

Status of the software implementation:

Overall, we as a group are incredibly proud of the current status of our project, as we have successfully completed all the features we set out to finish over the project's span. After never partaking in a project of this scale, it was amazing to see the growth and sense of accomplishment upon completing these weekly tasks as a team. The minimum requirements that we were given are as follows:

- Our Alerting system, a key feature of our software, was implemented using Twilio. It utilizes our predictions model and SQLite database to send email and text message alerts to users, notifying them of hurricane activity based on their preferences.
- Visualizations (at least three different types): We implemented an interactive map, heatmap as well and chart visualizations for users and admins to interact with
- We have successfully created two distinct roles, user and Admin. Each role comes with a unique set of features and permissions, ensuring a tailored user experience for each user type.
- Filtering - Users can filter hurricane data by date and location.
- Predictions - Created a hurricane risk prediction model that relies on the previous data to predict upcoming hurricane data for a user-specified location.
- Dashboards - We implemented a user and admin dashboard, giving different permissions and features to the roles based on how we wanted these users to interact with the system.
- Login/Registration System - We implemented a login and registration system that utilizes basic validation.
- Database: Implemented an SQLite database that stores user and hurricane data, making our system usable and scalable.

After completing our requirements, we have 0 tests left in the backlog since we have accomplished everything we set out to do!

Reflections:

Once the requirements and overall project started getting into motion, our overall project management worked very well. The hardest part was the lack of direction offered at the beginning of the course surrounding the project, which set us back at the start. The only thing we would change about our process next time would be a better understanding of our tech stack before stepping in. Better file structure and testing would have helped us save time later on, which was an exciting learning experience through this project. Initially, the initial requirements needed to be better detailed, but the project felt much more apparent once we received clarification from our TA in early February. We did not miss or overlook any of the requirements. The only thing that needed to be added in our initial planning was direction, and it would have been good if we looked more into Flask before the coding process. As a team, we will deal with testing differently in the future. Though most of our struggles here were because we lacked understanding of Pytest and Selenium, it would have been better to fix our testing process earlier (i.e. ensuring they all passed and writing more beforehand) to make the project closing a more straightforward process. Overall, the effort to complete the project was about what was expected; we did feel that being expected to put in 8 hours a week consistently was unfair since everyone has other things going on in their lives, but overall with finishing on time, I feel that we correctly understood the amount of work that needed to be put into the project. Our team is proud of the many features that we implemented, especially getting everything connected into one web app. We are most proud of our prediction model/alerting system. This is because these are both things

that we knew very little about before starting the project, so it was interesting to learn how to do this and then see it all in action.