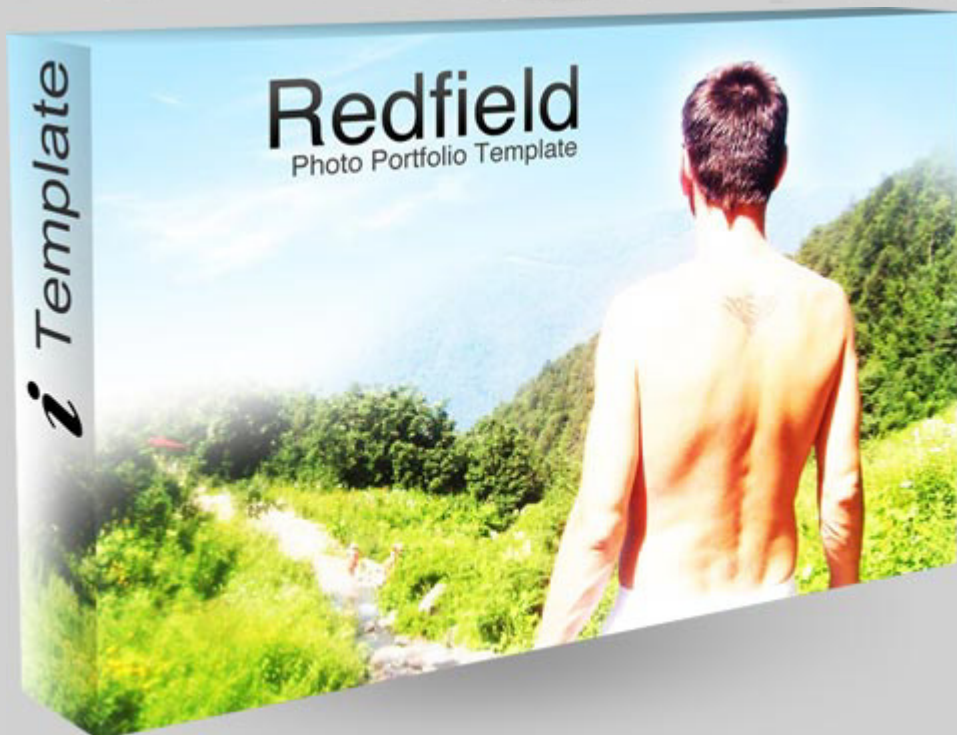




# DANGERO.US

PREMIUM  
SITE TEMPLATES  
SCRIPTS  
GRAPHICS



## **“Redfield – Photo Portfolio”**

### **Premium Template By iDango.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 “Redfield”
2. Redfield Files & Folders Structure
3. Installation
4. HTML Structure
5. Inner Linking (How it works?)
6. CSS Files and Structure
7. JavaScripts
8. Photo Gallery
  - 8.1 Photo Gallery (HTML & JavaScript)
  - 8.2 Image Controller (HTML & JavaScript)
9. Ajax Contact Form
10. Social.Bar
11. PSD Files and Structure
12. Licensing Terms

# 1. About “Redfield”

---

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

This template is intended for use as a photo portfolio with brief information about the author (photographer). The template is fully animated without the use of Flash technology. All animation realized through the use of jQuery and CSS3.

All content loaded on the fly, your users do not need to wait the reloading of entire site.

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

Photo Gallery has the great image controller. Using this controller you can move, resize and rotate the viewed photos with good looking animation.

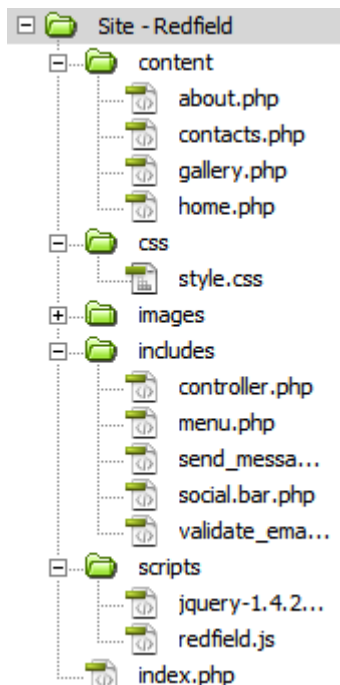
Also **Redfield** is powered with an Ajax Contact Form with validation. Here is a couple of screenshots:



## 2. Redfield Files Structure

---

Redfield files are located in the **redfield/** folder from the downloaded archive:



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

**css:** folder with a .css files

**images:** here you can find layouts images and photos

**includes:** two php scripts used for Ajax contact form, social.bar, image controller and menu html layouts

**scripts:** JavaScripts used in this template

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

## 3. Installation

---

To install **Redfield** template you have to copy all files & folders from the **redfield/** 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/PHP structure of Redfield 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-->
<div id="background" ></div>
<!--End of additional Background-->

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


<!--Main content section-->
<div class="wrapper">
    <div id="enter"></div>
    <div id="menu">
        <?php include("includes/menu.php") ?>
    </div>
    <div id="content">
        <div class="content-inner">
            <?php include("content/home.php") ?>
        </div>
    </div>
    <div id="photo-gallery">
        <?php include("content/gallery.php") ?>
        <?php include("includes/controller.php") ?>
    </div>
</div>
<!--End of main content section-->

<!--Social.Bar-->
<div id="social-bar">
    <?php include("includes/social.bar.php") ?>
</div>
<!--End of Social.Bar-->

<div id="swirls"></div>
<div id="white-layer"></div>

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

Enter button is displayed on the Home Page:



Content section. All files with html content will be loaded here

Here is the Photo Gallery section (chapter 8)

Social.Bar section (chapter 10)

These layers are used in the Photo Gallery

## 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-->
```

## Main Content Section

This is the main section of the site. Here is menu, content section and the photo gallery section.

### Menu

HTML content of the menu is included form the **includes/menu.php** file:

```
1 <a class="menu-link " href="home">Home</a>
2 <a class="menu-link" href="about">About Me</a>
3 <a class="menu-link" id="gallery" href="gallery">Photo Gallery</a>
4 <a class="menu-link" href="contacts">Contacts</a>
```

To create your own menu link, first of all you need to create <a> tag with the “menu-link” class. Then you have to create a .php file with HTML content in the **content/** folder. Name it for example **mypage.php**. Then add the **href** attribute with a value equal to the filename. It must be look like this:

```
<a class="menu-link" href="mypage">My Page</a>
```

Href attribute specifies the .php **filename**, which will be loaded from the **content/** folder.

Link with the id="gallery" works in other way, because the gallery is loaded by default, but it is hided.

## Content Section

This section is used to load HTML content (by Ajax) from the .php files located in **content/** folder. Home page content is included by default.

```
<div id="content">
    <div class="content-inner">
        <?php include("content/home.php") ??>
    </div>
</div>
```

## Photo Gallery Section

Photo gallery explanation is in the [chapter 8](#).



## Footer Section

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

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

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. 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/about.php** file directly from the simple link in the text.

In these cases, there is a JavaScript function **loadContent()**. This function has two attributes: **\$id** and **\$uri**.

**\$id** is used to determine type of loaded content – gallery or simple text content. If you wish to load the gallery page you should use it with the value equal to “gallery”. In other cases leave it blank.

**\$uri** attribute is used to specify the **filename**(only filename) of the file you want to load. If you want to load a **content/about.php** file you have to use \$uri attribute with the value equal to “about”.

So for this example we need to use the following formatting for the links:

```
<a href="about" onclick="loadContent('', 'about'); return false">About</a>
```

Any value could be here

Blank value must be here

This means that we want to load content from the **content/about.php** file

Used to prevent browser's default action for the links

**Example 2.** We want to load Photo Gallery page directly from the simple link in the text.

We should use the following formatting for the links:

```
<a href="gallery" onclick="loadContent('gallery', ''); return false">Gallery</a>
```

Any value could be here

This means that we want to load Photo Gallery content

Any value could be here

Used to prevent browser's default action for the links

## 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, fonts, link colors, layout images, logo, footer etc.*

```
/*-----Menu----- */
```

*This section contains rules for menu*

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

*Social.Bar icons styling*

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

*All rules for Image Gallery*

```
/*----- Image Controller -----*/
```

*Rules for Image Controller*

```
/*----- 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

---

Redfield template uses two JavaScript files.

