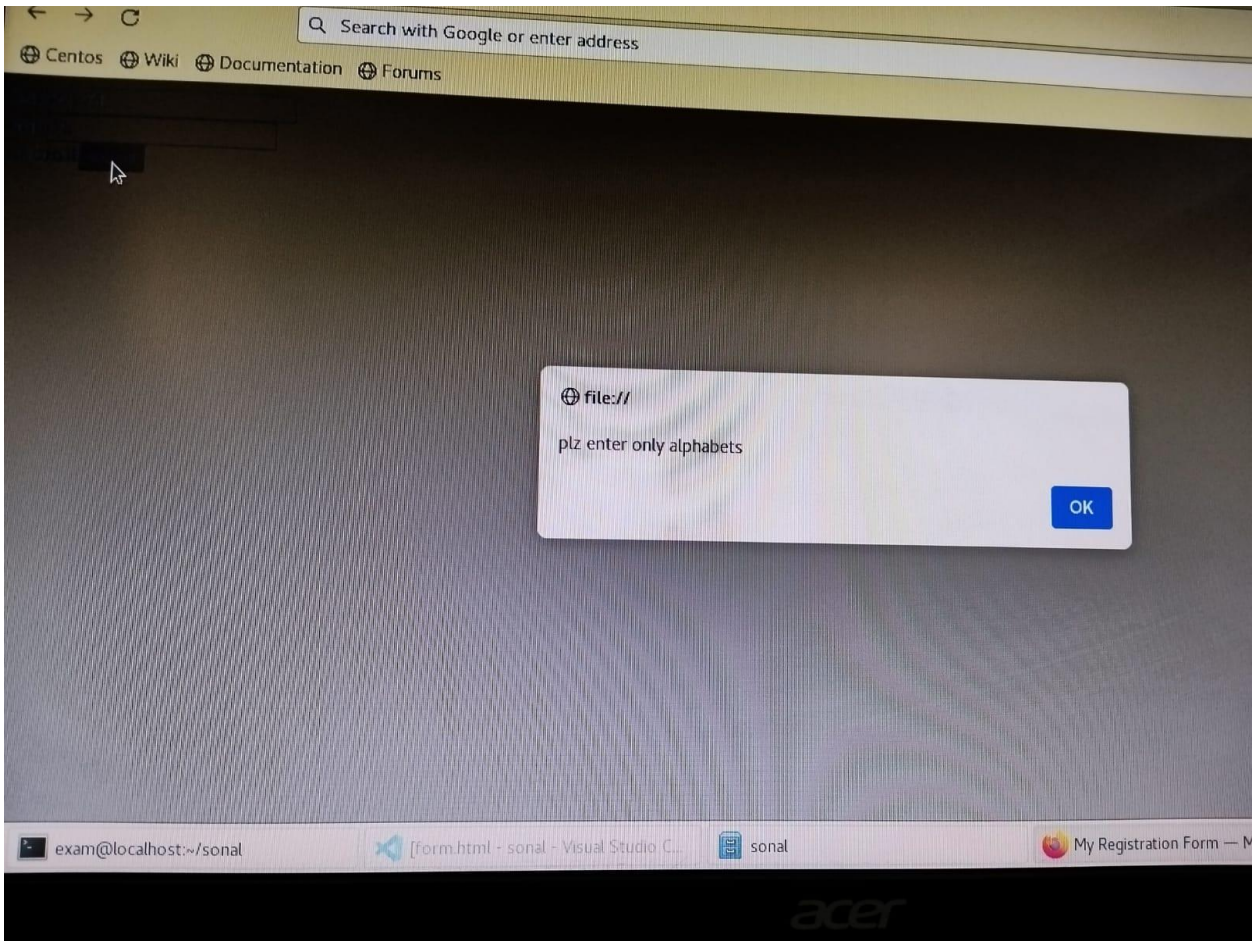
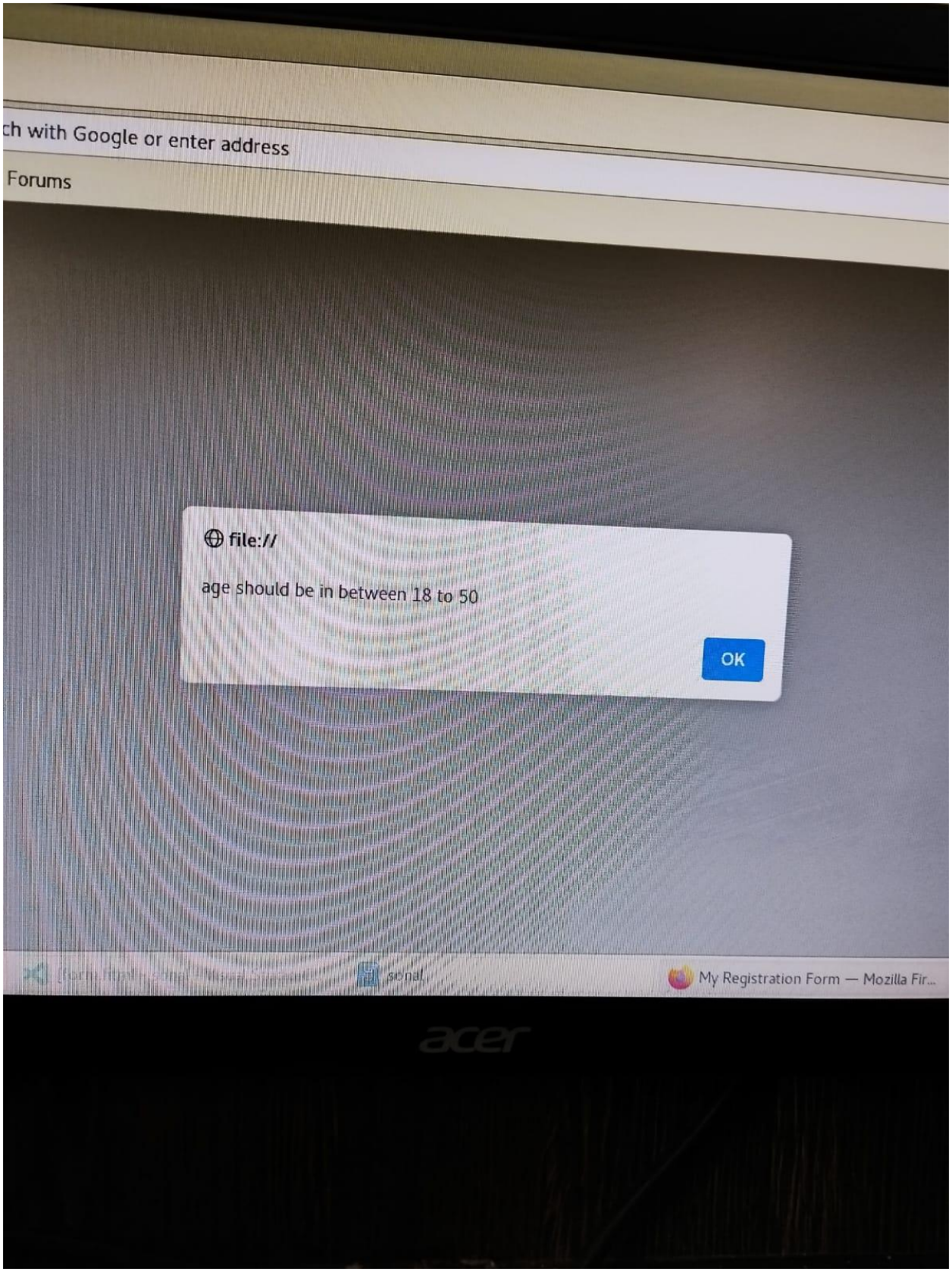
1. Create an HTML form that contain the Student Registration details and write a JavaScript to validate Student first and last name as it should not contain other than alphabets and age should be between 18 to 50.

```
<html>
    <head>
        <title>
            Registration Form
        </title>
        <script>
            function f()
            {
                var fname=document.myform.fname.value;
                var lname=document.myform.lname.value;
                var age=document.myform.age.value;

                if(!(/[a-zA-Z]+/).test(fname))
                {
                    alert("please enter valid first name");
                    return false;
                }

                if(!(/[a-zA-Z]+/).test(lname))
                {
                    alert("please enter valid last name");
                    return false;
                }

                if(isNaN(age)||age<18||age>60)
                {
                    alert("please enter age between 18 to 60");
                    return false;
                }
            }
        </script>
    </head>
```

```
    <body>
        <form name="myform" onsubmit="f()">
            fname<input type="text" name="fname"><br><br>
            lname<input type="text" name="lname"><br><br>
            age<input type="text" name="age" ><br><br>
            <input type="submit" value="submit">



        </form>
    </body>
</html>
```

