

Scavenger Hunt User Manual

This document is intended for the use of maintenance and update of the Montgomery County Scavenger Hunt.

Map:

```
1. map = new Map("map", {
2. //you may change the basemap here
3.   basemap: "streets-vector",
4.   zoom: 16,
5.   slider: false
6. });
```

Change or add the map attributes here.

Layer:

```
1. //add layer
2. var agoServiceURL = "http://montgomeryplans.org/arcgis4/rest/services/Overlays/M
   CAtlas_Park_Information/MapServer";
3. var agoLayer = new ArcGISTiledMapServiceLayer(agoServiceURL);
4. map.addLayer(agoLayer);
```

Change the layer or lay URL

Point of interest on the map (one example)

Connect to the database and check if the user has already been to this point of interest.

The query returns one row which means this user has been to this point of interest (point no.1 in this case). Set the color variable to grey.

If it returns not one row, the user has not been to this point of interest. Set the color variable to red.

Create symbol variable.

Set symbol attribute – the information of site 1.

Set geolocation for this symbol.

Add this symbol to the map with its site info, color, shape, and geolocation.

The rest of points of interest marks follow the same logic.

```

1. //adding points of interest on the map, there are 92 of them, the code are repe
   ative with different geolocation info for each point
2. //before adding the points, check if the user has been to this point. If YES, s
   et color of the point to grey, if No, set color of point to red
3. //refer the park info document to see the list of site names, their info and th
   eir ID, the table sites in the database is also help but not comprehensvie
4.
5. //very first point
6.
7.                                     <?php include ('mysqli_connect.php');
8.                                     //get all the points of interest this user has been to
9.                                     $query = "SELECT * FROM sites as tp
10.                                    LEFT JOIN visits as up ON
11.                                    tp.site_id = up.site_id and up.user_id = '$user'
   order by tp.site_id" ;
12.                                    $sql = mysqli_query($dbc, $query);
13.                                    if (!$sql) {
14.                                        printf("Error: %s\n", mysqli_error($dbc));
15.                                        exit();
16.                                    }
17.                                    while($row = mysqli_fetch_array($sql, MYSQLI_ASSOC)) {
18.                                        //check if user's visited points include point "1"
19.                                        if ($row['user_id'] == $user && $row['site_id'] == '
   1'){
20.                                            //if Yes, set the point color to grey
21.                                            $color = array (150, 151, 149, 0.5);
22.                                            $color2 = array(150, 151, 149);
23.                                            break;
24.                                        }
25.                                        //if No, set the point color to red
26.                                        else {
27.                                            $color = array (255, 0 ,0, 0.5);
28.                                            $color2 = array (255, 0 ,0);
29.                                        }}
30.                                        ?>
31.                                    //use function to add point "1" to the map
32.                                    function addPointGraphics(){
33.
34.                                        let color = new Color (<?php echo json_encode($color) ; ?>)
35.                                        let color2 = new Color (<?php echo json_encode($color2) ; ?>)
36.                                        let symbol = new SimpleMarkerSymbol (SimpleMarkerSymbol.STYLE_CIRCL
   E, 9,
37.                                        new SimpleLineSymbol(SimpleLineSymbol.STYLE_SOLID, color,9), color
   2);
38.                                    //set the site info
39.                                    let attr = {"site name": "\"A Dahlia Garden\"", "park name": "Agricu
   ltural History Farm Park", "compass coordinates": "39 9'51\"N 77 8'2\"W", lat:
   -77.132439, long: 39.165291} ;
40.                                    //mark this site to the following geolocation
41.                                    let loc= new Point(-77.132439, 39.165291);
42.
43.                                    let graphic = new Graphic(loc, symbol, attr, infoTemplate);

```

```

44.
45.         map.graphics.add(graphic);
46.
47.     }
48.         map.on("load", addPointGraphics);
49.
50.
51.
52. //end first

```

Question forms

Each site has two questions for users upon arrival.

By default, questions are hidden by CSS class.

When user arrives, the first question pops up.

The users are mandatory to choose an answer out of four choices. If they don't, they receive a hint for correction.

