Name:- Sejal Godse

Roll No. :- SH011

Subject :- Web Framework Practicle

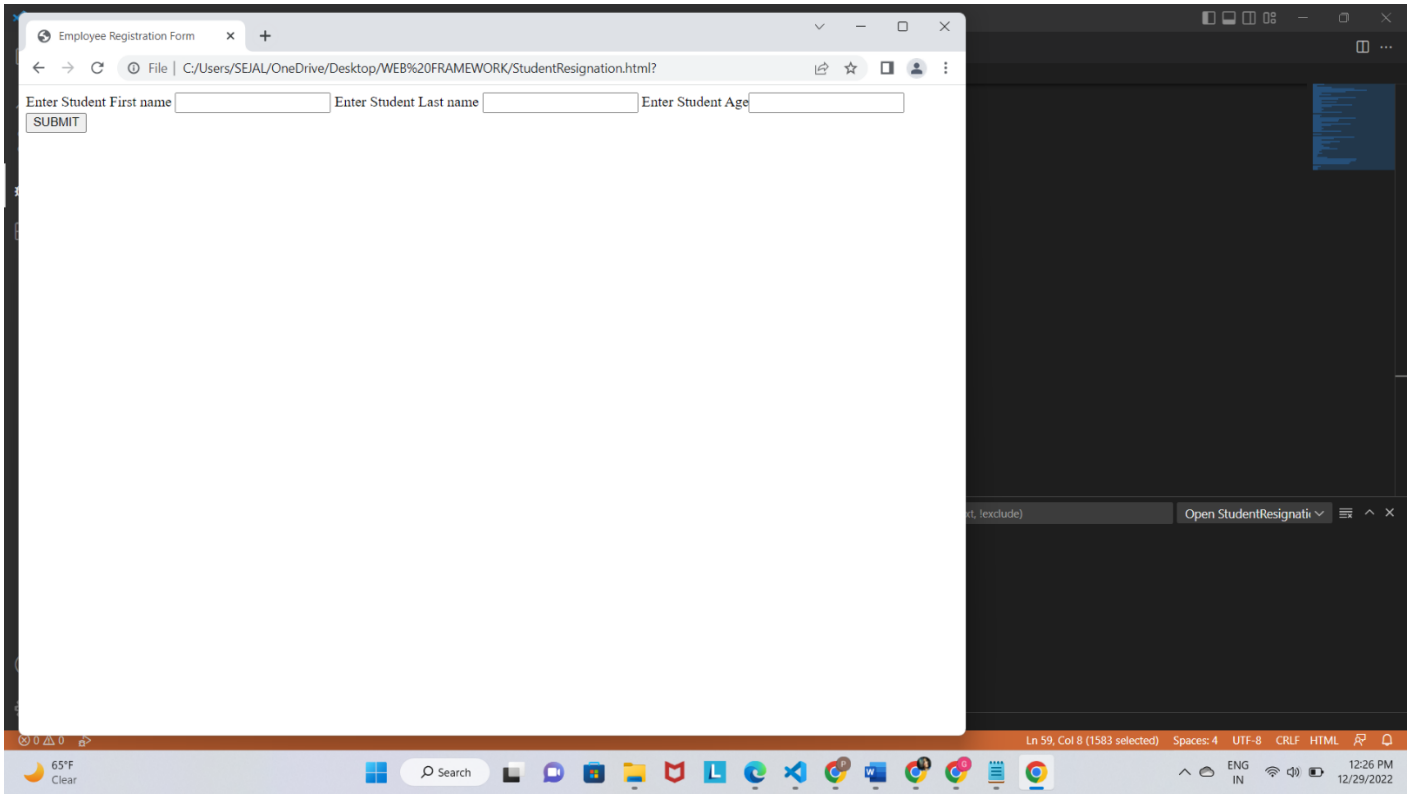
Q1) Create an HTML form that contain the Student Registration details and write a JavaScript to validate Student first and last name as it should not contain other than alphabets and age should be between 18 to 50.

```
<html lang="en">
<head>
   <title>Employee Registration Form</title>
   <script>
      function Submit()
      {
         var fname = document.getElementById("s1").value;
         var patt = /[A-Za-z]+$/;
         if(!fname.match(patt))
         {
          alert("First name should be alaphabatic");
           return false;
         }
         else
         {
          alert("First name Entered");


         }


         var lname = document.getElementById("s2").value;
         var patt2 = /[A-Za-z]+$/;
          if(!lname.match(patt2))
         {
```

```
            alert("Last name should be alphabatic");
