Project Report

Theme: Using Firebase to Manage Starbucks Dataset

1. Project Idea

Now, many people want to know: What city or country has the highest number of Starbucks stores per capita? What two Starbucks locations are the closest to one another? What location on Earth is farthest from a Starbucks? How has Starbucks expanded overseas?

So, in order to let people to get this information, I get the dataset from Kaggle and create a Web application. “Starbucks Locations Worldwide” dataset includes a record for every Starbucks or subsidiary store. It contains 256,000 data and each data contains 13 attributes, and it is CSV format and website presents the dataset in a tabular format.

In this web application user can sort the data up and down according to different attributes; in addition, they also can filter the data through search bar and range slider according their own needs. These functions will help them easier to get information about Starbucks.

2. Working component screenshot and implement details

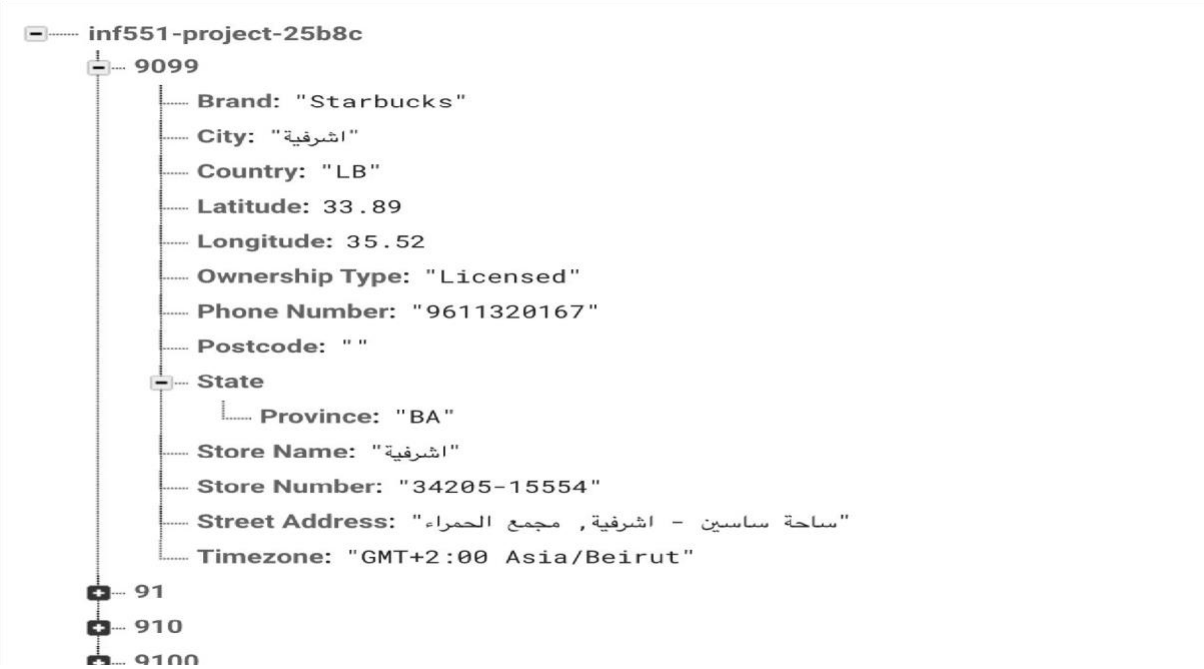
1) Data researching and selecting

In this project, I choose an interesting data set ,“Starbucks Locations Worldwide”, from Kaggle.com . The dataset is in CSV format and Website presents the dataset in a tabular format, which includes a record for every Starbucks or subsidiary store location currently in operation as of February 2017. In this dataset, there are many attributes of each Starbucks, such as store name, ownership type, location and other attributes.