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/redfield.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 (redfield.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 effects and rules for handling menu links, animation of Social.Bar, Ajax loader, Photo Gallery and Image Controller actions.

```
1 // JavaScript Document
```

```
2 $(function() {
```

```
3     $("#background").css({height:$(document).height()})
```

```
4     $(window).resize(function() {
```

```
5         setHeight()
```

```
6     })
```

```
7     /*-----Enter Button-----*/
```

```
8     $("#enter").hover(function() {
```

```
9         $(this).fadeTo(200,0.4)
```

```
10     },
```

```
11     function() {
```

```
12         if(!$(this).hasClass('entered')) {
```

```
13             $(this).fadeTo(200,1)
```

```
14         }
```

```
15     })
```

```
16     $("#enter").click(function() {
```

```
17         $(this).addClass("entered").fadeOut(600)
```

```
18         intro()
```

```
19     })
```

```
20     /*-----Menu-----*/
```

```
21     $(".menu-link").css({opacity:"0.4"}).hide()
```

```
22     $(".menu-link").hover(function() {
```

```
23         $(this).fadeTo(300,1).animate({marginLeft:"-10px"},{ queue:false, duration:200 })
```

```
24     },
```

```
25     function() {
```

```
26         $(this).fadeTo(300,0.4).animate({marginLeft:"0px"},{ queue:false, duration:200 })
```

```
27     })
```

```
28     $(".a.menu-link").click(function(e) {
```

```
29         e.preventDefault()
```

```
30         $(".menu-link").not($(this)).removeClass("active-link")
```

```
31         $(this).addClass("active-link")
```

```
32         loadContent($(this).attr("id"),$(this).attr("href"))
```

```
33     })
```

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

```
35     $("body").ajaxStart(function() {
```

```
36         $(this).prepend('<div class="ajax-loader"><h3>Loading...</h3></div>');
```

```
37     })
```

```
38     $("body").ajaxComplete(function() {
```

```
39         $(".ajax-loader").remove();
```

```
40     })
```

```
41     /*-----Social Bar-----*/
```

```
42     $(".bar-heading").click(function() {
```

```
43         var $bar=$("#social-bar")
```

```
44         if ($bar.css('left')=="-130px") {
```

```
45             $bar.animate({left:'0px'},300)
```

```
46         }
```

```
47         else $bar.animate({left:"-130px"},300)
```

```
48     })
```

```
49     $(".bar-icons a").hover(
```

```
50         function() {
```

```
51             $(this).children("span").fadeIn(200)
```

```
52             $(this).animate({top:"-5px"},200)
```

```
53         },
```

```
54         function() {
```

```
55             $(this).animate({top:"0px"},200)
```

```
56             $(this).children("span").fadeOut(200)
```

```
57         }
```

```
58     })
```

```
59     /*-----Photo Gallery-----*/
```

```
60     /*--- See in chapter 8.1 ---*/
```

```
61     /*-----Image Controller-----*/
```

```
62     /*--- See in chapter 8.2 ---*/
```

```
63 }
```

Assigning the background height equal to the height of the document.

Fade effect for the Enter button

When the Enter button is clicked than the **intro()** function will be executed

Used to initialize menu links. By default they will be hidden and faded to opacity 0.4.

Animation for menu links. When you'll hover on it, it will slide left and fade to opacity 1 for 0.2 seconds

When you'll take away mouse from it, it will slide back and fade to opacity 0.4

Menu links handling. When you'll click the menu link the **loadContent()** function will be executed.

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.

Animation for Social.Bar block. When we'll click the Social.Bar image, then the social bar block with icons will be shown using the slide animation for 0.3 seconds.