Output:



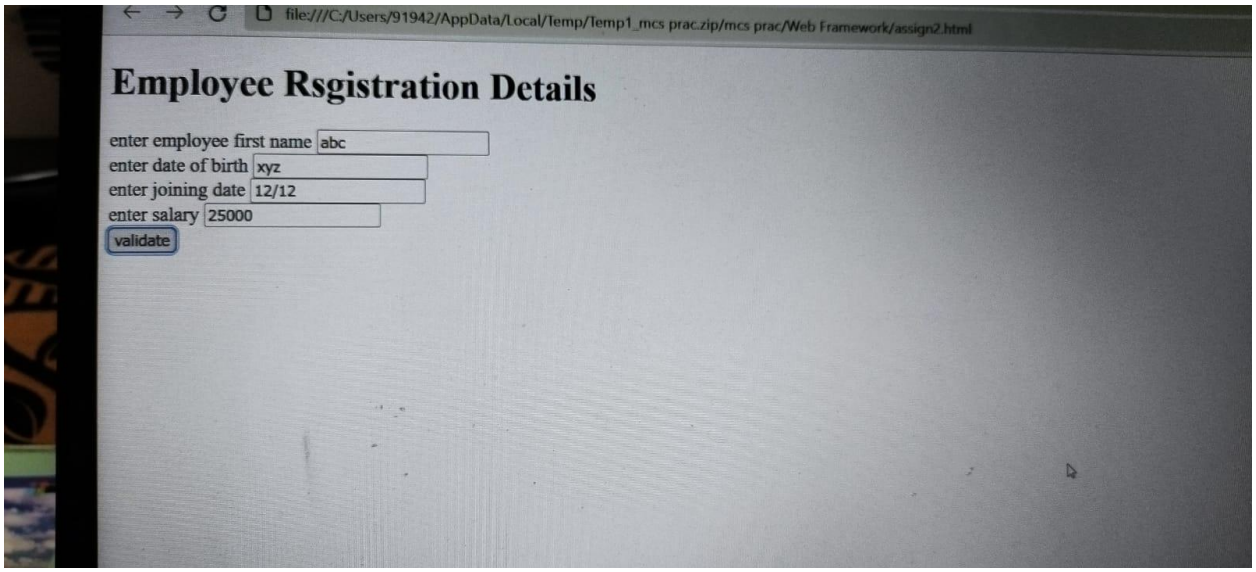

ch with Google or enter address
Forums
file://
age should be in between 18 to 50
OK
My Registration Form — Mozilla Fir...

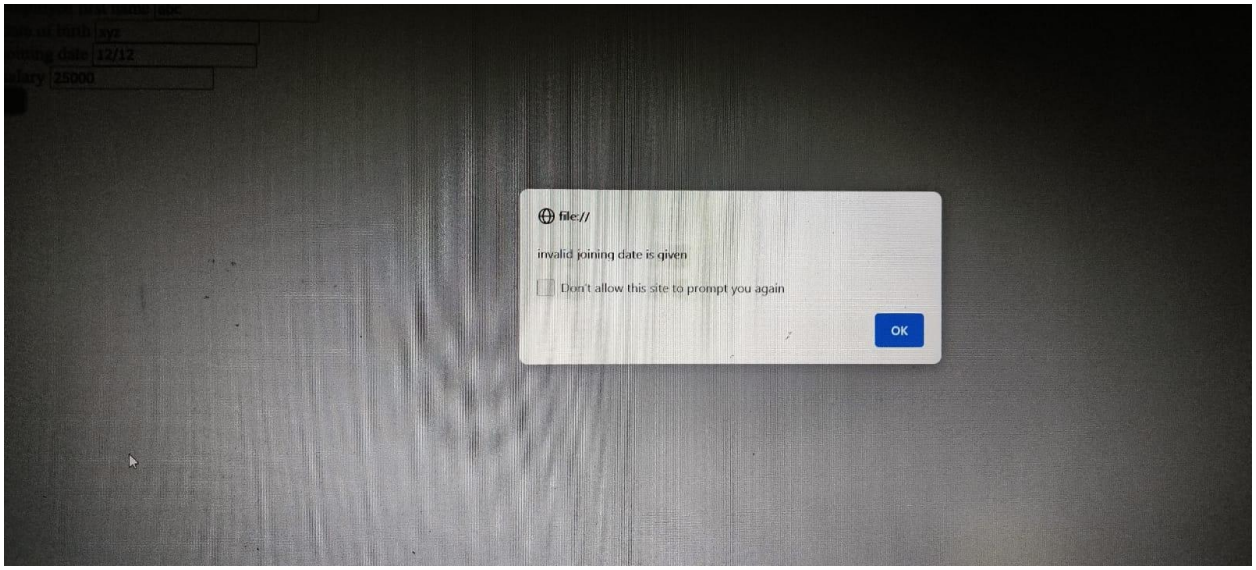
2.Create an HTML form that contain the Employee Registration details and write a JavaScript to validate DOB, Joining Date, and Salary.

