client

Redux State

-appState, *keeps track of loading state and any messages for the user*

-user, *stores the current logged in user AND the user that is being displayed when visiting another's page*

-locations, *stores the available categories, the selected categories, the returned locations from the last query, and the 5 most recent locations*

⬇

components

navbar — this component is present on all pages

checks if the user is logged in and changes appearance accordingly

LocationCard

calls an external flag api AND external geocoding api to show the closest town to the location

RecentLocations

sends a request and recieves the 5 most recent locations, displays them as LocationCards

Friends

all users this user is following, displays them in a list with their picture minimized

Map

shows a leaflet map with all datapoints in the location

LocationPictures

takes the id of all datapoints in the location and uses those to get the url to the pictures of those datapoints, puts them in a carroussel

LocationUsers

sends a request to the server and gets all users that have this location in their favorites list

SelectedCategories

component that displays the categories the user has selected to use in the search, the available categories are rendered directly on the page

pages

- Home
- Results
- User
- Location Details
- Search

/locations

/ -get- *recieves an array of categories and returns the resulting locations*

/categories -get- *checks all available categories to choose from in the server and returns them to be displayed on the search page*

/favorites -post- *toggles a location in and out of a user's favorites list*

/:locationId/users -get- *return all users that have a specific location favorited*

/recent -get- *returns the 5 latest locations that were added*

/auth

/signup -post-
/login -post-
/me -get- *check user*

/user

/pic -patch- *adds the picture url to a user instance after one has been uploaded*

/descr -patch- *adds a description to the user instance*

/:id -get- *returns a specified user*

/add -patch- *put a user in or out of another user's followings list*

the core function that calculates locations

locationFinder.js

server

stores all users, locations and datapoints incl. their relations

remote postgres DB

database

stores all individual datapoints to calculate results with

data/

miro