

WEB222 Final Assessment – Winter 2021

Overview

The Final Assessment has **three parts** and builds on your final assignment. Specifically, you will be asked to do the following:

1. Research and Implement an HTML Search Form (worth 5%)
2. Research and Implement an HTML Data Entry form with Custom Validation (worth 10%)
3. Research and Implement a Static Hosting solution for your final code, so that it can be viewed online (worth 5%)

Due Date

The Final Assessment is due on **Monday April 19th by midnight**. Please see Submission instructions at the end of this document. You **may** consult your notes and use the web, but you **may not** discuss any part of this final assessment with other students. All work must be your own.

Part 1 and 2: HTML Forms and Form Validation

You are asked to modify the solution to Assignment 5 in order to provide additional features. Specifically, you are asked to add two HTML Forms:

1. A Search Form
2. A New Observation Data Entry Form

Use the code in the ZIP file provided, which has TODO comments indicating where to do each of the necessary steps discussed below.

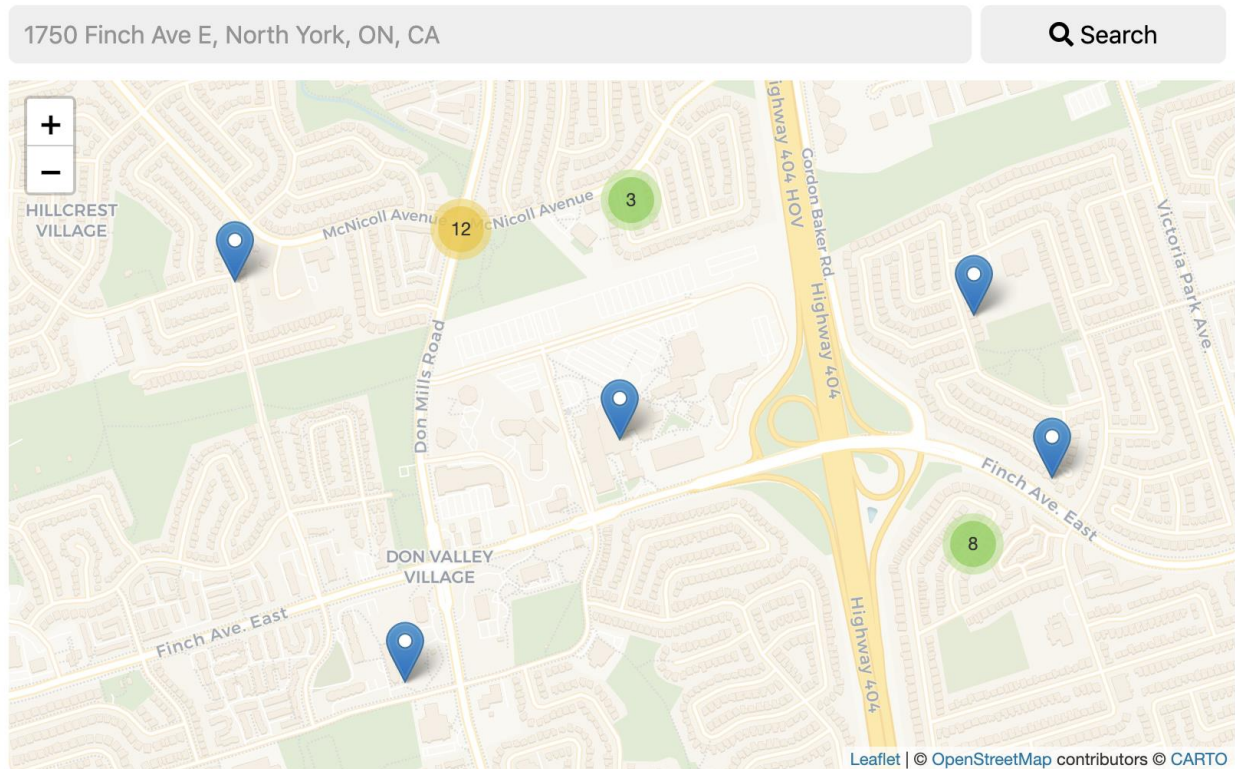
1.1 Search Form (5%)

The Search Form allows a user to enter an address and search for observation data on iNaturalist. For example, a user can enter a street address, or be more general and use a city name. All of the following are valid searches:

- 1750 Finch Ave E, North York, ON, CA
- 70 The Pond Rd, North York, ON, CA
- Oakville, ON, CA
- Paris, France
- Melbourne, Australia

Your Search Form should be placed at the top of the map, and look like this:

Map

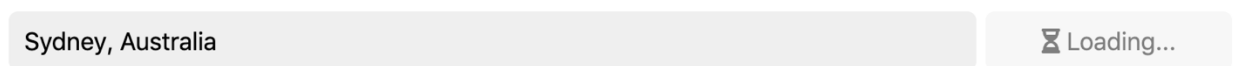


Modify the following files in order to create the form:

- **index.html**: finish the markup for the form (see the TODO comment)
- **js/app.js**: when the form is submitted, you need to get the text that the user has entered in the form field, and use it to invoke the **search()** function, which will download and display the results. NOTE: **search()** is already defined for you in **src/search.js**, you only need to call it.
- **css/style.css**: add any CSS styles you need in order to make the form look like the image above.

