<!DOCTYPE html>

<html lang="en">

<head>

<title>Trails Map</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no"/>

<meta charset="utf-8"/>

<!-- Leaflet CSS -->

<link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.1.0/css/font-awesome.min.css" rel="stylesheet"/>

<link rel="stylesheet" href="https://unpkg.com/leaflet@1.0.1/dist/leaflet.css"/>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/leaflet.markercluster/1.5.3/MarkerCluster.css"/>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/leaflet.markercluster/1.5.3/MarkerCluster.Default.css"/>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/leaflet-search/2.9.8/leaflet-search.min.css" />

<script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet-search/2.9.8/leaflet-search.min.js"></script>

<!-- Custom CSS -->

<link rel="stylesheet" href="leaflet-sidebar.css"/>

<link rel="stylesheet" href="stylem.css"/>

<style>

body {

padding: 0;

margin: 0;

}

html, body, #map {

height: 100%;

font: 10pt "Helvetica Neue", Arial, Helvetica, sans-serif;

}

.lorem {

font-style: italic;

color: #aaa;

}

</style>

</head>

<body>

<div id="sidebar" class="sidebar collapsed">

<!-- Nav tabs -->

<div class="sidebar-tabs">

<ul role="tablist">

<li><a href="#home" role="tab"><i class="fa fa-bars"></i></a></li>

<li><a href="#settings" role="tab"><i class="fa fa-gear"></i></a></li>

</ul>

</div>

<!-- Tab panes -->

<div class="sidebar-content">

<div class="sidebar-pane" id="home">

<h1 class="sidebar-header">

<span class="sidebar-close"><i class="fa fa-caret-left"></i></span>

</h1>

<div class="search-container">

<input type="text" id="searchInput" placeholder="Search for a location...">

<button id="searchButton">Search</button>

<div id="suggestionsContainer"></div>

</div>

<div id="sidebarContent"></div>

<div class="container">

<div>

<div class="art-board\_\_containe lorem" id="cardsContainer">

<!-- Dynamically generated cards will be inserted here -->

</div>

</div>

</div>

</div>

<!--<div class="sidebar-pane" id="settings">

<h1 class="sidebar-header">

Settings<span class="sidebar-close"><i class="fa fa-caret-left"></i></span>

</h1>

</div>-->

</div>

</div>

<div id="map" class="sidebar-map"></div>

<!-- Leaflet JS -->

<script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.7.1/leaflet.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet.markercluster/1.5.3/leaflet.markercluster.js"></script>

<script src="leaflet-sidebar.js"></script>

<script src="app.js"></script>

</body>

</html>

App.js

var map = L.map("map").setView([11.1271, 78.6569], 7);

L.tileLayer("https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}/{y}?access\_token=pk.eyJ1IjoibWRhbGFtaW43MDY2ODgiLCJhIjoiY2txeW96aXF4MWFmMDJvbzFpbjBsYTQxcCJ9.mhnxoz7okQGZ9pMHMVUH0w", {

attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors, Imagery &copy; <a href="https://www.mapbox.com/">Mapbox</a>',

maxZoom: 18,

id: "mapbox/streets-v11",

tileSize: 512,

zoomOffset: -1,

}).addTo(map);

map.addEventListener("drag", (e) => console.log(map.getCenter()));

map.addEventListener("zoom", (e) => console.log(map.getCenter()));

var greenIcon = new L.Icon({

iconUrl: "loc.jpeg",

shadowUrl: "https://cdnjs.cloudflare.com/ajax/libs/leaflet/0.7.7/images/marker-shadow.png",

iconSize: [50, 50],

iconAnchor: [23, 33],

popupAnchor: [1, -33],

shadowSize: [41, 41],

});

var mcg = L.markerClusterGroup({

chunkedLoading: true,

spiderfyOnMaxZoom: false,

});

const sheetID = '1zkxjCUv4EhFl34AuN8HhNE3DazbinGkKrR17OPBlunU';

const sheetRange = 'mapdata!A2:F30';

const apiKey = 'AIzaSyAJBec7\_E9UNpNbGnFjrcCr4oS8gdVgtck';

const sheetURL = `https://sheets.googleapis.com/v4/spreadsheets/${sheetID}/values/${sheetRange}?key=${apiKey}`;

fetch(sheetURL)

.then(response => response.json())