            return false;
           }
           else
           {
            alert("Last name Entered");


           }


           var age = document.getElementById("s3").value;
           if(age<18||age>50)
           {
            alert("Enter valid age");
           return false;
           }
           else
           {
             alert("Age Entered");
           }
         }
    </script>
</head>
<body>
    <form action="">
        Enter Student First name <input type="text" id="s1">
        Enter Student Last name <input type="text" id="s2">
        Enter Student Age<input type="text" id="s3">
```

```
    <button onclick="Submit()">SUBMIT</button>

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

OUTPUT



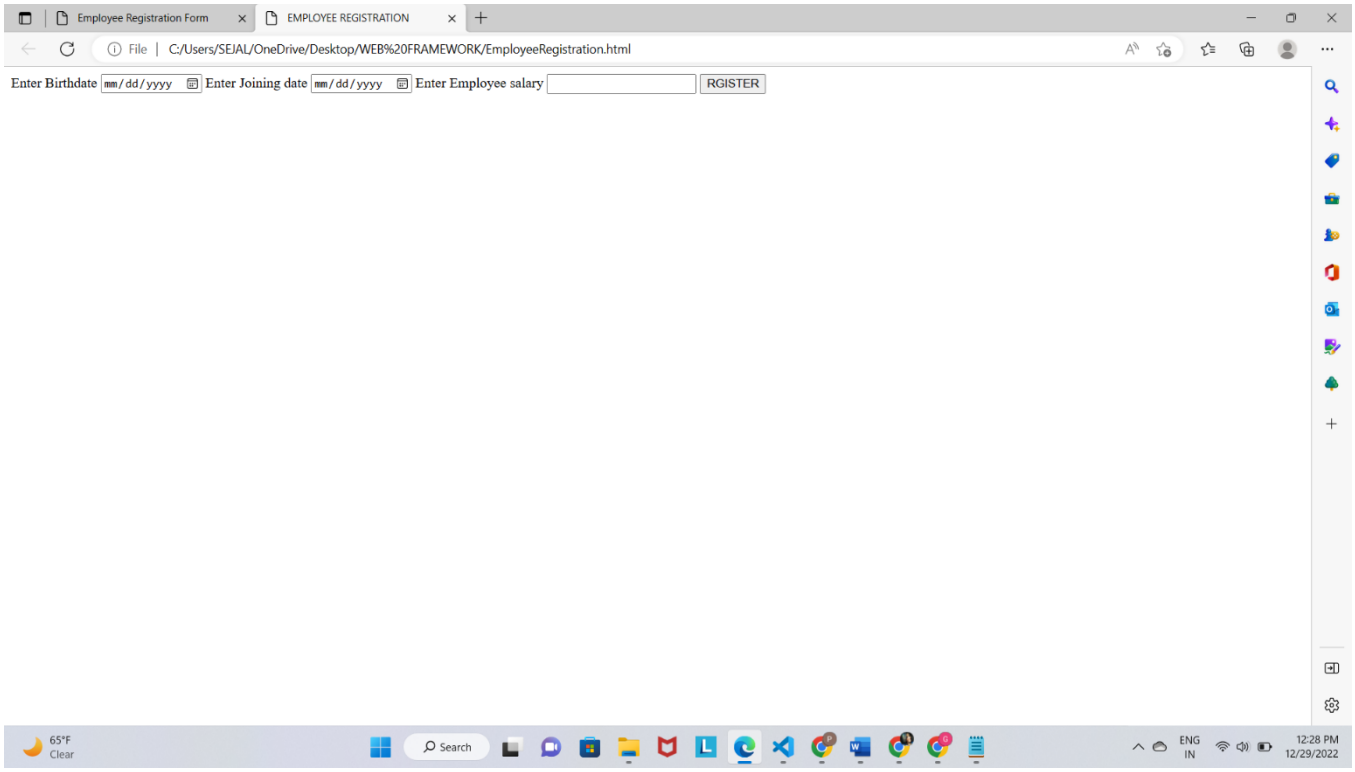
Q2) Create an HTML form that contain the Employee Registration details and write a JavaScript to validate DOB, Joining Date, and Salary.

