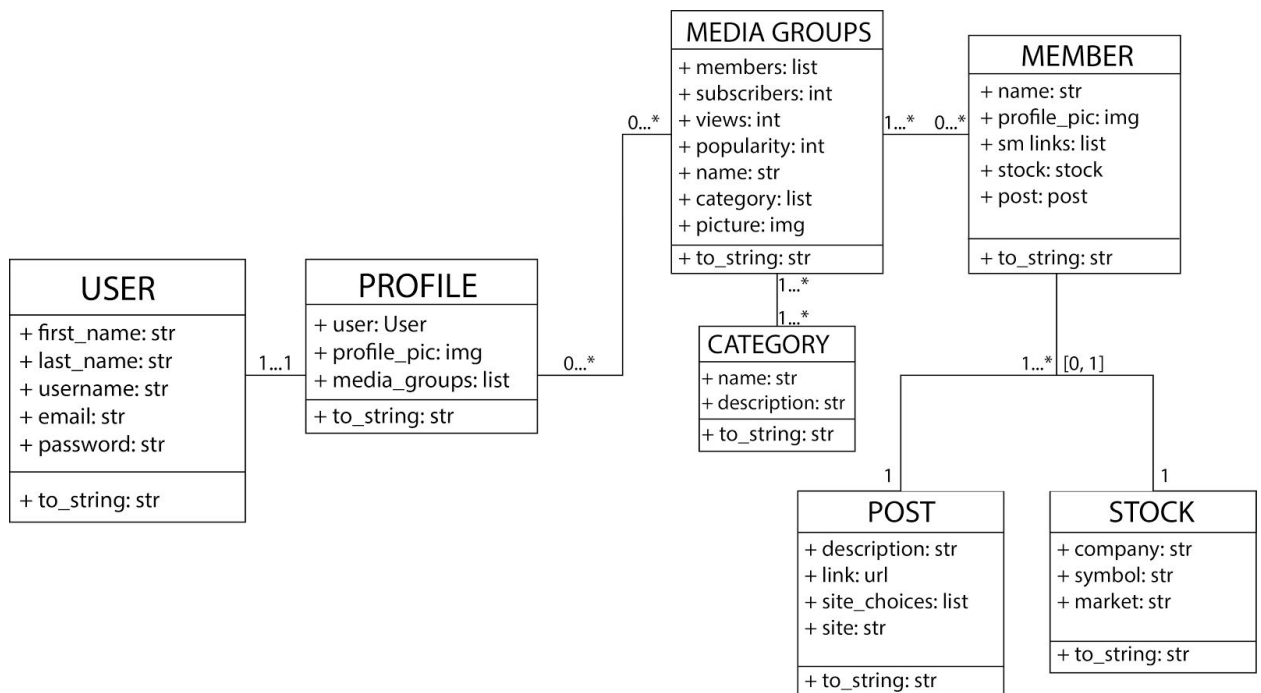- **Title:** Anonymous
- **Subtitle:** FlexFeed
- **Semester:** Fall 2017
- **Overview:** FlexFeed is a dynamic social media aggregator that allows users to see only what they want from their social media outlets. With FlexFeed, social media becomes a viable source of news from your favorite categories, like Technology, Music, Cars, etc., free of ads, suggested posts, updates from friends and family, and anything else a user seeking information on their favorite worldly topics wouldn't want to see.
- **Team Members:** Cory Knopp, Ray Leo, Lauren Bornfeld, Sebastian Lacki, Sasha Krutiy, Catherine Mangeym
- **Github Repository:** https://github.com/coryknopp/FlexFeed
- **User Interface:** *A detailed user interface breakdown can be found at the end of this report.*
- **Data Model:**

- **URL Routes/Mappings:**
  All routes are prefaced with /flexfeed

/	This is FlexFeed's homepage which, by default, displays social media posts from each group the user is subscribed to.
/<group_id>	This is FlexFeed's homepage with a group_id used as a parameter. This results in a homepage which displays social media posts from one group as selected by the user.