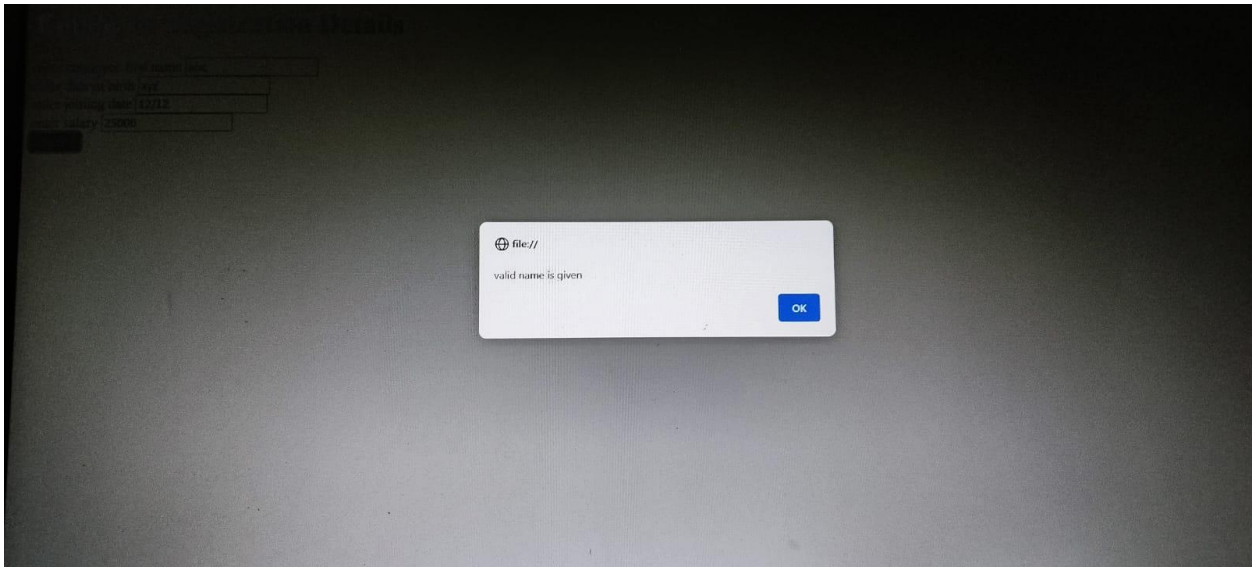
```
<html>
        <head>
        <title>Employee Form</title>
        <body>
        <form name="employee_form" onsubmit="f()"
align="center">
                Name:<input type="text" id="name"/><br/>
                DOB:<input type="text" id="dob"/><br/>
                Date Of Joining:<input type="text"
id="doj"/><br/>
                Salary:<input type="text" id="sal1"/><br/>
                Submit<input type="submit" id="submit"
value="Submit"/><br/>
        </form>
        </body>
        <script>
        function f(){
                var
pattern=/^(0?[1-9]|[1-2][0-9]|3[0-1])[\/](0?[1-9]|1[0-2])[\/]\d{
4}$/;
                var
dob=document.getElementById("dob").value;
                if(!pattern.test(dob))
                alert("Enter valid dob");
                var
doj=document.getElementById("doj").value;
                  if(!pattern.test(doj))
                alert("Enter valid doj");
```

```
                var
sal=document.getElementById("sal1").value;
            if(isNaN(sal)||sal<18000||sal>50000)
                alert("Enter valid salary");
                else
                        alert("Submitted Successfully");
        }
        </script>
        </head>
</html>
```

Output:







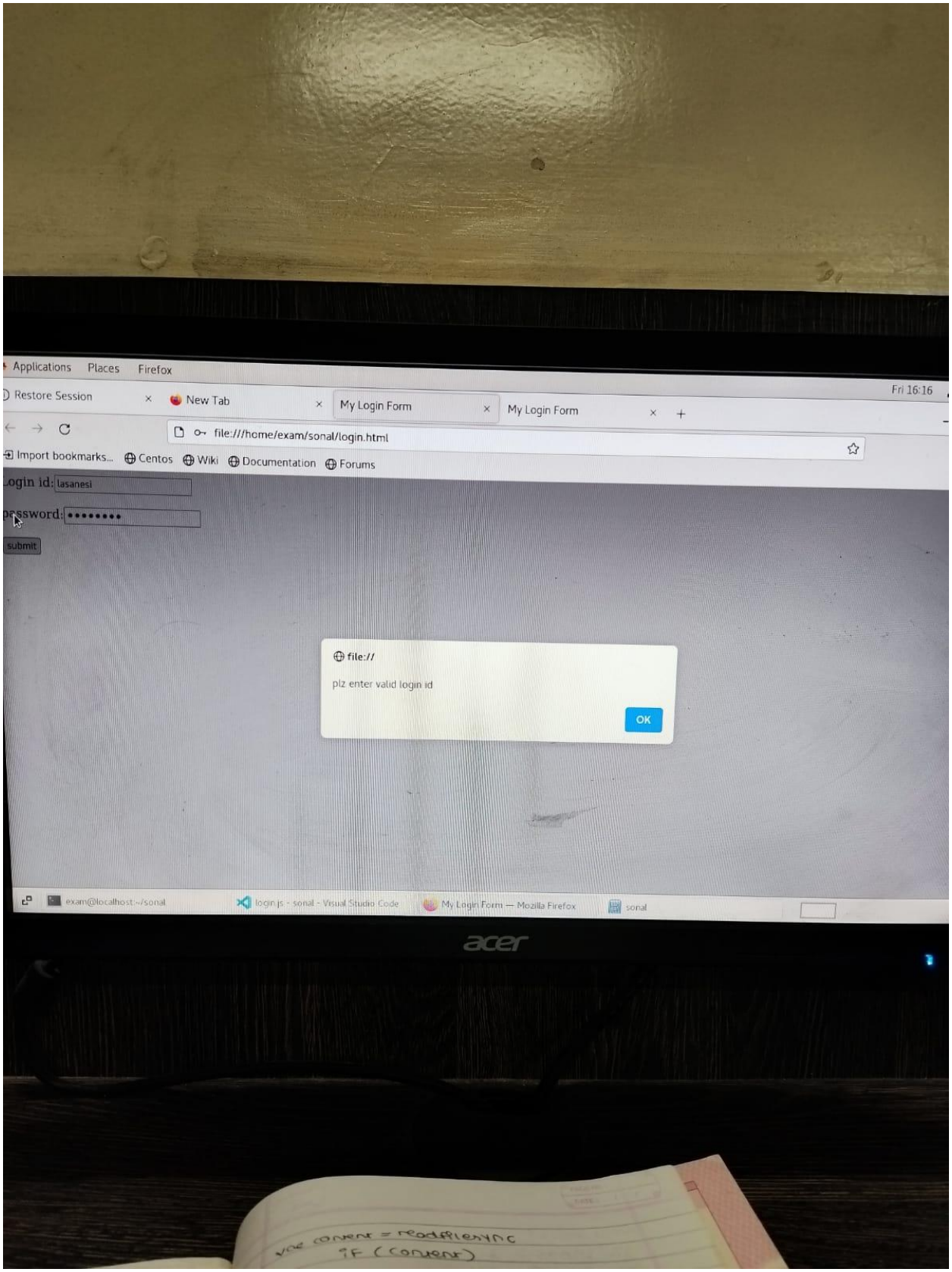
3. Create an HTML form for Login and write a JavaScript to validate email ID using Regular Expression.

