

LATE BHAUSAHEB HIRAY S.S. TRUST’S INSTITUTE OF COMPUTER APPLICATION

**ISO 9001-2008 CERTIFIED**

**S.N. 341, Next to New English School, Govt. Colony, Bandra (East), Mumbai – 400051, Tel: 91-22-26570892/3181**

### Date:11/04/2022

**CERTIFICATE**

### This is to certify that Mr. Saurabh Bhausaheb Gawali

**Roll No. 2021105 is a student of FYMCA Semester-I has completed successfully full-semester practical/assignments of subject Web Technology for the academic year 2021– 22.**

### Subject In-Charge Director

### 

### External Examiner

**LBHSS’s**

**Hiray Institute of Computer Application**

**WEB TECHNOLOGY PRACTICAL – JOURNAL**

**F.Y.MCA**

|  |  |
| --- | --- |
|  | **Index** |
| 1 | Create an application to demonstrate Node.js Modules |
| 2 | Create an application to demonstrate various Node.js Events |
| 3 | Create an application to demonstrate Node.js Functions |
| 4 | Using File Handling demonstrate all basic file operations (Create, write, read, delete) |
| 5 | Create an HTTP Server and perform operations on it |
| 6 | Create an application to establish a connection with the MySQL database and perform basic database operations on it |
| 7 | Create an application using Filters |
| 8 | Create an application to demonstrate directives |
| 9 | Demonstrate controllers in Angular.js through an application |
| 10 | Demonstrate features of Angular.js forms with a program |
| 11 | Create a SPA (Single Page Application) |

Practical No : 1

1. **Create an application to demonstrate Node.js Modules.**

**There are two types of modules in Nodejs**

1. **Build-in Module**
2. **Local Module**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

1. **Build-in Module:**
2. **http module**

**Coding:-**

 const http= require('http');

 const server= http.createServer((req,res) =>{

     if(req.url==='/')

     {

         res.write('Hello Everyone');

         res.end();

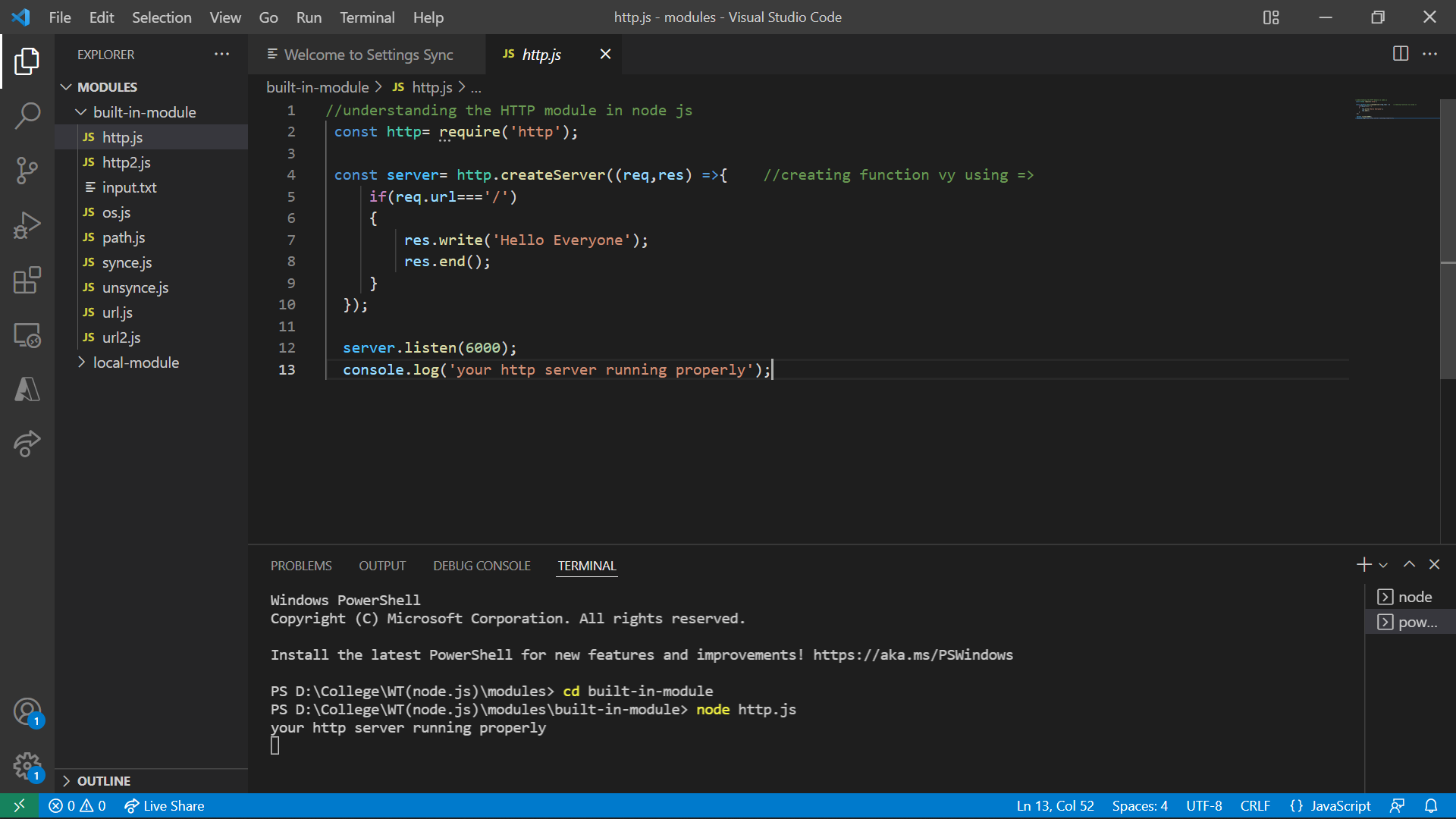
     }

  });

  server.listen(6000);

  console.log('your http server running properly');

**Output:-**



**2) OS Module**

**Coding:-**

var os=require('os');

console.log(os.EOL);

console.log(os.arch());

console.log(os.hostname());

console.log(os.freemem());

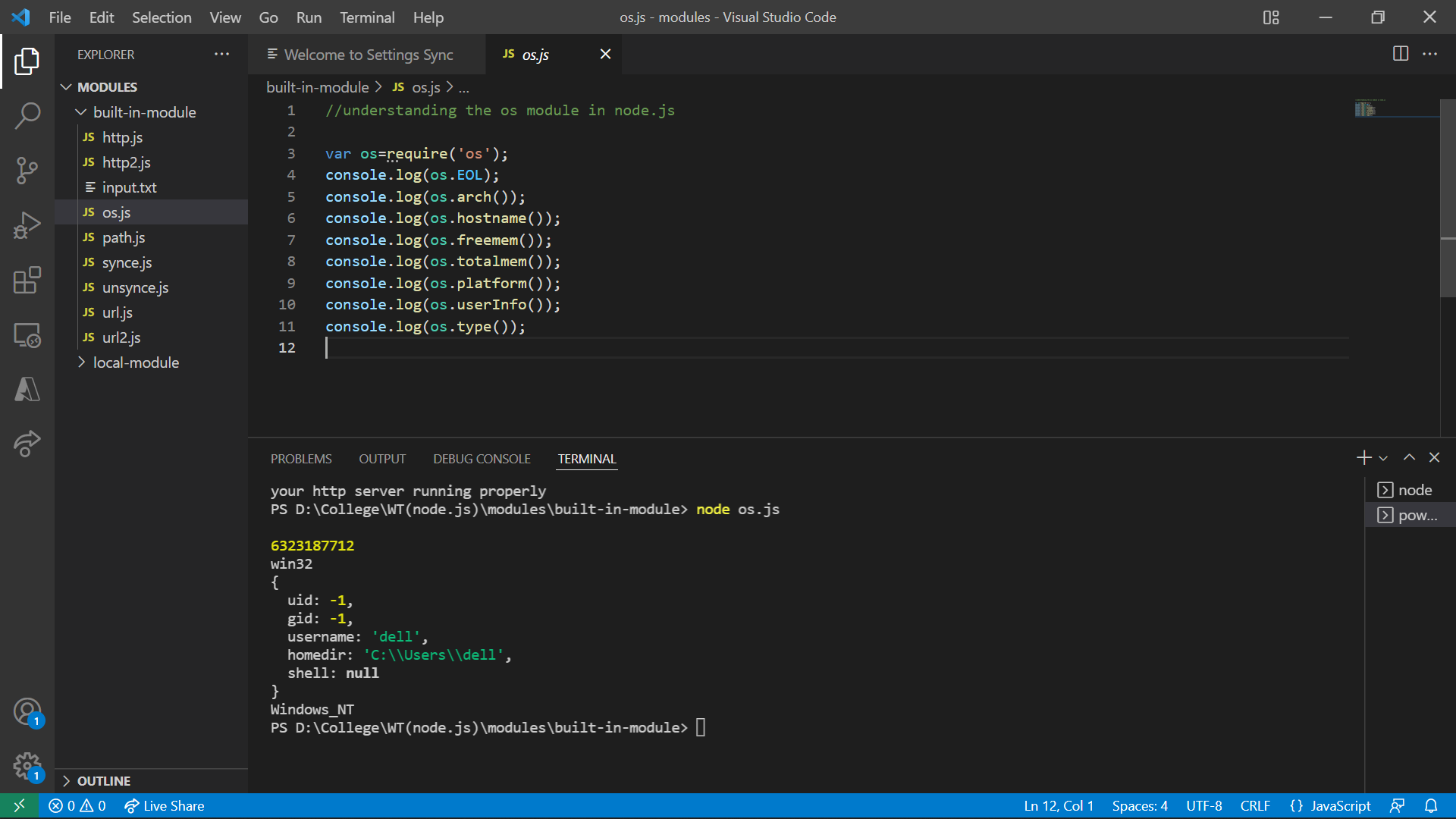
console.log(os.totalmem());

console.log(os.platform());

console.log(os.userInfo());

console.log(os.type());

**Output:-**



**3) Path Module**

**Coding:-**

var path= require('path');

console.log(path.dirname('C:/Users/HP/Desktop/WT/WT practical/Module/Build-in-module/Path.js'));

console.log(path.basename('C:/Users/HP/Desktop/WT/WT practical/Module/Build-in-module/Path.js'));

console.log(path.extname('C:/Users/HP/Desktop/WT/WT practical/Module/Build-in-module/Path.js'));

console.log(path.parse('C:/Users/HP/Desktop/WT/WT practical/Module/Build-in-module/Path.js'));

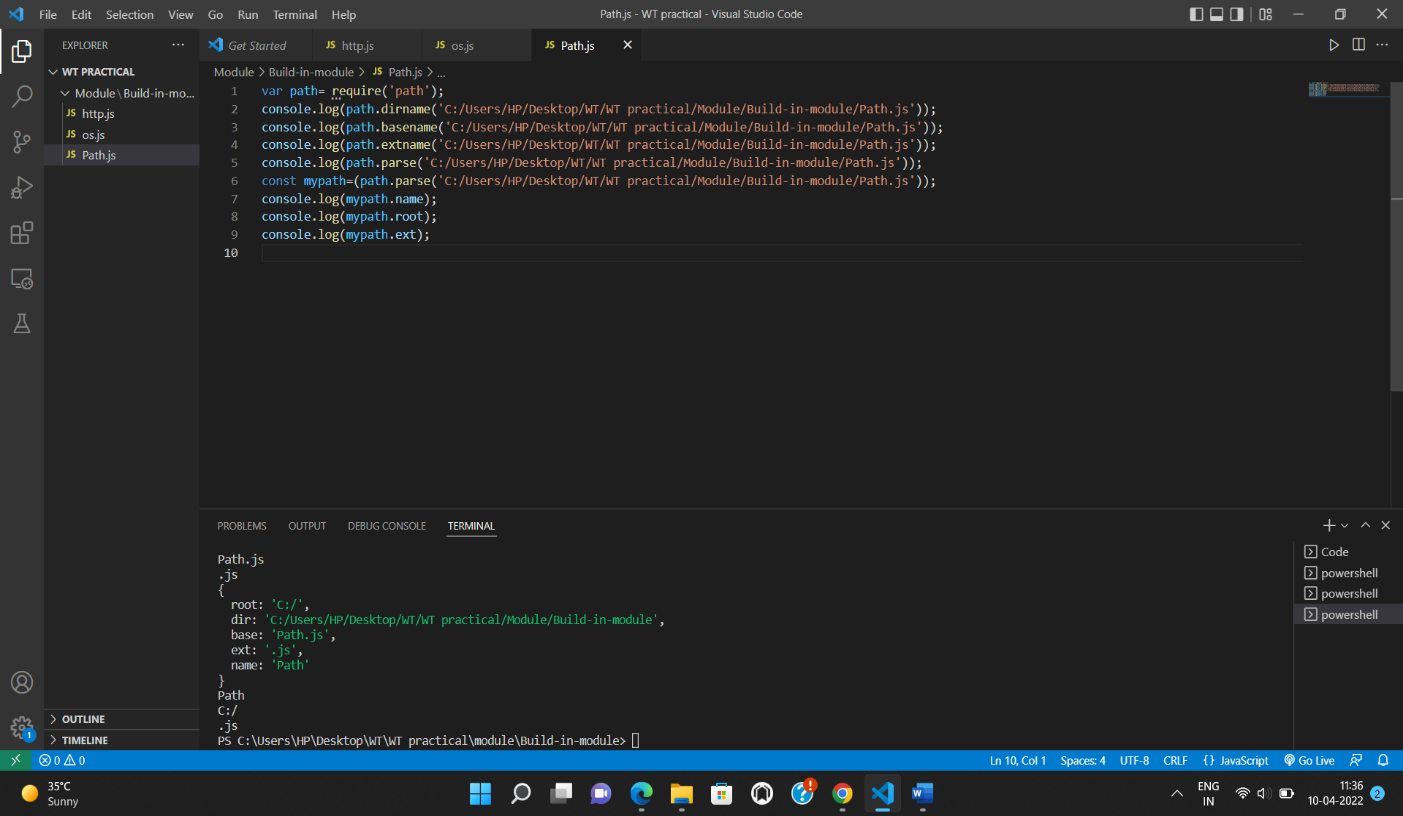
const mypath=(path.parse('C:/Users/HP/Desktop/WT/WT practical/Module/Build-in-module/Path.js'));

console.log(mypath.name);

console.log(mypath.root);

console.log(mypath.ext);