.then(data => {

const rows = data.values;

const cardsContainer = document.getElementById('cardsContainer');

rows.forEach((row, index) => {

const coords = row[0].split(',').map(Number);

const name = row[1];

const image = row[2];

const address = row[3];

const link = row[4];

const phone = row[5];

var marker = L.marker(coords, { icon: greenIcon });

var popupContent = `

<div class="card">

<div class="card\_\_image">

<img src="${image}" alt="${name}" style="width: 100%; height: auto;" />

</div>

<div class="card\_\_info">

<h5>${name}</h5>

<p><b>Address:</b> ${address}</p>

<p><b>Phone:</b> ${phone}</p>

<a href="${link}" target="\_blank">Visit Us</a>

</div>

</div>

`;

marker.bindPopup(popupContent);

mcg.addLayer(marker);

var card = document.createElement('div');

card.className = 'card';

card.id = `card${index + 1}`;

card.innerHTML = `

<div class="card\_\_image">

<img src="${image}" alt="${name}" />

</div>

<div class="card\_\_info">

<div class="car\_\_info--title">

<p>${name}</p>

</div>

</div>

`;

cardsContainer.appendChild(card);

card.addEventListener("click", () => {

map.flyTo(coords, 18);

marker.openPopup();

sidebar.close(); // Close sidebar when a card is clicked

});

});

map.addLayer(mcg);

})

.catch(error => console.error('Error fetching data:', error));

var sidebar = L.control.sidebar('sidebar').addTo(map); // Ensure sidebar is correctly initialized

// Function to handle search suggestions

function searchSuggestions(event) {

const keyword = event.target.value.trim().toLowerCase();

const suggestionsContainer = document.getElementById('suggestionsContainer');

suggestionsContainer.innerHTML = '';

if (keyword.length >= 1) {

fetch(sheetURL)

.then(response => response.json())

.then(data => {

const rows = data.values;

const matchingRows = rows.filter(row => {

const name = row[1].toLowerCase();

return name.includes(keyword);

});

matchingRows.forEach((row, index) => {

const name = row[1];

const suggestionItem = document.createElement('div');

suggestionItem.textContent = name;

suggestionItem.classList.add('suggestion-item');

suggestionItem.addEventListener('click', () => {

// Set the selected suggestion as the search input value

document.getElementById('searchInput').value = name;

// Clear suggestions

suggestionsContainer.innerHTML = '';

});

suggestionsContainer.appendChild(suggestionItem);

});

})

.catch(error => console.error('Error fetching data:', error));

}

}

// Add an event listener for input event on the search input field to handle suggestions

document.getElementById('searchInput').addEventListener('input', searchSuggestions);

function performSearch(keyword) {

fetch(sheetURL)

.then(response => response.json())

.then(data => {

const rows = data.values;

// Find the matching row

const matchingRow = rows.find(row => {

const name = row[1].toLowerCase();

return name.includes(keyword.toLowerCase());

});

// Hide all existing cards

const cardsContainer = document.getElementById('cardsContainer');

cardsContainer.innerHTML = '';

if (matchingRow) {

const coords = matchingRow[0].split(',').map(Number);

const name = matchingRow[1];

const image = matchingRow[2];

const address = matchingRow[3];

const link = matchingRow[4];

const phone = matchingRow[5];

// Create the card content for the matching row

const cardContent = `

<div class="card">

<div class="card\_\_image">

<img src="${image}" alt="${name}" style="width: 100%; height: auto;" />

</div>

<div class="card\_\_info">

<h5>${name}</h5>

<p><b>Address:</b> ${address}</p>

<p><b>Phone:</b> ${phone}</p>

<a href="${link}" target="\_blank">Visit Us</a>

</div>

</div>

`;

// Display the corresponding card in the sidebar

cardsContainer.innerHTML = cardContent;

// Add click event to the card to collapse sidebar and zoom to location

document.querySelector('.card').addEventListener('click', () => {

document.getElementById('sidebar').classList.add('collapsed');

map.flyTo(coords, 18);

});

} else {

console.log('No matching location found');

}

})

.catch(error => console.error('Error fetching data:', error));

}

// Function to handle clicking the search button

function handleSearch() {

const keyword = document.getElementById('searchInput').value.trim().toLowerCase();

performSearch(keyword);