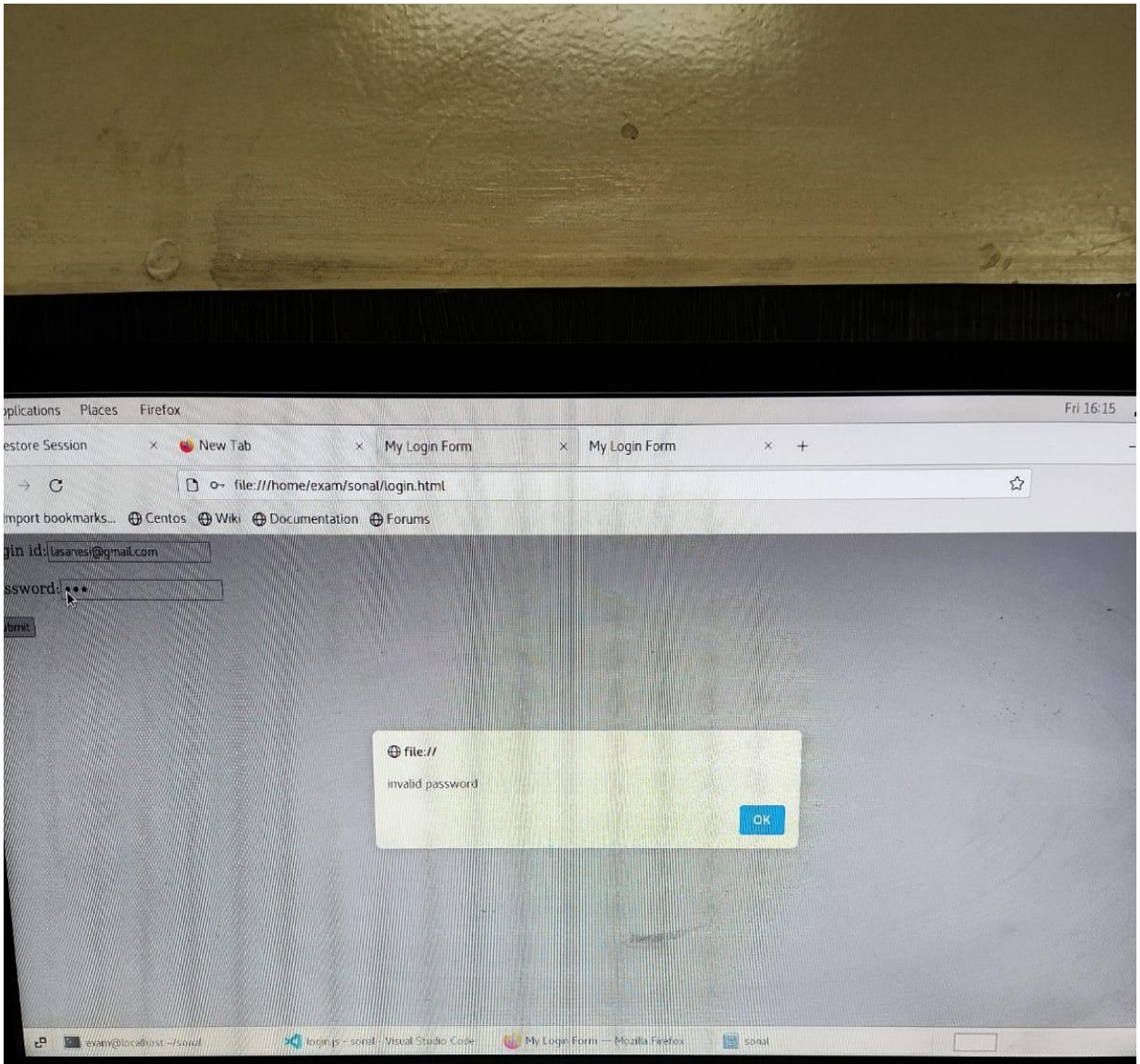
```
<!DOCTYPE html>

<html lang="en">
```

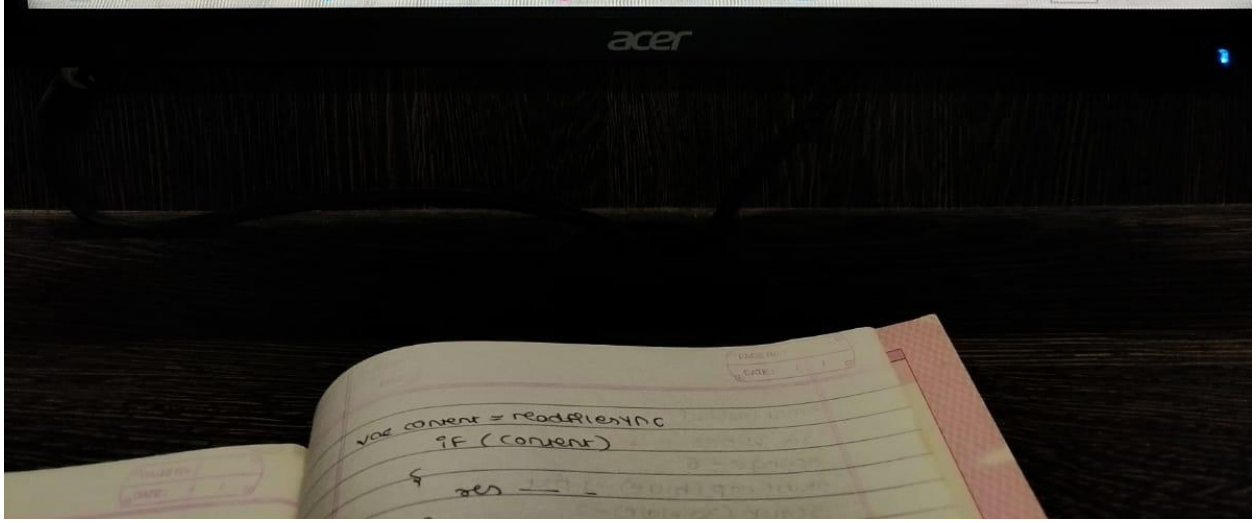
```
<head>

    <meta charset="UTF-8">

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

    <meta name="viewport" content="initial-scale=1.0">

    <title>my login form</title>

    <script>

        function f()

        {

            var lid=document.getElementById("lid").value;

            var pwd=document.getElementById("pwd").value;


            if(!(/([a-zA-Z0-9])+@([a-zA_Z])+.([a-zA_Z])/.test(lid))){

                alert("plz enter valid id");

                return false;

            }

            var l=password.length;

            if(password==null ||l<8)

            {

                alert("invalid password");

                return false;

            }


            alert("successfully login...");
```

```
        }

    </script>

</head>

<body>

    loginid:<input type="text" name="login id" id="lid"><br>

    password:<input type="password" name="password" id="pwd"><br>



    <input type="button" value="submit" onclick="f()">

</body>

<script>

    console.log("hello world");

    </script>

</html>
```

