COMP 3512 Assignment #2

Due Sunday Dec 9ish

Version 0.9, Oct 30 2018

Overview

This assignment is a group project that expands your first assignment in functionality and scope. It is broken into several milestones with different dates. The milestones are in place to ensure your group is progressing appropriately.

Some of the specific details for some of the milestones and pages will develop over time; that is, this assignment specification is a living document that will grow over the next several weeks.

Composition

You can work in groups of three or four for this assignment. It is also possible to work individually or in a pair, but I do discourage it; please talk to me about this if you are planning on working by yourself or as a pair.

If working in a group, each member needs to take responsibility for and complete an appropriate amount of the project work. **Be sure to consult the instructor at least one week prior to any due date if your group is experiencing serious problems in this regard.**

I feel foolish saying this in a third-year university course, but it is your responsibility to read all the assignment instructions thoroughly and carefully. If you are unclear about something, ask me. But before you do, read the material in question again!

Beginning

You will be using JavaScript, PHP, and MySQL in this assignment. Eventually this will mean your assignment will *likely* be submitted via Cloud9, since I need a way to view your web pages *and* view your programming code. If you wish to use a different hosting environment than Cloud9, you can do so as long as there is a way for me to view the pages and the source code.

If you do use Cloud9, each group member (and me as well) will have to be given RW access to the workspace via the Share button. Cloud9 allows multiple people to simultaneously edit the same file, so you will have to be careful about overwrites if you have two people editing the same block of code.

GitHub

Each group member will need their own Github account. You will also need to create a private repo on Github for the assignment. You have a couple of ways to do this. One way would be for one member in your group to create it (they would thus "own" the repo), and then add other members as collaborators. The free GitHub account doesn't allow private repos; if you sign up for the Student Developer Pack (https://education.github.com/pack) you can have free private repos while a student. The other way is to email me, and I can create a private repo under our department's github organization (https://github.com/MountRoyalCSIS) for your group. You would need to supply the github names or emails for each member.

You will want to push out updates fairly frequently (1-2 times a day when working on it, or more frequently if lots of work is being done). I will examining the commits when marking. You can push your content from Cloud9 to GitHub via the terminal, using the following commands:

```
git init     [only need to do this one command once for your assignment]

git add *

git commit –m "Fixed the rocket launcher"     [alter message and name as appropriate]

git remote rm origin [just in case your cloud9 workspace still linked to my github]

git remote add origin https:... your-repo-url.git      [specify URL of github repo … do this just once]

git push –u origin master   [login using your own individual github credentials]
```

For more information about Git and GitHub, read pages 571-577 of textbook (2nd Edition). There are many online guides to git (for instance, https://guides.github.com/introduction/git-handbook/).

Milestones

You will need to implement certain functionality by specific dates. These won't be marked but you will incur penalties (deductions in marks) if the milestone is not completed on time. These will be periodically updated. You will show me your completed milestones in the lab on the dates shown below.

Milestone 1. You must decide on your group members, set-up your github accounts, and create your workspace in c9 (if you are using it) that will be used collectively by your group. Create a simple version of your About page (see description below). The design and content of this page will change later as you progress through the assignment. But for now, indicate each the group member's name and github page (as a link). Also provide a link to the main github page for the assignment. *Due November 9.*

Milestone 2. Complete a preliminary design and implement it with CSS. Your design needs to be workable at both mobile and desktop browser sizes. This means you will need to make use of media queries. See textbook pages 290-293 for more information. In Lab 12, look at the provided CSS for lab12-test01 to see simple media queries at work. *Due November 16.*

Milestone 3. Create the following APIs in PHP (place them in folder named `services`).
Due November 23.

`services/genre.php` – with no parameter, return JSON representation of all genres. If supplied with `id` parameter, then return just JSON data for single specified genre.

`services/artist.php` – with no parameter, return JSON representation of all artists. If supplied with `id` parameter, then return just JSON data for single specified artist.

`services/gallery.php` – with no parameter, return JSON representation of all galleries. If supplied with `id` parameter, then return just JSON data for single specified gallery.

`services/painting.php` – with no parameter, return JSON representation of all paintings. If supplied with `id` parameter, then return just JSON data for single specified painting. If supplied with `artist` parameter, then return painting data for specified artist id. If supplied with `gallery` parameter, then return painting data for specified gallery id. If supplied with `genre` parameter, then return painting data for specified genre id.

`services/review.php` – Must be supplied with `painting` parameter; return JSON data for all reviews for specified painting.

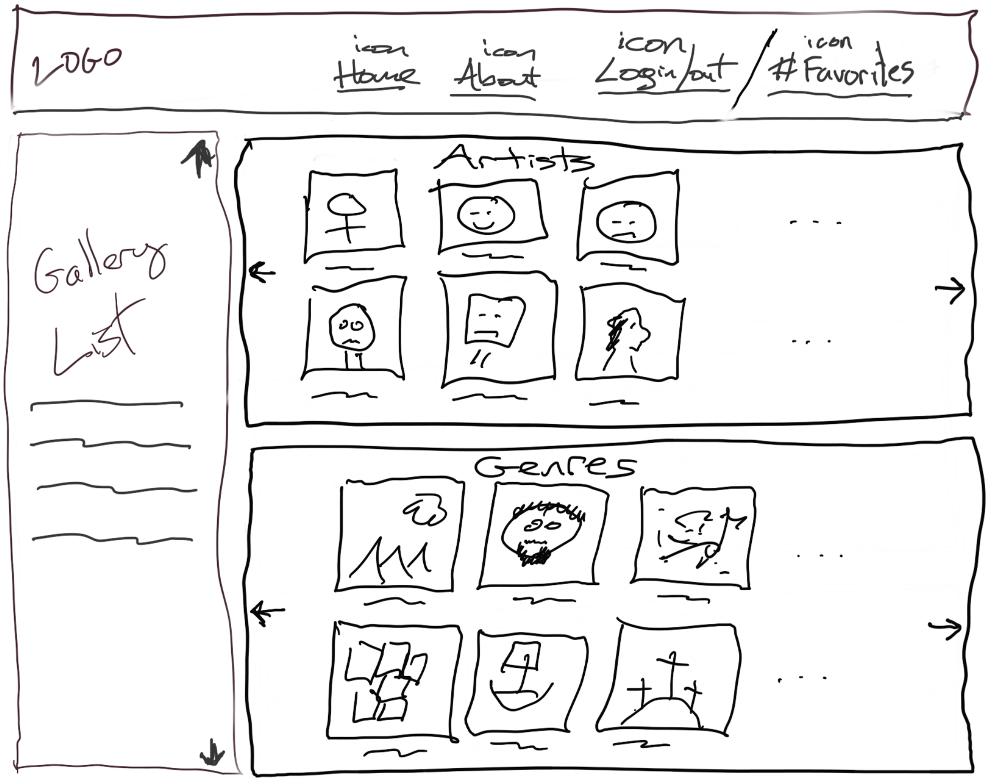
Functionality

Eventually, your assignment will have roughly the following functionality:

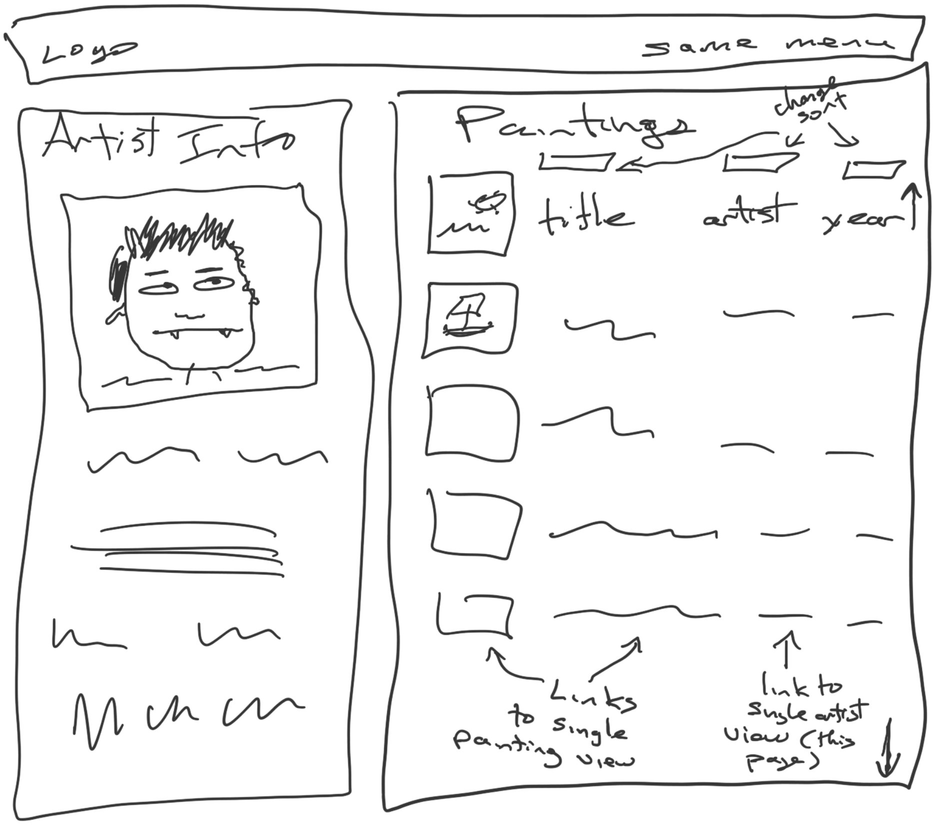
Home Page: will display list of all galleries, list of all artists, and a list of all genres. *I recommend making this page quite simple initially and then add in the horizontal scrolling towards the end.*

The data for these lists will eventually come from an API you create in PHP that retrieves information from a database. Artists and Genres lists will be horizontally scrolling lists of links. When the user clicks a gallery, artist, or genre, the user will be taken to the appropriate single item view.

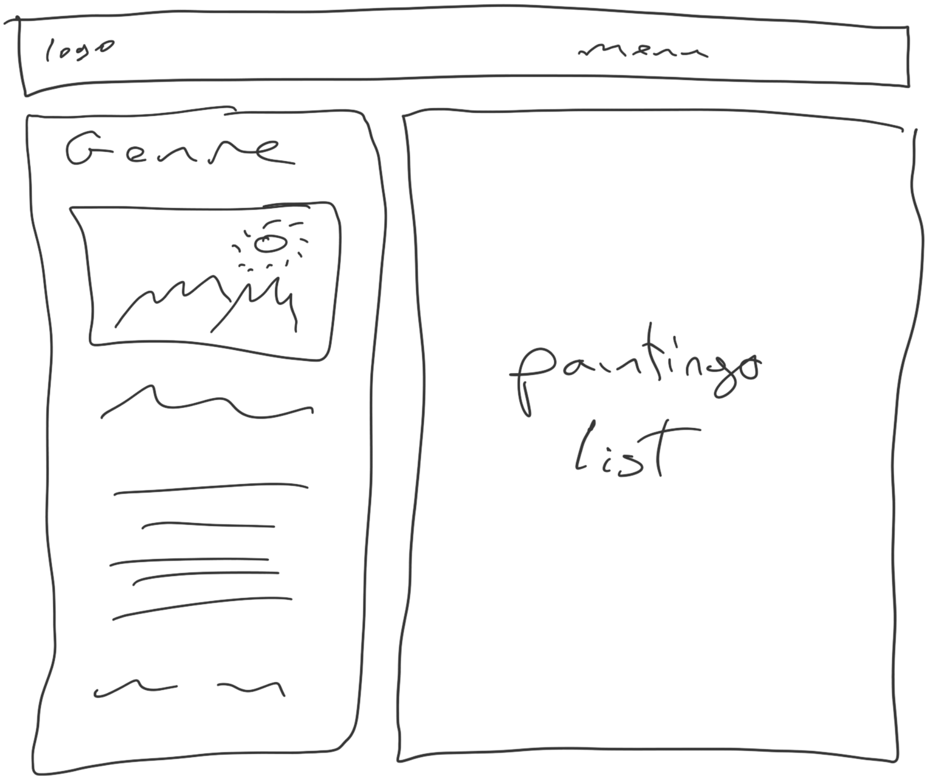
Your assignment will need some type of consistent header at the top of each page that contains a logo/title, and a menu consisting of links for home page, about page, to login/logout, and to view the favorite list. The options in this list will vary depending if logged in. If not logged in, the options will be: Home, About, Login. If logged in, the options will be: Home, About, Logout, Favorites. It might be nice to use icons and text for these menu labels.

