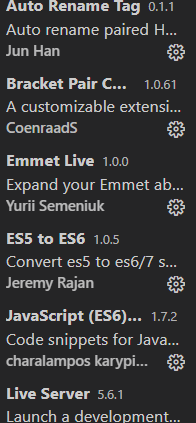
.container





Shift+alt(option on Mac)+downarrow to copy at down the line

cntrl down to select number of items with same name

<!DOCTYPE html> //!doc type chesi enter chesthay html format osthundhi

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <link rel="stylesheet" href="style.css" /> //link enter and give style sheet reference

    <title>Form Validator</title>//give your titlr

  </head>

  <body>

    <div class="container"> //.container type chesi enter chey

      <form id="form" class="form"> // form#id.class ani enter chey

        <h2>Register With SivanCloud</h2>

        <div class="form-control">// .formcontrol enter it

          <label for="username">Username</label>

          <input type="text" id="username" placeholder="Enter user name" />

          <small>Error Message</small>

        </div>

        <div class="form-control">

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

          <input type="text" id="email" placeholder="Enter email" />

          <small>Error Message</small>

        </div>

        <div class="form-control">

          <label for="password">Password</label>

          <input type="password" id="password" placeholder="Enter password" />

          <small>Error Message</small>

        </div>

        <div class="form-control">

          <label for="password2">Password2</label>

          <input

            type="password"

            id="password2"

            placeholder="Enter password again"

          />

          <small>Error Message</small>

        </div>

        <button type="submit">Submit</button>

      </form>

    </div>

    <script src="script.js"></script>

  </body>

</html>

Style.css

@import url("https://fonts.googleapis.com/css?family=Open+Sans&display=swap");

:root {

  --success-color: #2ecc71;

  --error-color: #e74c3c;

}

\* {

  box-sizing: border-box;

}

body {

  background-color: #f9fafb;

  font-family: "Open Saans", sans-serif;

  display: flex;

  align-items: center;

  justify-content: center;

  min-height: 100vh;

  margin: 0;

}

.container {

  background-color: #fff;

  border-radius: 5px;

  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.3);

  width: 400px;

}

h2 {

  text-align: center;

  margin: 0 0 20px;

}

.form {

  padding: 30px 40px;

}

.form-control {

  margin-bottom: 10px;

  padding-bottom: 20px;

  position: relative;

}

.form-control label {

  color: #777;

  display: block;

  margin-bottom: 5px;

}

.form-control input {

  border: 2px solid #f0f0f0;

  border-radius: 4px;

  display: block;

  width: 100%;

  padding: 10px;

  font-size: 14px;

}

.form-control input:focus {

  outline: 0;

  border-color: #777;

}

.form-control.success input {

  border-color: var(--success-color);

}

.form-control.error input {

  border-color: var(--error-color);

}

.form-control small {

  color: var(--error-color);

  position: absolute;

  bottom: 0;

  left: 0;

  visibility: hidden;

}

.form-control.error small {

  visibility: visible;

}

.form button {

  cursor: pointer;

  background-color: #3498db;

  border: 2px solid #3498db;

  border-radius: 4px;

  color: #fff;

  display: block;

  font-size: 16px;

  padding: 10px;

  margin-top: 20px;

  width: 100%;

}

Script.js

const form = document.getElementById("form");

const username = document.getElementById("username");

const email = document.getElementById("email");

const password = document.getElementById("password");

const password2 = document.getElementById("password2");

//show input error message

function showError(input, message) {

  const formControl = input.parentElement;

  formControl.className = "form-control error";

  const small = formControl.querySelector("small");//The querySelector() method only returns the first element that matches the specified selectors. To return all the matches, use the [querySelectorAll()](https://www.w3schools.com/jsref/met_document_queryselectorall.asp) method instead.

  small.innerText = message;

}

//Show success outline

function showSuccess(input) {

  const formControl = input.parentElement;//parentelement will get parent container

  formControl.className = "form-control success";

}

function isValidEmail(email) {

  var re = /^(([^<>()[\]\\.,;:\s@\"]+(\.[^<>()[\]\\.,;:\s@\"]+)\*)|(\".+\"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/;

  return re.test(String(email).toLowerCase());

}

//Event listners follows below

form.addEventListener("submit", function (e) {

  e.preventDefault(); //which actually prevents default submission

  if (username.value === "") {

    showError(username, "Username is required");

  } else {

    showSuccess(username);

  }

  if (email.value === "") {

    showError(email, "Email is required");

  } else if (!isValidEmail(email.value)) {

    showError(email, "Email is not valid");

  } else {

    showSuccess(email);

  }

  if (password.value === "") {

    showError(password, "Password is required");

  } else {

    showSuccess(password);

  }

  if (password2.value === "") {

    showError(password2, "Password should match");

  } else {

    showSuccess(password2);

  }

});

const form = document.getElementById("form");

const username = document.getElementById("username");

const email = document.getElementById("email");

const password = document.getElementById("password");

const password2 = document.getElementById("password2");

//show input error message

function showError(input, message) {

  const formControl = input.parentElement;

  formControl.className = "form-control error";

  const small = formControl.querySelector("small");

  small.innerText = message;

}

//Show success outline

function showSuccess(input) {

  const formControl = input.parentElement;

  formControl.className = "form-control success";

}

function checkEmail(input) {

  const re = /^(([^<>()[\]\\.,;:\s@\"]+(\.[^<>()[\]\\.,;:\s@\"]+)\*)|(\".+\"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/;

  if (re.test(input.value.trim())) {

    showSuccess(input);

  } else {

    showError(input, "Email is not valid");

  }

}

function checkRequired(inputArr) {

  inputArr.forEach(function (input) {

    if (input.value.trim() === "") {

      showError(input, `${getFieldName(input)} is required`);

    } else {

      showSuccess(input);

    }

  });

}

//check input length

function checkLength(input, min, max) {

  if (input.value.length < min) {

    showError(

      input,

      `${getFieldName(input)} must be at least ${min} characters`

    );

  } else if (input.value.length > max) {

    showError(

      input,

      `${getFieldName(input)} must be less than ${max} characters`

    );

  }

}

//Check Passwords match

function checkPasswordMatch(input1, input2) {

  if (input1.value !== input2.value) {

    showError(input2, "Passwords do not match");

  }

}

//Get fieldname

function getFieldName(input) {

  return input.id.charAt(0).toUpperCase() + input.id.slice(1);

}

//Event listners follows below

form.addEventListener("submit", function (e) {

  e.preventDefault();

  checkRequired([username, email, password, password2]);

  checkLength(username, 3, 15);

  checkLength(password, 6, 25);

  checkEmail(email);

  checkPasswordMatch(password, password2);

});

/\*

  if (username.value === "") {

    showError(username, "Username is required");

  } else {

    showSuccess(username);

  }

  if (email.value === "") {

    showError(email, "Email is required");

  } else if (!isValidEmail(email.value)) {

    showError(email, "Email is not valid");

  } else {

    showSuccess(email);

  }

  if (password.value === "") {

    showError(password, "Password is required");

  } else {

    showSuccess(password);

  }

  if (password2.value === "") {

    showError(password2, "Password should match");

  } else {

    showSuccess(password2);

  }

});\*/