

Styling

We recommend you customize your theme styles using the user style files. You can customize the theme's styles with the following approaches:

Gulp based workflow

You can add your own SCSS and override the theme style in the `src/scss/user.scss` file.

user.scss

To make broader changes to the design of the theme, such as changing the color scheme or font sizes, use `src/scss/_user-variables.scss`. Any variable from `node_modules/bootstrap/scss/variables` or `src/scss/theme/_variables.scss` can be overridden with your own value.

_user-variables.scss

To remove bootstrap components, update `src/scss/_bootstrap.scss` file.

_bootstrap.scss

If you are not using Gulp based workflow

You can add your own CSS and override the theme style in the `public/assets/css/user.css` file.

user.css

Changing theme colors

Changing theme colors `public/assets/css/user.css` file.

user.css

Changing theme colors

Gulp based workflow

If you are using gulp base work flow, you can update your theme colors with two different approaches.

1. Using scss variable:

You can bring your necessary variable colors form `src/scss/theme/_colors.scss` and paste it into `src/scss/_user-variables.scss` then update variables as your necessity. We recommend to follow this approach.

_user-variables.scss

Light mode is default theme style in phoenix. So, if you update any bootstrap's SCSS variables, it will effect in Light mode. If you want to update any color, find the corresponding variable for this color and place theme to _user-variables.scss file. For example:

```
/*-----  
|| Theme Colors  
/*-----*/  
  
$primary: $blue !default;  
$secondary: $gray-600 !default;  
$success: $green !default;  
$info: $cyan !default;  
$warning: $yellow !default;  
$danger: $red !default;  
$light: $gray-100 !default;  
$dark: $gray-1100 !default;
```

Most of the background and text colors used from body colors, to update the colors change the corresponding variables.

```
/*-----  
|| Body Colors  
/*-----*/  
  
$body-color: $gray-900 !default;  
$body-bg: $gray-50 !default;  
  
$body-secondary-color: $gray-800 !default;  
$body-secondary-bg: $gray-200 !default;  
  
$body-tertiary-color: $gray-700 !default;  
$body-tertiary-bg: $gray-300 !default;  
  
$body-quaternary-color: $gray-500 !default;  
$body-quaternary-bg: $gray-400 !default;  
  
$body-emphasis-color: $gray-1100 !default;  
$body-emphasis-bg: $white !default;  
  
$body-highlight-color: $gray-1000 !default;  
$body-highlight-bg: $gray-100 !default;
```

If you want to update any theme colors for dark mode, update `$theme-colors-dark` variable and to update the grays colors you have to update the `$grays-dark` variable.

```
/*-----  
| Theme Styles  
/*-----*/  
  
$theme-colors-dark: (  
  'primary': $primary-dark,  
  'secondary': $secondary-dark,  
  'success': $success-dark,  
  'info': $info-dark,  
  'warning': $warning-dark,  
  'danger': $danger-dark,  
) !default;
```

All Colors

All **Phoenix** colors are available as Sass variables and Sass map in `src/scss/theme/_colors.scss` file. To avoid increased file sizes, we don't create text or background color classes for each of these variables. Instead, we choose a subset of these colors for a theme palette.

You'll find all the colors used in the theme in a sass map name `$theme-color-shades` in `src/scss/theme/_colors.scss` file. If you want to use any **Phoenix** color to generate `text/bg` utility class, Add the color variable in `$theme-color-shades` map. If you want to add/change color for dark mode, update the `$theme-color-shades-dark` map.

```
$theme-color-shades: (  
  'primary-lighter': $blue-200,  
  'primary-light': $blue-300,  
  'primary-dark': $blue-600,  
  'primary-darker': $blue-700,  
  
  'secondary-lighter': $gray-200,  
  'secondary-light': $gray-300,  
  'secondary-dark': $gray-1000,  
  'secondary-darker': $gray-700,  
  
  'warning-lighter': $orange-200,  
  'warning-light': $orange-300,  
  'warning-dark': $orange-600,  
  'warning-darker': $orange-700,  
  
  'danger-lighter': $red-200,  
  'danger-light': $red-300,  
  'danger-dark': $red-600,  
  'danger-darker': $red-700,  
  
  'success-lighter': $green-200,  
  'success-light': $green-300,  
  'success-dark': $green-600,  
  'success-darker': $green-700,  
  
  'info-lighter': $cyan-200,  
  'info-light': $cyan-300,  
  'info-dark': $cyan-600,  
  'info-darker': $cyan-700,  
) !default;
```

2. Using CSS variable:

_user.scss

To update theme color with this approach, you have to update CSS variable for both modes. If you change any color you have to assign its corresponding RGB color too. Please note that sometimes you may need to update corresponding component's CSS variable. For example:

```
/*-----  
| Theme Styles  
/*-----*/  
  
$success: #00d27a;  
$danger: #e63757;  
:root, [data-bs-theme='light'] {  
  --phoenix-primary: #{$success};  
  --phoenix-primary-rgb: #{to-rgb($success)};  
  ...  
  ...  
}  
[data-bs-theme='dark'] {  
  --phoenix-primary: #{$danger};  
  --phoenix-primary-rgb: #{to-rgb($danger)};  
  .card {  
    --phoenix-card-bg: #{$success};  
  }  
}
```

If you are not using Gulp based workflow

You can add your own CSS and override the theme style in the `public/assets/css/user.css` file. You can update all theme colors including grays using css variables. If you change any color you have to assign its corresponding rgb color. Sometimes you may need to update corresponding component's css variable too.

```
/*-----  
| Theme Styles  
/*-----*/  
  
:root, [data-bs-themes='light'] {  
  --phoenix-primary: #00d27a;  
  --phoenix-primary-rgb: 0, 210, 122;  
  ...  
  ...  
  /* grays */  
  --phoenix-gray-100: #f9fafd;  
  --phoenix-gray-100-rgb: 249, 250, 253;  
}  
[data-bs-theme='dark'] {  
  --phoenix-primary: #e63757;  
  --phoenix-primary-rgb: 230, 55, 87;  
  .card {  
    --phoenix-card-bg: #00d27a;  
  }  
}
```