/groups/	This page displays blocks representing the groups the user is subscribed to. Clicking on groups displays the social media feed of that group. From this window, the user can click on to manage their groups.
/login/	This is a simple login form that users are directed to when attempting to log in.
/discovery/	This page displays social media groups as blocks that the user has not subscribed to. It servers as a way to "discover" what other users are looking at.
/settings/	This is a simple settings page which displays the user's settings with an option to change them.
/editprofile/	This is the window with with the user can change their settings (name, email, etc.)
/groups/edit_group/<group_id>	This is the url mapping used to edit groups the user is subscribed to. When given no argument, it displays a list of icons representing each of the user's groups. When it is supplied a group id (when the user clicks on a specific group), it displays the same list, but with a breakdown of the social media accounts in the selected group on the right hand side of the page.
/groups/delete/<group_id>	This URL route deletes a group from the user's list of subscribed groups by group id.
/groups/add/<group_id>	This URL adds a group the user's list of subscribed groups.

- **Authentication/Authorization:** FlexFeed only uses two permissions levels: admin and user. In terms of using the website, admins see the same things that users does, the only difference is that admin has access to the administration interface.

  Non-users can access FlexFeed, but are limited to being directed to log in. Without logging in, non-users cannot view a feed or save groups. As a result, they do not have access to the discovery page or groups page. Additionally, they do not have access to the settings page either as non-users have no settings.
- **Team Choice:** Our team choice feature adds a live stock market feed to the top of our homepage. To make this work we used a third party widget from tradingview.com. Before this, we had used mock data to create a banner of stock information, and therefore had to store that mock data in our database. With the widget, however, we were able to change our data model to only include the name of the company and the stock market abbreviation of that company.

  We also added a search functionality in the edit groups page that allows users to search for members they want to add to their groups.

- **Conclusion:** The highlight of our team's experience was that communication is key. Almost all of the issues we encountered were a result of bad communication. These issues ranged from not being on the same page about what features we needed to implement and disagreeing with design choices to literal merge conflicts with code and database changes. By the end of the project we had all become better communicators and ran into issues like this much less, and we learned that it was particularly helpful to have one or two voices that directed the rest of the team. For example, one member began outlining tasks that needed to be done for a given project, and assigned them to team members. As a result of this initiative, each of us were faced with a digestible task as opposed to staring at a long list of to-dos, and we found that this was the most efficient way to work without conflict. Knowing this before we started the project would have certainly made things easier.

  As far as technical hurdles go, GitHub was a source of many of our challenges despite how useful it was. At the beginning of the project, none of us had substantial experience with git, and as such we were left to our own devices to research things like merge conflicts, using branches efficiently, and how to rollback changes after something disastrous happened. We also constantly ran into database conflicts as git cannot merge sqlite files which we eventually resolved by leaving it up to only two members to handle creating data and modifying the database. Just like communication, however, we learned from these technical challenges, and we all now know more about using git.

