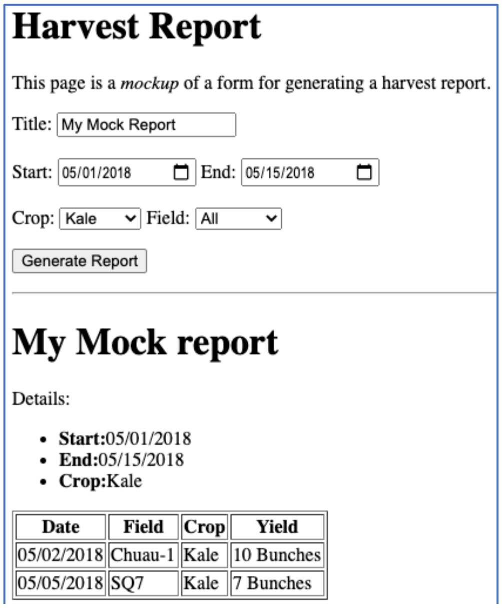
FarmData2 Onboarding Activity 03
HTML Technology Spike

Introduction:

HTML (Hypertext Markup Language) is the fundamental technologies used to structure forms and reports in the FarmData2 front-end. These activities introduce you to just enough essential HTML to get you on the path to doing FarmData2 development. They will also familiarize you with the resources and references that you can return to later to learn more as you need to. Future activities will provide similar introductions to just enough of the other key technologies necessary for FarmData2 front-end development. All of these activities will be done in the context of FarmData2 and will provide seeds for the features that you eventually work on.

Harvest Report Mockup

To learn and practice with HTML you will build a mockup of a Harvest Report function for FarmData2. It won't have all of the features of an actual Harvest Report, it won't actually work yet, and it won't look very pretty. We'll add to this form as we go and eventually make it almost fully functional. The page you build in these activities will look something like this:



Harvest Report

This page is a *mockup* of a form for generating a harvest report.

Title: My Mock Report

Start: 05/01/2018 End: 05/15/2018

Crop: Kale Field: All

My Mock report

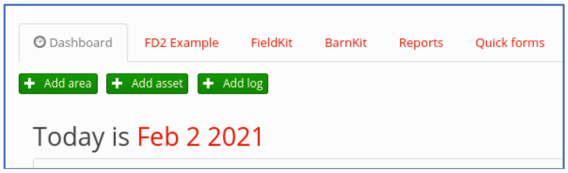
Details:

- **Start:**05/01/2018
- **End:**05/15/2018
- **Crop:**Kale

Date	Field	Crop	Yield
05/02/2018	Chuau-1	Kale	10 Bunches
05/05/2018	SQ7	Kale	7 Bunches

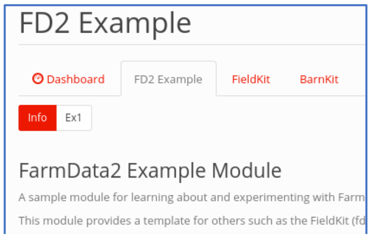
Adding a Tab to FarmData2:

When you launch FarmData2 after installation you should see three tabs that are FarmData2 specific: *FD2 Example*, *FieldKit* and *BarnKit* as shown below.



Today is Feb 2 2021

The FieldKit and BarnKit tabs are where the majority of FarmData2 features will live. The FD2 Example tab provides a playground for trying out new things and serves as an example of how to make new tabs and sub-tabs. Initially, this tab contains two sub-tabs: Info and Ex1.



FarmData2 Example Module

A sample module for learning about and experimenting with Farm

This module provides a template for others such as the FieldKit (fd

So, the first thing we'll do is add an additional sub-tab to the FD2 Example tab to give you a place to build your Harvest Report mockup.

1. Create a feature branch from your FarmData2 main branch. Give your new branch the name `<name>-Spikes` where `<name>` is your name. Also be sure to checkout your new branch. You will do your work for this activity in this feature branch.

