



“Croplane”

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 “Croplane”
2. Croplane Files & Folders Structure
3. Installation
4. HTML Structure
5. Inner Linking (How it works?)
6. CSS Files and Structure
7. JavaScripts
8. Default Content Layouts
 - 8.1 Windowed Layout Type
 - 8.2 Slideshow (HTML & JavaScript)
 - 8.3 Portfolio (HTML & JavaScript)
 - 8.4 Image Gallery (HTML & JavaScript)
 - 8.5 Offer Section
9. Ajax Contact Form
10. Social.Bar
11. PSD Files and Structure
12. Licensing Terms

1. About “Croplane”

Croplane is the Ajax based website template designed by iDangero.us team.

Croplane is an unusual template which is good for small projects, fashion sites, portfolios. You will not see here the usual home page or menu navigation. Instead, there are animated blocks with titles. Click on it and you'll get the desired page.

Page content is loaded dynamically with good looking animation. It all works thanks to Ajax technology and the use of jQuery JavaScript library.

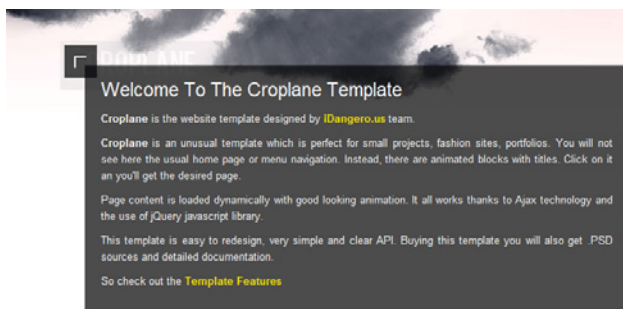
This template is easy to redesign, very simple and clear API.

The main feature of Croplane template is that it has five different content layouts:

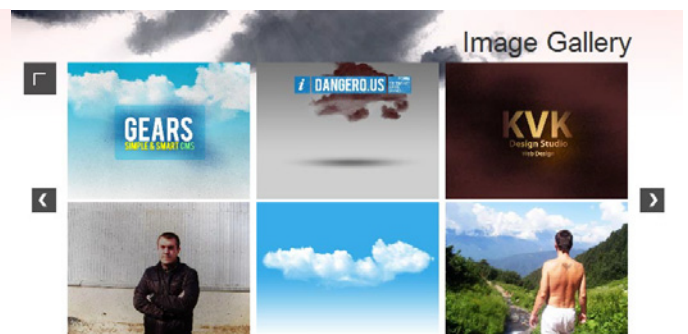
- Simple windowed content like this page (Windowed Layout Type)
- And 4 types of Content Layout Type:
 1. Slideshow layout type allows you to show content using animated slides
 2. Image Gallery. Show your photos in lightbox with a photo description
 3. Portfolio. Share your works in animated lightbox with a work description
 4. Offer Section. Actually you can put here not only images, but and any html content

The main difference between Windowed and Content Layout Types is that in the first type content loads into the window with a background. The Windowed Layout is good to use for text pages, other is good for customizable content.

Example of Windowed Layout Type



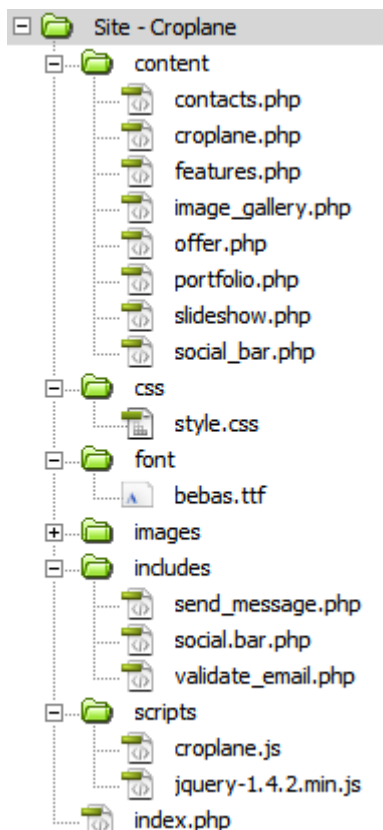
Example of Content Layout Type



Also Croplane powered with an Ajax Contact Form with validation

2. Croplane Files Structure

Template files are located in the **Croplane** folder from the downloaded archive:



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

css: folder with a .css files

font: folder with a additional Bebas font used in this template

images: here you can find layouts images and photos

includes: .two php scripts used for Ajax contact form and social.bar html layout

scripts: JavaScripts used in this template

and the main **index.php** in root folder

3. Installation

To install **Croplane** template you have to copy all files & folders from the **Croplane** 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 Croplane Template. Open index.php file with a text editor (Highly recommended to use Adobe® Dreamweaver® or Notepad ++ editors)

Index.php <body> structure:

```
<body>
<!--Additional Background with clouds-->
<div id="background"></div>
<!--End of additional Background with clouds-->

<!--Logo-->
<div id="logo" >
    <a href="index.php">
        
    </a>
</div>
<!--End of logo-->

<div class="wrapper">
    <!--Page Title-->
    <h2 class="page-title"></h2>
    <!--End of Page Title-->

    <!--HomePage Block Links-->
    <div class="blocks">
        -Block Links Here -

        <!--Social Bar Block-->
        <div class="bot_1 block social_bar">
            <?php include("includes/social.bar.php")?>
        </div>
        <!--End of Social Bar-->

    </div>
    <!--End of HomePage Block Links-->

    <!--Shadow image for block links-->
    
    <!--End of shadow image for block links-->

    <!--Section for window layout type -->
    <div id="window">
        
        <div class="window-content">
            -Window Content loaded by Ajax Here-
        </div>
        
    </div>
    <!--End of section for window layout type -->

    <!--Section for other layout types -->
    <div id="page">
        
        <div class="page-content">
            -Page Content loaded by Ajax Here-
        </div>
    </div>
    <!--End of section for other layout types -->
    <div class="clear"></div>
</div>
<!--Footer Section -->
<div id="footer">
    -Footer Content-
</div>
<!--End of footer Section -->
</body>
```


Logo Section

To change site **logo** you have to replace the logo.png (located in **images** folder) image to yours own or you could change the source of logo image to your image like this:

```
<!--Logo-->
<div id="logo" >
    <a href="index.php">
        
    </a>
</div>
<!--End of logo-->
```

