Ziyue Guo
Robert Dumitrescu
Sarah Lundell
Kevin Xu
Tongxin Zhu
Group number: 4

Project Milestone 5

Features to Test:

1) Entering a string location that is not a place for city search, latitude, and longitude.
2) Entering invalid signup data
3) Subscribe/Unsubscribe from the email list

How they are tested:

1) A key feature in our project is the search feature so users can find the weather at locations they are interested in. To do this, users can look up places by city name search field and out latitude/longitude search fields. If a user enters a location that is invalid, spells a city name wrong, or enters a latitude/longitude that is not supported by our map, we need to make sure they get an error message.
   To test this we will have to be sure to check any enterable value that may be invalid.
   Expected input data:
   City search - city name starting with a capital, spelled correctly.
   Latitude/longitude search - number values that correspond to latitudes/longitudes that appear within the state of Colorado.
   Input data values to test:
   City search - misspelled city name (ex: Buolder), non-capital letter starting input (ex: boulder), invalid input with numbers or other special characters (ex: 1,8493, #$*()!@#), blank input.
   Latitude/Longitude search - latitude/longitude values not in range or invalid, non-digit input, more than one value entered (ex: 103 402 [with the space])
   Currently, if a user enters an invalid input to any of these fields nothing happens in the view of the user. The API query is invalid so no data is returned. We need to have it fail loudly so the user knows that their input was not valid. Alerts should pop up telling the user what they did wrong. To test the functionality of the search features we manually input the data (described above) and noted the outcome.
2) On the user sign up page we want to make sure that the user enters valid information to our giving fields. If we get an invalid email and then try to send something to that address it could be an issue. Likewise, if a user enters a city that they would want information about and it is not in our database we will have problems so we need to make sure the data is valid. Currently, if you input a value that is invalid for a given field you are unable to submit the form as there is a regex checking for what type of input we require for each input.
   To test we need to input values into each field that test all possible cases:

Expected input data:
Email - standard email format (including @ and .)
Cities - a city name starting with a capital letter
Password - is a length of 8 characters (since we are using a hash function to store)
Input data values to test:
Email - value without @ and ., also just an invalid email to see what the email sender will try to do, and we should test a blank input
Cities - value that is not in our database (city not in CO or misspelled), city name without capital first letter (ex: boulder), blank input, a word with non letter characters
Password - try word lengths that are shorter and longer than the needed character length, try different characters to make sure the hash function maps correctly.

No major issues are caused by invalid inputs. The regex function checks all input fields so the user cannot submit the form unless the values are correctly formatted. For the cities field, a user can input an invalid city format, however, the API will not break from an invalid value, it will just not form a query.

3) We have a checkbox in the user signup where they can choose to receive emails about locations that they are interested in. They can also choose to unsubscribe later if they would like. To test this feature we need to make a new user and they subscribe to the email feature and make sure they begin to receive emails. Some edge cases to go along this are if they did not enter any cities to get emails about or if they entered more than one. After we validate that this is working we need to try to unsubscribe them and make sure that they stop receiving emails. We should also make sure that the user can subscribe after they signup so we should make an account this is not subscribed and they go back in and subscribe them later.