



# **“Designia”**

## **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 “Designia”
2. Designia Files & Folders Structure
3. Installation
4. HTML Structure
5. Typography
6. Inner Linking (How it works?)
7. Portfolio
8. CSS Files and Structure
9. JavaScripts
10. Popup Window
11. Ajax Contact Form
12. PSD Files and Structure
13. Licensing Terms

# 1. About “Designia”

---

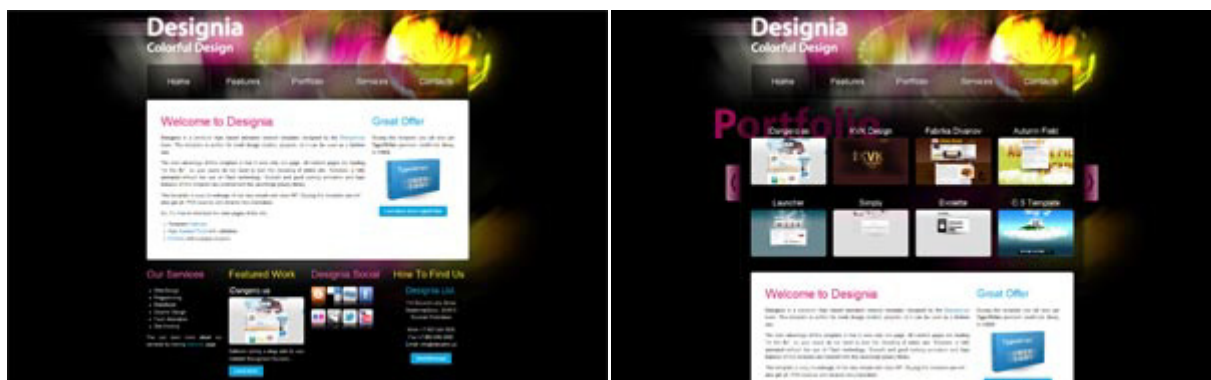
**Designia** is a premium Ajax based animated website template, designed by the iDangero.us team. This template is perfect for small design studios, projects, or it can be used as a fashion site.

The main advantage of this template is that it uses only one page. All content pages are loading "on the fly", so your users do not need to wait the reloading of entire site. Template is fully animated without the use of Flash technology. Smooth and good looking animation and Ajax features of this template are realized with the JavaScript jQuery library.

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

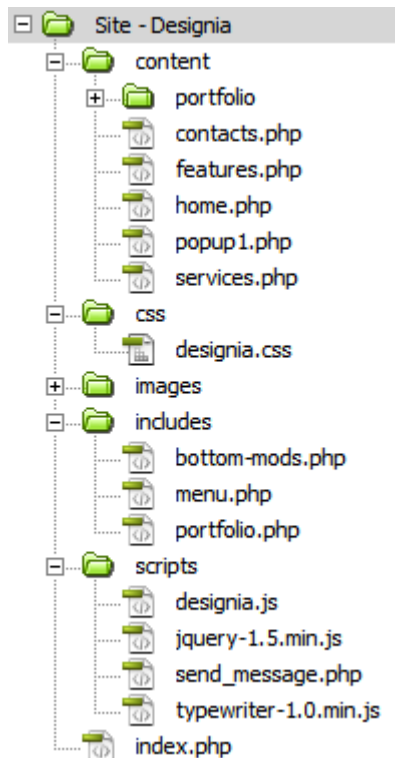
Also **Designia** is powered with an Ajax Contact Form with validation and a Popup Engine.

Here is a couple of screenshots:



## 2. Designia Files Structure

---



**Designia** files are located in the **designia/** 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 used in this Template

**includes:** php files with menu, bottom modules and portfolio content.

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

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

## 3. Installation

---

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

**Index.php** <body> structure:

```
11 <body>
12 <div class="wrapper">
13     <div class="ajax-loader">
14         Loading...
15     </div>
16     <div id="menu">
17         <?php include("includes/menu.php") ?>
18     </div>
19     <div class="portfolio-bg"></div>
20     <div id="portfolio">
21         <div class="folio-left"></div>
22         <div class="folio-right"></div>
23         <div class="folio-top">
24             <div class="folio-bot">
25                 <div class="folio-mid">
26                     <?php include("includes/portfolio.php") ?>
27                 </div>
28             </div>
29         </div>
30     </div>
31     <div class="main-content">
32         <div class="content-wrap">
33             <div class="dummy-content"></div>
34             <div class="content">
35                 <div class="content-inner">
36                     <?php
37                     if(empty($_GET['content'])) include ("content/home.php");
38                     else include ("content/".$_GET['content'].".php");
39                 <?>
40             </div>
41         </div>
42     </div>
43     <div id="bottomMods">
44         <?php include("includes/bottom-mods.php") ?>
45     </div>
46     <div class="footer">
47         <p align="center">&copy; 2011 Designia Premium Template by iDangero.us. All Rights Reserved.</p>
48     </div>
49 </div>
50 <div class="window">
51     <div class="close"></div>
53     <div class="pop-content"></div>
54 </div>
55 <div id="p-layer"></div>
56 </body>
```

Ajax loader container

Menu section, menu HTML content located in the **includes/menu.php** file

**potfolio-bg** contains big "portfolio" title image

Portfolio container

**folio-left** and **folio-right** divs contain arrow images to slide portfolio

Portfolio HTML content located in the **includes/portfolio.php** file

All pages content will be loaded in this div

Home page content must be included by default, but if the URL "content" variable is not empty then the appropriate file will be included here

Bottom modules section, their HTML content located in the **includes/bottom-mods.php** file

Footer Section.