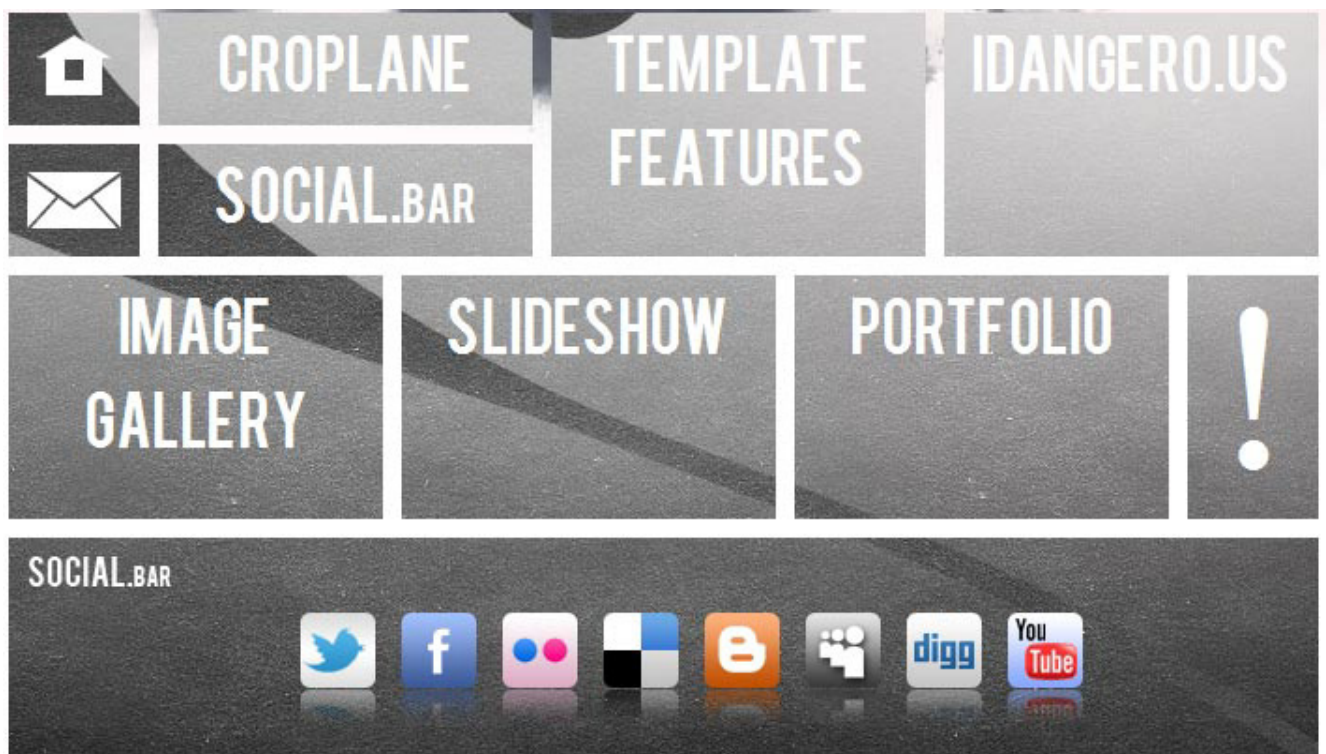
Page Title Section

```
<!--Page Title-->
<h2 class="page-title"></h2>
<!--End of Page Title-->
```

This section is used to display the Title of page loaded at this moment. Title text between <h2> and </h2> tags will be added automatically when a new page is loaded.

Block Links Section

Now let's look more closely at the **block links** section:



What are these blocks used for?

We see these blocks on the home page and they are used as a links to the site pages instead of the usual site menu. Here is the HTML code:

```
<div class="blocks">
    <div id="home" class="top_1 block">
        <a class="block-title" href="index.php"><img src=
            "images/layouts/home.png" width="40" height="32" alt="Home"
            vspace="12" /></a>
    </div>
    <div id="contacts" class="top_2 block inWindow" title="Contacts">
        <a class="block-title" href="#"><img src=
            "images/layouts/mail.png" width="50" height="30" alt=
            "Contacts" vspace="15" /></a>
    </div>
    <div id="croplane" class="top_3 block inWindow">
        <a class="block-title" href="#">Croplane</a>
    </div>
    <div id="social_bar" class="top_4 block inWindow">
        <a class="block-title" href="#">Social.<span style=
            "font-size:24px">Bar</span></a>
    </div>
    <div id="features" class="top_5 block inWindow" title=
        "Template&nbsp;Features">
        <a class="block-title" href="#">Template<br />Features</a>
    </div>
    <div id="idangerous" class="top_6 block" title="iDangero.us">
        <a class="block-title" href="http://www.idangero.us">
            iDangero.us</a>
    </div>
    <div id="image_gallery" class="m_1 block inContent" title=
        "Image&nbsp;Gallery">
        <a class="block-title" href="#">Image Gallery</a>
    </div>
    <div id="slideshow" class="m_2 block inContent" title="">
        <a class="block-title" href="#">Slideshow</a>
    </div>
    <div id="portfolio" class="m_3 block inContent" title="Portfolio">
        <a class="block-title" href="#">Portfolio</a>
    </div>
    <div id="offer" class="m_4 block inContent" title="Great Offer">
        <a class="block-title" href="#" style="font-family:georgia;
            font-size:120px;line-height:120px;">!</a>
    </div>
    <div class="bot_1 block social_bar">
        <?php include("includes/social.bar.php") ?>
    </div>
</div>
```

Each block link **div** must have class “block” and position class such a “top_1”, “top_2”, ..., “m_1”, “m_2” etc.

To change block title you have to change text between **<a>** and **** html tags and **title** attribute of the parent **div** tag. Read the next chapter to know how the inner linking works.

Window Layout Section

This section is used to display loaded content with a windowed layout type.

```
<!--Section for window layout type -->
<div id="window">
    
    <div class="window-content">
        -Window Content loaded by Ajax Here-
    </div>
    
</div>
<!--End of section for window layout type -->
```

Content with a windowed layout type will be loaded by Ajax in the **div** section with a **window-content** class attribute. “**Back To Home**” image provides an opportunity to return to the home page when you click on it.

Page Layout Section

This section is used to display loaded content with another layout type, such a slide-show, portfolio, image gallery or offer pages.

```
<!--Section for other layout types -->
<div id="page">
    
    <div class="page-content">
        -Page Content loaded by Ajax Here-
    </div>
</div>
<!--End of section for other layout types -->
```

“**Back To Home**” image provides an opportunity to return to the home page when you click on it.

Footer Section

Footer section is used for copyrights information of you company or project

```
<!--Footer Section -->
<div id="footer">
    -Footer Content-
</div>
<!--End of footer Section -->
```

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

5. Inner Linking (How it works?)

So let's see how the Inner Linking works on the example of the **block links** mentioned above. This you should to know first:

*All files with HTML content you want to load with Ajax must be located in the **content/** folder.*

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

Example 1. We want to load HTML content from the **content/features.php** file in the Windowed Layout Type.

We should use the following formatting for **block link**:

```
<div class="blocks">

    --- some other blocks here----

    <div id="features" class="top_5 block inWindow" title="Template Features">
        .....
        <a class="block-title" href="#">Template Features</a>
    </div>

    --- some other blocks here----

</div>
```

- **id** attribute is used to specify a filename that we want to load. For example, if we want to load the **content/contacts.php** file we need to change value of the **id** attribute to "**contacts**":

```
<div id="contacts" class="top_5 block inWindow" title="Contacts">
    .....
    <a class="block-title" href="#">Contacts</a>
</div>
```

- **inWindow** class for block link **div** used to load the file in Windowed Layout Type.
- **title** attribute used to display title of the page which you going to load. It will be displayed in the Page Title Section.

And after we click features block link we'll see animation and features.php file will be loaded in the Windowed Layout Type:

Template Features

"Croplane" Template Features

The main feature of Croplane template is that it has five different content layouts:

- Simple windowed content like this page.
- **Slideshow** layout type allows you to show content using animated slides.
- **Image Gallery**. Show your photos in lightbox with a photo description.
- **Portfolio**. Share your works in animated lightbox with a work description.
- **Offer Section**. Actually you can put here not only images, but and any html content.

Also Croplane powered with a Ajax **Contact Form** with validation.

Page title specified by the **title** attribute

HTML content from the **content/features.php** file specified by the **id="features"** attribute

Example 2. We want to load HTML content from the **content/features.php** file in the Windowed Layout Type directly from the simple link in the text.

In these cases, there is a JavaScript function **goTo()**. This function has two attributes: **\$filename** and **\$replaceContent**.

\$filename variable is used to specify filename from **content/** folder.

