**Local Foods Connection**

Albert Ford, Hongyuan Zhang, Martin Chamberlin, Priyanka Dangol, Yuya Kawakami

***Overview***

Our project is to develop a website for Local Foods Connection, a non-profit organization in Grinnell that works with local farmers to provide fresh, locally sourced food and provides information about events like farmers markets to Grinnell residents. With the current build of the website, we have a data table containing farms in Grinnell, a map page that shows the locations of farms and events, an events page that displays event information (linked to Laurel’s Google Calendar), and a user system that allows farmers to sign up and sign in.

***Major Parts of the Project + Gems***

Home page: Display all farms’ information in a collection of farm cards. Farms are searchable via the search bar.

Individual farm pages: Shows information (name, picture, description, address, phone number, etc.) of a farm.

Events page: Links with Laurel’s Google Calendar to populate the page with events she has on her calendar. We also distinguish between recurring and non-recurring events so we display them in separate columns.

Map page: Displays a map of Grinnell and markers corresponding to farms or events. It also has an interactive sidebar that displays information about the marker clicked.

Sign up page: Allows farmers to sign up for a farm on the website.

Sign in page: Allows registered farm owners to sign in and edit their own farm’s information; also allows admin to sign in, approve farms, edit farm information, delete farms, etc.

How these fit together: A Grinnell resident (not a farm owner) can browse local farms on the home page, and if interested in a particular farm, he/she can search for it and go to that farm’s individual page by clicking on that farm’s card on the home page. If he/she is interested in local farmer events, he/she can go to the ‘Events’ page. If he/she is interested in locations of local farms and farmer events, he/she can check the ‘Map’ page. For a farm owner, if not a registered user, he/she can sign up by going to the ‘Sign Up’ page and filling out the farm form. Once registered, farm owners can edit farm information by signing in via the ‘Sign In’ Page. Laurel can approve and edit farm information by signing in via the same page.

Gems: The main gems we used are ActiveAdmin & Devise for user system, Pundit for authorization, Rspec & Capybara for testing, and Google API Client Libraries for accessing Google Calendar events. More gems and documentation can be found in the Gemfile.

***Workflow***

* What worked

We met up with our alum mentor every week through Google Hangouts. This was very useful for us as we could share the screen with him and show him the exact code instead of just telling him about the problems we were facing. This allowed him to give specific code feedback, and he was usually able to answer most of our problems during our 45-minute Hangout sessions. We also met up with Laurel (our community partner) at least once every sprint which assured us that we were on the correct trajectory regarding her expectations for the project. During class time, we worked together as a group and also practiced pair programming. We kept each other updated about progress through facebook group, slack, and a group documentation doc.

* What didn’t

At the beginning we split up the work into individual tasks, which sometimes worked, but sometimes it was easy to get stuck on a problem and not have other perspectives to help solve the problem. Problems tended to be less daunting when we were helping each other and pulling our knowledge together to find solutions.

***Remaining Work***

* Layout for Maps Page

The sidebar that pops up after you click on the icons on the map page could be improved in design. A good CSS Grid resource is W3schools tutorial.

* Testing

We mainly used RSpec (a testing framework) and Capybara for our testing, so for the next team we would highly recommend familiarizing yourself with these two libraries before testing new features. The ‘/spec’ folder is the folder where most of the testing for our project is and while there are tests in the test folder, those are the ones that are automatically generated by Rails itself and are not that helpful. Running rspec on the command line can help you give a better idea of what to test as it shows the coverage for the main files and gives the exact lines for which tests need to be written. The tests are also organized by the features of the website instead of the actual names of the files. Here is the link to a Capybara cheat-sheet that might help you:

<https://gist.github.com/zhengjia/428105> and an RSpec cheat-sheet:

<https://www.anchor.com.au/wp-content/uploads/rspec_cheatsheet_attributed.pdf>

* Separating admin sign in page from farmer sign in page

With the current setup, we do not have separate sign in pages for the farmers and the admins. They both use the same sign in page provided by ActiveAdmin. This raises issues since farmer and admins will then both have access to admin tools which isn’t ideal. There should be a individual sign in portal for farmers, without using ActiveAdmin, that, once logged in, redirects them to a farm edit page where farmers can edit their own farms. Our alumni mentor Alex told us that, ideally, the farmer sign in page should not use ActiveAdmin since they aren’t admins.

* Putting photos on website with AWS

Currently, photos of farmers that users see when they open the individual farm pages are pulled from the original local foods connection page. Future work can be done on migrating this functionality to AWS. If done, users, when sign up, can upload a photo of their choice that would be shown on the website.

***Retrospective***

* Accomplishments

We are proud to have created an entirely new functionality, the calendar for local farmers’ events. We encountered a lot of difficulties (e.g. use gem or Google Calendar, display events as a list or on a calendar, Google Calendar API Key, differentiate between recurring and non-recurring events, parse and reformat information pulled from Google Calendar API, etc.) along the way, but teamwork made it work!

* Tools and Learning Resources

Alumni mentors are really awesome! They are experienced and very willing to help you. Consulting your alum mentor is a time efficient solution to possibly most of your problems. Another helpful resource is the web. Hartl’s tutorial, stack overflow, youtube videos are all helpful.

* Success Tips

The only way to learn a new codebase is to start working in it. Things will be confusing at first, especially with Rails and Javascript being new to us, but as you work on each problem you will figure things out as you need to by googling and figuring things out on your own. That said, don’t mess things up if you don’t understand them. Also don’t be afraid to learn legacy code and entirely new programming languages/frameworks.