When the **search()** function is called, it will use the **toggleLoading()** function in **js/ui.js** to alter the Search button, changing the icon and text, and disabling the button:

Map



Implement the **toggleLoading()** function in **js/ui.js** with the following logic:

- When `isLoading` is **true**:
 - o change the icon to use the Font Awesome hourglass-half icon (see <https://fontawesome.com/icons/hourglass-half?style=solid>)
 - o change the text to “Loading...”
 - o disable the button, so the user can’t click it again
- When `isLoading` is **false**:
 - o change the icon to use the Font Awesome search icon (see <https://fontawesome.com/icons/search?style=solid>)
 - o change the text to “Search”
 - o re-enable the button, so the user can click it

When your page is first loaded, you should show the “Search” version by default.

1.2 Add New Observation Form (10%)

The second form you need to add allows a user to submit a new observation.

*NOTE: we **won’t** be submitting real data to iNaturalist, only simulating it using <https://formspree.io>, which allows developers to collect responses from user-submitted forms. You need to sign-up for a free account so you can receive form submissions. See <https://help.formspree.io/hc/en-us/articles/360053239754-Getting-started-with-projects>.*

Your form should submit to <https://formspree.io/> using you’re formspree account, so you get an email with the form contents. This will help you know if your code is working.

Your new form should be created in the **add-observation.html** file, and you can put any CSS you need in the **css/form.css** file. You will also use the **js/validate.js** file in order to write some custom, client-side form validation.

Here is what your form should look like (NOTE: this screenshot was taken in Chrome on macOS, so your browser might look different, which is fine). Pay attention to the styling and order, etc:

Observation Details

* Required field

Observation

Species Guess*

Great Horned Owl

Description*

Roosting on fence post at the edge of a field

Date Observed*

yyyy-mm-dd



Notes

Notes about observation...

Location

Latitude*

43.7955

Longitude*

-79.3496



Share location information (i.e., it's ok to make this public)

Photo

Photographer Name

Karen Shu

Photographer Email

kshu@photos.com

Photo

Choose File

No file chosen

License ([help on choosing a license](#))

Creative Commons Attribution License



Submit

Form Markup

Here are all the details about how to build this form in HTML markup:

1. Use HTML **label** elements to define each form field. Make sure you associate them with the control they go with using the **for** attribute.
2. All **required** fields should have the appropriate HTML attribute set, and use a red * beside their label. Use HTML and CSS to do this.
3. Grouped sections of the form should use HTML **fieldset** and **legend** elements.
4. Use appropriate **placeholder** attributes on all field elements that can use them.
5. Use the most appropriate HTML form element type for each of the fields:
 - a. **Species Guess**: required text field, must be 200 or fewer characters in length
 - b. **Description**: required text field
 - c. **Date Observed**: required date field
 - d. **Notes**: optional free-form text field. Show 3 lines of text.
 - e. **Latitude**: required text field (see custom validation details below)
 - f. **Longitude**: required text field (see custom validation details below)
 - g. **Share Location**: checkbox field
 - h. **Photographer Name**: optional text field, must be 100 or fewer characters in length
 - i. **Photographer Email**: optional email field
 - j. **Photo**: optional file upload field
 - k. **License**: drop-down menu of options for Creative Commons Licenses (<https://creativecommons.org/choose/>). The possible values are:

License Code	License Name
CC-BY	Creative Commons Attribution License
CC-BY-NC	Creative Commons Attribution-NonCommercial License
CC-BY-SA	Creative Commons Attribution-ShareAlike License
CC-BY-ND	Creative Commons Attribution-NoDerivs License
CC-BY-NC-SA	Creative Commons Attribution-NonCommercial-ShareAlike License
CC-BY-NC-ND	Creative Commons Attribution-NonCommercial-NoDerivs License

Custom Validation

In addition to the standard HTML form validation, you should also create **custom JavaScript form validation** for the **Latitude** and **Longitude** values.

If a user tries to submit a form with invalid Longitude or Latitude values, you should stop the form from being submitted and use HTML and CSS to display an error message beside the label:

Latitude* must be a valid Latitude (-90 to 90)

abc

Longitude* must be a valid Longitude (-180 to 180)

-181.135

Your JavaScript validation should be written in the **js/validate.js** file (see the `validate()` function). You should validate the form according to these rules:

- **Latitude:** must be a valid number between -90 and 90
- **Longitude:** must be a valid number between -180 and 180

Part 2. Static Hosting (5%)

You are asked to research and implement a static hosting solution for your completed assignment. Your final code should be accessible via a public URL. You do not need to spend any money to achieve this, since many free hosting services exist:

1. Vercel <https://vercel.com/>
2. Netlify - <https://www.netlify.com/>
3. GitHub Pages - <https://pages.github.com/>
4. Firebase - <https://firebase.google.com/docs/hosting>

Please submit the public URL for your project. All pages, images, etc. must work and not return 404s or other errors.

Submission

Please run the **npm run prepare-submission** script and upload your **submission.zip** file to Blackboard. Please also **include the URL** to the publicly hosted version of your project.