

Wendy Wood Photography

Website Redesign

by
Charlie Frost
Andy Husak
Aaron Verstraete

Executive Summary

The goal of this project was to redesign the Wendy Wood Photography website. The old site was several years old and the client wished to have it updated to Flash. By using Flash the group was able to give the site visual appeal. Along with making the site more interactive the group was tasked with making the site easy to update from outside Flash. The group has achieved both goal of the project The new site has the visual appeal of Flash. The content for the site is load dynamically into the Flash file so that the content can be easily changed.

Table of Contents

1. Abstract	1
2. Project introduction	1
3. Requirements and Specifications	2
3.1 Functional requirements	2
3.2 Non-functional requirements	3
4. Design/solution approach	4
4.1 Overview of web pages	4
a. Splash page	4
b. Home	4
c. Testimonials, Artist, Biography, Session Information, and Clothing Tips	5
d. Gallery	6
e. Boutique	7
f. Investment	8
g. Contact	8
4.2 Flash design	10
a. Layers	10
b. ActionScript	10
c. Flash transitions	11
4.3. Functional modules	11
a. Slideshow	11
b. Gallery	12
c. Dynamic text	13
d. Submission form	14
e. Music player	14
5. Implementation details and results	16
5.1 Programming decisions	16
a. Flash vs. HTML	16
b. XML documents	16
c. Site visitor statistics	17
5.2 Obstacles	17
a. Flash learning curve	17
b. Looping of background music	18
c. Image resolutions	18
6. Future direction	19
7. Conclusion	19
8. Glossary	20

1. Abstract

Wendy Wood is a solo photographer specializing in various events such as weddings, family portraits, and graduations. Her current website is several years old and in need of a redesign and expansion of features. The client desires a remodeled website to accommodate a lighter color scheme, newly designed logos, additional and expanded pages, enhanced user interactivity, and visual flair. Our team designed this new website in Adobe Flash, creating architectural transitions, dynamically loaded text, and the display of images using various techniques. The new website incorporates all of the enhanced features our client has requested while still allowing for easily implemented future changes.

2. Project introduction

Photographer Wendy Wood currently owns a website with a layout that is several years old. The site is an HTML-based creation with navigation that relies upon five image-based buttons located along the top of the page which change accordingly to the visitor's current location (e.g. the buttons will transition to several gallery categories during the viewing of gallery slide shows). The homepage design involves a simple, stationary slide show of multiple images that contain minimal visual transitions. The color scheme contains shades of black filling the background of the site with light gray tones for the navigation buttons and logos and white text to fill the necessary pages. Users are able to access proofs through a password-enabled external site.

The client is requesting a remodeled and updated website. The newly formed site will contain nine accessible sections (home, galleries, testimonials, session information, artist, biography, investment, boutique, clothing tips, and a contact form). The updated design will feature a lighter, lavender color scheme. Images and various pages within the site will have an increased amount of visual transitions.

3. Requirements and Specifications

The client's website needs to be remodeled and updated from the current, several year old format. Our group has been given a list of requirements and preferences that the client expects to see applied to the new page. The client has also provided us with the URL to another photographer's website to serve as a rough model of what she expects from her newly remodeled site.

3.1 Functional requirements

- The homepage must have a splash screen which displays our client's business card, and includes a link to the main site.
- The main site must display an expanded number of sections to the user (home, galleries, testimonials, session information, artist, biography, investment, boutique, clothing tips, and a contact form). Each section will require various components, such as a text area, a slideshow of images, a gallery, or a submission form.
- The main site must play music files. These songs must loop, and allow the user to pause/play, and advance to the next song.
- The main page on the main site must display a series of photos which rotate in a loop. The border on each image must visually expand or contract to fit each image.
- A gallery page must display various sub galleries, a main large image, and thumbnails of all available photos. The page must rotate automatically through the images, but allow the user to pick a specific one, or to pause the slide show.
- A boutique page must be created, displaying a number of products available for purchase. Each product should display an image, and descriptive text such as name and pricing information.
- A contact page must be created which allows the user to submit a form with information to our client. This information must be sent to our client as an email.
- User statistics such as number of page views should be kept track of through a separate process.

