
Table of Contents

Introduction	1.1
HTML	1.2
Html Structure	1.3
Header	1.3.1
Left Side Menu	1.3.2
Right Side Menu	1.3.3
Main Content	1.3.4
Buttons	1.3.4.1
Font Icons	1.3.4.2
Progress Bars	1.3.4.3
Maps	1.3.4.4
Lock Screen	1.3.4.5
FAQ's	1.4
SCSS	1.5
CSS Files	1.6
Javascript Files	1.7
Plugins	1.8
Start New Page	1.9
Menu Bar Fold	1.9.1
Boxed and Moveable Header	1.9.2
Moveable Header	1.9.3
Boxed and Fixed Header	1.9.4
Fixed Header and Menu	1.9.5
Vue Version	1.10
Basic Installation	1.10.1
Clear 1.7 installation	1.10.1.1
Basic Structure	1.10.2
Layout structure	1.10.2.1
Leftmenu structure	1.10.2.2
Start New Page	1.10.2.3

Changing layouts	1.10.2.4
Changing Color Scheme	1.11
Laravel Spark Skin	1.12
Spark Installation	1.12.1
Using .vue files	1.12.2
Adding External Plugins	1.12.3
Laravel Version	1.13
Changing skin	1.13.1
Changing Layouts	1.13.2

Clear Documentation

Welcome to the Clear documentation.

please click next button to start reading.

This Document takes you briefly through the description about features that are included the Clear Admin Template.

Theme has the following main structural units:

- 1) Header**
- 2) Left Side Menu**
- 3) Right Side Menu**
- 4) Main Content**

Header:

Header has the following design:

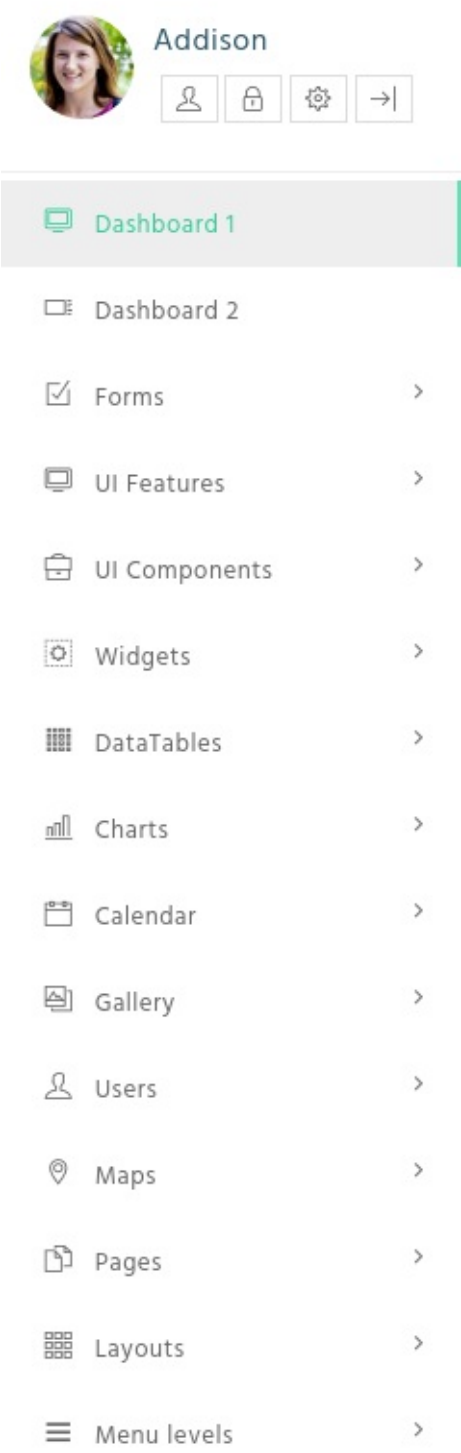


It has the following structure:

```
<header class="header">
  <nav class="navbar navbar-static-top">
    content...
  </nav>
</header>
```

Left Side Menu

It has the following design:

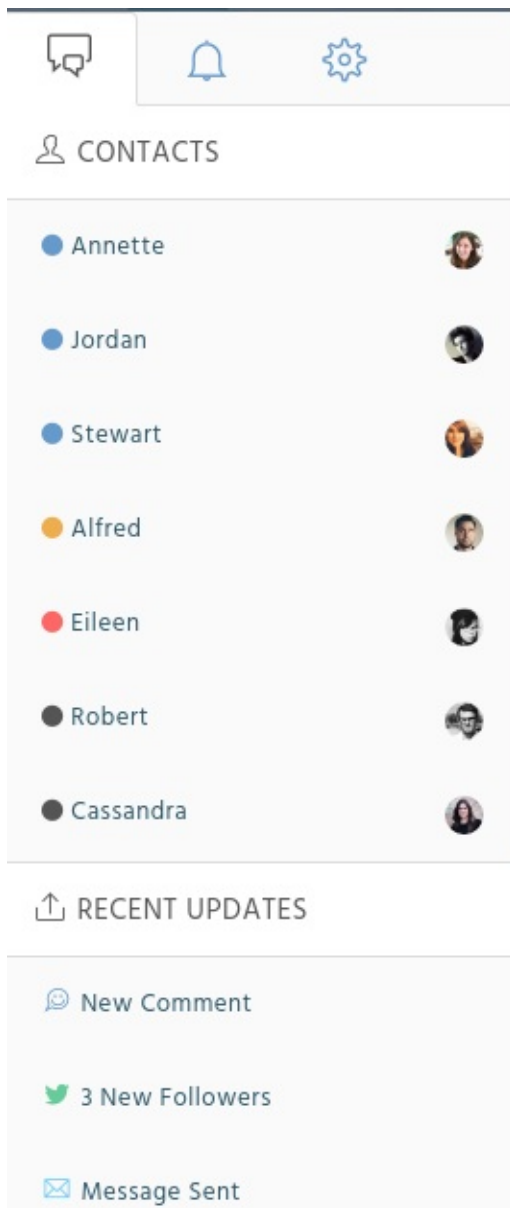


The Side Bar has the following structure :

```
<section class="sidebar">
  <div id="left" role="navigation">
    ...
  </div>
</section>
```


Right Side Menu

It has the following design:



The Side Bar has the following structure :

```
<div class="rightsidebar-right">
  <div class="rightsidebar-right-content">
    ...
    ...
  </div>
</div>
```


Main Content

The Main Content has the following structure :

```
<div class="wrapper row-offcanvas row-offcanvas-left">
  <aside class="left-side sidebar-offcanvas">
    ...
    ...
  </aside>
  <aside class="right-side">
    <section class="content-header">
      ...
    </section>
    ...
  </aside>
</div>
```

Buttons

There are twelve variants of buttons based on the colors used in this template with an option to change the color of the button just by replacing `btn-primary` with `btn-default``btn-info``btn-success``btn-warning``btn-danger` in the code below.

1) Radiused Buttons:

we can change effect of any button by replacing `hvr-buzz` with the effect we would like `hvr-hang` `hvr-sink` `hvr-pop` and other in the code below.

Radiused Buttons



It has the following Structure:

```
<button class="button button-rounded button-flat hvr-buzz">Button</button>
```

2) Rounded Buttons:

we can change effect of any button by replacing `hvr-shrink` with the effect we effect we want like `hvr-pulse` `hvr-grow` `hvr-push` and other in the code below.

Rounded Buttons



It has the following Structure:

```
<button class="button button-pill button-primary-flat hvr-shrink">shrink</button>
```

3) Rectangle Buttons:

Rectangle Box



It has the following Structure:

```
<button class="button button-primary-flat hvr-wobble-bottom">Button</button>
```

4) Circle Buttons:

Circle Buttons



It has the following Structure:

```
<button class="button button-circle button-primary-flat hvr-skew">Button</button>
```

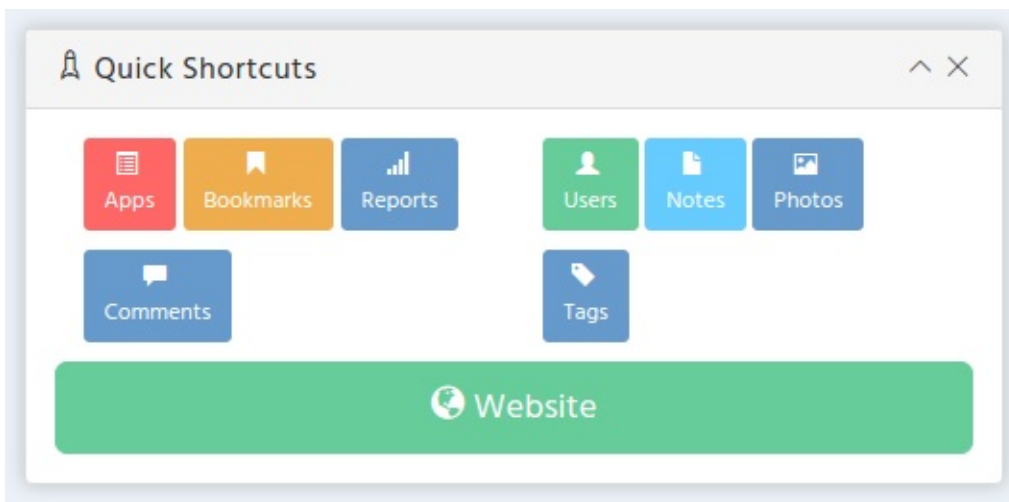
5) Shadow and Glow Transitions Buttons:



It has the following Structure:

```
<button class="button button-glow button-rounded button-flat hvr-float-shadow">Button</button>
```

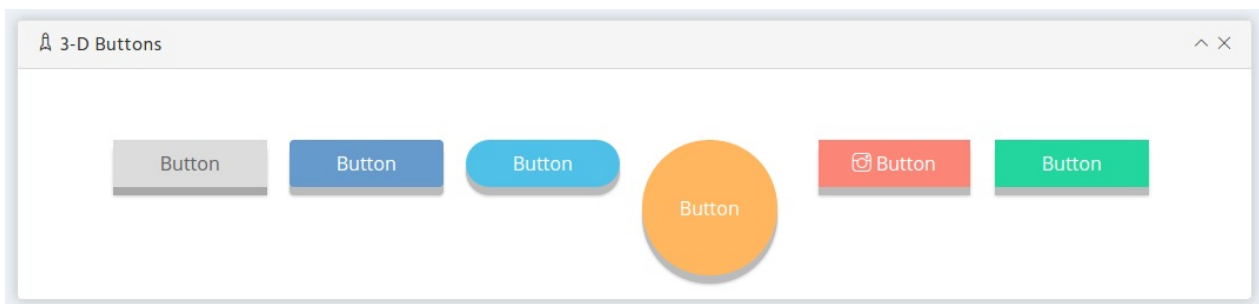
6) Quick Shortcuts:



