



PREMIUM SITE TEMPLATES



## “LauncherX”

# Premium Template By iDangero.us Documentation

Thank you for purchasing this item. If you have any questions that are beyond the scope of this help file, please feel free to contact us via support ticket form [here](#).

# Table of Contents

---

1. About “LauncherX”
2. LauncherX Files & Folders Structure
3. Installation
4. HTML Structure
5. Typography
6. Inner Linking (Popup Window)
7. CSS Files and Color Themes
8. PSD Files and Structure
9. Customization (Change Logo, Backgrounds, etc)
10. JavaScripts
11. “Send Email” Script & Ajax Contact Form
12. Licensing Terms

# 1. About “LauncherX”

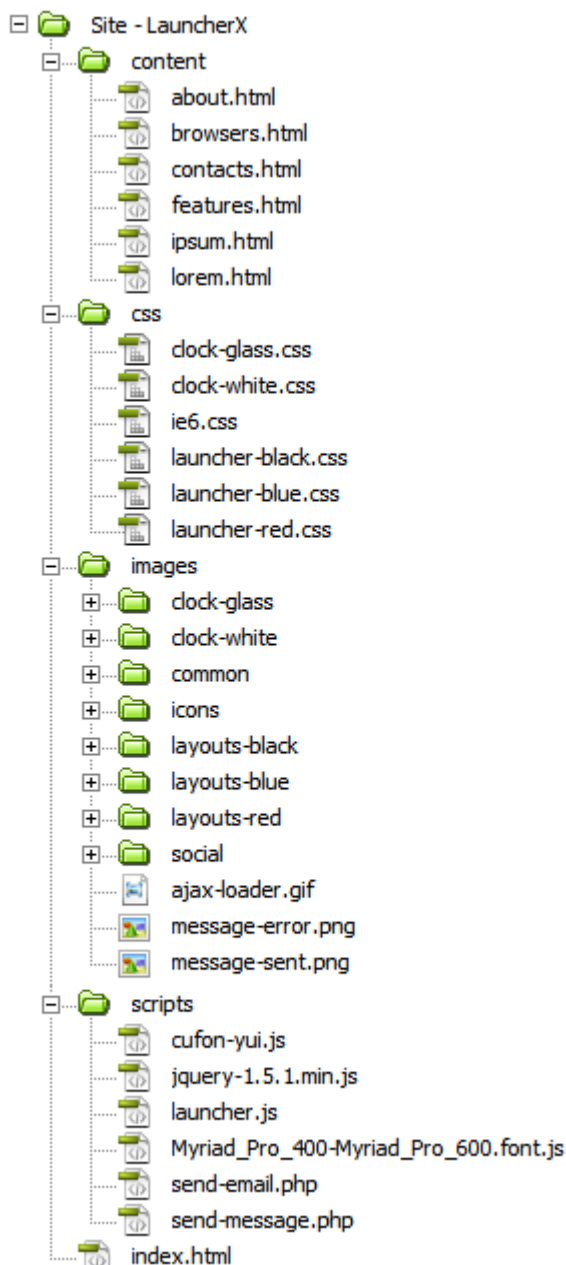
---

**LauncherX** is the premium Ajax based countdown template designed by iDangero.us. It is designed for temporary usage until the main site is under construction. It comes with an awesome animated clock with a countdown to the grand opening date, brief information about the main site (or company) and Ajax Contact form with validation. Also, there is an opportunity for visitors to leave their email address to be informed when the site will open.

This template is easy to redesign, it has very simple and clear API.

## 2. LauncherX Files & Folders Structure

---



**LauncherX** files are located in the **launcher-x/** folder from the downloaded archive:

**content:** in this folder there are .html files with content pages.

**css:** folder with a .css files for different color themes and clock skins.

**images:** here you can find layouts images and icons used in this Template and for other color themes.

**scripts:** JavaScripts and PHP scripts used in this template

And the main **index.html** in the root folder

## 3. Installation

To install **LauncherX** template you have to copy all files & folders from the **launcher-x/** folder to your web-site root folder on the server (It could be a real server or local testing server).

## 4. HTML Structure

Let's look at the main HTML structure of the LauncherX template. Open index.html file with a text editor (Highly recommended to use Adobe® Dreamweaver® or Notepad ++ editors).

**Index.html** structure:

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <link rel="stylesheet" type="text/css" href="css/launcher-black.css" />
6 <link rel="stylesheet" type="text/css" href="css/clock-glass.css" />
7 <script type="text/javascript" src="scripts/jquery-1.5.1.min.js"></script>
8 <script type="text/javascript" src="scripts/cufon-yui.js"></script>
9 <script type="text/javascript" src="scripts/Myriad_Pro_400-Myriad_Pro_600.font.js"></script>
10 <script type="text/javascript" src="scripts/launcher.js"></script>
11 <script type="text/javascript">
12     Cufon.replace('.cufoned');
13 </script>
14 <!--[if IE 6]>
15 <link rel="stylesheet" type="text/css" href="css/ie6.css" />
16 <script type="text/javascript">
17 isIE6=true;
18 </script>
19 <![endif]>
20 <title>LauncherX</title>
21 </head>
22 <body class="black">
23 <div id="background"></div>
24 <div id="wrapper">
25     <div id="logo"></div>
26     <div class="clock">
27         <h1 class="will-open">We will open after</h1>
28         <div class="clock-inner">
29             <div class="divider">:</div>
30             <div class="days">
31                 <p>Days</p>
32                 <div class="d-bg">
33                     <div class="d-wrap">
34                         <div class="day" style="top:0px;"></div>
35                     </div>
36                     <div class="mask3"></div>
37                 </div>
38     </div>
```

CSS and JavaScript files attachment

All elements with "cufoned" class will be replaced with Myriad Pro font

CSS style sheet file used for IE6 browser only

And if the visitor's browser is IE6 then the **isIE6** variable will be equal to **true**

Site Title

Section with an additional big clock background

Wrapper div contains whole site content

clock section contains the countdown clock

Blinking colon

Div block with a "days" class contains a number of days to the target date

Number of days will be placed here by JavaScript.

Mask layer is used to simulate obfuscation in gradient white clock skin

Div block with a "hours" class contains a number of hours to the target date

Number of hours will be placed here by JavaScript.

Mask layer is used to simulate obfuscation

Div block with a "minutes" class contains a number of minutes to the target date

Number of minutes will be placed here by JavaScript.

Mask layer is used to simulate obfuscation

Div block with a "seconds" class contains a number of seconds to the target date

First character of seconds' number will be placed into the div with a "second1" class and second character into the div with a "second2" class

Mask layer is used to simulate obfuscation

Container with a mirrored clock. It is used in Glass clock skin.

Container with a "glass panes" right under the clock

First glass pane with a "Learn More" button image

Middle glass pane with a "mail" image. It is used to open contact form

And the last pane with social icons