3.2 Non-functional requirements

- The design must be visually appealing and much more interactive. This in turn requires the site be created in Adobe Flash.
- The website must be low cost, which constrained us to only use a trial version of Flash. Our client will then need to find a licensed copy from which to export the final product.
- The website must have the ability to update images and text as necessary, requiring we load files externally into the Flash file.
- The website must be easy to update.

4. Design/solution approach

4.1 Overview of web pages

The website consists of an introductory splash image which then links to the Flash content. All content of the website is contained on one HTML page, embedded within a single Flash object, allowing increased interactivity and graphic effects. The general site layout is displayed in Figure 1.

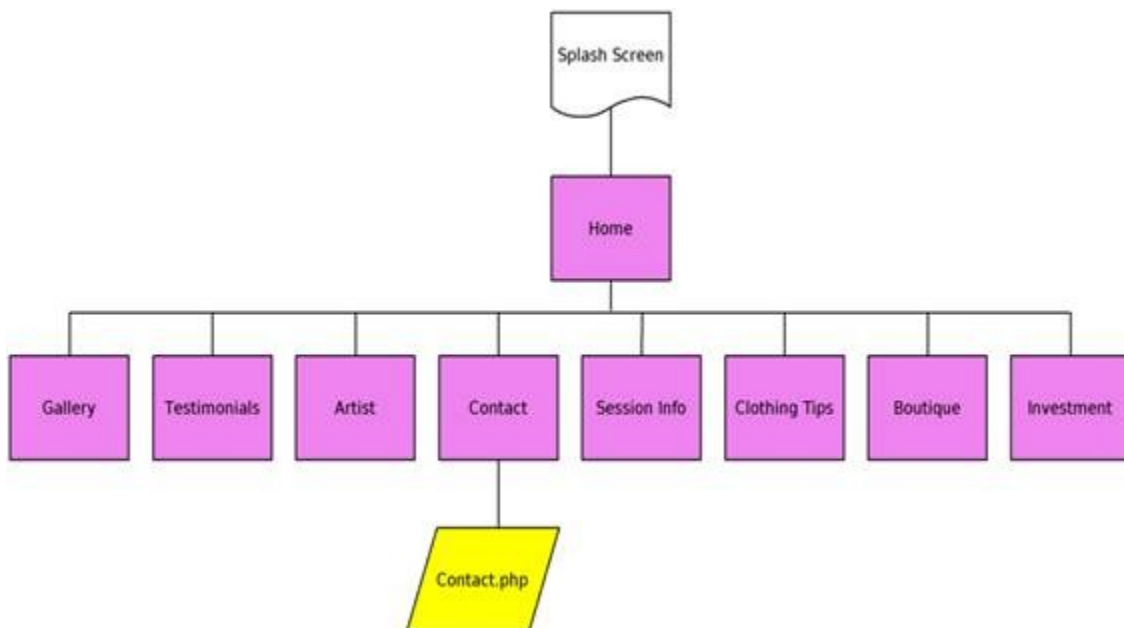


Figure 1: Site Map

a. Splash page

The splash page is the web page initially greeting the customer. It contains an image of Wendy Wood's business card. This page links directly to the main Flash page, which consists of all the other pages which will follow. This page is coded in HTML.

b. Home

The first page displayed once a user is linked from the splash page is Home. Home's layout is a three-column design consisting of pattern-slide show-pattern. The

slide show will display six images of our client's choice, which will automatically be rotated through, using a fade effect. In addition, users will immediately have the ability to navigate across the website with buttons located on the bottom of the page. A screenshot of this layout is shown in Figure 2.



Figure 2: Home Page

c. Testimonials, Artist, Biography, Session Information, and Clothing Tips

These pages are text related and will have two content boxes. The first of these boxes will contain a slide show that will cycle through pictures and will be located on the left. The second content box located on the right will take a text file form outside the Flash file and load it into a text field. This text field will cover most of the box leaving room for the scroll buttons that will be located at the bottom of the page to allow users to read text that does not fit in the space provided. The file is located outside the Flash file so that the client may edit the text with ease and so that the Flash file itself need not be accessed. An example screenshot of the testimonials page is provided in Figure 3.



