**Dog Tinder - Woofer**

As a user I want to see pictures of random adoptable dogs within a defined area when I put in a location. As a user I want to be able to select or reject the images I see. As a user I want to be able to see the details of the dog whose picture I am seeing before I select or reject the dog. As a user I don’t want to see dogs I have previously rejected appear in my search results after I have rejected them. As a user I want to see a list of the dogs I have previously selected and have the list visually notify me if dogs I have selected have been adopted.

As a developer … (see MVP).

Further filters can be provided to control the random dog displayed.

**MVP**

* Access Petfinder API to get a list of dogs by location
* Track which users liked/disliked dogs. Linked tables.
* Allow the user to click an information icon to get more information about the displayed dog before rejecting or selecting.
* Allow user to select or reject the images of dogs that are generated from search results.
* Save the rejected dogs to a SQL table so that rejected dogs will not appear in future search results.
* Save the chosen dogs to a SQL table and render the dogs to the “Wooflist”
* When a user revisits the site check the API to confirm if any saved dogs are no longer available for adoption.
* Update the dogs table and display the adopted logo on the dogs list.
* Allow the user to remove selected dogs from the “Wooflist”. Save the removed dog to the rejected dogs table.
* Responsive CSS.

**Stretch Goals**

* Allow the user to create a profile of preferred dog characteristics (size, age, gender, breed) and/or allow the user to filter search criteria based on provided select menus.
* Create or find a database of breed information so that users can research the dog breed that would be best for them. Generate suggested breeds from information supplied in the user profile.
* Create a background of randomly generated dog images from Dog CEO that also generate in random locations on the page and have dynamic transitions.
* Create swipe left/swipe right functionality for the mobile view (like Tinder)

**Conflict and Communication Plan**

Availability – Availability 4/5 to 11:30 PM. – all class days.

Core functionality is only worked on as a group

No more than 2 people should be working on a feature unless we hit a roadblock.

Check slack (if possible) – between noon and 1 PM daily.

All features and tasks will be on Trello. Including features and timeline goals with delegated or chosen tasks.

Majority rules ... No more than 5 minutes to give your argument (for each side). Take a break before final discussion.

Breaks – take a break when you need a break.

Communication – meet at beginning of class day and end to review progress and any problems.

Update Trello before the end of the night.

Two people to review any merge conflicts.

Use a development branch and merge working code only to master.

So all branches pushed to development every-time you get up from your computer for more than 10 minutes.

Put your name on your branch (name-feature) and use descriptive commits.