

COMPOSITE 2 ASSIGNMENT: DYNAMIC WEBSITE

Assignment Description

The composite 2 assignment is an involved group effort that results in a finished dynamic web application. The goal is to use all of the programming and design knowledge in combination, to produce a content-managed website that is both easy to use, and visually appealing.

The key to a content-managed site is the ability to log in and change the content without having to use any kind of editing software. This will be made possible with the use of a database to store content and a user-friendly design for the administrative section of the site.

This site will require a lot of planning and organization as well as coordination between group members. As an element of advice, try to keep the scope of the project as small as possible, and polish it to near-perfection in terms of design and programming. A small well-polished piece will look great to a client or potential employer, however a large, rough and unfinished one does not reflect well on the team, regardless of how ambitious it was.

Remember, your website will be built using CMS from class, unless you choose to build the system from scratch and meet all the requirements below. This is not negotiable.

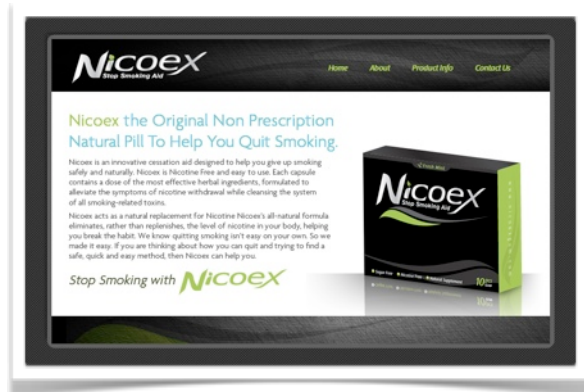
Building your own system from scratch is very ambitious and may take far longer than the time allotted for the project.



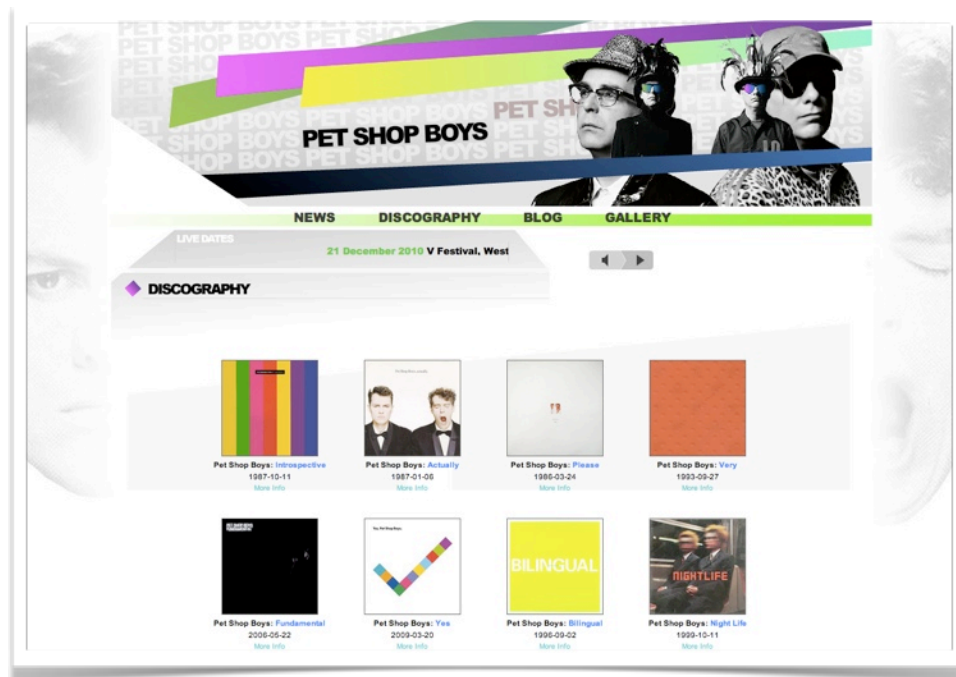
Assignment Options

You can choose any of the options below for your assignment. Keep in mind that despite the projects being vastly different in appearance, the same principles have to be applied for all of them to get them working.

- 1. Dynamic Website.** This is the most general option. You can build an entirely dynamic website which will allow your users to login and change pages as well as re-organize the site. The content of this page is up to you; it can be a client's site, or store, for example. In any case, you need to have a website's amount of content that will be displayed dynamically with a corresponding and appropriate design. Your content management system must have a complete and working backend. It must be styled with an administrative theme.



- 2. Content-Managed Band Discography.** This website will feature a database that manages the information needed to display a very detailed discography for a particular band. A minimum of 5 albums with complete info about songs, durations, artists, composers, etc, will be required. Also, accompanying album art is required, as well as concert photos. All of this content should be updateable through a password-protected admin section of the site. For brownie points, adding very short (and therefore legal) samples of the songs on each album, played through an embedded audio player is an extra, optional challenge.



