Indira College of Commerce and Science, Pune

M.Sc. II

Web Frameworks

Academic year 2022-23

Assignment 1

1. **Create an HTML form that contains the Student Registration details and write a JavaScript to validate Student’s first and last name as it should not contain other than the alphabet and the age should be between 18 to 50.**

|  |
| --- |
|  |
| <html> |
|  | <head> |
|  | <script type="text/javascript"> |
|  |  |
|  | function create\_account() |
|  | { |
|  | var fn=document.getElementById("fn").value; |
|  | var ln=document.getElementById("ln").value; |
|  | var emailid=document.getElementById("emailid").value; |
|  | var pn=document.getElementById("pn").value; |
|  | var age=document.getElementById("age").value; |
|  | var pass=document.getElementById("pass").value; |
|  | var cpass = document.getElementById("cpass").value; |
|  |  |
|  | var letters = /^[A-Za-z]+$/; |
|  | var email = /^([a-zA-Z0-9\_\.\-])+\@(([a-zA-Z0-9\-])+\.)+([a-zA-Z0-9]{2,4})+$/; |
|  | var phoneno= /^[0-9]+$/; |
|  |  |
|  | if(fn=='' || ln=='' || emailid=='' || pn=='' || pass=='' || cpass=='' || age=='') |
|  | { |
|  | alert("please fill each detail"); |
|  | } |
|  | else if(!letters.test(fn)) |
|  | { |
|  | alert("Enter only alphabets"); |
|  | } |
|  | else if(!letters.test(ln)) |
|  | { |
|  | alert("Enter only alphabets"); |
|  | } |
|  | else if(!email.test(emailid)) |
|  | { |
|  | alert("Enter valid email"); |
|  | } |
|  | else if(!phoneno.test(pn)) |
|  | { |
|  | alert("Enter valid phone"); |
|  | } |
|  | else if(age<18 || age>50) |
|  | { |
|  | alert("Enter age between 18 and 50"); |
|  | } |
|  | else if(pass != cpass) |
|  | { |
|  | alert("Enter valid password"); |
|  | } |
|  | else if(document.getElementById("pass").value.length > 12) |
|  | { |
|  | alert("Password maximum length is 12"); |
|  | } |
|  | else if(document.getElementById("pass").value.length < 6) |
|  | { |
|  |  |
|  | alert("Password minimum length is 6"); |
|  |  |
|  | } |
|  | else{ |
|  |  |
|  | alert("Your account has been created successfully..."); |
|  |  |
|  | window.location="success.html"; |
|  |  |
|  | } |
|  | } |
|  |  |
|  | </script> |
|  | </head> |
|  | <body bgcolor="pink"> |
|  | <table bgcolor="pink" align="center"/> |
|  | <tr> |
|  | <td>First Name</td> |
|  | <td><input type="text" id="fn"/></td> |
|  | </tr> |
|  | <tr> |
|  | <td>Last Name</td> |
|  | <td><input type="text" id="ln"/></td> |
|  | </tr> |
|  | <tr> |
|  | <td>Email\_Id</td> |
|  | <td><input type="text" id="emailid"/></td> |
|  | </tr> |
|  | <tr> |
|  | <td>Phone no</td> |
|  | <td><input type="text" id="pn"/></td> |
|  | </tr> |
|  | <tr> |
|  | <td>Age</td> |
|  | <td><input type="text" id="age"/></td> |
|  | </tr> |
|  | <tr> |
|  | <td>Password</td> |
|  | <td><input type="text" id="pass"/></td> |
|  | </tr> |
|  | <tr> |
|  | <td>confirm password</td> |
|  | <td><input type="text" id="cpass"/></td> |
|  | </tr> |
|  | <tr> |
|  | <td></td> |
|  | <td><input type="submit" value="Submit" onClick="create\_account()"/></td> |
|  | </tr> |
|  | </table> |
|  | </body> |
|  | </html> |
|  |  |

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

<!DOCTYPE html>

<html lang="en"><head>

<script src="employee.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>

employee.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 ufgen = document.registration.ufgen;

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,ufgen))

{

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;

}

}

/\*function underAgeValidate(births){

// it will accept two types of format yyyy-mm-dd and yyyy/mm/dd

var optimizedBirthday = births.replace(/-/g, "/");

//set date based on birthday at 01:00:00 hours GMT+0100 (CET)

var myBirthday = new Date(optimizedBirthday);

// set current day on 01:00:00 hours GMT+0100 (CET)

var currentDate = new Date().toJSON().slice(0,10)+' 01:00:00';

// calculate age comparing current date and borthday

var myAge = ~~((Date.now(currentDate) - myBirthday) / (31557600000));

if(myAge < 18)

{

alert("age is not validate");

}

else

{

alert("age is validate");

}

}

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

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>JavaScript form validation - checking email</title>

<link rel='stylesheet' href='form-style.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-validation.js"></script>

</body>

</html>

JavaScript Code

function ValidateEmail(inputText)

{

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

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;

}

}

1. **Create a Node.js file that will print the output "Hello World!" on the console.**

// the hello world program

console.log('Hello World');

1. **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); //3 - listen for any incoming requests

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

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

var http = require('http'); // includes the http module

var uc = require('upper-case'); // include the upper-case module

http.createServer(function (req, res) {

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

res.write(uc("hello world!")); // assign the upper-case module

res.end();

}).listen(8080); // port number

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

var fs=require('fs');

fs.readFile('first.txt',function(err,data)

{

fs.appendFile('second.txt',data,function(err)

{

if(err) throw err;

console.log('Saved');

});

});

//first.txt

Hello

//second.txt

World

helloworld

1. **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');

http.createServer(function (req,res)

{

fs.readFile('input.txt',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);

1. **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('./Modules');

http.createServer(function(req,res)

{

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

res.write("The date and time are currently"+dt.myDateTime());

res.end();

}).listen(8000);

***1.Module.js***

exports.myDateTime=function()

{

return Date();

};

1. **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.");