Figure 3: Testimonials Page

d. Gallery

The gallery page will display various galleries of selected photos to showcase Wendy's photographic talent. The page divides the content area into two main areas. The left area is largest and will be used for displaying the current large image. The right smaller area will contain a group of thumbnails of all images available to be viewed. Links will allow navigation between various sub-galleries such as newborns or weddings. A screenshot of an example gallery page can be seen in Figure 4.

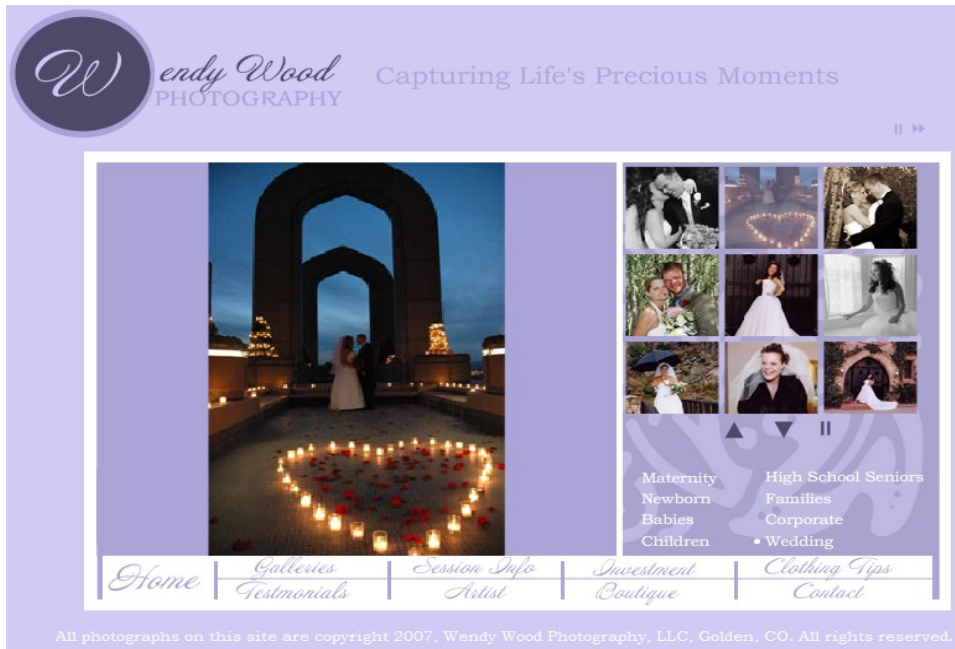


Figure 4: Gallery

e. Boutique

The boutique page showcases various products a customer can purchase other than photos, such as handbags and bracelets. The boutique has two areas, one for thumbnails of the various products and their descriptions, and another for a larger image of the currently selected product. As a product is selected from the thumbnails, that specific product's description is displayed in an area below all of the thumbnails, and its large image is displayed as well. The Boutique is shown in Figure 5.



Figure 5: Boutique

f. Investment

The investment page gives a visual presentation of the various photo packages and their pricing. Due to the detail of the content, the layout consists of only a small pattern on the side, with the majority of the area dedicated to the content.

g. Contact

The contact page will have two content boxes. The first content box which contains the input fields, will take the information that the user provides and upon clicking the submit button, will send that information to the contact.php form on the server. The contact.php page, invisible to the user, sends the information in an e-mail to the client. The second box on this page will have the slide show that will display several pictures that the client will specify. These pictures will dynamically load from an external file on the server and can be changed later. The layout of the page is shown in Figure 6.



Figure 6: Contact Page

4.2 Flash design

a. Layers

Since the background for most of the web pages will remain the same from page to page the layers for the background will be reused for each page. This will ensure that the static elements of the site will not move during page transitions. These layers will include the music player, a site feature that the client wants displayed throughout the entire site. In the diagram of Figure 7, the layers can be seen in their proper stacking order from top to bottom.

