick="valid()">
</body>
</html>
```

Output:-

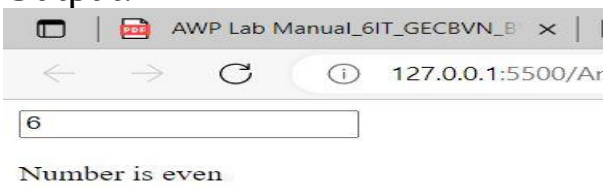
Angular JS Program

1. Write an AngularJS code which takes number as an input and display that number is odd or even.

Code:-

```
<html>
<head>
  <title>OddEven</title>
  <script src="../angular-1.8.2/angular.min.js"></script>
  <script>
    var app = angular.module("myapp",[]);
    app.controller("myctrl", function($scope)
    {
      $scope.oddeven=function(result)      // first we use only num so we got error
      {
        if(result == null)
        {
          return(document.getElementById('output').innerHTML="<br>"+"*Please Enter a
number to check number is Odd or Even");
        }
        else if(result % 2 == 0)
        {
          return(document.getElementById('output').innerHTML="<br><br>"+"Number is
even");}
        else {
          return(document.getElementById('output').innerHTML="<br><br>"+"Number is
Odd");}}});
  </script>
</head>
<body>
    <div ng-app="myapp" ng-controller="myctrl">
      <input type="number" ng-model="num">
      <!-- {{num}} -->
      <span id="output" ng-bind="oddeven(num)"></span>
      <!-- {{oddeven()}}   expression -->
    </div>
</body>
</html>
```

Output:-



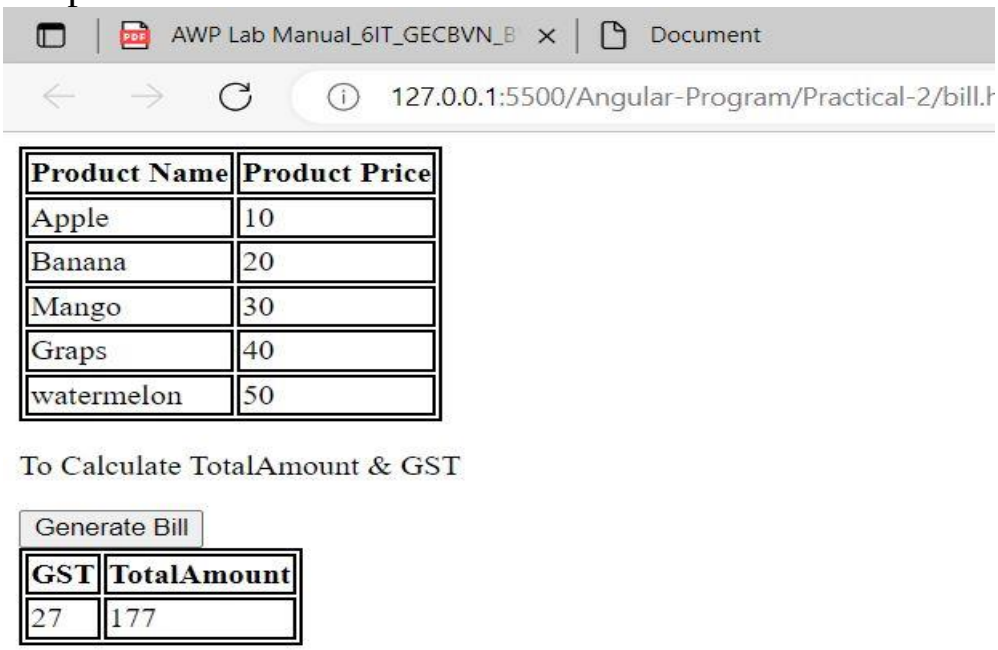
2. Design Order Form with a total price updated in real time, which contains name of five products and their prices. Create a bill amount for all the products and calculate GST on the billing amount and display total amount.

Code:-

```
<!DOCTYPE html>
<html lang="en" ng-app="myapp">
<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>
    <style>
        table, th, td{
            border: 2px solid black;
        }
    </style>
    <script src="../angular-1.8.2/angular.min.js"></script>
<script>
        var app=angular.module("myapp",[]);
        app.controller("myctrl", function($scope)
        {
            $scope.product = [
                {name: "Apple", price: 10},
                {name: "Banana", price: 20},
                {name: "Mango", price: 30},
                {name: "Graps", price: 40},
                {name: "watermelon", price: 50}
            ]
            $scope.totalamount=0;
            $scope.gst=0;