It has the following Structure:

```
<div class="btn btn-danger btn-sm btn-responsive" role="button">
  <span class="glyphicon glyphicon-list-alt">Apps</span>
</div>
```

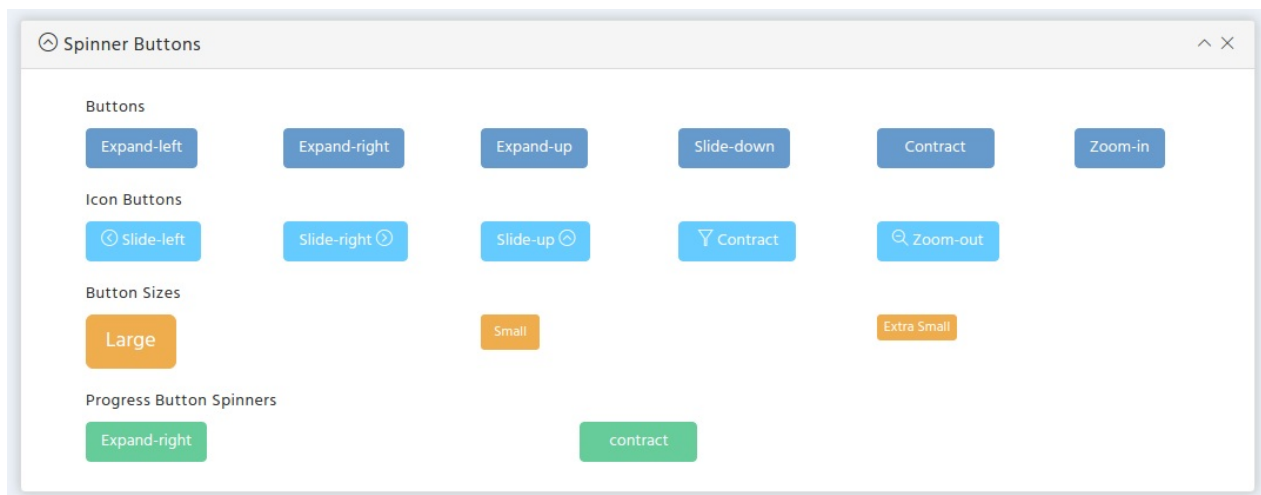
7) 3-D Buttons:



It has the following Structure:

```
<button class="button button-3d">Button</button>
```

8) Spinner Buttons:



It has the following Structure:

```
<button class="ladda-button btn btn-primary button_normal" data-style="expand-left">
  <span class="ladda-label">Expand Left</span>
  <span class="ladda-spinner"></span>
</button>
```

9) Icon Button & Dropdown:

Icon Button & Dropdown



It has the following Structure:

```
<button type="button" class="btn btn-icon btn-default m-r-50">
  <i class="icon fa fa-fw fa-map-marker" aria-hidden="true"></i>
</button>
```

10) Button Animation:

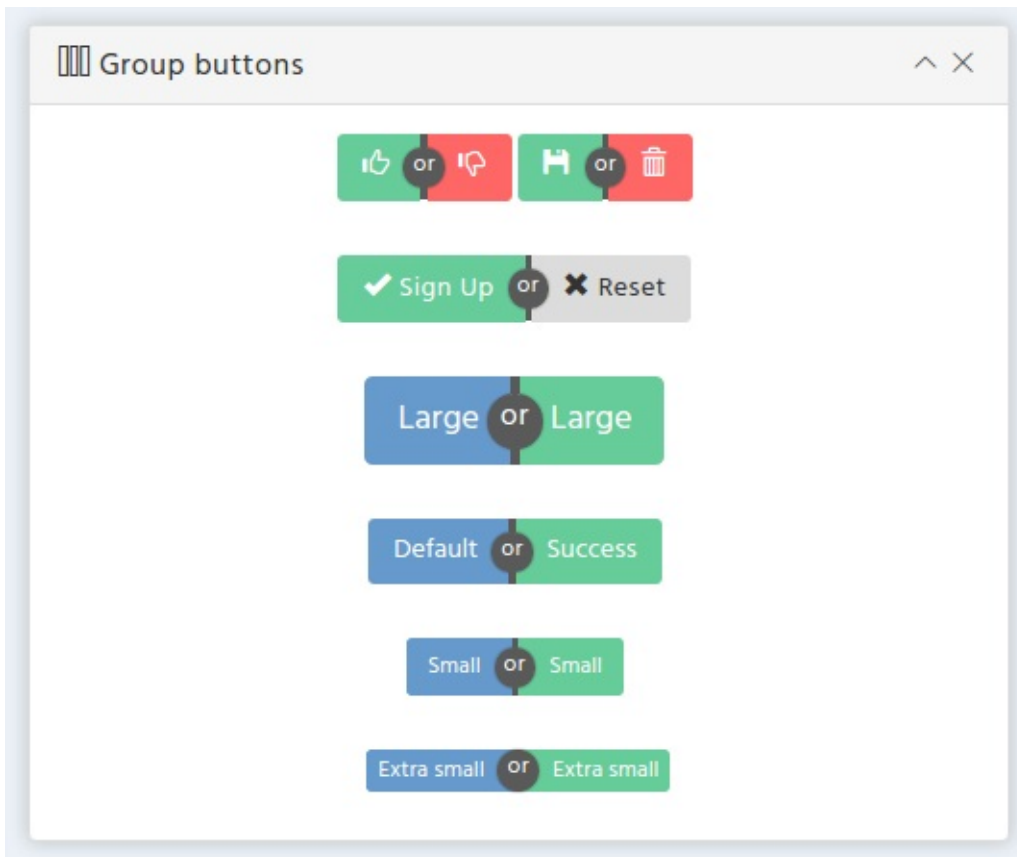
Button Animation



It has the following Structure:

```
<button type="button" class="btn btn-animate btn-animate-side btn-success m-r-50">
  <span>
    <i class="icon fa fa-fw fa-map-marker" aria-hidden="true"></i>Side Animation
  </span>
</button>
```

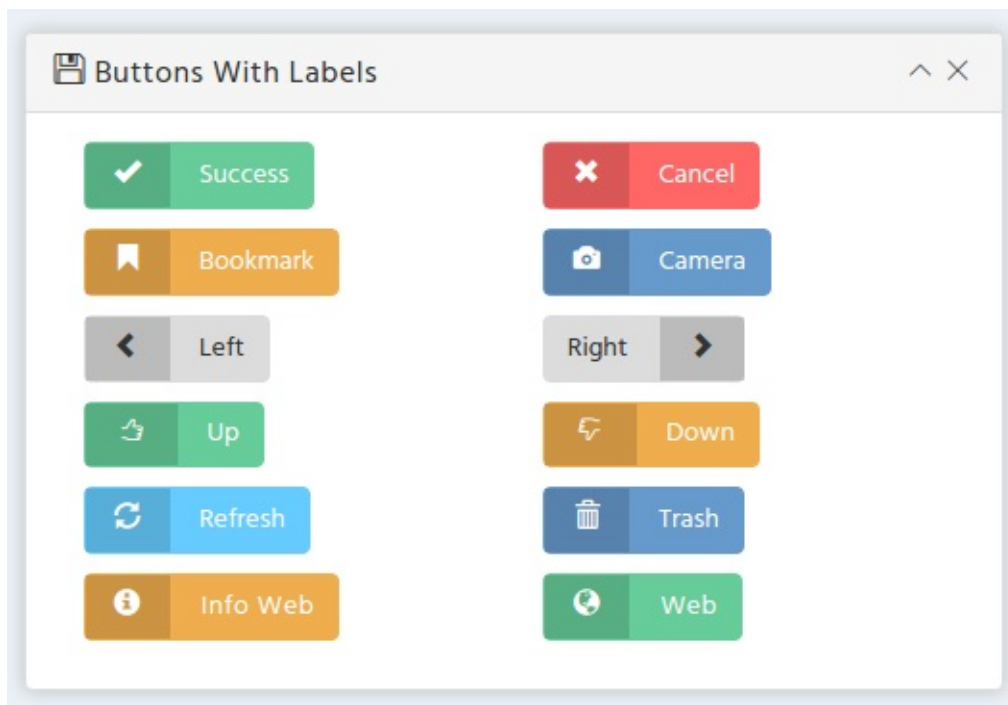
11) Group Buttons:



It has the following Structure:

```
<div class="ui-group-buttons">
  <a href="#" class="btn btn-success" role="button">
    <span class="glyphicon glyphicon-thumbs-up">.....</span>
  </a>
  ...
  ...
</div>
```

12) Buttons With Labels:



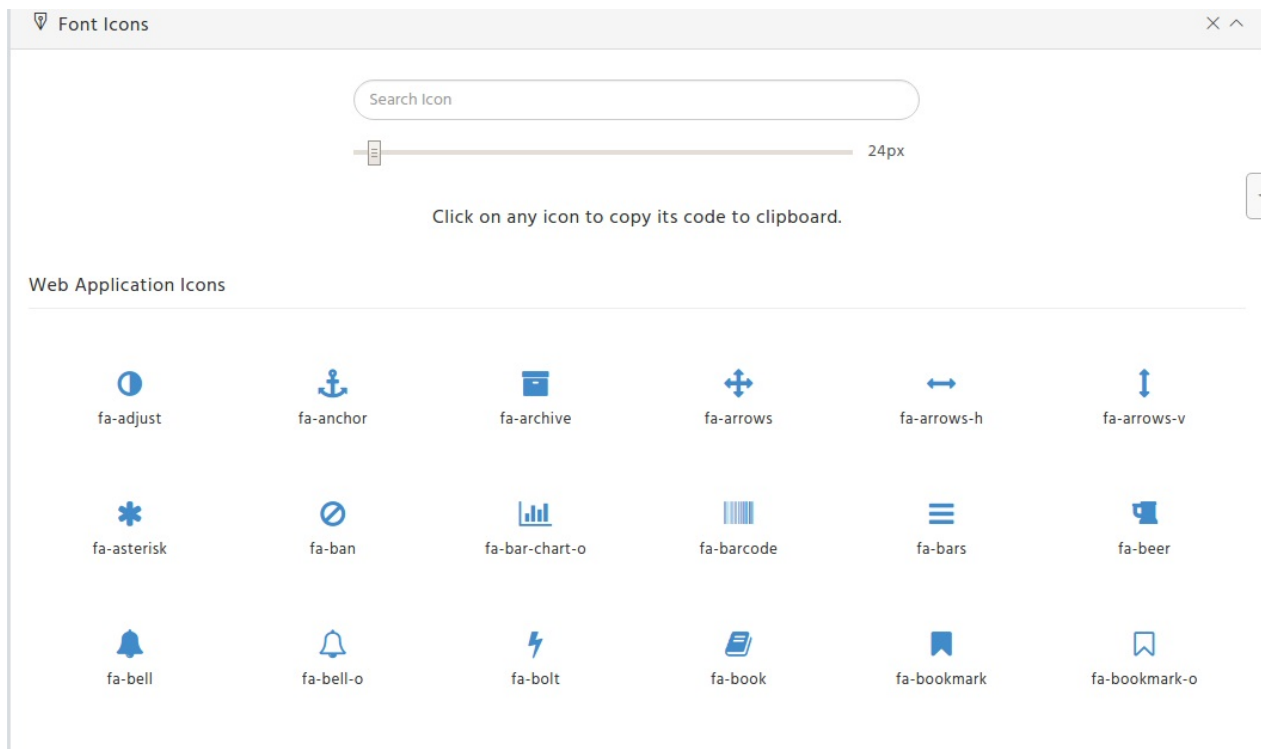
It has the following Structure:

```
<button type="button" class="btn btn-labeled btn-success">
  <span class="btn-label">
    <i class="glyphicon glyphicon-ok" aria-hidden="true"></i>Success
  </span>
</button>
```

Font Icons

There are four different types of icons included in this theme. In this page we gave an option for maximizing the size of a font and upon a click, the font gets copied to the clipboard and can be pasted anywhere. We can see the demo for this in the page [Font Icons](#) of this Template.

