IT390 - Skylar Zhao

Culminating Project

For my culminating project, I decided to recreate a project I made for UXD 225, this time in React with MUI elements.

Summary

Here's the original site:
https://studentweb.cdm.depaul.edu/~yzhao92/uxd225/mapux/index.html

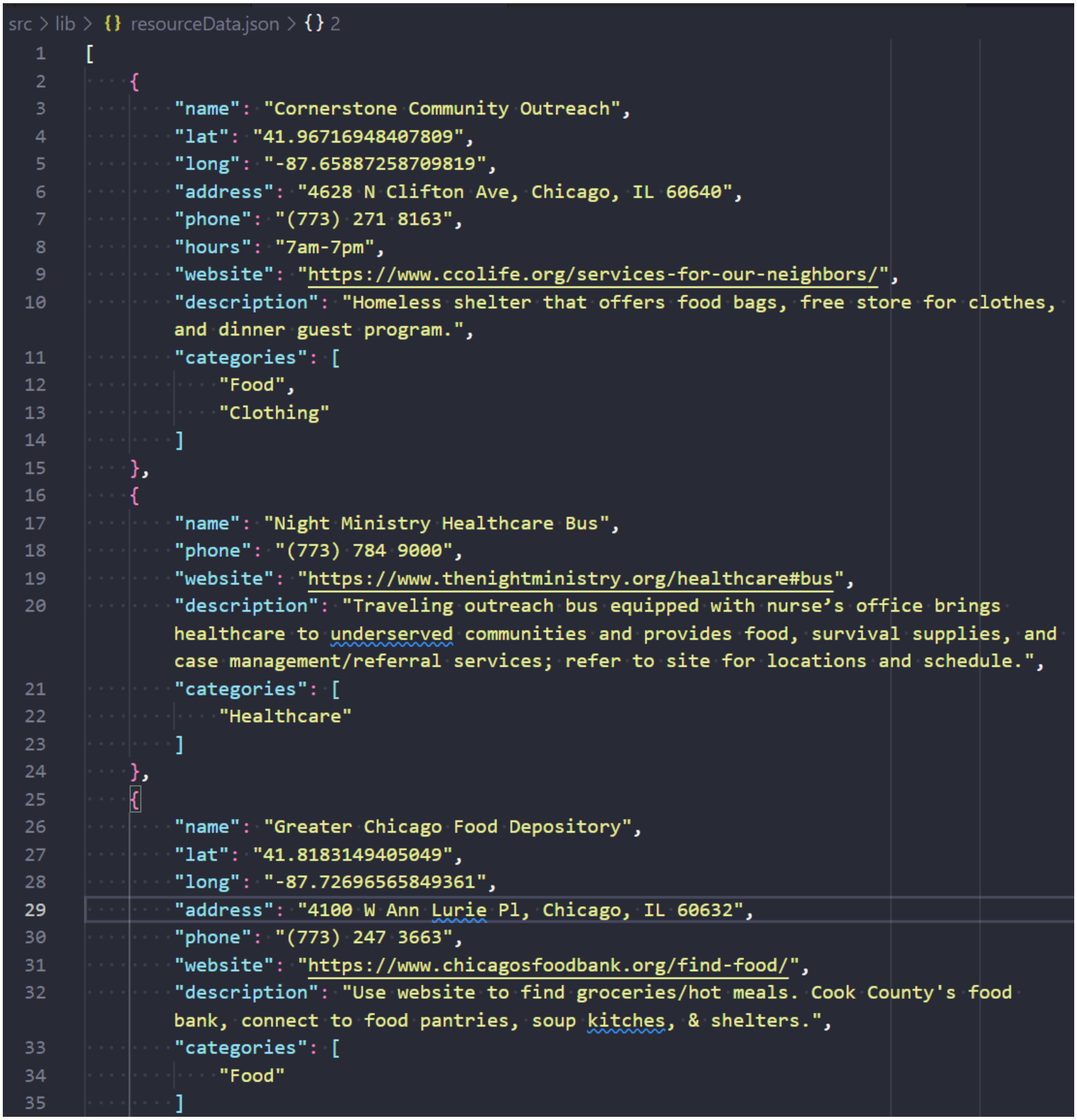
The original project was to practice using Leaflet API in a basic website. I decided to use the opportunity to visually show some data I already had lying around from a job at a nonprofit I had. I decided to take a list of shelters, food banks, mental health facilities, and other resources around Chicago for people in need.

The application shows resources in Chicago that you can filter out based on multiple categories that you can check, and shows the results in both a list and as markers on a map. The original site was written in vanilla JavaScript with direct DOM manipulation.

For this project, I fully rewrote the site in React using Material UI components.

