Assignment #3
CSCI 201 Spring 2019
4.5% of course grade

Title
Weathermeister Back-End API Integration

Topics Covered

Java Classes	HTML	CSS
Basic Java Topics	Java Servlets	JSP
JavaScript	AJAX	Databases
SQL	JDBC	

Overview
Assignment 3 builds off the previous web-based assignment, WeatherMeister. This assignment focuses more on the backend portion of your website, and you will need to use MySQL, Google Maps API, and OpenWeatherMap API.

OpenWeatherMap API & Google Maps API
Instead of reading in data through a hard-coded text file, you will use the OpenWeatherMap API to retrieve weather data. In other words, the weather data should now be pulled from this API, and you will not have to read in data from a text file. You can learn more about the API here: https://openweathermap.org/api. You can learn more about the retrieved data at this site: https://openweathermap.org/current. The data will be returned in a JSON or XML format.

Use the Google Maps API to generate a map overview when searching by latitude and longitude. You can learn more about the API here: https://developers.google.com/maps/documentation/javascript/tutorial. You will need this to create the overlay displayed in Figure 3.

You will have to generate API keys for both sites, so make sure you do this early. Validation can take up to a few hours, so *do not start this assignment on the last day*.

Parsing JSON
The data from the API is going to be returned as a JavaScript Object Notation (JSON) file. JSON is a lightweight data-interchange format. In other words, it is a syntax that allows for easy storage and organization of data. It is commonly used to exchange information between client and server, and it is popular because of its language independence and human readability. JSON is built upon two basic data structures that you should already be familiar with: dictionaries (maps) and ordered lists. An object in JSON is represented by an unordered set of name/value pairs (i.e. dictionary). Objects are contained by braces, { }, inside of which will list the object's attributes (with the syntax `name : value`), using a comma as the separator.

There are quite a few Java JSON parsing APIs out there. Unfortunately, the JDK does not have built-in support for JSON, but some of the more popular ones include GSON, Jackson, and JSON.simple. Here are a couple of blogs discussing the merits of choosing one of these APIs over another if you are interested:

http://blog.takipi.com/the-ultimate-json-library-json-simple-vs-gson-vs-jackson-vs-json/
http://javarevisited.blogspot.com/2016/09/top-5-json-library-in-java-JEE.html

In short, it seems GSON is known for its ease and flexibility of converting Java objects into JSON objects (and vice versa), and it is simple and straightforward to use. You may want to start there. No matter which API you choose (and you are not limited to choosing from the ones mentioned on these instructions), you will need to download a JAR file and add it to your Eclipse project. Below are links to the JAR file download for the APIs mentioned above:

GSON: https://mvnrepository.com/artifact/com.google.code.gson/gson/2.8.0 (under 'Files', click download JAR)
Jackson: https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-core/2.2.3 (under 'Files', click download JAR)
JSON.simple: https://mvnrepository.com/artifact/com.googlecode.json-simple/json-simple/1.1.1 (under 'Files', click download BUNDLE)

You will need to add the JAR file to your Java Build Path in Eclipse to use the library. First, move the JAR file to the top directory of your Eclipse project. In other words, if your project is named `Assignment3`, put the JAR file into the `Assignment3` directory in your Eclipse workspace directory. Next perform the following steps in Eclipse:

1. Right click on your Eclipse project.
2. Click "Refresh", which should make your JAR file show in the Package Explorer.
3. Right click on your Eclipse project again.
4. Choose "Properties".
5. Select "Java Build Path".
6. Click the "Libraries" tab.
7. Click "Add JARs".
8. Find the JAR in your project directory and add it.
9. Click "Okay".

MySQL

In this assignment, you will also track user data. You should construct a simple MySQL database that stores usernames, passwords, search queries for each user, and a timestamp of when the search occurred. You will need to display this data on the Profile Page for each user when logged in (see Figure 10).

Updated Images

All images needed for this assignment have been posted on the Assignments page of the web site.