}

// Add an event listener for the search button click

document.getElementById('searchButton').addEventListener('click', handleSearch);

leaflet-sidebar.js

/\* global L \*/

/\*\*

\* @name Sidebar

\* @class L.Control.Sidebar

\* @extends L.Control

\* @param {string} id - The id of the sidebar element (without the # character)

\* @param {Object} [options] - Optional options object

\* @param {string} [options.position=left] - Position of the sidebar: 'left' or 'right'

\* @see L.control.sidebar

\*/

L.Control.Sidebar = L.Control.extend(/\*\* @lends L.Control.Sidebar.prototype \*/ {

includes: (L.Evented.prototype || L.Mixin.Events),

options: {

position: 'left'

},

initialize: function (id, options) {

var i, child;

L.setOptions(this, options);

// Find sidebar HTMLElement

this.\_sidebar = L.DomUtil.get(id);

// Attach .sidebar-left/right class

L.DomUtil.addClass(this.\_sidebar, 'sidebar-' + this.options.position);

// Attach touch styling if necessary

if (L.Browser.touch)

L.DomUtil.addClass(this.\_sidebar, 'leaflet-touch');

// Find sidebar > div.sidebar-content

for (i = this.\_sidebar.children.length - 1; i >= 0; i--) {

child = this.\_sidebar.children[i];

if (child.tagName == 'DIV' &&

L.DomUtil.hasClass(child, 'sidebar-content'))

this.\_container = child;

}

// Find sidebar ul.sidebar-tabs > li, sidebar .sidebar-tabs > ul > li

this.\_tabitems = this.\_sidebar.querySelectorAll('ul.sidebar-tabs > li, .sidebar-tabs > ul > li');

for (i = this.\_tabitems.length - 1; i >= 0; i--) {

this.\_tabitems[i].\_sidebar = this;

}

// Find sidebar > div.sidebar-content > div.sidebar-pane

this.\_panes = [];

this.\_closeButtons = [];

for (i = this.\_container.children.length - 1; i >= 0; i--) {

child = this.\_container.children[i];

if (child.tagName == 'DIV' &&

L.DomUtil.hasClass(child, 'sidebar-pane')) {

this.\_panes.push(child);

var closeButtons = child.querySelectorAll('.sidebar-close');

for (var j = 0, len = closeButtons.length; j < len; j++)

this.\_closeButtons.push(closeButtons[j]);

}

}

},

/\*\*

\* Add this sidebar to the specified map.

\*

\* @param {L.Map} map

\* @returns {Sidebar}

\*/

addTo: function (map) {

var i, child;

this.\_map = map;

for (i = this.\_tabitems.length - 1; i >= 0; i--) {

child = this.\_tabitems[i];

var sub = child.querySelector('a');

if (sub.hasAttribute('href') && sub.getAttribute('href').slice(0,1) == '#') {

L.DomEvent

.on(sub, 'click', L.DomEvent.preventDefault )

.on(sub, 'click', this.\_onClick, child);

}

}

for (i = this.\_closeButtons.length - 1; i >= 0; i--) {

child = this.\_closeButtons[i];

L.DomEvent.on(child, 'click', this.\_onCloseClick, this);

}

return this;

},

/\*\*

\* @deprecated - Please use remove() instead of removeFrom(), as of Leaflet 0.8-dev, the removeFrom() has been replaced with remove()

\* Removes this sidebar from the map.

\* @param {L.Map} map

\* @returns {Sidebar}

\*/

removeFrom: function(map) {

console.log('removeFrom() has been deprecated, please use remove() instead as support for this function will be ending soon.');

this.remove(map);

},

/\*\*

\* Remove this sidebar from the map.

\*

\* @param {L.Map} map

\* @returns {Sidebar}

\*/

remove: function (map) {

var i, child;

this.\_map = null;

for (i = this.\_tabitems.length - 1; i >= 0; i--) {

child = this.\_tabitems[i];

L.DomEvent.off(child.querySelector('a'), 'click', this.\_onClick);

}

for (i = this.\_closeButtons.length - 1; i >= 0; i--) {

child = this.\_closeButtons[i];

L.DomEvent.off(child, 'click', this.\_onCloseClick, this);

}

return this;

},

/\*\*

\* Open sidebar (if necessary) and show the specified tab.