The users have infinite opportunities to find out the right answer. Each wrong attempt will result a hint for correction.

The answer input with value 1 is the right answer.

After passing the first question, the first question is hidden, and the second question pops up.

Some logic applies to the second question regarding the answer verification.

After passing the second question, the second question is hidden as well. All done.

All the questions for the 92 sites follow the same logic.

```

1. <!-- site questions forms, there are 92 of them-->
2.
3.
4. <!-- first site questions-->
5.
6. <!-- get the site questions -->
7. <div id="Q1" class="modal">
8. <div class="modal-content">
9. <!-- dispaly the first question -->
10. <form id="Q1.1" class ='firstquestion'>
11. <h2><strong>You have arrived! Answer the following two questions.</strong></h2>
12. <h2>What material are the flower beds made of?</h2>
13. <p>Question 1/2</p>
14. <!-- check if answer is correst for the first question -->

```

```

15. <p id = 'inputerror1.1'></p>
16. <!-- the right answer's value is 1 -->
17. Wood <input type="radio" name="yesno" value = '1'><br>
18. Brick <input type="radio" name="yesno" value = '2'><br>
19. Concrete <input type="radio" name="yesno" value = '3'><br>
20. Stone <input type="radio" name="yesno" value = '4'><br>
21. <!-- fade the first question and display the second question -->
22. <br><input onclick="return validateForm1q1()" type="button" value="Next">
23. </form>
24. <!-- display the second question -->
25. <form id="Q1.2" class = 'secondquestion'>
26. <h2>When the flowers are in bloom, what are two colors you see?</h2>
27. <p>Question 2/2</p>
28. <!-- check if answer is correct for the second question -->
29. <p id = 'inputerror1.2'></p>
30. <!-- the right answer's value is always 1 -->
31. Any combination of two of the colors <input type="radio" name="yesno2" value =
    '1'><br>
32. Pink <input type="radio" name="yesno2" value = '2'><br>
33. Yellow <input type="radio" name="yesno2" value = '3'><br>
34. Orange <input type="radio" name="yesno2" value = '4'><br>
35. <!--
    fade the question pop window, initiate php and record the user has done site 1 qu
    estion to prevent further pop-up upon revisit -->
36. <br><input onclick="return validateForm1q2();" type="button" value="Submit">
37. </form>
38. </div>
39. </div>
40.
41.

```

User's geolocation and the comparison against all the 92 sites.

Var global variables lat and long to store user's updated geolocation.

Var global variables lat4Decimals and long4Decimals to store user' updated geolocation as 4 decimals numbers without rounding those numbers.

Compare user's geolocation against 92 sites by calling the "compare[site number]" function every 5 seconds. The following only shows 8 of them as demonstration.

```

1.     var lat;
2.     var long;
3.     var lat4Decimals;
4.     var long4Decimals;
5.
6.     window.onload=function(){
7.         getLocation();
8.     }
9.
10.    var x = document.getElementById("demo");
11.    function getLocation() {
12.    if (navigator.geolocation) {
13.        navigator.geolocation.watchPosition(showPosition);
14.    } else {

```

```

15.         x.innerHTML = "Geolocation is not supported by this browser.);}
16.     }
17.
18.     function showPosition(position) {
19.         x.innerHTML="Current Latitude: " + position.coords.latitude +
20.         "<br>Current Longitude: " + position.coords.longitude;
21.         lat = position.coords.latitude;
22.         long = position.coords.longitude;
23.         //trim lat and long down to 4 deciamls without rounding
24.         lat4Decimals = lat.toString().match(/^-\?\d+(?:\.\d{0,4})?/)[0];
25.         long4Decimals = long.toString().match(/^-\?\d+(?:\.\d{0,4})?/)[0];
26.
27.         //run functions that compare user's current geolocaton against all 92 every 5 se
         conds
28.         setInterval(compare1,5000);
29.         setInterval(compare2,5000);
30.         setInterval(compare3,5000);
31.         setInterval(compare4,5000);
32.         setInterval(compare5,5000);
33.         setInterval(compare6,5000);
34.         setInterval(compare7,5000);
35.         setInterval(compare8,5000);

```

compare user's geolocation against site 1's geolocation.