Figure 1 Home Page (city)

Figure 2 Home Page (lat/long)

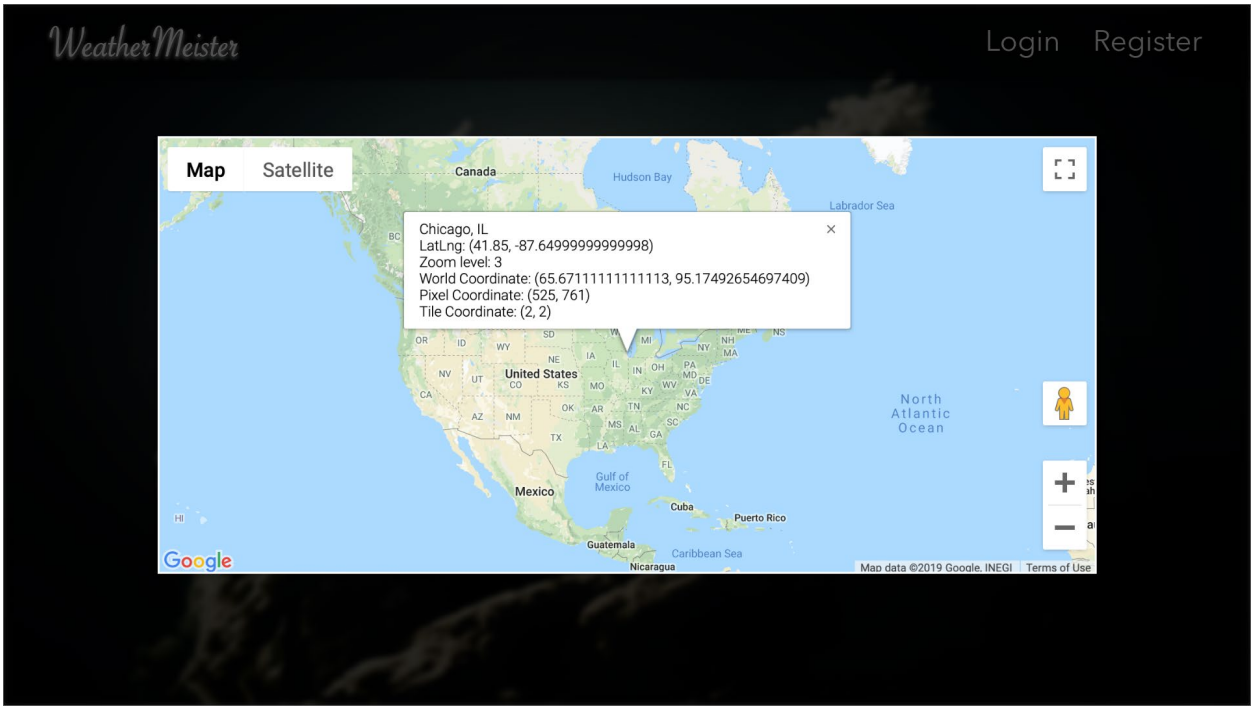

Figure 3 Home Page (Google Maps)

Home Page

In addition to the functionality from Assignment 2, there should now be "Login" and "Register" links in the top right corner of the page (Figure 1). Clicking "Login" will direct the user to the login page (Figure 4). Clicking "Register" will direct the user to the signup page (Figure 5).

If the user clicks the “Location” radio button, the page will create a Google Maps icon to the right of the search bar (Figure 2). The user can click on that icon, which will create a Google Maps overlay for the page (Figure 3). You can do this by creating three new tags: one for the map with a percentage sizing in the center of the viewport, another for the background, and one more to contain the other two. Clicking on the map should make the map disappear, returning the user to Figure 2, and auto-populate the “latitude” and “longitude” fields with the chosen latitude and longitude.

Note: The “Display All” button should be removed.

