

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.

```
//student.js

function formValidation()
{
    var first=document.registration.first;
    var last=document.registration.last;
    var rollno=document.registration.rollno;
    var uadd =document.registration.address;
    var age=document.registration.age;
    if(allLetter(first))
    {
        if(allLetter(last))
        {
            if(alphanumeric(rollno))
            {
                if(allnumeric(age))
                {
                    }}}}
    return false;
}

function allLetter(first)
{
    var letters = /^[A-Za-z]+$;/
    if(first.value.match(letters))
    {
        alert('Student first name submitted');
        return true;
    }
}
```

```
}  
else  
{  
    alert('Student first name must have alphabet characters only');  
    first.focus();  
    return false;  
}  
}  
  
function allLetter(last)  
{  
    var letters = /^[A-Za-z]+$/;  
    if(last.value.match(letters))  
    {  
        alert("Student last name submitted");  
        return true;  
    }  
    else  
    {  
        alert('Student last name must have alphabet characters only');  
        last.focus();  
        return false;  
    }  
}  
  
function alphanumeric(rollno)  
{  
    var letters = /^[0-9]+$/;  
    if(rollno.value.match(letters))  
    {  
        alert("Student Rollno submitted");
```

```
return true;
}
else
{
    alert('Student Rollno must have alphanumeric characters only');
    uadd.focus();
    return false;
}
}

function allNumeric(age)
{
    var number = /^[0-9]+$/;
    if(age.value.match(number))
    {
        var input = age.value;
        if(input>=18&&input<=25) {
            return true;
        }
        else {
            alert("Age must be between 18 and 25 | You have entered "+input);
            return false;
        }
    }
}
else
{
    alert('Age must have numeric numbers only');
    no.focus();
    return false;
}
```

```
}
```

```
//student.html
```

```
<!DOCTYPE html>
```

```
<html lang="en"><head>
```

```
<meta charset="utf-8">
```

```
<title>Student Registration</title>
```

```
<meta name="keywords" content="Student Registration" />
```

```
<meta name="description" content="this is student registration " />
```

```
<link rel='stylesheet' href='student.css' type='text/css' />
```

```
<script src="student.js">
```

```
</script>
```

```
</head>
```

```
<body onload="document.registration.userid.focus();" bgcolor="orange">
```

```
<h1>Studednt Registration Form</h1>
```

```
<form name='registration' onSubmit="return formValidation();">
```

```
<ul>
```

```
<li><label for="first">First Name:</label></li>
```

```
<li><input type="text" name="first" size="50" /></li>
```

```
<li><label for="last">Last Name:</label></li>
```

```
<li><input type="text" name="last" size="50" /></li>
```

```
<li><label for="empid">RollNO:</label></li>
```

```
<li><input type="text" name="rollno" size="50" /></li>
```

```
<li><label for="no">Age:</label></li>
```

```
<li><input type="number" name="age"></li>
```

```
<li><input type="submit" name="submit" value="Submit" /></li>
```

```
</ul>
```

```
</form>
```

```
</body>
```

```
</html>
```

```
//student.css
```

```
h1 {
```

```
    margin-left: 70px;
```

```
}
```

```
form li {
```

```
    list-style: none;
```

```
    margin-bottom: 5px;
```

```
}
```

```
form ul li label{
```

```
    float: left;
```

```
    clear: left;
```

```
    width: 100px;
```

```
    text-align: right;
```

```
    margin-right: 10px;
```

```
    font-family: Verdana, Arial, Helvetica, sans-serif;
```

```
    font-size: 14px;
```

```
}
```

```
form ul li input, select, span {
```

```
    float: left;
```

```
    margin-bottom: 10px;
```

```
}
```

```
form textarea {
```

```
    float: left;
```

```
    width: 350px;
```

```
height: 150px;  
}
```

```
[type="submit"] {  
clear: left;  
margin: 20px 0 0 230px;  
font-size:18px  
}
```

```
p {  
margin-left: 70px;  
font-weight: bold;  
}
```