            $scope.bill=function() {
                var total=0;
                for(var i=0; i< $scope.product.length; i++)
                {
                    var p = $scope.product[i];
                    total += p.price;
                }

                $scope.gst = total * 0.18;
                $scope.totalamount = total + $scope.gst;;
            }
        });
    </script>
</head>
```

```
<body ng-controller="myctrl">
    <table>
        <tr>
            <th>Product Name</th>
            <th>Product Price</th>
        </tr>
        <tr ng-repeat="p in product">
            <td>{{p.name}}</td>
            <td>{{p.price}}</td>
        </tr>
    </table>

    <p>To Calculate TotalAmount & GST</p>
    <button ng-click="bill()">Generate Bill</button>
    <table>
        <tr>
            <th>GST</th>
            <th>TotalAmount</th>
        </tr>

        <tr>
            <td>{{gst}}</td>
            <td>{{totalamount}}</td>
        </tr>
    </table>

</body>
</html>
```

Output:-

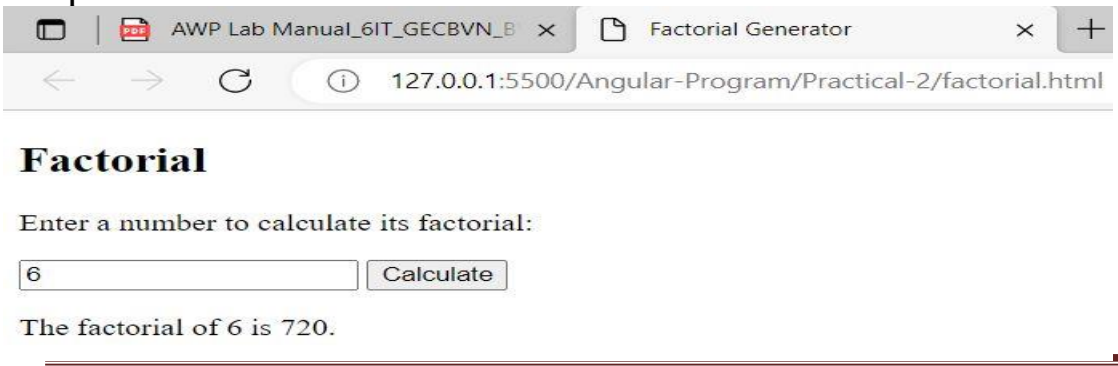


3. Design a webpage which takes one number as an input and generate its factorial number (use module, controller)

Code:-

```
<!DOCTYPE html>
<html lang="en" ng-app="myApp">
<head>
    <meta charset="UTF-8">
    <title>Factorial Generator</title>
    <script src="../angular-1.8.2/angular.min.js"></script>
    <script>
        var myApp = angular.module('myApp', []);
        myApp.controller('myctrl', function($scope) {
            $scope.num = null;
            $scope.result = null;
            $scope.factorial = function() {
                var num = $scope.num;
                var result = 1;

                for (var i = 2; i <= num; i++) {
                    result *= i;
                }
                $scope.result = result;
            }
            });
    </script>
</head>
<body ng-controller="myctrl">
    <h2>Factorial</h2>
    <p>Enter a number to calculate its factorial:</p>
    <input type="number" ng-model="num">
    <button ng-click="factorial()">Calculate</button>
    <p ng-if="result !== null">The factorial of {{num}} is {{result}}.</p>
</body>
</html>
```

Output:-



4. Design a webpage which takes inputs product name, product quantity and price. Generate table of entered values. When user clicks on table column title, it should sort that column values.(use filter, array)

Code:-

```
<html ng-app="myapp" >
<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>Product</title>
    <style>
    table,th, td{
            border: 1px solid black;
        }
        /* button {
            border: none;
        } */
    </style>
     <script src="../angular-1.8.2/angular.min.js"></script>
     <script>
        var app=angular.module("myapp",[]);
        app.controller("myctrl",function($scope, $filter){