User Interface Breakdown
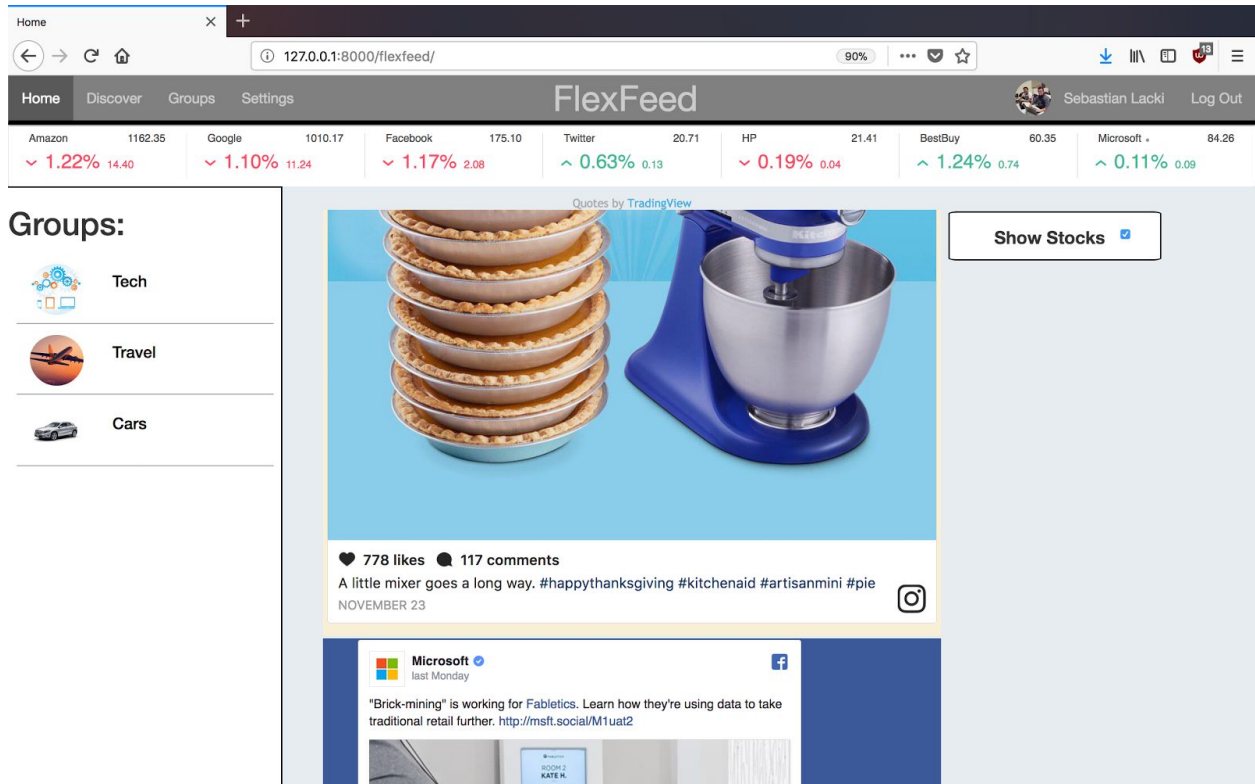
This is FlexFeed's homepage or *index* page. This page is intended to display the feed that our application is built around. Here the user can see a feed combining their choice of posts from different social media outlets. It also contains a live stock market feed pertaining to the accounts the user is viewing on their feed.