Popup window

All "popup" content will be placed here

This is the dark layer, which will fade in when the popup window is opened

## Menu Section

---

Menu HTML content located in the **includes/menu.php** file:

```
1 <ul class="menu">
2   <li>
3     <span class="link-bg"></span>
4     <a href="?content=home">Home</a>
5   </li>
6   <li>
7     <span class="link-bg"></span>
8     <a href="?content=features">Features</a>
9     <span class="linkDescr" title="Learn more about Designia features"></span>
10  </li>
11  <li class="openFolio">
12    <span class="link-bg"></span>
13    <a href="#">Portfolio</a>
14    <span class="linkDescr" title="Check out latest works made by Designia"></span>
15  </li>
16  <li>
17    <span class="link-bg"></span>
18    <a href="?content=services">Services</a>
19    <span class="linkDescr" title="Awesome Designia services"></span>
20  </li>
21  <li>
22    <span class="link-bg"></span>
23    <a href="?content=contacts">Contacts</a>
24  </li>
25 </ul>
```

To create your own menu link, first of all you have to create list element inside the **ul** list with a “menu” class using the **<li>** tag. Then, add an empty span element with a “link-bg” class. After that put the link title inside using **<a>** *and* **</a>** tags. If you want to use typewriter effect to display link’s description you have to add one more span element with a “linkDescr” class and to specify link’s description inside of “title” attribute of this span tag.

Then you have to create new PHP file inside the **content/** folder with a HTML content of new page. Name it for example – **mypage.php**. So all you have to do is to specify link to this file in the created link with a HREF attribute. But you should use the following rule: **You have to specify dynamic path to this file, not direct.**

So your new link will look like this one:

```
<li>
  <span class="link-bg"></span>
  <a href="?content=mypage">My Page</a>
  <span class="linkDescr" title="My Page description"></span>
</li>
```

Don't need to write it like this: **href="content/mypage.php"** Use it like in example above. JavaScript automatically will include content from the **PHP** file which is located in the **content/** folder with filename specified in **content** variable (in this case is the **mypage**)

We need to use such formatting to make these pages visible for search engines and users with disabled JavaScript.

## Portfolio Section

---

Read more about portfolio section in chapter 7

## Bottom Modules

---

HTML content of Bottom Modules is located in the **includes/bottom-mods.php** file:

Additional "first" class must be added to the first module

Any HTML content here

You can add additional "yellow" class to make module title yellow.

Additional "last" class must be added to the last module

```
<div class="module first">
  <h3 class="module-title">First Module Title</h3>
  --- First Module Content ---
</div>
<div class="module">
  <h3 class="module-title yellow">Second Module Title</h3>
  --- Second Module Content ---
</div>
<div class="module">
  <h3 class="module-title">Third Module Title</h3>
  --- Third Module Content ---
</div>
<div class="module last ">
  <h3 class="module-title yellow">Fourth Module Title</h3>
  --- Fourth Module Content ---
</div>
<div class="clear"></div>
```

## Popup Window

---

All content pages (files) will be loaded into the div block with a "pop-content" class.

## Footer Section

---

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

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

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

## 5. Typography

---

### Headings

---

This is the content heading. You should use the following formatting: `<h2 class="content-heading"> ... Some Heading ... </h2>`

Designia Features

This is the inner heading. You should use the following formatting: `<h3 class="inner-heading"> ... Some Heading ... </h3>`

Animated Page Loading

Every page is loading with a nice smooth fade and slide animation. Try these links for content with a different size: [Home](#), [Services](#), [Contacts](#)

### Button

---

Send Message

This is the button. You should use the following formatting:  
`<a class="button"> ... Button text ... </a>`  
`<input type="submit" class="button" value="Button Text" />`

### Sidebar

---

If you are using page WITHOUT sidebar you have to use following formatting:



```

<!-- Page Title -->
<h2 class="content-heading">Page Title</h2>
<!-- Some Page Content -->

<p align="justify">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer a libero at diam consequat vulputate feugiat vel ipsum...</p>

<!-- End of Page Content -->

<!-- End of Page -->

```

But if you want to use sidebar you have to put page content into the DIV block with a “**main-col**” class, like here:

```

<div class="main-col">
  <h2 class="content-heading">Page Title</h2>
  <p>Some Page Content</p>
</div>
<div class="sidebar">
  <div class="module">
    <h3 class="module-title">First Module Title</h3>
    <p>Some Module Content</p>
  </div>
  -----
  <div class="module">
    <h3 class="module-title">Second Module Title</h3>
    <p>Some Module Content</p>
  </div>
</div>
<div class="clear"></div>

```

This page will look like this:

This is the module title. You should use the following formatting: `<h3 class="module-title"> ... Some Heading ... </h3>`

## Welcome to Designia

**Designia** is a premium Ajax based animated website template, designed by the [iDangero.us](#) team. This template is perfect for small design studios, projects, or it can be used as a fashion site.

