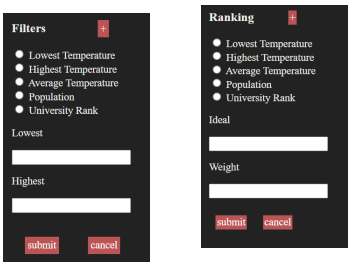
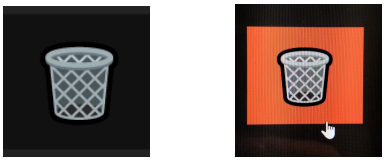
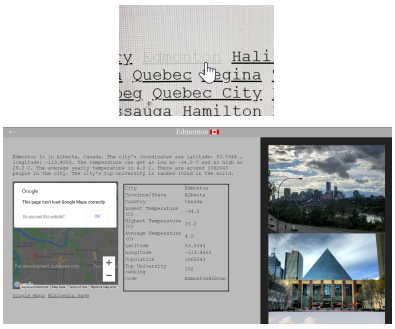
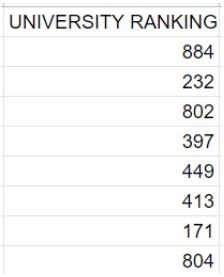
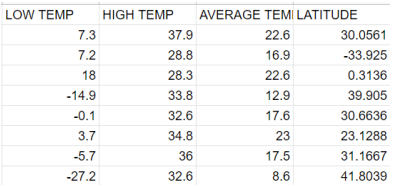
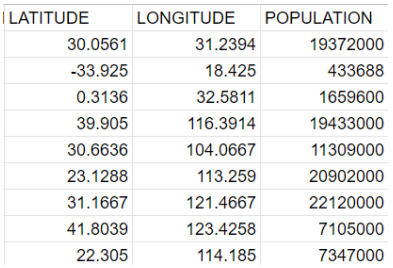
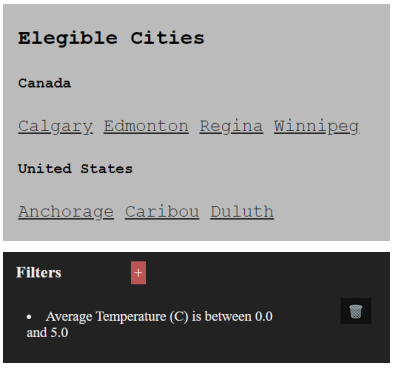
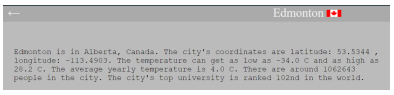
Criterion E: Evaluation

Success Criteria

An input form which allows the user to add filters/ranking criteria on the list of cities		The submit buttons of these forms consistently send the user to the correct page. There is also error checking for values that will not work or do not make sense. This success criteria has been met!
An input which allows the user to remove a filter/ranking criteria on the list of cities		The button consistently sends the user to the correct page to remove the item. When a user hovers over the button it changes to red as to increase the button's intuitivity. This success criteria has been met!
An input which allows the user to be sent to an information page for a specific city		When each city in the list is clicked the user is consistently redirected to it's information page. The page adapts to the amount of information wikipedia and the city database has on it so that there are no errors. The photos, flag, and map appear in the page along with the database information to give the user a more complete understanding of the city. This success criteria has been met!
A function that retrieves information from a university ranking database and adds it to a master city database		The program consistently adds the university rankings from the university rankings database to the full city database. This success criteria has been met!
A function that retrieves information from a global city weather database and adds it to a master city database		The program consistently processes the climate information from the climate database and adds it to the full city database. This success criteria has been met!

A function that retrieves information from a population and coordinates database and adds it to a master city database		The program consistently adds the population, longitude, and latitude from the population and coordinates database to the full city database. This success criteria has been met!
The cities are properly ranked using the ranking criteria		When the criterias are received the program consistently ranks and displays them correctly. Makers separating groups of cities are used to make the rankings easier to understand. This success criteria has been met!
ineligible cities are filtered out using the user imputed filters		When the filters are received the program consistently filters out ineligible cities and displays them correctly. This success criteria has been met!
A function that returns a paragraph about a specific city		The city information page consistently displays an accurate description of the city using the data in the city database. This success criteria has been met!

Recommendations for Further Development

The database is incomplete: Many cities are only mentioned in one of the databases which may exclude cities from rankings and fiiters that they should have been eligible or high ranking in. To fix this the program should source the city information from a more comprehensive database.

Important information is not included in the database: The database is missing a lot of important information such as the spoken language of the city or the presence of a democracy. These were not mentioned explicitly by the client but it can be inferred as important. To solve this a sample population could be queried on what they care about for a city and databases for that information should have been added.

The program is not made easily modular so it is difficult to add new features: Ideally adding new city information should be as simple as dragging a new database into the database folder. The reality of this project is that a new function must be written per each database, the input for each type of information must be individually written, and the names of each type of information must be defined in several areas of several scripts. To significantly improve this website the databases should not be hard coded and should instead use generalized functions that are applied to all databases.

Word Count: 220 (excluding tables and titles)