LAB2 ASSIGNMENT

NAME: RAVI SANKAR ARE

ST.ID :16144391.

1. I Implemented complex direction service .
2. I created two labels for input and output to get desired directions.
3. When user typed the input and output then it calls Jquery function

$('#control').ready(function () {

initialize(41.850033, -87.6500523);

$('#submit').click(function () {

calcRoute();

1. Whenever user click submit button then it calls google direction function API

And then accordingly it find the directions.

HTML CODE:

<script src="https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=true"></script>

<body>

<div id="panel">

<input type="text" name="start" id="start" value="" />

<input type="text" name="end" id="end" value="" />

<input type="submit" id="submit" value="submit" />

</div>

<div id="map-canvas" style="height:500px;width:100%;"></div>

<div id="warnings\_panel" style="width:100%;height:10%;text-align:center"></div>

</body>

JAVA SCRIPT CODE:

$('#control').ready(function () {

initialize(41.850033, -87.6500523);

$('#submit').click(function () {

calcRoute();

});

});

var map;

var directionsDisplay;

var directionsService;

var stepDisplay;

var markerArray = [];

function initialize(lat,lan) {

// Instantiate a directions service.

directionsService = new google.maps.DirectionsService();

// Create a map and center it on Manhattan.

var manhattan = new google.maps.LatLng(lat,lan);

var mapOptions = {

zoom: 13,

mapTypeId: google.maps.MapTypeId.ROADMAP,

center: manhattan

}

map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);

// Create a renderer for directions and bind it to the map.

var rendererOptions = {

map: map

}

directionsDisplay = new google.maps.DirectionsRenderer(rendererOptions)

// Instantiate an info window to hold step text.

stepDisplay = new google.maps.InfoWindow();

}

function calcRoute() {

// First, remove any existing markers from the map.

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

markerArray[i].setMap(null);

}

// Now, clear the array itself.

markerArray = [];

// Retrieve the start and end locations and create

// a DirectionsRequest using WALKING directions.

var start = document.getElementById('start').value;

var end = document.getElementById('end').value;

var request = {

origin: start,

destination: end,

travelMode: google.maps.TravelMode.WALKING

};

// Route the directions and pass the response to a

// function to create markers for each step.

directionsService.route(request, function(response, status) {

if (status == google.maps.DirectionsStatus.OK) {

var warnings = document.getElementById('warnings\_panel');

warnings.innerHTML = '<b>' + response.routes[0].warnings + '</b>';

directionsDisplay.setDirections(response);

showSteps(response);

}

});

}

function showSteps(directionResult) {

// For each step, place a marker, and add the text to the marker's

// info window. Also attach the marker to an array so we

// can keep track of it and remove it when calculating new

// routes.

var myRoute = directionResult.routes[0].legs[0];

for (var i = 0; i < myRoute.steps.length; i++) {

var marker = new google.maps.Marker({

position: myRoute.steps[i].start\_point,

map: map

});

attachInstructionText(marker, myRoute.steps[i].instructions);

markerArray[i] = marker;

}

}

function attachInstructionText(marker, text) {

google.maps.event.addListener(marker, 'click', function() {

// Open an info window when the marker is clicked on,

// containing the text of the step.

stepDisplay.setContent(text);

stepDisplay.open(map, marker);

});

}

CSS CODE:

html, body {

height: 100%;

margin: 0;

padding: 0;

}

#map-canvas, #map\_canvas {

height: 100%;

}

@media print {

html, body {

height: auto;

}

#map-canvas, #map\_canvas {

height: 650px;

}

}

#panel {

position: absolute;

top: 5px;

left: 50%;

margin-left: -180px;

z-index: 5;

background-color: #fff;

padding: 5px;

border: 1px solid #999;

}.