1) Font Awesome Icons :



It has the following Structure:

```
<a href="#">
  <i class="fa fa-phone"></i>
</a>
```

Icons Size - determines the size of icons, can take such values: fa-lg, fa-2x, fa-3x, fa-4x and fa-5x

```
<a href="#">
  <i class="fa fa-phone fa-lg"></i>
</a>
```

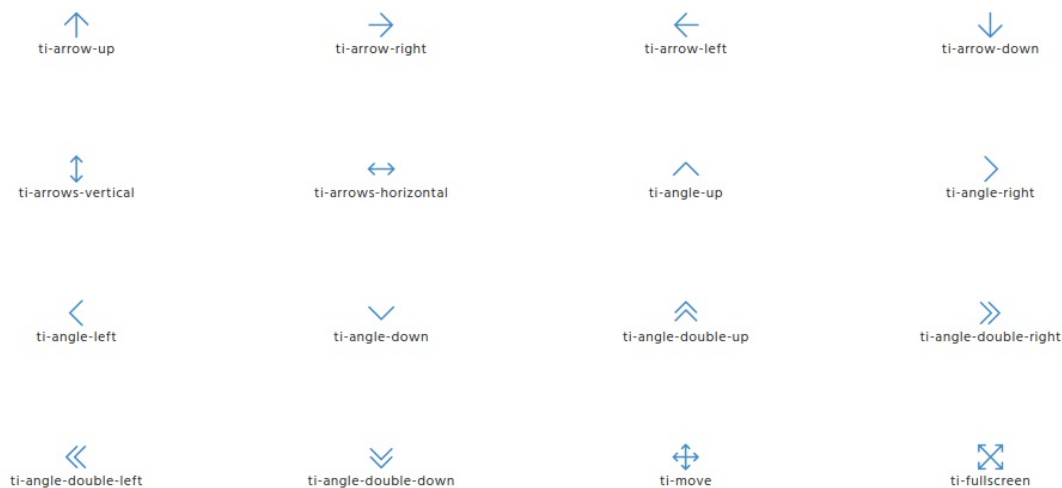
Stacked icons - It stacks the multiple icons. we can also control the size of the icons using the parameters : fa-stack-1x fa-stack-2x.

```
<span class="fa-stack fa-lg">
  <i class="fa fa-circle fa-stack-2x text-info"></i>
  <i class="fa fa-home fa-stack-2x fa-inverse"></i>
</span>
```

2) Themify Icons

Themify Icons

Arrows & Direction Icons



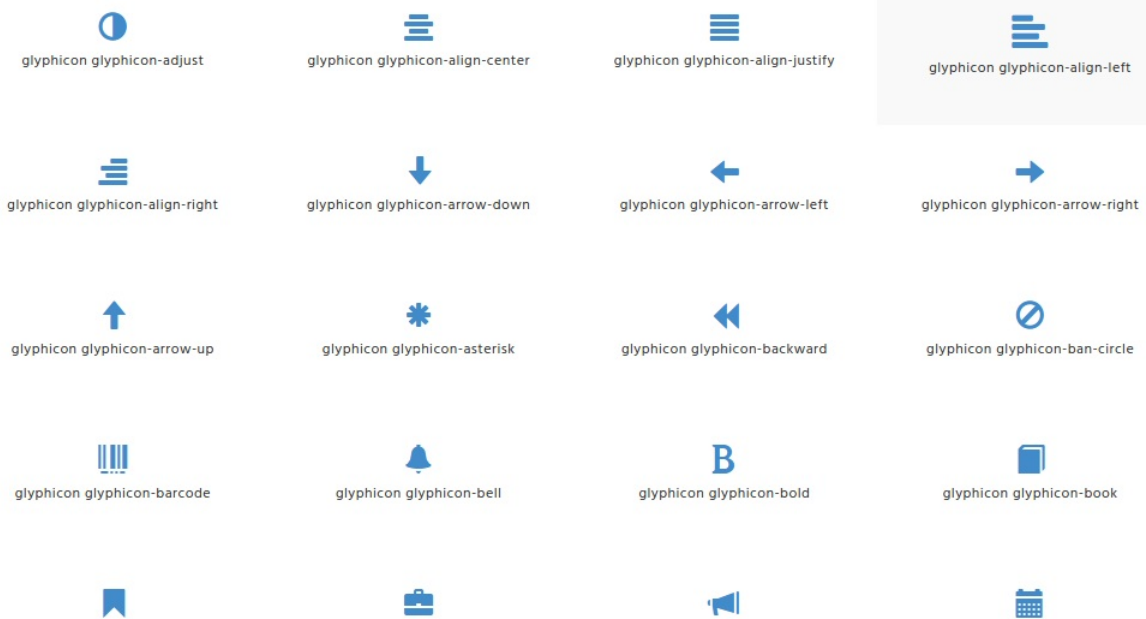
It has the following Structure:

```
<a href="#">
  <i class="ti arrow-up"></i>
</a>
```

We can import our desired icon just by replacing `arrow-up` in the above code with the icon we want.

3) Glyphicons

Glyphicons provided by Bootstrap

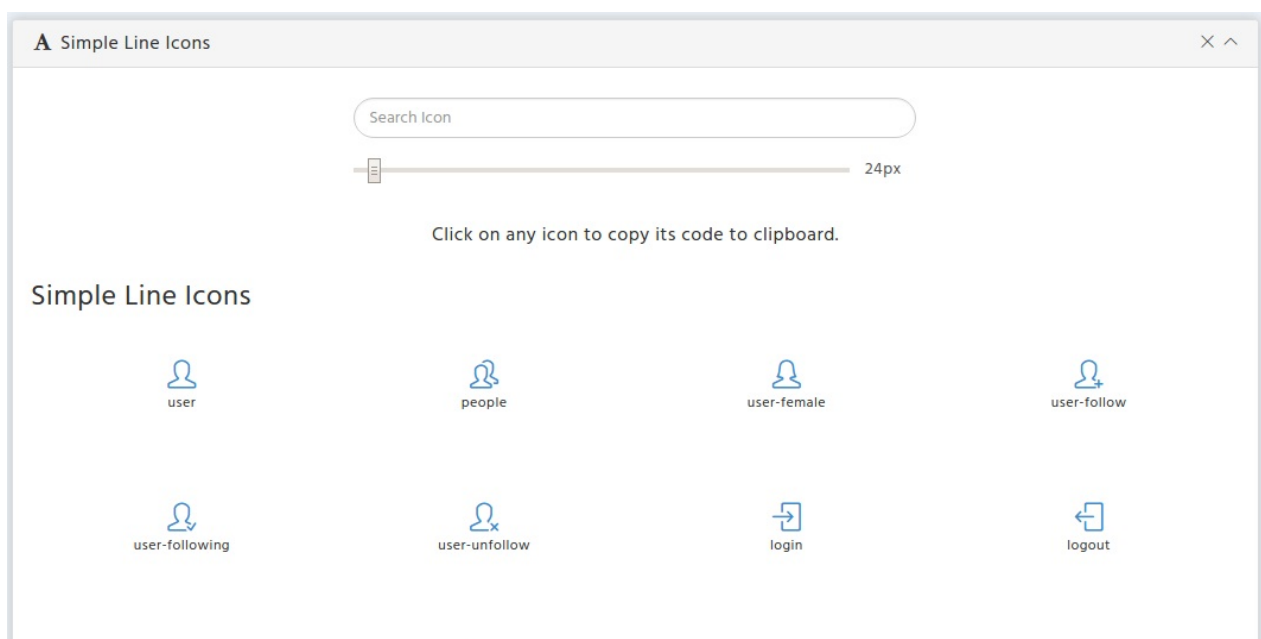


It has the following Structure:

```
<li>
  <span class="glyphicon glyphicon-cloud-download"> </span>
  <span class="glyphicon-class">glyphicon glyphicon-cloud-download</span>
</li>
```

We can import our desired icon just by replacing `glyphicon-cloud-download` in the above code with the icon we really need.

4) Simple Line Icons

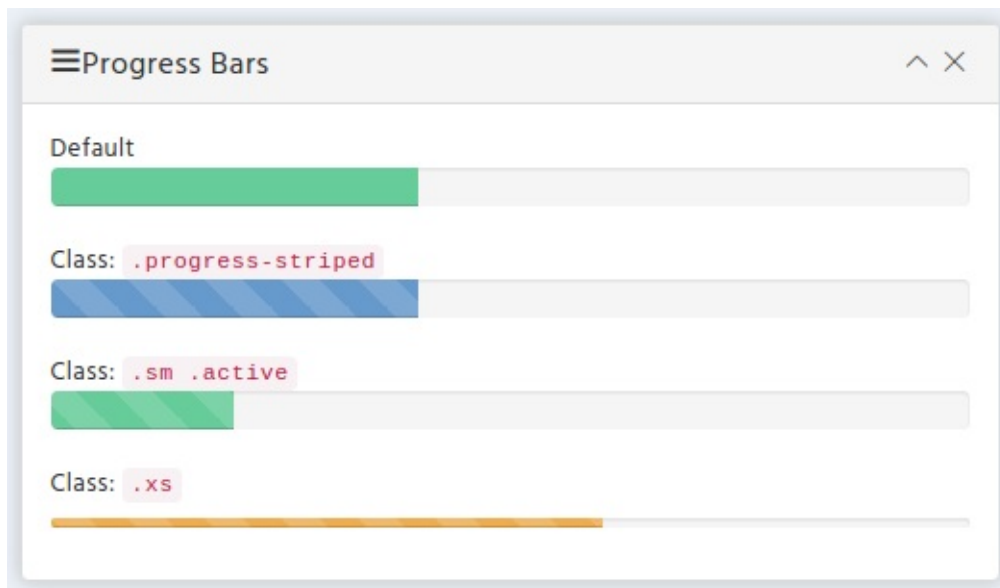


It has the following Structure:

```
<div class="preview">
  <i class="icon-user icons"></i>
  <span class="name">user</span>
</div>
```

Progress Bars

It has the following design:



Horizontal Progress Bar

```
<div class="progress">
  <div class="progress-bar" style="width: 70%;">Content</div>
</div>
```

parameters:

1) Background color - sets the color of the background. Can take values - progress-info, progress-success, progress-warning, progress-danger.

```
<div class="progress">
  <div class="progress-bar progress-bar-success" style="width: 70%;">Content</div>
</div>
```

2) striped - uses a gradient to create a striped effect.

```
<div class="progress progress-striped">
  <div class="progress-bar" style="width: 70%;">Content</div>
</div>
```

3) animated striped - add `.active` to `.progress-striped` to animate the stripes right to left.

```
<div class="progress progress-striped active">
  <div class="progress-bar" style="width: 70%;">Content</div>
</div>
```

4) **animated** - sets the animation by adding the attribute `data-appear-progress-animation`.