\*

\* @param {string} id - The id of the tab to show (without the # character)

\*/

open: function(id) {

var i, child;

// hide old active contents and show new content

for (i = this.\_panes.length - 1; i >= 0; i--) {

child = this.\_panes[i];

if (child.id == id)

L.DomUtil.addClass(child, 'active');

else if (L.DomUtil.hasClass(child, 'active'))

L.DomUtil.removeClass(child, 'active');

}

// remove old active highlights and set new highlight

for (i = this.\_tabitems.length - 1; i >= 0; i--) {

child = this.\_tabitems[i];

if (child.querySelector('a').hash == '#' + id)

L.DomUtil.addClass(child, 'active');

else if (L.DomUtil.hasClass(child, 'active'))

L.DomUtil.removeClass(child, 'active');

}

this.fire('content', { id: id });

// open sidebar (if necessary)

if (L.DomUtil.hasClass(this.\_sidebar, 'collapsed')) {

this.fire('opening');

L.DomUtil.removeClass(this.\_sidebar, 'collapsed');

}

return this;

},

/\*\*

\* Close the sidebar (if necessary).

\*/

close: function() {

// remove old active highlights

for (var i = this.\_tabitems.length - 1; i >= 0; i--) {

var child = this.\_tabitems[i];

if (L.DomUtil.hasClass(child, 'active'))

L.DomUtil.removeClass(child, 'active');

}

// close sidebar

if (!L.DomUtil.hasClass(this.\_sidebar, 'collapsed')) {

this.fire('closing');

L.DomUtil.addClass(this.\_sidebar, 'collapsed');

}

return this;

},

/\*\*

\* @private

\*/

\_onClick: function() {

if (L.DomUtil.hasClass(this, 'active'))

this.\_sidebar.close();

else if (!L.DomUtil.hasClass(this, 'disabled'))

this.\_sidebar.open(this.querySelector('a').hash.slice(1));

},

/\*\*

\* @private

\*/

\_onCloseClick: function () {

this.close();

}

});

/\*\*

\* Creates a new sidebar.

\*

\* @example

\* var sidebar = L.control.sidebar('sidebar').addTo(map);

\*

\* @param {string} id - The id of the sidebar element (without the # character)

\* @param {Object} [options] - Optional options object

\* @param {string} [options.position=left] - Position of the sidebar: 'left' or 'right'

\* @returns {Sidebar} A new sidebar instance

\*/

L.control.sidebar = function (id, options) {

return new L.Control.Sidebar(id, options);

};

Leaflet-sidebar.css

.sidebar {

position: absolute;

top: 0;

bottom: 0;

width: 100%;

overflow: hidden;

z-index: 2000;

}

.sidebar.collapsed {

width: 40px;

}

@media (min-width: 768px) {

.sidebar {

top: 10px;

bottom: 10px;

transition: width 500ms;

}

}

@media (min-width: 768px) and (max-width: 991px) {

.sidebar {

width: 305px;

}

}

@media (min-width: 992px) and (max-width: 1199px) {

.sidebar {

width: 390px;

}

}

@media (min-width: 1200px) {

.sidebar {

width: 460px;

}

}

.sidebar-left {

left: 0;

}

@media (min-width: 768px) {

.sidebar-left {

left: 10px;

}

}

.sidebar-right {

right: 0;

}

@media (min-width: 768px) {

.sidebar-right {

right: 10px;

}

}

.sidebar-tabs {

top: 0;

bottom: 0;

height: 100%;

background-color: #fff;

}

.sidebar-left .sidebar-tabs {

left: 0;

}

.sidebar-right .sidebar-tabs {

right: 0;

}

.sidebar-tabs,

.sidebar-tabs > ul {

position: absolute;

width: 40px;

margin: 0;

padding: 0;

list-style-type: none;

}

.sidebar-tabs > li,

.sidebar-tabs > ul > li {

width: 100%;

height: 40px;

color: #333;

font-size: 12pt;

overflow: hidden;

transition: all 80ms;

}

.sidebar-tabs > li:hover,

.sidebar-tabs > ul > li:hover {

color: #000;

background-color: #eee;

}

.sidebar-tabs > li.active,

.sidebar-tabs > ul > li.active {

color: #fff;

background-color: #0074d9;

}

.sidebar-tabs > li.disabled,

.sidebar-tabs > ul > li.disabled {

color: rgba(51, 51, 51, 0.4);

}

.sidebar-tabs > li.disabled:hover,

.sidebar-tabs > ul > li.disabled:hover {

background: transparent;

}

.sidebar-tabs > li.disabled > a,

.sidebar-tabs > ul > li.disabled > a {

cursor: default;