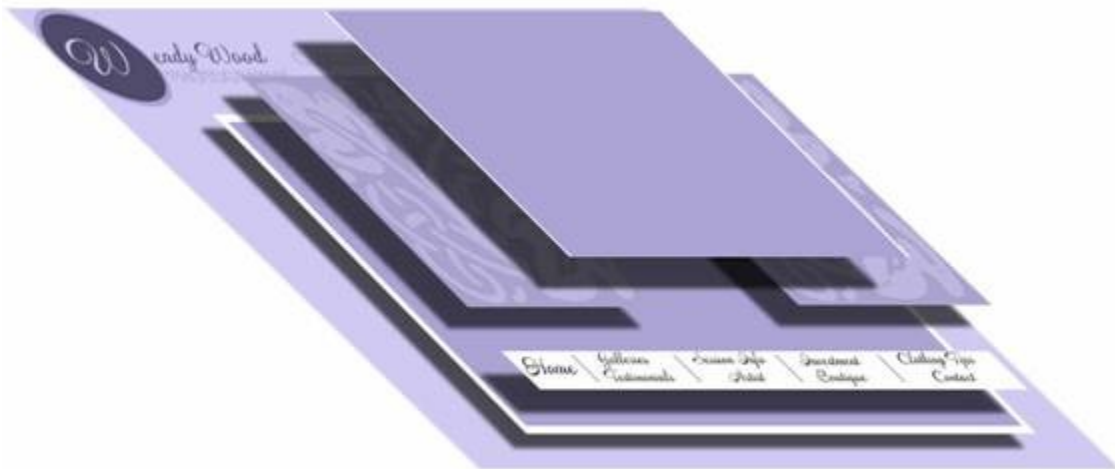


Figure 7: Separated Page Layers

b. ActionScript

ActionScript is the programming language of Flash and will handle the gallery, dynamic text, and the functionality of the music player. ActionScript can be applied in any context within a Flash movie: on its own layer where it's applied in a scene-wide fashion, or on an object or movie clip, where its context is specific to that item. ActionScript can act independently from the timeline progression of a Flash movie. Most of the ActionScript will go on its own layer to help keep things organized. The frames of the dynamic text module have the ActionScript instead of the ActionScript layer.

c. Flash transitions

Having the entire site being contained on a single page requires the implementation of transitions from page to page. Each of the nine pages is architecturally formatted in one of four ways: a pattern-slideshow-pattern scheme, a purple background split into two halves, a gallery layout with a large purple background for a large image and a smaller pattern window for thumbnails, and a pattern-purple box scheme. In order to display these four layouts, the Flash object will need to resize and reshape according to the user's navigation across various sections. To resolve this issue, many 20-frame transitions were created, traveling to and from each of the four possible page formats. Next, commands were assigned to each button with ActionScript, allowing the page to transition properly upon clicking each button at any time. For each of the nine pages, static frames were created as destinations for the ActionScript's navigation upon transitioning. Creating the static pages separately also served as an effective method for the overall organization of the Flash project.

4.3. Functional modules

In the course of designing the various Flash pages, portions of code or content were easily reusable across multiple pages. The design of each of these modules follows, giving a more in depth technical description of each.

a. Slideshow

The slideshow module is used to display various images in rotation. These images are stored externally on the server in an images folder. An XML file is used to keep track of which images are to be used on a per-page basis. Flash files can be considered "sandboxed" content: Flash is designed such that it has very limited and controlled access outside of itself. The reason for using an XML file is that due to this sandboxing, Flash cannot simply iterate through a directory. The XML file contains a list of image locations instead.

Once the XML file is loaded, those entries are added into an array. This array is simply looped through. Using ActionScript, each image is loaded into the Flash file, and displayed in a designated area. These images are put into a movie clip symbol. This

symbol in Flash allows images to be loaded into it, and allows modification to the symbol size, shape, and transparency without affecting the image itself. Care was taken to scale the image dynamically, so that an unknown image size will fit properly within an area which may change between pages. Timer functionality within ActionScript is used such that after a certain time has passed, a new image is loaded.