The main advantage of this template is that it uses only one page. All content pages are loading "on the fly", so your users do not need to wait the reloading of entire site. Template is fully animated without the use of Flash technology. Smooth and good looking animation and Ajax features of this template are realized with the JavaScript jQuery library.

This template is easy to redesign, it has very simple and clear API. Buying this template you will also get all .PSD sources and detailed documentation.

So, it's time to checkout the main pages of the site:

- Template [Features](#)
- Ajax [Contact Form](#) with validation
- [Portfolio](#) with a popup projects

## Great Offer

Buying this template you will also get **TypeWriter** premium JavaScript library for **FREE**:



[Learn more about TypeWriter](#)

## 6. 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. We should use the following formatting:

```
<a href="index.php?content=about" class="ajax">About</a>
```

Filename of the **php** file from the **content/** folder specified as a value of the **content** variable.

When we'll specify an "**ajax**" class, the script will load file content with Ajax into the content section

**Example 2.** We want to load HTML content from the **content/folio/folio\_item1.php** file directly from the simple link in the text. We should use the following formatting:

```
<a href="index.php?content=folio/folio_item1" class="ajax">First Item</a>
```

Filename of the **php** file from the **content/** folder specified as a value of the **content** variable.

When we'll specify an "**ajax**" class, the script will load file content with Ajax into the content section

**\*Example 3.** We want to load HTML content from the **content/about.php** file directly from the simple link in the text. We also can use the following formatting:

```
<a href="content/about.php" class="ajax direct">About</a>
```

Direct file location

When we'll specify an additional "**direct**" class, the script will load file specified directly in the HREF attribute

But this way (with direct file location) is highly not recommended to use

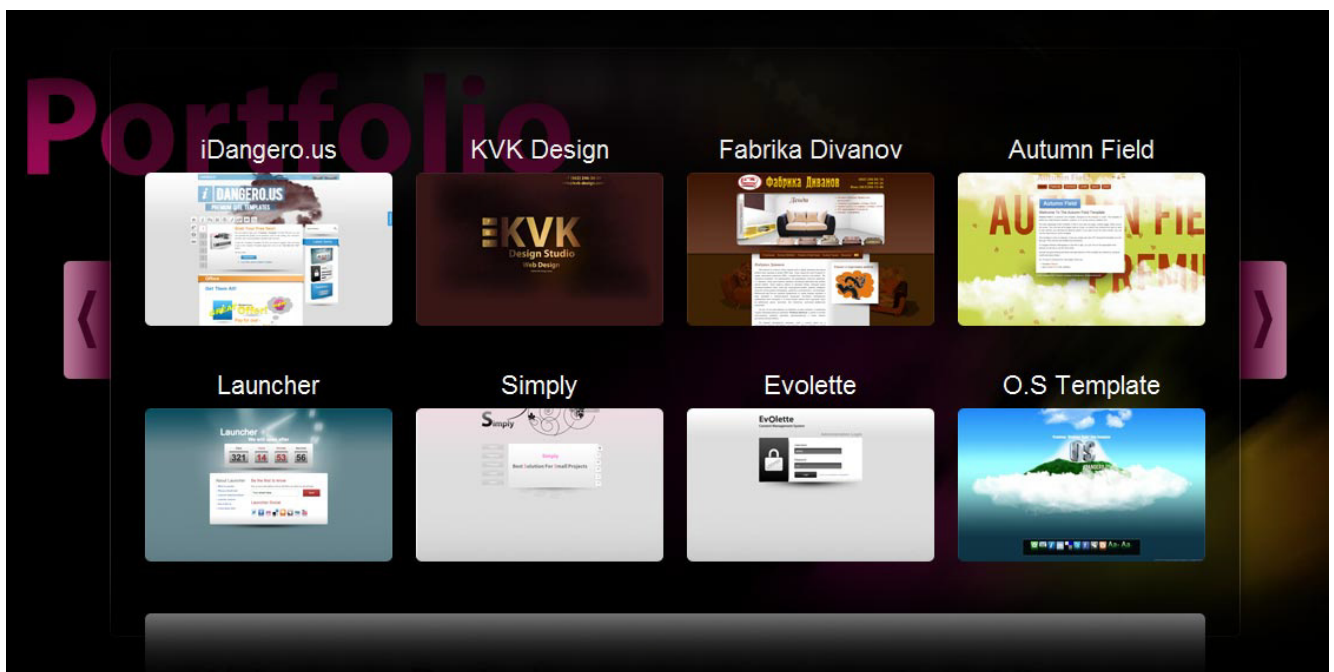
As you can see it is very easy to use Inner Linking. All you need is to specify filename as a value of the content variable using the **href** attribute and to specify the “**ajax**” class to load page with Ajax.

## 7. Portfolio Section

---

### *Portfolio appearance*

---



To open portfolio you can click on the “Portfolio” menu link or use inline link with a special class like this one:

```
<a href="#" class="openFolio">Go to portfolio</a>
```

When we'll specify an “**openFolio**” class, then the portfolio section will open through the animation queue.