Output:

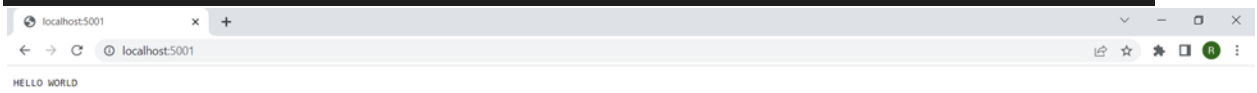

Applications Places Firefox
Fri 16:16
Restore Session
New Tab
My Login Form
My Login Form
file:///home/exam/sonal/login.html
Import bookmarks... Centos Wiki Documentation Forums
Login id: lasanesi
password: ••••••••
submit
file://
plz enter valid login id
OK
exam@localhost:~/sonal
login.js - sonal - Visual Studio Code
My Login Form — Mozilla Firefox
sonal
acer


Applications Places Firefox
Fri 16:15
Restore Session
New Tab
My Login Form
My Login Form
file:///home/exam/sonal/login.html
Import bookmarks...
Centos
Wiki
Documentation
Forums
Login id: lasanesi@gmail.com
Password:
Submit
file://
invalid password
OK
exam@localhost:~/sonal
login.js - sonal - Visual Studio Code
My Login Form — Mozilla Firefox
sonal


var content = readfilesync
if (content)
{
res

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

```
var http = require('http');  // includes the http module

var uc = require('upper-case'); // include the upper-case module

http.createServer(function (req, res) {

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

    res.write(uc("hello world!")); // assign the upper-case module

    res.end();

}).listen(5001);
```

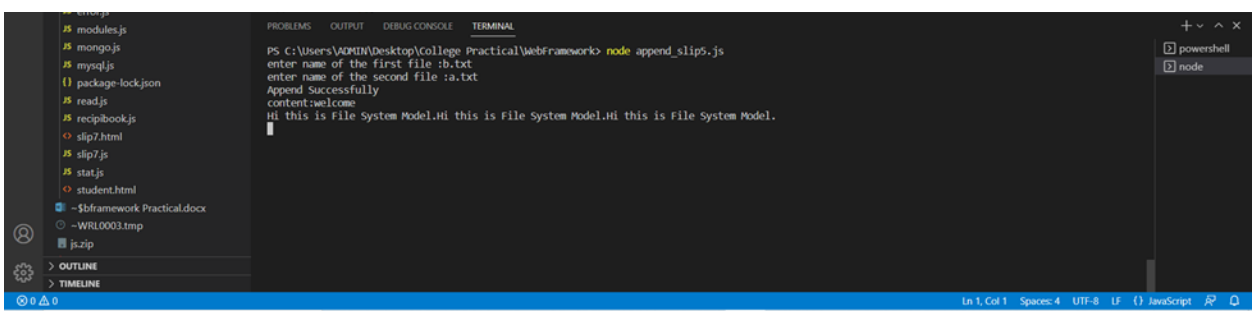


5.Using nodejs create a web page to read two file names from user and append contents of first file into second file.

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

```
console.log("\nFile Contents of file before append:",
a=fs.readFileSync("file1.txt", "utf8"));

fs.appendFile("file2.txt", a, (err) => {
if (err) {
    console.log(err);
}
else {

    console.log("\nFile Contents of file after append:",
    fs.readFileSync("file2.txt", "utf8"));
}
});
```

output:

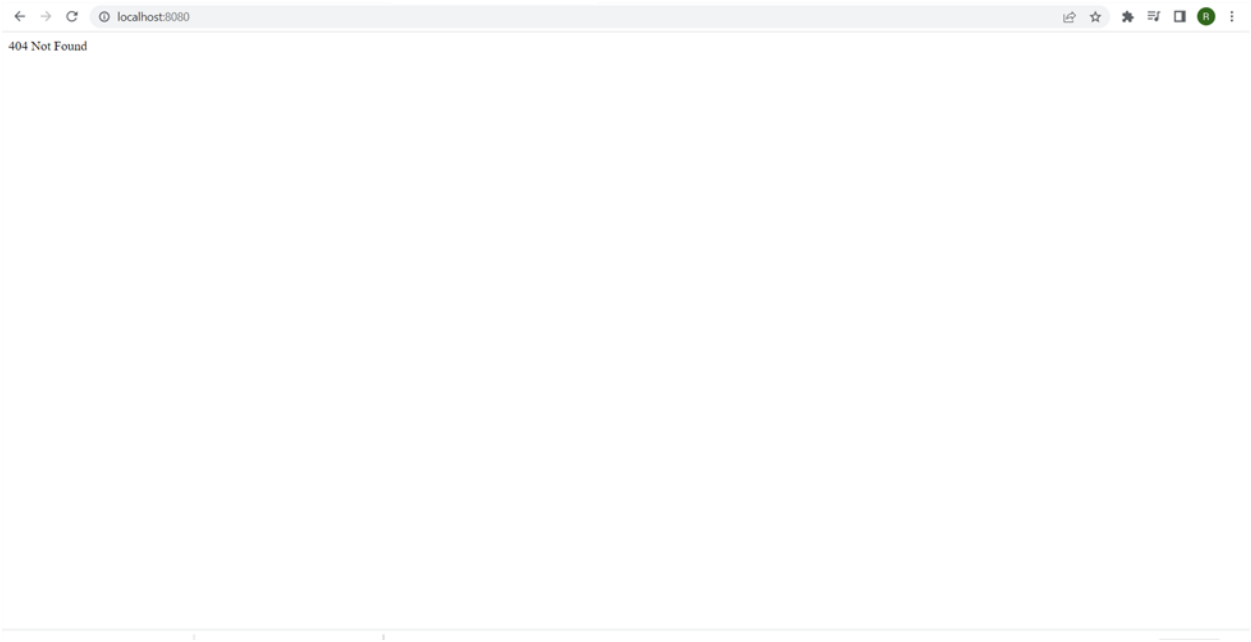


6.Create a Node.js file that opens the requested file and returns the content to the client. If anything goes wrong, throw a 404 error