The slideshow method mentioned above is not used on the homepage due to the need to change the page's overall format upon the transition of each of the six images. Instead, this portion of the site was manually animated and then looped accordingly until the user makes the decision of navigating elsewhere on the site. The slideshow effect is implemented on every other page with the exception of the gallery, boutique, and investment pages. The client is a photographer and therefore, prefers that her work is displayed on the majority of her website in order to familiarize her potential customers with the style of art she provides.

b. Gallery

The gallery module displays a series of images like a slideshow, but also includes thumbnail references of all images available. Images are located and loaded in the same fashion as the slideshow, using XML. An area is dedicated to where the thumbnails will go, and each image is diminished to a small size and added into a multi-column group of thumbnails all contained within another movie clip symbol. Once a thumbnail is clicked, the main picture is updated to display a full sized version of that same image. Buttons are added to scroll through the thumbnails. The thumbnail movie clip is kept behind a mask. A mask is a layer where you can specify what content can and cannot be seen, rather than the content itself. Once something moves out of the visible area, it turns invisible. When the thumbnails move up and down when scrolling the image changes its y-coordinate. The mask only displays a few of the thumbnails while the rest is kept out of view. The end result is what appears to be a scrollable area. An example of a layer mask is shown in Figure 8.

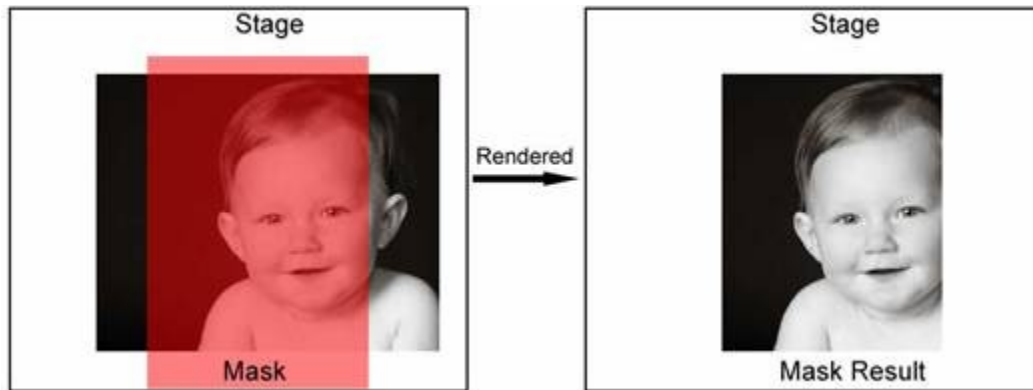


Figure 8: Mask Rendering

On the gallery page, where there are subcategories, each subcategory is displayed by going to a new page and loading a new XML file (as opposed to adding the subcategory functionality into the gallery code itself).

The boutique uses a modified version of the gallery. Within the XML file, each item has an associated image URL as well as a new description tag. This new description tag is loaded into an array, and when a thumbnail is clicked, the main image is loaded and this associated text is displayed in another area. The array index of both image and description is kept consistent in order to associate both.

c. Dynamic text

The text pages of the site are composed of four frames that handle the loading of text to the text box on the Flash page. In the first frame the ActionScript call the loadNumVars function to find the text that will display in the text box. The second frame checks to see if there is or isn't text to left to load. If there is no text to load, the scene jumps to frame four (named "end") where there is a stop() call so that the Flash movie will stop playing. If there is still text to load when the movie reaches frame three, the movie will jump back to frame one (named "load") with a call to the gotoAndPlay() function, which will display the text, however another layer must be added to allow the user to scroll. The additional layer contains the two buttons that will enable scrolling through the text and a movie clip called "controller". The two buttons that are on the page have ActionScript applied to them that tell the controller what button has been pressed. The code for each button is very similar with the exception of the buttons having different

names. Consisting of five frames, the controller is simple as well. Four of these frames are empty and the fifth contains ActionScript that activates appropriate "if" statements to allow the text to scroll in an appropriate direction. The controller finishes by returning to the first frame.

d. Submission form

The form on the contact page is fairly straightforward. Flash components such as text fields and buttons were used to create the form itself. When the submit button is pressed, it sends the content of each of these fields via POST to a PHP file. The flow of data is illustrated in Figure 9.