2. Connect a browser tab to the Theia IDE at `localhost:3000` and use the "open workspace" option to open the FarmData2 folder.

3. Find the `README.md` file in the `FarmData2/farmdata2_modules/fd2_tabs/fd2_example` directory. You can open this in the GitHub repository or in the Theia IDE. If you open it in Theia, you'll see the markdown source code. To see it rendered as a document click the tiny M↓ icon on the right.

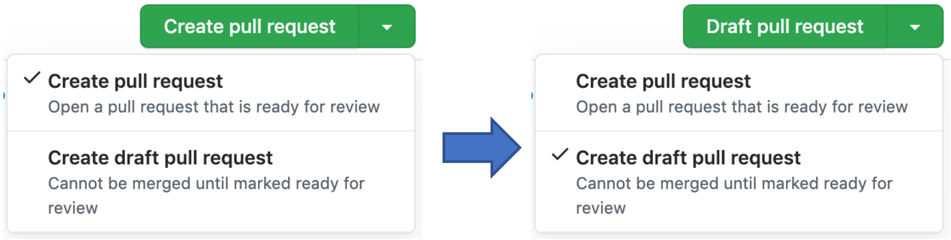
4. Read the first paragraph of this document just for context. Skip over the "FarmData2 Module Structure" section. Then follow steps 1-5 in the "Adding a New Sub-Tab to a FarmData2 Module" section to add a new sub-tab to the `fd2_example` module. Your new tab should have the name "A02" and the contents of the tab should be contained in a file named `a02.html`. Include your name in the content of the `a02.html` file. Paste a screen shot of the FD2 Example tab with your A02 sub-tab open below:

5. You'll be adding to your new sub-tab as you go through these activities. As you finish particular questions you'll be asked to make a commit to your repository with a commit message about the question. This is one of those key points. Stage (i.e. add) and commit to your feature branch the changes you have made to create your sub-tab. **Please include a commit message that clearly indicates what you have done.** You can use the `–m` flag on your

`git commit` command. Or for longer messages it will be more convenient to omit the `-m`, which will cause the nano editor to open for you to type your commit message.

6. Push your feature branch to your origin.

7. On GitHub create a *Draft Pull Request* for your branch to the upstream FarmData2 repository. You create a Draft Pull Request just like you create a normal Pull Request, but use the drop down on the green “Create pull request” button and choose “Create draft pull request” as shown below:


Create pull request
✓ Create pull request
Open a pull request that is ready for review
Create draft pull request
Cannot be merged until marked ready for review
Draft pull request
Create pull request
Open a pull request that is ready for review
✓ Create draft pull request
Cannot be merged until marked ready for review

A Draft Pull Request is just like a pull request but is designated as not being ready to be merged into the `main` branch. Since the work you do in these activities is for learning the technologies and will not become part of FarmData2 using a Draft Pull Request here makes good sense.

Learning Some HTML:

The basic structure of all of the FarmData2 specific pages is specified using HTML. These first activities will introduce you to the basics of HTML and how it is used to define the elements that make up the content on a page.

Use the Mozilla Developer Network (MDN) HTML Basics guide linked below to complete the following activities:

- https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics

You should read through the guide understanding the contents of each section. There are a few questions below drawn from this guide that are included to ensure that you get the key vocabulary and concepts. Once you have those, you’ll add some content to your new A02 sub-tab.

8. Consider the following *strikethrough* (`<s>`) *HTML element*, which is similar to the paragraph (`<p>`) element example in the guide except that the text will be ~~rendered with a strikethrough~~:

```
<s>This text has a strikethrough</s>
```

Name the three main parts of an HTML element and identify each in the above element.

9. Consider the following HTML element

```
<p draggable=”true”>this paragraph is draggable</p>
```

a. What is the *name* of the attribute on this paragraph element?

b. What is the *value* of the attribute you identified in b?