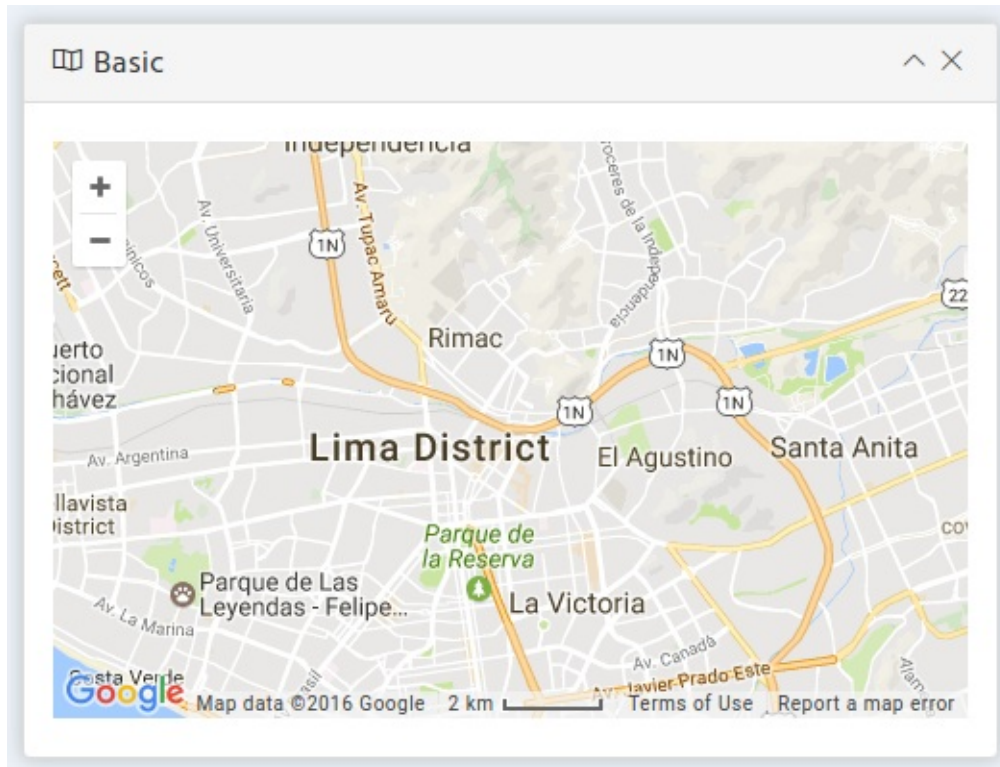
```
<div class="progress" data-appear-progress-animation="70%">
  <div class="progress-bar">Content</div>
</div>
```

Maps

There are three divergent maps used included in this template.

1) Google Maps :

We can find different styles of this kind in the page [Google Maps](#).



It has the following structure:

```
<div id="gmap-top" class="map">
</div>
```

parameters:

i) **zoom** - This attribute sets the scale of the map (by default - '8'). This option is available in the js file.

```
zoom : 8
```

ii) **center [lat , lng]** - these attributes set the coordinates of the center of the map and the point of the map (by default - '17.400408 and '78.507905'). This option is available in Js file.

```
center : [17.400408, 78.507905]
```


iii) **hue** - sets the hue of the feature to match the hue of the color supplied. Note that the saturation and lightness of the feature is conserved, which means that the feature will not match the color supplied exactly. Valid values: An RGB hex string, i.e. '#ff0000'. This option is available in the js file.

```
hue : '#0075c2'
```

iv) **Address** - This feature allows us to mention the current location. This option is available in the js file.

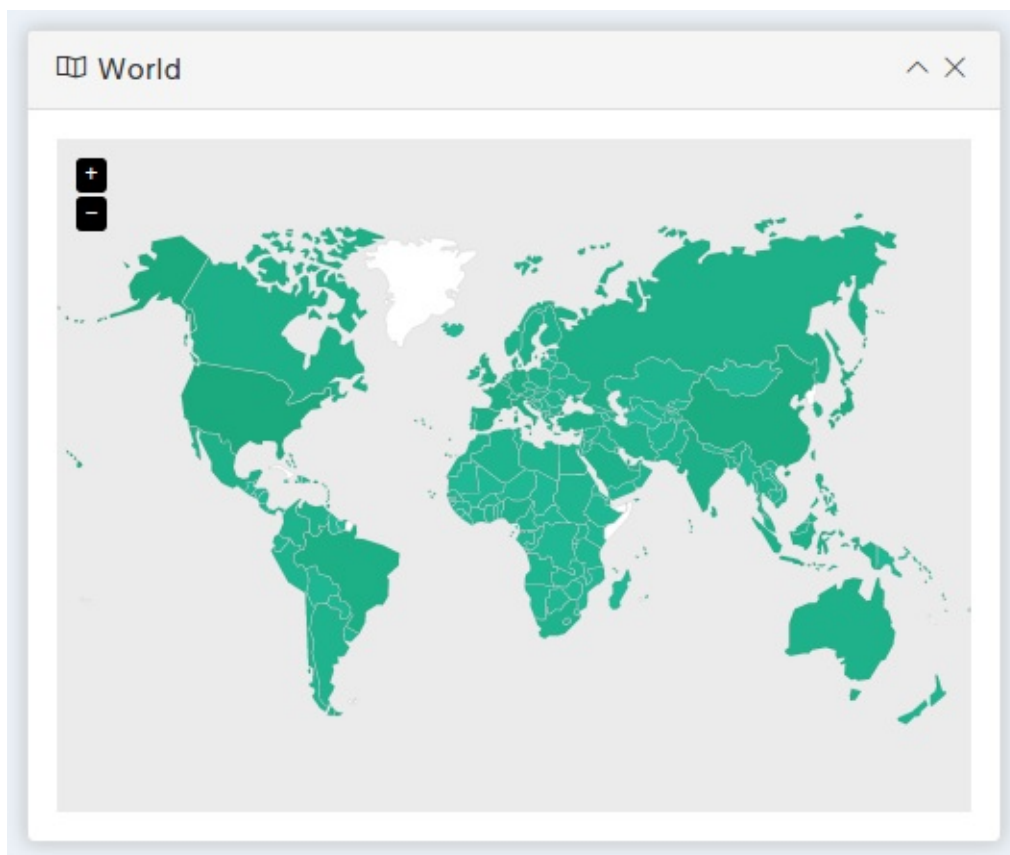
```
address : "nallakunta,hyderabad"
```

v) **Marker** - This feature allows the user to explicitly indicate the location. This option is available in the js file.

```
icon : "images/location-mark.png"
```

2) Vector Maps :

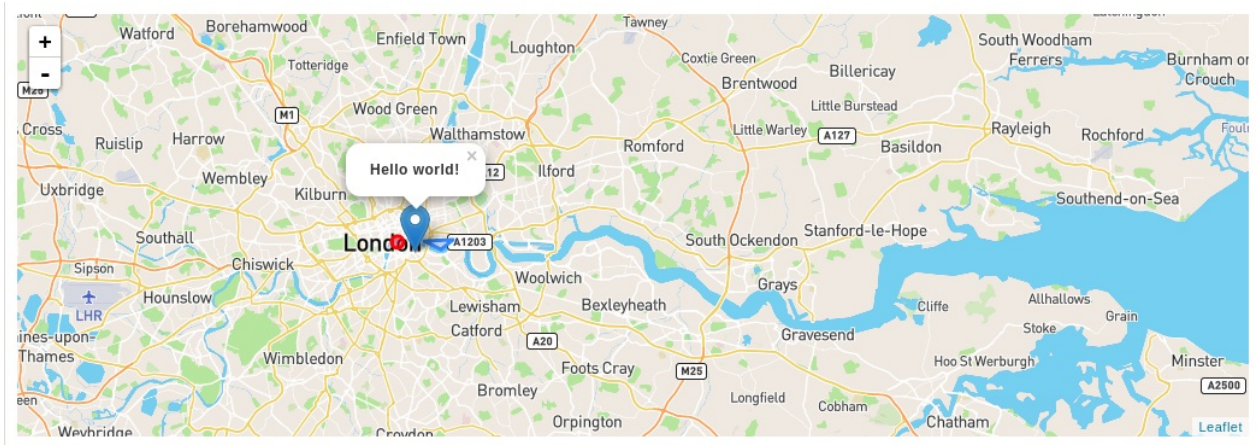
We can find different styles of this kind in the page [Vector Maps](#).



It has the following Structure

```
<div id="vmapworld" class="map_size">
  <svg>...</svg>
  ....
  ....
</div>
```

3) Advanced Maps :

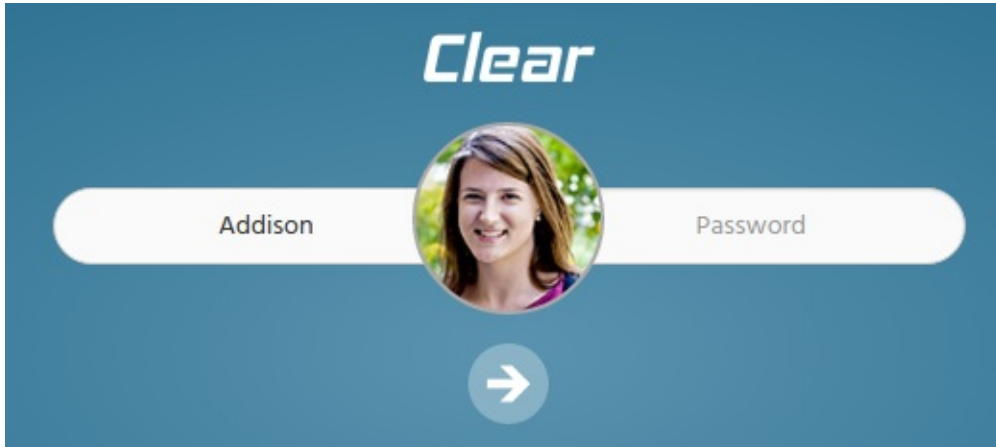


It has the following Structure

```
<div id="advanced_map" class="gmap leaflet-container leaflet-fade-anim leaflet-grab leaflet-touch-drag">
  ...
</div>
```

Lock Screen

This is the unique feature in this template which generally is designed for the Authentication purpose. This element can be found in `lockscreen` pages of this template.