Documentation

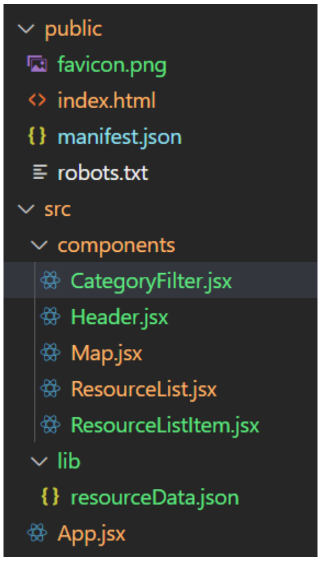
Here’s the resourceData,json file, which is the most important information I reused from the original site. It contains information for each resource, and looks like this:



```
[
    {
        "name": "Cornerstone Community Outreach",
        "lat": "41.96716948407809",
        "long": "-87.65887258709819",
        "address": "4628 N Clifton Ave, Chicago, IL 60640",
        "phone": "(773) 271 8163",
        "hours": "7am-7pm",
        "website": "https://www.ccolife.org/services-for-our-neighbors/",
        "description": "Homeless shelter that offers food bags, free store for clothes, and dinner guest program.",
        "categories": [
            "Food",
            "Clothing"
        ]
    },
    {
        "name": "Night Ministry Healthcare Bus",
        "phone": "(773) 784 9000",
        "website": "https://www.thenightministry.org/healthcare#bus",
        "description": "Traveling outreach bus equipped with nurse’s office brings healthcare to underserved communities and provides food, survival supplies, and case management/referral services; refer to site for locations and schedule.",
        "categories": [
            "Healthcare"
        ]
    },
    {
        "name": "Greater Chicago Food Depository",
        "lat": "41.8183149405049",
        "long": "-87.72696565849361",
        "address": "4100 W Ann Lurie Pl, Chicago, IL 60632",
        "phone": "(773) 247 3663",
        "website": "https://www.chicagosfoodbank.org/find-food/",
        "description": "Use website to find groceries/hot meals. Cook County's food bank, connect to food pantries, soup kitches, & shelters.",
        "categories": [
            "Food"
        ]
```

General Structure

- First, I created the general structure of my project.
- I wanted each component to be in its own file for easier development and reuse if I ever decide to revisit this project. Each component is a separate, distinct section of the page.
- The Header, CategoryFiler, ResourceList, and Map components are used in App. The resourceData.json file is loaded into App as well.



```
public
  favicon.png
  index.html
  manifest.json
  robots.txt
src
  components
    CategoryFilter.jsx
    Header.jsx
    Map.jsx
    ResourceList.jsx
    ResourceListItem.jsx
  lib
    resourceData.json
  App.jsx
```

Components

- Header
  - Written using the AppBar component in MUI
- CategoryFilter
  - I need to get all the categories that could be present in any resource in ResourceData. In App, I wrote getCategories to do just that:
    - It iterates over every resource and every category for said resource (as one resource can have multiple categories)
    - If the “allCategories” array does not include that category, then it appends it into the array.
    - Finally, “getCategories()” is assigned to a variable, categories, and used as a prop for ResourceData
  - I used the FormControlLabel component in MUI to generate checkboxes with labels:
    - https://mui.com/material-ui/react-checkbox/#label
  - Now, for the filtering logic, the function “onClickCheckboxHandler()”:
    - I used https://codingbeautydev.com/blog/react-check-if-checkbox-is-checked/ https://reactjs.org/docs/handling-events.html as a resource to learn how to see if a checkbox is checked and to handle that event.
    - The purpose of this function is to handle a checkbox click event and update the selected categories based on the state of the checkbox.
    - When the user clicks the checkbox:

      - If the checkbox becomes checked, it adds the name of the category (obtained from the "name" attribute of the checkbox element) to an array called "selectedCategories", and uses the concat method to generate a new array with the selected categories so as to not mutate the original array.
      - If the checkbox becomes unchecked, the function removes the name of the category from the "selectedCategories" array. The function creates a new array called "newCategories" using the "filter" method on the "selectedCategories" array. The "filter" method returns a new array that includes only the elements that pass the test specified by the callback function, which in this case is to exclude the category that has the same name as the unchecked checkbox.
    - I then put the function as an onChange property for the generated checkboxes, so that it is called every time the checkbox is checked or unchecked.
- ResourceList
  - It takes each resource as a prop and maps it to a ResourceListItem. It keeps the separation between components neat and tidy.
- ResourceListItem
  - It take a resource as a prop and renders it with MUI elements
  - You can have more than one element in the secondary property for the ListItemText component: https://stackoverflow.com/a/52402350
- Map
  - The original site's map is written in vanilla Leaflet; for this project, I used React-Leaflet
  - https://react-leaflet.js.org/docs/start-installation/ and https://react-leaflet.js.org/docs/example-popup-marker/ covered basic installation of the map and implementation of markers
  - I made a marker variable filtering resources by keeping those that have both lat and long properties, as some resources do not have a physical location or do not publish their location.
  - I then mapped out these resources to markers on the map that shows a popup with their name and address when clicked.

- App
  - The application itself. The topmost component that represents the site.
  - Aside from the aforementioned “getCategories” method and the implementation of the previous components, I also wrote some code here to filter out the list of resources by the selected category.
  - I originally thought the filtering should be in the ResourceList component, but it is more convenient to lift the state to App to handle both the state of the checked categories in CategoryFilter and filter the resources to pass to ResourceListItem and Map for them to display the filtered results.
    - I first create an empty array, and filter out the resource based on if the resource does not include the category, so that the relevant resources are included.
    - When multiple categories are checked, the resources shown must match all the checked categories instead of just having one of the categories.
    - If no checkboxes are selected, then all resources are shown.
  - The “selectedCategories” state and its associated setter callback are passed as props in the CategoryFilter component.

Other Comments:

- I used the Box component extensively for this site, and the sx property. Both are for layout and styling.
- https://mui.com/system/react-box/
- https://mui.com/system/getting-started/the-sx-prop/