To slide portfolio slides you need to click on the pink arrows on both sides of portfolio.

### *Portfolio HTML Content*

---

Now let's look at the portfolio HTML content which is located in the “**includes/portfolio.php**” file:

"folio-slides-wrap" and "folio-slides" containers are used for styling and to make hidden overflow for other portfolio slides.

Each "folio-slide" div container contains 8 folio items

This is a dummy container used to add additional spacing

"folio-item" div contains one folio item. Use additional "first" class to the first folio-item in the line.

Folio item title here

Link to folio-item file with HTML content. Use "popup" class to open it in the popup window, or "ajax" class to place it inside of content section

Use additional "last" class to the last folio-item in the line.

Second "folio-slide" div container with another 8 folio items

```
1 <div class="folio-slides-wrap">
2   <div class="folio-slides">
3     <div class="folio-slide">
4       <div style="height:50px"></div>
5       <div class="folio-item first">
6         <h3 class="folio-title">iDangero.us</h3>
7         <a href="?content=portfolio/id" class="popup"></a> </div>
8       <div class="folio-item">
9         <h3 class="folio-title">KVK Design</h3>
10        <a href="?content=portfolio/kvk" class="popup"></a> </div>
11        <div class="folio-item">
12          <h3 class="folio-title">Fabrika Divanov</h3>
13          <a href="?content=portfolio/fabrika" class="popup"></a> </div>
14        <div class="folio-item last">
15          <h3 class="folio-title">Autumn Field</h3>
16          <a href="?content=portfolio/autumn" class="popup"></a> </div>
17        <div class="folio-item first">
18          <h3 class="folio-title">Launcher</h3>
19          <a href="?content=portfolio/launcher" class="popup"></a> </div>
20        <div class="folio-item">
21          <h3 class="folio-title">Simply</h3>
22          <a href="?content=portfolio/simply" class="popup"></a> </div>
23        <div class="folio-item">
24          <h3 class="folio-title">Evolette</h3>
25          <a href="?content=portfolio/evolette" class="popup"></a> </div>
26        <div class="folio-item last">
27          <h3 class="folio-title">O.S Template</h3>
28          <a href="?content=portfolio/os" class="popup"></a> </div>
29      </div>
30      <div class="folio-slide">
31        <div style="height:50px"></div>
32        <div class="folio-item first">
33          <h3 class="folio-title">iDangero.us</h3>
34          <a href="?content=portfolio/id" class="popup"></a> </div>
35        <div class="folio-item">
36          <h3 class="folio-title">KVK Design</h3>
37          <a href="?content=portfolio/kvk" class="popup"></a> </div>
38        <div class="folio-item">
39          <h3 class="folio-title">Fabrika Divanov</h3>
40          <a href="?content=portfolio/fabrika" class="popup"></a> </div>
41        <div class="folio-item last">
42          <h3 class="folio-title">Autumn Field</h3>
43          <a href="?content=portfolio/autumn" class="popup"></a> </div>
44        <div class="folio-item first">
45          <h3 class="folio-title">Launcher</h3>
46          <a href="?content=portfolio/launcher" class="popup"></a> </div>
```

```

47     <div class="folio-item">
48         <h3 class="folio-title">Simply</h3>
49         <a href="?content=portfolio/simply" class="popup"></a> </div>
50     <div class="folio-item">
51         <h3 class="folio-title">Evolette</h3>
52         <a href="?content=portfolio/evolette" class="popup"></a> </div>
53     <div class="folio-item last">
54         <h3 class="folio-title">O.S Template</h3>
55         <a href="?content=portfolio/os" class="popup"></a> </div>
56     </div>
57 </div>
58 </div>
59 <div class="clear"></div>

```

As you can see it's easy to add (change) your own folio items into the portfolio section.

## 8. CSS Files and Structure

---

There is 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/designia.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.

**designia.css** file contains all general layouts for the template. It is separated into appropriate sections.

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.

## 9. JavaScripts

---

Designia Template uses three 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-1.5.min.js"></script>
<script type="text/javascript" src="scripts/typewriter-1.0.min.js"></script>
<script type="text/javascript" src="scripts/designia.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 (typewriter-1.0.min.js) is the premium JavaScript library. But you got it for free. You can learn more about this library and its usage from the **typewriter/** folder from the downloaded archive.

And the third script file (designia.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.

First of all let's look at the custom script more closely.

The first part of the script is executed when the document is loaded. Here are the effects and rules for handling menu links, its animation, Ajax loader, Content pages initialization, function used for popup window etc.



We have to load this function (see below) to active some features for inline content such as ajax loading for links, popup for links etc.

"Blink" effect for links with a "blink" class, when we'll hover such link it will fade to opacity 0.5 for 300ms

To make the Typewriter work we need to add unique "id" attribute to every "linkDescr" span equal to its index number.

We need to remove "divider" bg image from the last menu link, and to add additional margin and padding for the first link

Hover animation for menu links. When we'll take the mouse cursor on the link, then the "link-bg" span will fade in for 300ms, and if menu link has the children "linkDescr" span with its description the **typeWriter()** function will be executed

And when we take mouse cursor away from link its background (span with a **link-bg** class) will fade out for 150ms. TypeWriter function will be stopped and HTML content of the children "linkDescr" span will be removed.

