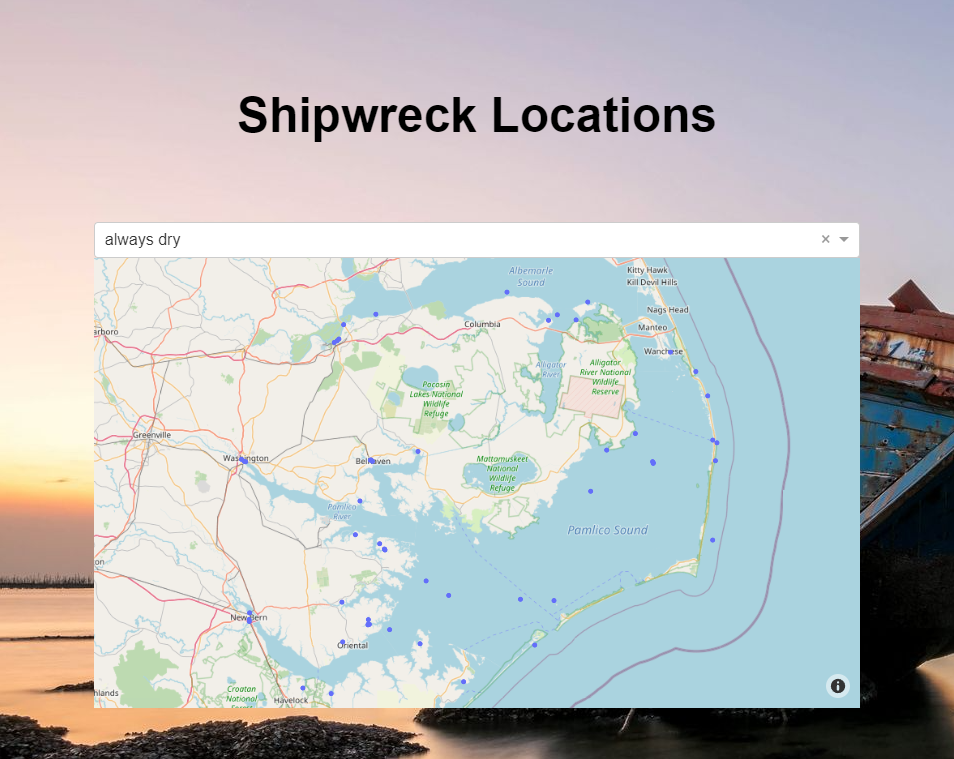
Aaron J Walls

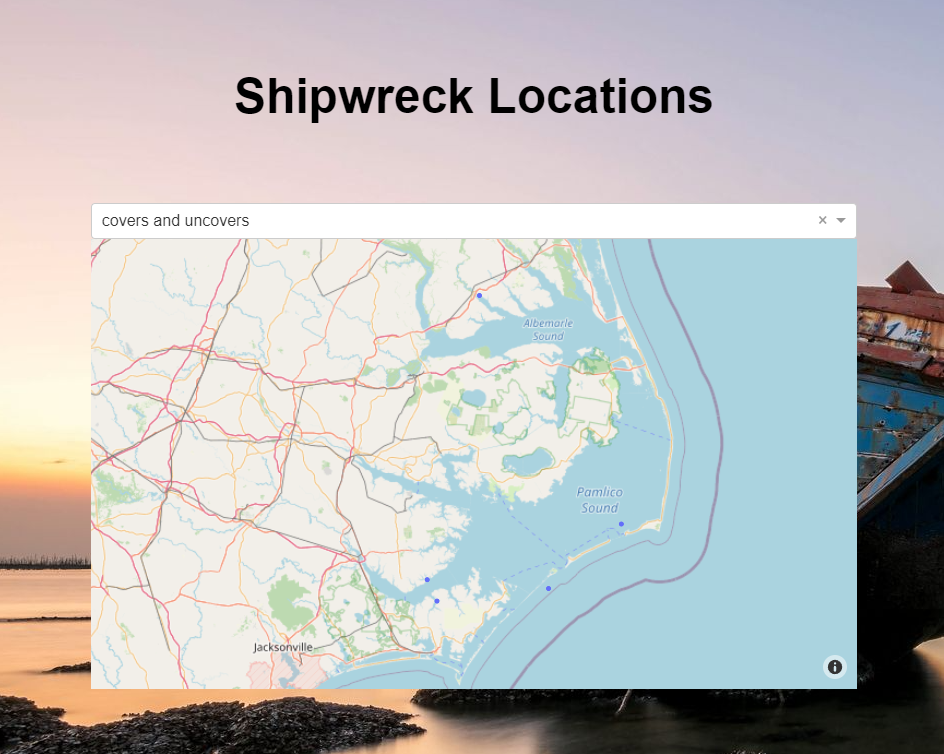
Professor Brooke Goggin

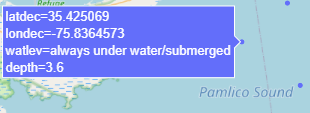
CS 499

November 28, 2021

MS4: Narrative

The “Shipwreck Locations” application is a python application that retrieves data from a Mongo Database. It uses Plotly, Dash, and Mapbox to display where the shipwrecks are according to the latitude and longitude queried. Users can filter which shipwrecks are displayed by the “watlev” value contained in the database. Below is what the application looks like when ran:

By default, it shows the always dry shipwrecks, but from the drop-down, you can select shipwrecks that are covered and uncovered depending on the tide, shipwrecks that are awash, shipwrecks that are always submerged, or even shipwrecks that are only partially submerged during high tide. There is also an option for no filter. 

If you hover over a shipwreck, and the depth is known, it will display the depth. This application demonstrates connecting and working with a database. It also demonstrates how that data can be handled appropriately and represented graphically using Dash and Plotly. Lastly, it also updates based on user input using app callbacks so that the data can be further filtered.

I had to adapt to the planned enhancements of my original project. This project is completely new from the original attempt. To get started I set up a local instance of MongoDB and imported sample data and configured SCRUM authentication. The MongoCRUD class used to query the database is nearly identical to the AnimalShelter class in the original assignment and they perform the same function: parameterize queries. This class would be used to establish a connection to my database, but also do any reading, writing, creating, or deleting I would need to do. The body of the application is similar as well, but with respect to the content. The original project had a bug where the map would not populate until a query was performed. This attempt solves that issue.



This block of code creates the map that is displayed in the application. It takes ‘watlev’ which value is one of the options from the drop-down menu. It then uses ‘watlev’ to filter the data frame. The code is not perfect. I am not the best at python and I am sure there exists a shortcut or shorthand way to condense the if statements. I am much more suited to strongly typed languages, I think this block of code reflects that. Both these projects were difficult because of python. Python is powerful (especially with all of its libraries) but it takes practice. Most of my time spent doing this project was in the documentation getting familiar with the libraries that Python has to offer. I also spent a fair amount of time setting up the environment to get all the packages to work with one another. I was able to improve my understanding of Dash and data presentation with this second attempt.