```
39 <div class="hours">
40 <p>Hours</p>
41 <div class="h-bg">
42 <div class="h-wrap">
43 <div class="hour" style="top:0px;"></div>
44 </div>
45 <div class="mask2"></div>
46 </div>
47 </div>
48 <div class="minutes">
49 <p>Minutes</p>
50 <div class="m-bg">
51 <div class="m-wrap">
52 <div class="minute" style="top:0px;"></div>
53 </div>
54 <div class="mask2"></div>
55 </div>
56 </div>
57 <div class="seconds">
58 <p>Seconds</p>
59 <div class="s-bg">
60 <div class="s-wrap">
61 <div class="second1" style="top:0px;"></div>
62 <div class="second2" style="top:0px;"></div>
63 </div>
64 <div class="mask2"></div>
65 </div>
66 </div>
67 </div>
68 <div class="clock-mirror"></div>
69 </div>
70 <div class="glass-panes">
71 <div class="glass-pane big-pane">
72 <h3 class="glass-title cufoned">More About LauncherX</h3>
73 <a class="openContent" href="#">
74 
75 </a>
76 </div>
77 <div class="glass-pane small-pane">
78 <h3 class="glass-title cufoned">Contacts</h3>
79 <a class="popup" href="#contacts">
80 
81 </a>
82 </div>
83 <div class="glass-pane big-pane" style="margin-right:0px;">
84 <h3 class="glass-title cufoned">Launcher.Social</h3>
85 <a href="#" class="soc-icon"></a>
86 <a href="#" class="soc-icon"></a>
87 <a href="#" class="soc-icon"></a>
88 </div>
89 <div class="clear"></div>
90 </div>
```



Container with additional content on homepage.

Footer

"content-wrap" section contains main content which is hidden by default

Left content column with information links inside.

Right content column with "send email address" form and additional content

Send Email form with one input field and span button

JavaScript email validation will be executed on "onkeyup" event.

After the message with visitor's email address will be sent, the status message will be added to the div block with a "message-status" class.

Dark layer will fade in when the popup will open

Popup window container

Close image

Content of the popup window will be placed here

Here will be placed the same content as inside the popup window to calculate popup window size before it will open. Dummy block is always hidden.

```
91 <div class="home-content">
92   <h2 class="home-heading cufoned">LauncherX - Premium Countdown Template</h2>
93   <p><strong>LauncherX</strong> is the premium Ajax based countdown template designed by
<strong>iDangero.us</strong>. It is designed for temporary usage until the main site is
under construction. It comes with an awesome animated clock with a countdown to the grand
opening date, brief information about the main site (or company) and Ajax Contact form with
validation. Also, there is an opportunity for visitors to leave their email address to be
informed when the site will open. <a class="openContent" href="#">Learn more</a> about
LauncherX</p>
94 </div>
95 <div class="footer">
96   <p>&copy; 2011 LauncherX Premium Template by iDangero.us. All Rights Reserved.</p>
97 </div>
98 <div class="content-wrap">
99   <div class="content-top">
100     <div class="content-bot">
101       <div class="content-mid">
102         <div class="left-col">
103           <h2 class="heading cufoned" style="color:#555">About LauncherX</h2>
104           <ul class="ul-links">
105             <li><a class="popup" href="#about">What is LauncherX</a></li>
106             <li><a class="popup" rel="600" href="#features">LauncherX Features</a></li>
107             <li><a class="popup" rel="500" href="#browsers">Browser Compatibility</a></li>
108             <li><a class="popup" href="#contacts">Ajax Contact Form</a></li>
109             <li><a class="popup" href="#lorem">Lorem Ipsum Dolor</a></li>
110             <li><a class="popup" href="#ipsum">Phasellus Nulla</a></li>
111           </ul>
112         </div>
113         <div class="right-col">
114           <h2 class="heading cufoned">Be the first to know</h2>
115           <p>Give us your email address and we will inform you when our site will open:</p>
116           <div class="sendEmail">
117             <input value="Your email here" type="text" id="email" onkeyup=
"validateEmail('#email')" size="20" />
118             <span class="button-l" id="sendEmail"><span class="button-m"><span style="
padding:0px 40px" class="button">Send</span></span></span>
119             <div class="message-status"></div>
120             <div class="clear"></div>
121           </div>
122           <h2 class="heading cufoned">Some additional content</h2>
123           <p style="text-align:justify">Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Proin id mi massa. Aliquam eleifend justo non sapien dapibus vehicula. Curabitur et
nisl nunc, nec porta dolor. Curabitur eu <a href="#about" class="popup">lobortis</a> nisi.
Mauris ullamcorper enim vel ipsum pellentesque ultricies. Sed erat erat, blandit sed posuere
et, egestas eu diam. Mauris gravida condimentum quam, nec condimentum leo vestibulum vel.
Aenean libero ipsum, elementum id iaculis vel, imperdiet varius dui.</p>
124         </div>
125       </div>
126     </div>
127   </div>
128 </div>
129 </div>
130 </div>
131 <div class="dark-layer"></div>
132 <div id="popup">
133   
134   <div class="popup-content"></div>
135 </div>
136 <div style="display:none;" class="dummy-block"></div>
137 </body>
138 </html>
```

## Footer Section

---

Footer section is used for copyrights information of you company or project. It will be automatically added to any loaded page.

