Shortcode

Magic Login Pro · https://handyplugins.co/docs/magic-login-shortcode/

You can easily display the Magic Login form anywhere on your website using the following shortcode:

```
[magic_login_form]
```

In most cases, it will blend nicely with your site's design out of the box. However, if you wish to adjust the appearance, you can fully customize it with your own CSS.

Shortcode Attributes

Starting from **version 2.1**, Magic Login includes additional shortcode attributes for greater flexibility.

Attribute	Description
`redirect_to`	The target URL after login. By default, it redirects to the page where the shortcode is placed. *(Note: If redirection rules are enforced via plugin settings, they will override this value.)*
`hide_logged_in`	Controls whether to display the form to logged-in users. Accepts `"true"` (default) or `"false"`.
`error_message`	Custom error message displayed when a user does not exist.
`info_message`	Informational text displayed by default on the form.
`success_message`	Message shown after a successful email send.
`label`	Custom label text for the input field.
`button_text`	Custom text for the submit button.
`class`	Add custom HTML classes to the form wrapper for styling.

Hooks

You can control the form’s target URL programmatically using:

```
magic_login_shortcode_form_action
```

Or set a specific redirect address directly in the shortcode:

```
[magic_login_form redirect_to="https://example.com/account"]
```

Using in Custom Templates

You can render the form directly within PHP templates:

```
<?php echo do_shortcode('[magic_login_form redirect_to="https://example.com/account"]'); ?>
```

Displaying the Form for Logged-in Users

By default, the shortcode doesn’t display anything for logged-in users. This helps prevent showing a login form where it’s not needed (e.g., on “My Account” pages). If you want the form to appear even for logged-in users, set hide_logged_in="false"

```
[magic_login_form hide_logged_in="false"]
```

Custom Styling

You can easily restyle the Magic Login form using the available IDs and classes.
Here’s a minimal CSS example you can use as a starting point:

```
#magic-login-register .registration_result .error,
#magic-login-register .registration_result .info,
#magic-login-register .registration_result .success,
#magic-login-shortcode #login_error,
#magic-login-shortcode .message,
#magic-login-shortcode .success {
  border: none !important;
```

```
  padding-left: 0 !important;
}

#magic-login-shortcode {
  width: auto !important;
}

#magic-login-shortcode input {
  font-family: 'Open Sans', Helvetica, Arial, sans-serif !important;
  font-size: 16px !important;
  background: #fff !important;
  color: rgba(0, 0, 0, 0.6) !important;
  border-radius: 5px !important;
  border: 1px solid #dcdcdc !important;
  padding: 12px 20px !important;
}

#magic-login-shortcode .magic-login-submit {
  background: #2f265c !important;
  color: #fff !important;
  font-weight: 700 !important;
  font-size: 16px !important;
  padding: 12px 24px !important;
  border-radius: 5px !important;
  cursor: pointer !important;
  margin-top: 10px !important;
}
```

Tips for Custom CSS

- Target the wrapper IDs `#magic-login-shortcode` or `#magic-login-register` for general form styling.
- Use `.magic-login-submit` for the submit button, and standard input selectors for text fields.
- Keep overrides minimal — most designs can be achieved by adjusting colors, borders, or typography.
- To ensure compatibility, use `!important` only when necessary.