```
<html lang="en">
<head>
    <title>EMPLOYEE REGISTRATION</title>
   <script>
      function register()
      {
         var bd = document.getElementById("d1").value;
         var bdlen = bd.length;
         if(bdlen==0)
         {
            alert("Please Enter BirthDate");
            return false;
         }
         else
         {
            alert("BirthDate Entered\n"+bd);
         }

         var jd = document.getElementById("d2").value;
         var jdlen = jd.length;
         if(jdlen==0)
         {
            alert("Please Enter Joining Date");
            return false;
```

```
        }
        else
        {
            alert("Joining Date Entered\n"+jd);
        }

        var sal = document.getElementById("d3").value;
        if(sal==0)
        {
            alert("Salary Must be entered");
            return false;
        }
        else
        {
            alert("Salary Entered\n"+sal);
        }


    }
  </script>
</head>
<body>
  <form action="">
  Enter Birthdate <input type="date" id="d1">
  Enter Joining date <input type="date" id="d2">
  Enter Employee salary <input type="number" id="d3">
  <button onclick="register()">RGISTER</button>
  </form>
```

```
</body>

</html>
```

OUTPUT

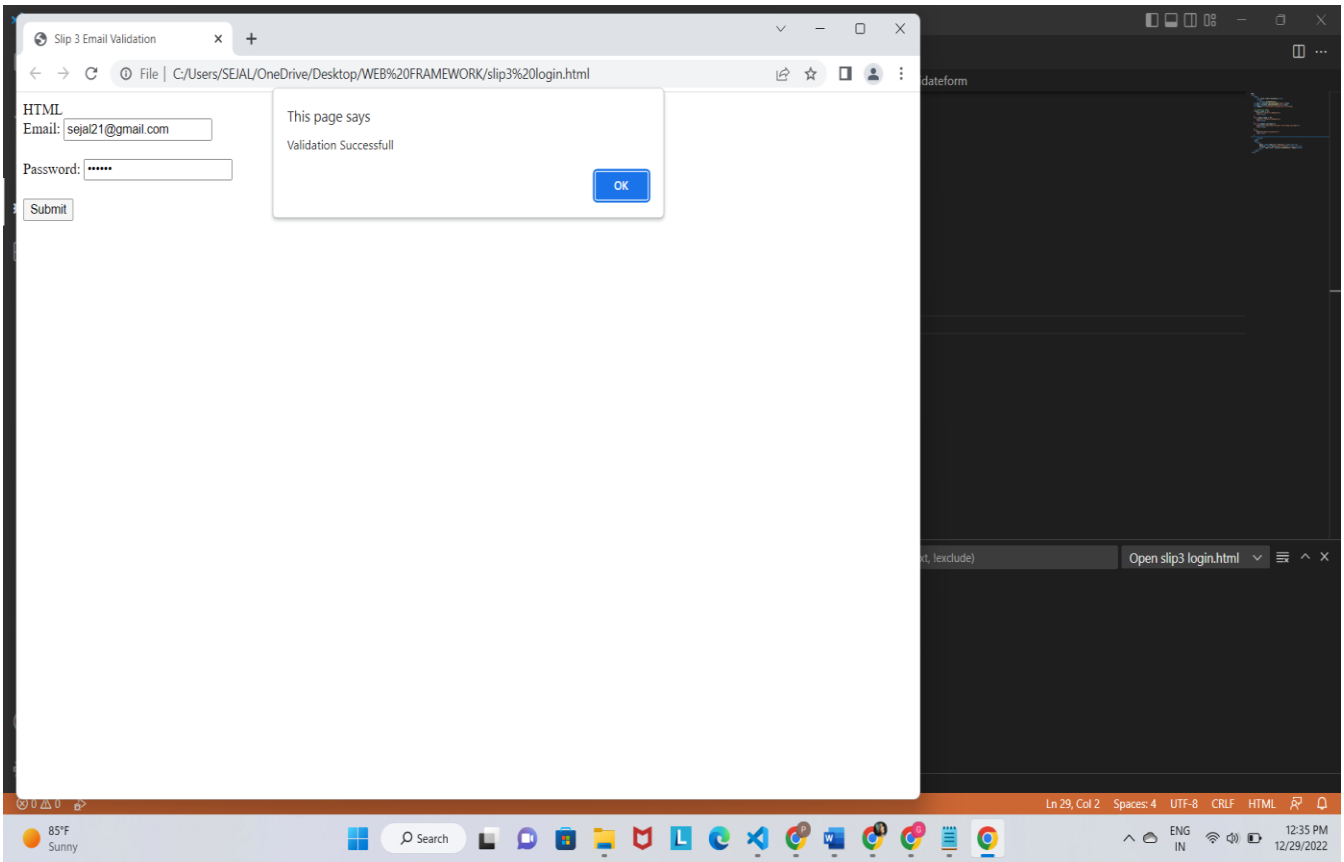


Q3) Create an HTML form for Login and write a JavaScript to validate email ID using Regular Expression.

```
<html>
  <head>
    <title>Slip 3 Email Validation</title>
    <script>
      function validateform(){
  var email = document.getElementById("email").value;
  var pass = document.getElementById("pass").value;
  var regEmail = /^([a-zA-Z0-9\._]+)@([a-z]+)(.[a-z]+)?$/;

  //if Fields are empty
  if(email.length == 0){
    alert("All Fields are Mandatory");
    return false;
  }
  else if(pass.length == 0){
    alert("All Fields are Mandatory");
    return false;
  }
  else if(!regEmail.test(email)){
    alert("Enter Name Format Correctly \"First Name Last Name\"");
    return false;
  }
  else{
    alert("Validation Successfull");
    return true;
```