}

.sidebar-tabs > li > a,

.sidebar-tabs > ul > li > a {

display: block;

width: 100%;

height: 100%;

line-height: 40px;

color: inherit;

text-decoration: none;

text-align: center;

}

.sidebar-tabs > ul + ul {

bottom: 0;

}

.sidebar-content {

position: absolute;

top: 0;

bottom: 0;

background-color: rgba(255, 255, 255, 0.95);

overflow-x: hidden;

overflow-y: auto;

}

.sidebar-left .sidebar-content {

left: 40px;

right: 0;

}

.sidebar-right .sidebar-content {

left: 0;

right: 40px;

}

.sidebar.collapsed > .sidebar-content {

overflow-y: hidden;

}

.sidebar-pane {

display: none;

left: 0;

right: 0;

box-sizing: border-box;

padding: 10px 20px;

}

.sidebar-pane.active {

display: block;

}

@media (min-width: 768px) and (max-width: 991px) {

.sidebar-pane {

min-width: 265px;

}

}

@media (min-width: 992px) and (max-width: 1199px) {

.sidebar-pane {

min-width: 350px;

}

}

@media (min-width: 1200px) {

.sidebar-pane {

min-width: 420px;

}

}

.sidebar-header {

margin: -10px -20px 0;

height: 40px;

padding: 0 20px;

line-height: 40px;

font-size: 14.4pt;

color: #fff;

background-color: #0074d9;

}

.sidebar-right .sidebar-header {

padding-left: 40px;

}

.sidebar-close {

position: absolute;

top: 0;

width: 40px;

height: 40px;

text-align: center;

cursor: pointer;

}

.sidebar-left .sidebar-close {

right: 0;

}

.sidebar-right .sidebar-close {

left: 0;

}

.sidebar-left ~ .sidebar-map {

margin-left: 40px;

}

@media (min-width: 768px) {

.sidebar-left ~ .sidebar-map {

margin-left: 0;

}

}

.sidebar-right ~ .sidebar-map {

margin-right: 40px;

}

@media (min-width: 768px) {

.sidebar-right ~ .sidebar-map {

margin-right: 0;

}

}

.sidebar {

box-shadow: 0 1px 5px rgba(0, 0, 0, 0.65);

}

.sidebar.leaflet-touch {

box-shadow: none;

border-right: 2px solid rgba(0, 0, 0, 0.2);

}

@media (min-width: 768px) {

.sidebar {

border-radius: 4px;

}

.sidebar.leaflet-touch {

border: 2px solid rgba(0, 0, 0, 0.2);

}

}

@media (min-width: 768px) {

.sidebar-left ~ .sidebar-map .leaflet-left {

transition: left 500ms;

}

}

@media (min-width: 768px) and (max-width: 991px) {

.sidebar-left ~ .sidebar-map .leaflet-left {

left: 315px;

}

}

@media (min-width: 992px) and (max-width: 1199px) {

.sidebar-left ~ .sidebar-map .leaflet-left {

left: 400px;

}

}

@media (min-width: 1200px) {

.sidebar-left ~ .sidebar-map .leaflet-left {

left: 470px;

}

}

@media (min-width: 768px) {

.sidebar-left.collapsed ~ .sidebar-map .leaflet-left {

left: 50px;

}

}

@media (min-width: 768px) {

.sidebar-right ~ .sidebar-map .leaflet-right {

transition: right 500ms;

}

}

@media (min-width: 768px) and (max-width: 991px) {

.sidebar-right ~ .sidebar-map .leaflet-right {

right: 315px;

}

}

@media (min-width: 992px) and (max-width: 1199px) {

.sidebar-right ~ .sidebar-map .leaflet-right {

right: 400px;

}

}

@media (min-width: 1200px) {

.sidebar-right ~ .sidebar-map .leaflet-right {

right: 470px;

}

}

@media (min-width: 768px) {

.sidebar-right.collapsed ~ .sidebar-map .leaflet-right {

right: 50px;

}

}

Stylem.css

@import url("https://fonts.googleapis.com/css2?family=Montserrat:wght@600;700&display=swap");

body {

font-family: "Montserrat", sans-serif;

}

.leaflet-popup-content-wrapper {

width: 350px;

background:white;

color:grey;

}

.container {

display: flex;

justify-content: center;

align-items: center;