10. Write HTML for a well-formed strikethrough element that has the content “My Old Book Title” and two attributes named *class* and *id*, with the values *revised* and *book-title* respectively.

11. Give properly nested HTML code for a strikethrough element that is strongly emphasized and is part of a paragraph.

12. Rearrange the tags in your previous answer so that they are not properly nested.

Starting the Harvest Report Spike:

13. Let’s put the content of the HTML basics guide to use… Complete the following tasks inside of the <body> tags of the `a02.html` file that appears in your A02 sub-tab. Be sure that the HTML source code that you write is nicely formatted (e.g. indent it similar to what is shown in the HTML basics guide).

a. Add the “Harvest Report” heading.

b. Add the paragraph that starts “This page is a” (ignore the italics for now).

c. Add the “My Mock Report” heading.
d. Add the “Details:” including the bulleted list and bold face type.

14. Stage and commit to your feature branch the changes you have made to your Harvest Report mockup. **Please include a commit message that clearly indicates what you have done.** Then push your feature branch to your origin.

Note that pushing a branch for which a pull request has been made will automatically update the pull request. Thus, the pull request at the upstream will now contain two commits, one for the new sub-tab and one for the start of the Harvest Report.

More Basic HTML:

HTML defines many elements besides those that were described in the guide. You can find a complete reference to all of the HTML tags in the HTML elements reference:

- https://developer.mozilla.org/en-US/docs/Web/HTML/Element

15. Use the HTML elements reference to answer the following questions. Write your answers in your own words.

a. What does the <br> tag do?

b. What does the <hr> tag do? Hint: The description here is not very helpful. Clicking on the tag name will take you to a page that describes that tag and gives examples.

c. What tag should be used to offset idiomatic or technical terms on a page? How is text tagged in this way typically displayed? Hint: Try using the browser’s search function to search the reference page to find the appropriate tag. Then click thorough to its page and read more about it.

Adding to the Harvest Report Spike 1:

16. Improve your Harvest Report page by:
a. Making the term *mockup* into italics.

b. Adding the horizontal line between the Harvest Report Form at the top and the actual report at the bottom.
c. Adding line breaks at appropriate places to space out the content.

17. Stage and commit to your feature branch the changes you have made to your Harvest Report mockup. **Please include a commit message that clearly indicates what you have done.** Then push your feature branch to your origin.

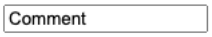
HTML Form Elements:

FarmData2 will need to accept user input in a variety of situations. When requesting a harvest report the user will need to enter things such as the date range and the crop or the field (or both) to generate the report. When performing a harvest, the user will have to enter information including the field and crop being harvested and the quantity that was harvested. HTML form elements provide the mechanisms by which users provide these types of inputs.

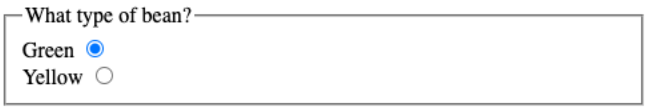
18. Use the MDN Basic native form controls guide linked below to complete the following activities:

- https://developer.mozilla.org/en-US/docs/Learn/Forms/Basic_native_form_controls

a. What HTML code would you add to your page to produce the following form element? You can test it in your sub-tab to be sure it works, but you should then remove it.



b. What HTML code would you add to your page to produce the following set of radio buttons? Ensure that only one of the buttons can be selected at a time. You can test it in your sub-tab to be sure it works, but you should then remove it.


What type of bean?
Green ●
Yellow ○

19. Notice that the text field does not have a label the way that each radio button does. Using the radio buttons as a model, give the HTML to add the label “Comment:” to your text field above. Note that this approach can also be used to label most form elements. You can test it in your sub-tab to be sure it works, but you should then remove it. Hint: Figure out what attribute connects the label to the element!

Adding to the Harvest Report Spike 2:

20. Improve your Harvest Report page by:
    a. Add the text field for the title with the "Title" label.
    b. Add the "Generate Report" button as an anonymous button.