```
    }
}
        </script>
    </head>
    <body>
        <form>
            Email: <input type="text" id="email"></input></br></br>
            Password: <input type="password" id="pass"></input></br></br>
            <button type="submit" onclick="validateform()">Submit</button>
        </form>
    </body>
</html>
```

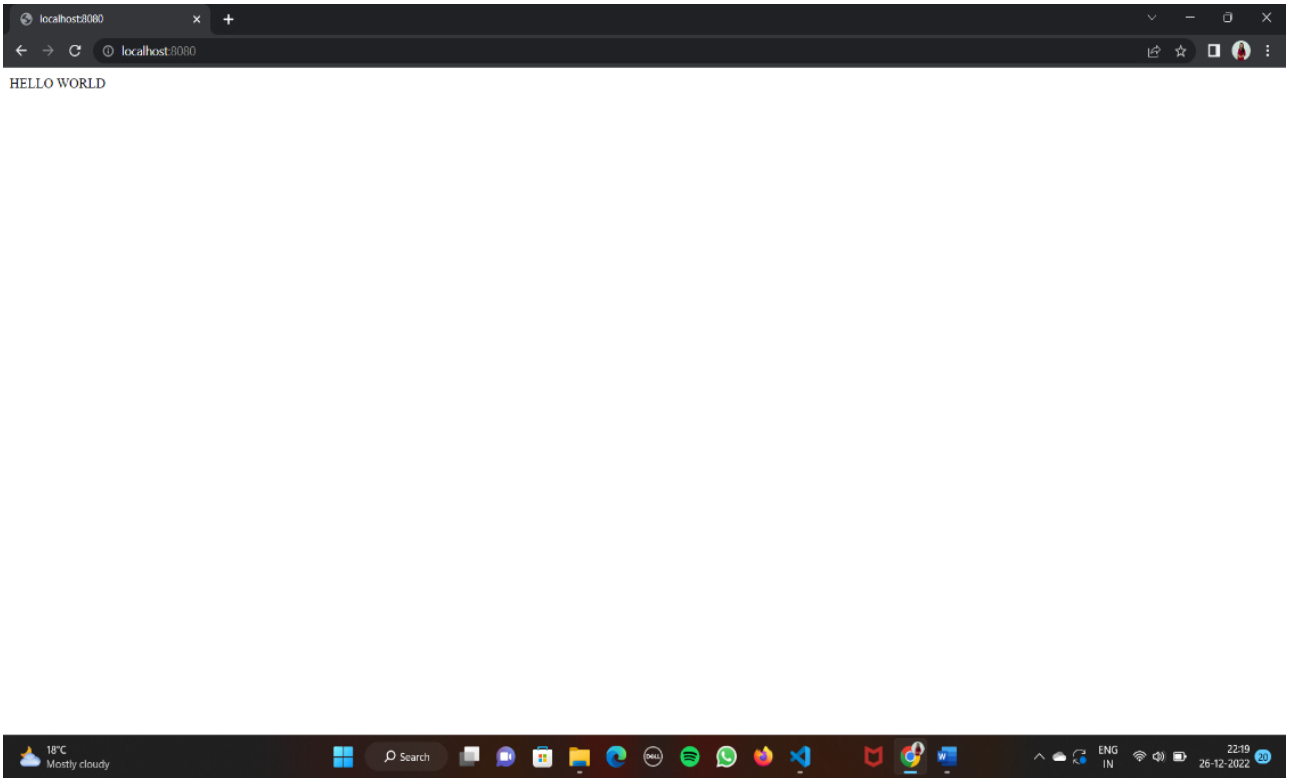
OUTPUT



Q4) Create a Node.js file that will convert the output "Hello World!" into upper-case letters.

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

var uc = require('upper-case');

http.createServer(function(req,res){

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

   res.write(uc.upperCase("Hello World...!"));

   res.end();

}).listen(8080);
```

OUTPUT



Q5) Using nodejs create a web page to read two file names from user and append contents of first file into second file.

```
var fs = require('fs');
var path = require('path')
var data = fs.readFileSync("Sample1.txt","utf8");
fs.appendFile("Sample2.txt", data,(err) =>{
    if(err) {
        console.log(err);
    }
    else{
        console.log("File Content after appending:
",fs.readFileSync("Sample2.txt","utf8"));
    }
});
```