\$replaceContent variable can take two values:

- **0** To use Windowed Layout Type
- **1** To use another Layout Type

In this example we should use the following formatting for the links from text:

```
<a href="javascript:goTo('features',0)">Features</a>
or
<a href="#" onclick="goTo('features',0)">Features</a>
or

```

Example 3. We want to load HTML content from the **content/features.php** file in the Content Layout Type.

We should use the following formatting for **block link**:

```
<div class="blocks">

    --- some other blocks here---

    <div id="features" class="top_5 block inContent" title="Template Features">
        <a class="block-title" href="#">Template Features</a>
    </div>

    --- some other blocks here---

</div>
```

The difference with the first example is that the block link uses a different class – **inContent** instead **inWindow** class. In this case file content will be loaded in Content Layout Type:

Page title specified by the **title** attribute

HTML content from the **content/features.php** file specified by the **id="features"** attribute

Example 4. We want to load HTML content from the **content/features.php** file in the Content Layout Type directly from the simple link in the text.

As in the example 2 we have to use the `goTo()` JavaScript function but with the **\$replaceContent** attribute must be equal to 1.

We should use the following formatting for the links somewhere in text:

```
<a href="javascript:goTo('features',1)">Features</a>
or
<a href="#" onclick="goTo('features',1)">Features</a>
or

```

Example 5. We want to use the block link as an external link.

In this case all you need is to remove the **inWindow** or **inContent** class from the block link div and to specify **href** attribute in the children **<a>** tag:

```
<div id="features" class="top_5 block" title="Template Features">
    <a class="block-title" href="http://somesite.com">Template Features</a>
</div>
```

6. CSS Files and Structure

There is only one CSS file used in this Template. You can find css attachment in the HEAD section of index.php file.

```
<head>
---
<link rel="stylesheet" type="text/css" href="css/style.css" />
---
</head>
```

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

style.css file is separated into sections:

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

There are common rules in this section, such as font size, font faces, link colors etc. Also in this section you'll find the following rule:

```
@font-face { font-family: Bebas; src: url('../font/bebas.ttf'); } - this rule allows to attach an not standart font Bebas.
```

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

Some rules here are used for logo.

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

Rules for Footer section

```
/*-----Block Links----- */
```

*This section contains information about **block links** – their positions and styling*

```
/* -----Social.Bar----- */
```

Social.Bar icons styling

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

Some rules for Windowed Layout Type

```
/*----- Image Gallery -----*/
```

All rules for Image Gallery

```
/*----- Slideshow -----*/
```

Slideshow styling

```
/*----- Portfolio -----*/
```

Rules for Portfolio Layout Type

```
/*----- Shadows -----*/
```

A couple of rules for shadow images

```
/*-----Back Button-----*/
```

Rules for "Back To Homepage" buttons

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

Rules for contact form and buttons

```
/*-----Ajax Loader-----*/
```

Ajax Loader styling

```
/*-----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.

7. JavaScripts

Croplane template uses two JavaScript files and three inline scripts (for image gallery, slideshow and portfolio – read more about inline scripts in [chapter 8](#)).

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

```
<head>
-----
<script type="text/javascript" src="scripts/jquery-1.4.2.min.js"></script>
<script type="text/javascript" src="scripts/croplane.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).

The second script file (croplane.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 this custom script more closely.

The first part of the script is executed when the document is loaded. Here is the blink effect for menu block links, animation of Social.Bar, rules for handling menu links and the Ajax loader.

```

1 // JavaScript Document
2 $(function() {
3     /*-----Blink Effect for Block Links-----*/
4     $(".block:not(.social_bar)").mouseover(
5         function() {
6             if(!$(this).hasClass("active")) {
7                 $(this).fadeOut(200,0.7).fadeIn(200,1)
8             }
9         })
10    /*-----Menu Actions-----*/
11    $(".inWindow").click(function() {
12        if(!$(this).hasClass("active")) {
13            loadContent($(this).attr("id"),0)
14        }
15        return false;
16    })
17    $(".inContent").click(function() {
18        if(!$(this).hasClass("active")) {
19            loadContent($(this).attr("id"),1)
20        }
21        return false;
22    })
23    /*-----Back To Home functionality-----*/
24    $(".to-home").click(function() {
25        toHome();
26    })
27    $(".to-home-page").click(function() {
28        $("#page").fadeOut(600,function() {
29            $(".block").delay(300).fadeIn(900)
30            $(".blocks_shadow").delay(300).fadeIn(900)
31        })
32        $(".page-title").fadeOut(600);
33    })
34    /*-----Social Bar-----*/
35    $(".bar-icons a").hover(
36        function() {
37            $(this).children("span").fadeIn(300)
38            $(this).animate({top:"-20px"},300)
39        },
40        function() {
41            $(this).animate({top:"0px"},300)
42            $(this).children("span").fadeOut(300)
43        }
44    )
45    /*-----Ajax Loader-----*/
46    $("body").ajaxStart(function() {
47        $(this).prepend('<div class="ajax-loader"><h3>Loading...</h3></div>');
48    })
49    $("body").ajaxComplete(function() {
50        $(".ajax-loader").remove();
51    })
52 })

```

Blink effect for the Block Links. It means that when we direct the mouse on it, it fades to opacity 0.7 for 200 ms. And then it fades to opacity 1 for 200 ms.

This means that if the link has an **InWindow** class, then the content of the file will be loaded as a Windowed Layout Type.

This means that if the link has an **InContent** class, then the content of the file will be loaded as a Content Layout Type.

When the **Back To Home** link is clicked, toHome() function will be executed. This rule is for the Windowed Layout Type

The same rule as a previous one but for the Content Layout Type

Animation for Social.Bar icons. First function is for the **mouseover** action, and the second one for the **mouseout** action.

Ajax Loader. This code means that when the Ajax request is starting, the HTML code of AjaxLoader will be added to the BODY section of the document.

When the Ajax request is completed, the HTML code of AjaxLoader will be removed from the BODY section of the document.

The next part of code contains functions which could be called from any place of the document. These functions are used for Ajax requests, for submitting and validation of the contact form.

```

53 function loadContent($id,$replaceContent){
54     var $uri = "content/"+$id+".php";
55     if ($replaceContent==0) {
56         $blockLeft = $("#"+$id).css("left");
57         $blockTop = $("#"+$id).css("top");
58         jQuery.get($uri, function (content) {
59             $(".window-content").html(content);
60             $("#"+$id).addClass("active")
61             $(".block:not("#"+$id+"")").fadeOut(600)
62             $("#"+$id).delay(600).animate({left:"0px",top:"0px"},600).fadeTo(300,0.2,function(){
63                 $("#window").fadeIn(600,function(){
64                     $(".to-home").fadeIn(300);
65                     $(".page-title").html($("#"+$id).attr("title")).fadeIn(300);
66                 });
67             });
68             $("#blocks_shadow").fadeOut(600)
69         })
70     }
71     else {
72         jQuery.get($uri, function (content) {
73             $(".page-content").html(content);
74             $(".block").fadeOut(600)
75             $("#page").delay(900).fadeIn(900,function(){
76                 $(".page .to-home-page").fadeIn(300);
77                 $(".page-title").html($("#"+$id).attr("title")).fadeIn(300);
78             });
79             $("#blocks_shadow").fadeOut(600)
80         })
81     }
82 }
83 function toHome(){
84     $(".to-home, .page-title").fadeOut(600)
85     $(".active").delay(600).fadeTo(300,1,function(){
86         $(this).animate({left:$blockLeft,top:$blockTop},600,function(){
87             $(".block, #blocks_shadow").fadeIn(600);
88             $(this).removeClass("active")
89         })
90     })
91 }
92 function goTo($id,$replaceContent){
93     toHome()
94     function newWindow(){
95         loadContent($id,$replaceContent)
96     }
97     setTimeout(newWindow,2000);
98 }
99 function validateMessage(){
100     if(($.trim($("#contact_message").val()).length) < 10) {
101         $("#contact_message").addClass("required")
102     }
103     else $("#contact_message").removeClass("required")
104 }

```

loadContent() function is used for Ajax loading of the .php files with HTML content.