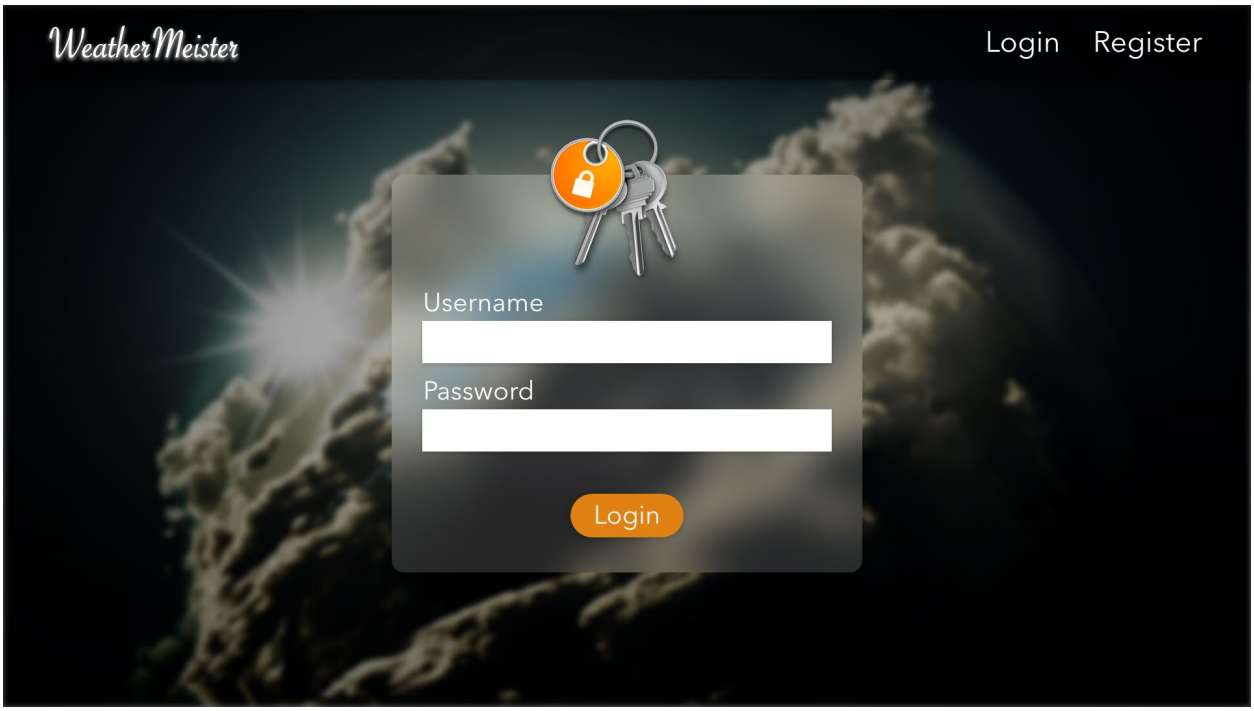

Figure 4 Login Page

Login Page

Once the user arrives to this page, the user can login with a pre-existing account. There are three different scenarios to account for:

- Wrong username: *This user does not exist.*
- Correct username, wrong password: *Incorrect password.*
- Correct username, correct password: *Successful login!*

Once the user enters the correct username/password information, redirect the user back to the home page (see Figure 6). Otherwise, keep the user on the login page, and display the error message between the password field and the “Login” button.

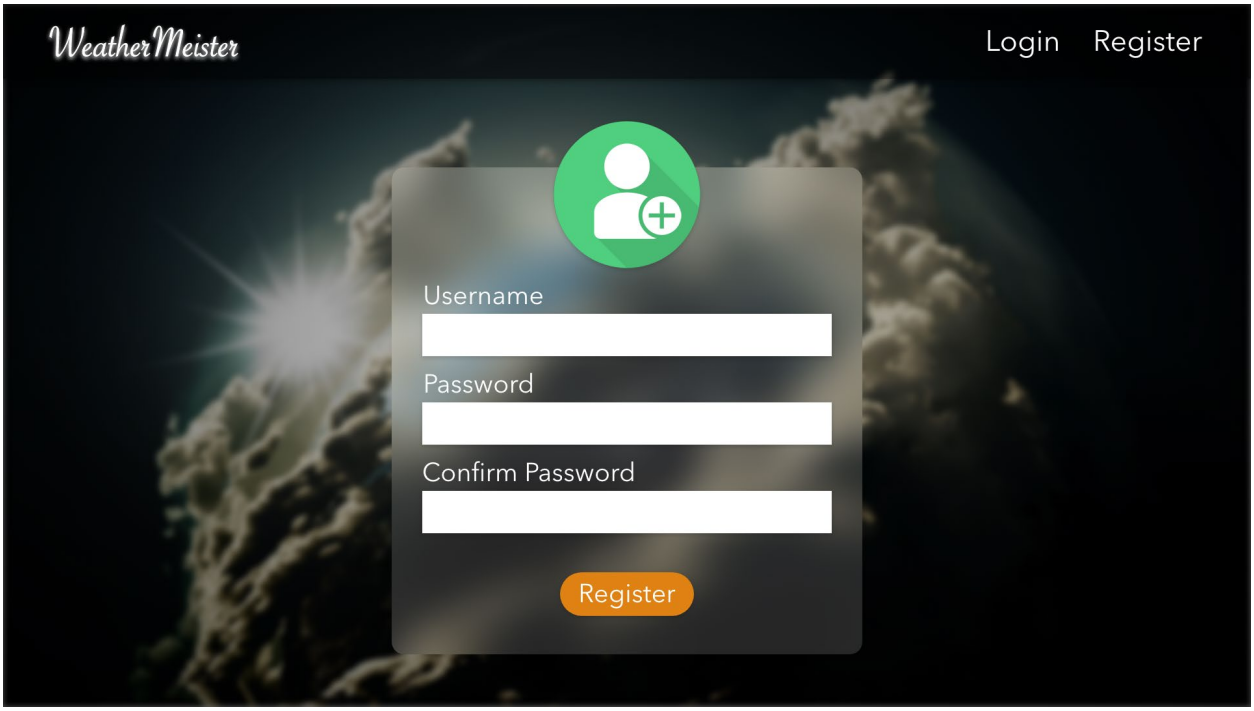

Figure 5 Register Page

Register Page

When the user clicks "Register", he will be directed to this page. The user can sign up for a new account on this page. There are three different scenarios to account for:

- Username already taken: *This username is already taken.*
- Passwords do not match: *The passwords do not match.*
- Username is unique, and passwords match: *Successfully created a new account!*

Once the user enters the correct username/password information, redirect the user back to the home page (see Figure 6). The user is automatically logged into this new account. If the data is not valid, leave the user on the register page. Display the error message between the "Confirm Password" field and the "Register" button.