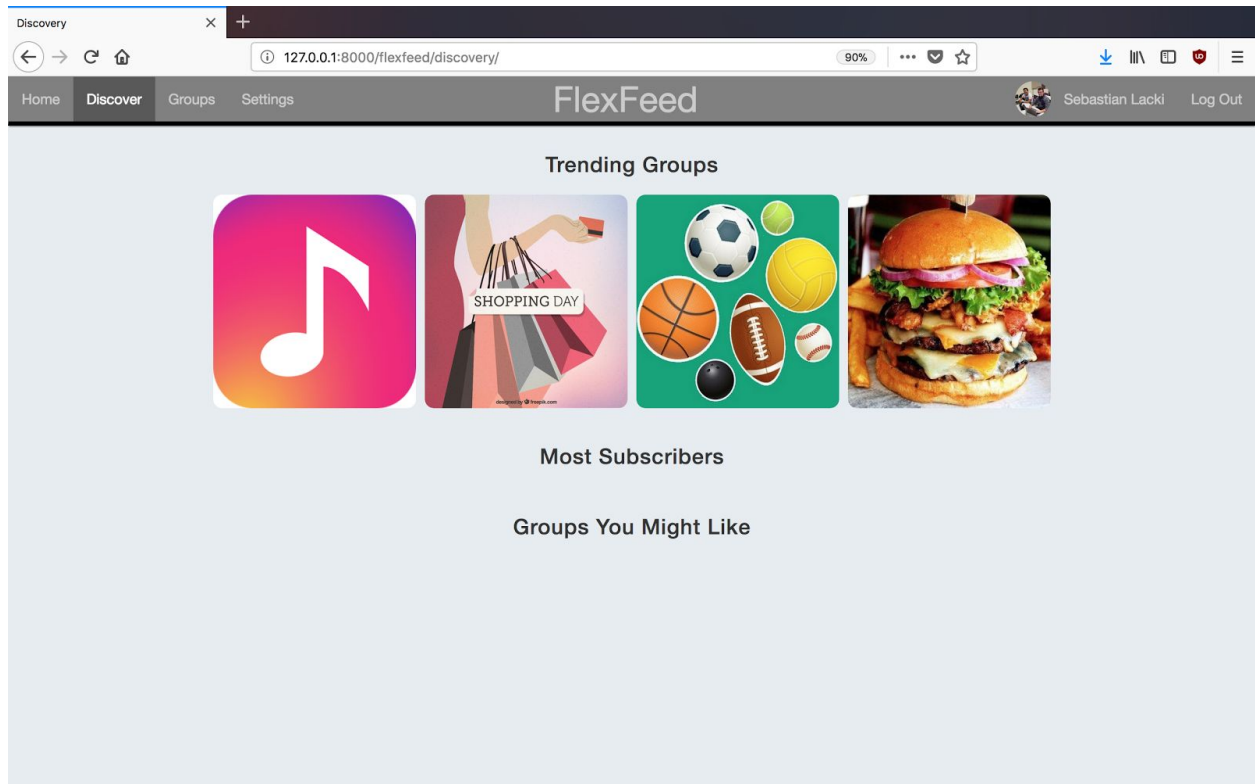
This is FlexFeed's discovery page. It is exists to display groups that a user is not subscribed to so that they may discover new ones to add to their feeds.
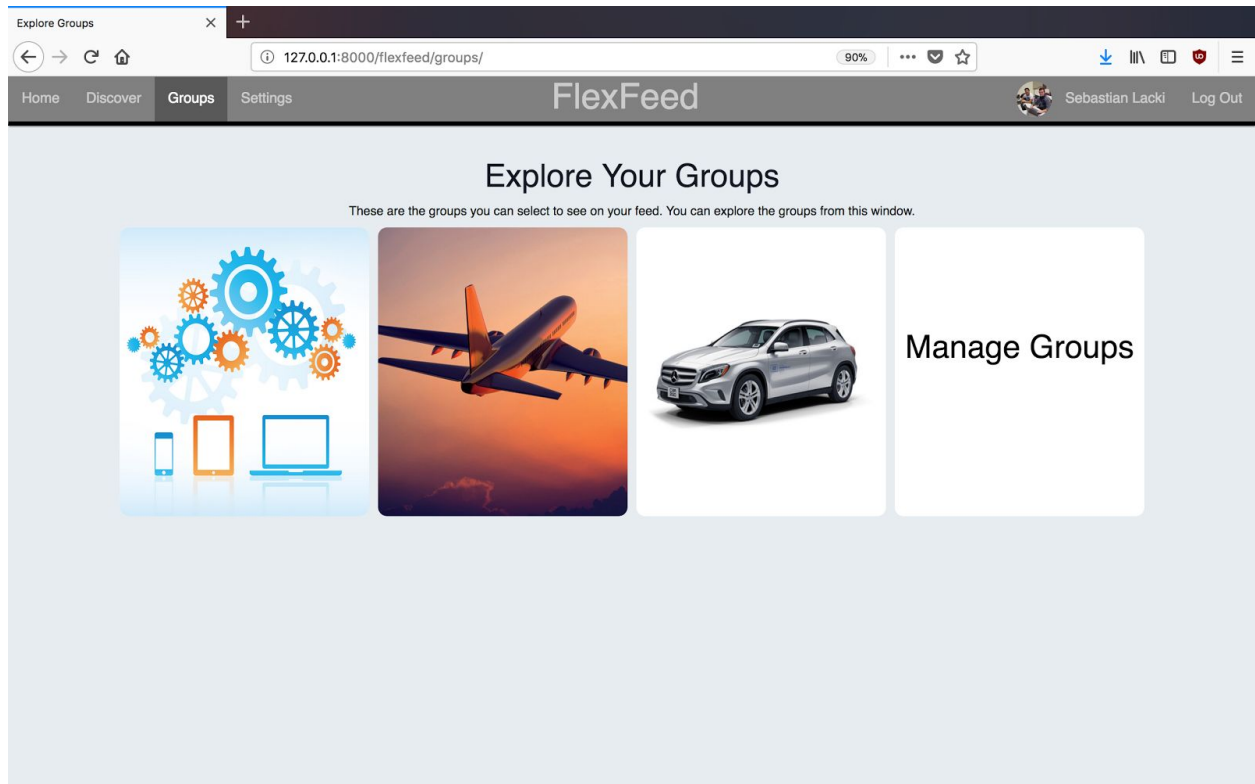
This is our *editprofile* page. It is a simple form which allows users to change their settings.