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

See the Photo Gallery and Image Controller scripts in chapter 8

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, for image gallery.

```

128 function setHeight() {
129     $("#background").css({height:$(document).height()})
130 }
131 function showImage($newImg) {
132     var $active = $(".photos img.active")
133     if ($active.index()==$newImg.index()) return false
134     else {
135         if($newImg.attr('title') == undefined) return false
136         else {
137             $active.fadeOut(600).removeClass("active")
138             $newImg.fadeIn(600).addClass("active")
139             $(".photo-title").fadeOut(300,function() {
140                 $(".photo-title").html($newImg.attr('title')).fadeIn(300)
141             })
142             $(".photo-descr").fadeOut(300,function() {
143                 $(".photo-descr").html($(".descriptions div:eq("+ $newImg.index()+")").html()).fadeIn(300)
144             })
145         }
146     }
147 }
148 function loadContent($id,$uri){
149     var $uri = "content/"+$uri+".php";
150     if($id != "gallery") {
151         jQuery.get($uri, function (content) {
152             $(".content").fadeOut(600,function() {
153                 $(".content-inner").html(content)
154                 $(".content").delay(300).fadeIn(600,function() {
155                     setHeight()
156                 })
157             })
158         })
159     }
160     else {
161         $(".content,#menu,#logo").fadeOut(600)
162         $(".white-layer").delay(600).fadeTo(600,0.5,function() {
163             $(".photo-gallery").fadeIn(300)
164             $(".img-info").delay(300).fadeIn(600)
165             $(".photo-box").delay(600).fadeIn(600,function() {setHeight()})
166             $(".film").delay(900).fadeIn(600)
167             $(".controller").delay(1500).fadeIn(300)
168             $(".swirls").delay(1500).fadeIn(600)
169         })
170     }
171 }
172 function toHome(){
173     $(".photo-gallery,.img-info,.photo-box,.film,.controller,#swirls").fadeOut(600)
174     $(".white-layer").delay(600).fadeOut(600,function() {
175         $(".logo").fadeIn(600)
176         intro()
177     })
178 }
179 function intro(){
180     $(".menu,#content").delay(700).fadeIn(600,function() {
181         for ($i=0;$i<$(".menu-link:last").index();$i++) {
182             $(".menu-link:eq("+ $i+")").delay(($i)*300).fadeIn(600,function() {setHeight()})
183         }
184     })
185     $(".footer").delay(1600).fadeIn(300)
186 }

```

setHeight() function is used for setting the Background height after the browser is resized

showImage() function is used in Photo Gallery to show images. Explanation in the chapter 8

loadContent() function is used for Ajax loading of the .php files with HTML content. It is also used to load the Photo Gallery page.

Ajax GET request

After the AJAX request, the HTML content of the file will be loaded to the div block with class "content-inner". After that the new window will be shown through the animation queue. And the setHeight() function will be executed.

This case used to load Photo Gallery page through the animation queue.

toHome() function provides an opportunity to return to the Home Page through the animation queue. This function is used in Photo Gallery.

Intro() function will be executed when you'll click the Enter button. This function is used to show home page through the animation queue.

```

187 function validateMessage() {
188     if (($("#contact_message").val()).length < 10) {
189         $("#contact_message").addClass("required")
190     }
191     else $("#contact_message").removeClass("required")
192 }
193 function validateEmail() {
194     var email = ($("#contact_email").attr("value"));
195     jQuery.post("includes/validate_email.php",{email: email},
196         function (data) {
197             if (data != 1) {
198                 $("#contact_email").addClass("required");
199             };
200             if (data == 1) {
201                 $("#contact_email").removeClass("required");
202             };
203         }
204     );
205 }
206 function sendMessage() {
207     validateEmail()
208     validateMessage()
209     if (!$("#contact_email,#contact_message").hasClass("required")) {
210         name = ($("#contact_name").attr("value"));
211         email = ($("#contact_email").attr("value"));
212         subject = ($("#contact_subject").attr("value"));
213         message= ($("#contact_message").val().replace(/<\/?[>]+>/gi, ''));
214         jQuery.post("includes/send_message.php",{
215             name : name, email : email, subject : subject, message : message},
216             function (status) {
217                 $("#message_status").slideDown(300).html(status);
218             }
219         );
220     }
221 }
222 function rotate($angel) {
223     $(".photo-box").css({
224         "-moz-transition-property":"-moz-transform",
225         "-moz-transition-duration": "1s",
226         "-webkit-transition-property": "-webkit-transform",
227         "-webkit-transition-duration": "1s",
228         "-o-transition-property": "-o-transform",
229         "-o-transition-duration": "1s",
230         "transition-property": "transform",
231         "transition-duration": "1s",
232         "-o-transform":"rotate("+ $angel +"deg)",
233         "-webkit-transform":"rotate("+ $angel +"deg)",
234         "-moz-transform":"rotate("+ $angel +"deg)",
235         "transform":"rotate("+ $angel +"deg)"
236     })
237     $(".rotate-ccw").attr({"onClick":"rotate("+ ($angel-5) +"")"})
238     $(".rotate-cw").attr({"onClick":"rotate("+ ($angel+5) +"")"})
239 }