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

```
//emp.js  
  
function formValidation()  
{  
var first=document.registration.first;  
var last=document.registration.last;  
var empid=document.registration.empid;  
var birth=document.registration.birth;  
var uadd =document.registration.address;  
var ucountry =document.registration.country;  
var no=document.registration.no;  
var jdate=document.registration.jdate;  
var uemail = document.registration.email;
```

```

var umgen = document.registration.umgen;
var ufgn = document.registration.ufgn;
var salary =document.registration.salary;
if(allLetter(first))
{
if(allLetter(last))
{
if(alphanumeric(empid))
{
if(allb(birth))
{
if(alphanumeric(uadd))
{
if(countryselect(ucountry))
{
if(allnumeric(no))
{
if(allnumeric(jdate))
{
if(ValidateEmail(uemail))
{
if(validgendor(umgen,ufgn))
{
if(allnumeric(salary))
{
}}}}}}}}}}
return false;
}
function allLetter(first)

```

```
{
var letters = /^[A-Za-z]+$/;
if(first.value.match(letters))
{
alert('employee name submitted');
return true;
}
else
{
alert('employee name must have alphabet characters only');
first.focus();
return false;
}
}
function allLetter(last)
{
var letters = /^[A-Za-z]+$/;
if(last.value.match(letters))
{
    alert("employee name submitted");
return true;
}
else
{
alert('employee name must have alphabet characters only');
last.focus();
return false;
}
}
```



```

function alphanumeric(empid)
{
var letters = /^[0-9a-zA-Z]+$/;
if(empid.value.match(letters))
{
    alert("employee id submitted");
return true;
}
else
{
alert('employee id must have alphanumeric characters only');
uadd.focus();
return false;
}
}

function allb(birth)
{
var birth_len = birth.value.length;
if (birth_len == 0)
{
alert("birth date should not be empty");
birth.focus();
return false;
}
alert("birth of date submitted");
return true;
}

function alphanumeric(uadd)
{

```

```
var letters = /^[0-9a-zA-Z]+$/;
if(uadd.value.match(letters))
{
    alert("address submitted");
return true;
}
else
{
    alert('address must have alphanumeric characters only');
    uadd.focus();
return false;
}
}

function countryselect(ucountry)
{
    if(ucountry.value == "Default")
    {
        alert('Select your country from the list');
        ucountry.focus();
        return false;
    }
    else
    {
        alert("country submitted");
        return true;
    }
}

function allnumeric(no)
{

```

```

var number = /^[0-9]+$/;
if(no.value.match(number))
{
    alert("Contact Number submitted");
    return true;
}
else
{
    alert('Contact no must have numeric numbers only');
    no.focus();
    return false;
}
}

function allNumeric(jdate)
{
    var jdate_len = jdate.value.length;
    if (jdate_len == 0)
    {
        alert("date of joining should not be empty");
        birthday.focus();
        return false;
    }
    alert("date of joining submitted");
    return true;
}

function ValidateEmail(uemail)
{
    var mailformat = /^[w+([\.-]?\w+)*@[w+([\.-]?\w+)*(\.w{2,3})+$/;
    if(uemail.value.match(mailformat))