            $scope.product = [];
            $scope.add = function(){
                   var pname=$scope.pname;
                   var quantity=$scope.quantity;
                   var price=$scope.price;
                   var arr = {
                    productname: pname,
                    productquantity: quantity,
                    productprice: price
                   };
                   $scope.product.push(arr);
                   $scope.pname="";
                   $scope.quantity="";
                   $scope.price="";
            };
            $scope.sortColumn = "productname";
            $scope.reverseSort=false;

            $scope.sortdata = function(column) {
                $scope.reverseSort = ($scope.sortColumn == column) ? !$scope.reverseSort
: false;
                $scope.sortColumn = column;
```

```
                $scope.product = $filter('orderBy')($scope.product, $scope.sortColumn,
$scope.reverseSort);
            };
        });
    </script>
</head>
<body ng-controller="myctrl">
    <form>
        <label for="name1">Product Name:</label>
        <input type="text" id="name1" ng-model="pname">
        <br>
        <label for="name2">Product Quantity:</label>
        <input type="number" id="name2" ng-model="quantity">
        <br>
        <label for="name3">Produc price:</label>
        <input type="number" id="name3" ng-model="price">
        <br>
        <button type="button" ng-click="add()">Add Data</button>
    </form>
        <table>
            <tr>
                <th ng-click="sortdata('name')">Product Name</th>
                <th ng-click="sortdata('quantity')">Product Quantity</th>
                <th ng-click="sortdata('price')">Product Price</th>
            </tr>

            <tr ng-repeat="p in product | orderBy:sortColumn:reverseSort">
                <td>{{ p.productname }}</td>
                <td>{{ p.productquantity }}</td>
                <td>{{ p.productprice }}</td>
            </tr>
        </table>
    </div>
</body>
</html>
```

Output:-

5. Design a webpage which display product name and product price using AngularJS $http Service from database. Display the content in tabular format.

Code:-

Database.html

```
<!DOCTYPE html>
<html>
<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>Database</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
</head>
<body ng-app="myapp"">
    <div ng-controller="myctrl">
        <table>
            <tr>
                <th>Product Name</th>
                <th>Product Price</th>
            </tr>
        <tr ng-repeat="product in productdata track by $index">
            <td>{{product.name}}</td>
            <td>{{product.price}}</td>
            </tr>
    </table>
    </div>

</body>
<script>
    var app = angular.module('myapp', []);
        app.controller("myctrl",function($scope, $http) {
            $http.get("product.php").then(function(response) {
                $scope.productdata = response.data;
            });
        });
</script>
</html>
```

product.php

```php
<?php
//create database connection
$host='localhost';
$username = 'root';
$password = '';
$dbname = 'mydb';

$con = mysqli_connect($host,$username,$password,$dbname);
if (!$con) {
    die('Could not connect: ' . mysqli_connect_error());
}
//fetch product data from database
$sql = "SELECT * FROM products";
$result = mysqli_query($con,$sql);
$data = array();
//put product data into array
if(mysqli_num_rows($result) > 0)
{
    while($row = mysqli_fetch_assoc($result))
    {
        $data[] = $row;
    }
    echo json_encode($data);
}
?>
```

Output:-

Product Name	Product Price
Graphs	50
Mango	40
Orange	30
Banana	10
guava	60

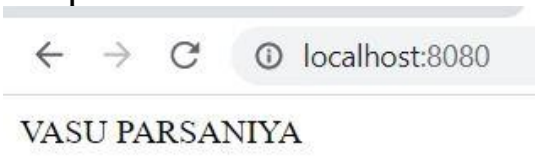
Node JS Program

1. Create a Node.js file that will convert the output "Hello World!" into upper-case letters.

Code:-

```
var http = require('http');
var upper=require('upper-case');

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write(upper.upperCase("Vasu Parsaniya"));
  res.end();
}).listen(8080);

// console.log("HelloWorld!");
```

Output:-



2. Write a Node.js module of calculator which will perform all the basic operations like add(), sub(), mul() and div().Use the module in a program and display the output.

Code:-

calculator.js

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

  sub: function(a, b) {
    return a - b;
  },

  mul: function(a, b) {
    return a * b;
  },

  div: function(a, b) {
    if (b === 0) {
      throw new Error('Division by zero is not allowed.');
    }
    return a / b;
  }
};
```

main.js

```
const calculator = require('./calculator');

console.log(calculator.add(2, 3));
console.log(calculator.sub(5, 3));
console.log(calculator.mul(2, 3));
console.log(calculator.div(6, 3));
console.log(calculator.div(6, 0));
```

Output:-

```
PS G:\Sem-6\Lab\Solution\AWP\Nodejs-Program\Pra-2> node main.js
5
2
6
2
G:\Sem-6\Lab\Solution\AWP\Nodejs-Program\Pra-2\calculator.js:43
      throw new Error('Division by zero is not allowed.');
      ^
```

3. Create a Node.js Application that uses user defined module circle.js which exports functions area() and circumference() and display details on console.

