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.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>ASS-1</title>

</head>
<body>
    <form name="VALIDATION FORM" text align="center">
      <fieldset>
    <label for="fName"> First Name: </label>
      <input type="text" id="fName" "><br><br>
    <label for="lName"> Last Name: </label>
      <input type="text" id="lName" "><br><br>
    <label for="age">   Age:  </label>
      <input type="number" id="age" "><br><br>
   <center> <button id="btn" type="button"
onclick="validate()">Submit</button></center>
 </fieldset>
    </form>
    <script src="./Student.js"></script>
</body>
</html>
```

Student .js

```
function validate()
{
        let fName=document.forms["VALIDATION FORM"]["fName"];
        let lName=document.forms["VALIDATION FORM"]["lName"];
        if ((fName.value== "" && lName.value== "" ) ||
(fName.value.length==0 && lName.value.length==0))
        {
            alert("Enter the First Name");
        }
        else
        {
            if (!/^[a-zA-Z]*$/g.test(fName.value)) {
                alert("Invalid First Name");
                fName.focus();
            }
            else
            {
                alert("first Name is Valid");
            }

            if (!/^[a-zA-Z]*$/g.test(lName.value)) {
                alert("Invalid Last Name");
                lName.focus();
```

```
            }
            else
            {
                alert("last Name is Valid");
            }
        }
        let age=document.forms["VALIDATION FORM"]["age"];
        if(age.value=="")
        {
            alert("Enter age First");
        }
        else
        {
            if(age.value<18 || age.value>50)
        {
            alert("Age is Invalid");
        }
        else
        {
            alert("Age is Valid");
        }
        }

}
```

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

Employee.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>ASSIGNMENT-2</title>
</head>
<body>
    <form name="EMPFORM" text align="center">
     <fieldset>

     <label for="Name"> Enter Name : </label>
       <input type="text" id="Name" ><br><br>
     <label for="dob">  Enter DOB (MM/DD/YYYY): </label>
       <input type="text" id="dob"><br><br>
        <label for="doj" >Select Date of joining :</label>
     <label for="doj"> </label>
        <input type="date" id="doj" ><br><br>
     <label> Enter Salary :</label>
        <input type="number" name="" id="salary" ><br><br>
     <center><button id="btn" type="button"
onclick="validate()">submit</button></center>
    </fieldset>
    </form>
    <script src="./Employee.js"></script>
</body>
</html>
```

Employee.js

```
function validate()
{
        let Name=document.forms["EMPFORM"]["Name"];

        if ( Name.value== "" || Name.value.length==0)
        {
            alert("Enter Name First");
            Name.focus();
        }

       let dob=document.forms["EMPFORM"]["dob"];
        let date = /^(0?[1-9]|1[0-2])[\/](0?[1-9]|[1-2][0-
9]|3[01])[\/]\d{4}$/;

        if (date.test(dob.value)) {
            alert("Date follows MM/DD/YYYY format");
        }
        else{
          alert("Invalid date format");
        }

        let doj=document.forms["EMPFORM"]["doj"];
        let ndoj=new Date(doj.value.toString());
        let currentDate= new Date();
        if(ndoj.getDate()<currentDate.getDate())
        {
            alert("Joining date should be future date");
        }
        else
        {
            alert("Correct Date Selected");
        }

        let sal=document.forms["EMPFORM"]["salary"];

        if(sal.value<1000 || sal.value>100000)
        {
            alert("Invalid Salary")
        }
        else
        {
            alert("Correct Salary");
        }
}
```

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

Mail.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>ASS-3</title>

</head>
<body>
    <center>
    <form name="Emailform" >
        <fieldset><b>
       <label for="Email">Enter Email :</label>
         <input type="email" id="Email" ><br><br>
       <label for="password"> Enter Password :</label>
         <input type="password" name="passwd" id="password" ><br><br>
       <center> <button id="btn" type="button"
onclick="validate()">submit</button></center>
   </b>
    </fieldset>
    </form>
</center>
    <script src="./Mail.js"></script>
</body>
</html>
```

Mail.js

```
function validate()
{
    let email=document.forms["Emailform"]["Email"];
    let passwd=document.forms["Emailform"]["password"];
    let Regex = /^[a-zA-Z0-9.!#$%&'*+/=?^_`{|}~-]+@[a-zA-Z0-9-
]+(?:\.[a-zA-Z0-9-]+)*$/;
        if (email.value.match(Regex))
        {
          alert("Valid email address!");
          email.focus();

        } else
        {

          alert("Invalid email address!");
          email.focus();
          return false;

        }
        if(passwd.value=="")
        {
          alert("Enter Password Please");
```

```
        }
        else
        {
          alert("login Successfully");
        }
}
```

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

```
var msg="hello World";
console.log(msg.toUpperCase());
```

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');
fs.writeFile('log.txt','Hello Node js', function(err)
{
    if(err) throw err;
    console.log('It\'s Saved!');
});
fs.appendFile('message.txt','Append Data into File', function(err)
{
    if(err) throw err;
    console.log('Saved!');
});
console.log(new Date().toISOString());
[...Array(1000)].forEach(function (item , index)
{
    fs.appendFile("append.txt", index+ "\n" , function(err)
    {
        if(err) console.log(err);
    });
});
console.log(new Date().toISOString());
```

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 (8000);
```

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(9517, console.log("Server is running..."));
```

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

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

var mysql = require('mysql');

var con = mysql.createConnection({

 host: "localhost",

 user: "root",

 password: ""

});

con.connect(function() {

 console.log("Connected!");

 con.query("CREATE DATABASE nodejs", function

(err, result) {


 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 mysql = require('mysql');

var con = mysql.createConnection({

 host: "localhost",

 user: "root",

 password: "",

 database: "nodejsdatabase"

});

con.connect(function(err) {

 if (err) throw err;

 console.log("Connected!");

 var sql = "CREATE TABLE customers (cid int primary key auto_increment,name VARCHAR(255), address VARCHAR(255))";

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

 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 mysql = require('mysql');

var con = mysql.createConnection({

 host: "localhost",

 user: "root",

 password: "",

 database: "nodejsdatabase"

});

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

```
 console.log("Connected!");
 con.query("INSERT INTO student(sname,address) VALUES
('ashu', 'sayyed')");
 con.query("INSERT INTO student(sname,address) VALUES
('madhu', 'nashik')");
 con.query("INSERT INTO student(sname,address) VALUES
('ram', 'satara')");
 con.query("INSERT INTO student(sname,address) VALUES
('raj', 'niphad')");
 console.log("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: "root",
 password: "",
 database: "nodejs"
});
con.connect(function(err) {
 if (err) throw err;
 var sql = "DELETE FROM customers WHERE cname ='Ram'";
 con.query(sql, function (err, result) {
 if (err) throw err;
 console.log("Number of records deleted: " + result);
 });
});
```

12. Create a Simple Web Server using node js

```
var http=require('http');
var server=http.createServer(function(request,response)
{
        response.write("Hello node js...");
        response.end();
})
server.listen(8000);
```