Міністерство освіти і науки України

Національний університет «Львівська політехніка»



**Лабораторна робота №15**

на тему:

«Розробка додатку Мешап»

з курсу:

«Алгоритмізація та програмування»

Виконав:

ст. гр. КН-110

Король Орест

Прийняв:

Кривенчук Ю.П.

Львів – 2017 р.

*Лабораторна робота №15*

***Тема роботи:*** *Розробити додаток Мешап*

***Мета роботи:*** покращити свої навички у роботі з веб-технологіями, навчитись їх правильно використовувати, створити перший повноцінний додаток

**Завдання**

Реалізувати усі інструкції наведені у останньому завданні від курсу цс50. Розробити веб-додаток, який працюватиме з геолокацією та новинами.

​ **Текст програм**

<?php

require(\_\_DIR\_\_ . "/../includes/config.php");

// get querystring params, urldecoding and trimming leading/trailing whitespace in the process

$params = array\_map('trim', explode(",", urldecode($\_GET["geo"])));

// build sql\_query statement

$sql\_query = "SELECT \* FROM places WHERE ";

for ($i = 0, $count = count($params); $i < $count; $i++) {

// if param is numeric, assume a postal code

if (is\_numeric($params[$i]))

{

$sql\_query .= 'POSTAL\_CODE LIKE "' . htmlspecialchars($params[$i], ENT\_QUOTES) . '%"';

}

else

{

$sql\_query .=

'(PLACE\_NAME LIKE "' . htmlspecialchars($params[$i], ENT\_QUOTES) . '%" OR ' .

(strlen($params[$i]) <= 2 ? 'ADMIN\_CODE1 LIKE "' . htmlspecialchars($params[$i], ENT\_QUOTES) . '%" OR ' : "") .

'ADMIN\_NAME1 LIKE "' . htmlspecialchars($params[$i], ENT\_QUOTES) . '%")';

}

if ($i < ($count - 1)) {

$sql\_query .= " AND ";

}

}

// search database for places matching $\_GET["geo"]

$places = query($sql\_query);

// output places as JSON (pretty-printed for debugging convenience)

header("Content-type: application/json");

print(json\_encode($places, JSON\_PRETTY\_PRINT));

?>

/\*\*

\* scripts.js

\*

\* Computer Science 50

\* Problem Set 8

\*

\* Global JavaScript.

\*/

// Google Map

var map;

// markers for map

var markers = [];

// info window

var info = new google.maps.InfoWindow();

// execute when the DOM is fully loaded

$(function() {

// styles for map

// https://developers.google.com/maps/documentation/javascript/styling

var styles = [

// hide Google's labels

{

featureType: "all",

elementType: "labels",

stylers: [

{visibility: "off"}

]

},

// hide roads

{

featureType: "road",

elementType: "geometry",

stylers: [

{visibility: "off"}

]

}

];

// options for map

// https://developers.google.com/maps/documentation/javascript/reference#MapOptions

var options = {

center: {lat:42.3770, lng: -71.1256}, // Cambridge

disableDefaultUI: true,

mapTypeId: google.maps.MapTypeId.ROADMAP,

maxZoom: 200,

panControl: true,

styles: styles,

zoom: 13,

zoomControl: true

};

// get DOM node in which map will be instantiated

var canvas = $("#map-canvas").get(0);

// instantiate map

map = new google.maps.Map(canvas, options);

// configure UI once Google Map is idle (i.e., loaded)

google.maps.event.addListenerOnce(map, "idle", configure);

});

/\*\*

\* Adds marker for place to map.

\*/

function addMarker(place)

{

// set the Lat/Lng of the place

var placeLatLng = new google.maps.LatLng(parseFloat(place.latitude), parseFloat(place.longitude));

var beachMarker = new google.maps.Marker({

position: placeLatLng,

map: map,

icon: image

});

// add the marker to the map

var image = "img/info.png";

var marker = new MarkerWithLabel({

position: placeLatLng,

map: map,

icon: image,

labelContent: place.place\_name + ", " + place.admin\_name1,

labelAnchor: new google.maps.Point(20, 0),

labelClass: "label",

labelStyle: {opacity: 1.0}

});

// set a listener for the info window

google.maps.event.addListener(marker, "click", function() { infoData(marker, place.place\_name + "," + place.admin\_code1); });

// add marker to markers[]

markers.push(marker);

}

/\*\*

\* Configures application.

\*/

function configure()

{

// update UI after map has been dragged

google.maps.event.addListener(map, "dragend", function() {

update();

});

// update UI after zoom level changes

google.maps.event.addListener(map, "zoom\_changed", function() {

update();

});

// remove markers whilst dragging

google.maps.event.addListener(map, "dragstart", function() {

removeMarkers();

});

// configure typeahead

// https://github.com/twitter/typeahead.js/blob/master/doc/jquery\_typeahead.md

$("#q").typeahead({

autoselect: true,

highlight: true,

minLength: 1

},

{

source: search,

templates: {

empty: "no places found yet",

suggestion: \_.template("<p><%- place\_name %>, <%- admin\_name1 %>, <%- postal\_code %></p>")

}

});

// update text field value with changed cursor value

$("#q").on("typeahead:cursorchanged", function(eventObject, suggestion, name) {

$('#q').val(suggestion.place\_name + ', ' + suggestion.admin\_name1 + ', ' + suggestion.postal\_code);

});

// re-center map and update UI after place is selected from drop-down

$("#q").on("typeahead:selected", function(eventObject, suggestion, name) {

map.setCenter({lat: parseFloat(suggestion.latitude), lng: parseFloat(suggestion.longitude)});

update();

});

// hide info window when text box has focus

$("#q").focus(function(eventData) {

hideInfo();

});

// re-enable ctrl- and right-clicking (and thus Inspect Element) on Google Map

// https://chrome.google.com/webstore/detail/allow-right-click/hompjdfbfmmmgflfjdlnkohcplmboaeo?hl=en

document.addEventListener("contextmenu", function(event) {

event.returnValue = true;

event.stopPropagation && event.stopPropagation();

event.cancelBubble && event.cancelBubble();

}, true);

// update UI

update();

// give focus to text box

$("#q").focus();

}

/\*\*

\* Hides info window.

\*/

function hideInfo()

{

info.close();

}

/\*\*

\* Removes markers from map.

\*/

function removeMarkers()

{

// iterate through markers[] and remove each of them

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

markers[i].setMap(null);

markers[i] = null;

}

// empty markers array

markers.length = 0;

}

/\*\*

\* Searches database for typeahead's suggestions.

\*/

function search(query, cb)

{

// get places matching query (asynchronously)

var parameters = {

geo: query

};

$.getJSON("search.php", parameters)

.done(function(data, textStatus, jqXHR) {

// call typeahead's callback with search results (i.e., places)

cb(data);

})

.fail(function(jqXHR, textStatus, errorThrown) {

// log error to browser's console

console.log(errorThrown.toString());

});

}

/\*\*

\* gets JSON of local weather

\*/

function infoData(marker, query)

{

// get weather for query

var parameters = { q: query };

$.getJSON("http://api.openweathermap.org/data/2.5/weather", parameters)

.done(function(data, textStatus, jqXHR) {

// build weather display data

var weatherInfo;

if (data) {

var temp\_c = Math.round(data.main.temp - 273.15);

var temp\_f = Math.round((temp\_c \* 9) / 5 + 32);

weatherInfo = "<p><b>Current Weather:</b><br/>" + temp\_f + "&degF / " + temp\_c + "&degC, " + data.weather[0].description + "</p>";

} else {

weatherInfo = "<p>(No weather data available)</p>";

}

// forward on to append news articles

articles(marker, query, weatherInfo);

})

.fail(function(jqXHR, textStatus, errorThrown) {

// log error to browser's console

console.log(errorThrown.toString());

});

}

/\*\*

\* gets JSON of news articles

\*/

function articles(marker, query, weatherInfo)

{

// get articles for query

var parameters = { geo: query };

$.getJSON("articles.php", parameters)

.done(function(data, textStatus, jqXHR) {

// build HTML list of articles for query

var articleList = "";

if (data.length > 0) {

articleList = "<ul>";

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

articleList += "<li><a href='" + data[i].link + "' target=\_blank>" + data[i].title + "</a></li>";

}

articleList += "</ul>";

} else {

articleList = "No news available";

}

//showInfo(marker, articleList);

showInfo(marker, weatherInfo + articleList);

})

.fail(function(jqXHR, textStatus, errorThrown) {

// log error to browser's console

console.log(errorThrown.toString());

});

};

/\*\*

\* Shows info window at marker with content.

\*/

function showInfo(marker, content)

{

// start div

var div = "<div id='info'>";

if (typeof(content) === "undefined")

{

// http://www.ajaxload.info/

div += "<img alt='loading' src='img/ajax-loader.gif'/>";

}

else

{

div += content;

}

// end div

div += "</div>";

// set info window's content

info.setContent(div);

// open info window (if not already open)

info.open(map, marker);

}

/\*\*

\* Updates UI's markers.

\*/

function update()

{

// get map's bounds

var bounds = map.getBounds();

var ne = bounds.getNorthEast();

var sw = bounds.getSouthWest();

// get places within bounds (asynchronously)

var parameters = {

ne: ne.lat() + "," + ne.lng(),

q: $("#q").val(),

sw: sw.lat() + "," + sw.lng()

};

$.getJSON("update.php", parameters)

.done(function(data, textStatus, jqXHR) {

// remove old markers from map

removeMarkers();

// add new markers to map

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

{

addMarker(data[i]);

}

})

.fail(function(jqXHR, textStatus, errorThrown) {

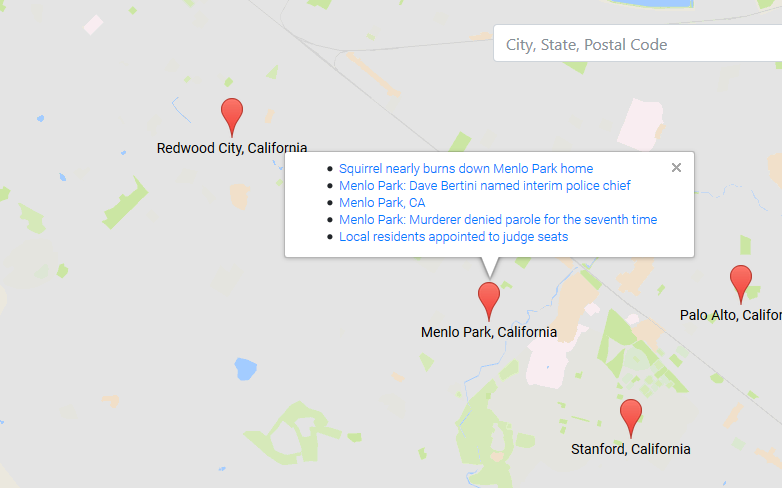
// log error to browser's console

console.log(errorThrown.toString());

});

};

**Приклад виконання програми**



***Висновок:*** завдяки набутим знанням з програмування, ми навчились реалізовувати складні програми та алгоритми.