}

.art-board {

border-radius: 40px;

border: 6px solid white;

box-shadow: -30px 70px 150px -20px rgba(51, 51, 51, 0.25);

}

.art-board\_\_container {

margin: 18px 0 18px 0;

width: 375px;

height: 812px;

overflow: auto;

display: flex;

flex-wrap: wrap;

justify-content: center;

align-items: center;

}

.card {

width: 295px;

overflow: hidden;

border-radius: 25px;

border: 1px solid white;

margin: 10px;

background-color:white;

box-shadow: 5px 5px 15px 5px rgba(230, 230, 250, 1);

}

.card\_\_image {

position: relative;

height: 160px;

}

.card\_\_image > img {

max-width: auto;

height: 150px;

width: 100%;

}

.card\_\_info {

display: flex;

align-items: flex-end;

justify-content: space-between;

padding: 0px 30px 20px 30px;

}

.card\_\_info h3 {

font-size: 18px;

font-weight: 700;

}

.card\_\_info p {

font-size: 14px;

font-weight: 600;

}

.card\_\_info--price {

text-align: right;

color: orangered;

}

.rect2 {

position: relative;

bottom: 315px;

right: 15px;

width: 325px;

height: 230px;

border: 1px solid white;

border-radius: 0 0px 80px 0;

background-image: url();

transform: rotate(10deg);

object-fit: cover;

}

.checked {

color: #ffd700;

}

::-webkit-scrollbar {

width: 8px;

}

::-webkit-scrollbar-thumb {

background: #cccccc;

border-radius: 10px;

}

::-webkit-scrollbar-track {

-webkit-box-shadow: inset 0 0 6px rgba(0, 0, 0, 0.3);

border-radius: 10px;

}

#map {

height: 600px;

width: 100%;

}

.leaflet-marker-icon leaflet-zoom-animated leaflet-interactive{

color: #68241C;

}

.leaflet-popup-tip-container {

width: 40px;

height: 11px;

position: absolute;

left: 50%;

margin-left: -7px;

overflow: hidden;

pointer-events: none;

}

.search-icon {

margin-left: 5px;

cursor: pointer;

}

.suggestions {

background-color: white;

border: 1px solid #ccc;

max-height: 150px;

overflow-y: auto;

display: none; /\* Hide by default \*/

position: absolute;

width: 100%;

}

.suggestions div {

padding: 5px;

cursor: pointer;

}

.suggestions div:hover {

background-color: #f0f0f0;

}

/\*https://docs.google.com/spreadsheets/d/e/2PACX-1vTm5v47OCV9XqXoysBSrRXHl74wLuQyDHO3s7gfYT9bkegNEoB9q6psyfV39KeJVRv0h738WM9CdGSD/pubhtml

[https://docs.google.com/spreadsheets/d/1PhVePG\_hSOgdlBZBpO0bNwXG5vQ12fo2h4SpyRXvdkw/edit?gid=1848024179#gid=1848024179\*/](https://docs.google.com/spreadsheets/d/1PhVePG_hSOgdlBZBpO0bNwXG5vQ12fo2h4SpyRXvdkw/edit?gid=1848024179#gid=1848024179*/)

Without direction

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Map with Sidebar and Search</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<script src='https://api.mapbox.com/mapbox-gl-js/v2.9.1/mapbox-gl.js'></script>

<link href='https://api.mapbox.com/mapbox-gl-js/v2.9.1/mapbox-gl.css' rel='stylesheet' />

<link rel="stylesheet" href="leaflet-sidebar.css"/>

<link rel="stylesheet" href="stylem.css"/>

<style>

body { margin: 0; padding: 0; }

#map { position: absolute; top: 0; bottom: 0; width:100%; height: 100%; }

#sidebar { position: absolute; top: 0; right: 0; width: 25%; height: 100%; overflow-y: scroll; background: white; display: none; }

.card { padding: 10px; border-bottom: 1px solid #ccc; cursor: pointer; }

.custom-marker { background-size: cover; width: 50px; height: 50px; border-radius: 50%; cursor: pointer; }

.autocomplete { position: relative; display: inline-block; }

.autocomplete-items { position: absolute; border: 1px solid #d4d4d4; border-bottom: none; border-top: none; z-index: 99; top: 100%; left: 0; right: 0; }

.autocomplete-items div { padding: 10px; cursor: pointer; background-color: #fff; border-bottom: 1px solid #d4d4d4; }

