Title: AnyCast

Team Members:

- Walker Narog
- Colin Carlson
- Calvin Kim
- Layne Hunt
- Regan Wilson

Project Description:

AnyCast is a new and exciting way to explore the weather brought to you by the members of group 08 in recitation 12. Many preexisting weather apps allow the user to check the weather either at their current location or by searching for a town or locality near them. AnyCast boldly breaks from this tradition by limiting user input and reducing decision anxiety. The two novel ways of interacting with AnyCast are either a randomizer button that transports the user to some spot on the Earth's surface or by entering the latitude and longitude of their desired location. This allows the user to interact with and get a taste of weather all around the globe in places they would not necessarily think or be able to visit. To keep the user comfortable in their weather exploration, we have also implemented a selection of 5 hand crafted avatars (buffalo, iguana, macaw, ferret, and koala) that any user with an account can choose from.

Project Tracker:

https://github.com/users/wjnarog/projects/1/views/1

Video:

https://clipchamp.com/watch/qBL8NvGsEbr

VCS:

https://github.com/wjnarog/AnyCast

Contributions:

- Walker
  - I worked on the back-end, documentation and the presentation. In particular, I implemented the login, register and site navigation api endpoints and the partials for our application. I created the unit test cases for the login and register api endpoints. I managed the project tracker to make sure the team stayed focused on their designated tasks. I oversaw the Github repository and designed the documentation stored there. Additionally, I consolidated the data and designed the presentation.
- Colin
  - I primarily worked on the backend and the database structure, such that it is. Specifically, I implemented the functions to handle the generation of the random coordinates and the functions to handle the api calls to weatherapi.com, along with the passing of the weather data into the ejs pages. Additionally, I built the

framework that allows the user's avatar choice to be rendered anytime the menu partial is rendered.

- Calvin
    - I handled the bulk of the front-end and contributed to some back-end work. Once most of the back-end was completed, I focused on formatting and designing aspects. Specifically, I designed the login page, register page, navbar, home page, and formatted the avatar page. Additionally, I incorporated a fun lively animated background for the home page.

- Layne
    - I initially worked on the ejs for the home page (mainly getting the data from the weather api to display to the viewer) and polished the implementation for our randomizer button. I also handled the implementation for the geo-apify api calls to get the location data for each randomly generated set of coordinates. Lastly, I handled the front end design of the avatar page and the initial implementation for the avatar page endpoints.

- Regan
    - I definitely struggled to find what to work on throughout the project, but I did help out with setting up the user session, as well as helping out with some of the documentation. I helped work on the test cases, as well as fixing bugs in the SQL.

Use Case Diagram:

Test Results:

Feature	Acceptance Criteria	Test	Test Method	Results
Login	Nothing entered	If the user attempts to login without entering any credentials, an error message will appear asking	Automated	When the user attempted to enter no credentials the message asking them to add something to both fields.

		them to enter credentials.		
	No password entered	If the user attempts to login without entering a password, an error message will be displayed asking the user to enter a password. We can set up a test account and run this test and those following using that test account.	Automated	When the user attempted to enter only a username the message asking them to add something to the password field
	Login with invalid credentials	If the user attempts to login with invalid credentials (i.e. to an account that does not exist, or with an incorrect password to an existing account), the user will be shown an error message saying they have entered invalid credentials.	Automated	When the user attempted to enter a username and password that do not correspond to any of the existing users, the user was redirected to the registration page.
	login with valid credentials	If the user attempts to login to an existing account with the correct credentials, they	Automated	When the user entered credentials corresponding to an existing account, the user was allowed to login

		should be logged in.		normally.
Registration	Missing Field	If the user attempts to register without filling out all the fields, an error message will ask them to fill in all the fields.	Automated	When the user attempted to skip any of the three fields on the registration page, an error message asked them to fill the remaining fields.
	Invalid Field	If the user attempts to enter an invalid input (i.e. invalid email address, password that does not conform to the standards, etc.), an error message will tell them what is incorrect and how to fix it.	Automated	When the user attempted to add an invalid email address, an error message asking them to correct the email address appeared.
	Valid Fields	If the user attempts to register with all valid fields, an account for them will be created and they will be logged in to that account.	Automated	When the user entered all the fields correctly and adhered to the restrictions (valid email, etc.) a new user account was created and the user was redirected to the login screen to log into their new account.
Randomizer Button	Pulls Location	When the user presses the button, a randomized	Manual	The user pressed the button 5 times (excluding the return to Boulder

		location should be generated and the closest weather station should be pulled from the api.		function) and each time the coordinates were pulled correctly and and correlated to a country, region, and town.
	Renders Weather Page	After the location has been pulled, the correct weather information along with the corresponding UI designs should be displayed.	Manual	After the user pressed the randomizer button, the weather data was pulled correctly and displayed on the home page.
	Recognizes Character Avatar	When the page is rendered, the user's avatar should be placed into the page properly if they are logged in, otherwise no avatar should be shown.	Automated	When the user changed their profile avatar, the changes were properly communicated to the database and rendered on each subsequent page.

Specific Use Cases Being Tested:

- Can the user log in from the home page?
- Can the user change their avatar?
- Can the user randomize the coordinates?
- Can the user enter their own desired coordinates?

General Observations:

- In our user testing, the users were mostly able to grasp how to interact with the website quickly and with little outside guidance. There was mild confusion on the general structure of the website, leading us to change the order in which pages are presented to the user, specifically changing the landing page from the randomizer page to the login page.

    - The user's reasoning for being confused on how to log in is because they already assumed they were logged in/didn't have to log in because they landed on the home page.
- Users commented that the limiting of user input that has been our underlying design philosophy for this project helped them to more easily and conveniently find the weather in some location.
- Users generally responded positively to the overall aesthetics of the website, with particular praise given to the hand-drawn avatars.

Overall, the users acted according to most of our assumptions, and we didn't make too many changes based on the feedback we received from our use case testing. The largest and most important issue we had that multiple test users struggled with was logging into the website. We solved this issue by changing out landing page to the home page

Deployment Link:

AnyCast