Nikki Jackson-Hebson

GEOG 777: Project 2 Report

***Problem Statement*** – Glacier Bay National Park has requested the development of a mobile-friendly interactive map application to enrich the visitor experience to Glacier Bay. The map application will depict features of interest within Glacier Bay National Park to include: name and location of glaciers, height and location of mountains, favorable wildlife sighting locations – to include sightings contributed by visitors, a marker for Glacier Bay’s Visitor Center, nearby accommodations, historical sites, and an easily accessible link to the official website for Glacier Bay National Park.

***Implementation Plan*** - The application will be written in HTML, CSS, and JavaScript to include a variety of map rendering libraries, and the use of CARTO to act as the database server. Geospatial data will be collected through open source research and imported into ArcGIS to manage and reduce excess data. Edited shapefiles will be imported into CARTO and symbolized within CARTO Builder to create a prototype of the intended map application. Before writing any text, proposed widgets for the web application are first built within CARTO Builder and documentation for JavaScript libraries are to be reviewed prior. ATOM, a desktop application text editor, can be used to host, edit, and preview the HTML project. The web application should incorporate the following map rendering libraries which are compatible with each other: CARTO-VL, MAPBOX GL JS, and AIRSHIP. CARTO-VL is a JavaScript library that interacts with different CARTO APIs to build custom applications leveraging vector rendering. MAPBOX GL JS is a JavaScript library that uses WebGL to render interactive maps from vector tiles and Mapbox styles. Lastly, AIRSHIP is a set of tools designed to enhance applications by offering layouts, templates, panels, and widgets. To note - AIRSHIP only works alongside CARTO.js or CARTO-VL. The completed application will function in both web and mobile environments and represent an aesthetically clean interface and provide valuable information to users.

***Process of executing Implementation Plan***:

* Geospatial data was collected through open source research and edited/exported in ArcMap to reduce excess data.
* Created ER diagram and logical schema.
* Edited shapefiles were imported into CARTO to use as a database server.
* The imported shapefiles were then symbolized in CARTO Builder to create a prototype of the intended finished application – as CSS code can be copied from CARTO Builder.
* Project folders were then created in ATOM where editing and testing began.
* Three map rendering libraries were included in the application: CARTO-VL, MAPBOX GL JS, and AIRSHIP.
* Enabled functions to perform with application buttons and widgets.
* Tested the application in multiple environments to ensure accuracy and consistency.
* ***ER Diagram and Logical Schema*** – The following figures *(1,2)* represent the ER diagram and logical schema for the web application database.

Diagram

Description automatically generated

Figure 1: ER Diagram

***Diagram

Description automatically generated***

Figure 2: Logical Schema

***Application design*** - The design of the application is depicted in *Figure 3* below. The application was designed to be user friendly and provide users with a description of Glacier Bay National Park as well as a hyperlink to the official park website. Users have the option to interact with a variety of tools and functions found throughout the application. The Draw Feature Tool allows the user to draw within the map to create points and/or polygons of wildlife sightings they experience while visiting Glacier Bay. The Wildlife View Tool allows users to toggle known wildlife areas by species of bear, bird, and seal. The Identify Glacier Tool identifies glacier polygons by name and location with the use of a hover function. An interactive histogram of mountain height in meters is included within the map face and updates automatically when a user zooms or pans. In addition, users are also able to interact with popups of a visitor center, several accommodations, and historical sites. Lastly, the MAPBOX navigation-guidance-day basemap was selected to show terrain and prominent land features.

