

## Harihar Subramanyam Fritter

### Grading Instructions

#### Accessing the App

Option 1: Go to this URL:

<http://fritter-hariharfitter.rhcloud.com/>

Option 2:

1. Clone the repo
2. Run `npm install`
3. Start MongoDB server with `sudo mongod`
4. Start app server with `npm start`
5. Navigate to <http://localhost:8080>

If you want to see the app in action, I've posted a demo of it on YouTube

<http://youtu.be/qM0cn2Haad0>

#### Using the App

##### Login/Register

1. First, you need to register, so enter your **username, password, and confirmed password** and click **Register**.

##### Tweeting

1. At this point, you'll be brought to your home screen.
2. To make a tweet, enter your tweet into the **textarea** and click **Make Tweet**. You'll see the tweet displayed in your feed below the button.
3. To edit a tweet, click the **Edit Tweet** button, click the text of the tweet, write something new, and click **Done**.
4. To delete the tweet, click the **Delete Tweet** button.

##### Following

1. From the home screen, click the **X Follower(s) / Y Followed** button.
2. This will take you to the Followers page. Now, type a **username (or leave the textarea blank to search all users)** and click **Search**
3. This will search through the users. From here, you can click **Follow** to follow the user.

4. If you ever want to unfollow the user, go to the followers page and click **Unfollow** next to the user's name.

#### Private Messages (Extra)

1. You can send private messages to friends (i.e. users who follow you, and whom you follow back)
2. From the home screen, click the **X Unread Messages** button.
3. This will take you to the messages page.
4. There is a toolbar on the left labeled **Friends**, it lists all your friends. If you don't have any friends, you'll need to follow a user and get them to follow you back (if you're testing the app, you can create two users and make them follow each other)
5. When you select a user, you can type a message into the **textarea** and click **Send** to send the message

#### Highlights

The first highlight is that this project as a Single-Page Application which fetches JSON data from a REST API. This way, the client and server code is nicely separated. For instance, all the client side code is in the `public/` directory and all the server side code is in `server.js`, `models/`, `routes/`, and `views/`.

Another highlight is that I use the Model-View-Controller design pattern effectively.

On the server side, the controllers are in the `routes/` directory, the models are in the `models/` directory, and the views (there is only one, since most of the view work is handled by the client) are in the `views/` directory.

On the client side, the controllers are in the `controllers/` directory, the models (i.e. the code connecting to the API) are in the `util/` directory, and the views are in the `views/` directory.

The final highlight is that the API endpoints are very simple, which makes the URLs clean and easy to remember. You can find them in the `routes/` directory. The endpoints are:

#### Authentication

`auth/login`  
`auth/register`  
`auth/validate_session`  
`auth/logout`

#### Following

`follow/followers`  
`follow/followed`

```
follow/friends  
follow/make  
follow/delete
```

```
Search  
search/users
```

```
Tweets  
tweets/since/:date  
tweets/make  
tweets/delete  
tweets/edit  
tweets/all  
tweets/followed
```

```
Messages  
messages/mine  
messages/send  
messages/unread  
messages/unread_names  
messages/from
```

## Help Wanted

There is no test suite for the application (we haven't learned how to do testing in Node yet). I know that Node has an assert module built in, but it's quite low level. Some questions I'd like help with are: 1) Which testing module should I use? 2) How do I do end-to-end testing of the application (i.e. interaction of both server and client), and 3) What techniques can I use to write easily testable code?

## Design

### Data Model

The data model is depicted below:

