Grading Directions

Best Parts:

- View elements common to many pages (footer, navigation, importing common css/js) are contained in their own files and imported into the pages, to avoid duplication of code and easier modification. (https://github.com/6170-fa14/tianm_proj2/blob/7620a1fd71c102ac290e5b25c438af62e7567480/views/navlinks.ejs, https://github.com/6170-fa14/tianm_proj2/blob/7620a1fd71c102ac290e5b25c438af62e7567480/views/imports.ejs, https://github.com/6170-fa14/tianm_proj2/blob/7620a1fd71c102ac290e5b25c438af62e7567480/views/footer.ejs)
- Editing, deleting, favoriting, and unfavoriting posts are done by JQuery Ajax calls so the page updates without reloading, with nice transition animations. (https://github.com/6170-fa14/tianm_proj2/blob/7620a1fd71c102ac290e5b25c438af62e7567480/public/javascripts/home.js)
- Routes are structured for easier management and understanding such that index.js (the default route, '/') deals with rendering pages, posts.js ("/posts") deals with managing (creating/editing/deleting/favoriting) posts, and users.js ("/users") deals with user account-related actions (logging out, logging in, creating new users, and following users).
- Additional feature: favoriting posts. Users can favorite/unfavorite posts on every page where you can view posts-- the homepage (which shows all posts by all users), the My Feed page (which shows posts by users that the logged in user is following), or the My Favorites page (which shows posts favorited by the logged in user). You cannot favorite your own posts, but can instead edit/delete them. On each of these pages, favorited and non-favorited posts by other people have different icons (filled star and empty star, respectively), which is consistent with external examples of favoriting.
- User profile pages show a "follow" button if the user is not currently followed by the logged in user, and "unfollow" if the user is followed. (https://github.com/6170-fa14/tianm_proj2/blob/7620a1fd71c102ac290e5b25c438af62e7567480/views/index.ejs#L17)

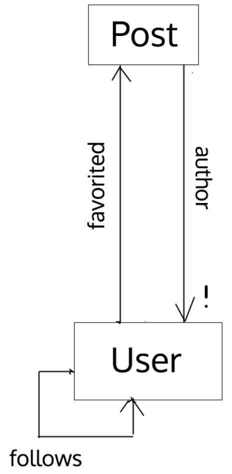
Design Challenges:

- How to organize routes in a logical way -- options included which HTTP method to use, and how to organize URLs. See the third item in Grading Directions for evaluation/which chosen.
- How to handle editing -- options included directing the user to a new page, or reloading the page with a new text box. I chose to keep everything on the same page to keep most of the activity on the website central to this one location (see Grading Directions #2)
- How to design the data model -- The data model I chose had:
  - A User (which has a username and password) follows multiple users, and favorites multiple posts. It is not aware of posts that it has posted, or users that follow it.
  - A Post (which has text for its content) is posted by a single user. It is not aware of who has favorited it.

  I went with this data model because it allows me to access posts for the "My Favorites" and "My Feed" page easily. Other options I considered included giving each User a list of posts, or giving each post a list of users which have favorited it. Giving each user a list of its own posts would be useful for the "My Fritters" page and profile pages, but finding all the posts posted by a user can be done with a single query so I didn't feel like adding the additional (likely large) data field to each User object would be necessary. Giving each post a list of users which have favorited it makes it necessary to look through each post's list for the current logged in user, which would be inefficient.

Data Model:

(see Design Challenges for explanation of data model)


Post
favorited
author
!
User
follows