**Q. 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>Student Registration</title>

 <script>

 function validate() {

 var Fname = document.getElementById("fname").value;

 let reF = /\d/;

 var Lname =document.getElementById("lname").value;

 let reL = /\d/;

 var Age = document.getElementById("age").value;

 if (reF.test(Fname)) {

 alert("Please use Alphabates to write first name.");

 return false;

 }

 else if ( reL.test(Lname)) {

 alert("Please use Alphabates to write last name.");

 return false;

 }

 else if(!(Age < 50 && Age > 18) )

 {

 alert('age should be between 18 to 50 ');

 return false;

 }

 alert("Registration Successful");

 return true;

 }

 </script>

</head>

<body bgcolor="yellow">

 <form onsubmit=validate()>

 <h1><b>Student Registration</b></h1>

 First Name:<input type="text" id="fname" /><br><br>

 Last Name:<input type="text" id="lname" /><br><br>

 Age:<input type="text" id="age" /><br><br>

 <input type="submit" >

 </form>

</body>

</html>

**Q. 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 Registration</title>

        <script>

            function validate(){

            const d=new Date();

            var DOB=document.getElementById("dob").value;

            let DOB1=new Date(DOB);

            var jd=document.getElementById("joining\_date").value;

            let jd1=new Date(jd);

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

            if(!(DOB1 < d)){

                alert("DOB should be less than current date");

                return false;

            }

            else if(!(jd1 >= d)){

                alert("jd should be greater than equal to current date");

                return false;

            }

            else if(!(salary >=10000)){

                alert("salary should be greater than 10000");

                return false;

            }

            alert("complete registration");

            return true;

            }

        </script>

    </head>

    <body bgcolor="pink">

        <form>

            <h1><b>Employee Registration</b></h1>

        name:<input type="text" id="name"/><br>

        dob:<input type="date" id="dob"/><br>

        joining date:<input type="date" id="joining\_date"/><br>

        salary:<input type="number" id="salary"/><br>

        <input type="submit" value="submit" onclick=validate()>

        </form>

    </body></html>

**Q. 3 Create an HTML form for Login and write a JavaScript to validate email ID**

**using Regular Expression.**

<html>

<head>

 <title> Login Form </title>

 <script>

 function validate() {

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

 var password = document.getElementById("pass").value;

 let re = new RegExp('^[a-zA-Z0-9.\_-]+@[a-zA-Z0-9.-]+\.[a-zAZ]{2,4}$')

 if (!re.test(username)) {

 alert("Please enter the username.");

 // return false;

 }

 if (password == null || password == "") {

 alert("Please enter the password.");

 // return false;

 }

 alert('Login successful');

 // return true;

 }

 </script>

</head>

<body bgcolor="sky blue">

 <form method="get"

onsubmit=validate()>

 username:<input type="text" id="username" /><br>

 Password:<input type="password" id="pass" /><br>

 <input type="submit" value="submit">

 </form>

</body>

</html>

**--------------------------------------------------------------------------------------------------------------------**

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

**letters.**

let output = "Hello World!"

console.log(output.toUpperCase())

**--------------------------------------------------------------------------------------------------------------------**

**Q. 5. Using nodejs create a web page to read two file names from user and append contents**

**of first file into second file.**

//npm install prompt-sync

var fs = require('fs');

const prompt = require("prompt-sync")({ sigint: true });

const firstFile = prompt("Enter Name Of First File : ");

const secondFile = prompt("Enter Name Of Second File : ");

// open destination file for appending

**var w = fs.createWriteStream(secondFile, {flags: 'a'});**

// open source file for reading

var r = fs.createReadStream(firstFile);

w.on('close', function() {

    console.log("done writing");

});

r.pipe(w);

**Q. 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 fs = require('fs');

const { error } = require('console');

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

            fs.open('input.txt', 'r+', function(err, fd) {

                if (err) {

                    err.status = 404;

                    console.error(err);

                    return res.end('File Not Found');

                } else {

                    console.log("File opened succefully");

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

                        if (!err) {

                            console.log('success ');