This case is for the Windowed Layout Type

Getting left and top positions of the clicked Block Link

Ajax GET request.

After AJAX request, the HTML content of the file will be loaded to the div block with class "window-content". After that the new window will be shown through the animation queue.

And the value of **title** attribute of the clicked block will be added to the **h2** tag with a "page-title" class

This case is for the Content Layout Type

After AJAX request, the HTML content of the file will be loaded to the div block with class "page-content". After that the new page will be shown through the animation queue.

toHome() function provides an opportunity to return to the Home Page through the animation queue. This function is used for Windowed Layout Type.

goTo() function provides an opportunity to load content directly from the simple link in the text. This function consist of two functions: toHome() and loadContent().The second function is executed with a 2-second delay.

validateMessage() function is used for validating message area in the contact form. If the message contains less than 10 characters, then the "required" class will be added to the contact form **textarea**. Otherwise the "required" class will be removed.

Code continues on the next page.

```

105 function validateEmail() {
106     var email = $("#contact_email").attr("value");
107     jQuery.post("includes/validate_email.php",{email: email},
108         function (data) {
109             if (data != 1) {
110                 $("#contact_email").addClass("required");
111             };
112             if (data == 1) {
113                 $("#contact_email").removeClass("required");
114             };
115         }
116     );
117 }
118 function sendMessage() {
119     validateEmail()
120     validateMessage()
121     if (!$("#contact_email,#contact_message").hasClass("required")) {
122         name = $("#contact_name").attr("value");
123         email = $("#contact_email").attr("value");
124         subject = $("#contact_subject").attr("value");
125         message= $("#contact_message").val().replace(/<\/?[^>]+>/gi, '');
126         jQuery.post("includes/send_message.php",{name : name, email : email,
subject : subject, message : message},
127             function (status) {
128                 $("#message_status").slideDown(300).html(status);
129             }
130         );
131     }
132 }
133 }

```

validateEmail() function is used for validating email field in the contact form. It uses Ajax request to the php script which returns "1" if the entered email address is correct and "0" if it's wrong. If it's wrong then the "required" class will be added to the email field. Otherwise it 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 the validateEmail() and validateMessage() functions will be executed. And then if the email field and message textarea will not have the "required" class, function will send an Ajax request to the php script used for sending email messages with the following variables: **name**, **email**, **subject** and **message**. Script will return the message with the mailing status, which will be added to the **div** block with id "message_status"

8. Default Content Layouts

8.1 Windowed Layout Type

Windowed Layout Type is the default layout. It's good to use for text pages



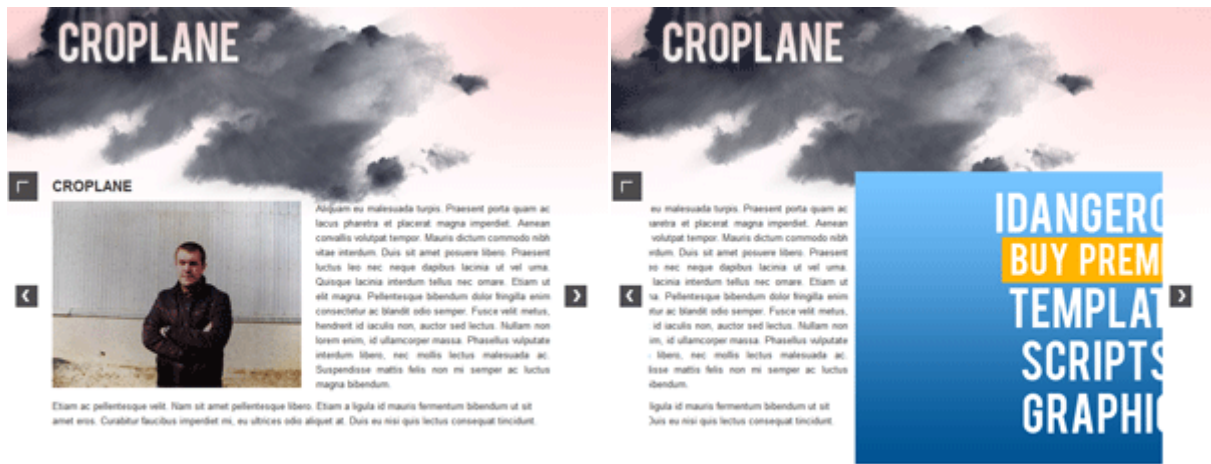
Default **Windowed Layout Type**. To use it just create a .php file with any HTML content inside and make link to it. Or call it with goTo() function where the **\$replaceContent** value is equal to "0"

8.2 Slideshow (HTML & JavaScript)

Let's see how the Slideshow Content Layout looks like and how it works:

First slide

Animated transition to the second slide



Second Slide



Now let's look more closely at the HTML and JavaScript code to learn how to create your own slides. Open **content/slideshow.php** file:

```

<script type="text/javascript">
jQuery(document).ready(function() {
    $(".slide-right").mousedown(function() {
        if($(".show-slide:last").position().left !=0) {
            $(".show-slide").animate({left:"-=700px"},1500)
        }
    })
    $(".slide-left").mousedown(function() {
        if($(".show-slide:eq(0)").position().left !=0) {
            $(".show-slide").animate({left:"+=700px"},1500)
        }
    })
})
</script>
<div class="slideshow">
    
    
    <div class="show-slides">
        <div class="show-slide">
            --- Some Slide Content---
        </div>
        <div class="show-slide">
            --- Some Slide Content---
        </div>
        <div class="show-slide">
            --- Some Slide Content---
        </div>
    </div>
</div>

```

Handling script for the **slide-left** and **slide-right** images. For example, if you will click **slide-right** image, script checks if this is the last slide or not. If not, then the all slides will move to the left at 700px for 1.5 seconds.

The same code for the **slide-left** image.

Slide-left and **slide-right** images.

First slide content. It could by any content: HTML code, Image, Flash, JavaScript, PHP etc.

Second slide content. It could by any content: HTML code, Image, Flash, JavaScript, PHP etc.

Third slide content. It could by any content: HTML code, Image, Flash, JavaScript, PHP etc.

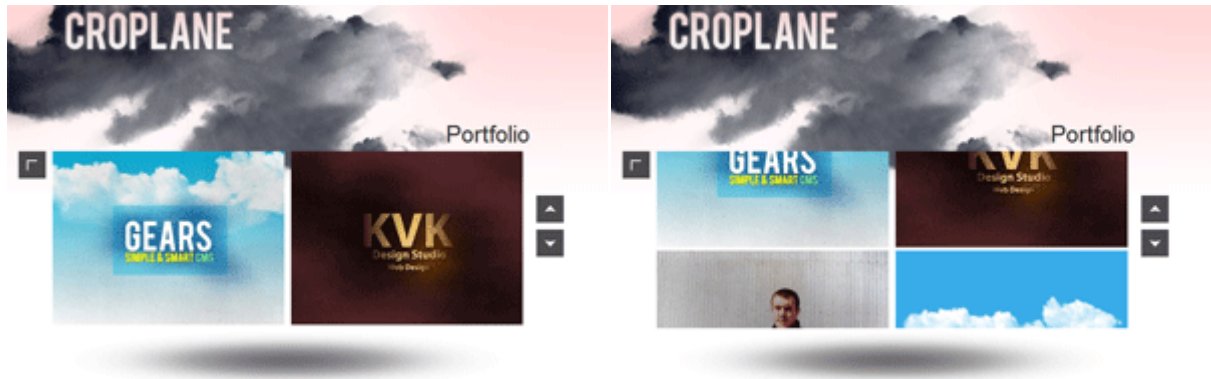
As you can see the slideshow section is very simple to use. If you want to add more slides, just add **div** block with the **class="show-slide"** with any content inside it. If you want to remove slide, remove the appropriate **div** block.

