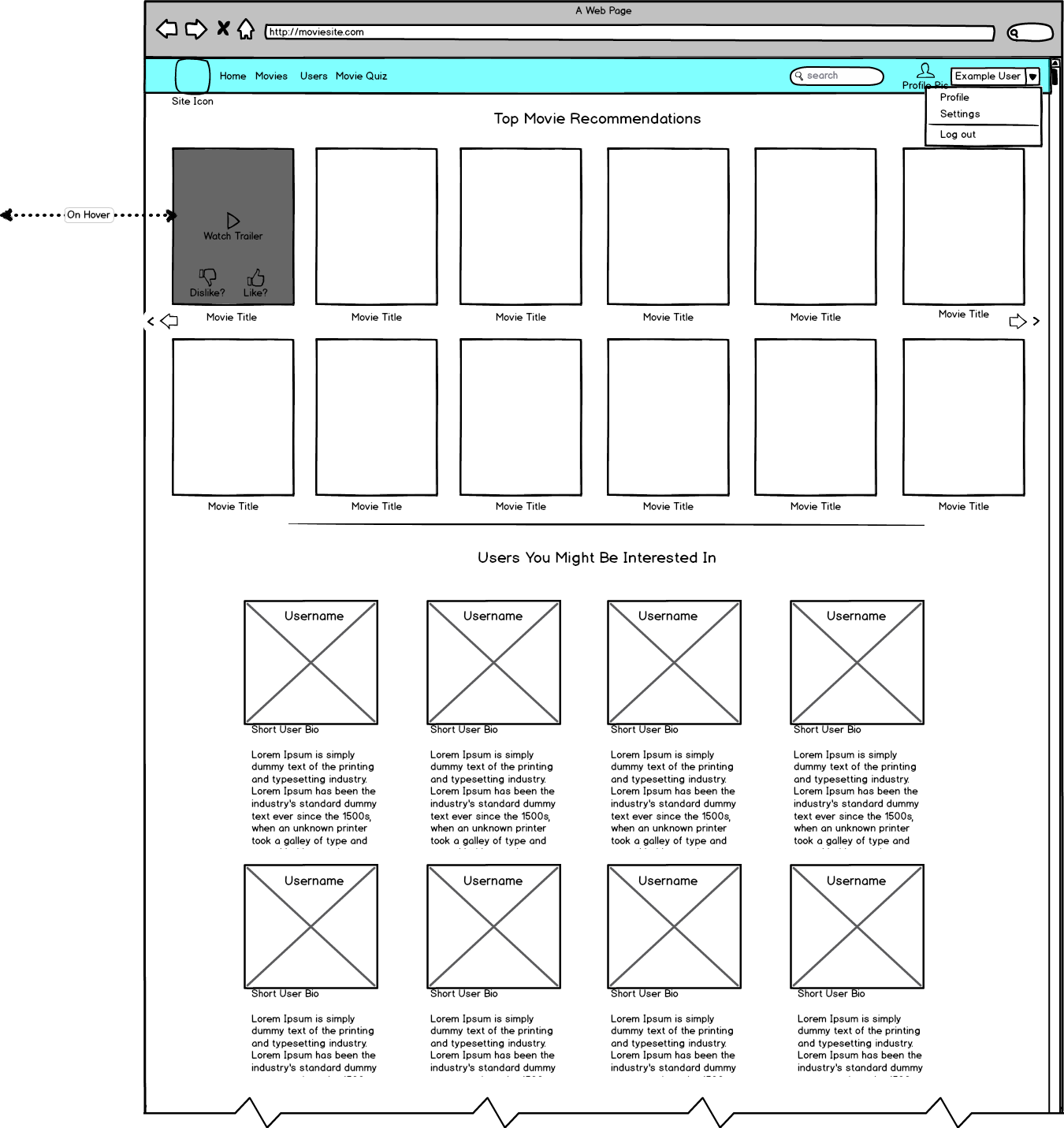
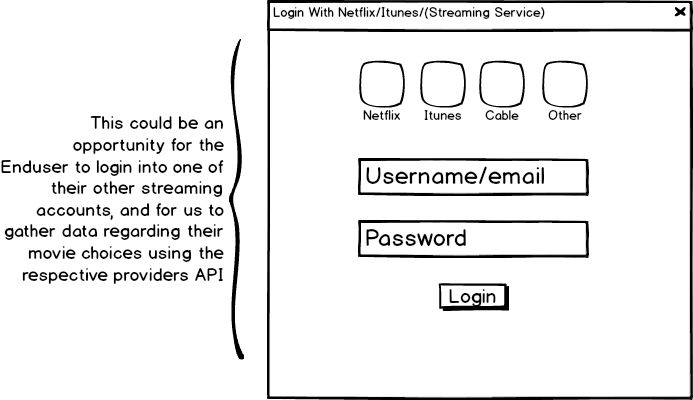
Panafold Movie Site Challenge