It has the following Structure:

```
<div class="row">
  <div class="col-lg-12 login_border_radius1 lockscreen_img">
    ...
    ...
  </div>
  ...
  ...
</div>
```

Frequently Asked Questions ? (FAQ's)

1) Right Side Bar Activation?

A: The Menu icon with a badge in the navbar will activate the Right side bar.

2) Toggle Left Menu?

A: The Hamburger in the nav will collapse the left menu if required..

3) How does the preloader Work?

A: In Every page just below `<body>` you can find `<div>` with `preloader` class .If you want to change it you can get the css code in custom.scss file ,there exists a code commented between "preloader".

4) How to get rid of Preloader?

A: well, you can remove the preloader div at the start of every page and the related css in custom.scss and there is an on load function in Rightsidebar.js at the end of the file.

5) Data of Dimple Charts?

A: you can get the dimple chart data in "chart-data" folder which exists in the root folder.

6) How to use Layouts?

A: The code for each layout page is in the respective pageload css and js files. you can use that particular code globally to change the layouts over all pages.

7) How does the preloader work?

A: In Every page just below `<body>` you can find `<div>` with `preloader` class .If you want to change it you can get the css code in custom.scss file ,there exists a code commented between "preloader".

SCSS

SCSS File is located in the folder `src/sass/`. SCSS is a CSS preprocessor that adds many features like variables, mixins, imports, color manipulation, functions and tons of other powerful features.

The entire compiler comes in a single class file ready for including in any kind of project in addition to a command line tool for running the compiler from the terminal.

- `custom.scss` - consists of all the custom css code which is used to enhance look and feel of this template
- This is a custom css which includes in all the pages and it can be found in `css/` i.e `custom.css`.

CSS Files

We have a css file called app.css which is a combination of the following files:-

- bootstrap.css
- font-awesome.min.css
- vendors/themify-icons/css/themify-icons.css
- metisMenu.css
- bootstrap-switch.min.css

app.css is placed in the destination directory (css/app.css) which is done with the help of gulp. It is placed in all pages and also every page has its own css file which can be found in src/css/custom_css/.

Javascript Files

Js files are located in the folder js/. They are:-

- jquery.min.js
- jquery-ui.min.js
- bootstrap.min.js
- raphael-min.js
- metisMenu.js
- jquery.nicescroll.min.js
- holder.min.js
- bootstrap-switch.min.js
- leftmenu.js
- rightside_bar.js

The above files are further combined into a single file called app.js and it can be found in destination directory(js/) which is done with the help of gulp.

app.js is placed in all pages. And also every page has its own js file which can be found in src/js/pages/.

Plugins Used

Below is the list of plugins used in Clear.

Name	Description
jQuery	Core Javascript library
Bootstrap	Sleek, intuitive, and powerful front-end framework for faster and easier web development.
animate	A cross-browser library of CSS animations. As easy to use as an easy thing.
font-awesome	The iconic font and CSS framework.
themify-icons	Pixel-perfect, hand-crafted icons that draw inspiration from Apple iOS.
weather-icons	215 weather themed icons inspired by Font Awesome and ready for Bootstrap.
simple-line-icons	You will get the complete Simple Line Icons Package here.
bootstrap-sass	bootstrap-sass is a Sass-powered version of Bootstrap 3, ready to drop right into your Sass powered applications.
bootstrap3-wysihtml5-bower	Bootstrap compatible wysiwyg editor
twitter-bootstrap-wizard	This twitter bootstrap plugin builds a wizard out of a formatter tabbable structure. It allows to build a wizard functionality using buttons to go through the different wizard steps and using events allows to hook into each step individually.
Buttons	A fully customizable CSS button library built with Sass
fullcalendar	Full-sized drag & drop event calendar.
clockface.js	Clock-like timepicker for Twitter Bootstrap.
clockpicker	A clock-style timepicker for Bootstrap.
bootstrap-colorpicker	Simple and customizable colorpicker component for Twitter Bootstrap.
countUp.js	It Animates a numerical value by counting to it.
datatables.net	DataTables for jQuery.
datatables.net-bs	DataTables for jQuery with styling for Bootstrap.
datatables.net-buttons-bs	Buttons for DataTables with styling for Bootstrap.

datatables.net-responsive	Responsive for DataTables for Bootstrap.
datatables.net-colreorder-bs	ColReorder for DataTables with styling for Bootstrap.
datatables.net-rowreorder-bs	RowReorder for DataTables with styling for Bootstrap.
datatables.net-scroller-bs	Scroller for DataTables with styling for Bootstrap.
bootstrap-datepicker	A datepicker for twitter bootstrap.
bootstrap-daterangepicker	It provides date range picker component for Bootstrap.
eonasdan-bootstrap-datetimepicker	Date/time picker widget based on twitter bootstrap.
fancybox	Fancybox is a tool that offers a nice and elegant way to add zooming functionality for images, html content and multi-media on your webpages.
Flotchart	Attractive JavaScript charts for jQuery
gmaps	gmaps allows you to use the potential of Google Maps in a simple way. No more extensive documentation or large amount of code.
holderjs	Holder renders image placeholders on the client side using SVG.
iCheck	Highly customizable checkboxes and radio buttons.
jasny-bootstrap	You get the missing components for your favorite front-end framework.
jquery.inputmask	jquery.inputmask is a jquery plugin which create an input mask.
blueimp-file-upload	File Upload widget with multiple file selection, drag&drop support, progress bar, validation and preview images.
bower-jvectormap	jQuery interactive vector maps.
jquery-knob	Nice, downward compatible, touchable, jQuery dial.
bootstrap-magnify	Small bootstrap js plugin to enhance porte-folios and image galleries.
bootstrap-maxlength	This plugin integrates by default with Twitter bootstrap using badges to display the maximum lenght of the field where the user is inserting text. Uses the HTML5 attribute "maxlength" to work.
bootstrap-multiselect	Twitter Bootstrap plugin to make selects user friendly.
	Drag and drop hierarchical list with mouse and touch

	compatibility.
select2	Select2 is a jQuery based replacement for select boxes. It supports searching, remote data sets, and infinite scrolling of results.
select2-bootstrap-theme	It gives a Select2 Theme for Bootstrap.
selectize	Selectize is a jQuery-based custom <select> UI control. Useful for tagging, contact lists, country selectors, etc.
sparkline	A lightweight plugin for the jQuery javascript library to generate small sparkline charts directly in the browser.
summernote	Summernote is a super simple WYSIWYG Editor on Bootstrap.
bootstrap-switch	It turns checkboxes and radio buttons into toggle switches.
toastr	Toastr is a Javascript library for non-blocking notifications. It is used to create simple core library that can be customized and extended.
bootstrap-touchspin	Bootstrap TouchSpin is a mobile and touch friendly input spinner component for Bootstrap.
trumbowyg	A lightweight and amazing WYSIWYG JavaScript editor.
bootstrapvalidator	The best jQuery plugin to validate form fields. Designed to use with Bootstrap.
x-editable	It provides in-place editing with Twitter Bootstrap, jQuery UI or pure jQuery.
jqvmap	jqvmap provides jQuery Vector Map Library.
metisMenu	It is a jQuery menu plugin with customised styles.
flot-spline	Flot plugin that provides spline interpolation for line graphs.
flot.tooltip	Tooltip for wonderful Flot plotting library.
blueimp-gallery-with-desc	blueimp Gallery is a touch-enabled, responsive and customizable image and video gallery, carousel and lightbox, optimized for both mobile and desktop web browsers. It features swipe, mouse and keyboard navigation, transition effects, slideshow functionality, fullscreen support.
jquery-mockjax	The jQuery Mockjax Plugin provides a simple and extremely flexible interface for mocking or simulating ajax requests and responses.
jquery-ui	Shim repository for jQuery UI.
jquery-selectric	Fast, simple and light jQuery plugin to customize HTML selects.
wow	It Reveal CSS animation on page scroll down.

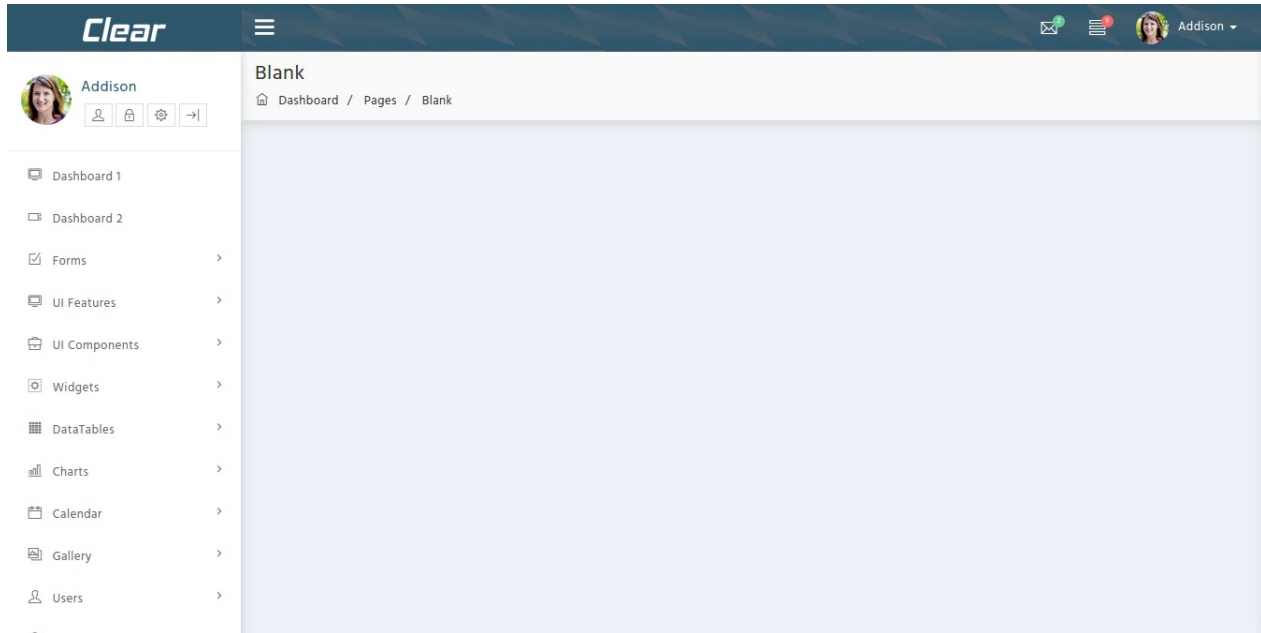
wow	It Reveal CSS animation on page scroll down.
moment	Moment is used to Parse, validate, manipulate, and display dates in javascript.
c3	It is a D3-based reusable chart library.
hover	A collection of CSS3 powered hover effects to be applied to links, buttons, logos, SVG, featured images and so on. Easily apply to your own elements, modify or just use for inspiration. Available in CSS, Sass, and LESS.
bootstrap-fileinput	An enhanced HTML 5 file input for Bootstrap with file preview, multiple selection, ajax uploads, and more features.
CoutUpCircle	jQuery Plugin that counts up a given number with Animation.
nvd3	These are Re-usable charts and chart components for d3.
chartist	It provides Simple responsive charts
jquery-date-range-picker	jQuery Date Range Picker is a jQuery plugin that allows user to select a date range.
datedropper	Datedropper is a jQuery plugin that provides a quick and easy way to manage dates for input fields.
timedropper	Timedropper is a jQuery UI timepicker. Manage time input fields in a standard form. Focus on the input to open a small interactive timepicker.
toolbar	A tooltip style toolbar jQuery plugin.
circliful	It is a jQuery circle statistic plugin.
datetimepicker	Using this plugin, you can add a datetimepicker, datepicker or timepicker dropdown to your forms.
air-datepicker	A lightweight, customizable, powerful jQuery datepicker.
awesome-bootstrap-checkbox	Font Awesome Bootstrap Checkboxes & Radios. Pure css way to make inputs look prettier.
dropify	dropify Overrides your input files with style.
gridforms	Data entry can be beautiful with gridforms.
pnotify	JavaScript notification plugin.
granlm.js	Create fluid and interactive gradients animations with this small (< 10 kB) js library.
prettyCheckable	A jQuery plugin to replace checkboxes and radios for custom images
bootstrap-calendar	Full view calendar with year, month, week and day views based on templates with Twitter Bootstrap.

