Title: Weatherdle

Who:

First and Last Name	GitHub Username	Email
Adam Adl	adamadl	adad8283@colorado.edu
Aryan Singh	arsi6998	arsi6998@colorado.edu
Peyton Barger	peba1207	peba1207@colorado.edu
Marissa Mills	Mami7538	mami7538@colorado.edu
Jamie Graves	JamieGraves	jagr3886@colorado.edu
Bryson Smith	brsm3129	brsm3129@colorado.edu

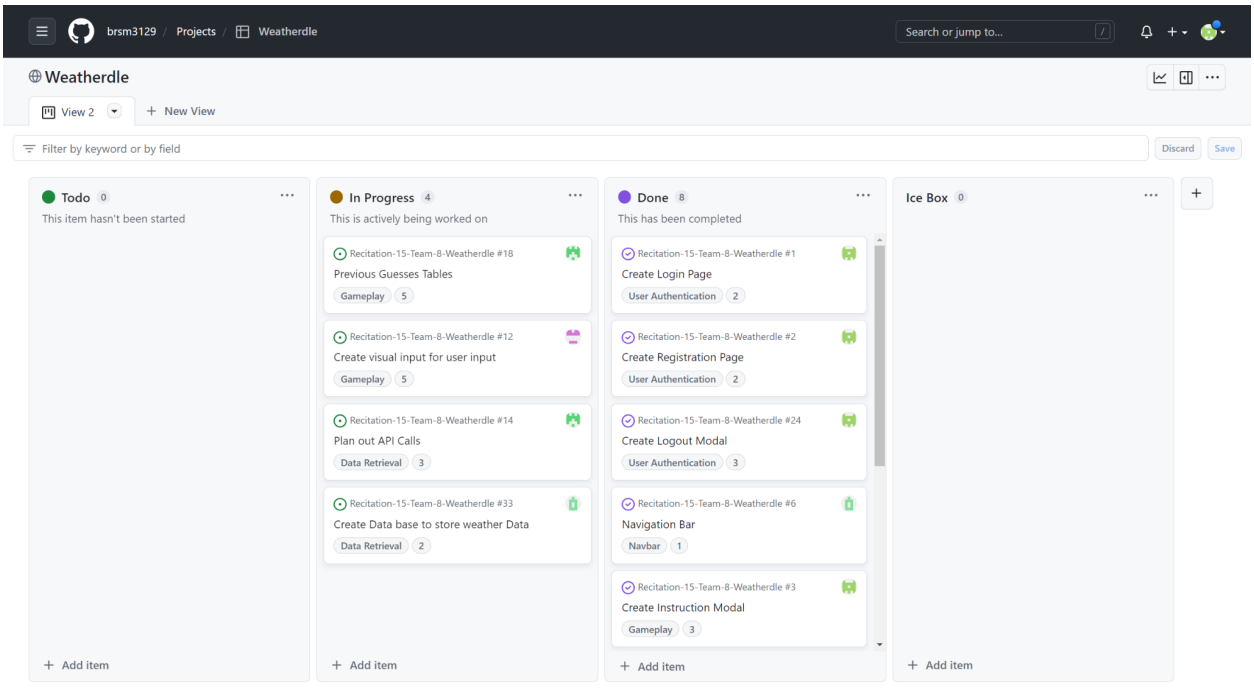
Project Description:

Upon opening the application, the user is prompted to register an account and login. This will bring the user to the home page which portrays the official game. Along with the game, there will be a button that allows the user to look at the instructions on how to play the game.

The game is called *"Weatherdle"* in which the user is given weather facts, including average temperature, average clouds, and yearly precipitation for guessing a city by weather information. Upon each guess the user gives, it is then compared to the actual answer, and the application outputs hints based on that guess, such as how high or low the average temperature is of the city they guessed compared to that of the answer.

After the user has correctly guessed the city, they are then prompted with a leaderboard showing their statistics such as the number of times played, win percentage, and current streak. They are also shown how many guesses it took them each time they played.

Project Track - GitHub Project Board:
https://github.com/users/brsm3129/projects/1



Video:
https://files.fm/u/sttpr6nv6

VCS Link to Git:
https://github.com/brsm3129/Recitation-15-Team-8-Weatherdle

Contributions:

Bryson - Directed the team on what needed to be done. Designed the architecture of the website. Built the main page of the game along with the logic behind the entries in the game table inside of the weatherdle app.get() function. Additionally I fixed issues with inserting new guesses into the table. Anytime there was something not working or needed direction, I was the one to answer the call to help.

Adam - Throughout the project, I mostly worked on the front-end aspect. I worked on the Login/Registration page and did the CSS files for that to make it look good, I also worked on the logout modal as well as the instruction modal. Additionally, I worked on the app.post() and app.get() for the Weatherdle page so that when a user makes a guess, the guess is inputted into the guesses table. I also worked a lot on the documentation of the project, such as the project board and the meeting logs.

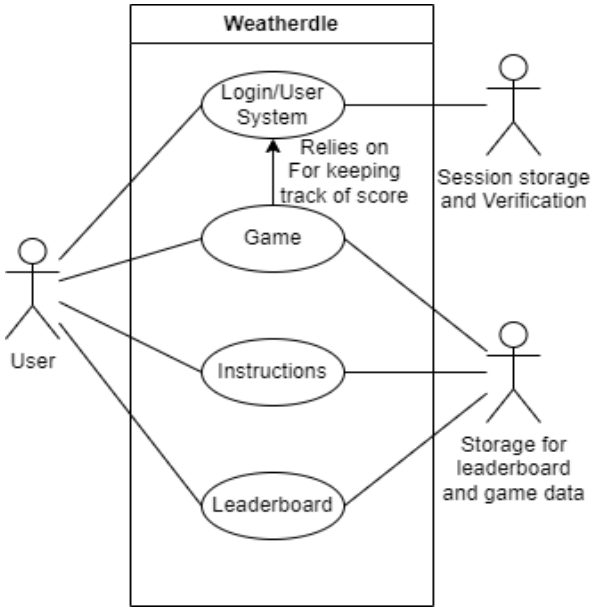
Jamie - Made user input system and found US State Capital City list with lat/longitude (sadly unneeded in end), Made semi-dynamically generated user input form in dropdown form for game.

Peyton - Made the leaderboard and profile page. Made the tables for the userdata.

Aryan- Throughout the project I made all the api calls to fetch the data from an external api, then filtered that data and populated the table with it to be used by the application. I also made the header and footer and also helped debug the app.post() in the Weatherdle page.

Marissa - Worked on session check, app.get() for the Weatherdle page, app.post() for the Weatherdle page, and completed testing.

Use Case Diagram:



Test Results:

1. ✓ Returns the default welcome message (60ms)
2. ✓ positive : /login (127ms)
3. ✓ Negative : /login. Checking invalid name
4. ✓ Negative: /register missing password
5. ✓ Positive: /register:

During lab 11, we decided to do test cases for the login and registration page. For login, we wanted to test that when a user inputs the correct username and password, they are redirected to the Weatherdle page, but when a user inputs an incorrect username and password, they receive an error message and stay on the login page. For the registration page, we wanted to test that when a user inputs any username and password that does not already exist, they get redirected to the login page, but when a user inputs a username with no password, they receive an error message and stay on the registration page. The result of these tests was successful, in that when running them, they all passed.

Deployment: **http://recitation-08-team-15.eastus.cloudapp.azure.com:3000/**
In order to run the application, clone the repository and cd into the repository/Source Code/src, then run docker-compose up. In any search engine of your choosing, run localhost:3000 to access the application