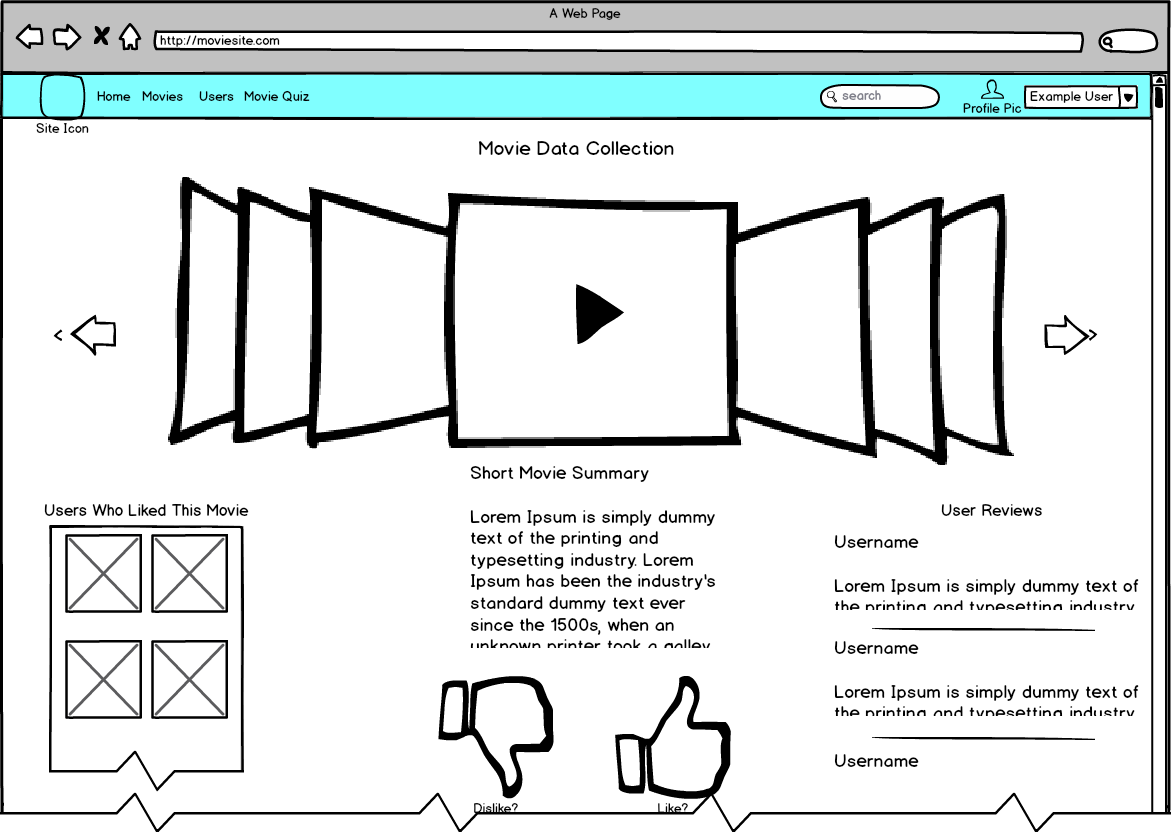
So I wasn’t entirely clear if you wanted me to write any code, but I assumed by “sketch interface” you wanted a visual representation of the code that I would implement. So I used a application called Balsamiq to create some mockups.

This is one of the easiest ways to get user info. By logging into other streaming accounts, we can use backend algorithms to convert their respective view history to something our site can parse

This is the main page. It is where the user can view the recommendations that have been generated through machine learning algorithms.

Up top, the user can scroll through movie recommendations tailored to him/her. On hover, options to like or dislike the selection appear, as well as an option to view trailer.

User follow recommendations are below, including the user icon and a short user bio



This is the manual method of gather data for recommendations. Initially, the user is provided with a random assortment of movies. The User can view the trailer and read a summary. Also, the user can see other users who liked that movie, and user reviews. The user can then indicate and interest or disinterest in the movie. As the user continues to do this, the suggestions become more tailored to their taste. For the user, they can build a queue of movies that they would like to see.

The core of the recommendations on this site is machine learning. Data containing the interest of each user has to be stored and updated as they user the site. I don’t have any real experience with machine learning, but I understand the basic concept. Also the use of data gathered by other streaming sites can play a large role, and save our site a lot of processing.

How Would I Register 500 New Users?

1. Using an established database API like parse. I use parse all the time, and it’s a fantastic way to get your site started with the initial few thousand users. The register form would be written with angular, and registration would be super simple with no server side code.
2. Using a form that connects to mysql using PHP. This gives us more control over our data than a database API, but it requires that we run server side PHP.
3. Using Django. Django allows us to manage User objects with python, and the site code is very dynamic and loads quickly. Django is very efficient and the modules make the site very easily changed and expanded. But the code can be complicated, and since the html is loaded through python, one small coding error can prevent the entire page from loading.