Site 1's geolocation is 39.1652, -76.1324. It is hardcoded into the function.

When matches, insert this instance to the table visits in database by calling the "post[site number]" function.

```

1.     //compare the user's geolcation against site 1's geolocation.
2.     function compare1()
3.
4.
5.     {
6.         if (lat4Decimals == 39.1652 && long4Decimals == -77.1324)
7.
8.         {
9.             //if they match, insert to the database - this user has arrived at site 1
10.            post1();
11.        }
12.    }

```

Calling “insert.php” in AJAX to insert this instance (this user has arrived at site 1).

```
1. //insert to the database using AJAX, get "data" back from the php('form' filed i
n sql), the value is 0. 0 means user has not answer the question form yet.
2.     function post1(){
3.     var userID = "<?php echo $user ?>";
4.     $.ajax({
5.     url: 'insert.php',
6.     type: 'POST',
7.     data:{
8.     userID: userID,
9.     // site 1
10.    placeId: 1
11.    },
12.    success: function(data) {
13.    //take value 0 in the function, the function sees 0, then dispaly the questio
n form for site 1
14.    showme1(data);
15.    },
16.    error: function(XMLHttpRequest, textStatus, errorThrown) {
17.    // handle any network/server errors here
18.    console.log("Status: " + textStatus);
19.    console.log("Error: " + errorThrown);
20.    }
21.    });
22.    }
23.
```

The “insert.php” insets the instance and it also returns the value of the “form” field. By default, the value is 0. The value is stored in variable “check”

This value controls if the user has answered the questions for site 1. By having default 0, it figures the user has no answered, so it will pop-up the questions for site 1.

After the user passing all the questions, the value will be update to 1, so the questions for site 1 will neve pop- up again for this user.

```
1. <?php
2.     include ('mysqli_connect.php');
3.
4.     # Always sanitize input from $_POST variable to prevent SQL injection
5.     $userId = $dbc->escape_string($_POST['userId']); // current_user_id
6.     $placeId = $dbc->escape_string($_POST['placeId']); // the_place_id
7.
8.
9.     $sql = "SELECT user_id, site_id FROM visits where (user_id = '". $userId.'" and
site_id = '". $placeId.'"");
10.    $result1 = @mysqli_query ($dbc,  $sql);
11.
12.    if ( $result1->num_rows == 0 ) {
13.
14.
15.    $query = "INSERT INTO visits (user_id, site_id, date) VALUES ('". $userId.'" ,
'". $placeId.'" , CURRENT_TIMESTAMP())";
```

```

16.
17.
18. $result = @mysqli_query ($dbc, $query); // Run the query.
19.
20.     if ($result) { // If it ran OK.
21.         // Print a message.
22.         echo '<h1 id="mainhead">Success!</h1>';
23.     }
24.     else { // If it did not run OK.
25.         // Print error.
26.         echo '<h1 id="mainhead">Error</h1>';
27.     }
28.
29.
30.
31.
32.         $query2 = "
33.             SELECT form FROM sites as tp
34.             JOIN visits as up ON
35.             tp.site_id = up.site_id where (up.user_id
36. = ' ".$userId." and tp.site_id = ' ".$placeId." )" ;
37.         $sql2 = mysqli_query($dbc, $query2);
38.         if (!$sql2) {
39.             printf("Error: %s\n", mysqli_error($dbc));
40.             exit();
41.         }
42.         while($row = mysqli_fetch_array($sql2, MYSQLI_AS
43. SOC)) {
44.             if ($row['form'] == '0'){
45.                 $check = '0';
46.                 // echo '<h2 id="mainhead">Success f
47. or check 0</h2>';
48.                 break;
49.             }
50.             else {
51.                 $check = '1';
52.                 ;
53.             }
54.             break;}
55.         }
56.
57.         echo $check;
58.
59.
60. ?>

```

If the value "check" is 0, it finds the questions' ID number (Q1 in this case). Then displaying the first question by changing CSS attribute.

```

1. //display question form
2. function showme1(check){
3.
4.   if(parseInt(check) == 0)
5.     {
6.       var modal = document.getElementById('Q1');
7.       modal.style.display = "block";
8.
9.     }
10.
11.    else { var modal = document.getElementById('Q1');
12.          modal.style.display = "none"}
13.
14.