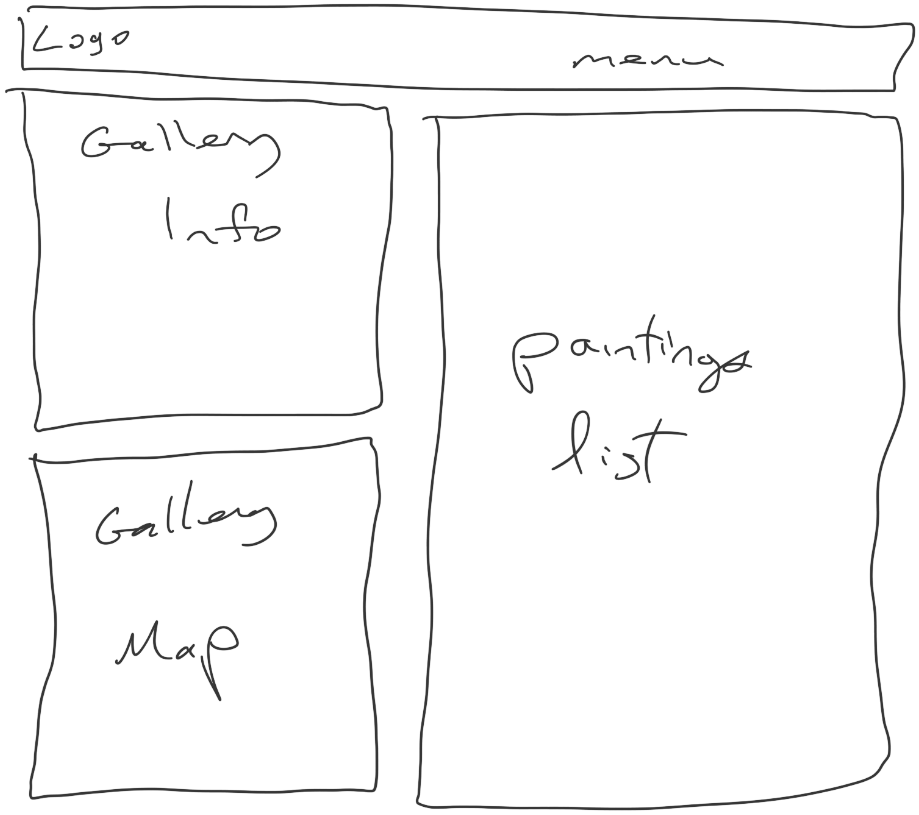


Single Artist Page: will display information for a single artist. Which artist? Presumably the artistID will be passed as a query string to the page. Information from the Artists table in the database will be displayed in the information section. As well, display a list of that artist's paintings. This painting list component will show up on multiple pages. The title and thumbnail image will be links to the Single Painting view; the artist name a link to the Single Artist Page. There should be some way to sort the list by title, artist, or year.