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


var url = require('url');


var fs = require('fs');





http.createServer(function (req, res) {


  var q = url.parse(req.url, true);


  var filename = "." + q.pathname;


  fs.readFile(filename, function(err, data) {


    if (err) {


      res.writeHead(404, {'Content-Type': 'text/html'});
```

```
      return res.end("404 Not Found");


    }


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


    res.write(data);


    return res.end();


  });


}).listen(8080);
```

Output:



7.Create a Node.js file that writes an HTML form, with an upload field

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


 http.createServer(function (req, res) {


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


  res.write('<form action="fileupload" method="post"
enctype="multipart/form-data">');


  res.write('<input type="file" name="filetoupload"><br>');


  res.write('<input type="submit">');
```

```
  res.write('</form>');



  return res.end();



}).listen(8080);
```

Output:

8.Create a Node.js file that demonstrate create database and table in MySQL.

```
var mysql=require("mysql");



var con=mysql.createConnection({

    host:"localhost",user:"root",password:"password"});
```

```
    con.connect(function(err)
    {
        if (err) throw err;
        console.log("connected");
        con.query("create database mydb",function(err)
        {
            if(err) throw err;
            else{
                console.log("created database");
                con.query("use mydb",function(err){
                })
            }
        })
    })
con.query("create table student(rno int primary key,name text)",
                        function(err){
                            if(err)
                            console.log("create table failed");
                            else
```

```
                                {
                                    con.query("insert into student
values(1,'sonal'),(2,'samarth'),(3,'prasad')",
                                    function(err)
                                    {
                                        if(err)
                                        console.log("insert values
failed!");
                                        else{
                                            console.log("insert values
successfully!");
                                            con.query("select * from
student",function(err,result){
                                                if(err) throw err;
                                                console.log(result);


                                            });
                                        }
                                    })
                                }
                                con.query("update student set
name='lasane' where rno=1",function(err){
                                    if(err) throw err;
                                    console.log("updated successfully");
                                });
```

```
});
```

9.Create a node.js file that Select all records from the "customers" table, and display the result object on console

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


var con = mysql.createConnection({
  host: 'localhost',
  user: "root",
  password: "sonal123",
  database:'employee'
});


con.connect(function(err) {
  if (err) throw err;
```

```
  console.log("Connected!");

});



con.query('SELECT * FROM emp', (err,rows) => {

    if(err) throw err;



    console.log('Data received from Db:');

    console.log(rows);

  });
```

10.Create a node.js file that Insert Multiple Records in "student" table, and display the result object on console.

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









var con = mysql.createConnection({

host: "localhost",

user: "root",

password: "sonal123",

database: "node"

});

con.connect(function(err) {
```

```
if (err) throw err;

console.log("Connected!");

var sql = "INSERT INTO student (rollno,name, percentage) VALUES ?";

var values = [



[1,'abc', 77.6],



[2,'def', 89.6],



[3,'ghi', 91.6]



];



con.query(sql, [values], function (err, result)



 {



    if (err) throw err;



    console.log("Number of records inserted: " +
result.affectedRows);



  });
```

```
con.query("SELECT * FROM student", function (err, result, fields) {


    if (err) throw err;


    console.log(result);


  });


});
```

11.Create a node.js file that Select all records from the "customers" table, and delete the specified record.

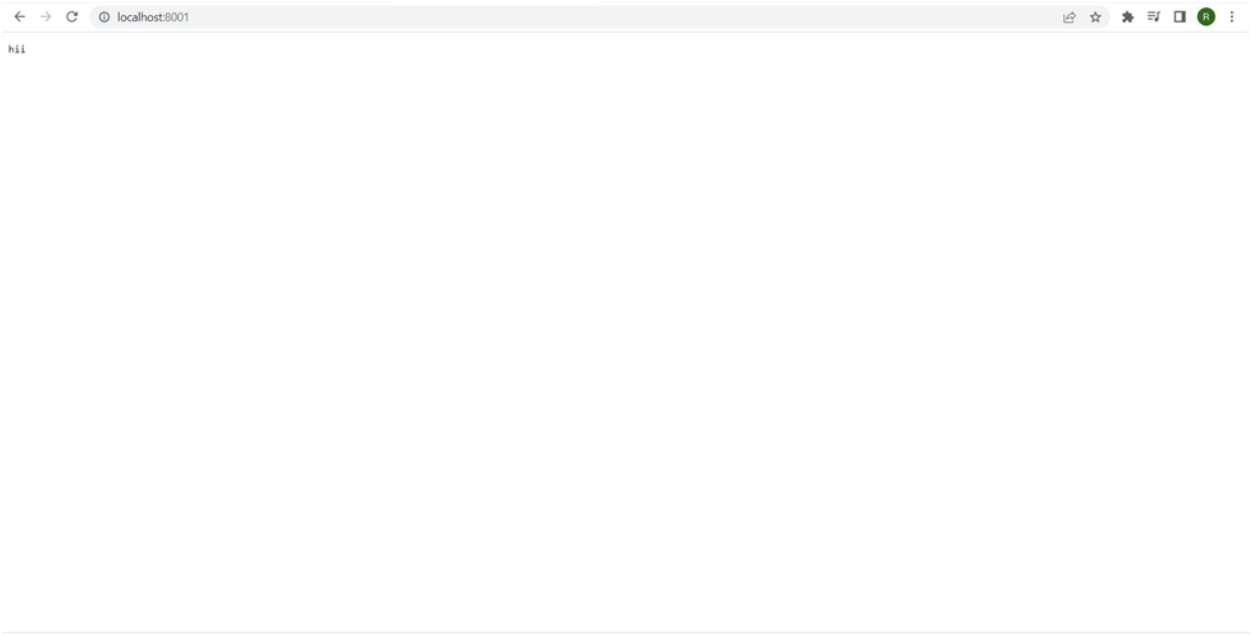
```
var mysql = require('mysql');


var con = mysql.createConnection({

  host: "localhost",

  user: "root",

  password: "sonal123",

  database: "employee"

});


con.connect(function(err) {

  if (err) throw err;
```

```
  var sql = "DELETE FROM emp WHERE name = 'dada'";

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

    if (err) throw err;

    console.log("Number of records deleted: " +
result.affectedRows);

  });

});
```