```

check the first question, if the user chooses the right answer (value = 1), then displays the second question.

Check the second question, if the user chooses the right answer, call the function “update[site numer]”.

```

1. //check user's answer for question 1
2. function validateForm1q1() {
3.   var radios = document.getElementsByName("yesno");
4.   var formValid = false;
5.
6.   var selectedVal = "";
7.   var selected = $("input[type='radio']+[name='yesno']:checked");
8.   selectedVal = selected.val();
9.
10.  var i = 0;
11.  while (!formValid && i < radios.length) {
12.    if (radios[i].checked) formValid = true;
13.    i++;
14.  }
15.
16.
17.  if (!formValid)
18.  {
19.
20.    document.getElementById("inputerror1.1").innerHTML = "<span style='color: red;'>
    Must select an option!</span>";
21.    return formValid;
22.  }
23.
24.
25.  else if (selectedVal!= '1')
26.
27.    document.getElementById("inputerror1.1").innerHTML = "<span style='color: red;'>
    Wrong answer, please try again!</span>";

```



```

28. else {
29.
30.     //show question 2 after checking
31.     next_step1();
32. }
33.
34. }
35.
36.
37. //show question 2 after checking
38. function next_step1() {
39. document.getElementById("Q1.1").style.display = "none";
40. document.getElementById("Q1.2").style.display = "block";
41.
42. }
43.
44.
45.
46.
47.
48. //check user's answer for question 2
49. function validateForm1q2() {
50. var radios = document.getElementsByName("yesno2");
51. var formValid = false;
52.
53. var selectedVal = "";
54. var selected = $("input[type='radio']")[name='yesno2']:checked";
55. selectedVal = selected.val();
56.
57. var i = 0;
58. while (!formValid && i < radios.length) {
59.     if (radios[i].checked) formValid = true;
60.     i++;
61. }
62.
63.
64. if (!formValid)
65. {
66.     document.getElementById("inputerror1.2").innerHTML = "<span style='color: red;
67.     >Must select an option!</span>";
68.
69.     return formValid;
70. }
71.
72. else if (selectedVal!= '1')
73. document.getElementById("inputerror1.2").innerHTML = "<span style='color: red;
74. >Wrong answer, please try again!</span>";
75.
76. else {
77.     //update 'from' filed from 0 to 1 in database. User has answered the site
78.     //1 questions, it wont show again
79.     update1();
80. }
81.

```

Call update1.php in update1 function.

```
1.
2. //update 'from' filed from 0 to 1 in database. User has answered the site 1 que
   stions, it wont show again
3.     function update1(){
4.
5.         var userID = "<?php echo $user ?>";
6.
7.         $.ajax({
8.             url: 'update1.php',
9.             type: 'POST',
10.            data:{
11.                userID: userID,
12.                placeId: 1
13.            },
14.            success: function(serverResponse) {
15.                // handle output from server here ('Success!' or 'Error' from PHP scr
ipt)
16.                console.log("Successful");
17.            },
18.            error: function(XMLHttpRequest, textStatus, errorThrown) {
19.                // handle any network/server errors here
20.                console.log("Status: " + textStatus);
21.                console.log("Error: " + errorThrown);
22.            }
23.        });
24.        window.location.reload();
25.    }
26.
27.
28.
29. //end of insert/update site 1 info
```

The Update1.php does:

Update the “form” field value from 0 to 1 for this user and for this site in visits table. So, the pop-up questions won’t be displayed again when this user revisits this site in future.

Update the “site_score” value from 0 to 1 for this user and for this site in visits table.

Check if the user has completed all the sites within each park. If the user has finished all the sites in a park. Update “park_score” value from 0 to 3 for this user and for this park. This update happens only on one of the site visit record that associated to the park.