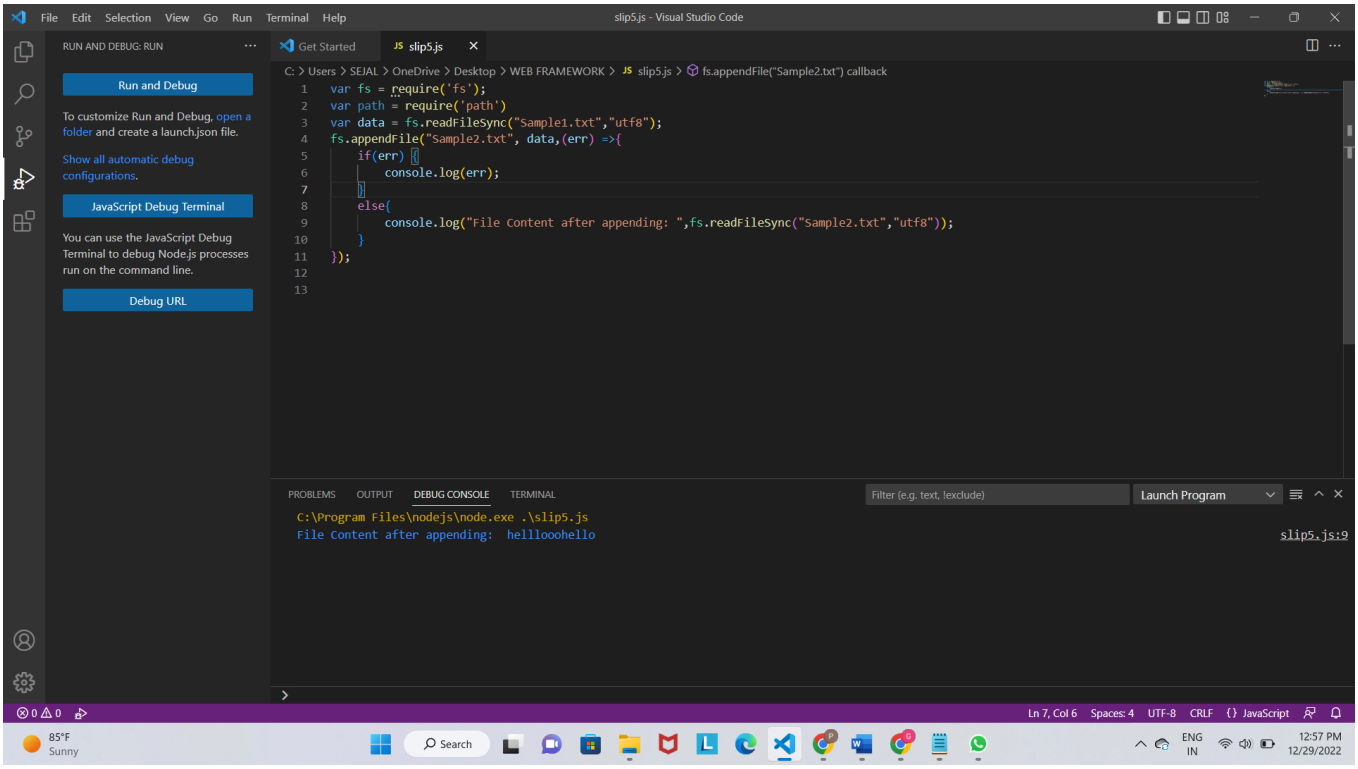
SAMPLE 1

```
hello
```

SAMPLE 2

```
hello
```

OUTPUT



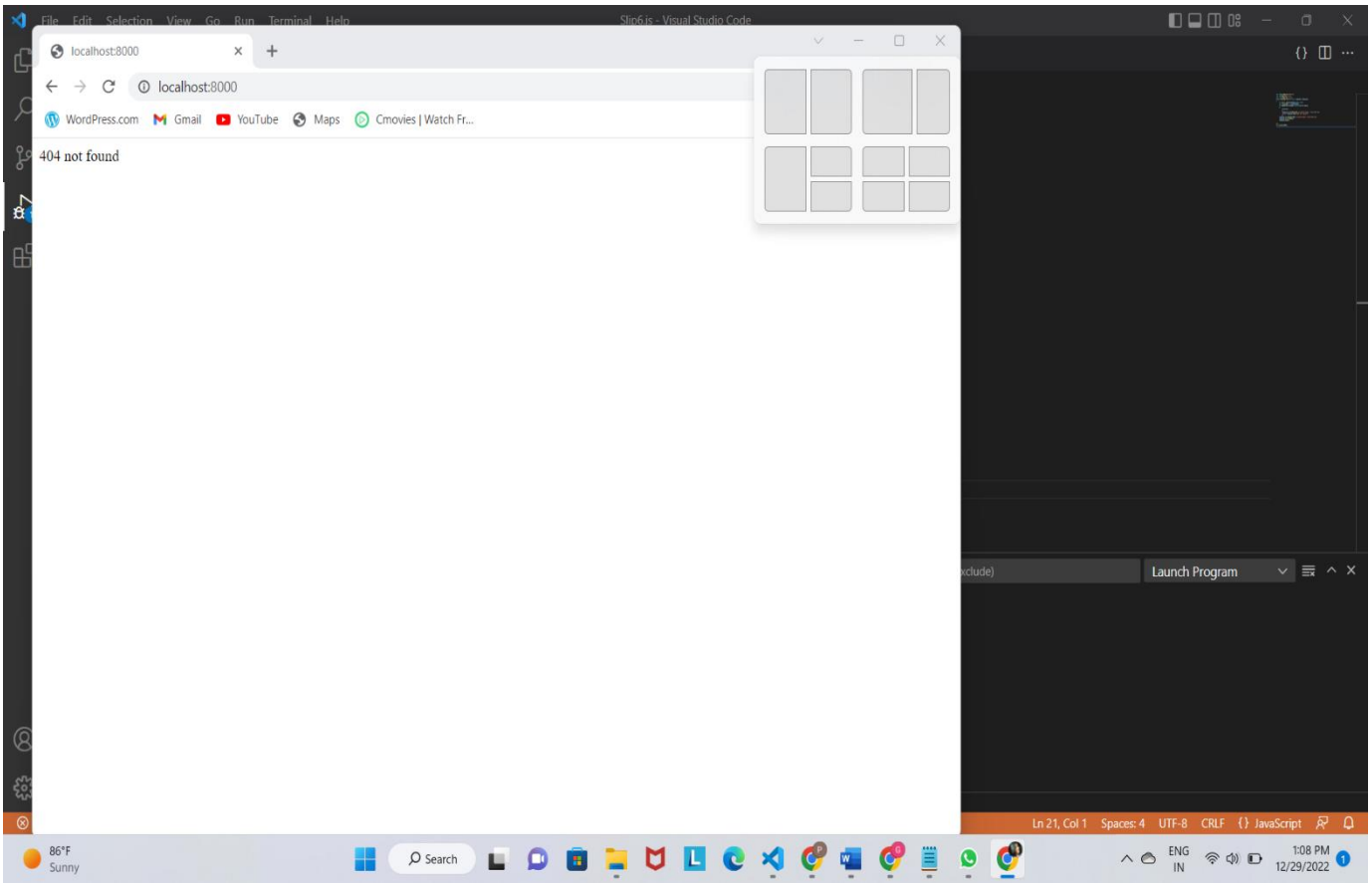
Q6) Create a Node.js file that opens the requested file and returns the content to the client. If anything goes wrong, throw a 404 error.

```
var http=require('http');
var fs=require('fs');
var url=require('url');
http.createServer(function (request,response)
{
    var q=url.parse(request.url,true);
    var filename="."+q.pathname;
    fs.readFile(filename,function(error,data)
    {
        if(error)
        {
        response.writeHead(404,{'content-type':'text/html'});
        return response.end("404 not found");
        }
    response.writeHead(200,{'content-type':'text/html'});
    response.write(data);
    response.end();

});
}).listen(8000);
```