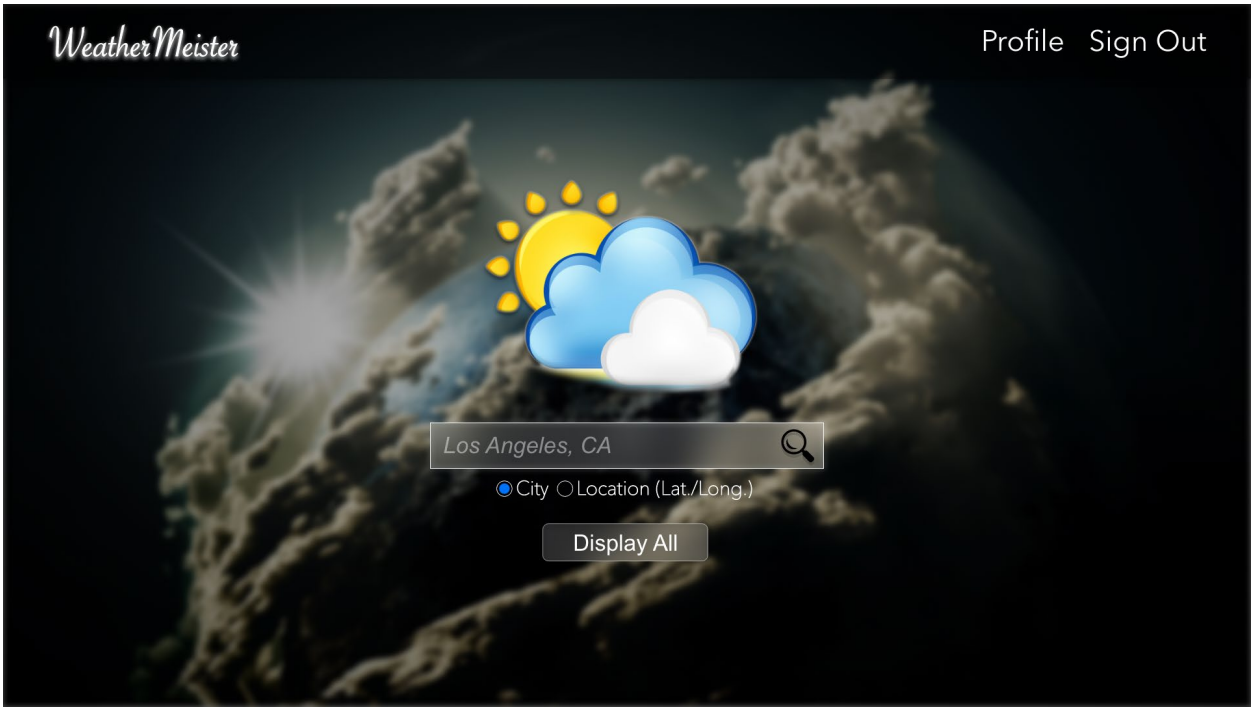

Figure 6 Home Page (after logging in)

Home Page (pt. 2)

Upon successfully logging into an account or successfully creating a new account, the "Login" and "Register" buttons should be replaced with "Profile" and "Sign Out", respectively. Clicking on the "Profile" button redirects the user to the profile page (see Figure 10). Clicking "Sign Out" will log out the user and redirect him to the home page (see Figure 1).

If the user performs a search while logged in, you should add this data into the MySQL table. If a search is completed while *not* logged in, do not track this data. Make sure to get the timestamp too. This process should be repeated whenever a search is completed from any page of the site.

Note: The "Display All" button should be removed.

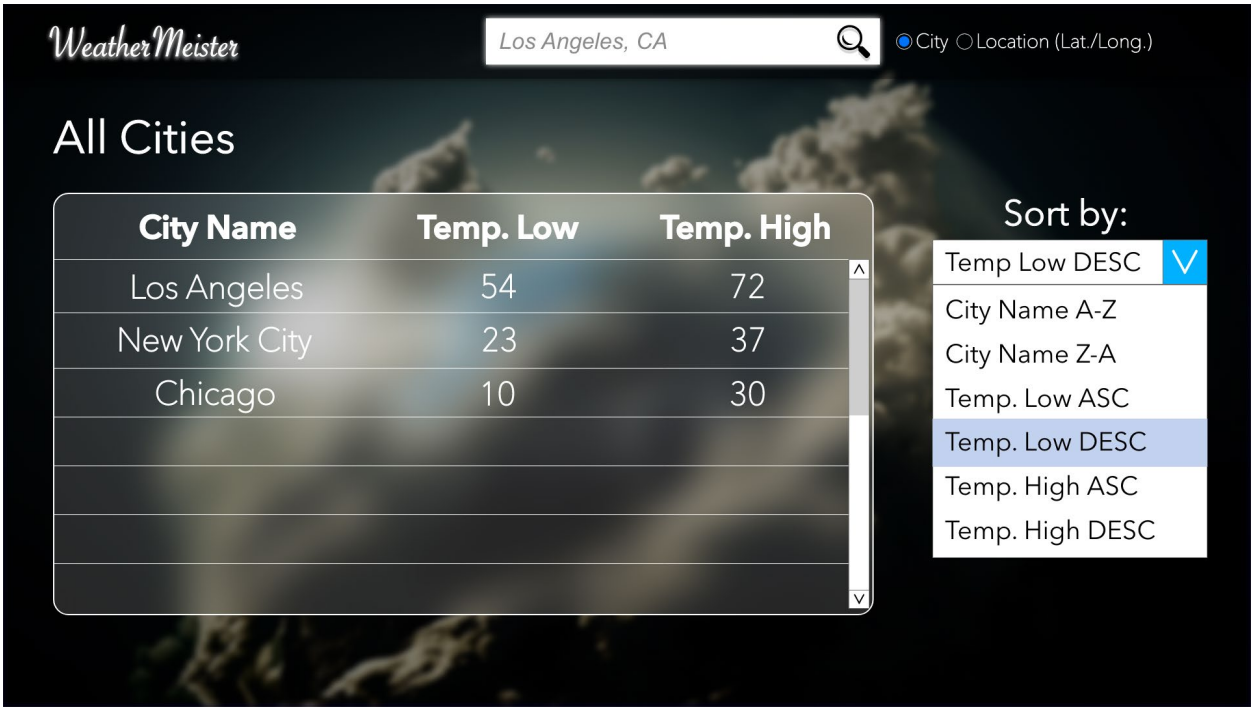

Figure 7 Search Results Page

Search Results Page

The search bar should be updated with the new Google Maps functionality. Clicking on "Location" should create a Google Maps icon to the right of the radio buttons (see Figure 9 for an example) and clicking on the icon will create the map overlay identical to Figure 3. Once again, this will populate the latitude and longitude fields once the user clicks on a location on the map. All other functionality remains the same as before.

