CS411 Team Assignment 1 - Proposal

We normally don't start a project without some idea of what it is that we want to accomplish. Working with your team, come up with **two** ideas for a web application that your team will plan and build this semester. We'll evaluate your proposal during a discussion section and green-light the project, possibly with a few suggestions.

Your proposal will be submitted on Gradescope, using a link to your team's Github repository. This means that you'll need to create a repository for the team.

I've listed some of the project requirements to give you an idea of the scope. Your proposal can be relatively short; I'm looking for just an idea of what you'd like to do. If you find in a few weeks that you need to modify the project, that's fine, this is just a planning document and not chiseled into stone.

Deliverables:

1. A link to the team's GitHub repo that includes a written overview of both of your team's ideas, one paragraph per idea. (I just need one paper per team.) Please create a 'docs' folder in your repo that will hold your team's assignments.

Here are a few high-level requirements for your application:

1. **It must utilize a database**. A simple way to meet this requirement is to require a user to store profile information in the database. You'll also be using it as a cache.

2. **It must correlate at least two publicly available data sets via API from the Internet**. Examples might be weather/climate data from NOAA, crime statistics from the FBI, and so on. A great place to get started is https://apigee.com/providers?apig_cc=1, which is a repository of datasets, and Postman's list at postman.com. Another good place to search for data is http://data.gov. The City of Boston also has data available at data.cityofboston.gov. Your application must correlate these data sets in some way; for example, pull a user's playlist from Spotify and correlate it with a feed that has concert dates to alert the user of bands that they like that are playing nearby. Use of the Google Maps or Geolocation service does **not** count toward your two APIs (it's a few simple lines of code, usually).

3. **It must use third-party authentication**, for example logging in with Google or Facebook using OAuth.

4. **It must have a decoupled architecture**, similar to what we looked at in class during the 'dogfooding' lecture. The implication is that you'll need a front end and a back end, and the two will communicate via a RESTful interface. It's too early to discuss technologies, but this does mean that there will be JavaScript in the front end. Since the back end is responding to requests and just returning data, it doesn't necessarily need to be in JavaScript…Python, Java, PHP, and so on would work.

Additional constraints may be placed on the project as the semester progresses. At this point in the project you should not be thinking much of the technical aspects of the site...this part of the project is a business function. The technical design will come later.

Just think about the user benefits for now, not the technical implementation. That being said, browsing through available APIs can be a good way to spark your imagination. A Google search for 'databases with public API' will give you a nice list in addition to those listed above.

One member of your group should submit your repo link on Gradescope by the due date and time.

Some ideas to get you started

Here are some project ideas that teams have worked on in the past:

> Transfer playlists between music services such as Spotify and Apple Music.
>
> Grab a user's Twitter timeline, send it to Watson for sentiment analysis, then create a Spotify playlist to match (or change) their mood.
>
> Find the safest running routes near you based on crime statistics, and generate a playlist for your desired running time.
>
> Find the least expensive flight for a sporting event that you'd like to attend.
>
> Based on playlist histories, suggest upcoming concerts / events that you might like.
>
> Determine whether bus, T, BUS, walking, or Uber/Lyft will deliver you to your class fastest.
>
> Find the nicest bathroom near you within a radius.
>
> Use facial recognition and analysis, replace faces in a photo with emoticons depicting their current mood.
>
> Create a menu / recipe based on calorie or dietary requirements (using food-related APIs), then hit Spotify to return a playlist that matches the 'style' of dish and its cooking time.

You can, of course, choose to do anything that you like, within the technical parameters. These ideas are just to get you thinking (but it's fine if you want to do one of these, too). Also, it's easier to remove features than to add them, so don't worry if your project seems large at this point.