# Assignment4
# This will be the readme.pdf that we eventually hand in
Food Share Application (without this text of course)

# Getting Started:
- navigate to the a4p2 directory
- npm install (this should install the required dependencies in package.json)
- mongod (start the database - should listen on the default port 27017)
- node server.js (start the server process)
- open browser, localhost:3000

# Testing using Mocha:
- We provided some basic tests in the test.js file. Run `mocha test.js` to run the tests.
See comments in top of this test.js file for detailed information about the tests.

# Notes:
- The database name is "foodshare", so if you want to see quickly what the database contents are, type mongo foodshare,
then run mongo queries

# Basic Usage / Tests:

// Signup and login
- The first thing you should probably do is sign up. In the nav bar, hover over "Sign Up", and select
"sign up as a user"
- Fill in the form, submit sign up form
- In the nav bar, go to the Log In page
- Login with the name and password you just signed up with
- You should be in the userProfile page now.
- Can press the "logout" button to sign out of your account.
- Try signing in as a deliverer now: Select "Sign u

p as a deliverer"
- Fill / submit form, then go log in like before
- Should end up in the delivererProfile page now
- Can press the "logout" button to sign out of your account

Notes:
- we provide some sample users, in users.json and deliverers.json. Can use mongoimport to load these documents into the foodshare database if you want (eg. mongoimport --db foodshare --collection users users.json)
- users have unique names, so you can't sign up as "Bob" twice. You can however, have a "Bob" user and a "Bob" deliverer

// Making an order
- Login as a "user", and fill out / submit the order form.
- Successful orders will appear in your order history

// Accepting an order (NOT DONE YET)
- Login as a deliverer, and use the search functionality on the deliverer profile page to search for orders to accept

// Leaving feedback
- We have a feedback page that allows anyone to leave feedback about any other user in the system. Using it should be quite self-explanatory. Note: you need to be logged in to give feedback.

// Admin functions (NOT DONE YET)
- Like the assignment handout says, go to http://localhost:3000/admin to see the admin apge.

- Follow the instructions given on the page

# To Do:

- Orders - User makes order from their profile page, deliverers can confirm orders on their profile page
- Admin

- Code cleanup. Notice the parallels between user and deliverer sign up / login code in server.js. Some of this is repeated code,
and can be optimized with functions / callback functions
- Testing
- Performance
- Writeups
- Video