**LOG IN FORM-**

Here's an example of a login form using HTML with a username input, password input, and two command buttons for submitting and resetting the form:

html

<!DOCTYPE html>

<html>

<head>

 <title>Login Form</title>

 <style>

 .container {

 max-width: 300px;

 margin: 0 auto;

 padding: 20px;

 border: 1px solid #ccc;

 }

 .container h2 {

 text-align: center;

 }

 .container input[type="text"],

 .container input[type="password"] {

 width: 100%;

 padding: 10px;

 margin-bottom: 10px;

 }

 .container input[type="submit"],

 .container input[type="reset"] {

 width: 100%;

 padding: 10px;

 background-color: #4CAF50;

 color: #fff;

 border: none;

 cursor: pointer;

 }

 .container input[type="submit"]:hover,

 .container input[type="reset"]:hover {

 background-color: #45a049;

 }

 </style>

</head>

<body>

 <div class="container">

 <h2>Login Form</h2>

 <form action="" method="post">

 <input type="text" name="username" placeholder="Username" required>

 <input type="password" name="password" placeholder="Password" required>

 <input type="submit" value="Submit">

 <input type="reset" value="Reset">

 </form>

 </div>

</body>

</html>

This HTML code creates a login form with a container div, a heading, and a form containing an input field for the username, an input field for the password, and two buttons for submitting and resetting the form. The CSS styles the form to provide a simple and clean look.

When the user fills in the form and clicks the "Submit" button, the form will be submitted to the specified action URL (replace `action=""` with the appropriate URL). The form data can be processed on the server-side using a server-side scripting language like PHP, Python, or Node.js to handle the login functionality.

The "Reset" button will clear the form fields and reset them to their initial state.

**OUTPUT**