Logo
same menu
Artist Info
Paintings
change sort
title
artist
year
Links to single Painting view
link to single artist view (this page)

Single Genre Page and Single Gallery Page: will display information for a single genre or gallery. Information to be displayed comes from the Galleries and Genres tables.

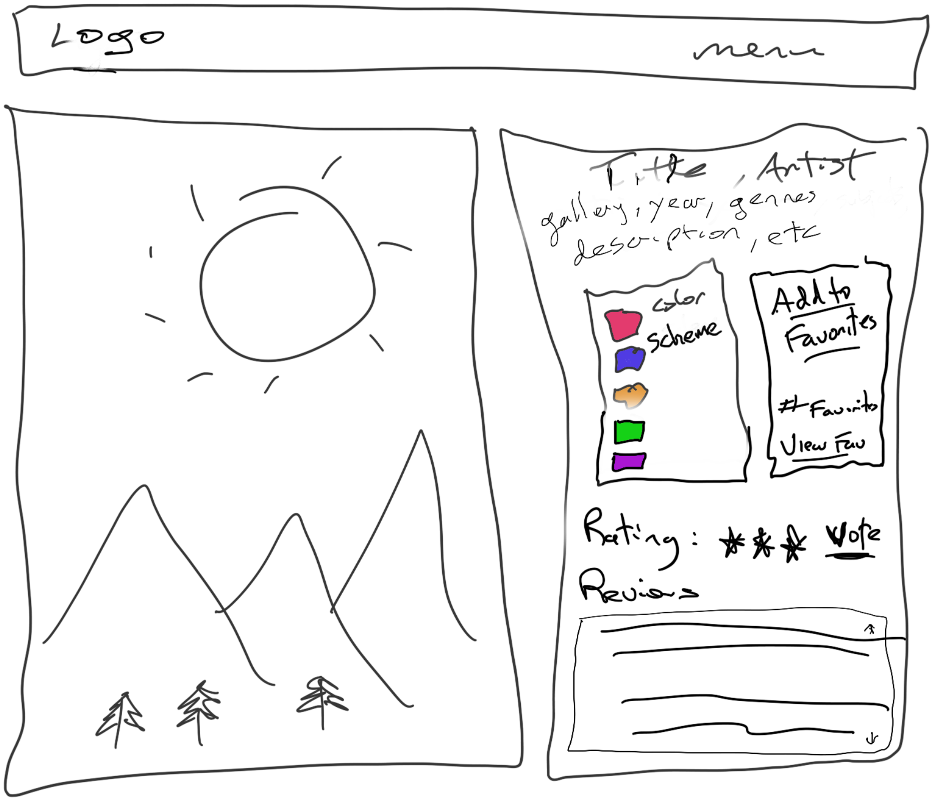




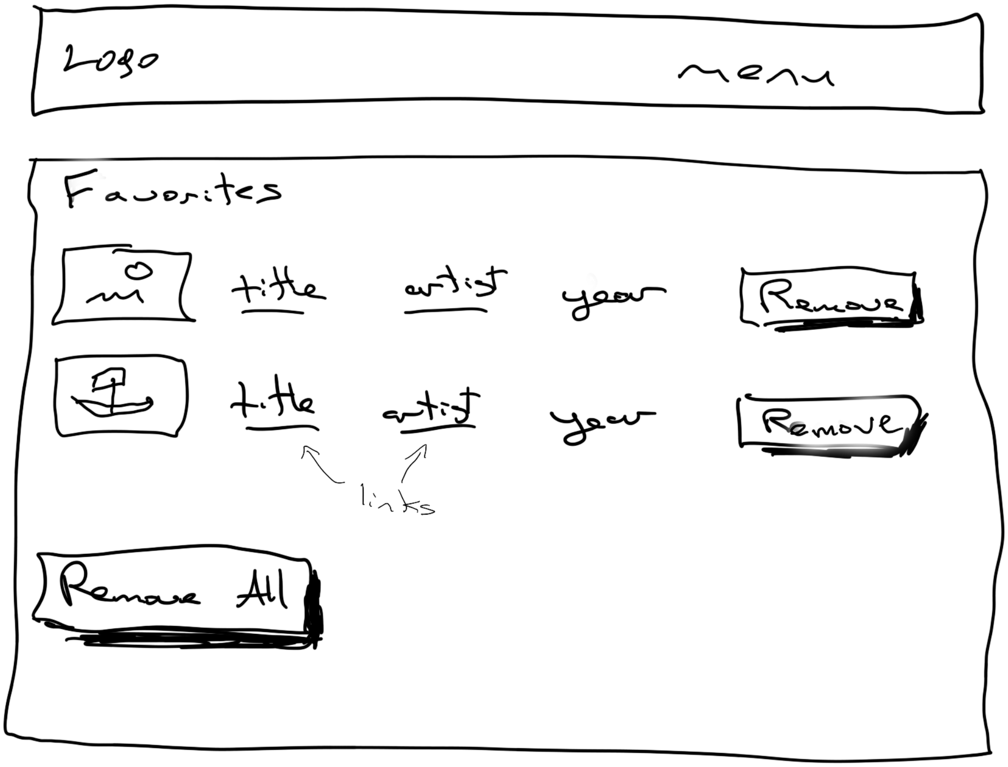
Single Painting Page: will display information for a single painting. Which artist? Presumably the paintingID will be passed as a query string to the page. Information from the Paintings table in the database will be displayed in the information section.

