var Addresses = [];

var JsonDataObj = [];

var new\_table;

var googleMatrixJSON;

var myData;

var unload\_Time;

var delivery\_Address = [];

var array1 = [];

var mytempFinalAddress = [];

var origin;

function Calculate\_Distance(value) {

var words = value.split(" ");

var distance = words[0];

var distance\_unit = words[1];

if (distance\_unit.includes("km")) {

distance = distance \* 1000;

}

else {

distance = distance \* 1;

}

return distance;

}

function Calculate\_Time(value) {

var total\_time;

if (value.includes("hour") || value.includes("hours")) {

var words = value.split(" ");

var hour = words[0] \* 60;

var min = words[2] \* 1;

total\_time = hour + min;

}

else {

var words = value.split(" ");

var min = words[0];

total\_time = min;

}

return total\_time;

}

function convertHTML\_JSON() {

var table = $('#tblGetDistance').tableToJSON();

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

new\_table = {

Origin: table[i].Origin,

Destination: table[i].Destination,

Distance: Calculate\_Distance(table[i].Distance),

Duration: Calculate\_Time(table[i].Duration)

};

googleMatrixJSON = JSON.stringify(new\_table);

JsonDataObj.push(googleMatrixJSON);

}

return myData = '[' + JsonDataObj + ']';

}

function GetUnloadTime() {

var num = unload\_Time;

var firstpart = Math.trunc(num);

var secondpart = Math.round(100 \* Math.abs(num - firstpart));

var total;

if (firstpart == 0) {

total = secondpart;

return total;

}

else {

total = firstpart \* 60 + secondpart;

return total;

}

}

function initMap() {

var mylength = document.getElementById("tblGetAddress").rows.length;

for (var i = 1; i < mylength; i++) {

Addresses.push(document.getElementById("tblGetAddress").rows[i].cells[1].innerHTML);

// console.log(Addresses);

}

var service = new google.maps.DistanceMatrixService();

service.getDistanceMatrix(

{

origins: Addresses,

destinations: Addresses,

travelMode: 'DRIVING',

unitSystem: google.maps.UnitSystem.METRIC,

avoidHighways: false,

avoidTolls: false,

}, callback);

function callback(response, status) {

// See Parsing the Results for

// the basics of a callback function.

if (status == 'OK') {

var origins = response.originAddresses;

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

mytempFinalAddress.push(origins[i]);

}

var destinations = response.destinationAddresses;

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

var results = response.rows[i].elements;

for (var j = 0; j < results.length; j++) {

var element = results[j];

var distance = element.distance.text;

var duration = element.duration.text;

var from = origins[i];

var to = destinations[j];

$("#tblGetDistance").append("<tr><td>" + from + "</td><td>" + to + "</td><td>" + distance + "</td><td>" + duration + "</td></tr>");

}

}

}

}

}

$(document).ready(function () {

$("#btnInsert").click(function () {

$.ajax({

//http://localhost/Service\_Database\_Connection/Service.svc/rest

url: "http://192.168.0.109/Back-End/Service1.svc/rest/InsertDistance\_Data",

type: "POST",

contentType: "application/json",

dataType: "json",

data: convertHTML\_JSON(), // googleMatrixJSON ,

success: function (result) {

// console.info(result);

// alert(result);

// alert("hehehe");

},

error: function (e) {

alert("something went wrong!");

}

}).then(function (result) {

// console.log(result);

// alert(result);

});

})

});

function fillRoute() {

$.ajax({

url: globalVariable.URL\_CSHARP + globalVariable.URL\_OPTIMAL\_ROUTE\_GET\_CONFIG\_OPTIMAL\_ROUTE,

type: "GET",

async: false,

Accept: "application/json",

success: function (resultdata) {

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

var Maximum\_Hour = resultdata[i].Maximum\_Hour;

var Name = resultdata[i].Name;

unload\_Time = resultdata[i].Unload\_Time;

console.log(unload\_Time);

}

},

error: function (e) {

alert("something went wrong!");

}

});

$.ajax({

url: globalVariable.URL\_CSHARP + globalVariable.URL\_OPTIMAL\_ROUTE\_GETOPTIMAL,

type: "GET",

async: false,

Accept: "application/json",

success: function (resultdata1) {

var total\_final\_time = 0;

var final\_hour;

var final\_minutes;

// console.log(unload\_Time);

//alert(unload);

var myunloadtime = GetUnloadTime();

// alert(myunloadtime);

// console.log(myunloadtime);

alert(myunloadtime);

for (var i = 0; i <= resultdata1.length; i++) {

var id = resultdata1[i].Id;

var origin = resultdata1[i].Origin;

var destination = resultdata1[i].Destination;

var distance = resultdata1[i].Distance;

var duration = resultdata1[i].Duration;

delivery\_Address.push(origin);

