<!DOCTYPE html>

<html>

<head>

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

<meta charset="utf-8">

<title>Places Searchbox</title>

<style>

/\* Always set the map height explicitly to define the size of the div

\* element that contains the map. \*/

#map {

height: 100%;

}

/\* Optional: Makes the sample page fill the window. \*/

html, body {

height: 100%;

margin: 0;

padding: 0;

}

#description {

font-family: Roboto;

font-size: 15px;

font-weight: 300;

}

#infowindow-content .title {

font-weight: bold;

}

#infowindow-content {

display: none;

}

#map #infowindow-content {

display: inline;

}

.pac-card {

margin: 10px 10px 0 0;

border-radius: 2px 0 0 2px;

box-sizing: border-box;

-moz-box-sizing: border-box;

outline: none;

box-shadow: 0 2px 6px rgba(0, 0, 0, 0.3);

background-color: #fff;

font-family: Roboto;

}

#pac-container {

padding-bottom: 12px;

margin-right: 12px;

}

.pac-controls {

display: inline-block;

padding: 5px 11px;

}

.pac-controls label {

font-family: Roboto;

font-size: 13px;

font-weight: 300;

}

#pac-input {

background-color: #fff;

font-family: Roboto;

font-size: 15px;

font-weight: 300;

margin-left: 12px;

padding: 0 11px 0 13px;

text-overflow: ellipsis;

width: 400px;

}

#pac-input:focus {

border-color: #4d90fe;

}

#title {

color: #fff;

background-color: #4d90fe;

font-size: 25px;

font-weight: 500;

padding: 6px 12px;

}

#target {

width: 345px;

}

</style>

</head>

<body>

<input id="pac-input" class="controls" type="text" placeholder="Search Box">

<div id="map"></div>

<script>

// This example adds a search box to a map, using the Google Place Autocomplete

// feature. People can enter geographical searches. The search box will return a

// pick list containing a mix of places and predicted search terms.

// This example requires the Places library. Include the libraries=places

// parameter when you first load the API. For example:

// <script src="https://maps.googleapis.com/maps/api/js?key=YOUR\_API\_KEY&libraries=places">

function initAutocomplete() {

var map = new google.maps.Map(document.getElementById('map'), {

center: {lat: -33.8688, lng: 151.2195},

zoom: 13,

mapTypeId: 'roadmap'

});

// Create the search box and link it to the UI element.

var input = document.getElementById('pac-input');

var searchBox = new google.maps.places.SearchBox(input);

map.controls[google.maps.ControlPosition.TOP\_LEFT].push(input);

// Bias the SearchBox results towards current map's viewport.

map.addListener('bounds\_changed', function() {

searchBox.setBounds(map.getBounds());

});

var markers = [];

// Listen for the event fired when the user selects a prediction and retrieve

// more details for that place.

searchBox.addListener('places\_changed', function() {

var places = searchBox.getPlaces();

if (places.length == 0) {

return;

}

// Clear out the old markers.

markers.forEach(function(marker) {

marker.setMap(null);

});

markers = [];

// For each place, get the icon, name and location.

var bounds = new google.maps.LatLngBounds();

places.forEach(function(place) {

if (!place.geometry) {

console.log("Returned place contains no geometry");

return;

}

var icon = {

url: place.icon,

size: new google.maps.Size(71, 71),

origin: new google.maps.Point(0, 0),

anchor: new google.maps.Point(17, 34),

scaledSize: new google.maps.Size(25, 25)

};

// Create a marker for each place.

markers.push(new google.maps.Marker({

map: map,

icon: icon,

title: place.name,

position: place.geometry.location

}));

if (place.geometry.viewport) {

// Only geocodes have viewport.

bounds.union(place.geometry.viewport);

} else {

bounds.extend(place.geometry.location);

}

});

map.fitBounds(bounds);

});

}

</script>

<script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyDCGsb3YabK4wGR7Ryx9t7dwbtu9YeC0Pw&libraries=places&callback=initAutocomplete"

async defer></script>

</body>

</html>