2) Data importing

First, I download the dataset from Kaggle and convert the CSV file into JSON format. Then, learning the data and edit the data to create index for this dataset. Finally, upload the dataset into Firebase using Python ‘request’ package. The following two screenshots are Python script and the Firebase result:

```
#read file
file = sys.argv[1]
df = pd.read_csv(file)

#remove the duplicated ones
ids = df["Store Number"]
d = df[ids.isin(ids[ids.duplicated()])].sort_values('Store Number')
df = df.drop(df.index[8028])
#df = df['store Number'].replace('-', ' ')
#df = df.set_index('Store Number')
df = df.sort_values('Store Number')

df.columns = ['Brand', 'Store Number','Store Name', '"Ownership
    Type', 'Street Address', 'City', 'StatesOrProvince', 'Country',
    'Postcode', 'Phone Number','Timezone', 'Lonitude', 'Latitude']

# into JSON file
jf = df.to_json('out.json', orient='index')
j = open('out.json','r')
rests = j.read()
print(rests)
#replace

# connect to firebase and load data
url ='https://starbucks-9101c.firebaseio.com/storeNumber.json'
response = requests.get(url)
data = rests
response = requests.put(url,data)
```



3) App designing

A. Filter function

The central concept of filter function is that when users enter keywords in the search bar under a certain attribute, the system will automatically loop each data of this column and match

with the input keyword, so as to output the data that the user wants. In addition, users can swipe range sliders of Latitude and Longitude to select range of values, thereby filtering out the data that meets the condition.

I use jQuery package to implement the range-slider filter function, and write JavaScript to implement search bar filter function. The following screenshots how I implement the two filter functions:

```
<!--filtering-->
<script type="text/javascript">
    function onSearch(obj, n) {
        setTimeout(function () {
            let DataTableId = document.getElementById('DataTable');
            let rowsLength = DataTableId.rows.length;
            let key = obj.value;
            let searchCol = n;
            for (let i = 1; rowsLength > i; i++) {
                let searchText = DataTableId.rows[i].cells[searchCol].innerHTML;
                if (searchText.match(key)) {
                    DataTableId.rows[i].style.display = '';
                } else {
                    DataTableId.rows[i].style.display = 'none';
                }
            }
        });
    }
</script>
```

Brand	City	Country	Latitude	Longitude	Ownership Type	Phone Number
Starbucks	Abu		24.48	54.38	Licensed	
Starbucks	Abu		24.51	54.54	Licensed	
Starbucks	Abu		24.4	54.49	Licensed	
Starbucks	Abu Dhabi	AE	24.4	54.49	Licensed	
Starbucks	Abu Dhabi	AE	24.46	54.61	Licensed	
Starbucks	Abu Dhabi	AE	24.19	55.69	Licensed	26670052
Starbucks	Abu Dhabi	AE	24.48	54.38	Licensed	
Starbucks	Abu Dhabi	AE	24.47	54.34	Licensed	
Starbucks	Abu Dhabi	AE	24.49	54.37	Licensed	26426280
Starbucks	Abu Dhabi	AE	24.19	55.69	Licensed	26359275

Sort
Sort ascending
Sort descending
Filter
Abu Dhabi

```
<!--    rangeSlider-->
<script type="text/javascript">
    $(document).ready(function () {
        $("#slider-range-latitude").slider({
            range: true,
            min: -46.4,
            max: 64.8,
            values: [-46.4, 64.8],
            slide: function (event, ui) {
                // in this function we can define what happens when a user changes the sliders
                $("#amount").val(ui.values[0] + " - " + ui.values[1]);

                let table = document.getElementById("DataTable");
                for (let i = 1, row; row = table.rows[i]; i++) {
                    let latitude = $('tr:eq(' + i + ') > td:eq(3)').text();
                    if (latitude >= ui.values[0] && latitude <= ui.values[1]) {
                        $(row).show();
                    } else {
                        $(row).hide();
                    }
                }
                $("#amount-latitude").val($("#slider-range-latitude").slider("values", 0) +
                    " - " + $("#slider-range-latitude").slider("values", 1));
            }
        });
        $("#amount-latitude").val($("#slider-range-latitude").slider("values", 0) +
            " - " + $("#slider-range-latitude").slider("values", 1));
```