```
<div class="footer">
  --- Footer Content ---
</div>
```

Change --- Footer Content --- to something like this: © **2010 My Company. All Rights Reserved**

## 5. Typography

---

### Additional Myriad Pro font usage

---

Device is powered with an awesome additional Myriad Pro font. Its usage is available due to the Cufon JavaScript library.

To use Myriad Pro font for template elements all you need is to add “**cufoned**” class to them.

### Headings

---

There is only one heading type used in this template. It used here for titles of in popup window and in content section titles:

Be the first to know

You need the following formatting for such headings:

```
<h2 class="heading">Some Heading</h2>
or
<h2 class="heading cufoned">Heading with Myriad Pro font</h2>
```

### Button

---

LauncherX Features

To create button you have to create “link” or “button” element with a “button” class and wrap it with two spans like in example below

You should use the following formatting for buttons:

```
<span class="button-l">
  <span class="button-m">
    <a href="index.php" class="button">Button Text</a>
  </span>
</span>
---- or ----
<span class="button-l">
  <span class="button-m">
    <input type="submit" class="button" value="Button Text">
  </span>
</span>
```

## 6. Inner Linking (Popup Window)

---

### *Popup Window*

---

As the LauncherX is “one-page” template, all additional content should be loaded into Popup Window. So let's see how the Inner Linking works. This you should to know first:

*All files with HTML content you want to put into Popup Window with Ajax request must be located in the **content/** folder.*

*All files must have a **.html** extension.*

**Example 1.** We want to put HTML content from the **content/about.html** file into the Popup Window. We should use the following formatting:

```
<a class="popup" href="#about">About LauncherX</a>
```

When we'll specify a “popup” class, the script will load file content with Ajax into Popup Window

Filename of the **html** file from the **content/** folder specified in the **href** attribute **after the hash character**.

**Example 2.** We want to put HTML content from the **content/about.html** file into the Popup Window. We also want to set popup window width to 900px. We should use the following formatting:

```
<a class="popup" href="#about" rel="900">About LauncherX</a>
```

We can set width of popup window with additional REL attribute. So in this case new window will open and it will be 900px in width

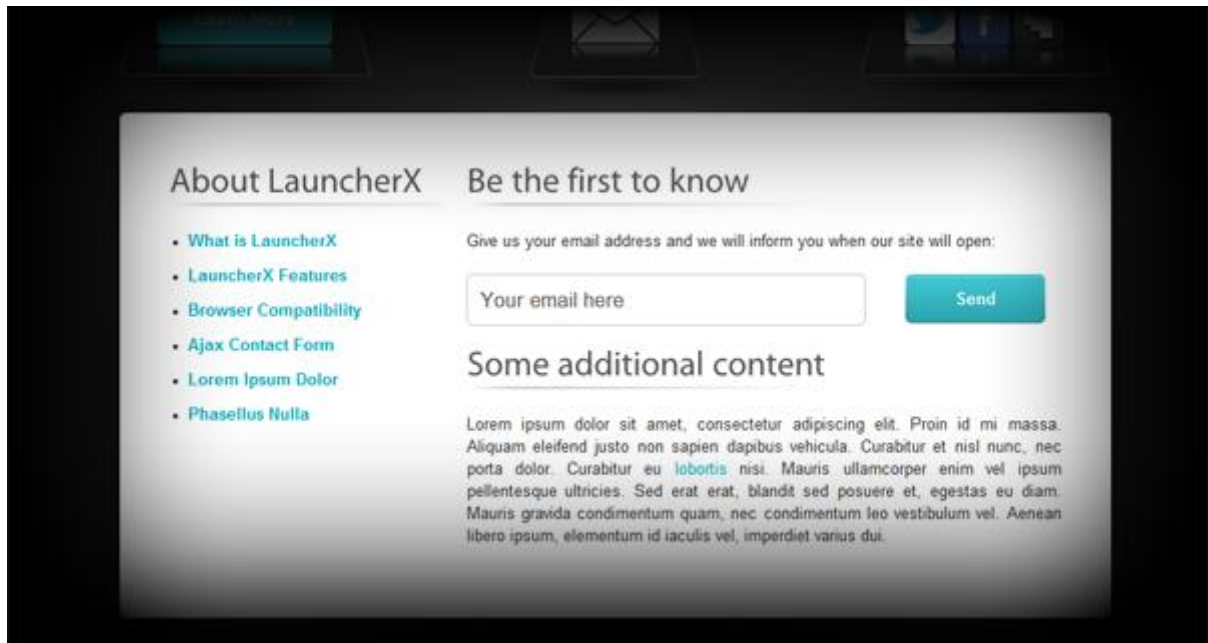


As you can see it is very easy to use Inner Linking. All you need is to specify filename in the HREF attribute after the hash character and to specify the “**popup**” class to load content into Popup Window with Ajax.

## Content Section

---

LauncherX has an additional content section:



This section will open after you'll click on “Learn More” button on the first glass pane under the clock. If you want to open this section with some other element you should add an “openContent” class to it. For example to open content section with some link from “Home Content” you should use the following formatting:

```
<a class="openContent" href="#">Open Content</a>
```

## 7. CSS Files and Color Themes

---

### CSS Files

---

There are three CSS files used in this Template and three additional CSS files used for color themes and clock skins. You can find css attachment in the HEAD section of index.html file.

```

<head>
-----
<link rel="stylesheet" type="text/css" href="css/launcher-black.css" />
<link rel="stylesheet" type="text/css" href="css/clock-glass.css" />
-----
<link rel="stylesheet" type="text/css" href="css/ie6.css" />
-----
</head>

```

By default they are located in the **css/** folder. You can easily change the path or filename in the **href** attribute of **link** tag.

**Launcher-black.css** file contains all general layouts for the template and its Black color theme. It is separated into sections:

```
/* --- Common Rules --- */
```

*There are common rules in this section, such as font size, fonts, link colors, layout images etc.*

```
/* --- Logo--- */
```

*Some rules for Logo*

```
/* --- Wrapper--- */
```

*All style rules for Wrapper*

```
/* --- Glass Panes--- */
```

*Rules for Glass Panes, their size and position.*

```
/* --- Home Content--- */
```

*Few rules for home content section, its size, position, heading*

```
/* --- Footer--- */
```

*Footer styling*

```
/* --- Content--- */
```

*All styling rules for content pages*

```
/* --- Forms--- */
```

*This section contains style rules for buttons and form elements*