If more than one city matches the search query (i.e. "Springfield" or "Washington"), the OpenWeatherMapAPI will return data for all cities with the same name. You should use the table to display data for all matching cities.

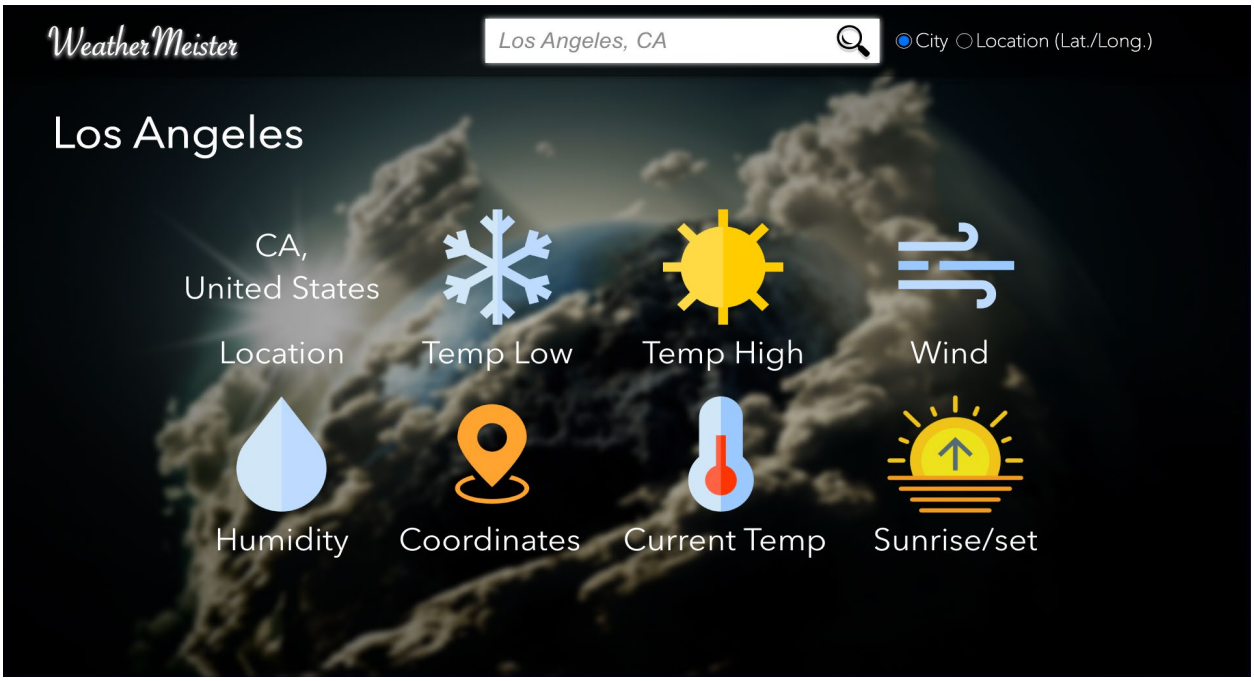

Figure 8 Details Page (city)

Figure 9 Details Page (Location)

Details Page

The search bar should be updated with the new functionality, as demonstrated in Figure 8 and Figure 9. Clicking on the Google Maps icon will create the map overlay as shown in Figure 3. All other functionality remains the same as before.