To customize slideshow section you have to use the **css/style.css** file and appropriate section there.

8.3 Portfolio (HTML & JavaScript)

First two works of portfolio

Animated Transition to the second two works of portfolio



Exposed portfolio work with its description.



So let's open content/portfolio.php file and look at the HTML and JavaScript code of portfolio (on the next page)


```

1 <script type="text/javascript">
2 jQuery(document).ready(function() {
3     $("#folio-top").mousedown(function() {
4         if($("#folio-slide:eq(0)").position().top !=0) {
5             $("#folio-slide").animate({top:"+=260px"},1500)
6         }
7     })
8     $("#folio-down").mousedown(function() {
9         if($("#folio-slide:last").position().top !=0) {
10             $("#folio-slide").animate({top:"-=260px"},1500)
11         }
12     })
13     $(".folio_img").click(function() {
14         $clicked=$(this)
15         $folio_left = $clicked.offset().left
16         $folio_left2 = $folio_left + $clicked.width()+10
17         $folio_top = $clicked.offset().top
18         if($clicked.index()==0) {
19             $(this).nextAll(".folio_img").fadeOut(600,function(){
20                 $clicked.addClass("folio-active").css({position:"fixed",left:$folio_left,top:$folio_top}).delay(400).animate({top:"50px",borderWidth:"5px"},600)
21                 $clicked.next(".folio_desc").addClass("folio-active").css({position:"fixed",left:$folio_left2,top:"50px"}).delay(1000).fadeIn(600)
22                 $('#close_box').css({position:"fixed",left:$folio_left-50,top:"50px"}).delay(1300).fadeIn(300)
23             })
24         }
25         if($clicked.index()==2) {
26             var $prevleft = $(".folio_img:eq(0)").offset().left
27             var $prevleft2 = $prevleft + $clicked.width()+10
28             $(this).prevAll(".folio_img").fadeOut(600,function(){
29                 $clicked.addClass("folio-active").css({position:"fixed",left:$folio_left,top:$folio_top}).delay(400).animate({top:"50px",left:$prevleft,borderWidth:"5px"},600)
30                 $clicked.next(".folio_desc").addClass("folio-active").css({position:"fixed",left:$prevleft2,top:"50px"}).delay(1000).fadeIn(600)
31                 $('#close_box').css({position:"fixed",left:$prevleft-50,top:"50px"}).delay(1300).fadeIn(300)
32             })
33         }
34         $("#dark-layer").delay(600).fadeTo(300,0.8)
35     })
36     $('#close_box,#dark-layer').click(function() {
37         $("#div.folio-active").fadeOut(400,function(){
38             $(this).removeClass('folio-active')
39         })
40         $('#close_box').fadeOut(300)
41         $(".img.folio-active").delay(400).animate({left:$folio_left,top:$folio_top,borderWidth:"0px"},600,function(){
42             $(".folio_img").not($clicked).fadeIn(600,function(){
43                 $clicked.css({position:"relative",left:0,top:0})
44             })
45         })
46         $("#dark-layer").delay(1000).fadeOut(300,function(){ $clicked.removeClass('folio-active') })
47         $(".folio_img").not($clicked).delay(1300).fadeIn(600)
48     })
49 })
50 </script>

```

Handling script for the **slide-up** and **slide-down** images. For example, if you clicked **slide-up** image, script checks if this is the first slide or not. If not, then the all folio slides will move to the top at 260px for 1.5 seconds.

If you clicked **slide-down** image, script checks if this is the last slide or not. If not, then the all folio slides will move to the down at 260px for 1.5 seconds.

General function for the portfolio animation. First of all, it is getting folio image position. We need it to return folio image on its place in future. Then we are checking if this first or second image in folio slide.

This case is for the first image in folio slide. Here is the animation queue to expose portfolio work: image animation, positioning and fade effect for the work description, positioning of "close" image

This case is for the second image in folio slide. Here is the animation queue to expose portfolio work: image animation, positioning and fade effect for the work description, positioning of "close" image

Adding the "Dark Layer" with fade animation

This function is used to return folio image on its initial place, and to close **exposed** layers. This function will be executed when you click at the **close** image or click on the **dark layer**.

Now let's look at the HTML code to learn how to create your own folio slides:

```

1 <div class="portfolio">
2   <div class="folio-slides">
3     <div class="folio-slide">
4       
5       <div class="folio_desc">
6         --- Work Description here ---
7       </div>
8       
9       <div class="folio_desc">
10        --- Work Description here ---
11      </div>
12    </div>
13    <div class="folio-slide">
14      
15      <div class="folio_desc">
16        --- Work Description here ---
17      </div>
18      
19      <div class="folio_desc">
20        --- Work Description here ---
21      </div>
22    </div>
23  </div>
24  
26  
28  
30  <div id="dark-layer"></div>
31  

```

First folio slide

First folio slide, image of the first work

Description of the first work (HTML content)

First folio slide, image of the second work

Description of the second work (HTML content)

Second folio slide

Close image, used in exposed view

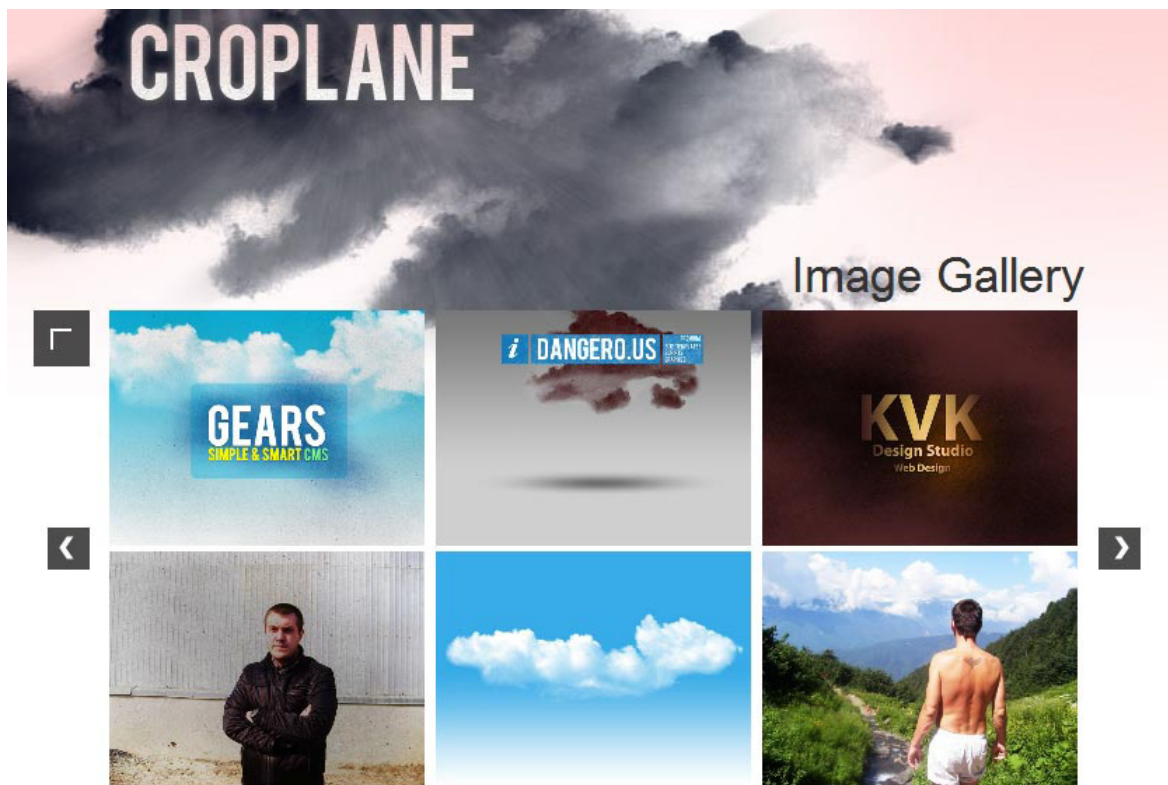
Arrow images, used to slide between folio slides