Code:-

circle.js

```
module.exports = {
    area: function(r) {
        return (22/7)*r*r;
    },

    circumference: function(r) {
        return 2*(22/7)*r;
    }
}
```

main.js

```
const circle = require('./circle');

console.log(circle.area(2));
console.log(circle.circumference(4));
```

Output:-

```
PS G:\Sem-6\Lab\Solution\AWP\Nodejs-Program\Pra-3> node main.js
12.571428571428571
25.142857142857142
```

4. Create a Node.js program to perform file operations like create a file, read a file, write to file and delete a file.

Code:-

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

// Create a file
fs.writeFile('example.txt', 'Vasu Parsaniya', (err) => {
  if (err) throw err;
  console.log('File created!');
});

// Read a file
fs.readFile('example.txt', 'utf8', (err, data) => {
  if (err) throw err;
  console.log('File contents:', data);
});

// Write to a file
fs.appendFile('example.txt', '\nThis is a new line.', (err) => {
  if (err) throw err;
  console.log('Data written to file!');
});

// Delete a file
fs.unlink('example.txt', (err) => {
  if (err) throw err;
  console.log('File deleted!');
});
```

Output:-

```
PS G:\Sem-6\Lab\Solution\AWP\Nodejs-Program\Pra-4> node .\file_curd.js
File deleted!
File created!
Data written to file!
File contents: Vasu Parsaniya
PS G:\Sem-6\Lab\Solution\AWP\Nodejs-Program\Pra-4>
```

5. Write Write Node.js code to perform Insert, update and delete operations on Employee database using Node.js and MongoDB.

Code:-

```
const { MongoClient} = require('mongodb');

const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);

async function createNewDatabase(dbName) {
  try {
    await client.connect().then(() => console.log("Connected With Server")).catch(err =>
console.log("Unable to connect"));
    var db = client.db(dbName);
    // var collection = db.createCollection('employeeCollection');
    console.log(`Created new database '${dbName}'`);
  } catch (err) {
    console.error(err);
  } finally {
    await client.close();
  }
}
//--------------------Insert data-----------------------------------------
async function insertData(dbName, collectionName) {

  //establish the connection with MongoDB database
  await client.connect().then(() => console.log("Again Connected")).catch(err =>
console.log("Unable to connect to database"));

  //We take database reference
  const myDB = client.db(dbName);
  /* we cannot use here createCollection() because it create only one time collection if
second time same code
  we run then give error So we us collection().
  -->if collection is not created then automatically collection() can create
collection*/

  const myColl = myDB.collection(collectionName);