```

**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.

**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**"

**rotate()** function provides an opportunity to rotate photos using the image controller. It's based on the CSS3 properties "transition" and "transform". It's mean that when you'll click **rotate-right** image, then the css property "**transform: rotate(10deg)**" will be added to the div block with a "photo-box" class. If you'll click **rotate-left** image, then the css property "**transform: rotate(-10deg)**" will be added etc. And after that, the onClick attribute will be changed to the same rotate() function, but with a value larger on the 5 degrees from the former ones.





```

1 
3 
5 <div class="img-info">
6   <h1 class="photo-title"></h1>
7   <div class="photo-descr"></div>
8 </div>
9 <div class="film">
10   <div class="film-thumbs">
11     <div class="thumbs">
12       
13       
14       
15     </div>
16   </div>
17   
19   
21 </div>
22 <div class="photo-box">
23   <div class="photos">
24     
25     
26     
27   </div>
28   
30   
32   
34 </div>
35 <div class="descriptions">
36   <div> --- HTML description of 1st Photo ---</div>
37   <div> --- HTML description of 2nd Photo ---</div>
38   <div> --- HTML description of 3rd Photo ---</div>
39 </div>

```

**toHome** image (button). When you'll click it, the **toHome()** function will be executed. And you'll get to the home page through the animation queue.

Photo Gallery Logo

This block is used to display information about selected photo: its title and the HTML description.

Photo Chooser block. It's look like a photo film.

Small thumbs of photos. It's recommended to use thumbs with a 70px in width.

**Film-left** and **film-right** arrows are used to scroll the film.

Container for the full size photos.

**Next** and **Previous** images are used to navigate between photos

Image of the shadow

Container for the photo descriptions

As you can see it's very easy to use your own photos in this Photo Gallery. First thing you have to do, is to create small thumbs icons for your photos and put them into the div block with a **"thumb"** class (It's recommended to use thumbs with a 70px in width) using the usual **<img>** tag.

Then you have to put all full size photos into the div block with a **"photos"** class in the same order as the thumbs icons. You have to specify the **"active"** class for the first photo. Also don't forget to specify the **"title"** attribute for all your photos. It will be shown in the Photo Information section.

And the final thing you have to do is to write the description for your photos. You have to create div blocks (with any HTML content inside) into the div with a **"descriptions"** class in the same order as the thumbs icons. These descriptions will be show in the Photo Information section.

Now let's look at the JavaScript used in Photo Gallery from the **scripts/redfield.js** file:

```

61  /*-----Photo Gallery-----*/
62  $(".thumbs img,.blink").hover(
63    function(){$(this).fadeTo(300,0.5)},
64    function(){$(this).fadeTo(300,1)}
65  )
66  $("#toHome").click(function(){toHome()})
67  $(".photo-title").html($(".photos img:eq(0)").attr('title'))
68  $(".photo-descr").html($(".descriptions div:eq(0)").html())
69  $(".film-left").mousedown(function(){
70    $(".thumbs").animate({left:["+=1000px","linear"]},8000)
71  })
72  $(".film-right").mousedown(function(){
73    $(".thumbs").animate({left:["-=1000px","linear"]},8000)
74  })
75  $(".film-left,.film-right").mouseup(function(){
76    $(".thumbs").stop(true,false)
77  })
78  $(".photo-next").mousedown(function(){
79    var $newImg = $(".photos img.active").next('img')
80    showImage($newImg)
81  })
82  $(".photo-prev").mousedown(function(){
83    var $newImg = $(".photos img.active").prev('img')
84    showImage($newImg)
85  })
86  $(".thumbs img").click(function(){
87    var $newImg = $(".photos img:eq("+$(this).index()+")")
88    showImage($newImg)
89  })

```

Blink effect for the Thumbs. It means that when we direct the mouse on it, it fades to opacity 0.5 for 300ms. And then it fades to opacity 1 for 300ms.

When you'll click the toHome image, the **toHome()** function will be executed.