**Output:-**



**4) Synch Module**

**Coding:-**

var fs = require('fs');

console.log('here we started');

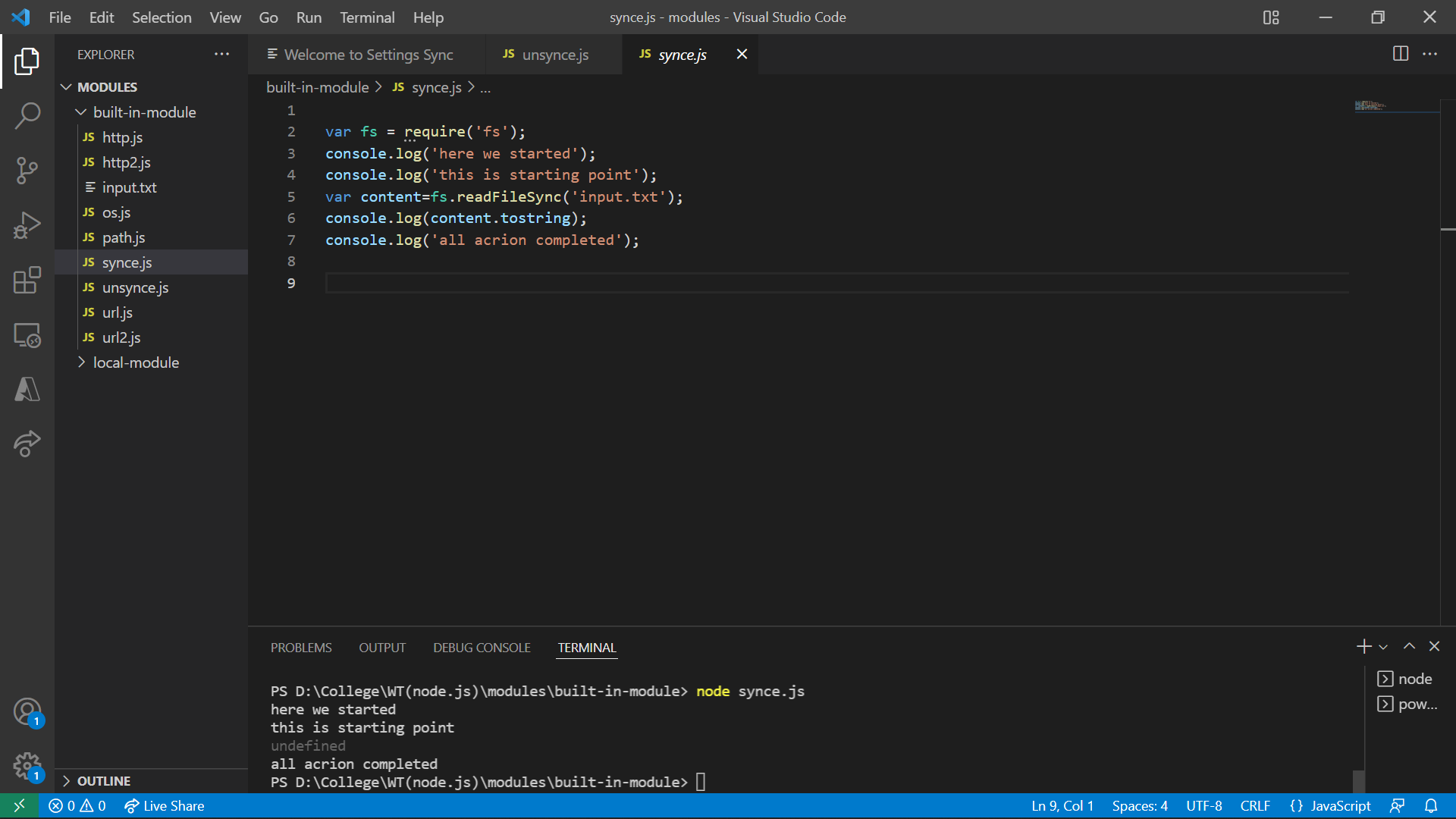
console.log('this is starting point');

var content=fs.readFileSync('input.txt');

console.log(content.tostring);

console.log('all acrion completed');

**Output:-**



**5)Unsynce Module**

**Coding:-**

var fs = require ('fs');

console.log('start');

fs.readFile('input.txt',function(error,data){

    if(error){

        console.error(error);

        return;

    }

console.log(data.toString());

});

console.log('endnnnnnnnnnnnnnnnn');

console.log('Hello Good Morning!!!!!!!!!!!!!!!!!!!!!!!');

**Output:-**



**6)URL Module**

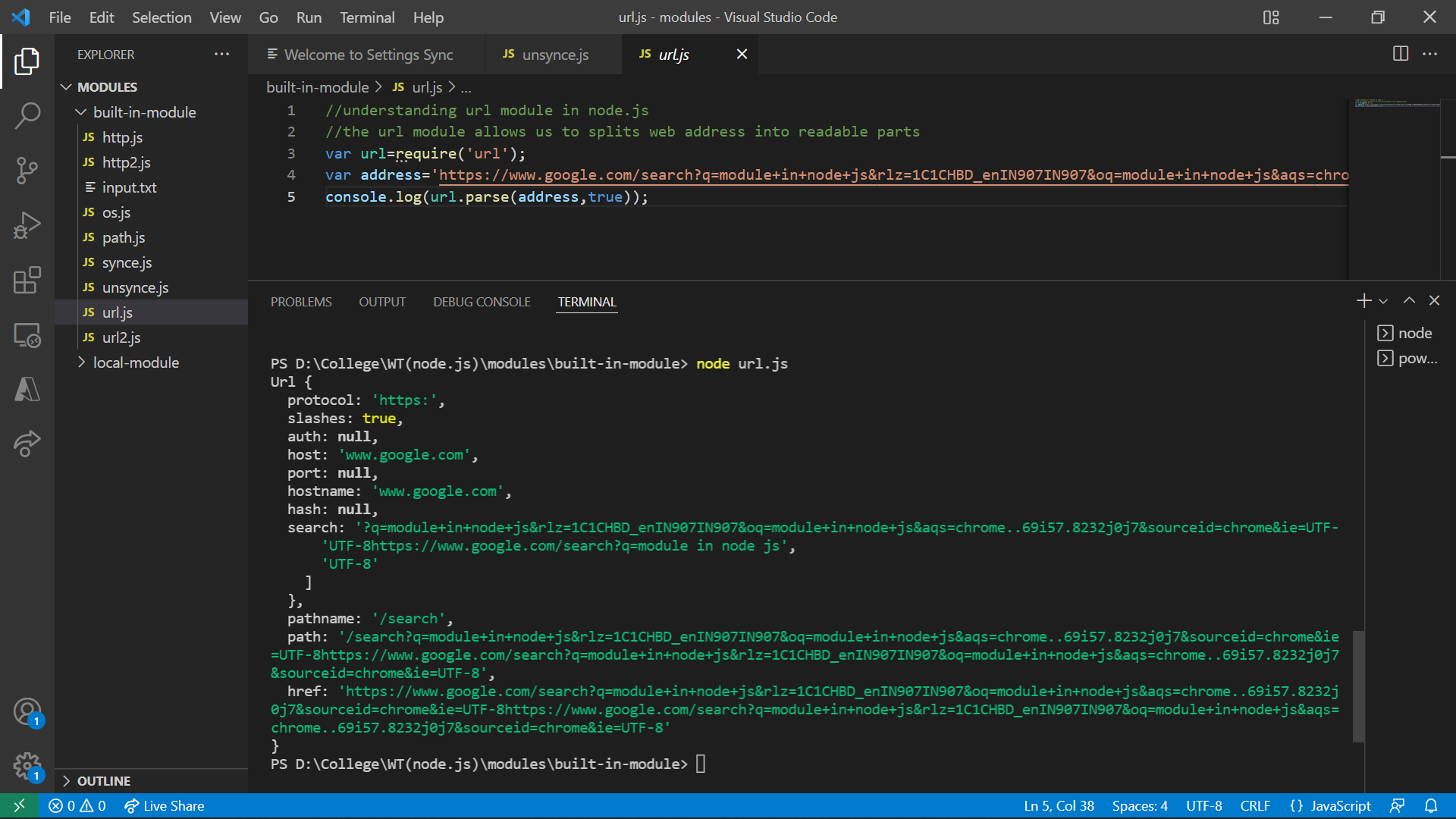
**Coding:-**

var url=require('url');

var address='https://www.google.com/search?q=module+in+node+js&rlz=1C1CHBD\_enIN907IN907&oq=module+in+node+js&aqs=chrome..69i57.8232j0j7&sourceid=chrome&ie=UTF-8https://www.google.com/search?q=module+in+node+js&rlz=1C1CHBD\_enIN907IN907&oq=module+in+node+js&aqs=chrome..69i57.8232j0j7&sourceid=chrome&ie=UTF-8';

console.log(url.parse(address,true));

**Output:-**



**b)Local Module**

Local Modules are not inbuilt module. We can say that Local Module created by users.

**Coding:-**

**local\_module1.js**

exports.add=function(x,y){

    return x+y;

};

exports.sub=function(x,y){

    return x-y;

};

exports.div=function(x,y){

    return x/y;

};

exports.mul=function(x,y){

    return x\*y;

};

**local\_module2.js**

var cal=require('./local\_module1.js');

 var x=50;

 var y=5;

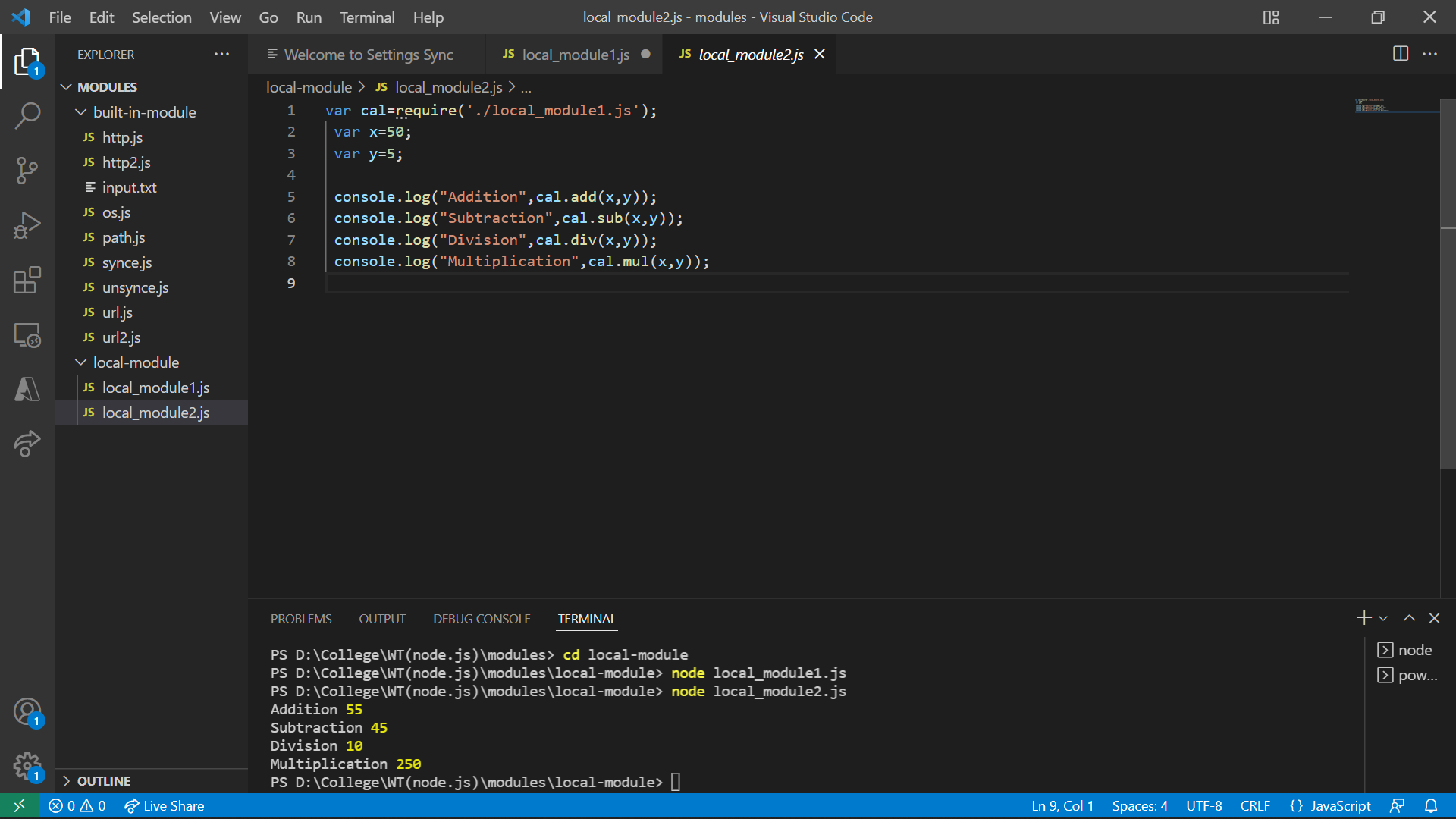
 console.log("Addition",cal.add(x,y));

 console.log("Subtraction",cal.sub(x,y));

 console.log("Division",cal.div(x,y));

 console.log("Multiplication",cal.mul(x,y));