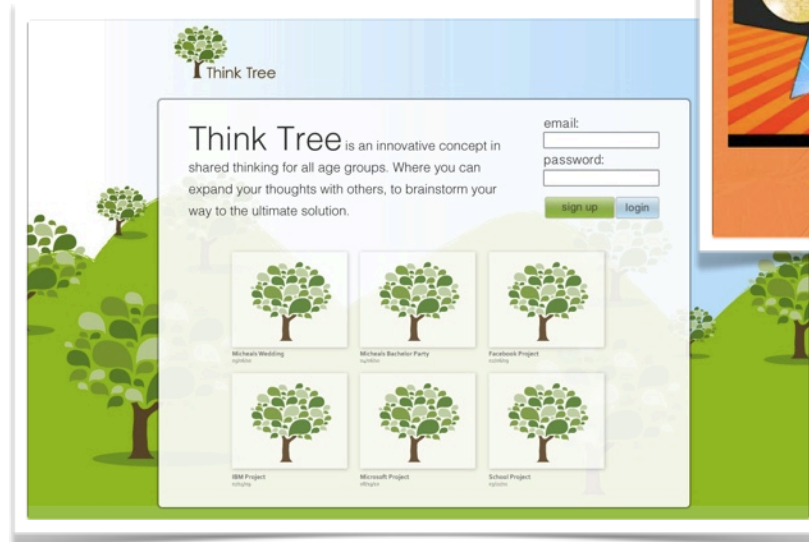
- 3. Dynamic Content-Managed Photo Site.** This will be a site that will allow the dynamic uploading of images, which will be cropped / resized and then thumb-nailed and displayed in a pleasing way. Your site will require an admin login that will allow you to edit the photos displayed (change them by uploading new ones, re-ordering them etc... not necessarily actual photo-editing). You should also be able to add titles and captions to all the photos, as well as display a photo credit. You can also connect to other web-apps, such as Flickr to fetch your photo content, however, you do have to store descriptions captions, credits in your own database.



you do have to store descriptions captions, credits in your own database.

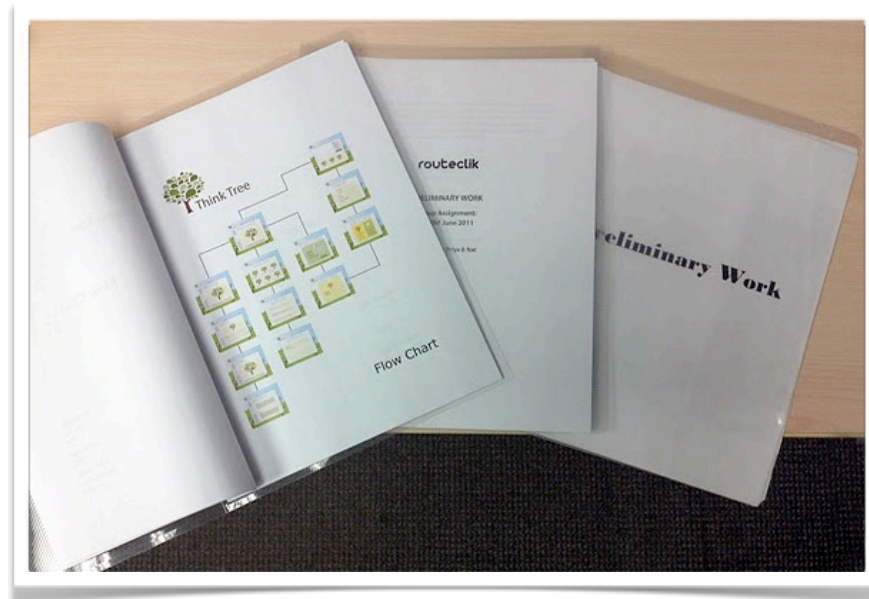
- 4. Multi-User Blog.** You are required to build a multi-user blogging system that will allow you to have a separate login for each member of your group. Each member of the group is required to write some content for their blog and display it on their site.

- 5. Custom Web Application.** If you find that you have an idea for a project that fits within the parameters of the above projects, write up a description of what you intend to do, and get approval from all your instructors, then you can build your own idea instead of one of these projects.



Design Requirements - 50%

1. 20% - All group members must submit a design concept for the project. The individual concepts must be developed at a mockup level. One of the prototypes will be selected by committee, specifically, group members and instructor will vote on the first scheduled composite milestone.
2. 10% - Design documentation package must contain:
 - A print out of each preliminary concept
 - A print out of a wireframe and final prototype of the chosen concept
 - A flow chart indicating site organization and functionality from a navigational stand point. A logic map or storyboard could also be used to describe the process of navigating the site.
 - A brief description of each individual team member's role in the project
 - A printed diagram describing all of the tables/fields in your system
 - Your design documentation will be marked on both the information it contains, as well as visual appearance and professionalism



3. 10% - All of the site pages must have a professional-looking graphic treatment. No boring text-only pages! Also, the graphic treatment of all pages must be consistent (not necessarily identical). Even the site administration pages need to have a look that relates visually to the rest of the site.
4. 10% - The design and execution must also incorporate all of the following:
 - One or more Flash animated elements
 - One or more Flash interactive elements using basic ActionScript 3.0
 - A layout structure, styled using only CSS code and marked up with div tags or HTML 5 semantic tags. The layout should be based on the mockup created for requirement #1. **Note: A table-layout based website will not be accepted, and will receive a grade of zero.**