Brand	City	Country	Latitude	Longitude	Ownership Type
Starbucks	Ras Al Khaimah	AE	25.79		Licensed
Starbucks	Ras Al Khaimah	AE	25.75		Licensed
Starbucks	Ras Al Khaimah	AE	25.68		Licensed
Starbucks	Umm Al Quwain	AE	25.53	55.54	Licensed
Starbucks	Ajman	AE	25.42	55.47	Licensed
Starbucks	Ajman	AE	25.39	55.47	Licensed
Starbucks	Sharjah	AE	25.35	55.39	Licensed
Starbucks	Sharjah	AE	25.35	55.4	Licensed
Starbucks	Sharjah	AE	25.33	55.38	Licensed

Sort
Sort ascending
Sort descending
Filter range:
6.6 - 31.6

B. Sorting function

Users can use “Sorting ascending” and “Sort descending” buttons to sort data. The system will automatically loop through each row of data, compare and then sort them, so as to output the data that the user wants.

I use write JavaScript to implement this sorting function, and use Bootstrap framework to set drop-down menu, button icon, and table roller. The following screenshots show how to implement sorting function:(e.g. Latitude)

```
<!--sorting-->
<script>
    function sortAscending(n) {
        let table, rows, switching, i, x, y, shouldSwitch, dir, switchcount = 0;
        table = document.getElementById("DataTable");
        switching = true;
        dir = "asc";
        while (switching) {
            switching = false;
            rows = table.rows;
            for (i = 1; i < (rows.length - 1); i++) {
                shouldSwitch = false;
                x = rows[i].getElementsByTagName("TD")[n];
                y = rows[i + 1].getElementsByTagName("TD")[n];
                if (x.innerHTML.toLowerCase() > y.innerHTML.toLowerCase()) {
                    shouldSwitch = true;
                    break;
                }
            }
            if (shouldSwitch) {
                rows[i].parentNode.insertBefore(rows[i + 1], rows[i]);
                switching = true;
                switchcount++;
            } else {
                if (switchcount == 0 && dir == "asc") {
                    dir = "desc";
                    switching = true;
                }
            }
        }
    }
```

```
function sortDescending(n) {
    let table, rows, switching, i, x, y, shouldSwitch, dir, switchcount = 0;
    table = document.getElementById("DataTable");
    switching = true;
    dir = "desc";
    while (switching) {
        switching = false;
        rows = table.rows;
        for (i = 1; i < (rows.length - 1); i++) {
            shouldSwitch = false;
            x = rows[i].getElementsByTagName("TD")[n];
            y = rows[i + 1].getElementsByTagName("TD")[n];
            if (x.innerHTML.toLowerCase() < y.innerHTML.toLowerCase()) {
                shouldSwitch = true;
                break;
            }
        }
        if (shouldSwitch) {
            rows[i].parentNode.insertBefore(rows[i + 1], rows[i]);
            switching = true;
            switchcount++;
        } else {
            if (switchcount == 0 && dir == "asc") {
                dir = "asc";
                switching = true;
            }
        }
    }
}
```

Brand	City	Country	Latitude	Longitude	Ownership Type
Starbucks	Wien	AT	48.24		Company Owned
Starbucks	Wien	AT	48.24		Company Owned
Starbucks	Wien	AT	48.22		Company Owned
Starbucks	Wien	AT	48.21		Company Owned
Starbucks	Wien	AT	48.21	16.37	Company Owned
Starbucks	Wien	AT	48.21	16.38	Company Owned