**Output:-**



Practical No: 2

1. **Create an application to demonstrate various Node.js Events**

**AIM:-** The main aim of events is call back function when event is triggered.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Coding:-**

var events = require('events');

var eventEmitter = new events.EventEmitter();

eventEmitter.addListener('msg', function() {

  console.log('Helloooooo');

});

eventEmitter.on('msg1',function() {

  console.log('Good Evening');

});

eventEmitter.removeListener('msg2',function(){

  console.log('MCA');

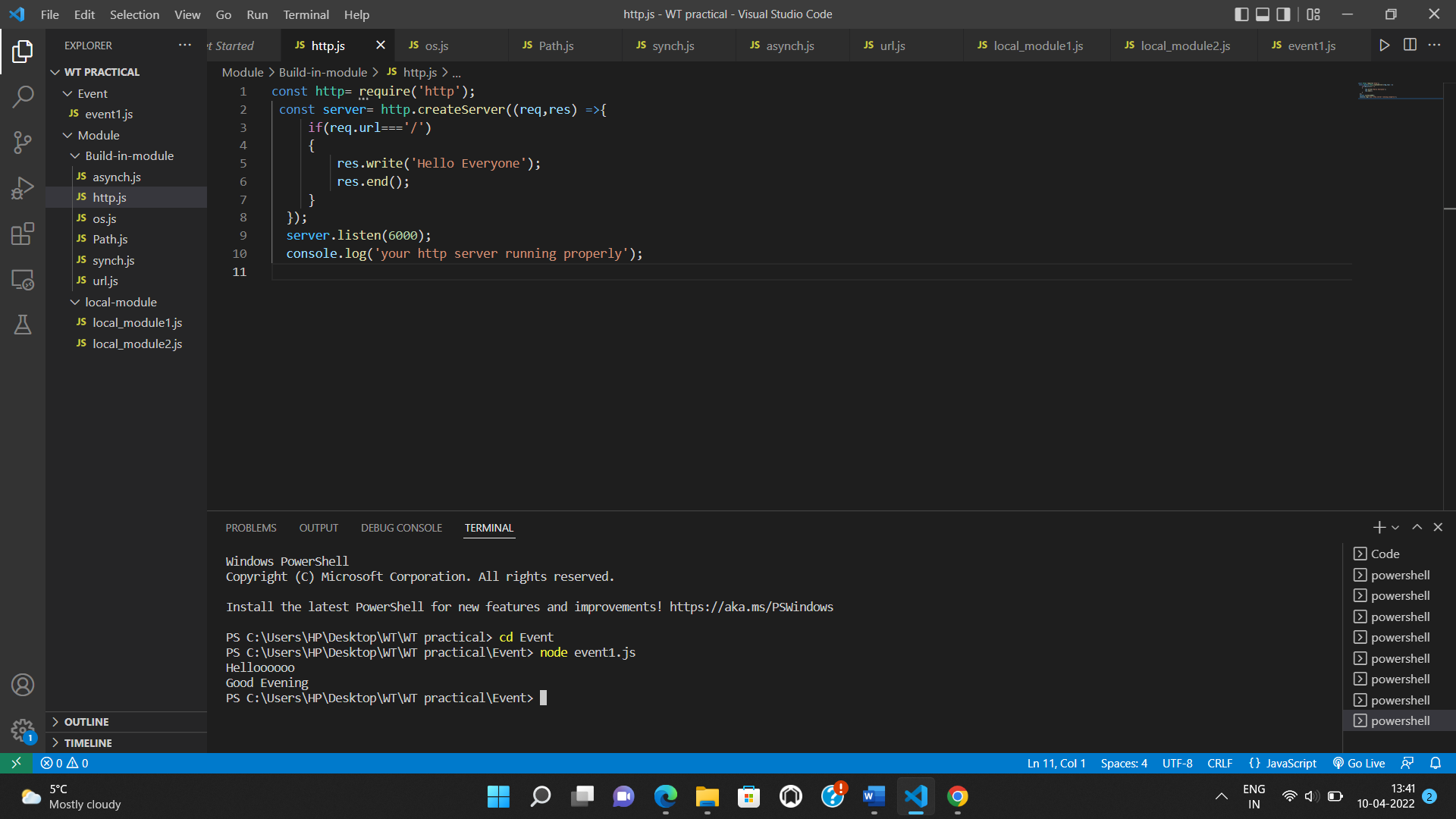
});

eventEmitter.emit('msg');

eventEmitter.emit('msg1');

eventEmitter.emit('msg2');

**Output:-**



Practical No: 3

1. **Create an application to demonstrate Node.js Functions**
2. **e.js Functions**

**AIM:-** Functions are used to get desire action in application.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Let’s create function with for calculating values

**Coding:-**

function cal(n1,n2)

{

    console.log(n1+n2);

    console.log(n1-n2);

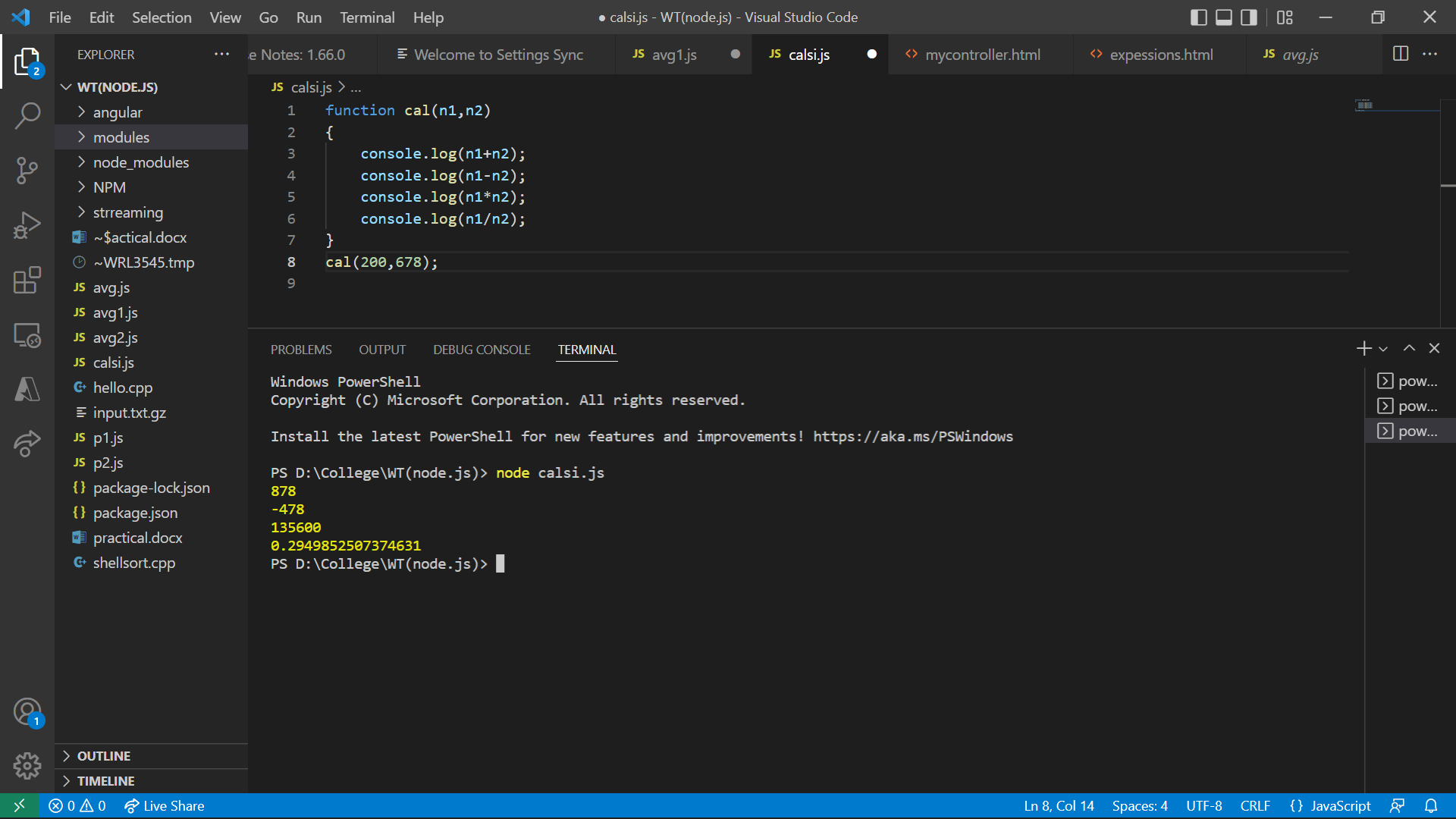
    console.log(n1\*n2);

    console.log(n1/n2);

}

cal(200,678);

**Output:-**



Practical No: 4

1. **Using File Handling demonstrate all basic file operations (Create, write, read, delete)**

**Create file**

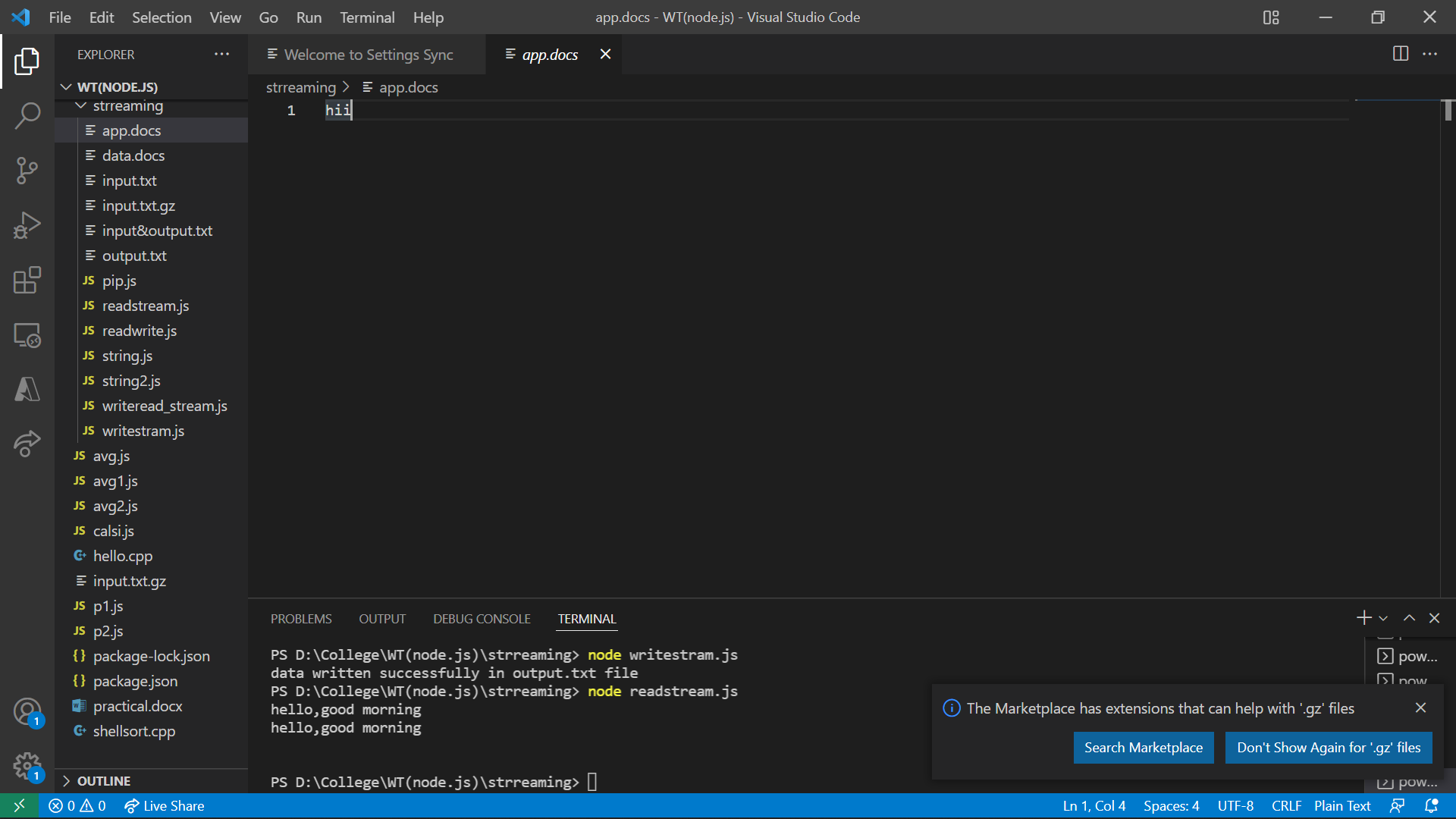
**AIM:-** File handling used to store the data in a file using peogram.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

In this first we read the file then we are going to create file.