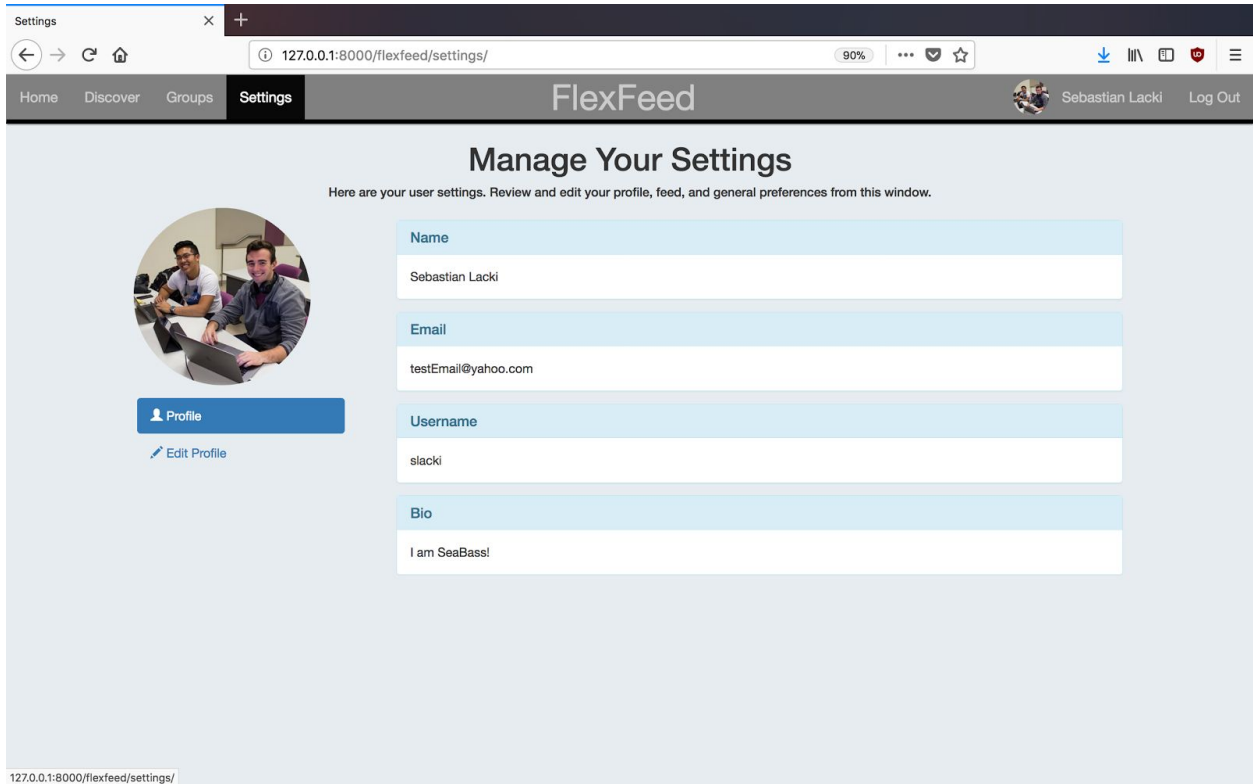
This is FlexFeed's settings page which exists to show users what their settings are.

