"LightShops" Documentation by "Nick Paul" v0.1.1

"LightShops"

Created: 12/09/2021
By: Nick Paul
Email: pianorockcover@gmail.com

Thank you for purchasing my theme. If you have any questions that are beyond the scope of this help file, please feel free to email via my user page contact form here. Thanks so much!

Table of Contents

A) File Structure - top

There is a theme file structure

- **css/** - contains compiled css files
- **images/** - contains images and icons (*.jpg, *.png and *.svg formats)
- **js/** - contains compiled JavaScript file
- ***.html** - any theme pages

B) HTML Structure - top

This theme is based on Bootstrap Grid System and you feel free to use all of its features like containers, rows and columns. The general template structure is the same throughout the template.

```html
1   <!DOCTYPE html>
2   <html lang="en">
3
4     <!-- BEGIN METATAGS -->
5     <head>
6       <meta name="viewport" content="width=device-width, initial-scale=1.0, minimum-scale=1.0">
7       <link rel="stylesheet" href="css/style-0.0.7.css">
8       <!-- ... -->
9     </head>
10    <!-- END METATAGS -->
11
12    <body>
13      <!-- BEGIN SIDEBAR -->
14      <div class="sidebar" id="catalogSidebar">
15        <!-- ... -->
16      </div>
17      <!-- END SIDEBAR -->
18
19      <!-- BEGIN HEADER -->
20      <div class="area">
21        <div class="header">
22          <!-- ... -->
23        </div>
24      </div>
25      <!-- END HEADER -->
26
27      <!-- BEGIN CONTENT -->
28      <div class="area">
29        <div class="section">
30          <div class="container-full">
31            <div class="section-title">
32              <!-- ... -->
33            </div>
34            <div class="row">
35              <div class="col"></div>
36              <!-- ... -->
37            </div>
38          </div>
39        </div>
40      </div>
41      <!-- END CONTENT -->
42
43      <!-- BEGIN FOOTER -->
44      <div class="area">
45        <footer class="footer">
46          <!-- ... -->
47        </footer>
48      </div>
49      <!-- END FOOTER -->
50
51      <!-- INCLUDE JAVASCRIPT FILES HERE -->
52      <script src="js/main-0.0.7.js"></script>
53    </body>
```

Common html structure contains

1. **Meta information** between <header>...</header> tags. Don't forget to include the LightShops style.css file and vieport metatag to provide responsiveness
2. **Sidebar** includes links to catalog pages, sales, goods and other pages. The rest of the page elements must be wrapped by element with `.area` class, so as not to be not overlapped by sidebar.
3. **Header** section includes logo, profile links, catalog search field and etc (i.g shopping cart, it depends on template).
4. **Content** part must also be wrapped by `.area` element and may contain several `.section` tags. Elemets with `.section` class provide top and bottom margins. You also should place `.container-full` element inside section to provide left and right paddings. Use `.section-title` with any header tags (like h1, h2, h3, h4...) to get well-styled section title. Inside `.container-full` element you can use Bootstrap Grid System elements (row and cols) to organize your content.
5. **Footer** section duplicates header information with some additional links, social networks and contacts.
6. **JavaScript** files must be included at the bottom of <body> tag. Don't forget to include the LightShops main.js file.

C) CSS Files and Structure - [top](top)

All css contents (including some helpful npm packages and own LiteShops styles) are compressed, minified and saved in single stylesheet file called `css/style-<version>.css` in order to provide better website loading speed. You feel free to override any css rules just placing your code at the end of this file or in another *.css files which should be included AFTER LiteShops stylesheet. But I

highly recommend you to use npm pakages like gulp, grunt or webpack to compile your css and put in to a single file before you deploy your project to production to archieve better perfomance.

There is a list of npm packages which are used in LiteShops

- [bootstrap](#) (Grid only)
- [animate.css](#)
- [bootstrap-icons](#)
- [normalize.css](#)

CSS classes naming rules

There are two simple rules used in LiteShops to name css-classes and organize them in html structure.

1. Divide page contents into small components like (product, article thumbnail, slider, sale banner etc.) and reuse them throughout all theme pages
2. Name of children elements must be prefixed with main component class. For example, there is `.product` component that visualize a single product in catalog. All product children element names should fit the pattern `.product-*`, like `.product-title` or `.product-tags`. See the example image below.

```html
<div class="product">
    <div class="product-content">
        <div class="product-labels">
            <!-- ... -->
        </div>
        <div class="product-image-wrapper">
            <!-- ... -->
        </div>
        <div class="product-details">
            <div class="product-price">
                <!-- ... -->
            </div>
            <div class="product-tags">
                <!-- ... -->
            </div>
            <div class="rate">
                <div class="rate-stars">
                    <!-- ... -->
                </div>
            </div>
        </div>
    </div>
</div>
```

Note that components can include each other and in that case nested components may not have a prefix in order to reuse them in other parts of the page. For example, in the previous image there is `.rate` component (customers rating) that may be used not only in products but also in customers feedback and etc. So `.rate` is a separate component and its class name not been prefixed with `.product-*`

D) JavaScript - [top](#)

Like stylesheets all JavaScript files (including npm packages and own LiteShops scripts) are compressed, minified, compiled with [babel](#) and saved into a single `js/main-<version>.js` file. All of own LiteShops scrips are written using JQuery. Almost all functionality is provided by LiteShops own scrips except sliders and image galleries.

There is a list of JavaScript npm packages that are used in LiteShops

- [jquery](#)
- [ion-rangeslider](#)
- [swiper](#)
- [simplelightbox](#)

Make things interactive

There are several ready-to-use data tags which can make elements on your page appear and disappear after a mouse click or hover. Look at the example below.

```
1   <div class="filters-view-all" data-toggle="#vendowFilters" data-trigger="click" data-effect="slide">
2       Show filters
3   </div>
4   <div class="hidden" id="vendowFilters">
5       <!-- ... -->
6   </div>
```

There is [data-toggle] attribute which indicates that all elemets that fit the selector in its value will be shown or hidden by clicking on the element. There are also some optional attributes that can customize this script behavior. [data-trigger] sets up the trigger (click or mouse hover) and [data-effect] an animation effect (fade or slide).

Show modals

To open a modal when clicking on a button or other element of a web page, you need to add an attribute [data-modal] containing a modal css selector. (Such as modal ID like in the example below)

```
1   <div class="product-image-wrapper" data-modal="#productViewModal">
2       <!-- ... -->
3   </div>
4   <div class="modal" id="productViewModal">
5       <!-- ... -->
6   </div>
```

Once again, thank you so much for purchasing this theme. As I said at the beginning, I'd be glad to help you if you have any questions relating to this theme.

Nick Paul