**Code:-**

**data.docs**



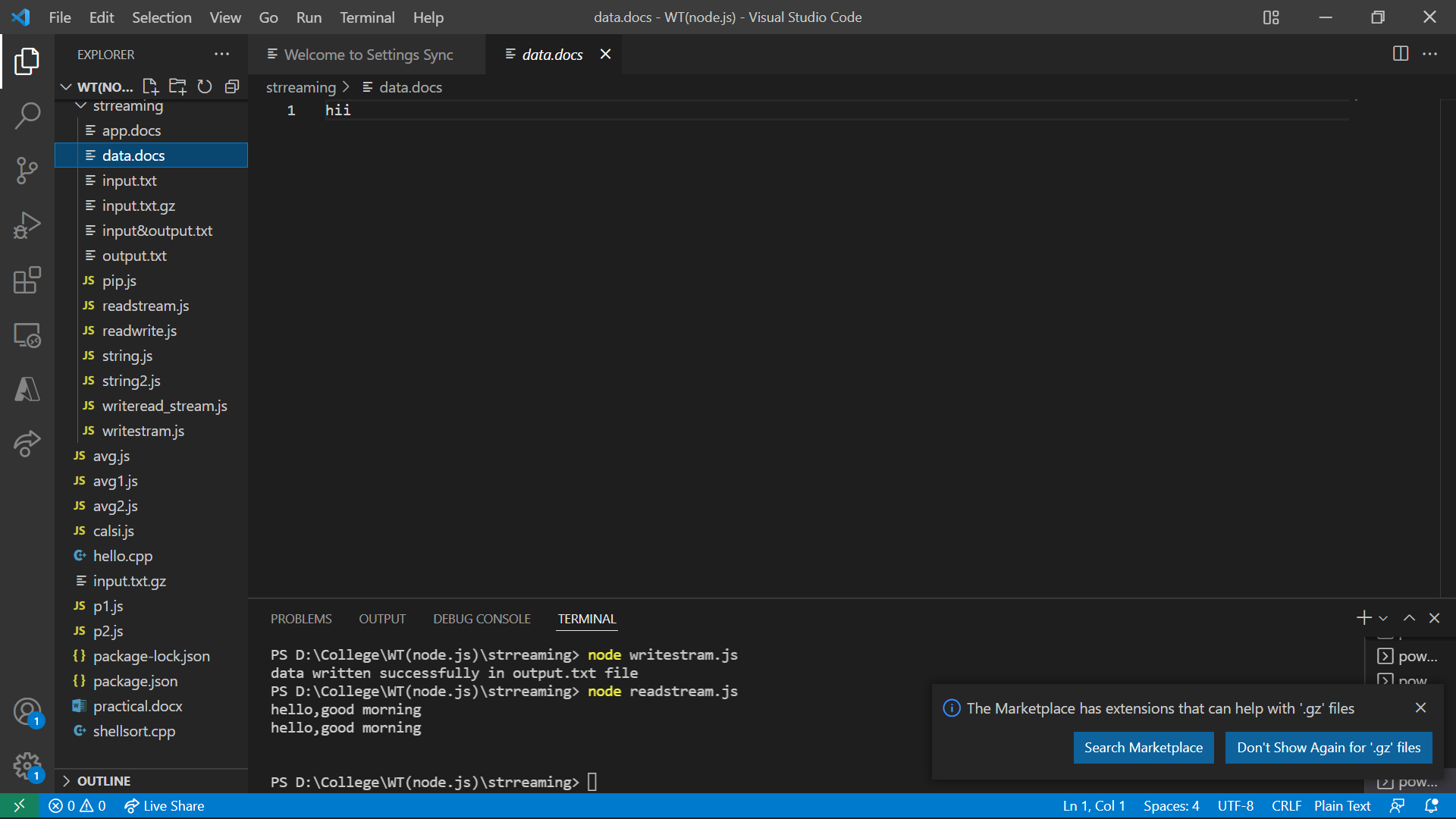
var fs = require('fs');

var readstream= fs.createReadStream('data.docs');

var writestream=fs.createWriteStream('app.docs');

readstream.pipe(writestream);

**app.docs**



Write

In this we are going to write some text in file

**Code:**

var fs =require('fs');

var data= 'hiii';

var writestream=fs.createWriteStream('output.txt');

writestream.write(data, 'utf-8');

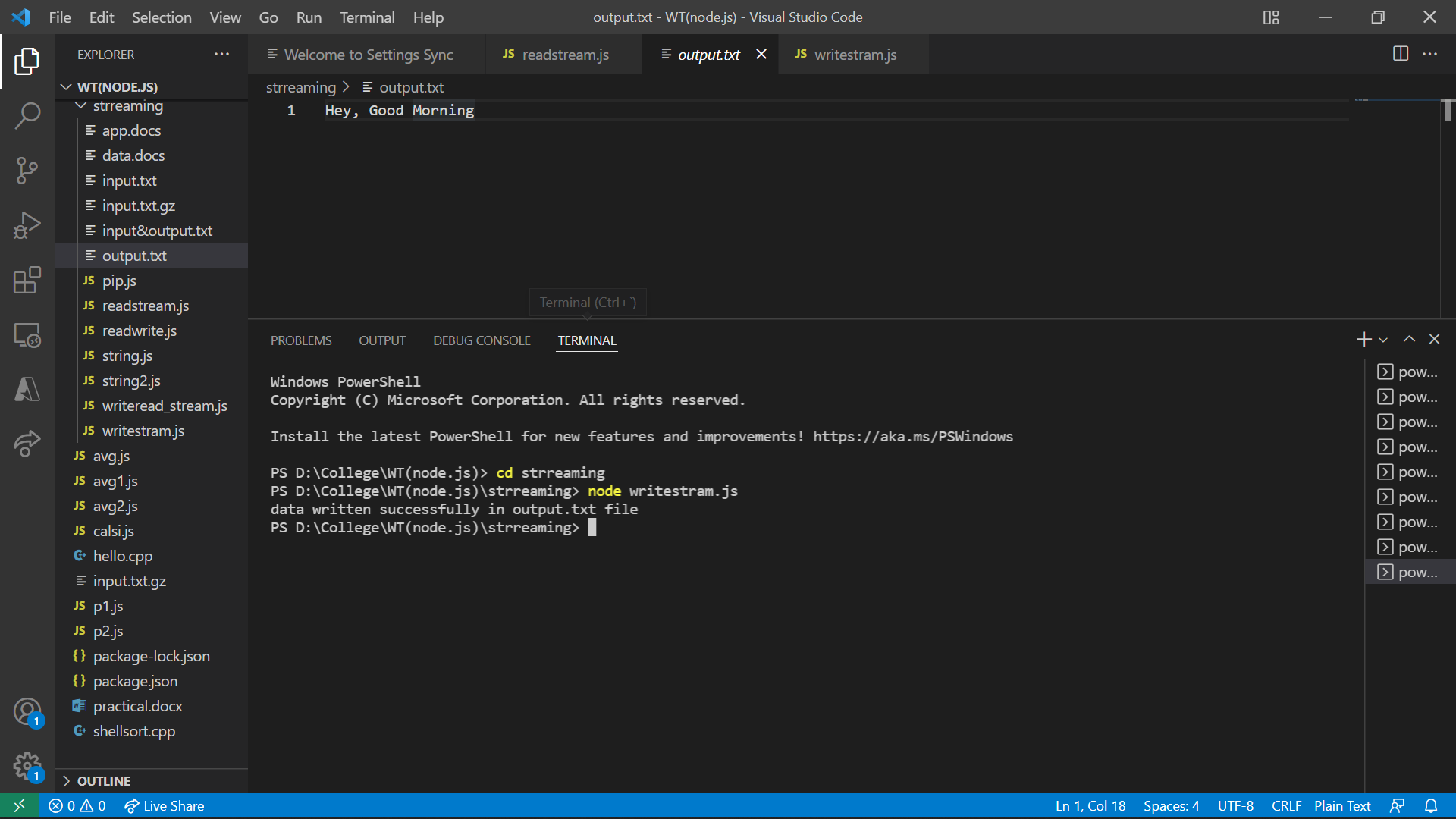
writestream.end();

writestream.on('finish',function(){

    console.log("data written successfully in output.txt file");

});

**Output:**



Read

Previously we write the file , now we are going to read that same file.

**Coding:-**

var fs =require('fs');

var data= '';

var readstream=fs.createReadStream('output.txt');

readstream.setEncoding('utf-8');

readstream.on('data',function(chunk){

    data=data+chunk;

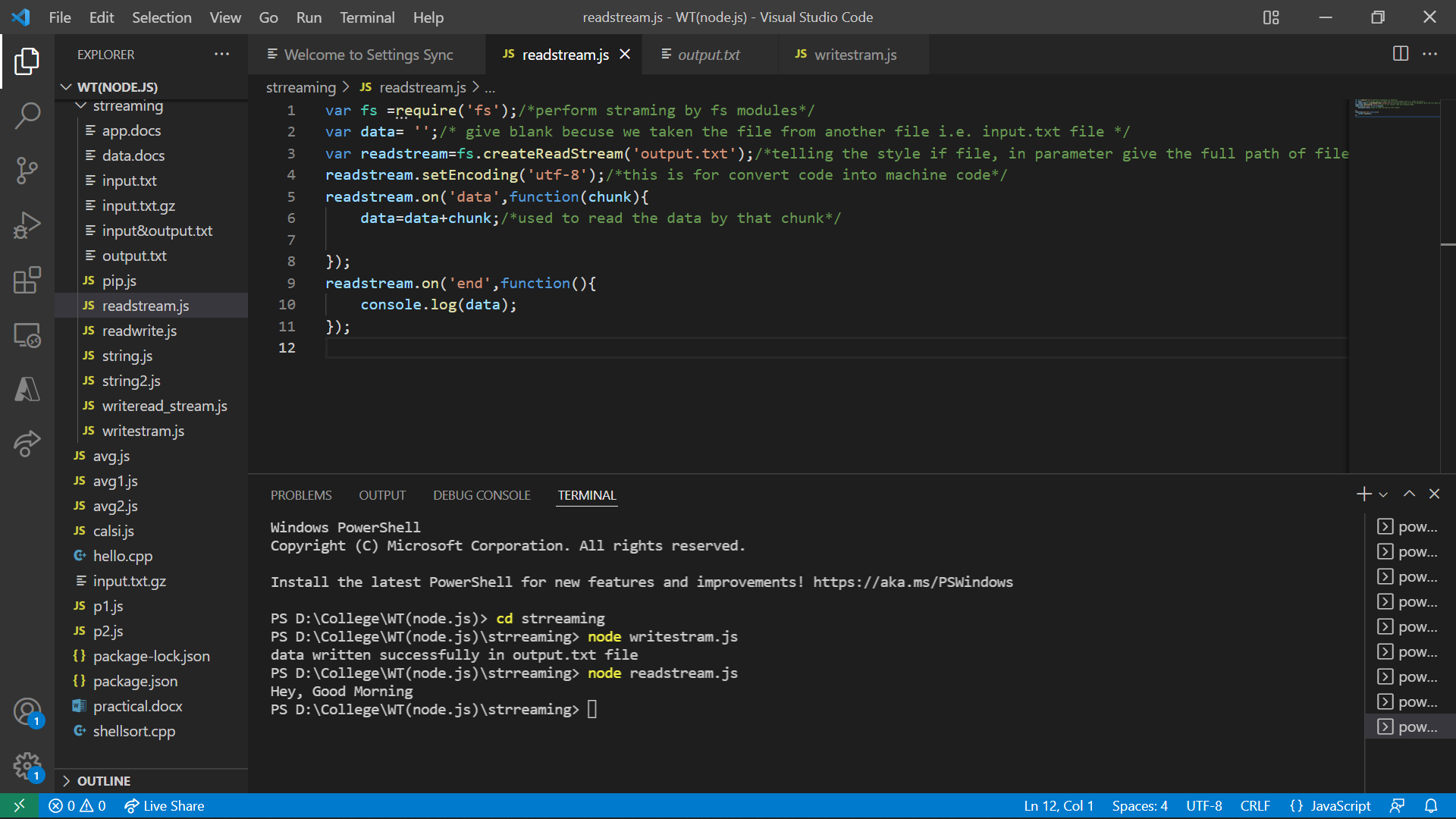
});

readstream.on('end',function(){

    console.log(data);

});

**Output:-**



Delete

Again now working with previous file , this time we are going to delete data which we are previously write and read in file.

**Coding:-**

var fs =require('fs');

try{

    fs.unlinkSync('output.txt');

    console.log("Data is deleted Successfully...");

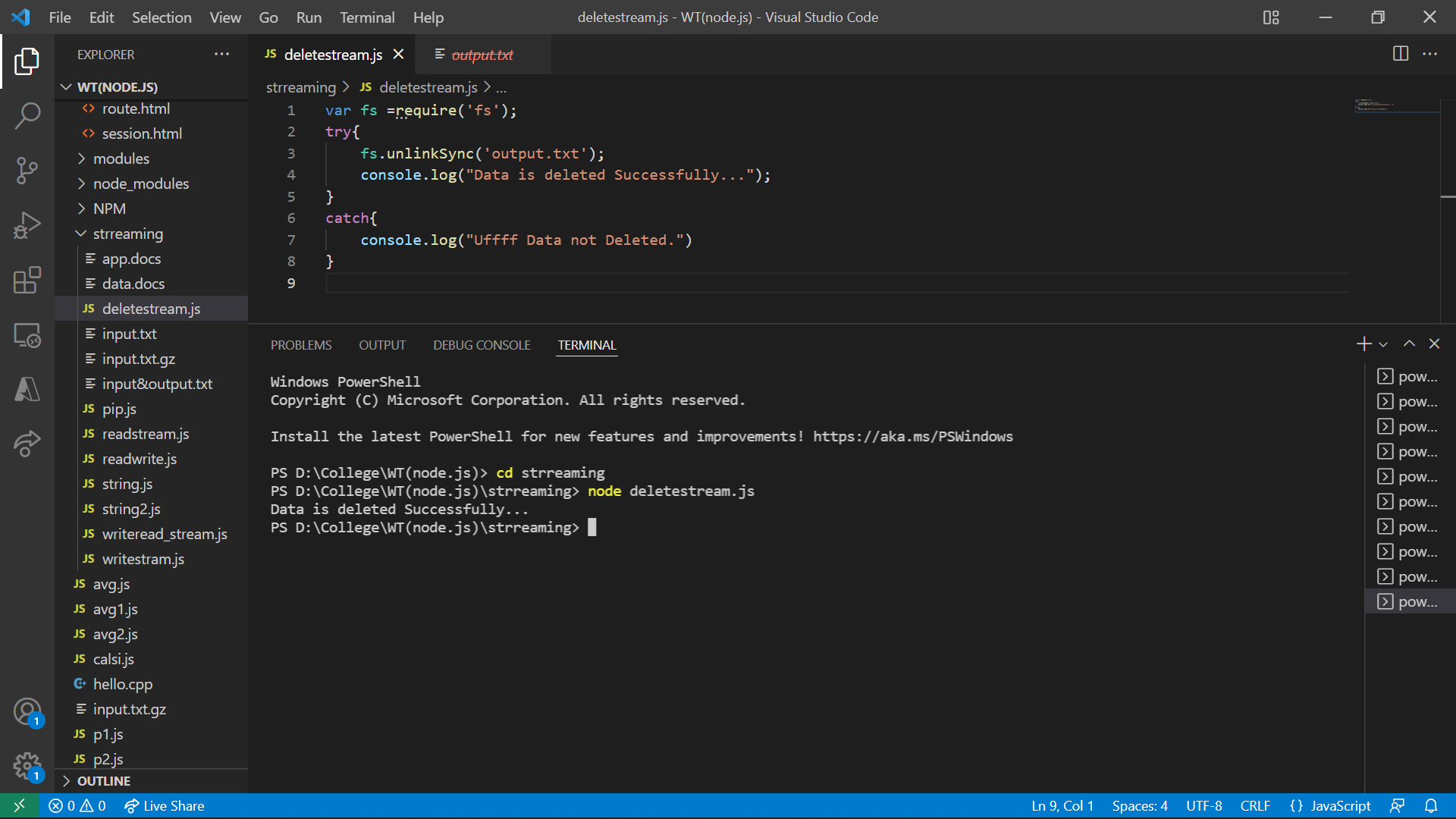
}

catch{

    console.log("Uffff Data not Deleted.")