When the gallery is loaded, the title and description of the first image will be added into the appropriate sections of the Photo Information Section

Animation used in the Film section. It allows you to scroll film to the right and to the left

Handling for the Next and Previous images(buttons). When you'll click on these images, the **showImage()** function will be executed to show new Photo.

Handling for the thumb icons. When you'll click on the thumb, the **showImage()** function will be executed to show new Photo.

```

131 function showImage($newImg) {
132   var $active = $(".photos img.active")
133   if ($active.index()==$newImg.index()) return false
134   else {
135     if($newImg.attr('title') == undefined) return false
136     else {
137       $active.fadeOut(600).removeClass("active")
138       $newImg.fadeIn(600).addClass("active")
139       $(".photo-title").fadeOut(300,function(){
140         $(".photo-title").html($newImg.attr('title')).fadeIn(300)
141       })
142       $(".photo-descr").fadeOut(300,function(){
143         $(".photo-descr").html($(".descriptions div:eq("+newImg.index()+")").html()).fadeIn(300)
144       })
145     }
146   }
147 }

```

**showImage()** function gets the Index Number of the image to show. Firstly it checks is the new image not the same as the active image. If not, it checks if the new image has the title. If it has the title, the active image will fade out for 600ms and the new image will fades in for 600ms. Then the Photo Title and the Photo Description will fade out for 300ms, after that the new values of title and description (which are taken from the new image title and the description div with the Index Number the same as the index number of new image) will be added to the appropriate div blocks. After that the Photo Title and the Photo Description will fade in for 300ms.

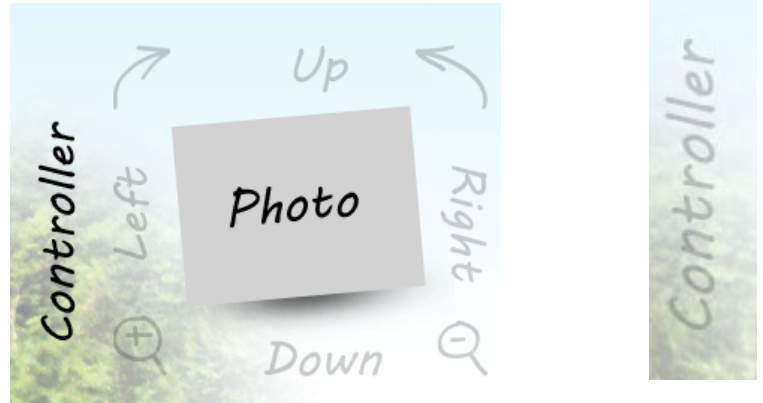


## 8.2 Image Controller (HTML & JavaScript)

---

To use Image Controller you have to click this button in the right on the screen

And it will be looked like this



**Image Controller** allows you to rotate photo, move it horizontally, vertically, enlarge and reduce its size.

Rotation feature is realized using the CSS3 properties "transition" and "transform", which are fully supported in the latest versions of Opera, Safari and Chrome browsers. Latest versions of 3<sup>rd</sup> FireFox will rotate image, but without animation. (It will work in the 4<sup>th</sup> version of FireFox). Internet Explorer 8 and older versions don't support this property. But it must be available in the Internet Explorer 9.

Now let's look at the HTML code of the Image Controller (**includes/controller.php**) (see the next page):

Here is the HTML code of the Image Controller:

```
1 <div class="controller">
2 
3 
4 
5 
6 
7 
8 
9 
10 
11 
12 </div>
```

Image Controller image. It's used to open Image Controller.

Image of the improvised Photo

These four images are used to move photo

Used to enlarge and reduce photo size

Last two images are used to rotate photo.

As you can see the last two images (used for photo rotation) have the **onclick** function **rotate()**. Default photo rotation angle is (-5) degrees. Rotate function provides an opportunity to rotate photo for 5 degrees in the each direction for one click. After the photo will be rotated, the **onclick** values will be changed by JavaScript.

