Group: Albert, Deren, Emma, Hayden, Giancarlo

Pitches:

#1. (Main Pitch)
Create an app/website that provides suggestions for beaches to visit based on information gathered from weather and water APIs. Using google places / maps to get the best beaches locations as well and based on that, offers ranked suggestions telling the user which/best beach to go to. Some of the planned features/data Includes: UV Index, tide, wave height, water temperature, air temperature, wind speed, sunrise/ sunset time

Requirements:

1. Have the user log-in via a created profile, which can include past beaches that he/she has gone to and maybe saved preferences (prefered wave height, etc)
2. APIs that can be utilized for this project include the Weather Channel API (https://www.programmableweb.com/api/weather-channel) and surfing forecasts APIs (https://www.programmableweb.com/api/msw-forecast)
3. Allows them to create a profile via Facebook, Twitter, etc. and the website/app can pull certain information from those profiles (location, etc).
4. This will use decoupled structure. REST getting API data as a backend, using Google Firebase as user auth and user storage, then a web app for frontend display using HTML, CSS, JS, AJAX.

#2.
Web app that takes the lyrics from the top songs from your Spotify account and curates a “mood board” based on the lyrics.

1. Users would have to login to Spotify account, have the option to export mood boards, and top songs would be stored in a database.
2. API’s we could use are Google/Bing image search API, Genius Lyrics API, Spotify API.
3. For third party authentication, pull the profile from spotify
4. This will use decoupled structure. REST getting API data as a backend, using Google Firebase as user auth and user storage, then a web app for frontend display using HTML, CSS, JS, AJAX.

Useful Links
Git Cheat Sheet

Spotify API

Interesting API's

Ideas

- Web app that takes the lyrics from the top songs in your spotify and curates a "mood board" based on the lyrics (Spotify -> Google Images)
- A Productivity/Organization tool, generates a dashboard with Google Calendar, todos list, news feed, stock market data, etc. (Similar to Notion/Monday.com/Trello)
- Browse a person's twitter account and recent posts and based on a series of recent posts, suggest a book (Twitter API + Goodreads API)
- Develop an application that allows users to enter in specific dishes they want to eat (not just the cuisine) and the application will browse through restaurants in the area for the specific dish (Restaurant menu API)
- Suggest which beaches to go to gathering tons of API data about weather and water data. Using google places / maps to get the best beaches as well. Then tell user which / best beach to go to.

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 datasets 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 APIs, and https://www.programmableweb.com (my favorite). 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.
3. It must use third-party authentication, for example logging in with Twitter 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.