SAMPLE 1

```
WHAT IS GOOGLE?
Google is a popular internet search engine. It scans the Web to
find Web pages that are relevant to the words you have typed in
the search box.
```

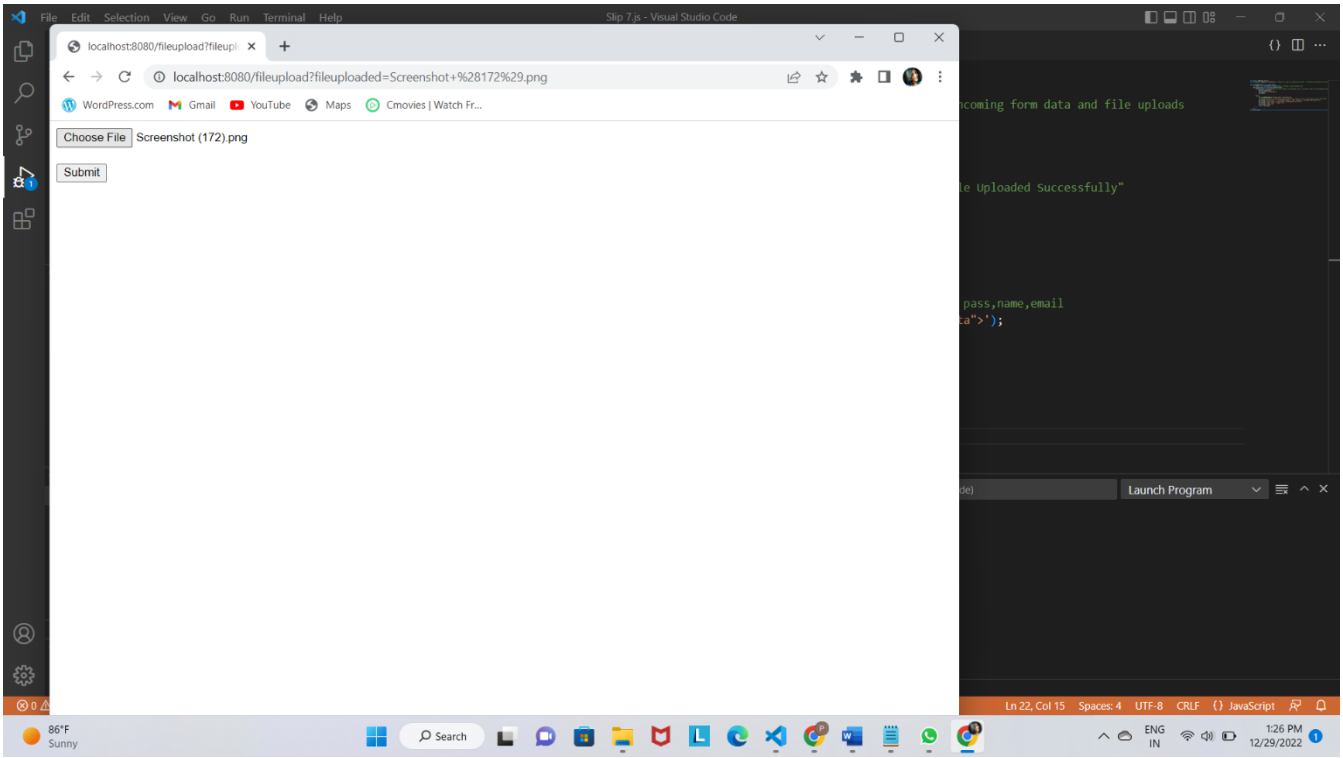
OUTPUT


File Edit Selection View Go Run Terminal Help
Slip6.js - Visual Studio Code
localhost:8000
localhost:8000
WordPress.com
Gmail
YouTube
Maps
Cmovies | Watch Fr...
404 not found
Launch Program
Ln 21, Col 1
Spaces: 4
UTF-8
CRLF
{} JavaScript
86°F
Sunny
Search
ENG
IN
1:08 PM
12/29/2022

Q7) Create a Node.js file that writes an HTML form, with an upload field.

```
var http = require('http');
var formidable = require('formidable'); //module is used for
parsing form data,  for handling incoming form data and file
uploads

http.createServer(function(req,res){
    var form = new formidable.IncomingForm(); //Creates a new
incoming form.
    form.parse(req,function(err,fields,files){
        if(req.url=='/fileupload'){ //if user request of
uploading file is successful then "File Uploaded Successfully"
            console.log(files);
            res.write('File Uploaded');
            res.end();
        }
        else{
            res.writeHead(200,{'Content-Type':'text/html'});
            //enctype multipart is used for dealing with files,
usually 'text' is the type for pass,name,email
            res.write('<form action = "fileupload" method =
"get" enctype = "multipart/form_data">');
            res.write('<input type = "file"
name="fileuploaded"><br><br>');
            res.write('<input type = "submit">');
            res.write('</form>');
            return res.end();
        }
    });
}).listen(8080);
```

OUTPUT


File Edit Selection View Go Run Terminal Help
Slip 7.js - Visual Studio Code
localhost:8080/fileupload?fileupl
localhost:8080/fileupload?fileuploaded=Screenshot+%28172%29.png
WordPress.com
Gmail
YouTube
Maps
Cmovies | Watch Fr...
Choose File
Screenshot (172).png
Submit
ncoming form data and file uploads
le Uploaded Successfully"
pass,name,email
Launch Program
Ln 22, Col 15
Spaces: 4
UTF-8
CRLF
JavaScript
86°F
Sunny
Search
ENG
IN
1:26 PM
12/29/2022

Q8) Create a Node.js file that demonstrates create database and table in MySQL.

DATABASE

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

var con = mysql.createConnection({
  host: "localhost",
  user: "yourusername",
  password: "yourpassword"
});

con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  con.query("CREATE DATABASE mydb", function (err, result) {
    if (err) throw err;
    console.log("Database created");
  });
});
```

TABLE

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

var con = mysql.createConnection({
  host: "localhost",
  user: "myusername",
  password: "mypassword",
  database: "mydb"
});

con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
```

```
  /*Create a table named "customers":*/
  var sql = "CREATE TABLE customers (name VARCHAR(255), address
VARCHAR(255))";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("Table created");
  });
});
```

Q9) 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: "yourusername",
  password: "yourpassword",
  database: "mydb"
});

con.connect(function(err) {
  if (err) throw err;
  con.query("SELECT * FROM customers", function (err, result, fields)
{
    if (err) throw err;
    console.log(result);
  });
});
```

Q10) Create a node.js file that Insert Multiple Records in "student" table, and display the result object on console.

insert_record.js

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

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);
  });
});
```

Q11) 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: "yourusername",
  password: "yourpassword",
  database: "mydb"
});

con.connect(function(err) {
  if (err) throw err;
  var sql = "DELETE FROM customers WHERE address = 'Mountain 21'";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("Number of records deleted: " + result.affectedRows);
  });
});
```

Q12) 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(5000);
```

Q14) Write node js script to interact with the filesystem, and serve a web page from a file .

```
<html>
<body>
<h1>My Header</h1>
<p>My paragraph.</p>
</body>
</html>

var http = require('http');
var fs = require('fs');
http.createServer(function (req, res) {
  fs.readFile('demofile1.html', function(err, data) {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write(data);
    return res.end();
  });
}).listen(8080);
```