Dark Layer

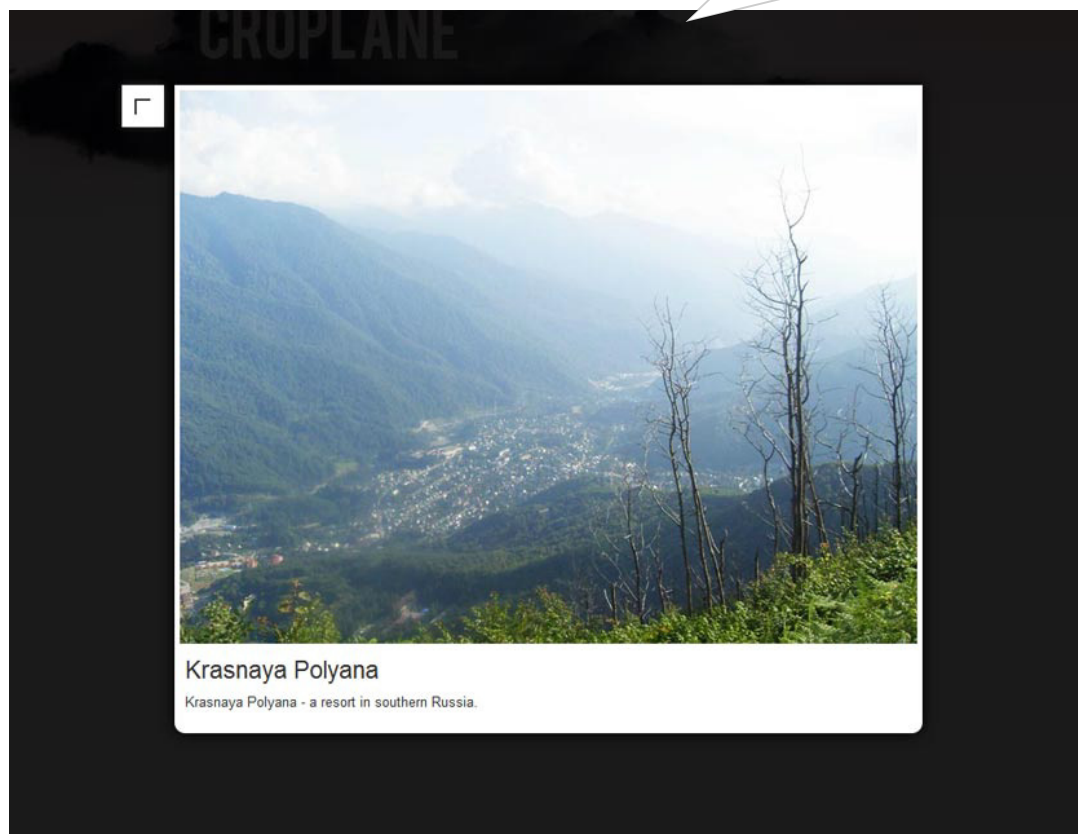
As you see it's very easy to create new folio works. First of all you have to create "**folio-slide**" div (div with a "folio-slide" class). You can put two works into the each slide. Then inside the **folio-slide** div you have to create you works in the following order:

1. Add image of the first work using the tag with a **folio_img** class ()
2. Add div block with a **folio_desc** class (<div class="folio_desc">...</div>). Put the description of the first work inside this block (It could be any HTML or Flash content)
3. Add image of the second work using the tag with a **folio_img** class ()
4. Add div block with a **folio_desc** class (<div class="folio_desc">...</div>). Put the description of the second work inside this block (It could be any HTML or Flash content)

8.4 Image Gallery (HTML & JavaScript)



Exposed image/photo with its description



Let's open **content/image_gallery.php** file and look at the JavaScript & HTML code of it:

```

1 <script type="text/javascript">
2 jQuery(document).ready(function() {
3     $(".image_s").hover(
4         function() {
5             if(!$(this).hasClass("img_active") && !$(this).hasClass("faded")) {
6                 $(this).fadeTo(200,0.5)
7             }
8         },
9         function() {
10             if(!$(this).hasClass("img_active") && !$(this).hasClass("faded")) {
11                 $(this).fadeTo(200,1)
12             }
13         })
14     $(".slide-right").mousedown(function() {
15         if($(".g-slide:last").position().left !=0) {
16             $(".g-slide").animate({left:"-=700px"},1500)
17         }
18     })
19     $(".slide-left").mousedown(function() {
20         if($(".g-slide:eq(0)").position().left !=0) {
21             $(".g-slide").animate({left:"+=700px"},1500)
22         }
23     })
24     $(".image_s").click(function() {
25         if(!$(this).hasClass("img_active") && !$(this).hasClass("faded")) {
26             $(this).addClass("img_active")
27             $(".image_s:not(.img_active)").addClass("faded").fadeTo(600,0.01)
28             $(".slide-left,.slide-right,.to-home-page,.page-title").fadeOut(600)
29             var $desc = $(this).next(".img_desc").html()
30             if($desc != null) {
31                 var $description = $desc
32             }
33             else var $description = ""
34             if($(this).attr('title') != '' ) {
35                 var $img_title = '<h3 class="img_title">'+$(this).attr('title')+'</h3>'
36                 $(".img_description").html($img_title+$description)
37             }
38             else {
39                 $(".img_description").html($description)
40             }
41             $(".big_image").html('')
43         }
44     })
45     $("#close_box, #dark-layer").click (function(){
46         closeLightbox()
47     })
48 })
49 function loadImage() {
50     var $left = (700 - $("#lightbox").width())/2;
51     $(".img_active").fadeTo(600,0.5,function() {
52         $("#lightbox").css({left:$left+"px"}).delay(300).fadeIn(600,function() {
53             $("#dark-layer").fadeTo(600,0.9)
54         })
55     })
56 }
57 function closeLightbox() {
58     $("#dark-layer, #lightbox").fadeOut(600)
59     $(".image_s").removeClass("img_active").removeClass("faded").delay(600).fadeTo(600,1)
60     $(".slide-left,.slide-right,.to-home-page,.page-title").delay(900).fadeTo(600,1)
61 }
62 </script>

```

Blink effect for images. First function is for the **mouseover** action, and the second one for the **mouseout** action.