}

**Output:-**



Practical No: 5

1. **Create an HTTP Server and perform operations on it**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Coding:-**

const http= require('http');

 const server= http.createServer((req,res) =>{

     if(req.url==='/')

     {

         res.write('Hello Everyone');

         res.end();

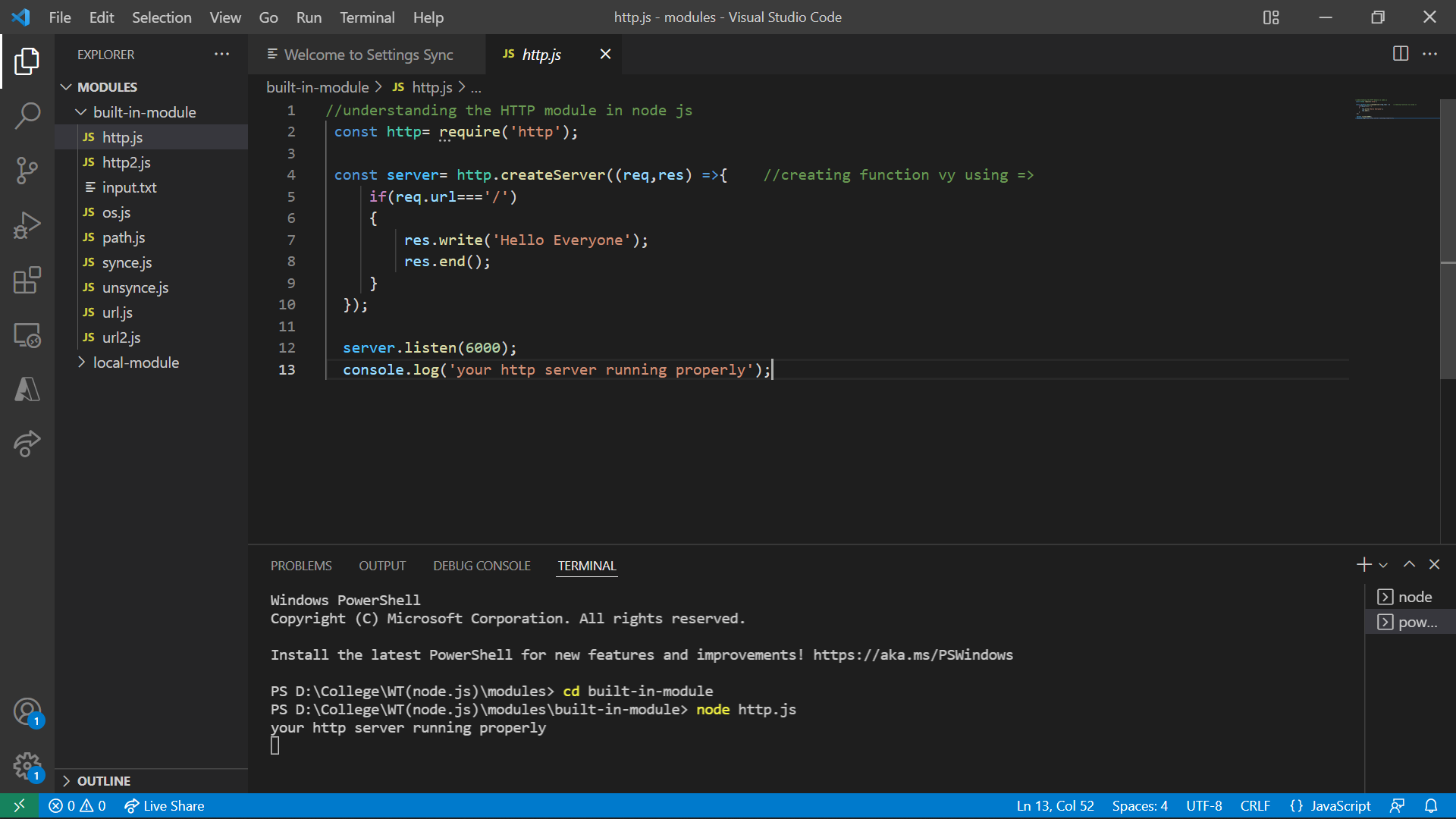
     }

  });

  server.listen(6000);

  console.log('your http server running properly');

**Output:-**



Practical NO: 6

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

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**AIM:-** Database is used to save the information.

**Coding:-**

Creating Database

var mysql=require('mysql');

var conn=mysql.createConnection({

    host:'localhost',

    user:'root',

    password:'',

    database:'collegedb'

});

conn.query("create database collegebd",

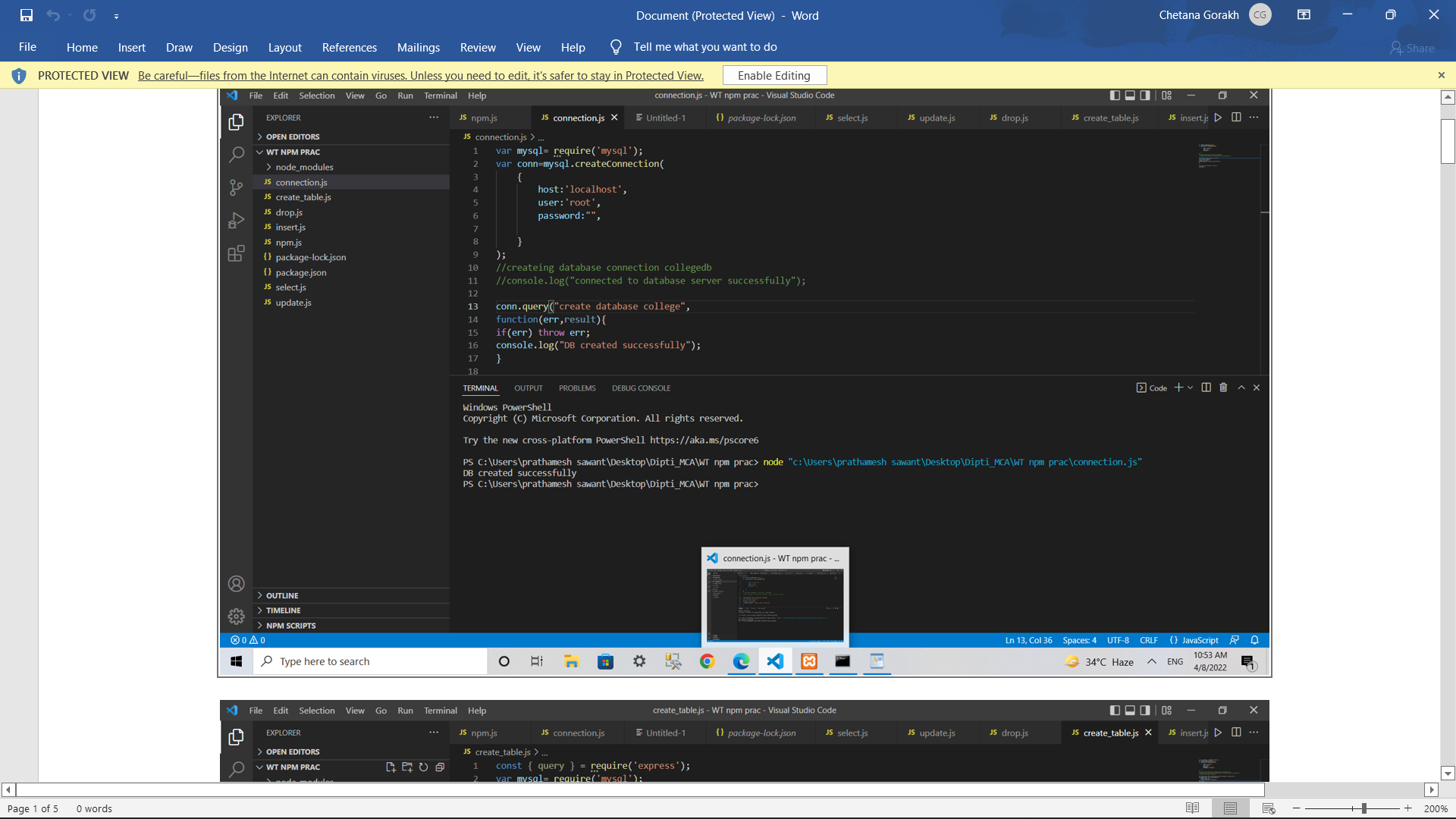
function(err,result){

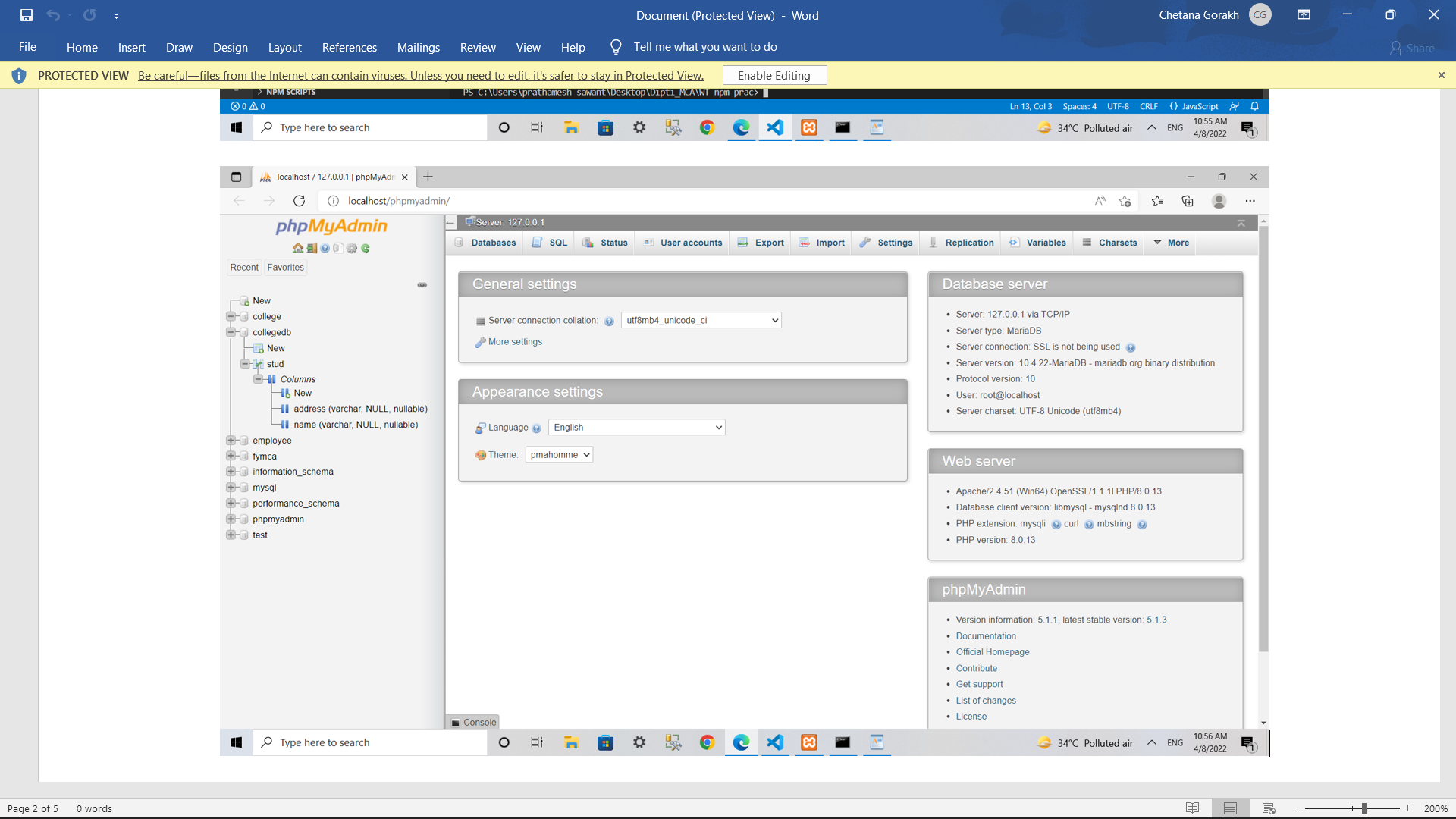
    if (err) throw err;

    console.log("Database cretaed successfully")

});

**Output:-**





Create table

var sql="create table stud (name varchar(10),address varchar(30))";