  try {
    const data = [
      { name: 'Vasu', age: 20 },
      { name: 'Ashneer', age: 30 },
      { name: 'Piyush', age: 40 },
```

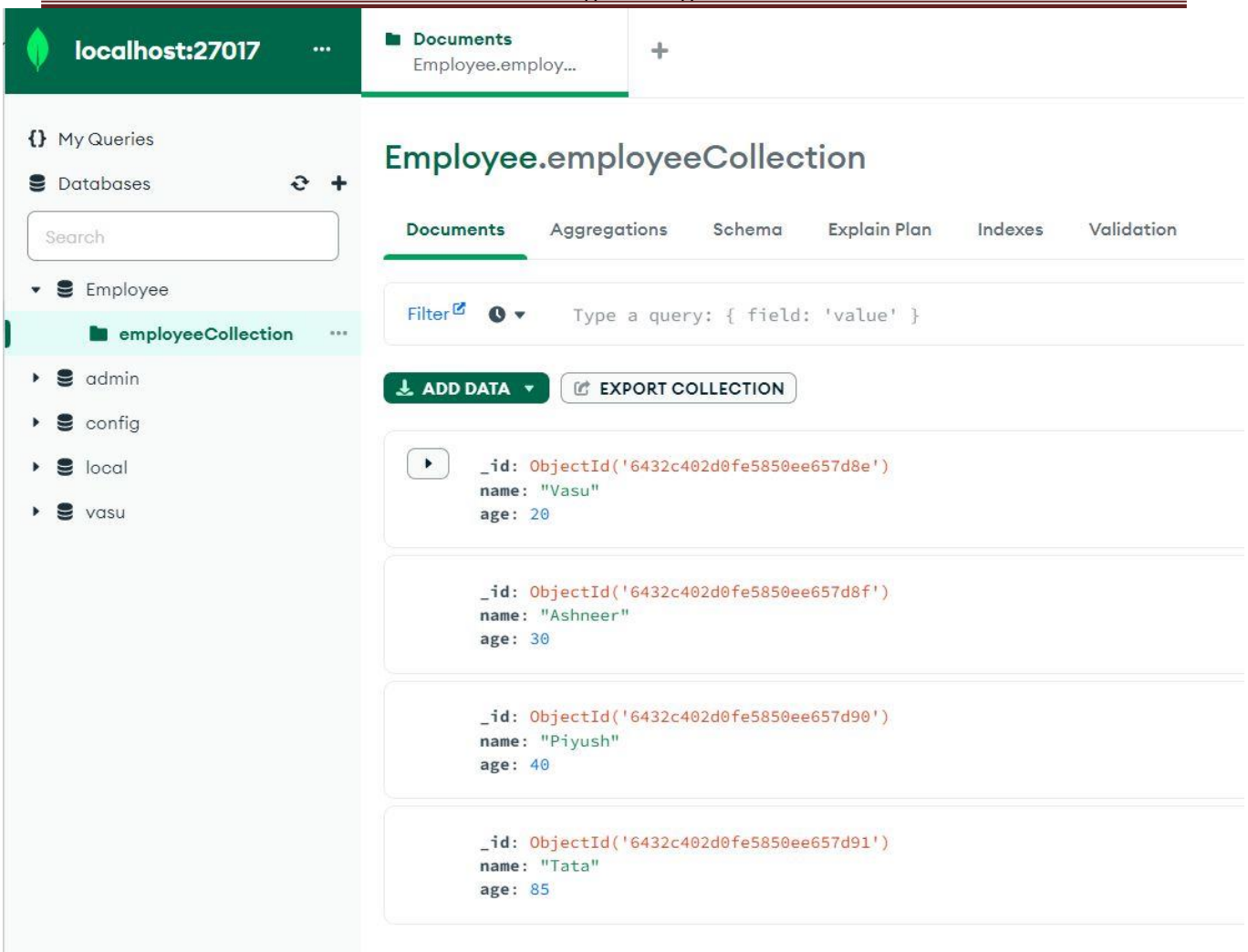
```
      { name: 'Tata', age: 85 }
    ];

    const insertManyresult = await myColl.insertMany(data);
    let ids = insertManyresult.insertedIds;
    console.log(`${insertManyresult.insertedCount} documents were inserted.`); //count
the number of inserted document
    for (let id of Object.values(ids)) {
      console.log(`Inserted a document with id ${id}`);
    }
  } catch (e) {
    console.log(e);
    // console.log(`A MongoBulkWriteException occurred, but there are successfully
processed documents.`);

    let ids = e.result.result.insertedIds;
    for (let id of Object.values(ids)) {
      console.log(`Processed a document with id ${id._id}`);
    }
    console.log(`Number of documents inserted: ${e.result.result.nInserted}`);
  } finally {
    await client.close();
  }
}


//----------------------Update data-------------------------------------


async function updateData(dbName, collectionName) {

  //establish the connection with MongoDB database
  try {
    await client.connect().then(() => console.log("Again Again Connected")).catch(err =>
console.log("Unable to connect to database"));