Handling for menu links. When you'll click on it, first of all, function will check if this link has an "openFolio" class (Portfolio link). If yes then it will check if the Portfolio opened or closed, if opened then it will be closed and otherwise

If it is not a "Portfolio" link then the loadContent() function will be executed to load new page content, and Portfolio will be closed.

Animation for the Portfolio images. When we'll hover such image it will fade to opacity 0.3 for 300ms

Animation used to slide Folio Slides. When we'll click on the "folio-right" div (arrow), then the all divs with a "folio-slide" class will slide left to -940px for 1.5 seconds.

When we'll click on the "folio-left" div, then the all divs with a "folio-slide" class will slide left to +940px for 1.5 seconds.

```
1 // JavaScript Document
2 jQuery(document).ready(function() {
3     pageScript()
4     /*-----Blink Effect-----*/
5     $("a.blink").hover(
6         function() {$(this).fadeTo(300,0.5)},
7         function() {$(this).fadeTo(300,1,function() {$(this).css("filter","")} )}
8     )
9     /*-----Menu-----*/
10    var linkDescr, linkIndex, menuLink;
11    $("#menu li").each(function() {
12        $(this).children(".linkDescr").attr({"id":"linkDescr"+$(this).index()});
13    })
14    $("#menu li:last").css('background','none')
15    $("#menu li:first").css({'margin-left':'10px','padding-left':'10px'})
16    $("#menu li").hover(
17        function() {
18            linkIndex = $(this).index();
19            $(this).children(".link-bg").fadeIn(300);
20            if($(this).children(".linkDescr").length>0) {
21                $(this).children("a").animate({top:"5px"},200);
22                linkDescr = $(this).children(".linkDescr").attr('title');
23                typeWriter({twID:linkIndex,text:linkDescr,id:'linkDescr'+linkIndex,delay:20})
24            }
25        },
26        function() {
27            $(this).children(".link-bg").fadeOut(150);
28            $(this).children("a").animate({top:"30px"},150);
29            typeWriter({twID:linkIndex,"stop"});
30            $(this).children(".linkDescr").html("");
31        }
32    )
33    $("ul.menu > li").click(function() {
34        if($(this).hasClass("openFolio")) {
35            if($("#portfolio").css('display')== 'block') {
36                closePortfolio()
37            }
38            else openPortfolio()
39        }
40        else {
41            if($("#portfolio").css('display')== 'block') closePortfolio()
42            menuLink = $(this).children("a");
43            loadContent("content/"+menuLink.attr("href").split("content=")[1]+".php")
44        }
45    })
46    $("#menu a").click(function(e) {e.preventDefault()})
47    /*-----Portfolio-----*/
48    $("#portfolio img").hover(
49        function() {
50            $(this).fadeTo(300,0.3)
51        },
52        function() {
53            $(this).fadeTo(300,1)
54        }
55    )
56    $(".folio-right").click(function() {
57        if($(".folio-slide").css("left")!="-" + $(".folio-slide").length-1)*940 + "px")
58            $(".folio-slide").animate({left:"-940px"},1500)
59    })
60    $(".folio-left").click(function() {
61        if($(".folio-slide").css("left")!="0px") $(".folio-slide").animate({left:"+940px"},1500)
62    })
63 }
```

This code means that when the Ajax request is starting, then the `openLoader()` function will be executed

When the Ajax request is completed, then the `closeLoader()` function will be executed

When we'll click on the dark layer, the popup window will be closed

Handling for links with a **"popup"** class. Width and height variables will be equal to the REL and REV attributes. If link has the "direct" class then the URL will be equal to the value of HREF attribute, otherwise it will be equal to the value of the **"content"** variable from HREF attribute of link. And the `popUp()` function will be executed.

Handling for links with a **"ajax"** class. . If link has the "direct" class then the URL will be equal to the value of HREF attribute, otherwise it will be equal to the value of the **"content"** variable from HREF attribute of link. And the `loadContent()` function will be executed.

```

63  /*-----Ajax Loader-----*/
64  $(".ajax-loader").ajaxStart(function() {
65      openLoader()
66  })
67  $(".ajax-loader").ajaxComplete(function() {
68      closeLoader();
69  })
70  /*-----Popup Window-----*/
71  $("#p-layer").click(function() {
72      closePopup();
73  });
74  $(".a.popup").click(
75  function(e) {
76      e.preventDefault();
77      var width=$(this).attr("rel");
78      var height=$(this).attr("rev");
79      if($(this).hasClass("direct")) var uri = $(this).attr("href");
80      else {
81          var url = "content/"+$(this).attr("href").split("content=")[1]+".php"
82      }
83      popUp(url,width,height)
84  })
85  /*-----Ajax Content-----*/
86  $(".a.ajax").click(
87  function(e) {
88      e.preventDefault();
89      if($(this).hasClass("direct")) loadContent($(this).attr("href"));
90      else loadContent("content/"+$(this).attr("href").split("content=")[1]+".php");
91  })
92  })

```

The next part of code contains functions which could be called from any place of the document. These functions are used to load files with content, for submitting and validation of the contact form, to call a popup window etc:

These two functions are used to show and to hide Ajax loader.