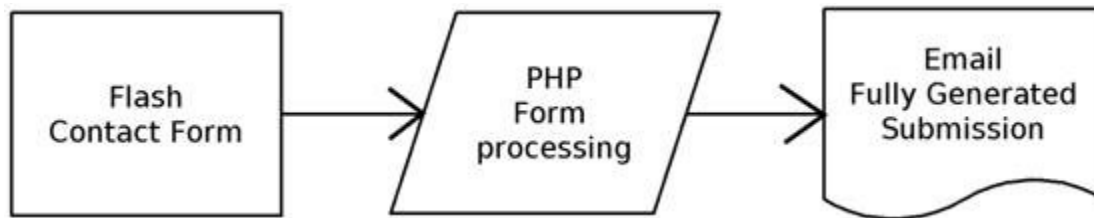


Figure 9: Flash Form Processing

The PHP file receives all the fields from the contact form in the form of POST variables. Each string comes with HTML tags embedded within it (due to Flash), and so these are stripped out. Each string is then used to create an email sent via the PHP file to our client. The email arrives in our clients inbox with a specific subject line and the "From" field set as the email address entered on the form itself.

e. Music player

The music player is present across all pages of the website. The music player has two pieces that give it functionality: The ActionScript, and the buttons that appear on the screen. The ActionScript does several things. It creates the sound object that will be playing and loads the MP3 names from the XML file on the server. The MP3 files are then loaded into an array as sound objects. The music player by default plays unless told to stop. Since Flash is unable to discontinue the music upon sound objects, the addition of a new function is required to pause and play the music. The first function, called `pauseIt()`, stops the music and stores the current position of the selected song using the

position variable in the sound object. When the music is restarted with the `unpauseIt()` function, the position is divided by 1000, due to the elapsed time being stored in milliseconds. The function call is then `.start(position/1000)`. To advance the player to the next track, the "play" function is called again, advancing the index in the array to the next value. The script for the button is also in the same area and defines how the button will look and which functions will be called upon being clicked. The play/pause toggle button and the next button have no ActionScript on them with the exception of the instance names that each contain. These are called within the ActionScript layer.

5. Implementation details and results

5.1 Programming decisions

a. Flash vs. HTML

Our group decided to code the client's website using Flash for several reasons. Flash files were capable of being created using a 30-day free trial of the program, a window of time suitable for the two to three week scope of the project's implementation process. Flash can handle playing music without the use of an outside player while HTML would have required the separate considerations of multiple music players such as Windows Media Player, QuickTime, etc. Using a single Flash object for the entire website would allow the site to reside on a single HTML file, making the creation process more organized with the handling of fewer pages. Finally, Flash enabled the group to easily create visual transitions between various portions of the site.

b. XML documents

Our group determined that the combination of external files with XML documents would serve as the most effective scheme for dynamically loading content into the Flash object. The first idea was to have Flash iterate through a directory, pulling out all necessary files. Flash is sandboxed, as such, directory iteration would not work. Assuming the consistent naming and proper organization of the text and image files, an XML document can easily load these separate pieces of content onto the client's website. In addition, the limited programming knowledge of the client was taken into consideration. With the external files combined with their corresponding XML documents, the client will be capable of editing the content of her site by modifying the external files themselves rather than having to trace through cumbersome Flash frames and ActionScript.

c. Site visitor statistics

Our client wanted access to some sort of private statistics of the number of visitors that were accessing her website. At first, numerous PHP and HTML methods were considered to achieve this. Upon logging into her web host, built-in modules were available for activation capable of tracking the number of site visitors as well as documenting various site statistics that were beyond what was requested.

5.2 Obstacles

a. Flash learning curve

Our group did not begin coding in Flash until the end of week three of the six-week field session. There were a few reasons for this. Licensing issues of the Flash editing software raised many concerns and made both the group and client skeptical about using it without running into potential legal troubles or spending a large amount of money. The client and group were unsure of certain elements of the site that she desired, creating a hesitation with using Flash. Furthermore, all group members were more familiar with the HTML language than they were with the Flash ActionScript language going into the six-week field session duration. However, after considering many tasks that would be easier to accomplish with Flash, the group finally decided to use Flash.