```

```
{  
    alert("email address is submitted");  
    return true;  
}  
  
else  
{  
    alert("You have entered an invalid email address!");  
    uemail.focus();  
    return false;  
}  
}  
  
function validgender(umgen,ufgen)  
{  
    x=0;  
    if(umgen.checked)  
    {  
        x++;  
    } if(ufgen.checked)  
    {  
        x++;  
    }  
    if(x==0)  
    {  
        alert('Select Male/Female');  
        umgen.focus();  
        return false;  
    }  
    else  
    {
```

```
window.location.reload()

return true;

}

}

function allnumeric(salary)
{
var sal = /^[0-9]+$;/
if(salary.value.match(sal))
{
alert("salary submitted");
return true;
}
else
{
alert('salry is not submitted');
salary.focus();
return false;
}
}
```

```
//emp.html
```

```
<html>

<html lang="en"><head>

<meta charset="utf-8">

<title>JavaScript Form Validation using a sample Employee registration form</title>

<meta name="keywords" content="example, JavaScript Form Validation, Sample registration form" />

<meta name="description" content="This document is an example of JavaScript Form Validation using a sample registration form. " />

<link rel='stylesheet' href='employee.css' type='text/css' />
```

```
<script src="emp.js">
</script>
</head>
<body onload="document.registration.userid.focus();" bgcolor="orange">
<h1>Employee Registration Form</h1>
<form name='registration' onSubmit="return formValidation();">
<ul>
<li><label for="first">First Name:</label></li>
<li><input type="text" name="first" size="50" /></li>
<li><label for="last">Last Name:</label></li>
<li><input type="text" name="last" size="50" /></li>
<li><label for="empid">Employee id:</label></li>
<li><input type="text" name="empid" size="50" /></li>
<li><label for="birth">Birth of date:</label></li>
<li><input type="date" id="birth" name="birth"></li>
<li><label for="address">Address:</label></li>
<li><input type="text" name="address" size="50" /></li>
<li><label for="country">Country:</label></li>
<li><select name="country">
<option selected="" value="Default">(Please select a country)</option>
<option value="AF">Australia</option>
<option value="AL">Canada</option>
<option value="DZ">India</option>
<option value="AS">Russia</option>
<option value="AD">USA</option>
</select></li>
<li><label for="no">Contact no:</label></li>
<li><input type="number" id="" name="no"></li>
<li><label for="jdate">Date of joining:</label></li>
```

```
<li><input type="date" id="" name="jdate"></li>
<li><label for="email">Email:</label></li>
<li><input type="text" name="email" size="50" /></li>
<li><label id="gender">Gender:</label></li>
<li><input type="radio" name="male" value="Male" /><span>Male</span></li>
<li><input type="radio" name="female" value="Female" /><span>Female</span></li>
<li><label for="salary">salary:</label></li>
<li><input type="number" id="salary" name="salary"></li>
<li><input type="submit" name="submit" value="Submit" /></li>
</ul>
</form>
</body>
</html>
```

```
//emp.css
```

```
h1 {
    margin-left: 70px;
}
form li {
    list-style: none;
    margin-bottom: 5px;
}
```

```
form ul li label{
    float: left;
    clear: left;
    width: 100px;
    text-align: right;
    margin-right: 10px;
```

```
font-family:Verdana, Arial, Helvetica, sans-serif;  
font-size:14px;  
}
```

```
form ul li input, select, span {  
float: left;  
margin-bottom: 10px;  
}
```

```
form textarea {  
float: left;  
width: 350px;  
height: 150px;  
}
```

```
[type="submit"] {  
clear: left;  
margin: 20px 0 0 230px;  
font-size:18px  
}
```

```
p {  
margin-left: 70px;  
font-weight: bold;  
}
```

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

//email.js

```
function ValidateEmail(inputText)
```



```

{
var mailformat = /^[a-zA-Z0-9.!#$%&'*/+=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/;
if(inputText.value.match(mailformat))
{
alert("Valid email address!");
document.form1.text1.focus();
return true;
}
else
{
alert("You have entered an invalid email address!");
document.form1.text1.focus();
return false;
}
}

```

//email.html

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>JavaScript form validation - checking email</title>
<link rel='stylesheet' href='email.css' type='text/css' />
</head>
<body onload='document.form1.text1.focus()>
<div class="mail">
<h2>Input an email and Submit</h2>
<form name="form1" action="#">

```

```
<ul>
<li><input type='text' name='text1' /></li>
<li>&nbsp;</li>
<li class="submit"><input type="submit" name="submit" value="Submit"
onclick="ValidateEmail(document.form1.text1)" /></li>
<li>&nbsp;</li>
</ul>
</form>
</div>
<script src="email.js"></script>
</body>
</html>
```

```
//email.css
li {list-style-type: none;
font-size: 16pt;
}
.mail {
margin: auto;
padding-top: 10px;
padding-bottom: 10px;
width: 400px;
background : #D8F1F8;
border: 1px solid silver;
}
.mail h2 {
margin-left: 38px;
}
input {
```

```
font-size: 20pt;
}
input:focus, textarea:focus{
background-color: lightyellow;
}
input submit {
font-size: 12pt;
}
.rq {
color: #FF0000;
font-size: 10pt;
}
```

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

```
//upp.js
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);
```

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

