EXPERIMENT NO: 09

AIM: Develop a web page for validation of form fields using regular expressions.

Lab Outcomes:

1. To familiarize the concept and component of regular expression
2. To write simple regular expression

THEORY:

JavaScript Regular Expression:

A regular expression is a sequence of characters that forms a search pattern. The search pattern can be used for text search and text replaces operations.

What Is a Regular Expression?

- A regular expression is a sequence of characters that forms a search pattern.
- When you search for data in a text, you can use this search pattern to describe what you are searching for.
- A regular expression can be a single character, or a more complicated pattern.
- Regular expressions can be used to perform all types of text search and text replace operations.

Syntax
/pattern/modifiers;

Example
var patt = /w3schools/i; Example explained:
/w3schools/i is a regular expression. w3schools is a pattern (to be used in a search).
i is a modifier (modifies the search to be case-insensitive).

Using String Methods
In JavaScript, regular expressions are often used with the two string methods: search() and replace().
The search() method :
uses an expression to search for a match, and returns the position of the match.
The replace() method
returns a modified string where the pattern is replaced.

Using String search() With a String
The search() method searches a string for a specified value and returns the position of the match:

Example
Use a string to do a search for "W3schools" in a string: var str = "Visit W3Schools!";
var n = str.search("W3Schools");

Using String search() With a Regular Expression

Example

Use a regular expression to do a case-insensitive search for "w3schools" in a string: var str = "Visit W3Schools";
var n = str.search(/w3schools/i); The result in n will be:
6

Using String replace() With a String

The replace() method replaces a specified value with another value in a string: var str = "Visit Microsoft!";
var res = str.replace("Microsoft", "W3Schools");

Use String replace() With a Regular Expression

Example

Use a case insensitive regular expression to replace Microsoft with W3Schools in a string: var str = "Visit Microsoft!";
var res = str.replace(/microsoft/i, "W3Schools"); The result in res will be:
Visit W3Schools!

Regular Expression Modifiers

Modifiers can be used to perform case-insensitive more global searches:

Modifier	Description
i	Perform case-insensitive matching
g	Perform a global match (find all matches rather than stopping after the first match)
m	Perform multiline matching

Regular Expression Patterns

Brackets are used to find a range of characters:
Expression Description
[abc] Find any of the characters between the brackets [^abc] Find any character NOT between the brackets [0-9] Find any of the digits between the brackets
[^0-9] Find any character NOT between the brackets (any non-digit) (x|y) Find any of the alternatives separated with |

Quantifiers

Quantifier	**Description**
n+	Matches any string that contains at least one n
n*	Matches any string that contains zero or more occurrences of n
n?	Matches any string that contains zero or one occurrences of n
n{X}	Matches any string that contains a sequence of X n's
n{X,Y}	Matches any string that contains a sequence of X to Y n's
n{X,}	Matches any string that contains a sequence of at least X n's
n$	Matches any string with n at the end of it
^n	Matches any string with n at the beginning of it

?=n Matches any string that is followed by a specific string n
?!n Matches any string that is not followed by a specific string n

RegExp Object Properties Property Description

constructor Returns the function that created the RegExp object's prototype
global Checks whether the "g" modifier is set
ignoreCase Checks whether the "i" modifier is set
lastIndex Specifies the index at which to start the next match
multiline Checks whether the "m" modifier is set
source Returns the text of the RegExp pattern

RegExp Object Methods

compile()
Deprecated in version 1.5. Compiles a regular expression
exec() Tests for a match in a string. Returns the first match
test() Tests for a match in a string. Returns true or false
toString() Returns the string value of the regular expression

Using test()
The following example searches a string for the character "e":
Example
var patt = /e/;
patt.test("The best things in life are free!");
Since there is an "e" in the string, the output of the code above will be:
true
You don't have to put the regular expression in a variable first. The two lines above can be shortened to one:
/e/.test("The best things in life are free!");

Using exec()
The exec() method is a RegExp expression method.
It searches a string for a specified pattern, and returns the found text as an object. If no match is found, it returns an empty (null) object.
The following example searches a string for the character "e": Example 1
/e/.exec("The best things in life are free!");

Example:

```
<html>

<head>
   <title>creating mailing system</title>
   <style>
```