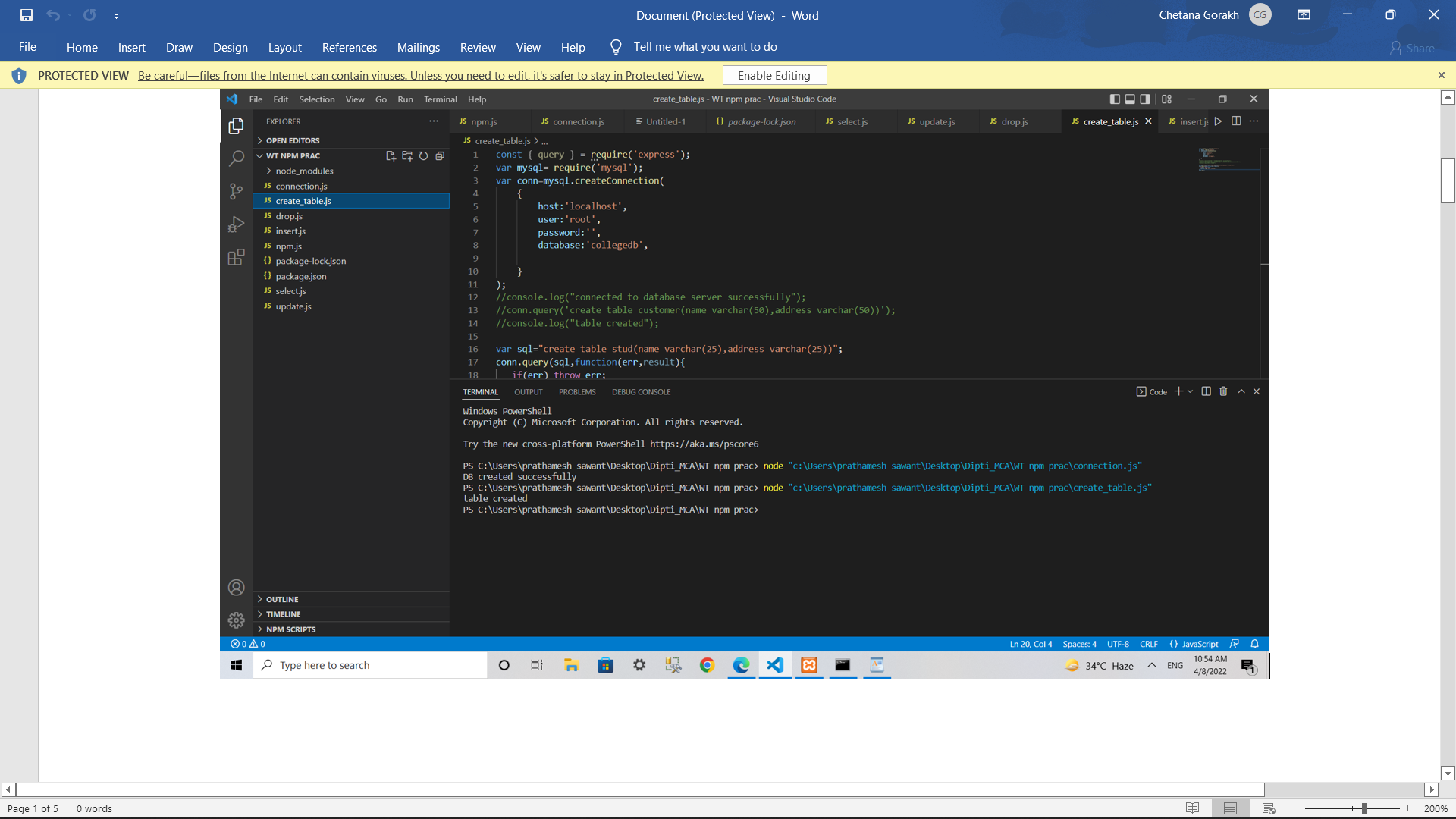
conn.query(sql,function(err,result){

    if(err) throw err;

    console.log("table created");

});

**Output:-**



Inserted data

var record=[

            ['dipti’,’USA’],

            ['trupti','mumbai']

            ];

var sql="insert into stud(name,address)values?";

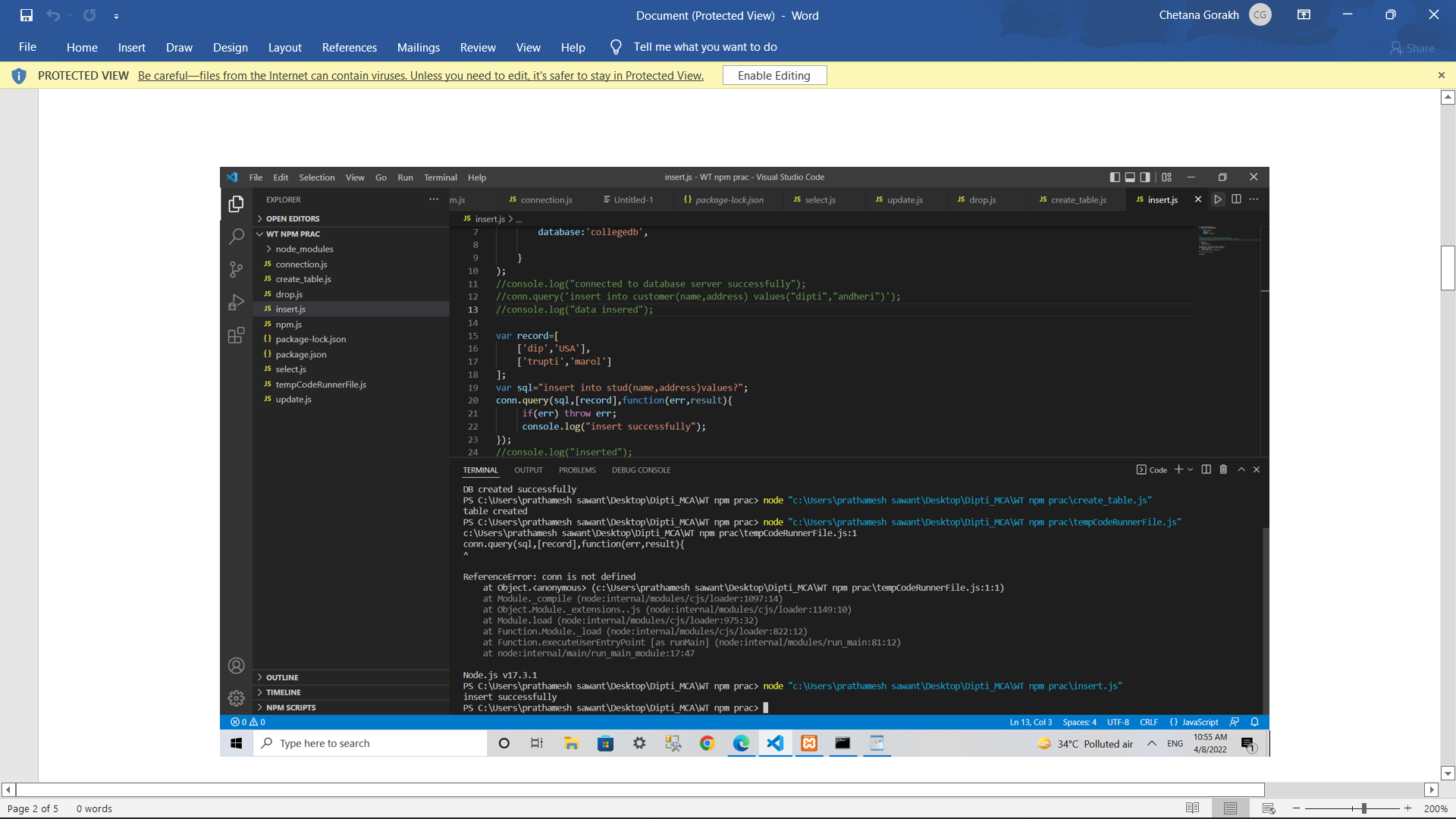
conn.query(sql,[record],function(err,result){

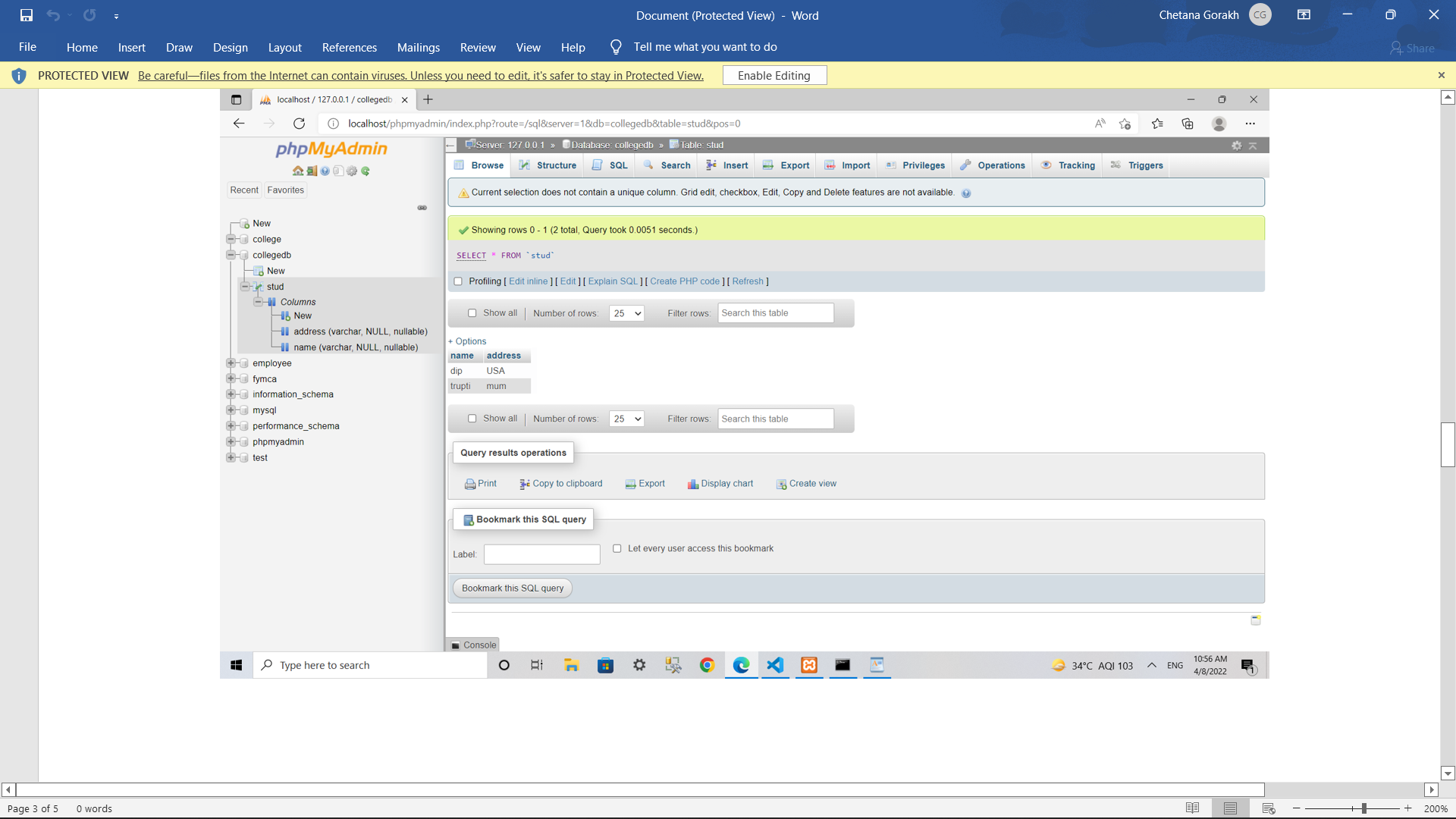
    if(err)throw err;

    console.log("inserted successfully");

});

**Output:-**





Drop db

var mysql=require('mysql');

var conn=mysql.createConnection({

    host:'localhost',

    user:'root',

    password:''

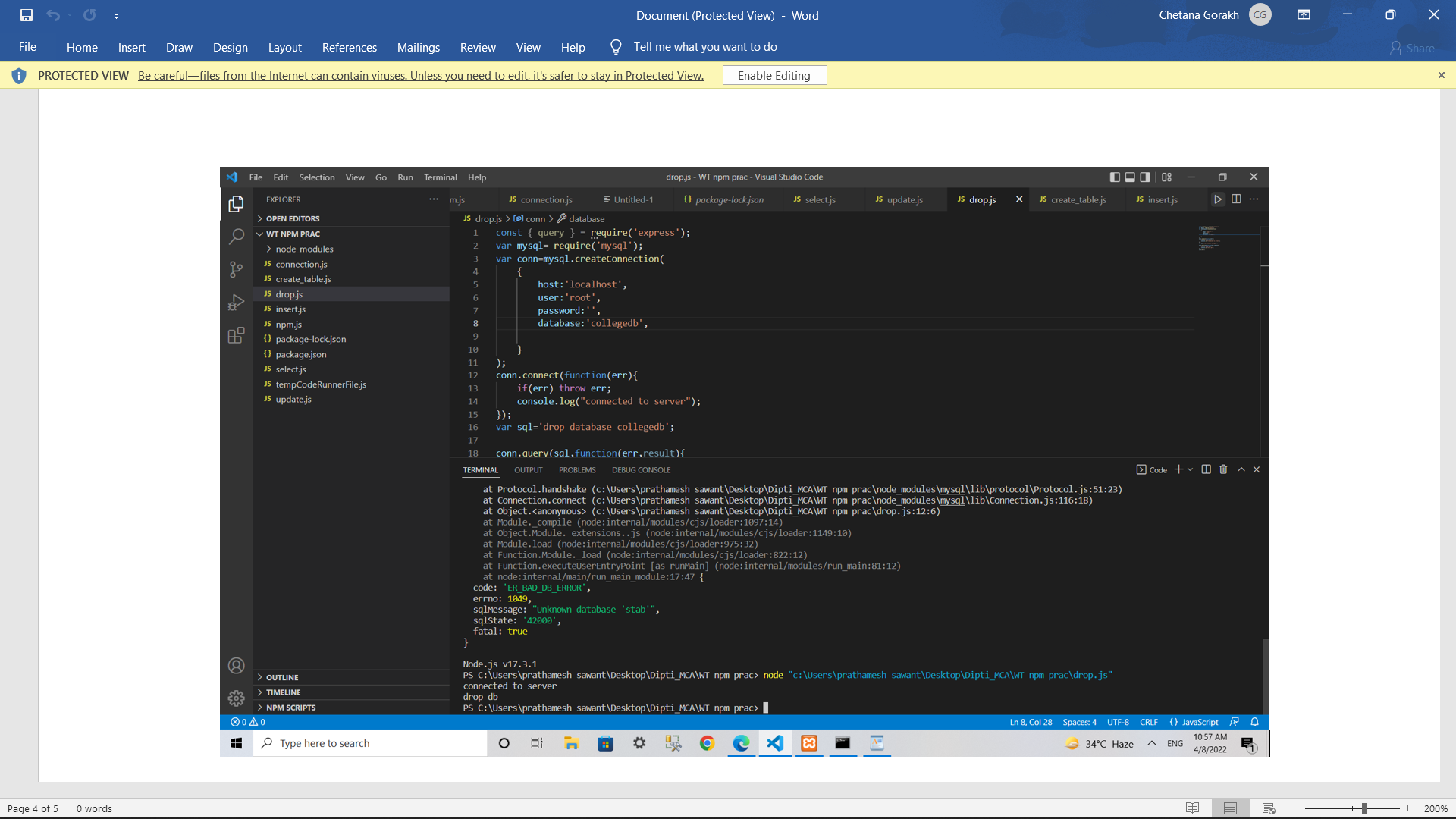
});