12.Create a Simple Web Server using node js.

```
var http = require('http'); // 1 - Import Node.js core module


var server = http.createServer(function (req, res) {   // 2 -
creating server


    //handle incomming requests here..




});


server.listen(5001); //3 - listen for any incoming requests


console.log('Node.js web server at port 5000 is running..')
```

Output:

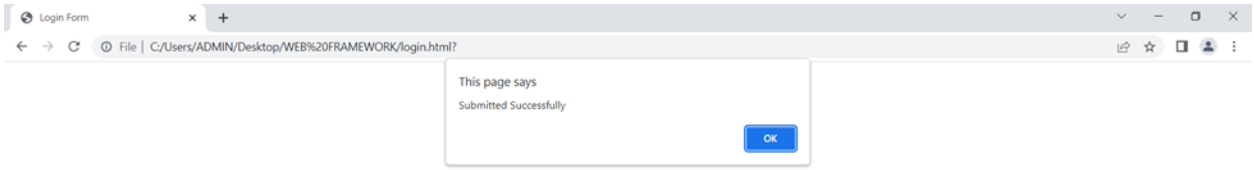


13.Using node js create a User Login System.

```
<html>
  <head>
      <title>
      login page
      </title>
      <script>
      function f(){
            var validRegex
=/^[a-zA-Z0-9.!#$%&'+/=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0
-9-]+)$/;
            var email=document.getElementById("email").value;
            if(!validRegex.test(email))
            alert("please enter valid email id");
            else
            alert("submitted succesfully");
            return false;
```

```
        }
        </script>
        </head>
        <body>
                <center><b>
                <form name="login" onsubmit="f()">
                <h1>Login here</h1>
                email id:<input type="text" id="email"/><br><br>
                password:<input type="password"
id="password"><br><br>
                submit<input type="submit" id="submit"
value="submit">
        </form>
        </b>
        </center>
</body>
</html>
```

Output:



14.Using node js create a eLearning System.

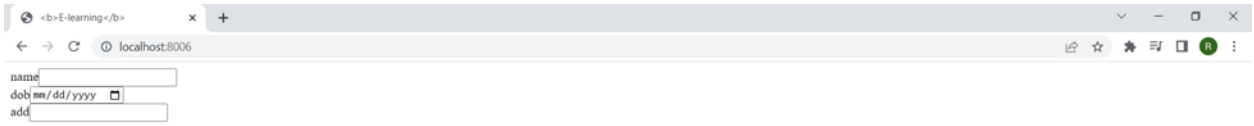
Elearning.html

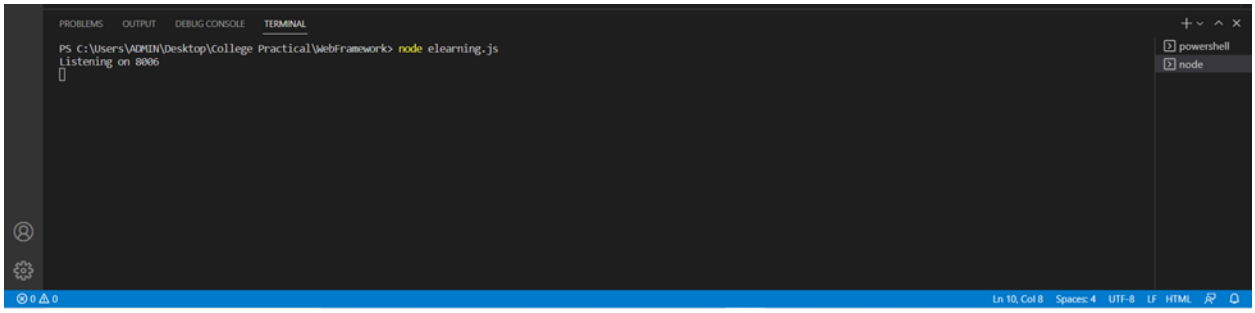
```
<html>
        <head>
       <title><b>E-learning</b></title>
        </head>
        <body>
        <img src="/home/exam/Downloads/image.png"/><br>
        name<input type="text"><br>
        dob<input type="date"><br>
        add<input type="text"><br>
        </body>
</html>
```

Elearning.js

```
var fs=require("fs");
var http = require('http');
http.createServer(function(req,resp)
{
   resp.writeHead(200,{"content-type":"text/html"});
        var content=fs.readFileSync("elearning.html");
        if(content)
        {
        resp.write(content);
        }
        else
        {
        resp.write("404 error");
        }
        resp.end()
}).listen(8006);
```

Output:





15.Using node js create a Recipe Book.

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

var http=require('http');

var con=http.createServer(function(req,res){

        if(req.url=='/')

        {

                fs.readFile('demo.html',function(err,data)
```

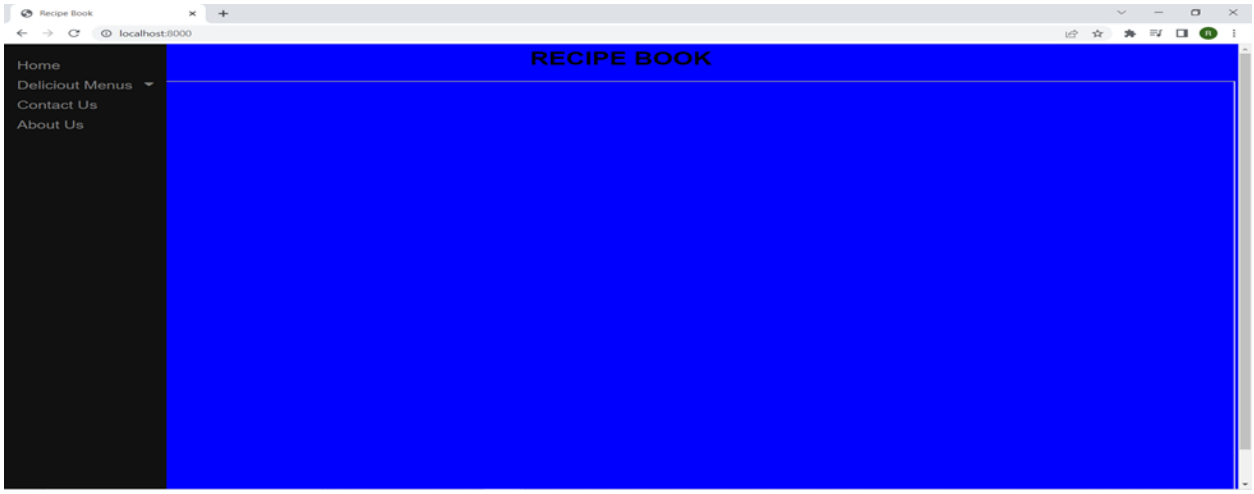
```
        {
                res.writeHead(200,{'content-Type':'text/html'});
                res.write(data);
                res.end();
        });
}