                            fs.close(fd);

                            return res.end(data);

                        }

                        else {

                            err.status = 404;

                            console.error(err);

                            return res.end('File Not Found');

                        }

                    });

                }

           });

    });

            server.listen(5000,()=>{

console.log(“server running at port 5000”)

});

**-------------------------------------------------------------------------------------------------------------------------------**

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

const http = require('http')

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

    res.setHeader("Content-Type" , "text/html")

    res.statusCode = 200

    res.write(`

    <html>

    <body>

    <input type="file" name="fileupload">

    <br>

    <input type="submit">

    </form>

    </body>

    </html>

    `)

    return res.end()

}).listen(2000 , ()=>{

    console.log("Server running at the port 2000")

})

**---------------------------------------------------------------------------------**

**Q. 8 Create a Node.js file that demonstrates create database and table in MySQL.**

// create database:

const mysql = require('mysql')

const conn = mysql.createConnection({

    host:3306,

    user:"root",

    password:"root12"

})

conn.connect((err)=>{

    if(err){

        console.log(err)

    }else {

        console.log("Connected")

        conn.query("CREATE DATABASE testdb", function(err, result){

            if(err){

                console.log(err)

            }else {

                console.log("created database")

                conn.end()

            }

        })

    }

})

// to create table

const mysql = require('mysql')

const conn = mysql.createConnection({

    host:3306,

    user:"root",

    password:"root12",

    database:"testdb"

})

conn.connect((err)=>{

    if(err){

        console.log(err)

    }else {

        const createTableQuery = "create table student(name varchar(20) , class varchar(20))"

        conn.query(createTableQuery , (err, result) =>{

            if(err){

                console.log(err)

            }else {

                console.log("Table Created ")

                conn.end()

            }

        })

    }

})

**-------------------------------------------------------------------------------------------------------------------------------------------------**

**Q. 9 Create a node.js file that Select all records from the "customers" table, and display the**

**result object on console.**

const mysql = require('mysql')

const conn = mysql.createConnection({

    port:3306,

    user:"root",

    password:"root12",

    database:"testdb"

})

conn.connect((err)=>{

    if(err){

        console.log(err)

    }else {

        const q = "select \* from customers2"

        conn.query(q , (err, result)=>{

            if(err){

                console.log(err)

            }else {

                result.forEach((e)=>{

                    console.log(`name : ${e.name} address : ${e.address}`)

                })

            }

            conn.end()

        })

    }

})

**--------------------------------------------------------------------------------------------------------------------------------------**

**Q. 10 Create a node.js file that Insert Multiple Records in "student" table, and display the**

**result object on console.**

const mysql = require('mysql')

const conn = mysql.createConnection({

    port:3306,

    user:"root",

    password:"root12",

    database:"testdb"

})

conn.connect((err)=>{

    if(err){

        console.log(err)

    }else {

        values = [

            ["ashwini", "mcs"],

            ["diksha", "datascience"],

            ["neha" , "mcs"]

        ]

        conn.query('insert into student(name , class) values?',[values], (err, result)=>{

            if(err){

                console.log(err)

            }else{

                console.log("inserted successfully")

                console.log(result)

                conn.end()

            }

        })

    }

})

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

**specified record.**

//npm install mysql2

var mysql = require('mysql');

var con = mysql.createConnection({

    host: "localhost",

    port: 3306,

    user: "root",

    password: "root12",

    database: "mydb"

});

con.connect(function (err) {

    if (err) throw err;

    console.log("Connected!");

    con.query("SELECT \* FROM customers", function (err, result, fields) {

        if (err) throw err;

        console.log(result);

    });

    con.query("delete FROM customers where name = 'sakshi'", function (err, result, fields) {

        if (err) throw err;

        console.log(result);

    });

});

**--------------------------------------------------------------------------------------------------------------------------**

**Q. 12 Create a Simple Web Server using node js**

let http = require('http')

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

    res.end("Hello World !")