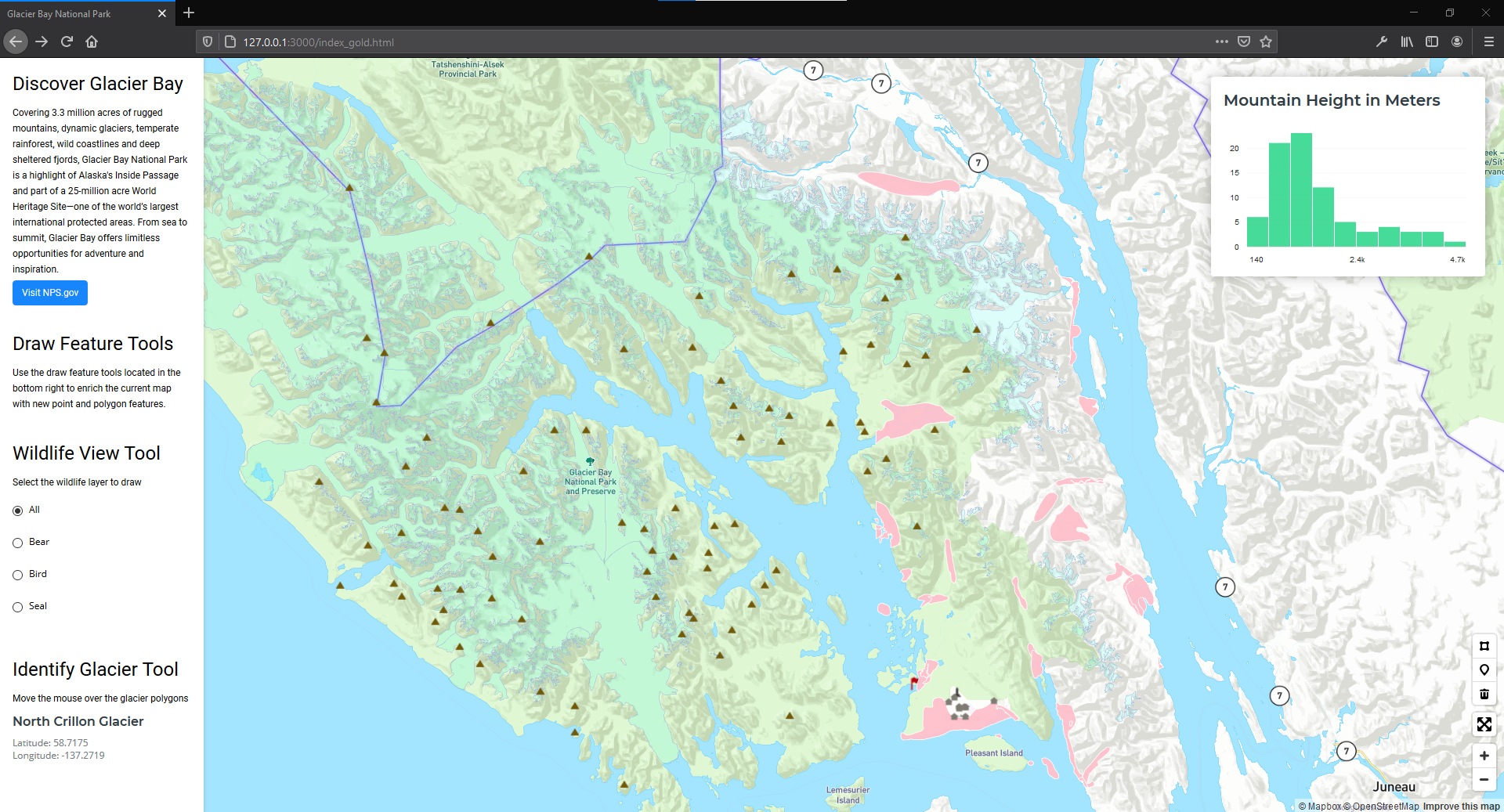


Figure 3: Design of Glacier Bay National Park interactive map application

***Functionality*** - The functionality of the application works as intended. Each button and widget within the web map application executes the intended function as defined. The user of the application is kept informed with prompts and alerts while adding and editing wildlife sightings in the map. Prompts for user input is depicted in *Figure 4* below.