```
/*----- Popup Window-----*/
```

*Styling rules for the popup window*

```
/*-----End of CSS File-----*/
```

If you would like to edit a specific section of the site, simply find the appropriate section in the CSS file, and then scroll down until you find the appropriate style that needs to be edited.

Second CSS file **clock-glass.css** contains all rules for glass skin of clock.

And the last CSS file **ie6.css** contains few rules for Internet Explorer 6 browser.

## Color Themes

---

LauncherX comes with three color themes – Black (default), Blue and Red. Now let's see how to apply color theme you need.

First of all you have to change main CSS file to the appropriate color theme CSS file:

```
<head>
----- Default (Black theme) is -----
<link rel="stylesheet" type="text/css" href="css/launcher-black.css" />

----- To use Blue color theme change it to: -----
<link rel="stylesheet" type="text/css" href="css/launcher-blue.css" />

----- To use Red color theme change it to: -----
<link rel="stylesheet" type="text/css" href="css/launcher-red.css" />
-----
</head>
```

After that you have to change <body> class to the appropriate color theme class:

```
<head>
----- Default (Black theme) is -----
<body class="black">

----- To use Blue color theme change it to: -----
<body class="blue">

----- To use Red color theme change it to: -----
<body class="red">
-----
</head>
```

That's done.

## Clock Skins

---

LauncherX comes with two clock skins: Elegant Glass (default) and Gradient White. Let's see how to use required clock skin. Actually all you need is to change the CSS file that is responsible for clock skin:

```
<head>
----- Default (Elegant Glass) is -----

<link rel="stylesheet" type="text/css" href="css/clock-glass.css" />

----- To use Gradient White clock skin change it to: -----

<link rel="stylesheet" type="text/css" href="css/clock-white.css" />

</head>
```

## 8. PSD Files & Structure

There are **five** layered **.psd** files comes with the **LauncherX** template. You can locate them into **psd/** folder form the downloaded archive. Using these files you can easily redesign template. To edit these files you'll need an Adobe™ Photoshop™ (Recommended is CS4 or higher version)

First three .psd files are **launcher-black.psd**, **launcher-blue.psd** and **launcher-red.psd**. They contain all general layouts for template styling for every color theme:



		Logo
		Open Soon
		Glass Panes
		Message
		Button
		Content
		Glass Clock
		White Clock
		Home Content
		Footer
		BGs

Fourth psd file is the **social-bar.psd**, it contains social icons styling:

And the final **psd** file is the **ajax-loader.psd**, which will help you to redesign Ajax loader.

All layers in these file are well organized and easy to understand and redesign.

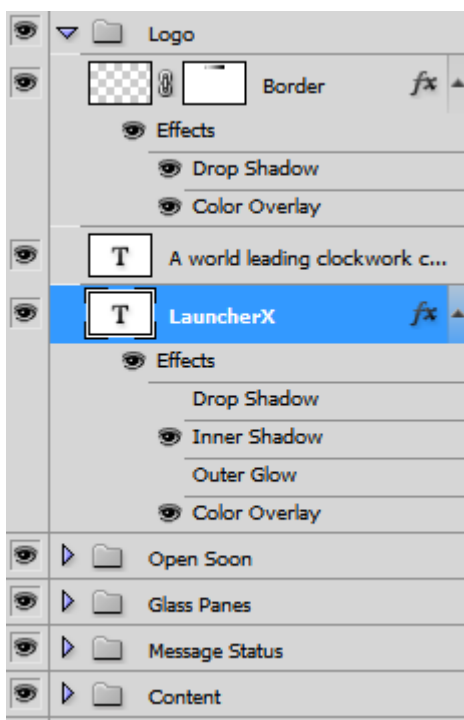
## 9. Customization

---

To customize LauncherX template you need to use included psd files and probably to make some changes in template CSS files.

Let's for example change logo and background color of black theme to orange.

First of all open *launcher-black.psd* file. Find "Logo" layers group and expand it, choose "Horizontal Type Tool", click on the "LauncherX" logo text and type your own:

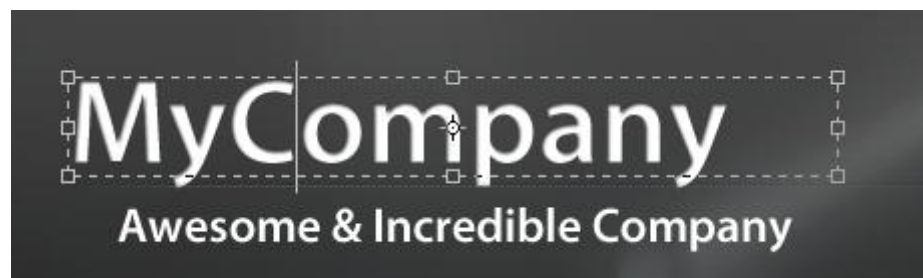


Horizontal Type Tool

Click on the "LauncherX" logo text

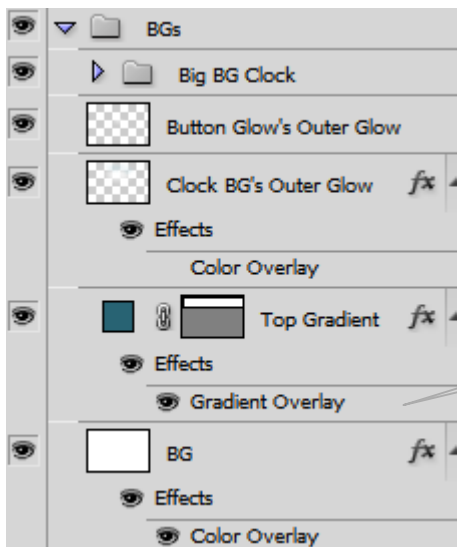


Change it to your own



Change the slogan in the same way

Now let's change background color to orange. Find the last "BGs" layers group and expand it, we will need here "Top Gradient" and BG layers:



Now we have Orange background color

Now we have to return to logo editing to fix color of the line between company title and its slogan. Find the "Border" layer inside of "Logo" layer group and change its color to the same as its background, but a little lighter:





Do the same actions with a footer line located in the “Footer” layers group and with Home heading border located in the “Home Content” layers group.

Now we have to cut these images for web usage. Go to **images/layouts-black/** folder. Find the appropriate files and pay attention to their size. It will be very good if it will be the same. In this folder we should replace **bg.png**, **bg.jpg**, **home-heading.png**, **footer.png** and **logo.png** images.

- Bg.png – is the 10px in width and full height cut of “Top Gradient” layer.
- To make the same bg.jpg image (with a same size) you need to merge BGs layer group, and to cut it in 1100x800px from top center.
- To make logo.png image you need to merger “Logo” layers group.
- To make home-heading.png and footer.png you need to convert them to Smart Object and then to Rasterize

After file replacement we need to open css/launcher-black.css file and to find rules for body background, logo, footer and home-heading.

Body style rule is the first in file. We need to change its background color