`loadContent()` function is used to load php files from **content/** folder and put their HTML content to site content section.

Ajax GET request to the specified php file from the **content/** folder

We have to put new content inside of "dummy-content" div to calculate new content page' height

Then the "content-inner" div will be animated to new height for 900ms and fade to opacity 0 for 300ms.

After it is hided then the new content will be placed into the "content-inner" div, and `pageScript()` function will be executed. After that the "content-inner" div will fade in for 300ms and "dummy-content" will be cleared.

```

93  /*-----Ajax Loader-----*/
94  function openLoader() {
95      $(".ajax-loader").css({"position":"absolute","top":$(window).scrollTop()+250}).show()
96  }
97  function closeLoader() {
98      $(".ajax-loader").hide()
99  }
100  /*-----Content Loading Functions-----*/
101  function loadContent(url) {
102      jQuery.get(url,function(newContent) {
103          $(".dummy-content").html(newContent)
104          $(".content-inner").animate({height:$(".dummy-content").height()},{queue:false,duration:
105  900})
106          $(".content-inner").fadeTo(300,0,function() {
107              $(".content-inner").html(newContent);
108              pageScript();
109              $(".content-inner").fadeTo(300,1,function(){$(".content,.content-inner").css("filter",
110  ""))})
111              $(".dummy-content").html("")
112          })
113      })
114  }

```



This function will be executed with every page load.

"Blink" effect for elements with a "blink" class, when we'll hover such link it will fade to opacity 0.5 for 300ms

```
113 function pageScript() {
114     $(".content .blink").hover(
115         function() {$(this).fadeTo(300,0.5)},
116         function() {$(this).fadeTo(300,1,function(){$(this).css("filter","")})})
117     )
118     $(".content a.popup").click(
119         function(e) {
120             e.preventDefault();
121             var width=$(this).attr("rel");
122             var height=$(this).attr("rev");
123             if($(this).hasClass("direct")) var uri = $(this).attr("href");
124             else {
125                 var url = "content/"+$(this).attr("href").split("content=")[1]+".php"
126             }
127             popUp(url,width,height)
128         })
129     $(".content a.ajax").click(
130         function(e) {
131             e.preventDefault();
132             if($(this).hasClass("direct")) loadContent($(this).attr("href"));
133             else loadContent("content/"+$(this).attr("href").split("content=")[1]+".php");
134         })
135     $(".content a.openFolio").click(
136         function(e) {
137             e.preventDefault();
138             openPortfolio()
139         })
140     $(".content a.pop-image").click(function(e) {
141         e.preventDefault();
142         var bigImgUrl = $(this).attr("href");
143         var bigImgWidth = $(this).attr("rel");
144         popUpImage({url:bigImgUrl,width:bigImgWidth ,smallImg : $(this).children("img") });
145     })
146 }
147 /*-----Functions to open and to close Portfolio section-----*/
148 function openPortfolio() {
149     $(".portfolio-bg").delay(1800).fadeIn(600)
150     $("#portfolio").slideDown(1000).animate({ "margin-top": "-19px" },600)
151     $(".folio-left,.folio-right").delay(1800).fadeIn(300)
152 }
153 function closePortfolio() {
154     $(".portfolio-bg").fadeOut(300)
155     $("#portfolio").animate({ "margin-top": "0px" },600).slideUp(1000)
156     $(".folio-left,.folio-right").fadeOut(300)
157 }
158 /*-----Popup Windows-----*/
159 function popUp(url,width,height) {
160     var width=width||600;
161     var height=height||"auto";
162     var windowTop = $(window).scrollTop() + 100;
163     var windowPadding = $(".window").css("padding").split("px")[0]
164     if(!windowPadding) windowPadding = $(".window").css("padding-left").split("px")[0]
165     jQuery.get(url,function(data) {
166         $(".pop-content").html(data);
167         $(".window").css({"width":width,"height":height});
168         $(".window").css({"left": ($(window).width()-$(".window").width())/2-windowPadding,
169 "top":windowTop}).delay(300).fadeIn(300);
170         $(".window .close").delay(600).fadeIn(200);
171         openLayer();
172     })
173 }
```

Handling for links with a "popup" class. The same principle like above.

Handling for links with an "ajax" class. The same principle like above.

Handling for links with an "openFolio" class. If we'll click on the same link then the Portfolio section will be opened (openPortfolio() function)

Handling for links with a "pop-image" class. It is used for special animated popup windows with an image inside. After such link is clicked then the popUpImage() function will be executed.

These two functions are used to open and to close portfolio section.

When the openPortfolio() function is executed then the "portfolio-bg" div will fade in for 600ms after the 1800ms delay. Div with a "portfolio" id will slide down for 1000ms. Arrows will fade in for 300ms after the 1800ms delay

Default width value (600px)

Default height value ('auto')

Top position for popup window

We need to calculate window padding to set the window correctly

Ajax GET request to the specified file. After that new content (from file) will be inserted into the div with a "pop-content" class

Then we need to set window's size and position. After that window will fade in for 300ms after the 300ms delay.