Check if the user has completed all the sites within each division. If the user has finished all the sites in a division. Update “division_score” value from 0 to 25 for this user and for this division. This update happens only on one of the site visit record that associated to the division.

```

1. //update form file from 0 to 1 for this site. no more questions
2. $query = "UPDATE visits
3. SET form = '1'
4. WHERE user_id = '". $userId.'" and site_id = '". $placeId.'"";
5.
6.
7. $result = @mysqli_query ($dbc, $query);
8.
9.     if ($result) { // If it ran OK.
10.         // Print a message.
11.         echo '<h1 id="mainhead">Success!</h1>';
12.     }
13.     else { // If it did not run OK.
14.         // Print error.
15.         echo '<h1 id="mainhead">Error</h1>';
16.     }
17.
18.
19.
20.
21. // update site piont from 0 to 1 for this site
22. $query2 = "UPDATE visits
23. SET site_point = '1'
24. WHERE user_id = '". $userId.'" and site_id = '". $placeId.'"";
25.
26.
27. $result2 = @mysqli_query ($dbc, $query2);
28.
29.     if ($result2) { // If it ran OK.
30.         // Print a message.
31.         echo '<h1 id="mainhead">Success!</h1>';
32.     }
33.     else { // If it did not run OK.
34.         // Print error.
35.         echo '<h1 id="mainhead">Error</h1>';
36.     }
37.
38.
39.
40. /*         extra points         */
41.
42. /*if the user has finished all the 5 sites in Agricultural History Farm Park, up
date park_point from 0 to 3 on site_id 1, site_id 1 is one of the site in Agricu
ltural History Farm Park*/
43. $query3 = "SELECT COUNT(visits.site_id) from visits join sites on visits.site_id
= sites.site_id where user_id = '". $userId.'" and park_name = 'Agricultural His
tory Farm Park'" ;
44. $result = mysqli_query($dbc, $query3);
45. $row = mysqli_fetch_assoc($result);
46.     if ($row['COUNT(visits.site_id)'] == '5')
47.
48.
49.     { $query4 = "update visits SET park_point = '3' where user_id = '". $userId.'"
' and site_id = '1'";
50.         mysqli_query($dbc, $query4);}
51.
52.
53.

```

```

54.
55. /*if the user has finished all the 49 sites in north division, update division_p
oint from 0 to 25 on site_id 79, site_id 79 is one of the site in north division
*/
56. $query3 = "SELECT COUNT(visits.site_id) from visits join sites on visits.site_id
= sites.site_id where user_id = '". $userId.'" and division_name = 'north' ";/*
division name */
57. $result = mysqli_query($dbc, $query3);
58. $row = mysqli_fetch_assoc($result);
59. if ($row['COUNT(visits.site_id)'] == '49') /* number of sites *
/
60.
61.
62. { $query4 = "update visits SET division_point = '25' where user_id = '". $use
rId.'" and site_id = '79'"; /* site id */
63. mysqli_query($dbc, $query4);}

```

Leaderboard.

The leaderboard.php does:

1. Calculates and displays current user's score.
2. Displays the top 5 scores and the associate users. Users' scores can tie.

Achievements

A visited point of interest will have a checked mark, and a not yet visited points of interest will have an unchecked mark.

```

1. $query = "SELECT site_id FROM visits where user_id = '". $user.'" order by site_i
d" ;
2.
3. $sql = mysqli_query($dbc, $query);
4. $rowcount=mysqli_num_rows($sql);
5. if (!$sql) {
6. printf("Error: %s\n", mysqli_error($dbc));
7. exit();
8. }
9. for ( $i= 1; $row = mysqli_fetch_array($sql, MYSQLI ASSO
C); ++$i)
10. {
11.
12. if ($row['site_id'] == '26')
13. {
14. echo "<span><input checked="" type="checkbox"/></span>"; break;}
15. else if ($i == $rowcount)
16. echo "<span><input type="checkbox"/></span>";
17. }?>
18.