For example, if the photo rotation is (-5) degrees (default), the **onclick** values will be the same as on the image above. After we will click the rotate-clockwise button, the photo will be rotated to 0 degrees. And the **onclick** values will be changed to these ones:

```


```

Now let's look at the JavaScript code of the Image Controller:

```
90  /*-----Image Controller-----*/
91  $(".imgController").click(function() {
92      var imContr=$("#controller")
93      if (imContr.css('right')=="-250px") {
94          imContr.animate({right:'20px'},300)
95      }
96      else imContr.animate({right:'-250px'},300)
97  })
98  $(".controller *:not(.c-photo),.bar-heading,#toHome").hover(
99      function() {$(this).fadeTo(300,1)},
100     function() {$(this).fadeTo(300,0.2)}
101 )
102 $(".move-left").mousedown(function() {
103     $(".photo-box").animate({left:["-=1000px","linear"]},4000)
104 })
105 $(".move-right").mousedown(function() {
106     $(".photo-box").animate({left:["+=1000px","linear"]},4000)
107 })
108 $(".move-up").mousedown(function() {
109     $(".photo-box").animate({top:["-=1000px","linear"]},4000)
110 })
111 $(".move-down").mousedown(function() {
112     $(".photo-box").animate({top:["+=1000px","linear"]},4000)
113 })
114 $(".move-left,.move-right,.move-up,.move-down").mouseup(function() {
115     $(".photo-box").stop(true,false)
116 })
117 $(".zoom-in").mousedown(function() {
118     $(".photo-box,.photos,.photos img.active").animate({width:["+=50px","linear"]},300,function() {
119         $(".photos img:not(.active)").css({width:$(".photos img.active").width()})
120     })
121     $(".photo-shadow").animate({width:["+=50px","linear"],bottom:["-=40px","linear"]},300)
122 })
123 $(".zoom-out").mousedown(function() {
124     $(".photo-box,.photos,.photos img.active").animate({width:["-=50px","linear"]},300,function() {
125         $(".photos img:not(.active)").css({width:$(".photos img.active").width()})
126     })
127     $(".photo-shadow").animate({width:["-=50px","linear"],bottom:["+=40px","linear"]},300)
128 })
```

This function is used for animated appearance of the Image Controller

Blink effect for the Image Controller elements, also used for social bar and toHome buttons.

This function is used to move div block with a "photo-box" class when the move-left image is clicked.

The same as a previous one, but for the move-right image.

The same function for the move-up image

The same function for the move-down image

This function is used to cancel move animation when the mouse button is released

This function is used to reduce photo size. When you'll click the zoom-in image, div blocks with a "photos", "photo-box" classes and the image with an "active" class will be reduced for 50px in width for 300ms. And after that not active (hidden) photos will be also resized to the same width as the "active" photo width. Shadow image will be resized too.

This function is used to enlarge photo. When you'll click the zoom-in image, div blocks with a "photos", "photo-box" classes and the image with an "active" class will be enlarged for 50px in width for 300ms. And after that not active (hidden) photos will be also resized to the same width as the "active" photo width. Shadow image will be resized too.

## 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 Redfield" readonly="readonly" />
20 </label>
21 </p>
22 <label>Message*:<br />
23 <textarea cols="60" rows="10" name="message" id="contact_message" onblur=
24 "validateMessage()" "></textarea>
25 </label>
26 <p>Fields marked with an asterisk(*) are required!</p>
27 <span class="button-l"><span class="button-m">
28 <input type="submit" name="submit" value="Send Message" />
29 </span></span>
30
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.

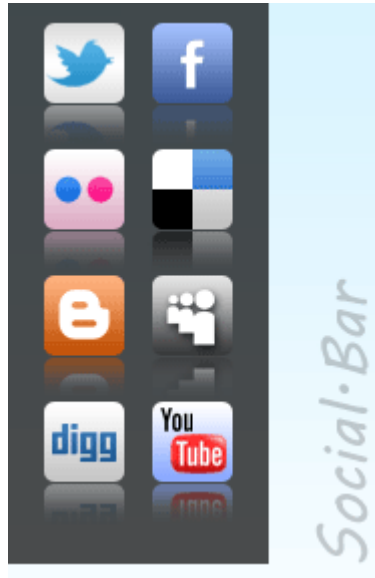
Otherwise script will return this HTML code with an error message.