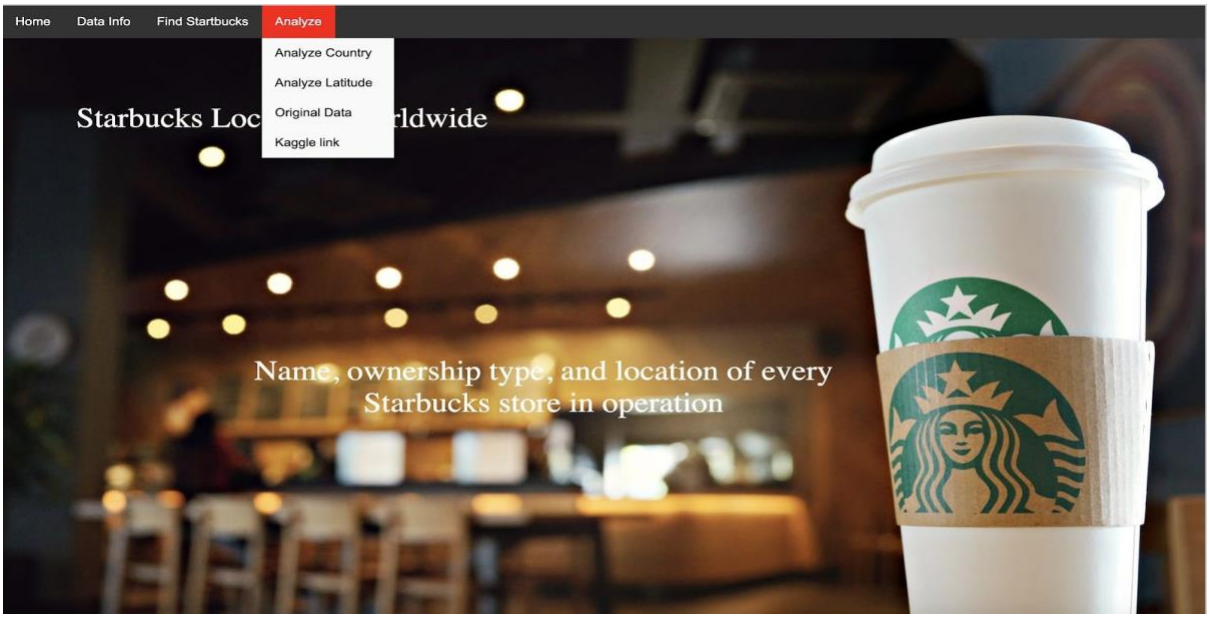
4) Web development and demo

I use Bootstrap framework and JavaScript to modify the web page, and use CSS to set the style of this Web application.

A. Home Page

This is the main page and lists navigation menu and drop-down menu. Users can click the corresponding menu to enter the corresponding interface to obtain information. The screenshots are HTML home page design and the result:

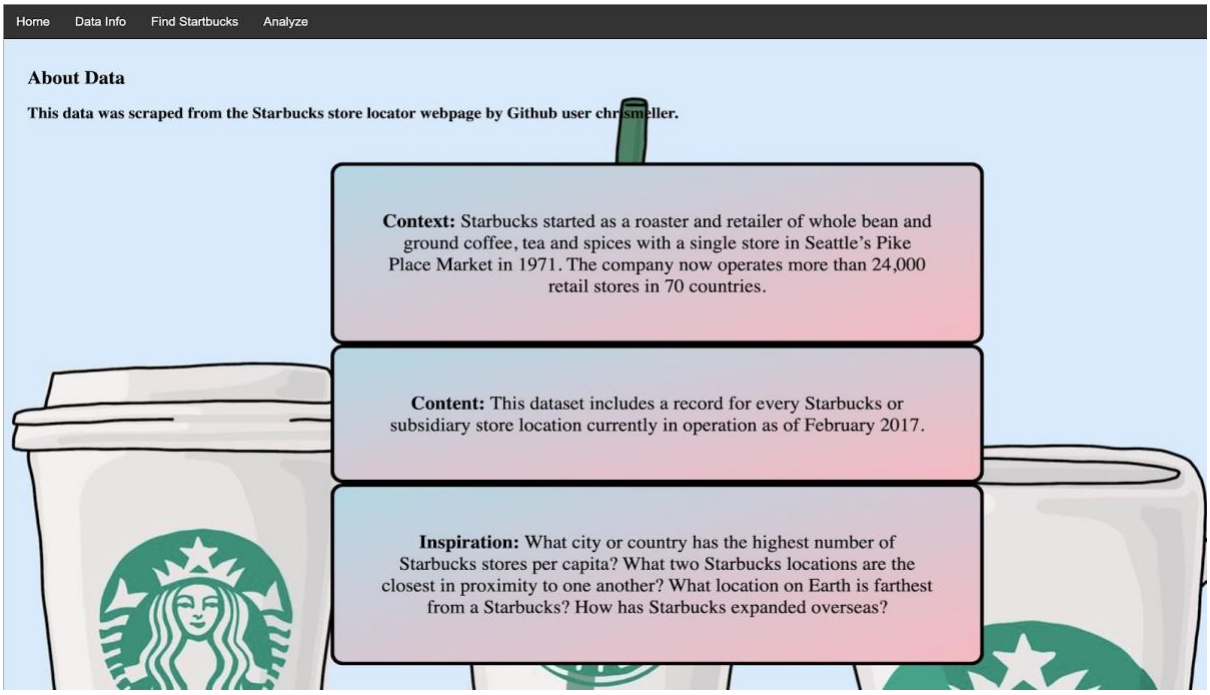
```
<body>
    <div class="navbar">
      <a href="home.html">Home</a>
      <a href="aboutData.html">Data Info</a>
      <a href="table.html">Find Startbucks</a>

      <div class="dropdown">
        <button class="dropbtn">Analyze
          <i class="fa fa-caret-down"></i>
        </button>
        <div class="dc">
          <a href="analyze.html">Analyze Country</a>
          <a href="analyze2.html">Analyze Latitude</a>
          <a href="test.html">Original Data</a>
          <a
              href="https://www.kaggle
              .com/starbucks/store-locations">Kaggle
              link</a>
        </div>
      </div>
    </div>

    <div class="container">
      <img src="1.jpg" alt="Snow" style="width:100%;">
      <div class="top-left">Starbucks Locations
          Worldwide</div>
      <div class="centered">Name, ownership type, and
          location of every Starbucks store in
          operation</div>
    </div>
</body>
```