"Close" image will fade in for 200ms after the 600ms delay

And openLayer() function will be executed

This function is used to show "dark-layer". It will fade to opacity 0.6 for 300ms

```
173 function openLayer() {  
174     $('#p-layer').fadeTo(300,0.6)  
175 }
```

**closePopup()** function provides an opportunity to close a Popup Window. Window will be closed using fade animation for 0.3 seconds

```
176 function closePopup() {  
177     $('.window .close,#p-layer').fadeOut(300)  
178     $(".window").fadeOut(300,function(){  
179         $(".window").css({width:"auto",height:"auto"})  
180     });  
181 }
```

This function is used to load animated popup window with an image inside. First of all it calculates position of small image and then it place new resized big image to the pop-content container with an onload event. When the image will be loaded then the popUpImage2() function will be executed

```
182 /*-----Popup Image Functions-----*/
```

```
183 function popUpImage(params) {  
184     openLoader()  
185     var smallLeft = params.smallImg.offset().left  
186     var smallTop = params.smallImg.offset().top  
187     $(".pop-content").html('')  
188 }
```

This function will animate popup window with image inside to center of the window

```
189 function popUpImage2(smallLeft,smallTop,newWidth) {  
190     closeLoader();  
191     var windowPadding = $(".window").css("padding").split("px")[0]  
192     if(!windowPadding) windowPadding = $(".window").css("padding-left").split("px")[0]  
193     $(".window").css({left:smallLeft,top:smallTop,padding:"0px"}).show()  
194     var newLeft = ($(window).width() - newWidth)/2  
195     $(".window").animate({top:$(window).scrollTop()+100, left:newLeft-windowPadding, padding:  
windowPadding},300)  
196     $(".pop-content img").animate({width:newWidth},300,function(){openLayer()})  
197     $(".window .close").delay(500).fadeIn(300)  
198 }
```

```
199 /*-----Validation And Send Message Functions-----*/
```

```
200 function validateEmail() {  
201     var email = $("input#email").attr("value");  
202     var re = /^\\w+([\\.-]?\\w+)*@(((([a-z0-9]{2,})|([a-z0-9]([a-z0-9]+)))[\\.]([a-z0-9])|([a-z0-9]+[-  
203     if(re.test(email)) $("input#email").removeClass("invalid");  
204     else $("input#email").addClass("invalid");  
205 }
```

**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 "invalid" class will be added to the email field (input field with an "email" id attribute). Otherwise it will be removed.

```
206 function validateMessage() {  
207     message = $("textarea#message").val();  
208     m_length = ($.trim(message)).length;  
209     if (m_length < 10) $("textarea#message").addClass("invalid");  
210     else $("textarea#message").removeClass("invalid");  
211 }
```

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

```
212 }  
213 function validateText(fieldId,length) {  
214     field = $("input#" + fieldId);  
215     text = field.val();  
216     m_length = ($.trim(text)).length;  
217     if (m_length < length) field.addClass("invalid");  
218     else field.removeClass("invalid");  
219 }
```

**validateText()** function is used to validating text fields in the contact form. It is used here to validate **Subject** and **Name**. If the field contains less than **length** characters, then the "invalid" class will be added to this field. Otherwise the "invalid" class will be removed.

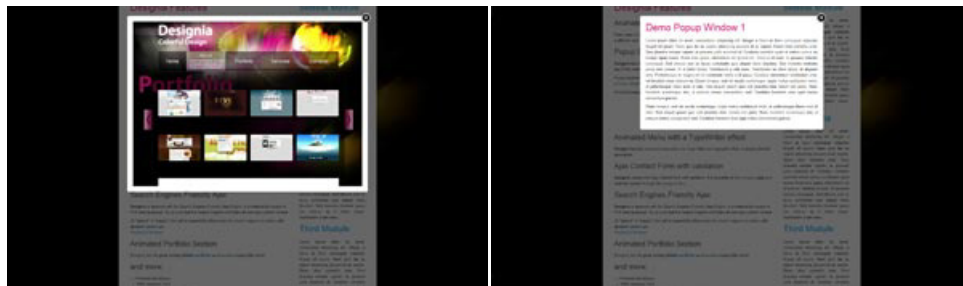
**sendMessage()** function provides an opportunity to send e-mail message by the Ajax request (without reloading of the site). First of all validate functions will be executed (to validate **Email**, **Name**, **Subject** and **Message**). And then if some of the field has the "invalid" class, message will not be sent.

```
220 function sendMessage() {
221     validateEmail();
222     validateMessage();
223     validateText('name',6);
224     validateText('subject',6);
225     if (!$("textarea#message,input#email,input#subject,input#name").hasClass("invalid")) {
226         name = $("input#name").attr("value");
227         email = $("input#email").attr("value");
228         subject = $("input#subject").attr("value");
229         message= $("textarea#message").val().replace(/<\/?[>]+>/gi, '');
230         jQuery.post("scripts/send_message.php",{name : name, email : email, subject : subject,
message : message},
231             function (status) {
232                 $("#contact-form").fadeOut(300,0,function() {
233                     $(this).css({"visibility":"hidden"})
234                     $(".message_status").html(status).fadeIn(300)
235                 })
236             }
237         );
238     }
239 }
```

If all fields are correct, then the function will send an 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 message with the mailing status, contact form will fade out and mailing status will fade in instead of form.