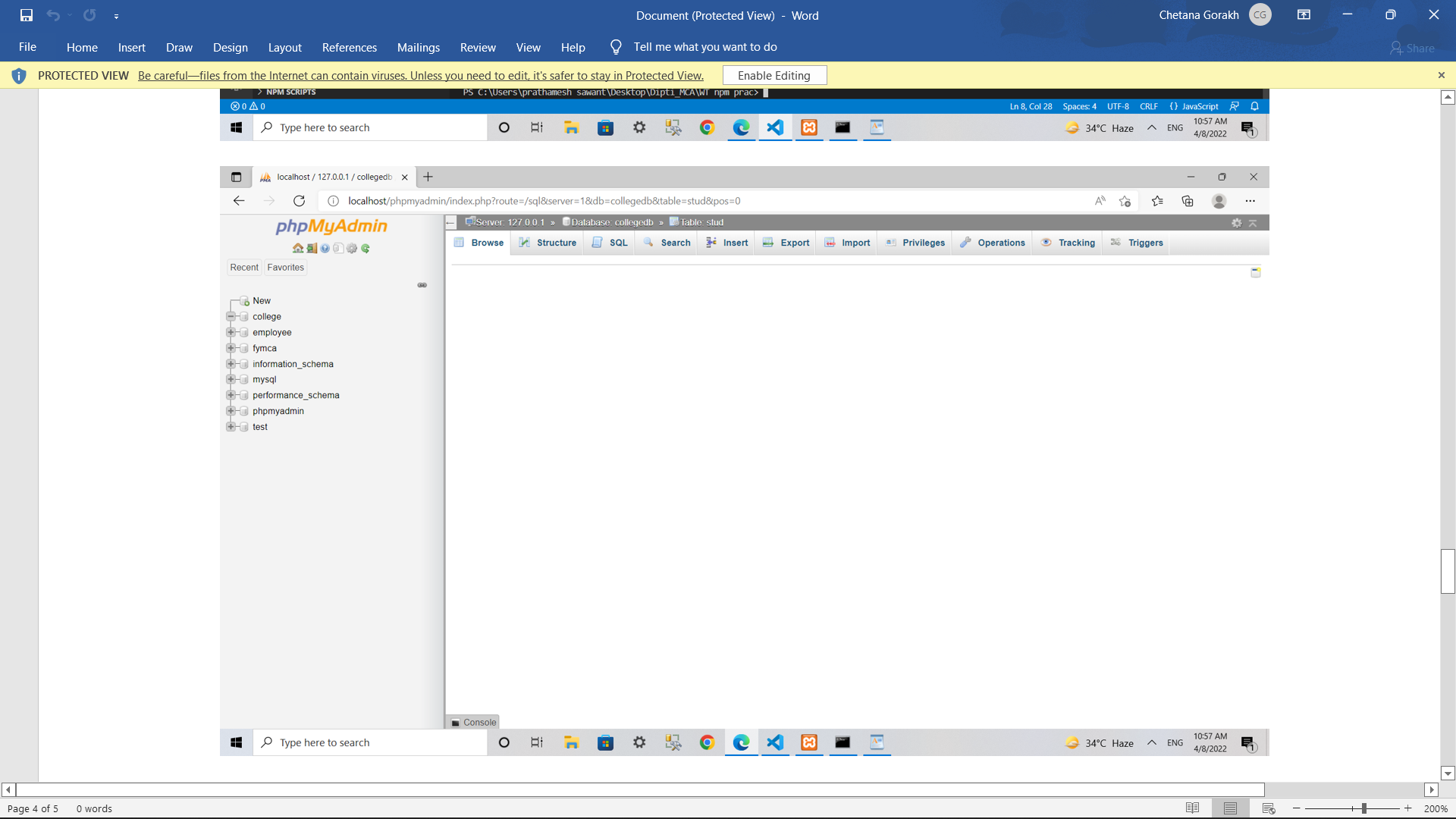
conn.query('drop database collegebd');

conn.end();

console.log('database deleted successfully')

**Output:-**





Practical No: 7

1. **Create an application using Filters**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**AIM:-** Filters are used in building the web-applications

**Coding:-**

<html>

   <head>

      <title>Angular JS Filters</title>

      <script src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js">

      </script>

   </head>

   <body>

      <div ng-app = "mainApp" ng-controller = "empController">

           <br> Name in Upper Case:{{employee.firstName | uppercase}}

           <br> Name in Lower Case:{{employee.lastName | lowercase}}

            <br>salary:{{employee.salary | currency}}

      </div>

      <script>

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

         mainApp.controller('empController', function($scope) {

            $scope.employee = {

               firstName: "Shradhha Kapoor",

               lastName: "Shradhha Kapoor",

               salary:"50000",

               }

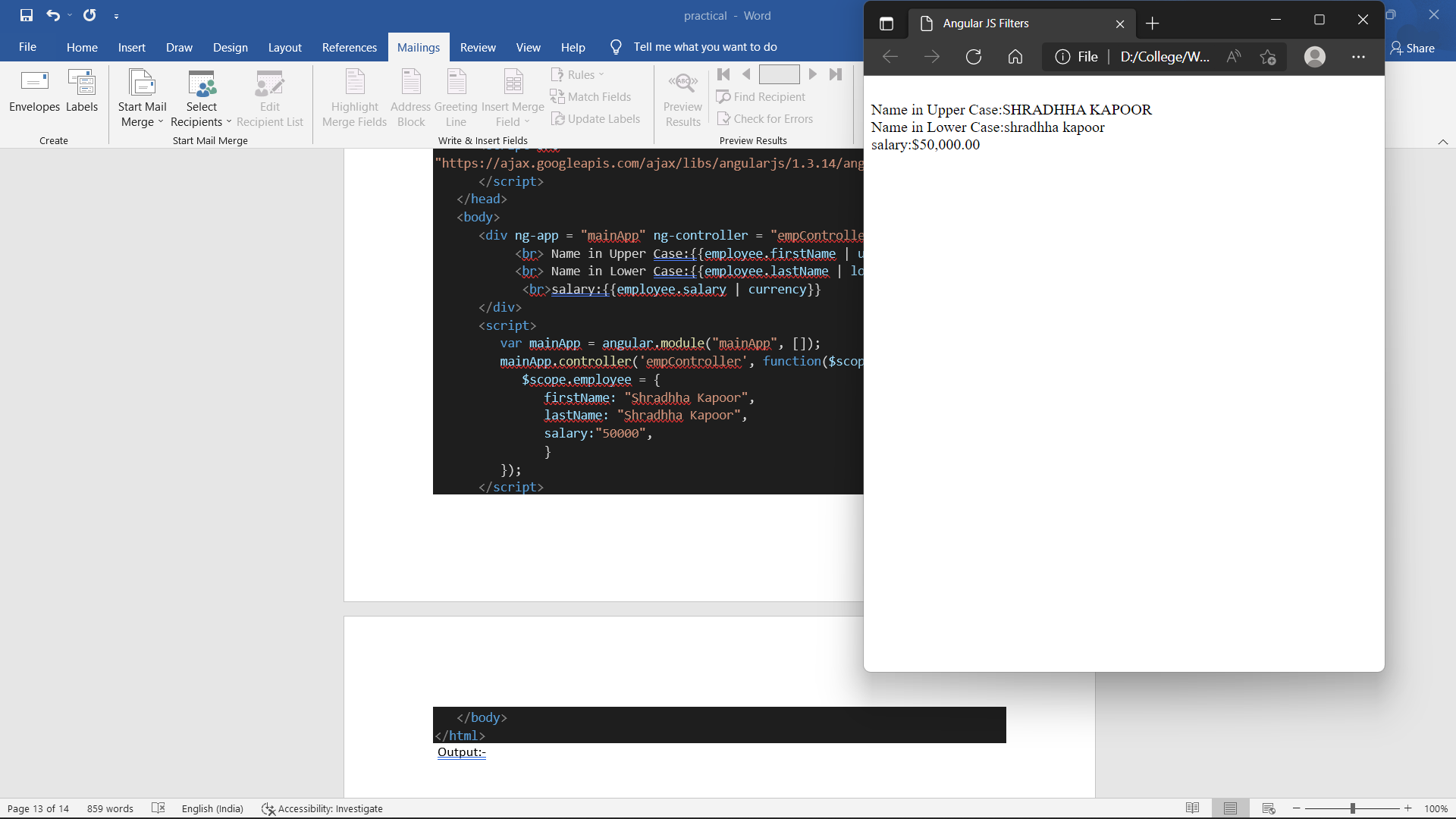
         });

      </script>

   </body>

</html>

**Output:-**



Practical No: 8

1. **Create an application to demonstrate directives**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**AIM:-** Directives are used to split the code of complex application into smaller pieces.

In Angular JS we used only one div tag

**Coding:-**

<html>

    <head>

        <meta charset="utf-8">

        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <title></title>

        <meta name="description" content="">

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

        <link rel="stylesheet" href="">

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

    </head>

    <body background-color="pink">

        <div ng-app="" ng-init="fn='Shraddha';ln='Kapoor'">

            <p> {{"Hello Everyone,"}}</p>

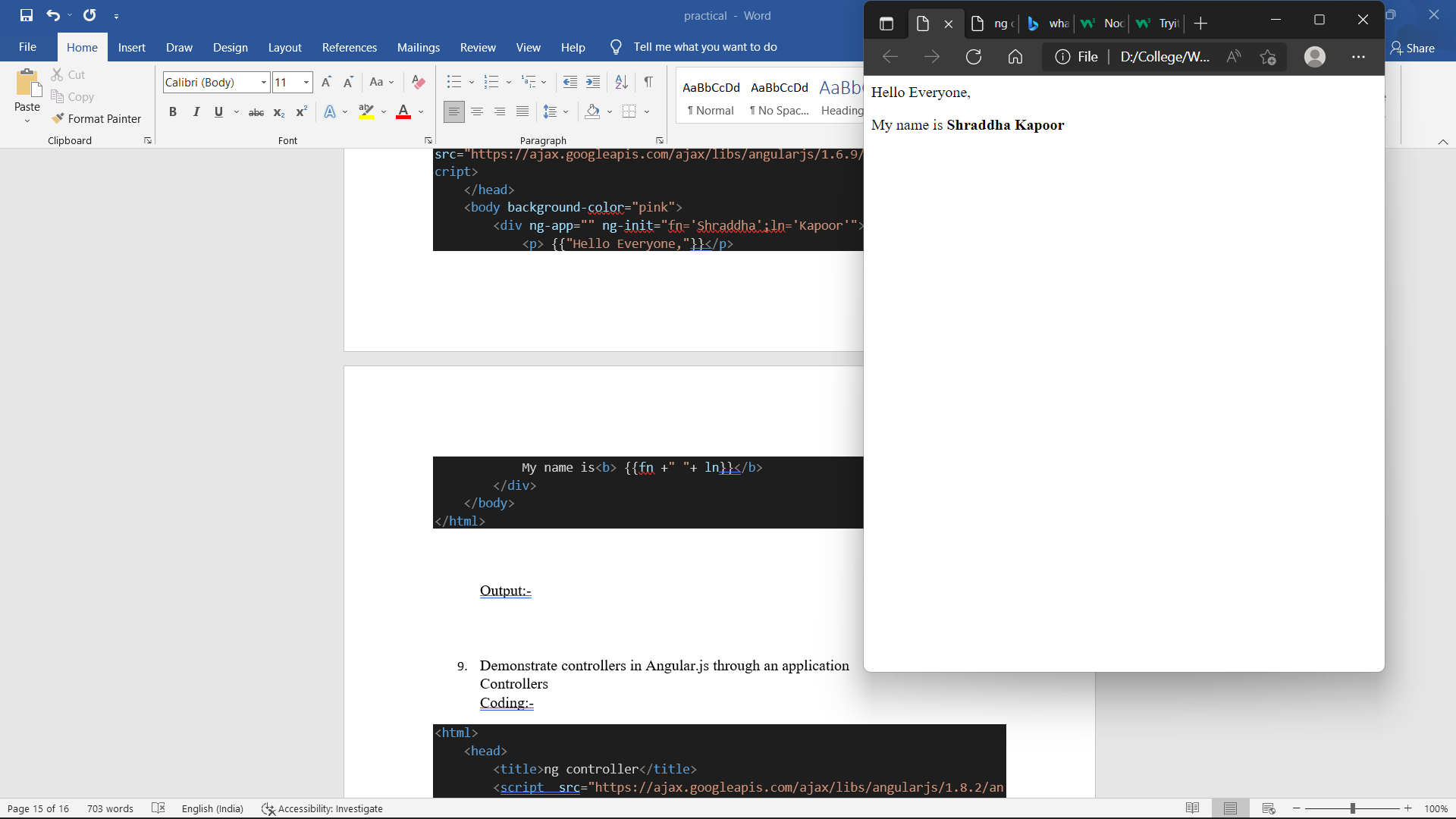
            My name is<b> {{fn +" "+ ln}}</b>

        </div>

    </body>

</html>

**Output:-**



Ng-mouseenter.