//console.log(unload\_Time);

var time = duration;

// var myunloadtime = 30;

if (i == (resultdata1.length - 1)) {

myunloadtime = 0;

}

total\_final\_time = total\_final\_time + time + myunloadtime;

var hour = Math.trunc(time / 60);

var minutes = time % 60;

final\_hour = Math.trunc(total\_final\_time / 60);

final\_minutes = total\_final\_time % 60;

// alert(delivery\_Address);

// console.log(delivery\_Address);

$("#tblGetOptimalRoute").append("<tr><td>" + origin + "</td><td>" + destination + "</td><td>" + distance / 1000 + " km" + "</td><td>" + hour + " hours " + minutes + " mins " + "</td><td>" + " " + final\_hour + " hours " + final\_minutes + " mins" + "</td></tr > ");

}

},

error: function (e) {

alert("something went wrong!");

}

});

}

//$(document).ready(function () {

// $("#btnOptimalRoute").click(function () {

// $.ajax({

// url:globalVariable.URL\_CSHARP + globalVariable.URL\_OPTIMAL\_ROUTE\_GET\_CONFIG\_OPTIMAL\_ROUTE,

// type: "GET",

// Accept: "application/json",

// success: function (resultdata) {

// for (var i = 0; i <= resultdata.length; i++)

// {

// var Maximum\_Hour = resultdata[i].Maximum\_Hour;

// var Name = resultdata[i].Name;

// unload\_Time = resultdata[i].Unload\_Time;

// console.log(unload\_Time);

// }

// },

// error: function (e) {

// alert("something went wrong!");

// }

// });

// $.ajax({

// url: globalVariable.URL\_CSHARP + globalVariable.URL\_OPTIMAL\_ROUTE\_GETOPTIMAL,

// type: "GET",

// Accept: "application/json",

// success: function (resultdata1) {

// var total\_final\_time = 0;

// var final\_hour;

// var final\_minutes;

// // console.log(unload\_Time);

// //alert(unload);

// var myunloadtime = GetUnloadTime();

// alert(myunloadtime);

// // console.log(myunloadtime);

// for (var i = 0; i <= resultdata1.length; i++) {

// var id = resultdata1[i].Id;

// var origin = resultdata1[i].Origin;

// var destination = resultdata1[i].Destination;

// var distance = resultdata1[i].Distance;

// var duration = resultdata1[i].Duration;

// delivery\_Address.push(origin);

// //console.log(unload\_Time);

// var time = duration;

// // var myunloadtime = 30;

// if (i == (resultdata1.length - 1) ) {

// myunloadtime = 0;

// }

// total\_final\_time = total\_final\_time + time+ myunloadtime;

// var hour = Math.trunc(time / 60);

// var minutes = time % 60;

// final\_hour = Math.trunc(total\_final\_time / 60);

// final\_minutes = total\_final\_time % 60;

// // alert(delivery\_Address);

// console.log(delivery\_Address);

// $("#tblGetOptimalRoute").append("<tr><td>" + origin + "</td><td>" + destination + "</td><td>" + distance / 1000 + " km" + "</td><td>" + hour + " hours " + minutes + " mins " + "</td><td>" + " " + final\_hour + " hours " + final\_minutes + " mins" + "</td></tr > ");

// }

// },

// error: function (e) {

// alert("something went wrong!");

// }

// });

// });

//});

$(document).ready(function () {

$("#btnDeliveryOptimalAddress").click(function () {

for (var i = 1; i <= delivery\_Address.length - 1; i++) {

var address = delivery\_Address[i];

$("#tblOptimalAddress").append("<tr><td>" + address + "</td></tr>");

}

});

});

$(document).ready(function () {

$("#btnNonDeliveryOptimalAddress").click(function () {

//var array1 = new Array("a", "b", "c", "d", "e", "f");

//var array2 = new Array("c", "e");

//var trytestADDRESS= ["Zeugstraat 92, 2801 JD, Gouda", "Jacob van Lennepstraat 46, 1053 HL, Amsterdam", "Lambertus Buddestraat 70, 7521 SB, Enschede", "Groningerweg 45/2, 9738 AB, Groningen"];

// console.log(Addresses);

// console.log(delivery\_Address);

// var array3 = array1.filter(val => !array2.includes(val));

// console.log(array3);

// console.log(Addresses);

// console.log(delivery\_Address);

var array4;

array4 = mytempFinalAddress.filter(val => !delivery\_Address.includes(val));

console.log(array4);

$("#tblNonDeliveryAddress").append("<tr><td>" + array4 + "</td></tr>");

});

});