**Assignment**

Name:-Suraj Shantaram Darde.

Class: - M.Sc (CS)-II

Div:-A

Subject:-Web framework Practical

Seat No-53

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>

<head>

<script type="text/javascript">

function validate()

{

var regName=/^[a-zA-z]+[a-zA-Z]+$/;

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

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

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

var mobno=document.getElementById("txtmobno").value;

if(age<18||age>50)

alert("student age must be 18 to 50");

if(!regName.test(fname))

alert("invalid name is given");

else

alert("valid name is given");

}

</script>

</head>

<body>

<form>

enter student first name <input type="text" name="txtfname" id="txtfname"><br>

enter student last name <input type="text" name="txtlname" id="txtlname"><br>

enter student age <input type="text" name="txtage" id="txtage"><br>

enter mobile no <input type="text" name="txtmobno" id="txtmobno"><br>

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

</form>

</body>

</html>

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

<html>

<head>

<script type="text/javascript">

function validate()

{

var regName=/^[a-zA-z]+[a-zA-Z]+$/;

var dateformatdob = /^(0?[1-9]|[12][0-9]|3[01])[\/\-](0?[1-9]|1[012])[\/\-]\d{4}$/;

var dateformatjdate = /^(0?[1-9]|[12][0-9]|3[01])[\/\-](0?[1-9]|1[012])[\/\-]\d{4}$/;

var salaryformat=/^\d{1,6}(?:\.\d{0,2})?$/

var name=document.getElementById("txtname").value;

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

var jdate=document.getElementById("txtjdate").value;

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

if(!regName.test(name))

alert("invalid name is given");

else

alert("valid name is given");

if(!dateformatjdate.test(jdate))

alert("invalid joining date is given");

else

alert("valid joining date is given");

if(!dateformatdob.test(dob))

alert("invalid date of birth is given");

else

alert("valid date of birth is given is given");

if(!salaryformat.test(salary))

alert("invalid salary");

else

alert("salary is valid");

}

</script>

</head>

<body>

<form>

<h1>Employee Rsgistration Details</h1>

enter employee first name <input type="text" name="txtfname" id="txtname"><br>

enter date of birth <input type="text" name="txtdob" id="txtdob"><br>

enter joining date <input type="text" name="txtjdate" id="txtjdate"><br>

enter salary <input type="text" name="txtsalary" id="txtsalary"><br>

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

</form>

</body>

</html>

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

Expression.

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

</style>

</head>

<body>

<script>

function validateform(){

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

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

if (/^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$/.test(email))

{

alert("Valid Email Id..")

return (true)

}

else{ alert("You have entered an invalid email address!")

return (false)

}

}

</script>

<form name="myform" onsubmit="return validateform()">

<div class="container">

<p>Please fill in this form to Login.</p>

<hr>

<label for="email"><b>Email</b></label>

<input type="text" autocomplete="off" placeholder="Enter Email" name="email" id="email" required>

<label for="psw"><b>Password</b></label>

<input type="password" autocomplete="off" placeholder="Enter Password" name="psw" id="psw" required>

<hr>

<button type="submit" class="registerbtn">Register</button>

</div>

</form>

</body>

</html>

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("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.

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"));

}

});

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 Not Found");

}

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

res.write(data);

return res.end();

});

}).listen(8080);

Q7.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);

Q8.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: "nikhil96",

database: "Node"

});

con.connect(function(err)

{

if (err) throw err;

console.log("Connected!");

con.query("CREATE DATABASE WFN", function (err, result)

{

if (err) throw err; console.log("Database created");

});

});

var sql = "CREATE TABLE customers2(name VARCHAR(25), address VARCHAR(25))"; 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: "root",

password: "nikhil96",

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);

});

Q10.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: "nikhil96",

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);

});

});