17-356/17-766: Software Engineering for Startups
Homework 5, Engineering Sprint
100 Points
March 19th, 11:59 pm
Part of this homework is to be done with your group.

In this homework, you will work with your team to "finish" building out your Dronuts project, specifically focusing-in on your project from an engineering perspective.

The ***deliverables*** are as follows:

1. **A short (approximately 1 page) writeup <u>(per person in the group)</u>**. Reflect on the implementation of stories you chose for your MVP (those you chose to build in HW4) from an engineering perspective. Answer the following questions: **a)** how did your choices for the MVP affect the way you and your team built out the application technically? **b)** What pieces were most difficult to build compared to your original designs and intentions? **(c)** What tools/frameworks/libraries were most difficult to use and which would you choose instead in your next startup/project?

2. **Engineering Walkthrough**. Your group will need to do a code and application walkthrough with Heather, Zeeshan, and Michael. We'll create a doodle/survey to find times for the week of 3/18. We're looking for groups to ***demo these concepts*** in their application:
   a. React/(Front-End) communication (e.g. Fetch) to Express/Loopback (Back-End) API to **get** and **add/update** data for your application.
   b. React/(Front-End) communication (e.g. Fetch) to some third-party/other API to **get** data for your application.
   c. API **get** and **update** from the [Dronut-Drone API](#).
   d. **<u>At least one</u>** transaction made in your application, interacting with the [CommerceFriend](#) API.
   e. Some form of persistent state being used, e.g. database (any one of SQL, in-memory, KV) storing orders and being queried by your API.
   f. More than 3 ***useful*** UI interactions within your Front-End application, e.g. click, drag, forms, where state of the application changes (and, yes!, this is related to **<u>a</u>**/**<u>b</u>**).
   g. [Google Analytics](#) integration or some alternative gathering analytics for your site
   h. A README document in your repository explaining *how to run the application locally and via docker-compose* (e.g. development vs production variations as well), *make and deploy changes through Travis CI*, *deploy on Azure VMs*, and how to *run linting*, *tests*, and *npm-cleanup scripts/tools*.