})

server.listen(4000,()=>{

console.log(“server running at 4000”)

});

**-----------------------------------------------------------------------------------------------------------------------------------------**

**Q. 13 Using node js create a User Login System.**

**login.js**

let express=require('express')

let bodyParser=require('body-parser')

let app=express()

app.use(bodyParser.urlencoded({extended:true}))

app.get("/",(req,res)=>{

    res.sendFile(\_\_dirname+"/login.html")

})

app.post("/login",(req,res)=>{

    const CorrectUser="admin"

    const CorrectPassword="admin"

    const {fname,password}=req.body

    if(CorrectUser==fname && CorrectPassword==password){

        res.sendFile(\_\_dirname+"/user.html")

    }else{

        res.sendFile(\_\_dirname+"/error.html")

    }

})

app.listen(3333,()=>{

    console.log("server listen to the port 3333")

})

**Login.html**

<!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>login html </title>

</head>

<body>

    <form method="post" action="/login">

      Enter username:  <input type="text" name="fname" ><br><br>

        Enter Password:<input type="text" name="password" >

        <button id="btn">login</button>

    </form>

</body>

</html>

**user.html**

<!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>

</head>

<body>

    login successfully!

</body>

</html>

**error.html**

<!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>

</head>

<body>

    invalid login

</body>

</html>

**------------------------------------------------------------------------------------------------------------------------------------------**

**Q. 14 Write node js script to interact with the filesystem, and serve a web page from a file**

let fs = require('fs/promises')

let http = require('http')

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

   let data = await fs.readFile('file.html')

   data = data.toString()

   res.setHeader('Content-Type', 'text/html')

   res.write(data)

   res.end()

})

server.listen(2000)

**file.html**

<!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>

</head>

<body>

    <h1>Hello</h1>

    <p>From html file</p>

</body>

</html>

**Q. 15 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.**

const http = require('http')

const getDate = require('./modules')

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

    res.end(getDate())

})

server.listen(2000)

**modules.js**

module.exports = function getDate(){

    return new Date().toLocaleString()

}

**Q. 16. 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.**

**main.js**

var events = require('events');

var eventEmitter = new events.EventEmitter();

//Create an event handler:

var myEventHandler = function () {

    console.log("Inside the callback function");

  console.log('I hear a event!');

}

//Assign the event handler to an event:

eventEmitter.on('fire', myEventHandler);

function main() {

//Fire the 'scream' event:

    setInterval(()=>{

        console.log("Firing an event ")

        eventEmitter.emit('fire')

    } , 2000)

}

main()

**--------------------------------------------------------------------------------------------------------------------**

**Q. 17 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.**

const express = require('express')

const app = express()

app.get("/" , (req , res)=>{

    res.download("file.txt")

})

app.listen(2000)

**file.txt**

**this is a file which has to be downloaded**

**-------------------------------------------------------------------------------------------------------------------------------**

**Q. 18 Create your Django app in which after running the server, you should see on the**

**browser, the text “Hello! I am learning Django”, which you defined in the index view.**

**views.py**

from django.http import HttpResponse

def hello(request):

    return HttpResponse("Hello")

**urls.py**

from django.contrib import admin

from django.urls import path

from.import views

urlpatterns = [

    path('admin/', admin.site.urls),

    path('hello/',views.hello),

]

**------------------------------------------------------------------------------------------------------------------------------------------**

**Q. 19 Design a Django application that adds web pages with views and templates.**

**views.py**

from django.http import HttpResponse

from django.template import loader

def slip(request):

    template=loader.get\_template("slip19.html")

    return HttpResponse(template.render())

**urls.py**

from django.contrib import admin

from django.urls import path

from.import views

urlpatterns = [

    path('admin/', admin.site.urls),

    path('slip/', views.slip),

]

**Setting.py**

Line no-14 import os

Line no-57 'DIRS': [os.path.join(BASE\_DIR, 'templates')],

**Create templates folder > inside it create “slip19.html” file>**

**Slip19.html**

<!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>Slip19</title>

</head>

<body>

    <h1>This is Slip19</h1>

    <p>This is from Slip19</p>

</body>

</html>

**-------------------------------------------------------------------------------------------------------------------------------**