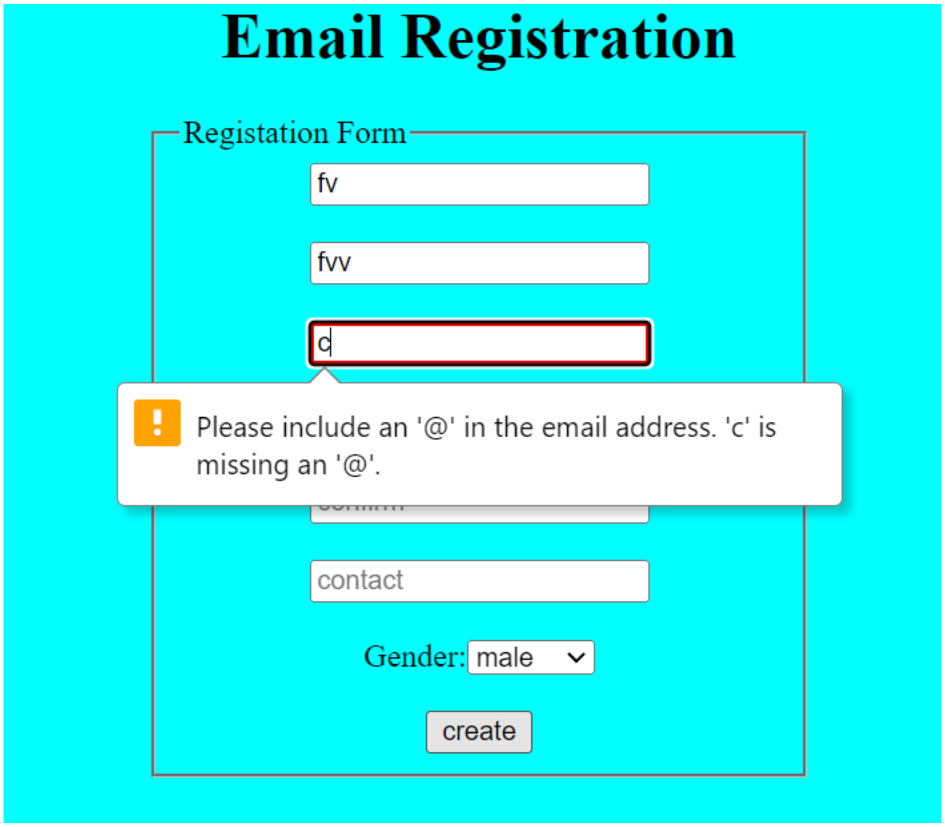
```
        legend {
            display: block;
            padding-left: 2px;
            padding-right: 2px;
            border: none;
        }
    </style>
    <script type="text/javascript">
        function validate() {

            var user = document.getElementById("e").value;
            var user2 = document.getElementById("e");
            var re = /^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/;
            if (re.test(user)) {
                alert("done");
                return true;
            }
            else {
                user2.style.border = "red solid 3px";
                return false;
            }
        }
    </script>
</head>

<body bgcolor="cyan">
    <center>
        <h1>Email Registration</h1>
        <form>
            <fieldset style="width:300px">
                <legend>Registation Form</legend>
                <table>
                    <tr>
                        <input type="text"
                            placeholder="firstname"
                            maxlength="10">
                    </tr>
                    <br><br>
                    <tr>
                        <input type="text"
                            placeholder="lastname"
                            maxlength="10">
                    </tr>
                    <br><br>
                    <tr>
                        <input type="email"
```

```
                            placeholder="username@gmail.com" id="e">
                    </tr>
                    <br><br>
                    <tr>
                        <input type="password" placeholder="password">
                    </tr>
                    <br><br>
                    <tr>
                        <input type="password" placeholder="confirm">
                    </tr>
                    <br><br>
                    <tr>
                        <input type="text" placeholder="contact">
                    </tr>
                    <br><br>
                    <tr>
                        <label>Gender:</label>
                        <select id="gender">
                            <option value="male">male</option>
                            <option value="female">female</option>
                            <option value="others">others</option>
                        </select>
                    </tr>
                    <br><br>
                    <tr><input type="submit"
                        onclick="validate()" value="create">
                    </tr>
                </table>
            </fieldset>
        </form>
    </center>
</body>
  </html>
```

OUTPUT



Email Registration

Registation Form

fv

fvv

c

Please include an '@' in the email address. 'c' is missing an '@'.

contact

Gender: male

CONCLUSION:

Thus, we have implemented regular expression to validate a form fields.