Flash itself is a unique design environment. It is meant both for interactively programmed or scripted uses and for graphical movies. It incorporates the film and animation industry concept of a timeline, stage, and frames. It can also allow scripting or full object oriented programming anywhere within the timeline, on objects, or across the entire movie. This blending of computer science and film made the program somewhat non-intuitive at first. With the shortened time frame, we did not have the luxury of sitting back and learning the fundamentals of flash before diving into coding and designing.

The concern dealt with learning and familiarizing ourselves with Flash and its ActionScripting process. Fortunately, the group was able to utilize many web-based tutorials in order to quickly learn about Flash and the newly encountered approach to the project.

b. Looping of background music

During week five, the client demoed a preliminary version of the website's final design and pointed out that the music was not playing properly. After inspecting the corresponding frames and ActionScript code, the Flash object was failing to stay within the designated frames for the homepage. As a result, the object defaulted by reaching the final frame and looping all the way back to the very first frame of the project, causing the music to restart shortly after loading the homepage. The music would also not itself loop through the provided playlist.

This issue was easily resolved with minimal ActionScript. A command was needed to force the Flash object to remain within the proper frames of the homepage rather than traveling to the first frame and forcing the restarting of the background music. As an extra measure, all code was reviewed to ensure that the Flash object would not return to the first frame upon being initially loaded.

c. Image resolutions

During the week five demo of the website, the client noticed a lack of quality in some of the images. These photos were being displayed in a low resolution and appeared to be muddy and pixilated. One major cause of this problem dealt with the differences in resolutions between the Flash editing stage and the user's browser. During the editing process, the images appeared to be of acceptable quality, however, once the Flash object was put into the setting of one's web browser, the size of the entire object was increased which resulted in the decline in quality of the images.

To fix this issue, the group requested higher resolution versions of the same images. The HTML which contained the embedded flash file was designed such that the flash file filled 100% of the width and height of the screen. This was adjusted to keep the file constrained to its original dimensions. This has resolved any image scaling issues.

6. Future direction

As for the future of our client's website, there are plenty of opportunities to expand upon what our group has created. As mentioned earlier, the client will be able to easily modify the text and images currently contained within her new website if necessary through the use of external files. The client had originally envisioned her new site having a personalized proof-viewing module along with the added ability of customers to purchase copies of their photographed images. Due to time constraints and fluctuating desires from the client, our group was unable to achieve this.

We researched some open source shopping cart solutions, and found a few that could assist the client. These included osCommerce, and Zen Cart. We also found a picture gallery system, aptly named Gallery. Gallery allows per-user albums, private viewing of each album, and watermarks on pictures. Gallery could easily be incorporated into the site to allow viewing of proofs. Integrating Gallery with the shopping cart systems became a problem. Those systems are designed for product-based stores like Amazon, Target, etc. where each product is available to everyone. Zen Cart in specific did not allow per-user private products. We did not have time to investigate if osCommerce would allow it.

7. Conclusion

After considering the finalized expectations from the client, our group believes that the completed work accomplished during the past six weeks has sufficiently fulfilled the client's needs. We have implemented a flash based website with dynamic content, with expanded sections. The site is more interactive, and incorporates music and enhanced images. Many changes were made to the initial memo received during the first week and the group worked efficiently to overcome these modifications. The overall duration of the six weeks has been a period involving learning curves, usual success, and a new standard being set for group based project development.

8. Glossary

Adobe Flash:	A multimedia authoring program capable of creating content for things such as web applications, games, and movies.
HTML:	(Hypertext Markup Language). The predominant markup language for the creation of web pages.
URL:	(Uniform Resource Locator). The address of a resource on the Internet.
PHP:	(PHP: Hypertext Preprocessor). A programming language designed for producing dynamic web pages.
Splash page:	A web page that welcomes the site visitor before directing them to the homepage.
ActionScript:	The programming language used by Adobe Flash.
XML:	A general purpose markup language.
Thumbnail:	A reduced-size version of an image, usually the size of a thumb's nail.