Handling script for the **slide-left** and **slide-right** images. For example, if you will click **slide-right** image, script checks if this is the last slide or not. If not, then the all slides will move to the left at 700px for 1.5 seconds.

The same code for the **slide-left** image.

General function for the gallery animation. First of all, all images will fade out excepts the image you have clicked.

Then the scripts checks for the photo description and for the **title** attribute. If they are not empty, they will be added to the div block with an **"img_description"** class.

After that, image with the attributes same as you've clicked will be added to the div block with **"big_image"** class.

This function is used to close exposed view. It will be executed when you click at the **close** image or click on the **dark layer**.

loadImage() function is used to expose image/photo, which will be shown after the animation queue.

closeLightbox() function is used to close the exposed through the animation queue.

Now let's look at the HTML code to learn how to create your own gallery slides:

```
1 <div id="gallery">
2   
3   
4   <div class="gallery-slides">
5     <div class="g-slide">
6       
7       <div class="img_desc"><p>Photo 1 Description</p></div>
8       
9       <div class="img_desc"><p>Photo 2 Description</p></div>
10      
11      <div class="img_desc"><p>Photo 3 Description</p></div>
12      
13      
14      
15    </div>
16    <div class="g-slide">
17      
18      
19      
20      
21      
22      
23    </div>
24  </div>
25 </div>
26 <div id="lightbox" style="display:none">
27   
29   <div class="big_image"></div>
30   <div class="img_description"></div>
31 </div>
32 <div id="dark-layer"></div>
```

Arrow images, used to slide between folio slides

Gallery slide. Each slide contains 6 photos/images.

First photo

Description of the first photo

Lightbox section. Exposed image with its description will be added here

Close image, used in exposed view

Dark Layer

As you see it's very easy to create or remake gallery. First of all you have to create **Gallery Slide** into the div block with a **"gallery-slides"** class. You should use the following formatting:

```
<div class="g-slide">
  --- Images will be here ---
</div>
```

Then put into the Gallery Slide image using the `` tag with an **"image_s"** class. Don't forget to specify image title with the **title** attribute. If you want to add image description, you have to add right after the image the div block with an **"image_desc"** class. You can put into it any HTML content. You can add up to 6 images(photos) into one Gallery Slide.

8.5 Offer Section

Here is the offer section:



Let's look at the HTML code of the offer section (open content/offer.php file):

```
1 
2 
```

As you see it is very very simple. There is only two images: one with offer text and one is the shadow image. To change the offer image to yours own just change the **src** attribute in the first **** tag.

9. Ajax Contact Form

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

```
1 <h2 class="content-heading">Contacts</h2>
2 <div id="message_status"></div>
3 
4 <p>You can contact us using the following contact form:</p>
5 <form action="javascript:sendMessage()" method="post" id="contact_form">
6   <p>
7     <label>Your Name:<br />
8     <input type="text" size="40" name="name" id="contact_name" />
9   </label>
10 </p>
11 <p>
12   <label>Email*:<br />
13   <input type="text" size="40" name="email" id="contact_email" onblur="validateEmail()" />
14 </label>
15 </p>
16 <p>
17   <label>Subject:<br />
18   <input style="color:#777" type="text" size="40" id="contact_subject" name="subject"
19   value="Message From CROPLANE" readonly="readonly" />
20 </label>
21 </p>
22 <p>
23   <label>Message*:<br />
24   <textarea cols="60" rows="10" name="message" id="contact_message" onblur="
25   validateMessage()"></textarea>
26 </label>
27 <p>Fields marked with an asteriks(*) are required!</p>
28 <span class="button-l"><span class="button-m">
29   <input type="submit" name="submit" value="Send Message" />
30 </span></span>
31 </form>
```

Message with the mailing status will be added here when the mail will be sent.

When we'll click the "Send Message" button, the sendMessage() JavaScript function will be executed.

Remove the **readonly** attribute. It's used in the demo site.

Use the following formatting for the beautiful buttons:

```
<span class="button-l">
  <span class="button-m">
    <input type="submit" ... />
  </span>
</span>
```

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'])) $message = strip_tags($_POST['message']);
6  $mail_to = "info@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     <div class="mail-sent">
15     <h4>
16     Thanks! Your message has been successfully sent.</h4>
17     </div>
18     <?php
19     }
20     else {
21     ?>
22     <div class="mail-error">
23     <h4>
24     Sorry! Error occurred. Try again later please.</h4>
25     </div>
26     <?php
27     }
28     ?>

```

Checking for existence of these variables.

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

Standard headers for the e-mail message.

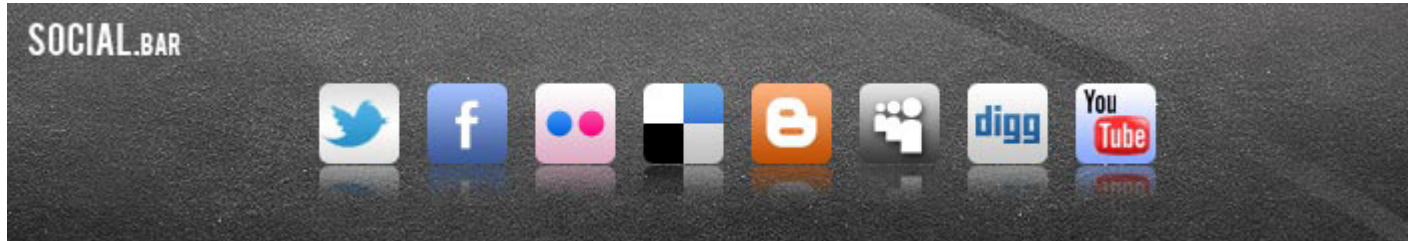
If the message was successfully sent, the script will return this HTML message.

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

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

10. Social.Bar

SOCIAL.BAR is a panel at the bottom of the home page, where you can see the animated social icons which you can link to your relevant social pages (profiles).



To edit Social.Bar just open the **includes/social.bar.php** file:

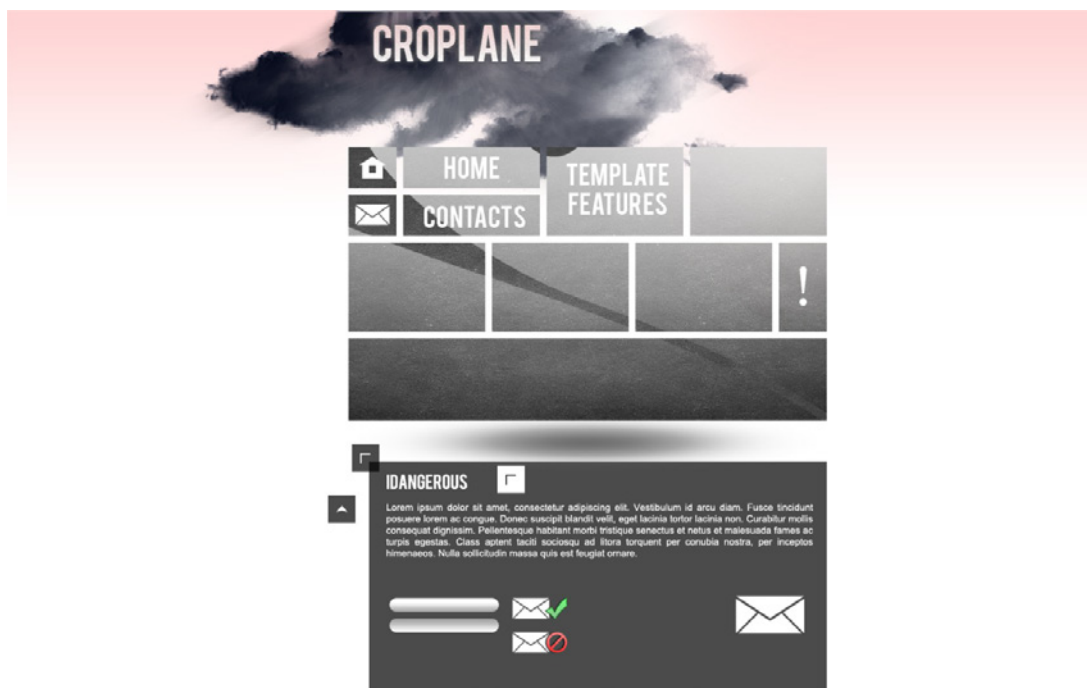
```