underscore	provides support for the usual functional suspects without extending any core JavaScript objects.
jquery-labelauty	A lightweight and beautiful jQuery plugin for radio and checkbox inputs.
dimple	An object-oriented charting API for d3
imagehover	Pure CSS Image Hover animation and transition Effects Library
editable	Tiny jQuery/Bootstrap widget that makes a HTML table editable.
amcharts	Programmed libraries and tools for all your data visualization needs with javascript charts and maps.
bootstrap-table	An extended Bootstrap table with radio, checkbox, sort, pagination, and other added features.
tableExport.jquery.plugin	jQuery plugin to export a html table to JSON, XML, CSV, TXT, SQL, Word, Excel, PNG, PDF.
mark.js	Highlight keywords using JavaScript. Intended for every use case. Can e.g. be used to mark text in search results.
datatables.mark.js	A DataTables plugin for mark.js to highlight search terms in tables.
gridstack	gridstack.js is a jQuery plugin for widget layout.
ladda-bootstrap	Buttons with built-in loading indicators, effectively bridging the gap between action and feedback.
sweetalert2	A replacement for JavaScript's popup boxes, supported fork of sweetalert.
insignia	Customizable tag input. Progressive.
leaflet	JavaScript library for mobile-friendly interactive maps.
cssgram	It is a CSS library for Instagram filters.
jquery-advanced-news-ticker	A powerful, flexible and animated vertical news ticker plugin for JQuery.
swiper	Most modern mobile touch slider and framework with hardware accelerated transitions.
ihover	iHover is a collection of hover effects using pure CSS, inspired by codrops article, powered by Sass.
jquery.nicescroll	Nicescroll is a jquery plugin, for nice scrollbars with a very similar ios/mobile style.
flip	A lightweight jQuery plugin to create 3d flip animation.
metrojs	It is a plugin to help make Metro interfaces on the web. Example features include Live Tiles that can fade, flip, slide

metrojs	at custom points or in a carousel.
screenfull.js	Simple wrapper for cross-browser usage of the JavaScript Fullscreen API.
LC-switch	Superlight jQuery plugin improving switch buttons for forms look and functionality.
card	Card let's you add an interactive, CSS3 credit card animation to your credit card form to help your users through the process.
wenk	Lightweight tooltip for the greater good with only css.

Blank Page

The design for Blank Page is shown below:

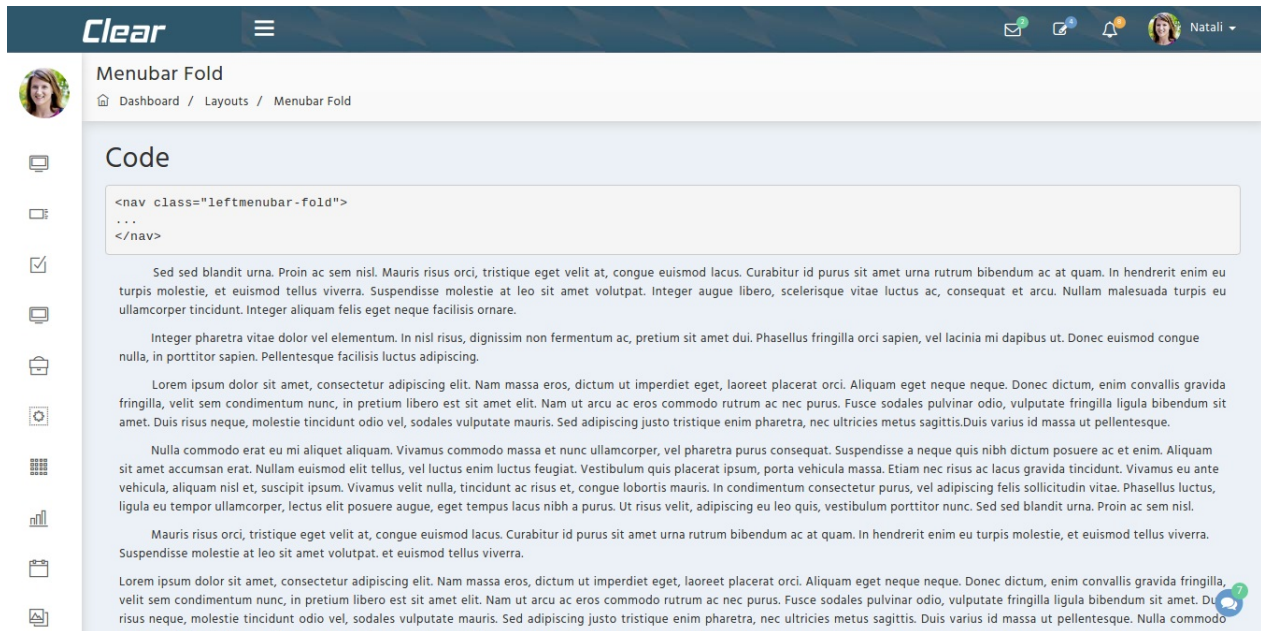


It has the following Structure:

```
<body class="skin-default">
  <div class="preloader">
    ...
  </div>
  <header class="header header_movable">
    ...
    ...
  </header>
  ...
  ...
</body>
```

Menubar Fold

The design for Menubar Fold is shown below:

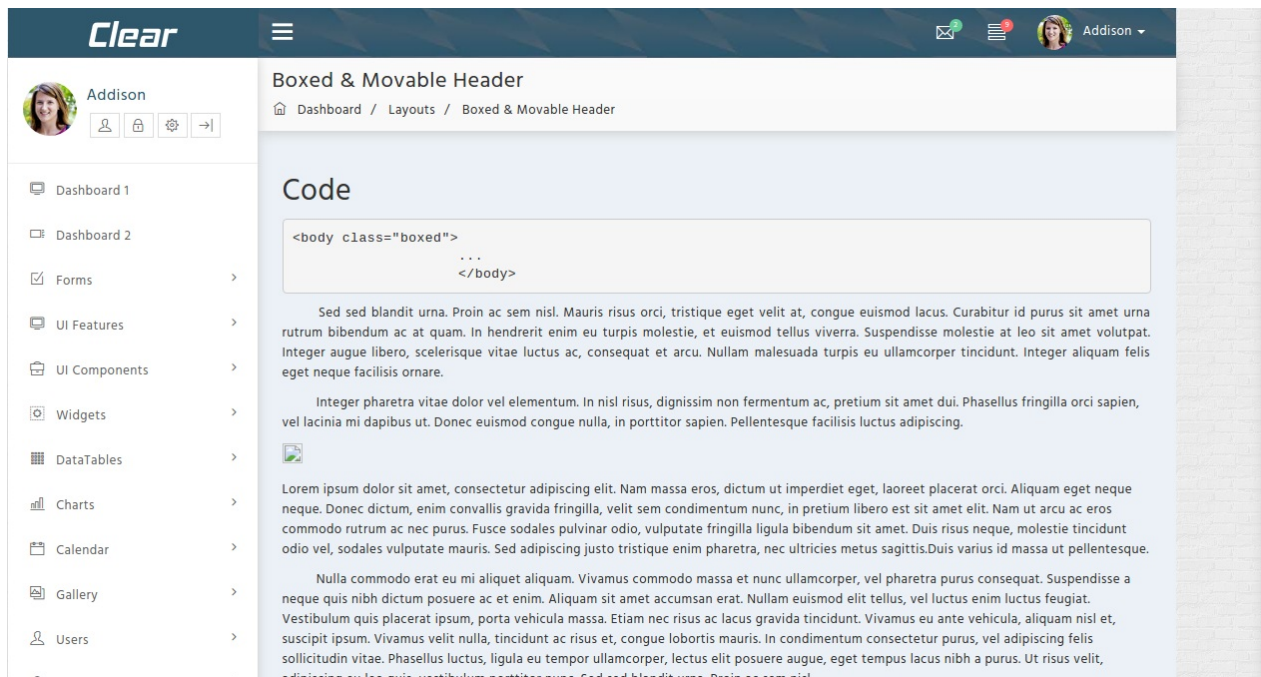


It has the following Structure:

```
<body class="skin-default">
  <div class="preloader">
    ...
  </div>
  <header class="header">
    <nav class="navbar navbar-static-top" role="navigation">
      ...
    </nav>
  </header>
  <div class="wrapper row-offcanvas row-offcanvas-left">
    <aside class="left-side sidebar-offcanvas">
      ...
    </aside>
    ...
  </div>
</body>
```

Boxed and Moveable Header :

The design for Boxed and Moveable Header is shown below:

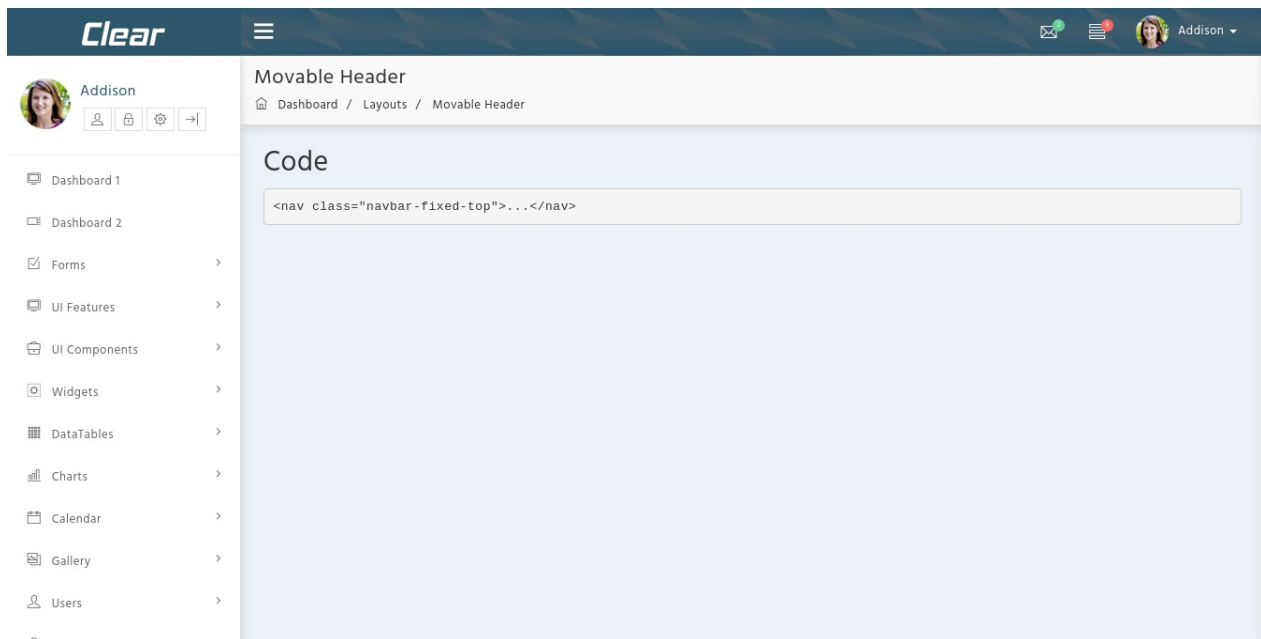


It has the following structure:

```
<body class="skin-default boxed movable-header">
<div class="preloader">
    ...
    ...
</div>
<header class="header header_movable">
    <nav class="navbar navbar-static-top" role="navigation">
        ...
    </nav>
</header>
<div class="wrapper row-offcanvas row-offcanvas-left">
    ...
</div>
</body>
```


Moveable Header :

The design for Menubar Fold is shown below:

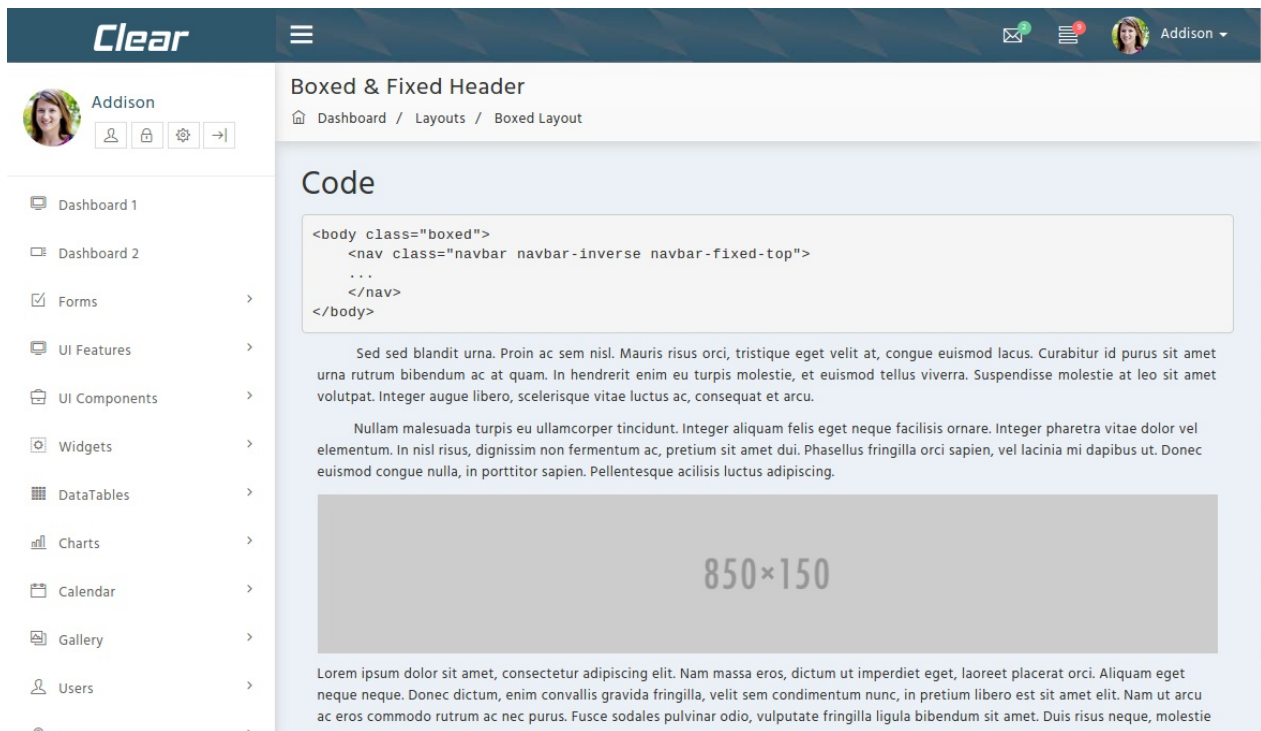


It has the following structure:

```
<body class="skin-default movable-header">
<div class="preloader">
    ...
    ...
</div>
<header class="header header_movable">
    <nav class="navbar navbar-static-top" role="navigation">
        ...
    </nav>
</header>
<div class="wrapper row-offcanvas row-offcanvas-left">
    ...
</div>
</body>
```

Boxed & Fixed Header

It has the following design:

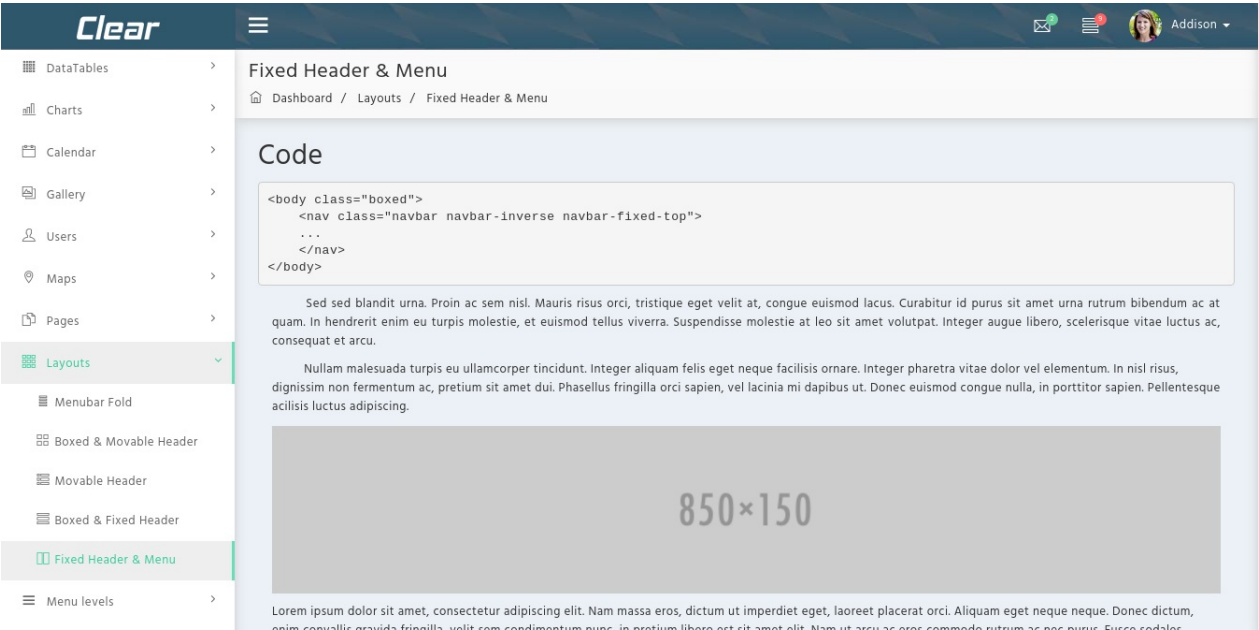


It has the following Structure:

```
<body class="skin-default boxed fixed-menu">
<div class="preloader">
    ...
</div>
<header class="header header_movable">
    <nav class="navbar navbar-fixed-top" role="navigation">
        ...
    </nav>
</header>
<div class="wrapper row-offcanvas row-offcanvas-left">
    ...
</div>
</body>
```

Fixed Header & Menu

It has the following design:



It has the following design :

```
<body class="skin-default fixed-menu">
<div class="preloader">
    ...
</div>
<header class="header">
    <nav class="navbar navbar-fixed-top" role="navigation">
        ...
    </nav>
</header>
<div class="wrapper row-offcanvas row-offcanvas-left">
    ...
</div>
</body>
```

This Document takes you briefly through the description about features that are included the Clear_vue Version.

Basic installation :

The Clear package comes with compiled assets. To view the project place the package contents in the root directory of your server or point a virtual host to the project directory and every thing should be working out of the box.

Note: To setup a virtual host follow this [guide](#).

And once you are ready to customize the project follow the below steps.

Follow below steps before you start customizing clear VueJS version.

Note :

Make sure you have [nodejs](#) installed in your system with minimum version `6.10.0` ,

you can check installed version by running `node -v` command in terminal.

If you are having older version, please install latest version from [nodejs.org](#)

1) Install Dependencies

Run the below command to get all the dependencies from package.json

```
npm install
```

2) Install Assets

Run the below command to get the plugins from bower.json

```
bower install
```

Note 1: If you logged in as `root` user you may need to run `bower install --allow-root`

Note2: We have removed bower in 4.2 version , so if you are using 4.2 or above version please ignore this step.

3) File Compilation

Run the following command to compile all the files.

```
npm run dev
```

now your vuejs version is ready.

Development tips

You need to run `npm run dev` everytime you want your changes to be applied but you can follow below tips to save time

File Compilation with watchers

Run the following command to set watchers for files so that they compile as soon as you save any changes

```
npm run watch
```

serve with hot reload at localhost:8080

The following command will create a local server at localhost:8080 with hot-module-replacement enabled, to reflect your changes without effecting the state of your application.

```
npm run hot
```

File compilation for production

Run the following command to compile all files with minification for production.

```
npm run production
```

Note: when using "dev" "watch" or "production" make sure that your application is in the root directory of the server.

If your files are being server from a sub folder in the server like `localhost/vue` then consider using the laravel-mix `mix.setResourceRoot('/');` option.

if you are serving the files like `localhost/vue` then set `mix.setResourceRoot('/vue/');`

Eventhough you can run `npm install` we suggest you to run `yarn install` since the package contains `yarn.lock` file which lets you get the exact version of packages that we have used.

