Environment Tracker Final Summary
Group 3: Alex Escatel, Mike Squires, Dan Alzeidan, Hamza Ali

Overview:

Environment Tracker UIC is a web application that utilizes several components including Arduinos with air quality sensors, react with javascript, google firebase, and leaflet api. These tools combined, create a real time web application that has an interactive map where users can see the data for specific sensors and get warnings if the air quality is bad. The map indicates which buildings have data available with markers. Clicking the markers allowS users to view certain air quality ratings within those buildings. Problem buildings/areas are marked with circles to get a quick idea of which areas to avoid.

Domain:

With the ever increasing usage of gasoline cars and industrial take over of the world, the air quality of where people live is becoming a concern for many people. Poor air quality can lead to health issues such as heart disease and respiratory issues. This project's aim is to accurately track the air quality so that people can avoid areas with dangerous air quality.

SE Practices/Tools

We utilized Jira to efficiently manage our work. Specifically, we chose the Kanban board option in Jira, as this allowed us to visually track our progress and prioritize tasks in real-time. The flexibility of the Kanban board also enabled us to integrate Scrum and Agile methodologies, which helped us to increase productivity and adapt to changing project requirements more effectively.

Deliverables:

For the first release we focused on making sure that the basics of the website were set up. We set up the Firebase, React, Leaflet, and started gathering Arduino parts. A basic map was displayed with mock data to test markers and popups.

The second release focused on bringing everything together. We got rid of the mock data and started pulling data collected from the Arduino's straight from the Firebase. The features we added were a navigation bar between pages and circles to indicate problem areas. We also experimented with different color schemes for the website in preparation for the third release.

The third release focused on modernizing the UI. We made the navigation bar more modern and interactable and made sure it appeared on all pages. The problem we had with white space was fixed by expanding the map and adding a background to the other pages. A skeleton page was also added for the news section we weren't able to finish.

Testing:

We focused on testing Arduino code and information pulled from the Firebase and the information added to it as well. We tested the Arduino's connection, the components that read different air quality measurements, and its connection to the Firebase. Testing for Arduino and Firebase went well, and made sure we were getting well formatted and consistent data for the front-end. As for front-end testing, most of it required user testing. The main thing we tested was making sure login worked and that the map worked properly when opening and closing pop-ups. Our testing revealed our current issue, a user being able to bypass the login screen.

Open Issues:

The main open issue that we have is the pinpoints don't update unless the user refreshes the entire page. This is a simple fix but it wasn't important for the goal of the final release. We also had some issues with the user being able to manipulate the URL to bypass the login page which is concerning, but again not important for the final release

Waiting Room:

- Fix login page bypass
- Fix pinpoint updating feature
- Add more Arduinos (needs more funding or alternative solution)
- Search feature on map
- News page

Retrospective:

Overall, we completed exactly what we set out to do. For each release we met all of our goals and did so efficiently. If we were to change one thing, it would be to have a uniform data form for sending data from arduino to firebase to the front end. We wasted a lot of time updating forms with this issue.

Architecture

collection of arduino nodes → Google Cloud Firestore → (receive / send) → React.js web app → leaflet react