    const myDB = client.db(dbName);
    const myColl = myDB.collection(collectionName);
    const filter = { name: 'Vasu' };
    // update the value of the 'quantity' field to 5
    const updateDocument = {
      $set: { age: 16 },
    };
    const result = await myColl.updateOne(filter, updateDocument);
  } catch (e) {
    console.log(e);
  } finally {
    await client.close();
  }
```

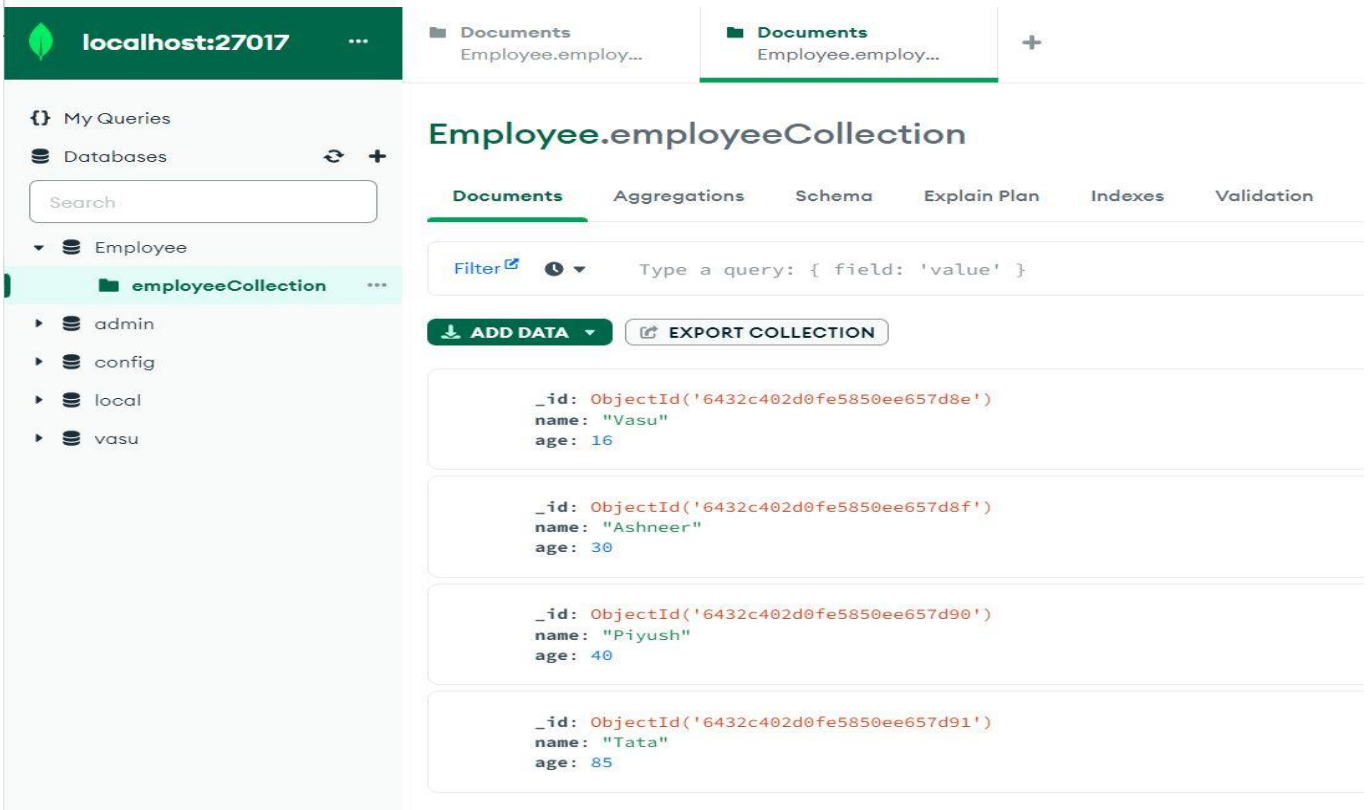
```
}
//---------------Delete data-------------------------------------

async function deleteData(dbName, collectionName) {

  //establish the connection with MongoDB database
  try {
    await client.connect().then(() => console.log("Again Again Again
Connected")).catch(err => console.log("Unable to connect to database"));

    const myDB = client.db(dbName);
    const myColl = myDB.collection(collectionName);

    const deleteData = { name: 'Vasu', age: 16 };

    const result = await myColl.deleteOne(deleteData);   //deleteOne() take parameter as
Object like JSON, javascript object
  } catch (e) {
    console.log(e);
  } finally {
    await client.close();
  }
}