```
body {  
    background:#111111 url(../images/layouts-black/bg.png) repeat-x left top;  
    font-family:Arial, Helvetica, sans-serif;  
    margin:0px;  
    font-size:13px;  
    line-height:1.4em;  
    color:#333;  
}
```

#111111 color is changed to #df7d1d (orange)

We have to change to:

```
body {  
    background:#df7d1d url(../images/layouts-black/bg.png) repeat-x left top;  
    font-family:Arial, Helvetica, sans-serif;  
    margin:0px;  
    font-size:13px;  
    line-height:1.4em;  
    color:#333;  
}
```

Actually if you replace logo and background images with same size images, don't need to change background properties for theme. Otherwise you probably will need to fix their size and position:

Here is rule for Background image:

```
#background {  
  background:url(../images/layouts-black/bg.jpg) no-repeat top;  
  width:1100px;  
  height:800px;  
  position:absolute;  
  top:0px;  
  left:50%;  
  margin-left:-550px;  
  z-index:-1;  
}
```

Probably you'll need to change Background's width, height, left and top positions

Rule for Logo:

```
#logo {  
  width:600px;  
  height:88px;  
  background:url(../images/layouts-black/logo.png) no-repeat left top;  
  position:absolute;  
  top:60px;  
  left:80px;  
}
```

Probably you'll need to change Logo's width, height, left and top positions

As you can see it's very easy to customize LauncherX template, all you need is to find appropriate layer in psd file, replace existing image and make some changes in CSS file.

## 10. JavaScripts

---

LauncherX template uses four JavaScript files

Attachment of JavaScript files can be found in the HEAD section of index.html file:

```
<head>  
-----  
<script type="text/javascript" src="scripts/jquery-1.5.1.min.js"></script>  
<script type="text/javascript" src="scripts/cufon-yui.js"></script>  
<script type="text/javascript" src="scripts/Myriad_Pro_400-Myriad_Pro_600.font.js"></script>  
<script type="text/javascript" src="scripts/launcher.js"></script>  
-----  
</head>
```

You can easily change the path or filename in the **src** attribute of **script** tag.

The first script file is the jQuery (JavaScript library that greatly reduces the amount of code that you must write).

Second script file (cufon-yui.js) is the Cufon library. Used here for text replacement with a Myriad Pro font. [Visit official website](#).

Third script file (Myriad\_Pro\_400-Myriad\_Pro\_600.font.js) is the encoded Myriad Pro font.

And the final script file (launcher.js) is the custom script used for animation, effects and Ajax requests. You can learn more about jQuery methods used here (`fadeOut()`, `fadeIn()`, `fadeTo()`, `animate()`, `jQuery.get()`, `html()`, `delay()`, `hover()` etc.) by visiting this [jQuery Documentation](#) Site.

Let's look at the custom script more closely.

The first part of the script is used for Ajax loader image initialization, to set target date, to get current date and to calculate difference between them.

```
1 // JavaScript Document
2 var isIE6
3 /*-----Ajax Loader initialization----- */
4 var ajaxLoader = new Image();
5   ajaxLoader.src = 'images/ajax-loader.gif';
6 /*-----Current and target dates----- */
7 var currentDate = new Date();
8 var targetDate = new Date();
9   targetDate.setUTCFullYear(2012); // Target year in UTC time zone
10  targetDate.setUTCMonth(0); // Target month in UTC time zone (from 0(Jan) to 11(Dec))
11  targetDate.setUTCDate(1); // Target day in UTC time zone (from 1 to 31)
12  targetDate.setUTCHours(0); // Target hours in UTC time zone (from 0 to 23)
13  targetDate.setUTCMinutes(0); // Target minute in UTC time zone (from 0 to 59)
14  targetDate.setUTCSeconds(0); // Target second in UTC time zone (from 0 to 59)
15 /*-----Difference between target and current date----- */
16 var diff = {};
17   diff.ms = targetDate.getTime() - currentDate.getTime();
18   diff.days = Math.floor((diff.ms)/(1000 * 60 * 60 * 24));
19   diff.hours = (diff.ms - diff.days*(1000 * 60 * 60 * 24))/(1000 * 60 * 60);
20   diff.fullHours = Math.floor(diff.hours);
21   diff.mins = (diff.hours - diff.fullHours)*60;
22   diff.fullMins = Math.floor(diff.mins);
23   diff.seconds = (diff.mins - diff.fullMins)*60;
24   diff.fullSec = Math.floor(diff.seconds);
25   if (diff.fullHours<10) diff.fullHours = "0"+diff.fullHours;
26   if (diff.fullMins<10) diff.fullMins = "0"+diff.fullMins;
27   if (diff.fullSec<10) diff.fullSec = "0"+diff.fullSec;
28   if (diff.days<100 && diff.days>=10) diff.days = "0"+diff.days;
29   if (diff.days<10) diff.days = "00"+diff.days;
```

Current date

In these variables you need to set target time according to UTC time zone. On this screenshot target date equal to 1<sup>st</sup> January, 2011 00:00 (New Year)

Some math to calculate difference between target and current dates

If some of calculated difference values is one-character we have to add "0" before it or "00" for number of days.

Next part of the script is executed when the document is loaded. Here are animation functions for the clock timer, some effects for links, rules for handling popup links, send email form handling and animation of social bar icons:

Here we have to insert calculated difference values inside div blocks with "day", "hour" and "minute" classes.

Here we have to insert calculated seconds difference. But first of all we have to split seconds number by characters, because first and second characters are in different div blocks.