else if(req.url=="/contact")
{
        fs.readFile("contact.html",function(err,data){
                res.writeHead(200,{'Content-Type':'text/html'});
                res.write(data);
                res.end();
        });
}
else if(req.url=="/about")
{
        fs.readFile("about.html",function(err,data){
                res.writeHead(200,{'Content-Type':'text/html'});
                res.write(data);
                res.end();
```

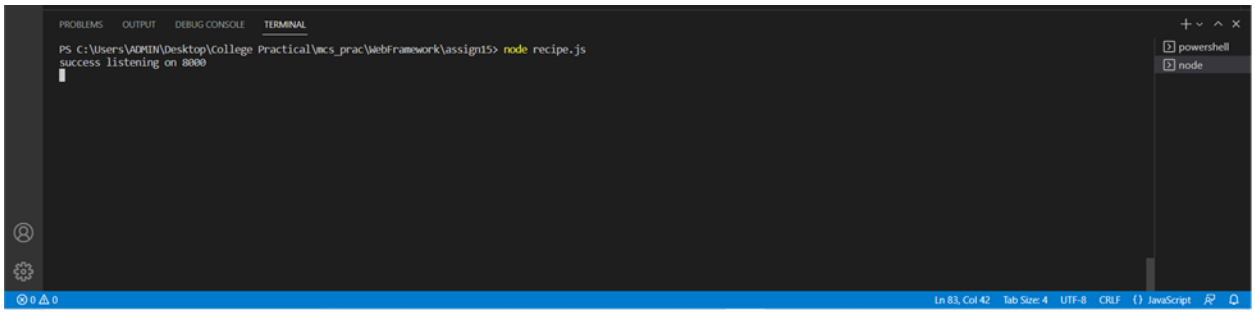
```
            });
        }
        else if(req.url=="/snacks")
        {
            fs.readFile("recipe_book.pdf",function(err,data){

res.writeHead(200,{'Content-Type':'application/pdf'});
                res.write(data);
                res.end();
            });
        }
        else if(req.url=="/cake")
        {
            fs.readFile("recipe_book.pdf",function(err,data){

res.writeHead(200,{'Content-Type':'application/pdf'});
                res.write(data);
                res.end();
            });
        }
        else if(req.url=="/rice")
        {
```

```
                fs.readFile("rice.pdf",function(err,data){

res.writeHead(200,{'Content-Type':'application/pdf'});
                        res.write(data);
                        res.end();
                });
        }
        else if(req.url=="/chicken")
        {
                fs.readFile("recipe_book.pdf",function(err,data){
                        res.writeHead(200,{'content-Type':'application/pdf'});
                        res.write(data);
                        res.end();
                });
        }
        else if(req.url=="/other")
        {
                fs.readFile("recipe_book1.pdf",function(err,data){
                        res.writeHead(200,{'content-Type':'application/pdf'});
                        res.write(data);
                        res.end();
                });
```

```
        }
        else if(req.url.match(".\jpg$"))
        {
                var filestream=fs.createReadStream("recipe.jpg");
                res.writeHead(200,{'Content-Type':'image/jpg'});
                filestream.pipe(res);
        }

        else
        {
                res.end("The end");
        }

}).listen(8000);
console.log("success listening on 8000");
```

Output:



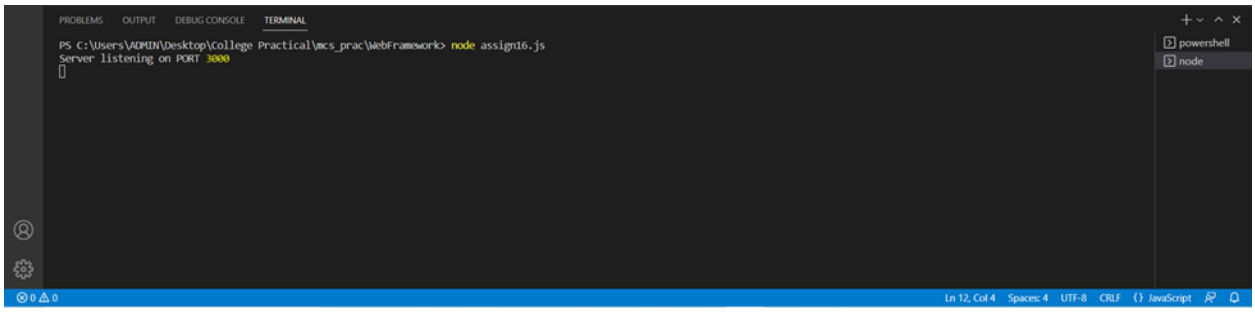


16.write node js script to interact with the filesystem, and serve a web page from a file.

```
var express = require('express');
var app = express();
var PORT = 3000;

app.get('/', function(req, res){
        res.download('hello.txt');
});

app.listen(PORT, function(err){
        if (err) console.log(err);
        console.log("Server listening on PORT", PORT);
```

```
});
```

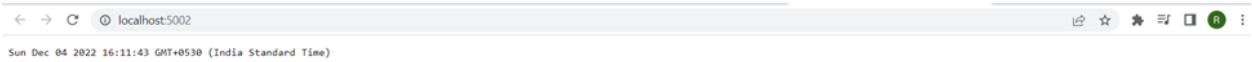
OUTPUT:



17.Write node js script to build Your Own Node.js Module. Use require (‘http’) module is a built-in Node module that invokes the functionality of the HTTP library to create a local server. Also use the export statement to make functions in your module available externally. Create a new text file to contain the functions in your module called, “modules.js” and add this function to return today’s date and time.

```
modules.js
module.exports.dt= new Date();

date.js
var http=require("http");
var d=require("./modules.js");
var s=http.createServer(function(req,resp){
   resp.writeHead(200,{"content_type":"text/plain"});
        resp.write(d.dt.toString());
        resp.end();
});
s.listen(5002);
console.log("open link http://localhost:5002/");
```

Output:



18.Create a js file named main.js for event-driven application. There should be a main loop that listens for events, and then triggers a callback function when one of those events is detected.

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

var myeventEmitter=new events.EventEmitter();




    myeventEmitter.on('myevent',function Listener1(){

        console.log('first event listener');


    });
```

```
        myeventEmitter.on('myevent',function Listener2(...args){



            console.log('listener2 executed.having parameters
${parameters}');

        });



        console.log(myeventEmitter.listeners('myevent'));



    myeventEmitter.emit('myevent',10,20);
```

Output:

19.Write node js application that transfer a file as an attachment on web and enables browser to prompt the user to download file using express js.

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

```javascript
var app=express();

app.get('/',function(req,resp){

    resp.download('hello.js');

});

app.listen(8051,function(err){

    if(err)

    console.log(err);



    console.log("server is running at http://127.0.0.1:8051");

});
```