Programming Requirements - 50%

1. 8% - Your site must feature well-formatted, organized and fully commented code. This goes without saying, especially for a project that may get significantly complicated. Furthermore, before beginning to build your project, you must create a diagram detailing the tables that will exist in the database. Each table should be labelled with its name, and the names of every field it contains. This diagram will be used as a basis of project approval for your programming course.
2. 6% - A user login system, featuring an admin user allowing you to log into the backend to make changes needs to be fully functional.
3. 10% - A content management system that is appropriate to the project. In any of the projects, you will require at least one table in the database to store the content, and another for the user login system. You will then have to use PHP to place the content in the appropriate place on your site. In most cases, you will have to build on the basic content table and add more tables and fields.
4. 4% - User and search-engine friendly URLs are a must; We have already done this in class, where the pages/section are all human-legible and easy to type in, and you do not see any URL variables or ".php" files.
5. 6% - Content submission through a form; you should be able to insert content into the system without having to use any other software other than your system backend. Make sure that you can't get to the admin section without having to log in.
6. 6% - Updating/Deleting existing content should be possible from the admin interface.
7. 8% - A template system should be present. In other words your layout and style for the site should not be replicated on each page. Rather, your content should be fed into at least one template file, which will then be used to generate the pages. You should also link the template to your style-sheet, JavaScript files and also to any images needed.
8. 2% - Your pages should validate in either XHTML Transitional or XHTML Strict mode of the W3C Validator.

```
34
35
36 /*
37 * getContent(): Load all the content and associated
38 * information for the current URL from the database.
39 */
40 function getContent()
41 {
42     $url = getUrl();
43
44     $query = "SELECT title, content, summary, timestamp, template
45             FROM " . T_CONTENT . " WHERE url='{$url}'";
46
47     $result = mysql_query( $query )
48             or die( mysql_error() );
49
50     // If there were no results from the database,
51     // then the page doesn't exist ... go to the 404 page.
52     if( mysql_num_rows( $result ) < 1 )
53     {
54         redirect( ERROR_404_PAGE_URL );
55     }
56
57     return mysql_fetch_assoc( $result );
58 }
59
60 /*
61 * redirect(): Redirects to the given address.
62 */
63 function redirect( $url )
64 {
65     // redirect to "/your_url_here"; this ensures that
66     // we are redirecting to the root of the site + the page's URL
67     header( "Location: /{$url}" );
68 }
69
70 /*
71 * getNavigation(): Generates an unordered list of links based
72 * on the contents of the menu table.
73 */
74 function getNavigation( $menuName )
75 {
```

"Do or Die" Milestones

Now. - Preparation

Deliverables: Meet with your group and:

- Determine what project you are going to be doing
- Exchange contact information
- Decide on your individual roles in the project
- Elect a team spokesperson, who will be maintaining contact with the faculty
- Decide whose hosting account will be used for development
- Start brainstorming!

November 30, 2011- Design Signoff

Deliverables: Individual concept designs. You will be voting with Amit as to which design will be the one used. Remember, everyone is responsible for his or her own version of the design. Immediately after the design has been chosen and approved, begin creating templates, graphics and tweaking the frontend design.

December 2, 2011- Functionality Signoff

Deliverables: Database diagram, and an overview of what the project will be. Your database diagram can be a simple sketch on a piece of paper. Each table in the system, as well as field names and data types should be explicitly written. Also, determine your roles in the project and elect a team spokesperson. Set up a sub-domain on one of your accounts, and install the CMS, to have it ready to start building. Also, define all the additional tables needed when your project has been approved.

December 14, 2011 - Final Tweaks, and Bug Fixes

Description: You should be finalizing the functionality and polishing up the remaining elements of the site. A discussion of how you will be presenting as a team should take place. Test the site on the instructor machine to be sure everything works as expected. Test, debug and test some more.

December 16, 2011 - Presentation Day

Deliverables: All the mentioned documentation in both the design and programming requirements, as well as the final live site. You will be presenting as a group, for 8-10 minutes.

Submission Details

Please comment EVERY file that you changed or added in your project like so:

```
<?php
/*  Group Members: John Smith, Jill Chan
    Date of Submission: <Put Due Date Here>
    Assignment: Composite 2
*/
?>
```

Once your site is finished, zip the entire thing, including images and put it on Bash in the designated folder. You will be required to present your assignment, but without the submission of your code, you will not receive credit or feedback or a grade for the coding component of the assignment.

Due Date

The Composite 2 Assignment is due on **December 16, 2011 9:00 AM Sharp.**

Please have any technical issues resolved by this time so that you are ready to present as a group and hand it in at the beginning of class.

Assigned Groups

The following are your groups for the Web Composite 2 assignment:

Group 1

- Anoop
- Victoria
- Stephen

Group 2

- Maxine
- Seyed
- Kirsten

Group 3

- Sara
- Nicki
- Sergey

Group 4

- Joya
- Zackary
- Sarvin

Group 5

- Priyanka
- Dylan
- Laura

Group 6

- Eric
- Michelle
- Mihai