<div class="bar-icons">
  <a href="#">
    
    <span>Twitter</span>
  </a>
  <a href="#">
    
    <span>Facebook</span>
  </a>
  <a href="#">
    
    <span>Flickr</span>
  </a>
  <a href="#">
    
    <span>Delicious</span>
  </a>
  <a href="#">
    
    <span>Blogger</span>
  </a>
  <a href="#">
    
    <span>My&nbsp;Space</span>
  </a>
  <a href="#">
    
    <span>DIGG</span>
  </a>
  <a href="#">
    
    <span>You&nbsp;Tube</span>
  </a>
</div>
```

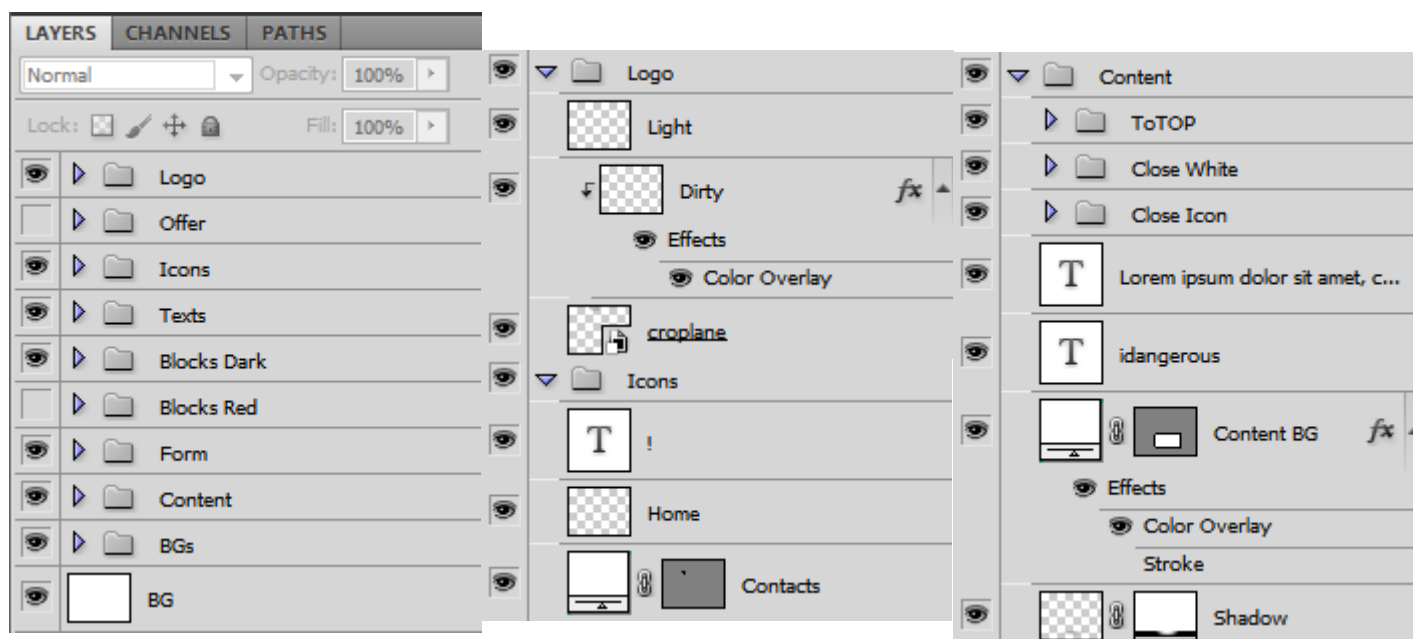
11. PSD Files & Structure

There are the two layered **.psd** files comes with the Croplane Template. You can locate them into the **psd/** folder from the downloaded archive. Using these files you can easily redesign Croplane. To edit this files you'll need an Adobe™ Photoshop™ (Recommended is CS4 or CS5 version)

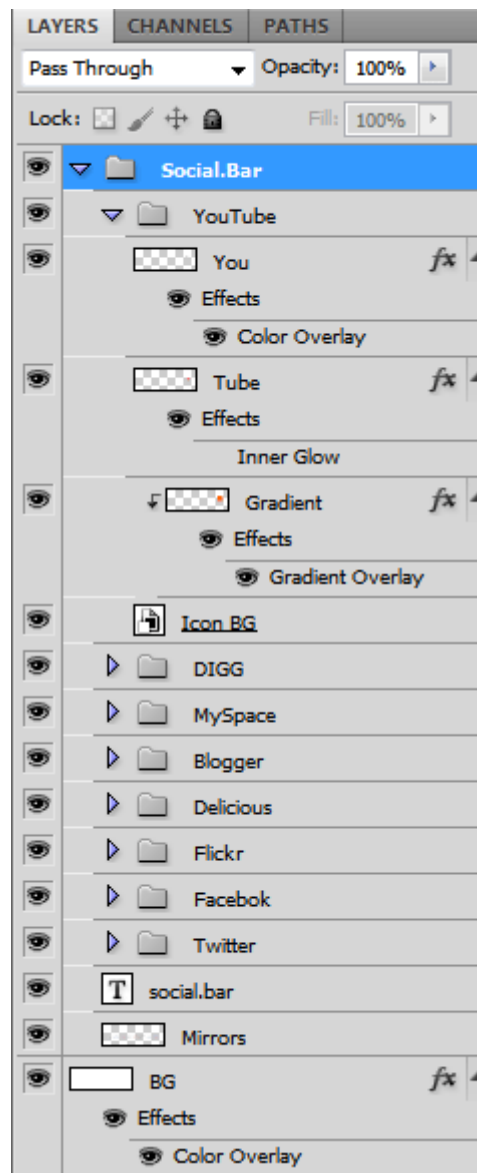
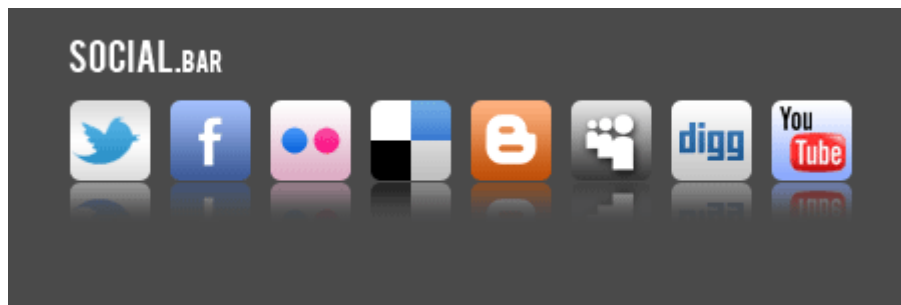
First **.psd** file is the **croplane.psd**. It contains all general layouts for template styling:



All layers in this file are well organized and easy to understand and redesign:



Second .psd file is the **social-bar.psd**. This file contains the Social.Bar layered icons



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 **Croplane** (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**.