B. Data Information Page

This page I show the basic information about Starbucks data. The screenshots are HTML data information page design and the result:

```
<div class="container">
  <img src="2.jpg" alt="Snow"
      style="width:100%;height:100%">
  <div class="top-left"><h1>About Data</h1><h3>This data
      was scraped from the Starbucks store locator
      webpage by Github user chrismeller.</h3></div>
  <div class="centered"><div>

  <p> <b>Context: </b>Starbucks started as a roaster and
      retailer of whole bean and ground coffee, tea and
      spices with a single store in Seattle's Pike Place
      Market in 1971. The company now operates more than
      24,000 retail stores in 70 countries.</p>
```



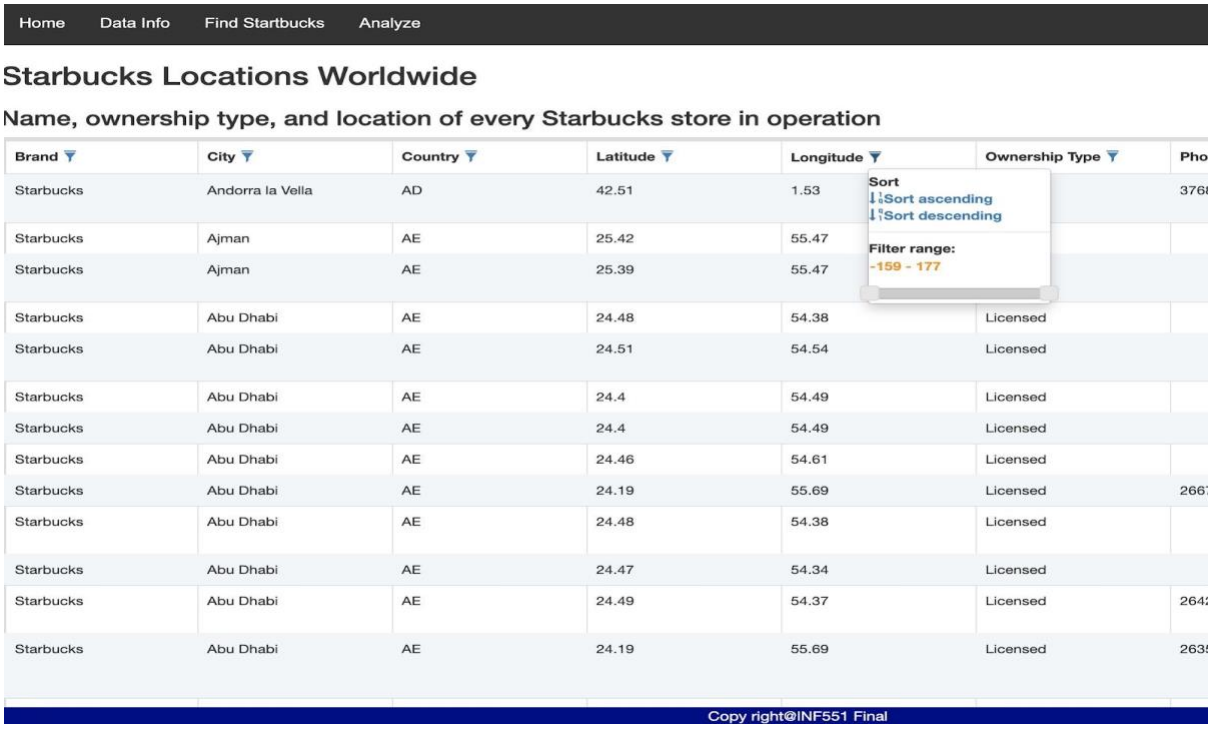
C. Dataset Page

This page shows the specific data of Starbucks. In order to implement this page, first, I create the table of HTML page, and then I connect the firebase and insert data to the table of HTML page. The following screenshots are method for creating page table using HTML, and JavaScript script to insert data from Firebase to HTML page table:

```
<div class="content-box_content-section">
    <div style="...">
        <table id="DataTable" class="table table-striped" border="1">
            <thead style="...">
            <tr>
                <div class="dropdown">

                    <th id="brand">Brand
                        <div class="btn-group">
                            <a href="#" data-toggle="dropdown">
                                <span class="glyphicon glyphicon-filter"></span>
                            </a>
                            <ul class="dropdown-menu" role="menu">
                                <div>Sort</div>
                                <div>
                                    <a href="#" onclick="sortAscending(0)">
                                        <span class="glyphicon glyphicon-sort-by-alphabet"></span>Sort ascending
                                    </a>
                                    <br>
                                    <a href="#" onclick="sortDescending(0)">
                                        <span class="glyphicon glyphicon-sort-by-alphabet-alt"></span>Sort
                                        descending
                                    </a>
                                </div>
                                <li class="divider"></li>
                                <div>Filter</div>
                                <div>
                                    <input name="key" type="text" onkeydown="onSearch(this,0)" value=""/>
                                </div>
                            </ul>
                        </div>
                    </th>
```