## 10. Popup Window

**Designia** template has its own engine to call popup windows. Here is screenshot:



Popup engine allows you to popup HTML content from the file.

**Example 1.** We need to call a popup window with HTML content from the *content/about.php* file. We can use the following formatting for the links:

Use "**popup**" class to call a popup window with HTML content from file

Specify **filename** as a value of the **content** variable in the **href** attribute

Use **rel** attribute to specify window width in px (not required – 600px is default)

And **rev** attribute for window height in px (not required – 'auto' is default)

```
<a class="popup" href="index.php?content=about" rel="800" rev="200">About</a>
```

JavaScript automatically will include content from the **PHP** file which is located in the **content/** folder with filename specified in **content** variable (in this case is the **about**)

**Example 2.** We need to call a popup window with HTML content **NOT** from the **content/** folder file. In these cases additional class "direct" is required! For example we need content from the **myfolder/about.php** file. We should use the following formatting for the links:

Additional class - "**direct**"

Specify **direct** file location in the **href** attribute

Use **rel** attribute to specify window width in px (not required – 600px is default)

And **rev** attribute for window height in px (not required – 'auto' is default)

```
<a class="popup direct" href="myfolder/about.php" rel="800" rev="200">About</a>
```

**Example 3.** We need to call a popup window with an image inside. We need small image (image\_s.jpg) and big image (image\_b.jpg). We should use the following formatting for the links:

Use "**pop-image**" class to call animated popup window with an image inside

Location of the big image

Big image width. **Required**

```
<a class="pop-image" href="images/image_b.jpg" rel="800">
  
</a>
```

Location of the small image

Width and height of the small image. **Required**

# 11. 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 class="message_status"> </div>
3 <form id="contact-form" action="javascript:sendMessage()" method="post">
4   <p><strong>You can contact us using the following contact form:</strong></p>
5   <p>
6     <label>Your Name: *<br />
7       <input type="text" size="50" name="name" id="name" onkeyup="validateText('name',6)" />
8     </label>
9   </p>
10  <p>
11    <label>Subject: *<br />
12      <input type="text" size="50" name="subject" onkeyup="validateText('subject',6)" id="subject" />
13    </label>
14  </p>
15  <p>
16    <label>E-mail: *<br />
17      <input type="text" size="50" name="email" onkeyup="validateEmail()" id="email" />
18    </label>
19  </p>
20  <p>
21    <label>Message: *<br />
22      <textarea id="message" onkeyup="validateMessage()" cols="30" rows="10"></textarea>
23    </label>
24  </p>
25  <p><em>Fields marked with an asterisk(*) are required</em></p>
26  <p>
27    <input type="submit" value="Send Message" />
28  </p>
29 </form>
```

Message status will be posted 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:



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

## 12. PSD Files & Structure

---

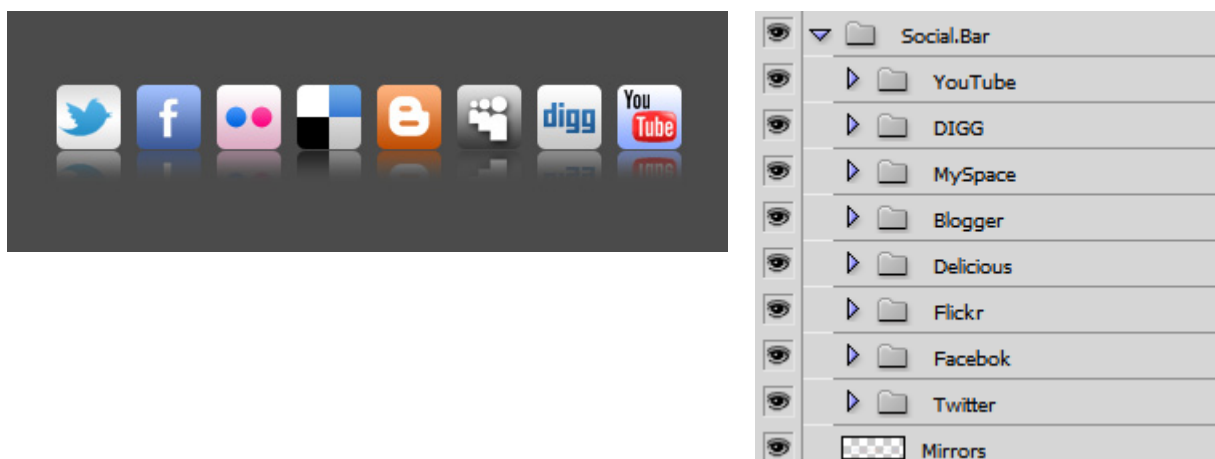
There are the **three** layered **.psd** files comes with the **Designia** template. You can locate them into the **psd/** folder form the downloaded archive. Using these files you can easily redesign template. To edit these files you'll need an Adobe™ Photoshop™ (Recommended is CS4 or CS5 version)



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



Second 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.

# 13. 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 **Designia** (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**.