There is quite a bit of data to be displayed so being able to do so effectively will impact the design mark of the assignment. Each painting has a single artist, but multiple genres and reviews. The color scheme will hopefully be generated using the Google Vision API. The rating can be calculated from the Reviews.

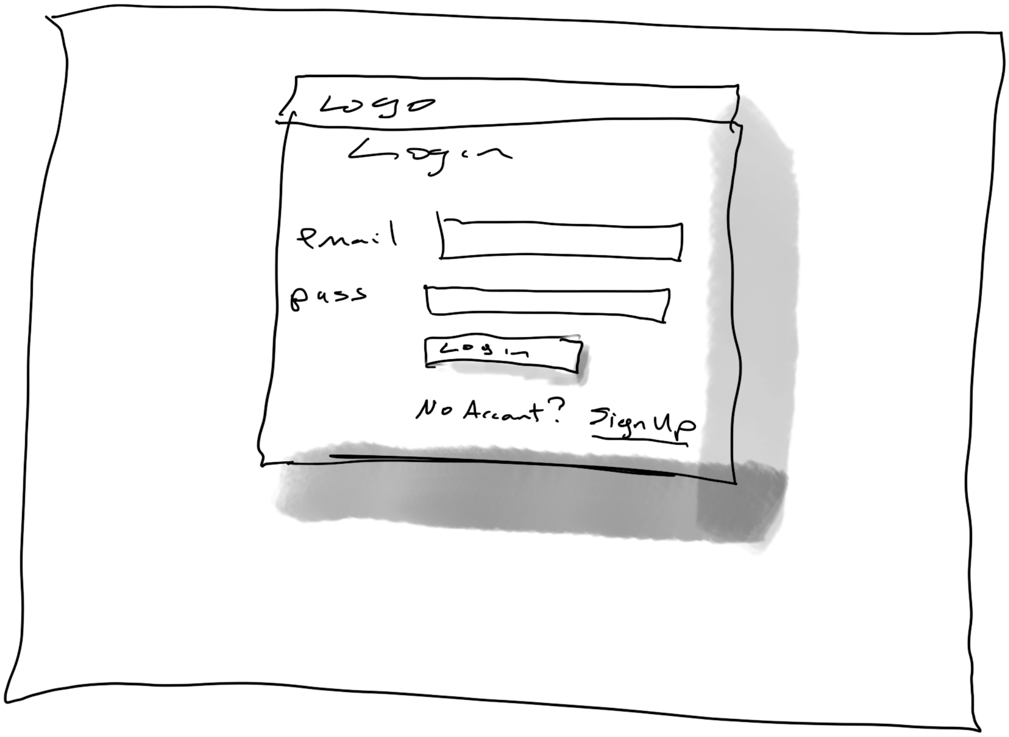
The user can add a painting to her favorites list. If the user is logged in, they will be able to add a review and rating to a painting.