Figure 4: Prompt for user input from executing draw tool processes in application

***User interface and experience*** – The interface was designed for the user to have a seamless experience when accessing and executing the tools within the application. The purpose of the application is stated in the panel as “Discover Glacier Bay” as well as a set of definitions provided to explain each tool the user can interact with. If a user inputs a wildlife sighting, they are then directed to add a species value within the prompt text box. If the user does not enter a value or cancels the prompt, the tool alerts the user with a “User cancelled the prompt” message and deletes the previously drawn feature. Thus, preventing any errors when adding wildlife sightings to the database. In addition, users are also able seamlessly interact with click and hover popups of historical sites *(Figure 5)*, a visitor center *(Figure 6)*, and several accommodations throughout the application. Furthermore, the design of the application was intended to be aesthetically pleasing with the selection of color hues as well as specifically placed buttons, text, and centering the area of interest within the application.

******

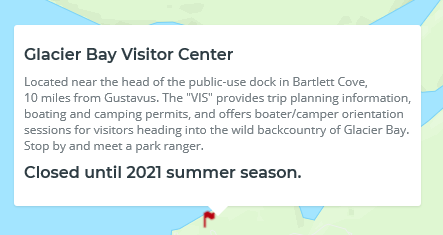
******

Figure 5: Hover popup of historical site

Figure 6: Hover popup of visitor center

***Mobile Interface*** – The application was created with responsive design methods to allow users to view the web map in multiple environments. The mobile interface is attributed with two tabs: Map and Tools. The map tab displays the main map and the mountain height histogram as shown in *Figure 7* while the Tools tab displays the interactive tools found in the panel. The Tools tab is depicted in *Figure 8* below.

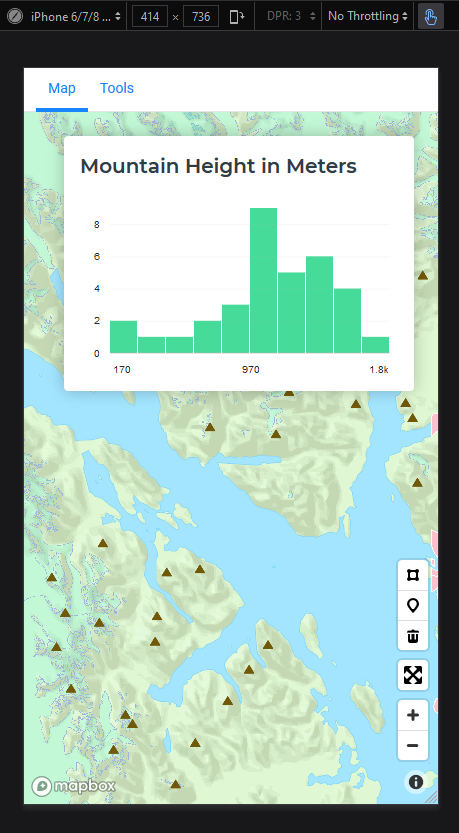
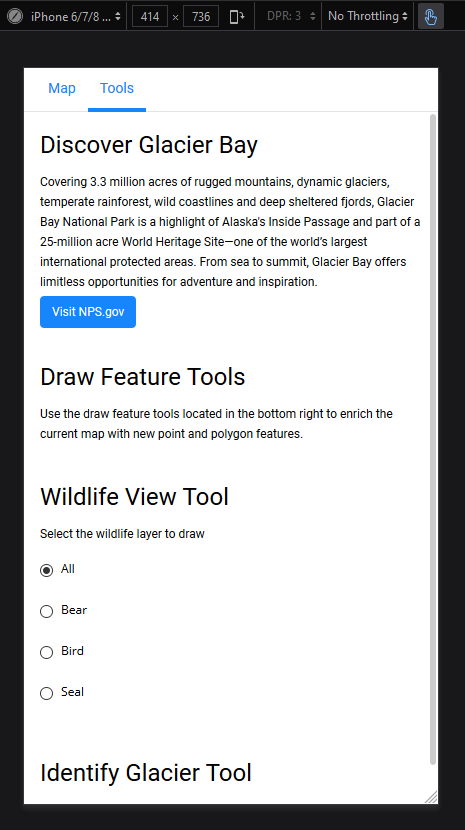
******

Figure 8: Mobile version displaying Tools tab

Figure 7: Mobile version displaying Map tab