```

Fraction number is used to indicate how many points of interest in a park the user has visited versus the total number of points of interest in that particular park.

```
1. $query = "SELECT count(visits.site_id) FROM visits
2. JOIN sites on sites.site_id = visits.site_id
3. where user_id = '". $user.'" and park_name = 'Agricultural History Farm Park' " ;
4.
5. $query1 = "SELECT count(site_id) FROM sites
6. where park_name = 'Agricultural History Farm Park' " ;
7.
8.
9.
10. $result = mysqli_query($dbc, $query);
11. $row = mysqli_fetch_assoc($result);
12.
13. $result1 = mysqli_query($dbc, $query1);
14. $row1 = mysqli_fetch_assoc($result1);
15. // print ".$row.'/'.'.$row1."
16. print $row['count(visits.site_id)'].' / '.$row1['count(site_id)'];
17.
```

The divisions also have a check or an unchecked mark to indicate user's accomplishment.

```
1. $query = "select
2. count(v.site_id), division_name
3. from sites s join visits v on s.site_id = v.site_id
4. where division_name = 'North' and v.user_id = '". $user.'"
5. group by division_name" ;
6.
7.
8. $result = mysqli_query($dbc, $query);
9. $row = mysqli_fetch_assoc($result);
10.
11. if ($row['count(v.site_id)'] == '49')
12. {
13.     echo "☑";}
14. else
15. { echo "☐";
16. }
```

Overall look.



North Division

5 / 5 Agricultural History Farm Park

"A Dahlia Garden"

Shade Garden

Habitat garden

Antique farm equipment

Pig Pen

5 / 5 Black Hill Regional Park

3 / 3 Darnestown Heritage Park

5 / 5 Dowden's Ordinary Special Park

5 / 5 Germantown Town Center Urban
Park

5 / 5 Little Bennett Regional Park

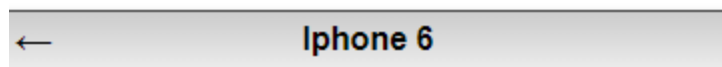
6 / 6 Olney Manor Recreational Park

5 / 5 Rock Creek Regional Park

5 / 5 South Germantown Recreational
Park

5 / 5 Woodlawn Manor Cultural Park

South Division



Rewards

Check if the user has completed all the sites in a park. If yes, print a button for getting the reward. If no, print a disabled button.

```
1.
2. $query = "select
3. count(v.site_id), park_name
4. from sites s join visits v on s.site_id = v.site_id
5. where park_name = 'Black Hill Regional Park' and v.user_id = ".$user."
6. group by park_name" ;
7.
8.
9.         $result = mysqli_query($dbc, $query);
10.        $row = mysqli_fetch_assoc($result);
11.
12.                if ($row['count(v.site_id)'] == '5')
13.                {
14.        echo "<input type='button' name='BH' id='BH' class=\"button\" value='Completed Black Hill Regional Park' />";
15.        echo "<br>";        echo "<br>";
16.
17.                }
18.                else
19.                { echo "<input type='button' disabled name='BH'
20.        id='BH' class=\"button\" value='Completed Black Hill Regional Park' />";
21.                echo "<br>";        echo "<br>";
22.                }
23.
```

The same logic for division rewards and the final reward.



Completed Agricultural History Farm Park

Completed Black Hill Regional Park

Completed Blockhouse Point Park

Completed Cabin John Park

Completed Calverton- Galway Local Park

Completed Darnestown Heritage Park

Completed Dowdens Ordinary Special Park

Completed Ellsworth Urban Park

User

Sign up contains the following features.

Check if all the required fields are filled.

Check if the username already exists.

Check if the Email already exists.

Check if the password contains least 1 number, 1 lower case, 1 capital case and being at least 8 characters long.

Check if two passwords match.

Check phone number format.

Check captcha.

Password retrieval

Check if the user exists

Check if the sent email link for password retrieval is expired – expires in 60 mins.

Check if two passwords match.

Migration:

The website needs to be hosted on a server with an SSL certificate – https is required for geolocation.

The host server needs to have an email address to send password retrieval emails.

Depending on the URL of the server, the password retravel link needs to be modified as well as the email verification URL. They can be located in forgot.php and verify.php.

db_connect.php and Mysqli_connect.php need to be modified for server connection.

The front page is named as index.php.

Document for sites.