```

31 $(function() {
32     /*-----Clock Time Initialization----- */
33     $(".day").html(diff.days);
34     $(".hour").html(diff.fullHours);
35     $(".minute").html(diff.fullMins);
36     if ((diff.fullSec+"").length==2) {
37         var second1 = (diff.fullSec+"").substr(0,1);
38         var second2 = (diff.fullSec+"").substr(1,1);
39     }
40     else {
41         var second1 = 0;
42         var second2 = diff.fullSec;
43     }
44     $(".second1").html(second1);
45     $(".second2").html(second2);
46     function changeDay() {
47         var currentDay = $(".day:first").html()*1;
48         var newDay = currentDay-1;
49         if (newDay < 100 && newDay >= 10) newDay = "0"+newDay;
50         if (newDay < 10) newDay = "00"+newDay;
51         $(".d-wrap").prepend('<div class="day">'+newDay+'</div>');
52         $(".day:eq(0),.day:eq(1)").animate({top:"+=80px"},2500,function(){$(".day:eq(1)"}
53     ).remove()});
54     }
55     function changeHour() {
56         var currentHour = $(".hour:first").html()*1;
57         var newHour = currentHour-1;
58         if (newHour < 0) {
59             newHour = 23;
60             changeDay();
61         }
62         if (newHour < 10 && newHour >= 0) newHour = "0"+newHour;
63         $(".h-wrap").prepend('<div class="hour">'+newHour+'</div>');
64         $(".hour:eq(0),.hour:eq(1)").animate({top:"+=80px"},2000,function(){$(
65     '.hour:eq(1)').remove()});
66     }
67     function changeMinute() {
68         var currentMin = $(".minute:first").html()*1;
69         var newMin = currentMin-1;
70         if (newMin < 0) {
71             newMin = 59;
72             changeHour();
73         }
74         if (newMin < 10 && newMin >= 0) newMin = "0"+newMin;
75         $(".m-wrap").prepend('<div class="minute">'+newMin+'</div>');
76         $(".minute:eq(0),.minute:eq(1)").animate({top:"+=80px"},1500,function(){$(
77     '.minute:eq(1)').remove()});
78     }
79     function changeFirstSecond() {
80         var currentSecond = $(".second1:first").html()*1;
81         var newSecond = currentSecond-1;
82         if (newSecond < 0) {
83             newSecond = 5;
84             changeMinute();
85         }
86         $(".s-wrap").prepend('<div class="second1">'+newSecond+'</div>');
87         $(".second1:eq(0),.second1:eq(1)").animate({top:"+=80px"},500,function(){$(
88     '.second1:eq(1)').remove()});
89     }
90 }

```

Here we have to insert first seconds' character inside div block with "second1" class, and second character inside the "second2" div.

changeDay() function is used to change day's number when the hour' number is changing from 0 to 23.

changeHour() function is used to change hours' number when the minutes' number is changing from 0 to 59.

changeMinute() function is used to change minutes' number when the first seconds' character is changing from 0 to 5.

changeFirstSecond() function is used to change first seconds' character when the 2<sup>nd</sup> seconds' character is changing from 0 to 9.

```

86 setInterval(function() {
87     var currentSecond = $(".second2:first").html()*1;
88     var newSecond = currentSecond-1;
89     if (newSecond < 0 ) {
90         newSecond = 9;
91         changeFirstSecond();
92     }
93     $(".s-wrap").prepend('<div class="second2">'+newSecond+'</div>');
94     $(".second2:eq(0), .second2:eq(1)").animate({top:"+=80px"}, 500, function() { $(".second2:eq(1)").remove() });
95 }, 1000);
96
97 setTimeout(function() { $(".divider").hide() }, 500)
98 setInterval(function() {
99     $(".divider").show()
100     setTimeout(function() { $(".divider").hide() }, 500)
101 }, 1000)
102 /*-----Glass panes animation----- */
103 $(".glass-panes").find("img").each(function() {
104     makeItAnimated($(this), $(this).width())
105 })
106 function makeItAnimated(img, imgWidth) {
107     var newSmallWidth = imgWidth*0.9;
108     img.hover(
109         function() {
110
111             img.animate({width: newSmallWidth}, 150)
112         },
113         function() {
114             img.animate({width: imgWidth}, 150)
115         }
116     )
117 }
118 /*-----"Open Content" animation----- */
119 $(".openContent").click(function(e) {
120     e.preventDefault();
121     $(".h1.will-open").fadeOut(300);
122     $(".clock").animate({top:"-80px"}, 600);
123     $(".glass-panes").animate({top:"-110px", width:800}, 600);
124     $(".glass-pane:eq(0), .glass-pane:eq(1)").animate({marginRight:128}, 600)
125     $(".home-content, .footer").fadeOut(600);
126     $(".content-wrap").delay(800).fadeIn(600)
127 })
128 /*-----"About" links animation----- */
129 $(".ul-links li a").hover(
130     function() {
131         $(this).animate({marginLeft:"5px"}, 150)
132     },
133     function() {
134         $(this).animate({marginLeft:"0px"}, 150)
135     }
136 )
137 /*-----Popup links handling----- */
138 $(".a.popup").click(function(e) {
139     e.preventDefault();
140     var filename = $(this).attr('href').substr(1);
141     popUp({url : "content/"+filename+".html", pWidth : $(this).attr("rel") });
142 })
143 $(".dark-layer, .close-popup").click(function() { closePopup() })

```

This is the main timer function, which is used to animate (scroll) 2<sup>nd</sup> character of seconds' number. It is executing with 1 second interval. With every second this function adds div block with a new second before existing character, then it animates them (scroll), and after, old second's character will be removed. If new second is < 0, then it will be equal to 9 and changeFirstSecond() function will be executed.

These function are used to animate "colon", to make it blink with every second.

When we'll click on image inside of div with a "glass-panes" then the makeItAnimated() function will be executed. We have to use such separated function to prevent "double-hover" bug in IE6 browser

Width of every image will be decreased to 90% of its real width for 150ms

When we'll click on link with an "openContent" class, then the "clock" will be animated to top position equal to "-80px" for 600ms, "will-open" title will be hided for 300ms, glass panes will change their top and width attributes, content section will fade in for 600ms after the 800ms delay.

Animation for list links. First function will be executed when we'll take the mouse cursor on the list element and second one is used when we'll take the mouse cursor away from it.

Handling rule for links with a "popup" class. When we'll click on it, then the filename value will be equal to HREF attribute of this link. And pWidth variable will be equal to REL attribute of this link. Then the popUp() function will be executed to load .html file from content/ folder.

When we'll click on the "dark-layer" or on the image with a "close-popup" class, then the closePopup() function will be executed.



When we'll click on the email field its value will be removed

When we'll click on the sendEmail button validateEmail() function will be executed. And if the entered email is correct then the input field and "submit" button will be hidden. Ajax-loader image will be added to "sendEmail" block. Then - Ajax request to "send email" php script, and after request is completed, the message with a message status will be added to "message-status" block.