<html>

    <head>

        <script src="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>

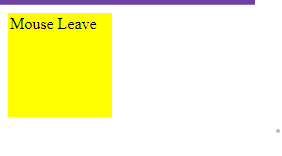
    </body>

</html>

**Output:**

Output:





Ng-mouseup

<html>

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

    </script>

    <body ng-app="">

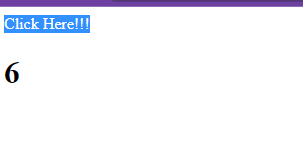
        <div ng-mouseup="count = count + 1" ng-init="count=0">Click Here!!!</div>

        <h1>{{count}}</h1>

        </body>

</html>

**Output:**



ng-mousedown

<html>

    <head>

        <title>

            Angular ng-mousedown Event

        </title>

        <script src="angular.min.js"></script>

        <script type="text/javascript">

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

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

                $scope.count = 0;

                $scope.getdetails = function () {

                    $scope.count = 0;

                    $scope.count = $scope.count + 1;

                }

            });

        </script>

    </head>

    <body>

        <div ng-app="ngmousedownApp" ng-controller="ngmousedownCtrl">

            <h2>AngularJs ng-mousedown Event Example.</h2>

            <input type="button"

            ng-mousedown="getdetails()" value="Click Here"><br /><br />

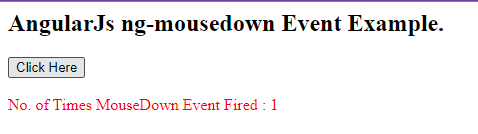
            <span style="color:Red">No. of Times MouseDown Event Fired : {{count}}</span>

        </div>

    </body>

</html>

Output:



Ng-Keydown

<html>

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

    <body ng-app="">

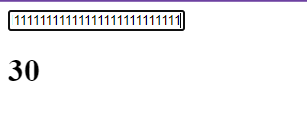
        <input ng-keydown="count = count + 1" ng-init="count=0"/>

        <h1>{{count}}</h1>

    </body>

</html>

**Output:**



Practical No: 9

1. **Demonstrate controllers in Angular.js through an application**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**AIM:-** Controllers used to attached controller to the view.

**Coding:-**

<html>

    <head>

        <title>ng controller</title>

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

    </head>

    <script>

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

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

            $scope.fname=""

            $scope.lname=""

            $scope.age=""

        });

    </script>

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

        Enter Your fname:<input type="text" ng-model="fname">

        <br>

        Enter Your 1name:<input type="text" ng-model="lname">

        <br>

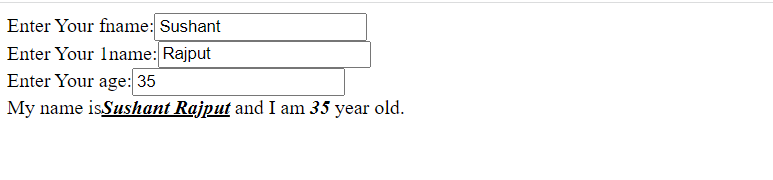
        Enter Your age:<input type="text" ng-model="age">

        <br>My name is<b><i><u>{{fname}} {{lname}}</u></i></b> and I am <b><i>{{age}}</i></b> year old.

    </body>

</html>

**Output:-**



Practical No: 10

1. **Demonstrate features of Angular.js forms with a program**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**AIM:-** May the forms are used for collecting information.

**Coding:-**

<html>

  <head>

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

  </head>

<body name ="form" ng-app="">

<form name="myForm">

Enter Your First Name: <input name="firstname" name="fn" placeholder="First Name" required>

<span style="color:red" ng-show="myForm.fn.$dirty && myForm.fn.$invalid">

  <span ng-show="myForm.fn.$error.required"></span>

  </span><br><br>

Enter Your Last Name:  <input type="text"  name="ln" placeholder="Last Name" required>

<span style="color:red" ng-show="myForm.ln.$dirty && myForm.ln.$invalid">

  <span ng-show="myForm.ln.$error.required"></span>

  </span><br><br>

Enter your Mail ID: <input type="email" placeholder="Email-ID" required><br><br>

Select your Previous Stream :<select>

                              <option value="">

                              <option value="Science">science

                              <option value="Commerce">Commerce

                            </select><br><br>

Select your class : MCA DIV-A <input type="radio" value="MCA Div-A">

                    MCA DIV-B<input type="radio" value="MCA Div-B"><br><br>

Select Your Subject :WT<input type="checkbox" value="WT" required>

ADBMS<input type="checkbox" value="ADBMS" required>

JAVA<input type="checkbox" value="JAVA" required>

DS<input type="checkbox" value="DS" required>

IOT<input type="checkbox" value="IOT" >

CC<input type="checkbox" value="CC" ><br><br>

<input type="submit"

ng-disabled="myForm.user.$dirty && myForm.user.$invalid ||

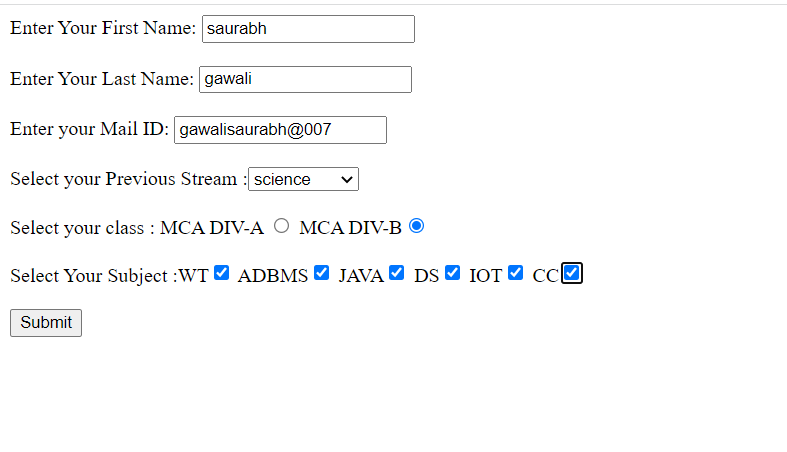
myForm.email.$dirty && myForm.email.$invalid">

</form>

</body>

</html>

**Output:-**



Basic Validation

<html>

<head>

<title>

AngularJs ng-minlength Form Validation Example

</title>

<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>

<script type="text/javascript">

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

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

$scope.sendForm = function () {

$scope.msg = "Form Validated";

};

});

</script>

</head>

<body>

<div ng-app="formApp" ng-controller="formCtrl">

<h1> Understandin AngularJs ng-minlength Form Validation with  Example</h1>

<form name="personForm" novalidate ng-submit="personForm.$valid &&sendForm()">

First Name:<input type="text" name="fname" ng-model="txtpin" ng-minlength="5" required />

<span style="color:Red" ng-show="personForm.fname.$error.required"> Required! </span>

<span style="color:Red" ng-show="personForm.fname.$dirty&&personForm.fname.$error.minlength">

     Minimum 5 Characters Required </span>

<br /><br />

<button type="submit">Submit Form</button><br /><br />

<span>{{msg}}</span>

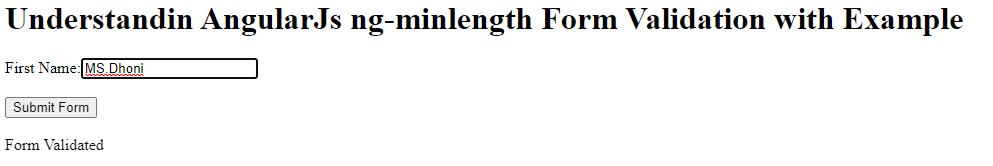
</form>

</div>

</body>

</html>

**Output:**



Input Validation Controls

<html>

<head>

<title>

AngularJs Form and Input Validation Controls

</title>

<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>

</head>

<body>

<div ng-app="">

<form name="personForm">

First Name: <input type="text" name="firstname" ng-model="person.fname" required />

<span ng-show="personForm.firstname.$touched && personForm.firstname.$invalid "> Required! </span><br /><br />

Last Name:<input type="text" name="lastname" ng-model="person.lname" required />

<span ng-show="personForm.lastname.$dirty && personForm.lastname.$valid"> Thanks for Text </span><br /><br />

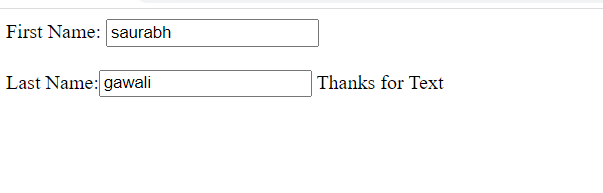
</form>

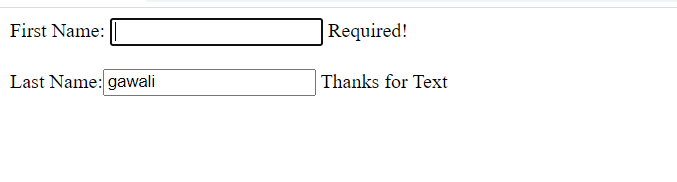
</div>

</body>

</html>

**Output:**





Form validation by using CSS in AngularJs

html>

<head>

<title>

AngularJs Form Input Fields Validation Example using CSS class

</title>

<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>

<script>

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

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

$scope.sendForm = function () {

$scope.msg='Form Submited Successfully';

};

$scope.getClass = function (color) {

return color.toString();

}

});

</script>

<style>

.valid.false {

background: red;

}

.valid.true {

background: green;

}

.error {

color: red;

}

</style>

</head>

<body ng-app="formApp" ng-controller="formCtrl">

<h3>Form validation by using CSS in AngularJs</h3>

<form name="personForm" ng-submit="sendForm()">

<label for="name">Name</label>

<input id="name" name="name" type="text" ng-model="person.name" required />

<span class="error" ng-show="personForm.name.$error.required"> Required! </span>

<br />

<br />

<label for="adress">Adress</label>

<input id="address" name="address" type="text" ng-model="person.address" required />

<span class="error" ng-show="personForm.address.$error.required"> Required! </span>

<br /><br />

<label for="contact">Contact No</label>

<input id="mobile" name="mobile" type="number" ng-model="person.mobile" required />

<span class="error" ng-show="personForm.mobile.$error.required">Required number!</span>

<span class="error" ng-show="personForm.mobile.$error.mobile">Invalid mobile!</span>

<br /><br />

<label for="email">Email</label>

<input id="email" name="email" type="email" ng-model="person.email" required />

<span class="error" ng-show="personForm.email.$error.required">Required!</span>

<span class="error" ng-show="personForm.email.$error.email">Invalid Email!</span>

<br /><br />

<input type="checkbox" ng-model="terms" name="terms" id="terms" required />

<label for="terms">I Agree to the terms.</label>

<span class="error" ng-show="personForm.terms.$error.required">You must agree to the terms</span>

<br /><br />

<button type="submit">Submit Form</button>

<br /><br />

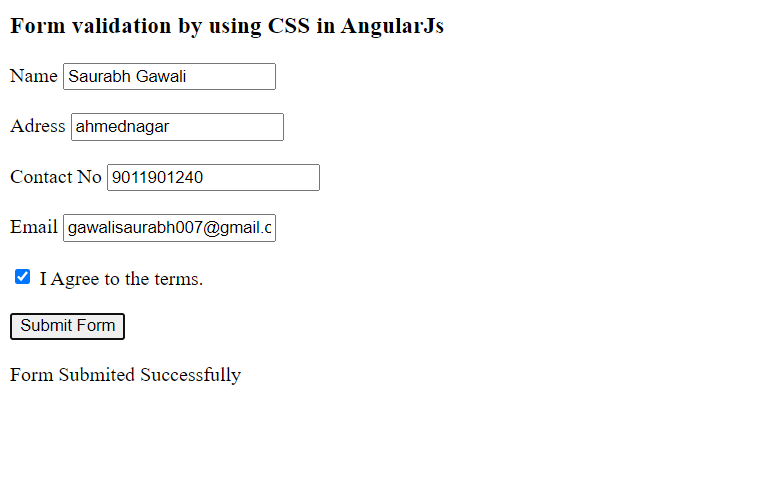
<span>{{msg}}</span>

</form>

</body>

</html>

Output:



Practical No: 11

1. **Create a SPA (Single Page Application)**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**AIM:-** SPA used for work faster

**Coding:-**

<html>

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

<body ng-app="" >

    <h3>

    <a href="home.html">HOME</a>

    <a href="about.html">ABOUT US</a>

    <a href="contact.html">CONTACT US</a>

    </h3>

</body>

</html>

<html>

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

<body>

<div ng-app="">

    <h1>This IS HOME page......</h1>

    <br>

    <br>

</div>

<button><a href="spa.html"> BACK </button>

</body>

</html>

<html>

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

<body>

<div ng-app="">

    <h1>This is ABOUTus pageee...</h1>

    <br><br>

</div>

<button><a href="spa.html"> BACK </button>

</body>

</html>

<html>

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

<body >

<div ng-app="">

<h1>This IS CONTACT Us pageeeeee</h1>

<br><br>

</div>

<button><a href="spa.html"> BACK </button>

</body>

</html>

**Output:-**