```
//append.js
var fs= require('fs');
fs.readFile('demo.txt',function(err,data)
{
```

```
fs.appendFile('demo2.txt',data.toString(),function(err)
{

});

fs.readFile('demo2.txt',function(err,data)
{
    console.log(data.toString())
});
});
```

//demo.txt

Hello

//demo2.txt

World

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 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 Page not found");
        }
        res.writeHead(200,{ 'Content-Type': 'text/html' });
```

```

        res.write(data);
        return res.end();
    });
}).listen(8080);

```

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

```

var http = require('http');
var formidable = require('formidable');
var fs = require('fs');
http.createServer(function (req, res)
{
    if(req.url == '/fileupload')
    {
        var form = new formidable.IncomingForm();
        form.parse(req, function(err,fields,files)
        {
            var oldpath = files.fileupload.path;
            var newpath = 'C:/Users/Saroj/'+files.fileupload.name;
            fs.rename(oldpath, newpath, function (err)
            {
                if(err) throw err;
                res.write('File uploaded and moved!');
                res.end();
            });
        });
    }
    else{
        res.writeHead(200,{ 'Content-Type': 'text/html' });
        res.write('<form action="fileupload" method="post" enctype="multipart/form-data">');
        res.write('<input type= "file" name= "fileupload"><br>');
        res.write('<input type= "submit">');
        res.write('</form>');
        return res.end();
    }
}).listen(8080);

```

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: "admin",
  password: "root@123"
});

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

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: "admin",
  password: "root@123",
  database: "mydatabase"
});

con.connect(function(err) {
  if (err) throw err;
  con.query("CREATE TABLE customer(cid int(10) primary key, cname varchar(50), cadd varchar(60));", function (err, result, fields) {
    if (err) throw err;
    console.log("Table created!!!");
  });
});
```

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: "admin",
  password: "root@123",
  database: "mydatabase"
});

con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!!");
  var sql = "INSERT INTO student VALUES (11,'Reema','Dange Chowk'),(12,'Tina','wakad'),(13,'Rina','pimpri)";
  con.query(sql, function (err, result, fields) {
    if (err) throw err;
    console.log("3 record inserted");
  });
});
```

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: "admin",
  password: "root@123",
  database: "mydatabase"
});

con.connect(function(err) {
  if (err) throw err;
  var sql = "DELETE FROM customer WHERE cid=3";
  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'); // Import Node.js core module

var server = http.createServer(function (req, res) { //create web
server
    if (req.url == '/') { //check the URL of the current request

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

        // set response content
        res.write('<html><body><p>This                is                home
Page.</p></body></html>');
        res.end();

    }
    else if (req.url == "/student") {

        res.writeHead(200, { 'Content-Type': 'text/html' });
        res.write('<html><body><p>This                is                student
Page.</p></body></html>');
        res.end();

    }
    else if (req.url == "/admin") {

        res.writeHead(200, { 'Content-Type': 'text/html' });
        res.write('<html><body><p>This                is                admin
Page.</p></body></html>');
        res.end();

    }
    else
        res.end('Invalid Request!');

});

server.listen(5000); //6 - listen for any incoming requests

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


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 EventEmitter = new events.EventEmitter();

// listener #1
var listener1 = function listener1() {
    console.log('listener1 executed.');
```

```
}

// listener #2
var listener2 = function listener2() {
    console.log('listener2 executed.');
```

```
}

// Bind the connection event with the listener1 function
eventEmitter.addListener('connection', listener1);

// Bind the connection event with the listener2 function
eventEmitter.on('connection', listener2);

var eventListeners = require('events').EventEmitter.listenerCount
    (eventEmitter, 'connection');
console.log(eventListeners + " Listener(s) listening to connection event");

// Fire the connection event
eventEmitter.emit('connection');

// Remove the binding of listener1 function
eventEmitter.removeListener('connection', listener1);
console.log("Listener1 will not listen now.");

// Fire the connection event
eventEmitter.emit('connection');

eventListeners = require('events').EventEmitter.listenerCount(eventEmitter, 'connection');
console.log(eventListeners + " Listener(s) listening to connection event");

console.log("Program Ended.");
```

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.