```

144  /*-----Send Email form----- */
145  $("#input#email").click(function() {
146      if($(this).val()=="Your email here") {$(this).val("")
147      }}
148  $("#sendEmail").click(function() {
149      validateEmail("#email");
150      if(!$("#input#email").hasClass("required")) {
151          $("#input#email,#sendEmail").hide()
152          $(".sendEmail").append('')
153          jQuery.get("scripts/send-email.php",{email:$("#input#email").val()},function(
status){
154              $(".sendEmail .ajax-loader").fadeOut(300,function(){$(".sendEmail
.ajax-loader").remove()})
155              $(".message-status").html(status)
156              $(".message-status").delay(400).fadeIn(600)
157          })
158      }
159  })
160 })

```

Next part of code contains functions which could be called from any place of the document.

These functions are used to validate form fields, to open and to close a popup window:

popUp() function provides an opportunity to call a Popup Window. First of all function calculates window size to set it in the center of screen with Ajax loader inside of it.

```

161 function popUp(params) {
162     var popupWidth = params.pWidth||700;
163     var popUpContent = $('#popup-content');
164     var popupLeftPadding = popUpContent.css("padding-left").split('px')[0];
165     popUpContent
166     .html('')
167     .css({width:"auto",height:"auto"})
168     .show()
169     $('#popup').css({marginLeft:-ajaxLoader.width/2-popupLeftPadding-10,top:$(window).
scrollTop()+100}).show()
170     jQuery.get(params.url,function(data){
171         $(".dummy-block").css({width:popupWidth-popupLeftPadding*2}).html(data);
172         $(".ajax-loader").css({opacity:0})
173         popUpContent.animate({width:$(".dummy-block").width(),height:$(".dummy-block").
height(),600,function(){
174             $(".dummy-block").html("");
175             popUpContent.html(data);
176             Cufon.replace(".cufoned");
177             if(isIE6) {
178                 $('#dark-layer').css({position:"absolute",height:$(window).height(),top:$(
window).scrollTop()})
179                 $(window).scroll(function(){
180                     $('#dark-layer').css({top:$(window).scrollTop()})
181                 })
182             }
183             $('#dark-layer').delay(100).fadeTo(300,0.7);
184             $('#popup-content a.popup').click(function(e){
185                 e.preventDefault();
186                 var filename = $(this).attr('href').substr(1);
187                 popUp({url : "content/"+filename+".html", pWidth : $(this).attr("rel")});
188             })
189         })
190         $('#popup').animate({marginLeft:-popupWidth/2-10},600)
191     })
192 }

```

Ajax GET request to specified file.

Then we have to add loaded HTML content to the dummy block to calculate window size for animation. Then Ajax loader will be hidden

"width" and "height" properties will be animated for 600ms, and after - loaded content will be inserted into the "popup-content" container

We need to replace popup window's fonts with a Myriad Pro font

This case is for IE6 browser, we have to add "absolute" position to "dark-layer" because the IE6 browser doesn't support "fixed" value.

"dark-layer" will fade to opacity 0.7 for 300ms after the 100ms delay

Handling rule for the links with a "popup" class inside of new window. The same as in code above

Also we have to animate "margin-left" property of the "popup" container



**closePopup()** function provides an opportunity to close a Popup Window. Window will be closed using fade animation for 0.3 seconds after the 200ms delay, and after it is closed, HTML content from the "popup-content" container will be removed.

**validateEmail()** function is used to validate email field in the contact form. Validation is based on the regular expressions. If the email address is wrong then the "required" class will be added to the email field (input field with an id attribute equal to emailFieldID variable). Otherwise it will be removed.

**validateMessage()** function is used to validate message length in the contact form. If the message contains less than 10 characters, then the "required" class will be added to the contact form **textarea** with a "contact\_message" id attribute. Otherwise the "required" class will be removed.

**sendMessage()** function provides an opportunity to send e-mail message by the Ajax request (without reloading of the site). First of all validate functions will be executed (to validate **Email** and **Message**).

If all fields are correct, then Contact form will fade out for 600ms. Ajax loader image will be added in the div with "contact-status" class

Ajax request to the php script used for sending e-mail messages with the following variables: **name**, **email**, **subject** and **message**. Script will return the status message in HTML format which will be added to ".contact-status" container and it will fade in for 600ms

```

193 function closePopup() {
194     $('#dark-layer').fadeOut(200)
195     $('#popup').delay(200).fadeOut(300,function() {
196         $('#popup-content').html('').css({width:"auto",height:"auto"});
197         $(this).css({filter:""})
198     });
199 }
200 function validateEmail(emailFieldID) {
201     var emailInput = $(emailFieldID)
202     var email = emailInput.attr("value");
203     var re = /^\\w+([\\.-]?\\w+)*@((((([a-z0-9]{2,})|([a-z0-9]([a-z0-9]+))\\.)[a-z0-9])|([a-z0-9]+[-]?)+[a-z0-9]+\\.([a-z]{2}|(com|net|org|edu|int|mil|gov|arpa|biz|aero|name|coop|info|pro|museum|mobi|travel)))$/i;
204     if(re.test(email)) emailInput.removeClass("required");
205     else emailInput.addClass("required");
206 }
207 function validateMessage() {
208     message = $("#textarea#contact_message").val();
209     m_length = ($.trim(message)).length;
210     if (m_length < 10) $("#textarea#contact_message").addClass("required");
211     else $("#textarea#contact_message").removeClass("required");
212 }
213 function sendMessage() {
214     validateEmail("#contact_email");
215     validateMessage();
216     if (!$("#contact_email,#contact_message").hasClass("required")) {
217         var name = $("#contact_name").attr("value");
218         var email = $("#contact_email").attr("value");
219         var subject = $("#contact_subject").attr("value");
220         var message= $("#contact_message").val().replace(/<\\/?[>]+>/gi, '');
221         $(".contact-form").fadeOut(600,function() {
222             $(".contact-status").html('').show()
223         })
224         jQuery.post("scripts/send-message.php",{name : name, email : email, subject :
subject, message : message},
225             function (status) {
226                 $(".contact-status").delay(600).fadeOut(300,function() {
227                     $(".contact-status").html(status).fadeIn(600)
228                 });
229             }
230         );
231     }
232 }

```

# 11. “Send Email” Script & Contact Form

## Send Email Script

If the all fields are correct **sendEmail** script will send and Ajax request to the php mailing script located in the **scripts/send\_email.php** file. This script will send a message and return mailing status message. Let's look at this script more closely:

```
1  <?php
2  if (isset($_GET['email'])) $email = $_GET['email'];
3
4  $mail_to = "demo@idangero.us"; //change this email to yours
5  $subject = "LauncherX subscription";
6  $message = "
7  New Launcher' visitor want to know when the site will be opened.
8  His email: $email
9  ";
10 $headers = "From: ".$email.">\r\n";
11 $headers .= "Reply-To: ".$email.">\r\n";
12 $headers .= "X-Mailer: PHPMailer.">\r\n";
13 $headers .= "Content-Type: text/plain; charset=\utf-8\"";
14 if (mail($mail_to, $subject, $message, $headers)) {
15     ?>
16
17     <div class="mail-sent">
18         <p>Thanks! Your email has been successfully sent!</p>
20     </div>
21     <?php
22     }
23     else {
24     ?>
25     <div class="mail-error">
26         <p>Error Occured. Try again later.</p>
28     </div>
29     <?php
30     }
31     ?>
```

Checking for existence of **\$email** variable

Message will be sent to this e-mail address. So you have to change it to yours

Default headers for the e-mail message.

If the message was successfully sent, the script will return this HTML message.

Otherwise script will return this HTML code with an error message.

You can easily edit this status messages using any HTML formatting.

# Ajax Contact Form

---

HTML code of the contact page is in the **content/contacts.html** file:

```
<h2 class="heading cufoned">Contacts</h2>
<div class="contact-status"></div>
<form name="contact" class="contact-form" action="javascript:sendMessage()" method="post">
  <p>You can contact us using the following contact form:</p>
  <p>
    <label> Your Name:<br>
      <input id="contact_name" type="text" class="text" name="name" size="30" />
    </label>
  </p>
  <p>
    <label> Your Email:<br>
      <input id="contact_email" onkeyup="validateEmail('#contact_email')" type="text"
class="text" name="email" size="30" />
    </label>
  </p>
  <p>
    <label> Subject:<br>
      <input id="contact_subject" type="text" class="text" name="subject" size="30" />
    </label>
  </p>
  <p>
    <label> Message:<br>
      <textarea id="contact_message" onkeyup="validateMessage()" name="message" class=
"textarea" cols="30" rows="10"></textarea>
    </label>
  </p>
  <p><em>Fields marked with an asterisk(*) are required</em></p>
  <span class="button-l"><span class="button-m">
    <input type="submit" class="button" value="Send Message" />
  </span></span>
</form>
```

Status message will be inserted here

If the all fields are correct sendMessage() script will send and Ajax request to the php mailing script located in the **includes/send\_message.php** file. This script will send a message and will return mailing status message. Let's look at this script more closely:

If the all fields are correct sendMessage() script will send and Ajax request to the php mailing script located in the **includes/send\_message.php** file. This script will send a message and will return mailing status message. Let's look at this script more closely:

```
1 <?php
2 if (isset($_POST['name'])) $name = $_POST['name'];
3 if (isset($_POST['email'])) $email = $_POST['email'];
4 if (isset($_POST['subject'])) $subject = $_POST['subject'];
5 if (isset($_POST['message'])) $message = $_POST['message'];
6 $mail_to = "demo@idangero.us"; // change this email to yours
7 $message = stripslashes($message);
8 $headers = "From: ".$name." <".$email.">\r\n";
9 $headers .= "Reply-To: ".$email."\r\n";
10 $headers .= "X-Mailer: PHPMailer".".\r\n";
11 $headers .= "Content-Type: text/plain; charset=\\"utf-8\\"";
12 if (mail($mail_to, $subject, $message, $headers)) {
13 }
14
15 <div class="mail-sent">
16 <p></p>
18 <p>Thanks! Your message has been successfully sent!</p>
19 </div>
20 <?php
21 }
22 else {
23 }
24 <div class="mail-error">
25 <p></p>
27 <p>Error Occured. Try again later.</p>
28 </div>
29 <?php
30 }
```

Checking for existence of these variables.

Message will be sent to this e-mail address. So you have to change it to yours

Default headers for the e-mail message.

If the message was successfully sent, the script will return this HTML message.

Otherwise script will return this HTML code with an error message.

You can easily edit this status messages using the any HTML formatting.

## 12. Licensing Terms

---

### Regular License (RL)

RL gives you as a customer non-exclusive & non-transferable right to use the product you've bought, in this case is the **LauncherX** (further "**Item**")

1. We do not limit the number of **Item**'s copies you are going to use. Using one **Item** you can create, for example, even 100 web-sites.
2. You can use the **Item** by itself or it's also possible to apply it in other project you work at.
3. You can use the **Item** for your own purpose as well as for your clients.
4. You can use the **Item** in commercial projects.
5. You can reproduce the **Item**:
  - on a web-site or as a web-site
  - as a part of software you create not for sale
  - as a printed variant
  - in digital(electronic) format (as a presentation or an e-book)
  - in video products
6. All photos used on iDangero.us for demonstration photo gallery functioning are our private property (if only there is nothing said about contrary). It's prohibited to use these photos wherever.
7. You are not allowed to sell, resell, license or give the **Item** free (any way) without our written consent. Please, do not offer to do it to any person.
8. You also do not have the right to use the **Item** in a project for selling (for example, for other templates, scripts, graphics and so on).
9. It's prohibited to rework / redesign / reproduce the **Item** (i.e.to rename it or change graphics & so on) and after this to sell it as your own.
10. In spite of reselling limitation you could claim money for the **Item** from your client.
11. If the **Item** (the whole **Item** or its parts) is created with materials used by GNU General Public License (GPL) (or some other license) it means you should follow all the terms of the license using the **Item**.