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

## 10. Social.Bar

---

SOCIAL.BAR is a panel at the left top 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

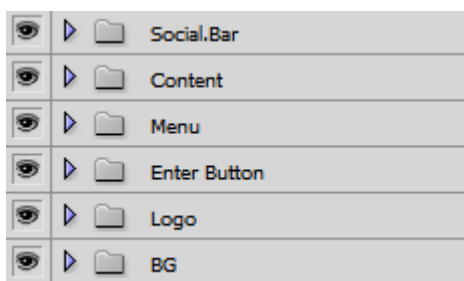
---

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

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

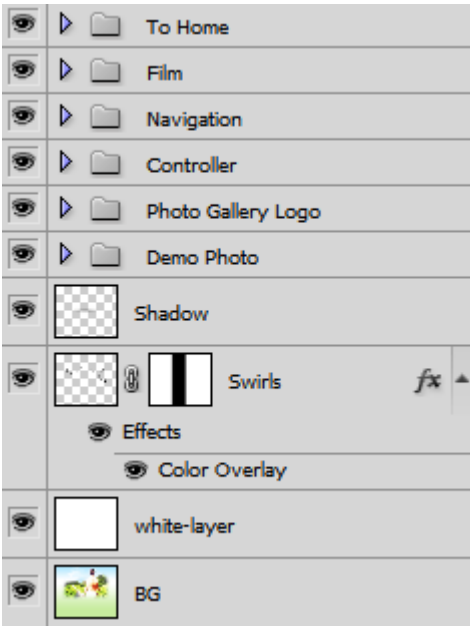


All layers in this file are well organized and easy to understand and redesign:

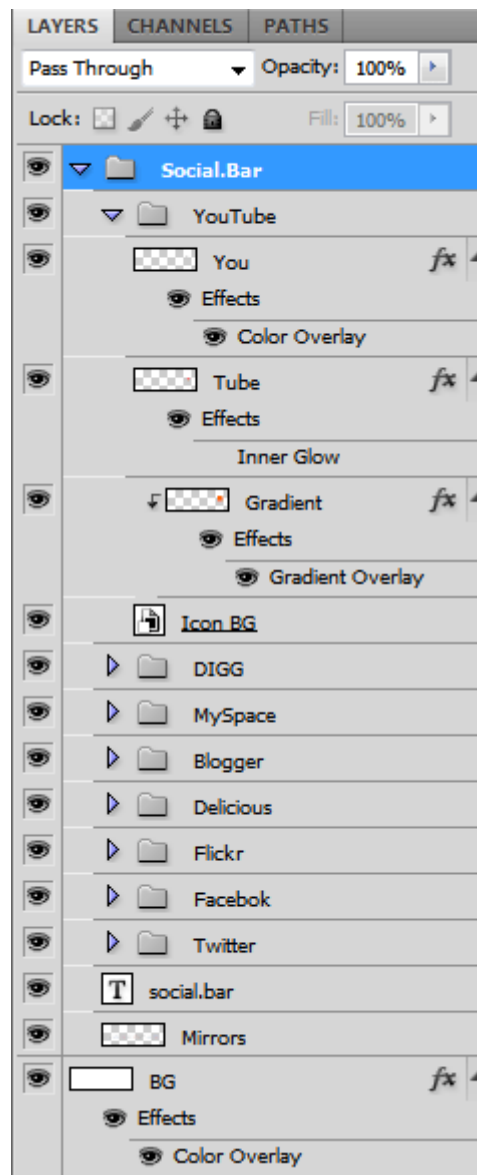
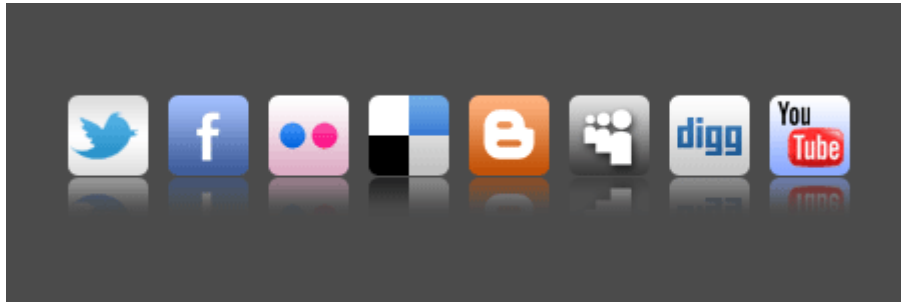




Second .psd file is the **gallery.psd**. It gives you full control to customize Photo Gallery page and all its elements:



Third .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 **Redfield Template** (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**.