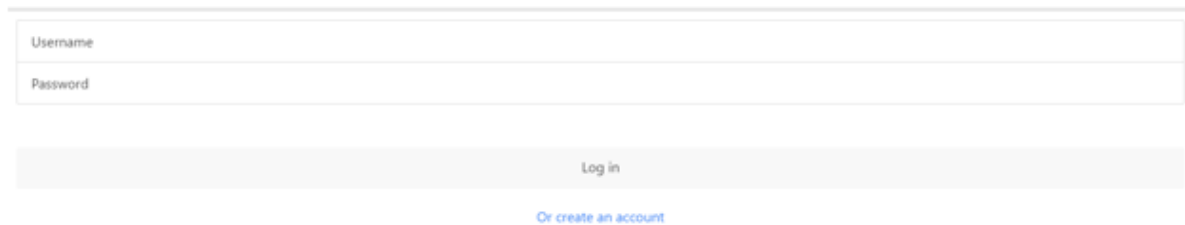


## Wiki page for Lab Asmt 9

1. Create a Private REST service that is Mashup of at least two public REST services.
2. Use this private web service in your web/mobile/hybrid application
3. Try to use a different web service and create some good use case
4. Application should have good UI/UX.

**Please find the screenshots below:**

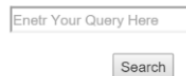
**Log-In Screen:**



A screenshot of a login interface. It features two input fields: the top one is labeled 'Username' and the bottom one is labeled 'Password'. Below these fields is a light gray rectangular button with the text 'Log in'. Underneath the button is a blue hyperlink that reads 'Or create an account'.

**Home Screen:**

**Search for Your Interests**



A screenshot of a search section. It includes a text input field with the placeholder text 'Enter Your Query Here'. Directly below the input field is a small, rectangular button labeled 'Search'.

## Search for Your Interests

### Oppenstein Brothers Memorial Park

**Address:** 12th St & Walnut St  
(undefined), US, Kansas City, MO,  
64106, United States,

**Contact:**

### Park University - Downtown Campus

**Address:** 911 Main St Ste 800  
(undefined), US, Kansas City, MO,  
64105, United States,

**Contact:** (816) 559-5601

## Kansas City Weather Report

Temaparature	35.6° F
Pressure	1017 hPa
Humidity	55 %
Wind	5.82 miles/hour
Description	Clear

### Columbus Square Park

**Address:** E Missouri Ave (at Holmes  
St), US, Kansas City, MO, 64106,  
United States,

**Contact:** (816) 513-7500

### Richard L Berkley Riverfront Park

**Address:** 1298 E Riverfront Dr. (btwn  
Heart of America & Paseo/Bond  
Bridges), US, Kansas City, MO,  
64105, United States,

**Contact:** (816) 221-0636

## Code:

```
client.js x server.js x
1 var http = require('http');
2 var fs = require('fs');
3 var url = require('url');
4 // Create a server
5 http.createServer( function (request, response) {
6   // Parse the request containing file name
7   var pathname = url.parse(request.url).pathname;
8
9   // Print the name of the file for which request is made.
10  console.log("Request for " + pathname + " received.");
11
12  // Read the requested file content from file system
13  fs.readFile(pathname.substr(1), function (err, data) {
14    if (err) {
15      console.log(err);
16      // HTTP Status: 404 : NOT FOUND
17      // Content Type: text/plain
18      response.writeHead(404, {'Content-Type': 'text/html'});
19    } else {
20      //Page found
21      // HTTP Status: 200 : OK
22      // Content Type: text/plain
23      response.writeHead(200, {'Content-Type': 'text/html'});
24
25      // Write the content of the file to response body
26      response.write(data.toString());
27    }
28    // Send the response body
29    response.end();
30  });
31 }
```