.autocomplete-items div:hover { background-color: #e9e9e9; }

.autocomplete-active { background-color: DodgerBlue !important; color: #ffffff; }

#searchContainer { position: absolute; top: 10px; left: 10px; z-index: 1; display: flex; flex-direction: column; }

#searchIcon { cursor: pointer; padding: 10px; border: 1px solid #ccc; background: white; }

#searchResults { margin-top: 10px; }

.close-icon { cursor: pointer; float: right; font-size: 20px; }

</style>

</head>

<body>

<div id='map'></div>

<div id='sidebar'>

<div id='cardsContainer'></div>

</div>

<div id="searchContainer">

<div style="display: flex; align-items: center;">

<div id="searchIcon">🔍</div>

<div class="autocomplete">

<input id="searchBar" type="text" placeholder="Search..." style="width: 300px; padding: 10px; border-radius: 5px; border: 1px solid #ccc;">

</div>

</div>

<div id="searchResults"></div>

</div>

<div id="3dButton" style="position: absolute; bottom:10px; right: 10px; z-index: 10; background: #fff; border: 1px solid #ccc; padding: 10px; border-radius: 5px; cursor: pointer; box-shadow: 0 2px 5px rgba(0,0,0,0.2);font-weight: bold;">

3D View

</div>

<script src="https://api.mapbox.com/mapbox-gl-js/v3.4.0/mapbox-gl.js"></script>

<script>

mapboxgl.accessToken = 'pk.eyJ1Ijoic3llZC1iYXNpbGEiLCJhIjoiY2x4cHg3dzM0MGQ2dDJyczRuOHRibnhpcyJ9.WpzsnLAEpVH2tCfTQgYqDQ';

const map = new mapboxgl.Map({

container: 'map',

style: 'mapbox://styles/syed-basila/clxpyopvr00mg01r217tz19xy', // Custom style URL

center: [ 78.695717,10.786447],

zoom: 12

});

map.addControl(new mapboxgl.NavigationControl());

function addMarkers(data) {

const markers = [];

const locations = [];

data.forEach((row, index) => {

const coords = row[0].split(',').map(Number);

const name = row[1];

const image = row[2];

const address = row[3];

const link = row[4];

const phone = row[5];

locations.push({ coords, name, image, address, link, phone, index });

const markerElement = document.createElement('div');

markerElement.className = 'custom-marker';

markerElement.style.backgroundImage = `url(${image})`;

const popupContent = `

<div class="popup">

<h3>${name}</h3>

<img src="${image}" alt="${name}" style="width: 100%; height: auto;" />

<p><b>Address:</b> ${address}</p>

<p><b>Phone:</b> ${phone}</p>

<a href="${link}" target="\_blank">Visit Us</a>

</div>

`;

const marker = new mapboxgl.Marker(markerElement)

.setLngLat(coords)

.setPopup(new mapboxgl.Popup().setHTML(popupContent))

.addTo(map);

markers.push(marker);

const cardElement = document.createElement('div');

cardElement.className = 'card';

cardElement.innerHTML = `

<div class="card\_\_image">

<img src="${image}" alt="${name}" />

</div>

<div class="card\_\_info">

<div class="car\_\_info--title">

<p>${name}</p>

</div>

</div>

`;

cardElement.addEventListener('click', () => {

map.flyTo({ center: coords, zoom: 18 });

marker.togglePopup();

});

document.getElementById('cardsContainer').appendChild(cardElement);

});

autocomplete(document.getElementById('searchBar'), locations);

}

function autocomplete(inp, arr) {

let currentFocus;

inp.addEventListener('input', function(e) {

let a, b, i, val = this.value;

closeAllLists();

if (!val) { return false; }

currentFocus = -1;

a = document.createElement('DIV');

a.setAttribute('id', this.id + 'autocomplete-list');

a.setAttribute('class', 'autocomplete-items');

this.parentNode.appendChild(a);

for (i = 0; i < arr.length; i++) {

if (arr[i].name.substr(0, val.length).toUpperCase() == val.toUpperCase()) {

b = document.createElement('DIV');

b.innerHTML = '<strong>' + arr[i].name.substr(0, val.length) + '</strong>';

b.innerHTML += arr[i].name.substr(val.length);

