## Project Goal:

We want to create a web application that displays upcoming concert locations for a user’s favorite Spotify artists on an interactive map. This map will allow the user to filter by artist, and view relevant concert information by hovering over the map marker.

## Data:

The main source of data for the project will be concert locations and times for various artists. We will gather this data by using the Songkick API. If this API is not available, we will simply scrape songkick.com. We will build the application’s user interface using Django, obtain the user’s Spotify information using the Spotify API, and display the concert location map using the Google Maps API.

## Anticipated Challenges:

One challenge is that gathering concert data for a large number of artists may take a long time. We will work on optimizing this process as much as possible. In particular, there may be significant overlap between different users’ favorite artists. So, we would like to keep track of these artists in a dictionary-like object (potentially JSON) that maps artist IDs to concert locations and times. This framework presents a few issues, such as name disambiguation, and the fact that concert data changes over time.

## Rough Timeline:

* Fourth week:
  + Figure out how to get Spotify information from the user via Django.
* Fifth week:
  + Determine whether or not gaining access to the Songkick API is feasible. If so, figure out how to use it; if not, optimize scraping routine.
* Sixth week:
  + Start working with the Google Maps API and see how it interacts with Django. Try to get geocoding and basic map display working.
* Seventh week:
  + Determine a suitable data type for the database and see if it improves speed. Work on name disambiguation, start using Songkick artist IDs
* Eighth week:
  + Refine the map display, add different marker styles and information upon hovering / clicking. See how difficult it is to implement filtering system.
* Ninth / Tenth week:
  + Make sure all the pieces work well together (database, songkick, spotify, django, maps). Do substantial user testing to see where code breaks (weird artist names, issues with users already being logged in, etc.)