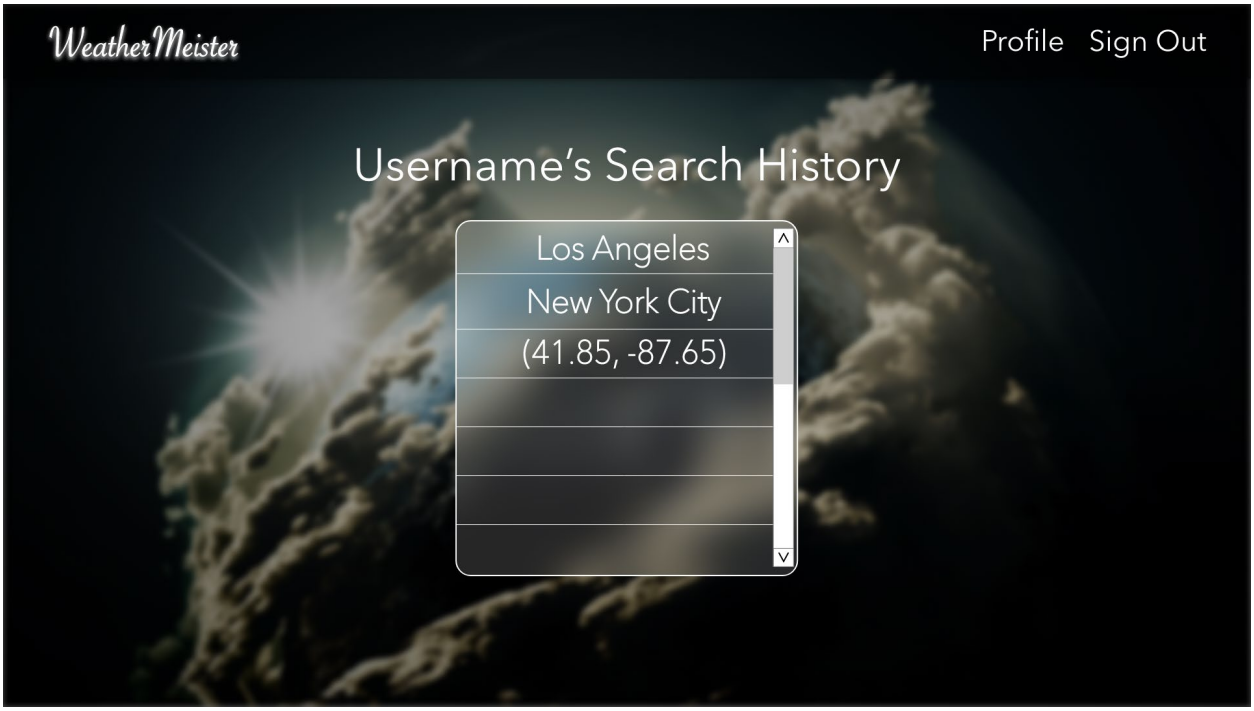

Figure 10 Profile Page

Profile Page

When the user clicks on "Profile," redirect the user to this page. This page will display the user's search history. Replace the "Username" with the actual username of the currently logged-in user. In the list below, display all of the searches made by the user. The searches are ordered from newest (at the top) to oldest (at the bottom).

Grading Criteria

The output must match as close as possible to the screenshots provided above.

User Login Functionality (0.9%)

0.1% - Login and Register links are on the home page and register page if a user is not logged in
0.2% - Profile and Sign Out links are on the home page, register page, and profile page if a user is logged in
0.1% - Login page looks like the screenshot
0.1% - Register page looks like the screenshot
0.2% - Error messages displayed properly on the Login page
0.2% - Error messages displayed properly on the Register page

Home Page (0.8%)

0.1% - When latitude/longitude radio button is clicked, the Google Maps icon is displayed
0.3% - Clicking the Google Maps icon displays the Google Map in the center of the page
0.4% - Clicking on the Google Map makes the map disappear and populates the latitude/longitude fields properly

Search Results Page (1.6%)
0.5% - Searching by city returns proper results
0.5% - Searching by latitude/longitude returns proper results
0.3% - Sorting functionality works properly
0.1% - clicking on city name (and not on temperatures) links to details page
0.1% - If no city matches search, the table and drop down menu are not shown, but an appropriate message is displayed.
0.1% - If only one city matches search, “All Cities” is not shown and drop down menu is not shown.

Details Page (0.5%)
0.5% - Details page shows proper live data

Top Search Bar (0.1%)
0.1% - When latitude/longitude radio button is clicked, the Google Maps icon is displayed

Profile Page (0.6%)
0.2% - Profile page shows all the searches performed by a user
0.1% - Profile page shows user’s name
0.2% - Results are sorted from newest to oldest
0.1% - Profile page looks like the screenshot