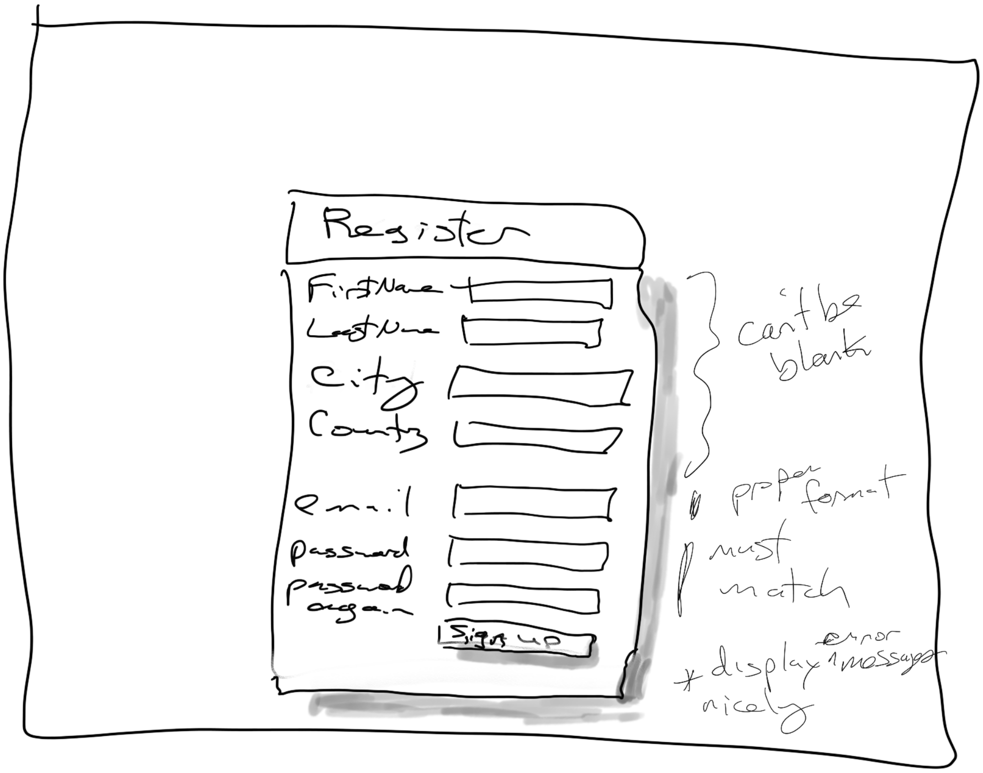
Favorites Page: will display list of logged-in user's favorited paintings. The user should be able to remove paintings singly or all at once from this list. You will implement this list via PHP sessions.



Login Page: will display a login screen. I will provide more information about how this is to be implemented after we cover security later in November. For now, simply check for the email abc@abc.ab and the password abcd.



Registration/Signup Page: will display a registration form. Perform the following client-side validation checks using JavaScript: first name, last name, city, and country can't be blank, email must be in proper format (use regular expressions), and the two passwords must match and be 6 characters long. Be sure to display any error messages nicely.


Register
FirstName
LastName
City
Country
email
password
password again
Sign up
can't be blank
proper format
must match
* display error messages nicely