Eric Gan
Project 2-1

Grading Directions:

There are a few parts of my code that I would like to highlight.

1) #L46-54 in users.js

   When a new user signs up I insert a new record into our database. Each record contains a username, password, and freets. Note that currently I am storing my freets in a list, which I know is not the best way to do this; however, since we learned about relative databases and mongoose after I already wrote my code, I wasn't able to use these tools and will make this change in the next part.

2) #L29-40 in profile.ejs

   I wanted to prevent users from posting on other peoples pages and only be able to post on their own profile page. To do this I passed in req.session.userName when I rendered the profile page. I also passed the username of the profile trying to be visited. Thus, if the two matched then I allowed access to post, edit, and delete; otherwise, I blocked access to these.

3) #L68-73 in users.js

   I first make sure that when a user tries to post a freet that it has length greater than 1, which prevents the user from posting an empty freet. Then to update the users freet, I find the record with the username and push this new freet onto their list of freets.

Design:

There were many design decisions I had to make. One was the way to do user authentication. To do this, I used express-sessions and created a session and stored the username in this session when a user successfully logged in. Then the user could only post, edit, and delete freets on his own profile page, but could still view the freets of the others. Then when the user logged out we would destroy this session. Other design choices I made were that there could not be any repeat usernames. To do this, when I stored usernames I just made them all lowercase and when a user logged in I converted his input to lowercase and then looked to see if it existed in the database. I also made sure all freets had length greater than 0 whether they were first being created or being edited. Finally, I chose to represent each user as a record in my database and each record kept track of a username, password, and list of freets. Note that currently I am storing my freets in a list, which I know is not the best way to do this; however, since we learned about relative databases and mongoose after I already wrote my code, I wasn't able to use these tools and will make this change in the next part. Also, we haven't learned this yet, but I would like to encrypt my passwords for part 2.

Credit:

Used Twitter Bootstrap: http://getbootstrap.com/2.3.2/
Used Font Awesome: http://fortawesome.github.io/Font-Awesome/
Used the TA's sample-express-app from GitHub: https://github.com/6170-fa14/sample-express-app/