If you are using version 1.7 of clear, Run the below commands

1) Install Dependencies

Run the below command to get all the dependencies from package.json

```
yarn install
```

2) Install Assets

Run the below command to get the plugins from bower.json

```
bower install
```

3) serve with hot reload at localhost:8000

The following command will launch a browser window with localhost:8000 with hot module replacement. To change the default port , edit the port in `config.index.js`

```
npm run dev
```

4) build for production with minification

The following command will build the project into the `dist` folder. Built files are meant to be served over an HTTP server.

Opening index.html over file:// won't work.

```
`npm run build`
```

Basic Structure :

index.html

This is the starting point of the project. It contains the vue app instance.

app.vue

This is the component that gets loaded into the index.html when vue app is started.

`app.vue` also contains preloader component and route-view to load basic pages and the main layout.

layout.vue

This gets loaded into app.vue when main paths are accessed.

This contains the main layout including the header, left-menu and the right-side content including the right-side menu.

main.js

This is the starting of the vue app. The vue app is instantiated at the end of this file. This file also contains the routes of the app to which it should re-direct and is controlled by the vue-router. The vue-router configuration is also done in this page.

Layout.vue

This contains the main layout which includes the header, left-menu and the right-side component. The right-side component contains the right-side bar.

importing components :

The components can be imported using the keyword `import`

eg:

```
import clear_header from "../components/layout/clear_header";
```

The component which is being imported is to declared in the components object:

eg:

```
components: {clear_header, left_side, right_side}
```

The component exporting is done using export default

eg:

```
`export default { ... }
```

,

Left Menu Structure :

It has the following structure:

The links in left menu has the following structure:

```
<router-link to="/user" tag="li" exact>
  <a class="logo"><i class="menu-icon ti-desktop"></i><span class="mm-text">Dashboa
rd 1</span></a>
</router-link>
```

this will render to:

```
<li class="">
  <a href="#/user" class="logo">
    <i class="menu-icon ti-desktop"></i><span class="mm-text">Dashboard 1</span>
  </a>
</li>
```

Start New Page

Open the blank.vue file in `/src/components` folder.

It has the following structure:

```
<template>
  <div>
  </div>
</template>
<script>
import Vue from "vue";

export default {
  // ===Component name
  name: "blank",
  // ===Components used by this component
  components: {},
  // ===Code to be executed when Component is mounted
  mounted() {},
  // ===Computed properties for the component
  computed: {},
  // ===Component data
  data() {
    return {

    }
  },
  // ===Component methods
  methods: {

  }
}
</script>
<!-- styles -->
<!-- adding scoped attribute will only apply the css to this component only -->
<style scoped></style>
```

Place your html content inside the div in template.

Note: Each template must only have one root element.

Then give the name of your component in the name value and your javascript in the appropriate lifecycle hook.

Then give the appropriate route in main.js routes section. If needed add the component path to left-side menu.

To Change Layouts:

To change the layouts to your desired one involves changing names in the routes.js file

1.Default Layout

To get the default layout modify line as below.

```
const routes = [{  
  path: '/',  
  component: resolve => require(['./default.vue'], resolve),  
}]
```

2.Fixed Header Layout

To get the fixed header layout modify line as below.

```
const routes = [{  
  path: '/',  
  component: resolve => require(['./fixed_header.vue'], resolve),  
}]
```

3.Fixed Header and Fixed Menu Layout

To get the fixed header and fixed menu layout modify line as below.

```
const routes = [{  
  path: '/',  
  component: resolve => require(['./fixed_header_fixed_menu.vue'], resolve),  
}]
```

4.Fixed Menu Layout

To get fixed menu layout modify line as below.

```
const routes = [{  
  path: '/',  
  component: resolve => require(['./fixed_menu.vue'], resolve),  
}]
```

5.Horizontal Layout

To get Horizontal menu layout modify line as below.

```
const routes = [{
  path: '/',
  component: resolve => require(['./horizontal.vue'], resolve),
}]
```

6.Menu Bar Folded Layout

To get Folded menu bar layout modify line as below.

```
const routes = [{
  path: '/',
  component: resolve => require(['./menubar_fold.vue'], resolve),
}]
```

7.Boxed Layout

To get boxed layout modify line as below.

```
const routes = [{
  path: '/',
  component: resolve => require(['./boxed.vue'], resolve),
}]
```

Note: There are different styles of header and menu layout files are available.

- Default layout
- Fixed Header layout
- Fixed Header and menu layout
- Fixed menu
- Horizontal layout
- Menu bar folded layout
- Boxed layout

Changing Color Scheme :

The default color scheme is dark, to change it to white scheme, we need to perform the following steps:

1) In src/layout.vue, change the following lines

```
@import ../assets/sass/custom.scss
```

to

```
@import ../assets/sass/custom_white.scss
```

And

```
import right_side from "../components/layout/right-side";
```

to

```
import right_side from "../components/layout/right-side_white";
```

To, change the logo of your site, you can edit the img tag in **clear_header.vue**

```

```

This Document takes you briefly through the description about Installation of Clear Admin Template skin for Laravel Spark .

Spark Installation

Make sure you have [nodejs](#) installed in your system with minimum version `6.10.0` ,

you can check installed version by running `node -v` command in terminal.

If you are having older version, please install latest version from [nodejs.org](#)

For laravel 5.4 , the minimum version of php required is `5.6.4` and

the following php extensions are required

- PHP `>= 5.6.4`
- OpenSSL PHP Extension
- PDO PHP Extension
- Mbstring PHP Extension
- Tokenizer PHP Extension
- XML PHP Extension

from 5.4 onwards, Laravel team decided to move to webpack from gulp

so assets compilation differs a bit.

They introduced a new npm package for webpack called `mix`

you can read more about it [here](#)

install local packages

```
yarn install
```

move assets to public

```
npm run dev
```

1) Installation of spark :

Initially, you need to Install the Laravel Spark, For the installation of Laravel spark, Please follow the official guide, the link for which is provided below

<https://spark.laravel.com/docs/4.0/installation>.

2) Applying the Clear skin:

After completing the installation of laravel spark, follow the below steps to apply the Clear skin

1. Remove node modules by running the following commands in your project root folder

```
rm -rf node_modules
```

2. Extract the files from the provided zip file.
3. copy all the files into your spark installation folder/directory.
4. Run `yarn install`
5. Run `npm run dev` or `npm run production` or `npm run watch` as per your requirement.
6. once, all the above steps are performed you will get a new look for your default spark website.

Note:

Eventhough you can run `npm install` we suggest you to run `yarn install` since the package contains `yarn.lock` file which lets you get the exact version of packages that we have used.

currently, we are providing support for spark version **v4.0.8**

Using .vue Files in spark

To use a .vue file from clear follow these steps:

- Copy the required .vue file from clear to "resources/assets/js/components" directory.
- Also copy the necessary css and js files to their respective folders.
- Then register the component in the "resources/assets/js/app.js" file.

eg:

```
var app = new Vue({
  mixins: [require('spark')],
  components: {
    'invoice': require('./components/clear.vue')
  }
});
```

- Then use the component in any of the blade files by using the component tag

```
<invoice></invoice>
```

Note:

- The links to the vendors css files and images should be adjusted to the new directory structure.

Adding External Plugins from Clear

To use plugins used in clear follow the below steps.

we are going with `select2` plugin which is already added to spark skin but we explain the steps here so you can follow it for other plugins aswell.

Install Plugins

first copy `select2` and `select2BootstrapTheme` packages from clear package.json and paste into spark package.json

Run `yarn install` which will copy those packages into `node_modules` folder

Check if the packages are downloaded to `node_modules` directory

Copying assets to public

- In `webpack.mix.js` use `mix.copy` to copy the plugins you needed to `public` directory.

First copy the vendor direcorey path in the `paths` object

```
var paths = {  
  'select2': vendors + 'select2/dist/',  
  'select2BootstrapTheme': vendors + 'select2-bootstrap-theme/dist/'  
};
```

- you can copy the path and the `mix.copy` from clear's `gulpfile.js`

```
mix.copy(paths.select2 + 'css/select2.min.css', destVendors + 'select2/css');
```

Also copy the necessary css files to `resources/assets/css` directory and js files to `resources/assets/js/pages` directory.

- then run `npm run dev` to copy the plugin to the `public/vendors` directoy.

Usage of Plugins

Use the links, HTML, CSS and js from clear HTML version in your view to get the desired plugin or widget.

The zip file contains all laravel files integrated with Clear, however you need to perform following steps to get vendors etc.

Get Composer packages

```
composer install
```

permissions

```
chmod -R 775 storage

chmod 775 bootstrap/cache
```

If you are on linux/ mac you can run below command to chown it.

```
chown -R www-data /var/www
```

database credentials

open `.env` and modify database details with yours

compile assets

If you don't have good knowledge on nodejs and npm, you can copy public folder files from codecanyon's downloaded files

Note : The above step is completely optional , if you are not comfortable with this, you can skip it and perform the below steps, still it works fine.

Make sure you have [nodejs](#) installed in your system with minimum version `7.0.0` ,

you can check installed version by running `node -v` command in terminal.

If you are having older version, please install latest version from [nodejs.org](#)

For laravel 5.5 , the minimum version of php required is `7.0.0` and

the following php extensions are required

- PHP >= 7.0.0
- OpenSSL PHP Extension
- PDO PHP Extension
- Mbstring PHP Extension
- Tokenizer PHP Extension

- XML PHP Extension

from 5.5 onwards, Laravel team decided to move to webpack from gulp

so assets compilation differs a bit.

They introduced a new npm package for webpack called `mix`

you can read more about it [here](#)

install local packages

```
yarn install
```

get bower components

```
bower install
```

move assets to public

```
npm run dev
```

Note: If bower gives warning that you are running as administrator, please run the command `bower install --allow-root`

Changing Skin of Clear :

To change the skin of clear from dark to light

- Comment the `mix.sass` line for compiling `custom.scss`
- Comment out the `mix.sass` line for compiling `custom_white.scss`

Changing Layouts of Clear :

To change any layout you just go through your file and in that you find the code

```
@extends('layouts/default')
```

Modify the above line as

```
@extends('layouts.menu_barfold')
```

or

```
@extends('layouts.mini_sidebar')
```

or

```
@extends('layouts.layout_movable_header')
```

or

```
@extends('layouts.layout_fixed')
```

or

```
@extends('layouts.layout_boxed_fixed_header')
```

or

```
@extends('layouts.horizontal_menu')
```

or

```
@extends('layouts.boxed_movable_header')
```

Note: There are different styles of header and menu layout files are available.

- Default Layout
- Menu_barfold Layout

- Mini Sidebar Layout
- Movable Header Layout
- Fixed Layout
- Boxed and Fixed Header Layout
- Horizontal Menu Layout
- Boxed Movable Header Layout