21. Stage and commit to your feature branch the changes you have made to your Harvest Report mockup. **Please include a commit message that clearly indicates what you have done.** Then push your feature branch to your origin.

Additional Input Types:

The `<input>` form element is very versatile and what type of form element it displays is controlled by its `type` attribute. The `<input>`: The Input (Form Input) element reference linked below provides a convenient list of all of the different type attributes that can be applied to the `<input>` element:

- https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input

22. Use the `<input>` element reference to answer the following questions:

    a. What `type` of `input` element should be used for entering numeric values? Why would it be better to use that type instead of just a `text` type?

    b. What `type` of `input` element should be used for the start and end dates of the Harvest Report?

    c. Note that not all input types are supported by all browsers. You can click the `type` name in the reference to get a page with information and examples for that `type`. At the bottom of that page there is a "Browser Compatibility" table. What desktop browsers do not support the input type that you found for part b?

d. Do you think FarmData2 should be concerned about this compatibility issue?

Additional Form Elements:

In addition to the basic native form controls provided by the `<input>` element there are many additional form elements available in HTML. The MDN Other form controls guide provides an introduction to a number of these additional HTML form elements.

- https://developer.mozilla.org/en-US/docs/Learn/Forms/Other_form_controls

23. What HTML element would be used to create the Crop and Field options?

Adding to the Harvest Report Spike 3:

24. Improve your Harvest Report page by:

a. Adding the "Crop" element with "Broccoli", "Kale" and "Peas" as the possible crops. Make "Kale" selected by default. Use a label to make "Crop:" appear next to this element.

b. Adding the "Field" element with "All", "Chuau-1" and "SQ7" as the possible fields. Make "All" selected by default and use a label to make "Field:" appear next to this element.

c. Add the elements for the Start and End date using the `date` input type. The start date should range from 01/01/2014 to 01/01/2022 and have the selected date of 05/01/2018. The end date should range from 05/01/2018 (i.e. not before the start date) up to 01/01/2022 and have the selected date of 05/15/2018.

25. Stage and commit to your feature branch the changes you have made to your Harvest Report mockup. **Please include a commit message that clearly indicates what you have done.** Then push your feature branch to your origin.

HTML Tables:

Many of the reports that are generated by FarmData2 will contain tables of data. For example, in a harvest report there will be one row for each harvesting of a particular vegetable during a time period. Each row will show information about that harvest (e.g. date, field, quantity, worker, etc…). Similarly, a report on a particular animal will show one row for each health check that that animal has had, with each row showing information about that health check. This section introduces you to the HTML elements that are useful for creating tables.

Use the MDN HTML table basics guide linked below to complete the following activities:

- https://developer.mozilla.org/en-US/docs/Learn/HTML/Tables/Basics

You should skim quickly down through the guide until you reach the *Active learning: Creating your first table* section. The details for how to create tables will be in the *Active learning: Creating your first table* section and the Adding headers with `<th>` elements section. You can stop before the Allowing cells to span multiple rows and columns. If the need arises to learn how to do that, we will come back to it later.

26. Answer the following questions based on the HTML table basics guide:

a. What HTML start and end tags surrounds all of the table content?

b. What HTML start and end tags surround all of the content of a row in the table?

c. What HTML start and end tags are used to surround the content of each column in a row?

d. The headers for a table are contained in the first row but require different tags. What HTML start and end tags are used to surround the header for each column in the table?

Adding to the Harvest Report Spike 4:

27. Improve your Harvest Report page by adding the HTML table. You may populate the table with the data shown or other data of your invention. Hint: You can add the attribute `border=1` to the `table` tag to get the borders. This is not technically the "right" way to do this, but we'll improve on it later.

28. Stage and commit to your feature branch the changes you have made to your Harvest Report mockup. **Please include a commit message that clearly indicates what you have done.** Then push your feature branch to your origin.