This is the *groups* page. It shows users the groups they are subscribed to, and hovering each group displays data about the groups. It also gives the user the option to manage their groups which directs them to the next page.
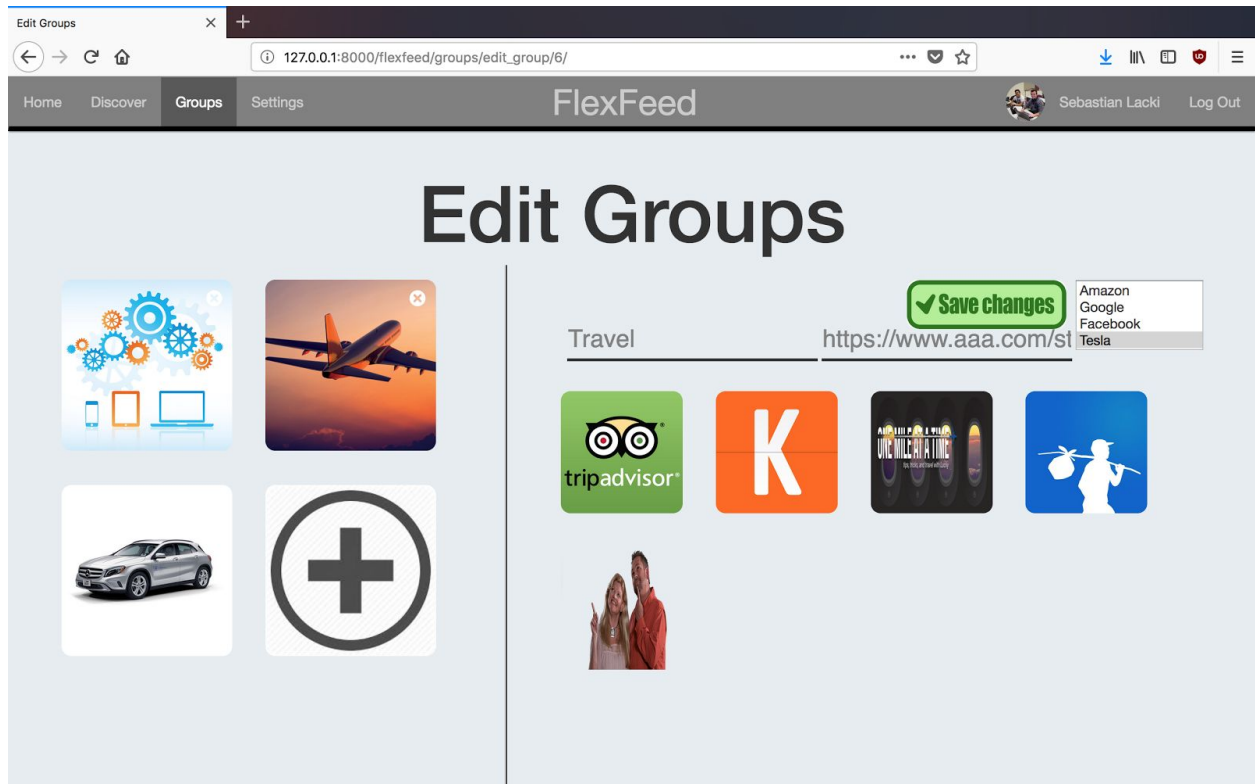
Finally, this is the page the user visits to edit the groups they are subscribed to. On this page, they can add new accounts to selected groups. They can create new groups altogether, and they can remove them from this interface.