createNewDatabase('Employee').then(() => console.log("Database Created")).catch(err =>
console.log("Unable to Create Database"));
insertData('Employee', 'employeeCollection').then(() =>
console.log("Inserted")).catch(err => console.log("Unable to Insert"));
updateData('Employee', 'employeeCollection').then(() =>
console.log("Updated")).catch(err => console.log("Unable to Update"));
deleteData('Employee', 'employeeCollection').then(() =>
console.log("Deleted")).catch(err => console.log("Unable to Update"));
```

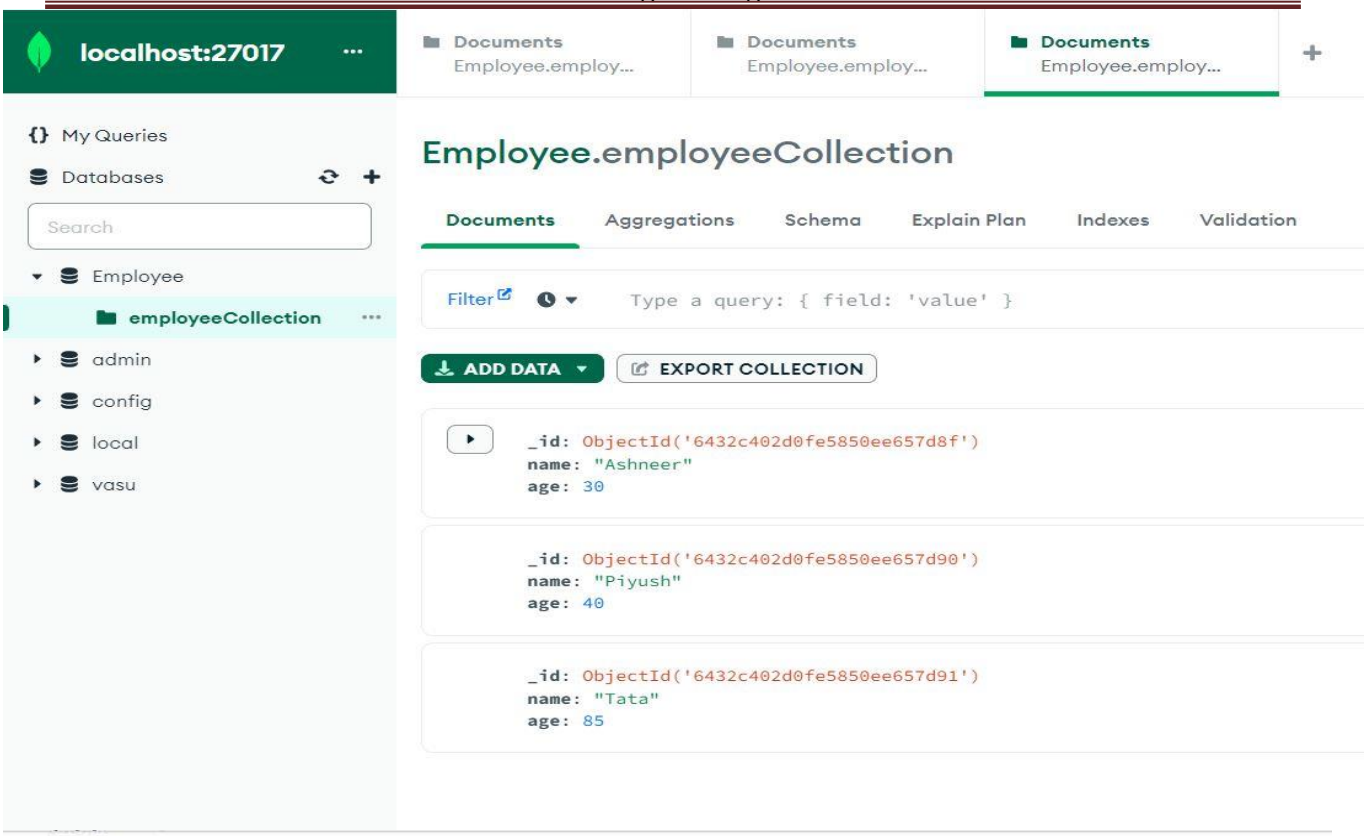
Output:-

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS G:\Sem-6\Lab\Solution\AWP\Nodejs-Program\Pra-5> node .\curd.js
Connected With Server
Created new database 'Employee'
Database Created
```