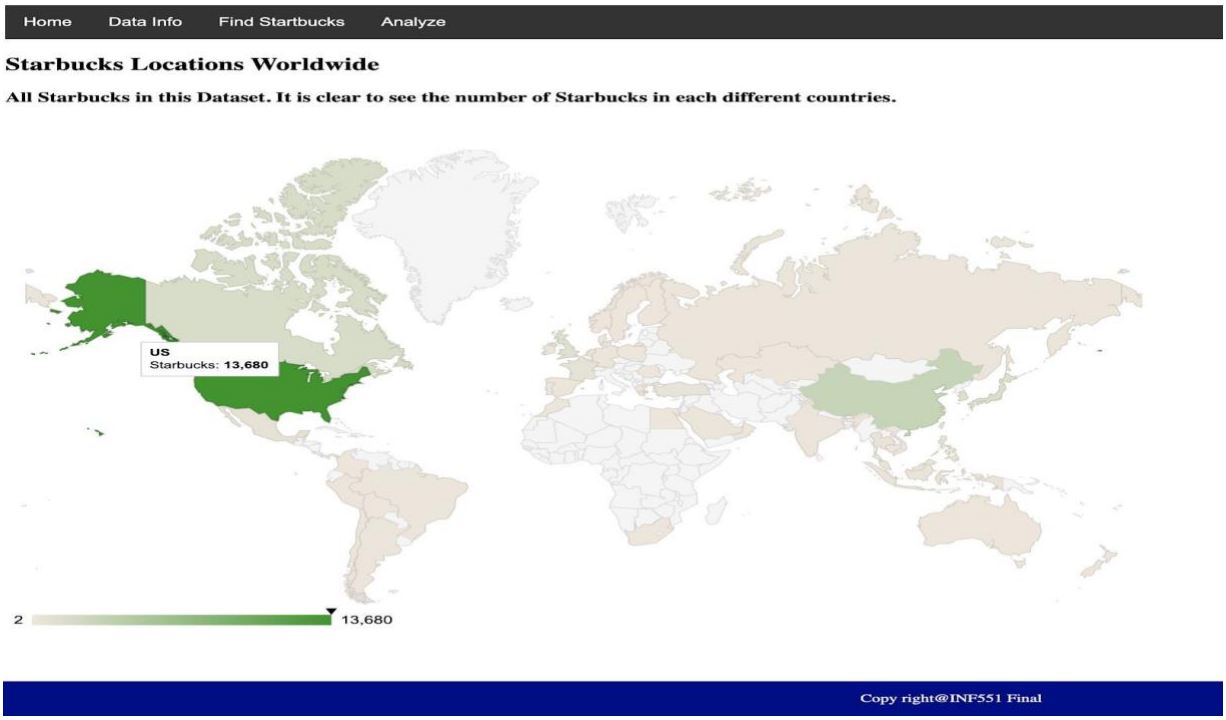
```
<script>
    // Your web app's Firebase configuration
    let firebaseConfig = {
        apiKey: "AIzaSyAGUMqpcbCNGBqTwDkFKSb5gedtcsb99CY",
        authDomain: "inf551-project-25b8c.firebaseapp.com",
        databaseURL: "https://inf551-project-25b8c.firebaseio.com",
        projectId: "inf551-project-25b8c",
        storageBucket: "inf551-project-25b8c.appspot.com",
        messagingSenderId: "764344443075",
        appId: "1:764344443075:web:86ebc82525c5eeff858276",
        measurementId: "G-7YCQJNHBPX"
    };
    // Initialize Firebase
    firebase.initializeApp(firebaseConfig);


    // https://firebase.google.com/docs/reference/js/firebase.database.DataSnapshot
    let dbRef = firebase.database().ref();
    dbRef.once('value')
        .then(function (snapshot) {
            snapshot.forEach(function (childSnapshot) {
                let brand = childSnapshot.val()['Brand'];
                let city = childSnapshot.val()['City'];
                let country = childSnapshot.val()['Country'];
                let latitude = childSnapshot.val()['Latitude'];
                let longitude = childSnapshot.val()['Longitude'];
                let ownership_type = childSnapshot.val()['Ownership Type'];
                let phone_number = childSnapshot.val()['Phone Number'];
                let postcode = childSnapshot.val()['Postcode'];
                let state_province = childSnapshot.val()['State']['Province'];
                let store_name = childSnapshot.val()['Store Name'];
                let store_number = childSnapshot.val()['Store Number'];
                let street_address = childSnapshot.val()['Street Address'];
                let timezone = childSnapshot.val()['Timezone'];
```

```
                let table = document.getElementById('DataTable');
                let row = table.insertRow();
                row.insertCell().innerHTML = brand;
                row.insertCell().innerHTML = city;
                row.insertCell().innerHTML = country;
                row.insertCell().innerHTML = latitude;
                row.insertCell().innerHTML = longitude;
                row.insertCell().innerHTML = ownership_type;
                row.insertCell().innerHTML = phone_number;
                row.insertCell().innerHTML = postcode;
                row.insertCell().innerHTML = state_province;
                row.insertCell().innerHTML = store_name;
                row.insertCell().innerHTML = store_number;
                row.insertCell().innerHTML = street_address;
                row.insertCell().innerHTML = timezone;

            });
        });
```



D. Analyze Page

This page I use JavaScript and CSS to visualize the data of "country" using world map, and "latitude" using histogram. When I put the mouse in different locations of the map and histogram, the corresponding starbucks data will be displayed automatically.



Starbucks Locations Worldwide

Show the visualization of all Starbucks' locations (Latitude) in this Dataset.