b.innerHTML += "<input type='hidden' value='" + arr[i].name + "' data-coords='" + arr[i].coords + "' data-index='" + arr[i].index + "' data-image='" + arr[i].image + "' data-address='" + arr[i].address + "' data-link='" + arr[i].link + "' data-phone='" + arr[i].phone + "'>";

b.addEventListener('click', function(e) {

inp.value = this.getElementsByTagName('input')[0].value;

const coords = this.getElementsByTagName('input')[0].getAttribute('data-coords').split(',').map(Number);

const index = this.getElementsByTagName('input')[0].getAttribute('data-index');

const image = this.getElementsByTagName('input')[0].getAttribute('data-image');

const address = this.getElementsByTagName('input')[0].getAttribute('data-address');

const link = this.getElementsByTagName('input')[0].getAttribute('data-link');

const phone = this.getElementsByTagName('input')[0].getAttribute('data-phone');

showSearchedCard(index, coords, image, address, link, phone);

closeAllLists();

});

a.appendChild(b);

}

}

});

inp.addEventListener('keydown', function(e) {

let x = document.getElementById(this.id + 'autocomplete-list');

if (x) x = x.getElementsByTagName('div');

if (e.keyCode == 40) {

currentFocus++;

addActive(x);

} else if (e.keyCode == 38) {

currentFocus--;

addActive(x);

} else if (e.keyCode == 13) {

e.preventDefault();

if (currentFocus > -1) {

if (x) x[currentFocus].click();

}

}

});

function addActive(x) {

if (!x) return false;

removeActive(x);

if (currentFocus >= x.length) currentFocus = 0;

if (currentFocus < 0) currentFocus = (x.length - 1);

x[currentFocus].classList.add('autocomplete-active');

}

function removeActive(x) {

for (let i = 0; i < x.length; i++) {

x[i].classList.remove('autocomplete-active');

}

}

function closeAllLists(elmnt) {

const x = document.getElementsByClassName('autocomplete-items');

for (let i = 0; i < x.length; i++) {

if (elmnt != x[i] && elmnt != inp) {

x[i].parentNode.removeChild(x[i]);

}

}

}

document.addEventListener('click', function (e) {

closeAllLists(e.target);

});

}

function showSearchedCard(index, coords, image, address, link, phone) {

const searchResults = document.getElementById('searchResults');

searchResults.innerHTML = ''; // Clear previous search results

const cardElement = document.createElement('div');

cardElement.className = 'card';

cardElement.innerHTML = `

<div class="close-icon">&times;</div>

<div class="card\_\_image">

<img src="${image}" alt="${name}" />

</div>

<div class="card\_\_info">

<div class="car\_\_info--title">

<p>${name}</p>

</div>

<p><b>Address:</b> ${address}</p>

<p><b>Phone:</b> ${phone}</p>

<a href="${link}" target="\_blank">Visit Us</a>

</div>

`;

// Close the card when the close icon is clicked

cardElement.querySelector('.close-icon').addEventListener('click', () => {

searchResults.innerHTML = ''; // Clear the card

});

// Handle card click

cardElement.addEventListener('click', () => {

// Close the card first

searchResults.innerHTML = '';

// Show location on map

map.flyTo({ center: coords, zoom: 18 });

// Find and click the marker

const markerElements = document.querySelectorAll('.mapboxgl-marker');

if (markerElements[index]) {

markerElements[index].click();

}

});

searchResults.appendChild(cardElement);

}

const sheetID = '1zkxjCUv4EhFl34AuN8HhNE3DazbinGkKrR17OPBlunU';

const sheetRange = 'mapdata!A2:F30';

const apiKey = 'AIzaSyAJBec7\_E9UNpNbGnFjrcCr4oS8gdVgtck';

const sheetURL = `https://sheets.googleapis.com/v4/spreadsheets/${sheetID}/values/${sheetRange}?key=${apiKey}`;

fetch(sheetURL)

.then(response => response.json())

.then(data => {

const rows = data.values.slice(1); // Remove header row

addMarkers(rows);

})

.catch(error => console.error('Error fetching data:', error));

document.getElementById('3dButton').addEventListener('click', () => {

const is3D = map.getPitch() > 0;

if (is3D) {

// Switch to 2D view

map.easeTo({

pitch: 0

});

document.getElementById('3dButton').innerText = '3D View';

} else {

// Switch to 3D view

map.easeTo({

pitch: 90

});

document.getElementById('3dButton').innerText = '2D View';

}

});

</script>

</body>

</html>