localhost:27017
Documents
Employee.employ...
My Queries
Databases
Search
Employee
employeeCollection
admin
config
local
vasu
Employee.employeeCollection
Documents
Aggregations
Schema
Explain Plan
Indexes
Validation
Filter
Type a query: { field: 'value' }
ADD DATA
EXPORT COLLECTION
_id: ObjectId('6432c402d0fe5850ee657d8e')
name: "Vasu"
age: 20
_id: ObjectId('6432c402d0fe5850ee657d8f')
name: "Ashneer"
age: 30
_id: ObjectId('6432c402d0fe5850ee657d90')
name: "Piyush"
age: 40
_id: ObjectId('6432c402d0fe5850ee657d91')
name: "Tata"
age: 85


localhost:27017
Documents
Employee.employ...
Documents
Employee.employ...
My Queries
Databases
Search
Employee
employeeCollection
admin
config
local
vasu
Employee.employeeCollection
Documents
Aggregations
Schema
Explain Plan
Indexes
Validation
Filter
Type a query: { field: 'value' }
ADD DATA
EXPORT COLLECTION
_id: ObjectId('6432c402d0fe5850ee657d8e')
name: "Vasu"
age: 16
_id: ObjectId('6432c402d0fe5850ee657d8f')
name: "Ashneer"
age: 30
_id: ObjectId('6432c402d0fe5850ee657d90')
name: "Piyush"
age: 40
_id: ObjectId('6432c402d0fe5850ee657d91')
name: "Tata"
age: 85



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS G:\Sem-6\Lab\Solution\AWP\Nodejs-Program\Pra-5> node .\curd.js
Connected With Server
Created new database 'Employee'
Database Created
PS G:\Sem-6\Lab\Solution\AWP\Nodejs-Program\Pra-5> node .\curd.js
Again Connected
4 documents were inserted.
Inserted a document with id 6432c402d0fe5850ee657d8e
Inserted a document with id 6432c402d0fe5850ee657d8f
Inserted a document with id 6432c402d0fe5850ee657d90
Inserted a document with id 6432c402d0fe5850ee657d91
Inserted
PS G:\Sem-6\Lab\Solution\AWP\Nodejs-Program\Pra-5> node .\curd.js
Again Again Connected
Updated
PS G:\Sem-6\Lab\Solution\AWP\Nodejs-Program\Pra-5> node .\curd.js
Again Again Again Connected
Deleted
PS G:\Sem-6\Lab\Solution\AWP\Nodejs-Program\Pra-5>
```

1. Use frames such that page is divided into 3 frames 20% on left to show contents of pages, 60% in center to show body of page, remaining on right to show remarks.

2. Create an HTML form to submit student data for Admission. Upload PDF copy of mark sheet, passport size photo and address proof (Use internal CSS).

3. Create your Resume using HTML tags also experiment with colors, text, link, size and also other tags you studied. (Use external CSS).

4. Write an HTML page that contains a selection box with a list of 5 countries, when the user selects a country, its capital should be printed next to the list; Add CSS to customize the properties of the font of the capital (color, bold and font size).

JAVA SCRIPT

1. Write a java script program which compute, the average marks of students then this average is used to determine the corresponding grade.

2. Write a Java Script code to display Fibonacci Series of given number. Number should be entered by user through text box.

3. Write a java script for loop that will iterate from 0 to 15 for each iteration, it will check if the current number is odd or even, and display a message to the screen.

4. To design the scientific calculator and make event for each button using java script.

5. Write JavaScript to validate the following fields of the above registration page. Name (Name should contains alphabets and the length should not be less than 6 characters). Password (Password should not be less than 6 characters length).E-mail id (should not contain any invalid and must follow the standard pattern name@domain.com)Phone number (Phone number should contain 10 digits only).

PHP & MYSQL

1. Write PHP programs

   - Write PHP program to change image automatically using switch case.

- Write PHP program to calculate current age without using any pre-define function.

- Write a PHP code to read data from JSON object and display in table form.

- Create simple login page in PHP. Let user login with predefined username and password. Logout user automatically after 5mins of user login and redirect back to login page.

- Write PHP code to perform CRUD operation using PHP.

Advanced Concepts:

1. Develop a web page which contains two list box. First list ask to select State and according to state selection second list box loads name of city. Develop it using AJAX.

2. Create a webpage that hides and show the image when clicked using jQuery.