

## User Registration Validation

### Objective:

Create a user registration form with an input validation using JavaScript. Handle errors and provide appropriate feedback to users.

### HTML:

Begin by completing the index.html file with a user registration form. Include the following fields with appropriate labels and input boxes/button with id strictly the same as:

Full Name, id = fullName

Email Address, id = email

Password, id = password

Confirm Password, id = confirmPassword

Submit Button, id = submit

The submit button on clicking should call the function validateForm() from the script.js file.

**Note:-** If the same ids are not followed, you will face problem(s).

Use the appropriate type attribute for each input box. The full name input box will have the type text, the email address input box will have the type email, and the password and confirm password input boxes will have the type password. The submit button will have the type button.

Follow this [Link](#) to look at possible input field type attributes.

### *Important Note:-*

An empty <div> container with id feedback is provided in index.html. Do not modify it. Use that container in your JavaScript file to give back error messages/feedback.

### CSS:

In style.css just make sure that every component is centered and keep the background colour of the body as azure, as shown in the above image

# User Registration Form

Full Name:

Email Address:

Password:

Confirm Password:

## JavaScript:

Complete the script.js file. Most of the instructions are written as comments in that file.

You must complete the validateName, validateEmail, validatePassword, ConfirmPassword, and validateForm functions. These functions don't return anything.

Whenever you have to give feedback about a successful entry or an error message, refer to the instructions in the **How to give your feedback/error message to the HTML page from a JavaScript file?**

Now, read the content below to learn more about the functions of the script.js file.

When **validateForm()** is called, the **try** block has code that runs in the following order:-

Checks if all fields are non-empty/filled or not. If not, throw the error message **Error: All fields are required..**

Calls the validateName() function

Calls the validateEmail() function

Calls the validatePassword() function

Calls the ConfirmPassword() function

Gives the feedback **Registration successful!** (in **green** colour) to the HTML page.

The **catch** block contains the code that gives feedback to the HTML page about the error message it catches. The error message, if caught, will either be **Error: All fields are required.**

or those thrown by any one of the `validateName()`, `validateEmail()`, `validatePassword()` or `ConfirmPassword()` functions.

**Note:-** The **try** block will reach step (6) successfully only if none of the steps (1) - (5) throw an error message.

The **validateName()** function :-

Check if the name is entered or not. If not, throw the error

The **validateEmail()** function :-

Check if the email entered is valid or not. An email is valid if and only if:-

It contains exactly 1 @ and,

There should be at least one character to the left of @. If yes, those characters should only be a-z or 0-9, and,

To the right of @ there should be exactly one . and,

Between@ and . there should be at least one character. If yes, those characters should be only from a to z, and,

To the right of . , there should be exactly 3 lowercase English characters.

This check is conveniently possible by Regex. Use the `test()` function of JavaScript to make it happen. Follow this [Link](#) to learn more about it.

If the email is invalid, throw the error message Error: Invalid Email Address..

The **validatePassword()** function :-

Check if the password entered is at least 8 characters long. If not, throw the error message

Error: Password must be at least 8 characters.

The **ConfirmPassword()** function :-

Check if the re-entered password matches or not. If not, throw the error message Error:

Passwords do not match.

### How to give your feedback/error message to the HTML page from a JavaScript file?

Using **innerHTML**, we can use the `<div>` feedback container of HTML to give the error messages/responses back. Follow this [link](#) to learn more about innerHTML.

When all fields are validated successfully, then on clicking Submit, **Registration successful!** should be printed in **green** color. If any one of the fields is invalidated, then on clicking Submit, an error message should appear in **red**.

We can use the **span** tag along with the innerHTML to adjust the color of the feedback. Follow this [link](#) to learn more about the span tag.

Example of a successful entry is shown in the above pic

# User Registration Form

Full Name:

Email Address:

Password:

Confirm Password:

Registration successful!