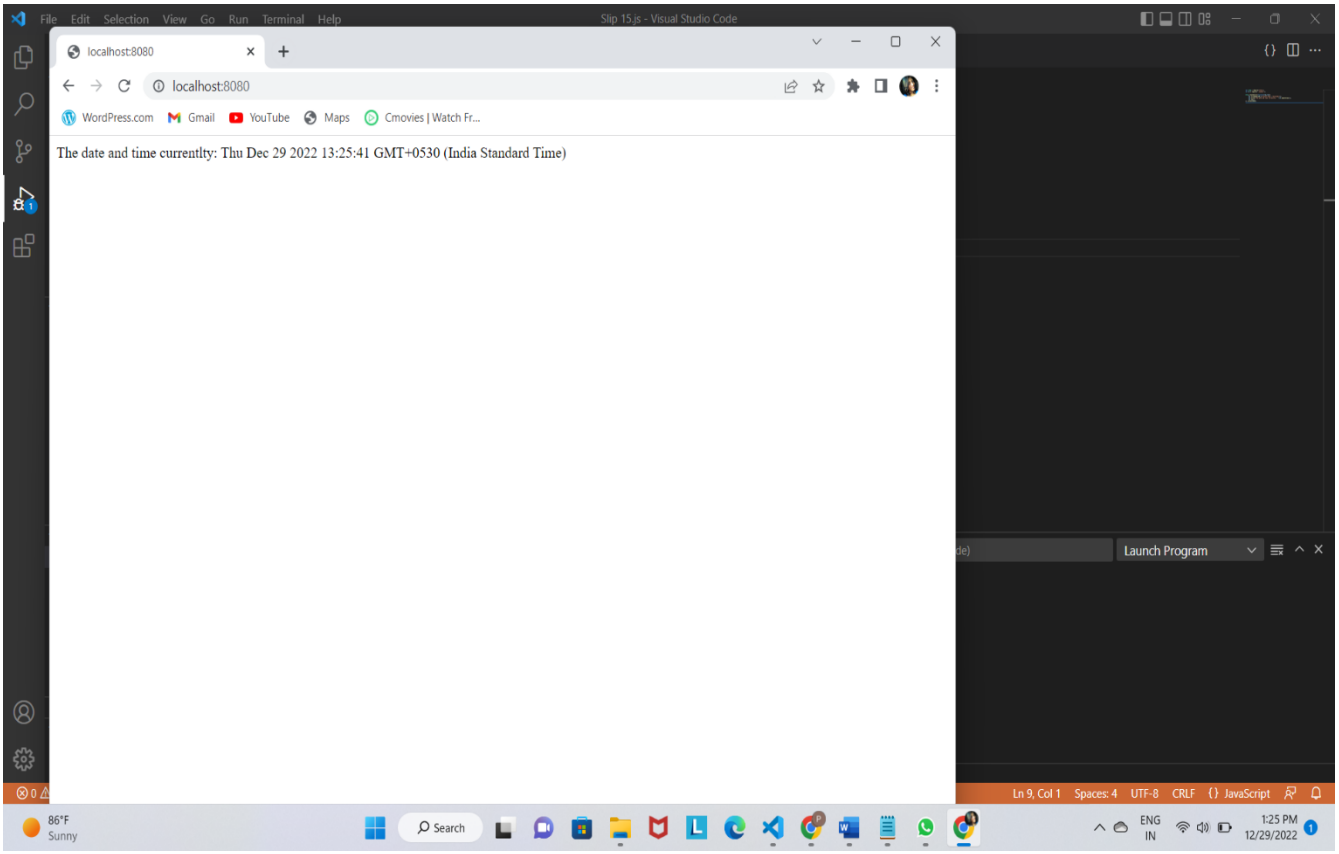
Q15) Write node js script to build Your Own Node.js Module. Use require ('http') module is a built-in Node module that invokes the functionality of the HTTP library to create a local server. Also use the export statement to make functions in your module available externally. Create a new text file to contain the functions in your module called, "modules.js" and add this function to return today's date and time.

```
var http = require('http');
var dt = require('./module');

http.createServer(function(req,res){
    res.writeHead(200,{'Content-Type':'text/html'});
    res.write('The date and time currentlty: ' +
dt.myDateTime());
    res.end();
}).listen(8080);
```

MODULE.JS

```
exports.myDateTime = function() {
    return Date();
}
```

OUTPUT


File Edit Selection View Go Run Terminal Help
Slip 15.js - Visual Studio Code
localhost:8080
localhost:8080
WordPress.com
Gmail
YouTube
Maps
Cmovies | Watch Fr...
The date and time currentlty: Thu Dec 29 2022 13:25:41 GMT+0530 (India Standard Time)
Launch Program
Ln 9, Col 1
Spaces: 4
UTF-8
CRLF
JavaScript
86°F
Sunny
Search
ENG
IN
1:25 PM
12/29/2022

Q16) 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.

```
// Import events module
var events = require('events');
// Create an eventEmitter object
var eventEmitter = new events.EventEmitter();

// Create an event handler as follows
var connectHandler = function connected() {
  console.log('connection succesful.');

  // Fire the data_received event
  eventEmitter.emit('data_received');
}

// Bind the connection event with the handler
eventEmitter.on('connection', connectHandler);
 // Bind the data_received event with the anonymous function
eventEmitter.on('data_received', function(){
  console.log('data received succesfully.');
});
// Fire the connection event
eventEmitter.emit('connection');
console.log("Program Ended.");
```

Q17) 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');
var app = express();
var PORT = 3000;

app.get('/', function(req, res){
    res.download('Unknown_file.txt', function(error){
        console.log("Error : ", error)
    });
});

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