Food Safety Site

Final Report

CS 4624: Multimedia, Hypertext, and Information Access

Virginia Tech

Blacksburg, Virginia 24061

5/12/2020

Instructor: Dr. Edward A. Fox

Client: Minh Duong

Team members: Zachary Gehr, Kiel Lew, Mahek Mehta, Prateek Mishra, Tim Nguyen, Chanaka Perera

Table of Contents

Abstract

The goal of our project is to provide a resource to empower farmers market vendors to incorporate safe food handling practices. Our client, Minh Duong, would like to target farmers market growers and vendors who do not fall under the Produce Safety Rule or Preventive Controls for Human Food Rule due to their farm size and annual revenue. Our mobile-friendly web application provides our target audience with a customizable experience with easily accessible and quick information. For example, if a farmer has livestock and grows tomatoes, the information and resources accessible on the website will be tailor-made to growing tomatoes and keeping livestock. We ensure this by providing a survey during account creation, which then determines which information to show based on the survey's results. We aimed to deliver on our project objectives by implementing a mobile-friendly application using a MERN stack, which comprises MongoDB, Express, React and Node.js. The website fmfoodsafety.cs.vt.edu contains a home page, FAQ page, recalls page, survey page, admin functionalities, and the ability for users to log in to save their data. Sourcing for these pages was directly from the client, or the Food and Drug Administration and Center for Disease Control websites. Testing the site occurred over a 10 day span involving both the team and the client. We were able to receive valuable feedback from the client throughout our implementation with weekly demos. We also used Postman to verify that the backend was implemented correctly.

List of Figures

List of Tables

Introduction

Farmers markets perform a vital role in connecting communities to their food system. These markets provide farmers the opportunity to sell fresh, local food to consumers and for those consumers to develop more awareness about where their food comes from. As such, it is important that proper procedures, as outlined by the Produce Safety Rule and Preventive Controls for Human Food Rule, are followed, to ensure that appropriate precautions are taken to minimize produce contamination. Some farms and vendors do not fall under either of these rules due to their size and revenue; however, it remains important that they take similar safety measures to limit risks.

Partnering with our client Minh Duong, a Doctoral Student in Food Science and Technology, our goal was to develop a website that can be used by vendors not covered by the aforementioned rules, to obtain information about best safety practices for their farm. Farmers will be able to enter custom characteristics for their farm and will be given tailored information that relates to those attributes. For example, a farmer could enter that they grow lettuce and tomatoes, have farm animals, and use well water. In return, they will be given specific procedures to follow pertaining to those characteristics. This relationship can be seen in the model provided by our client, shown in Figure 1.

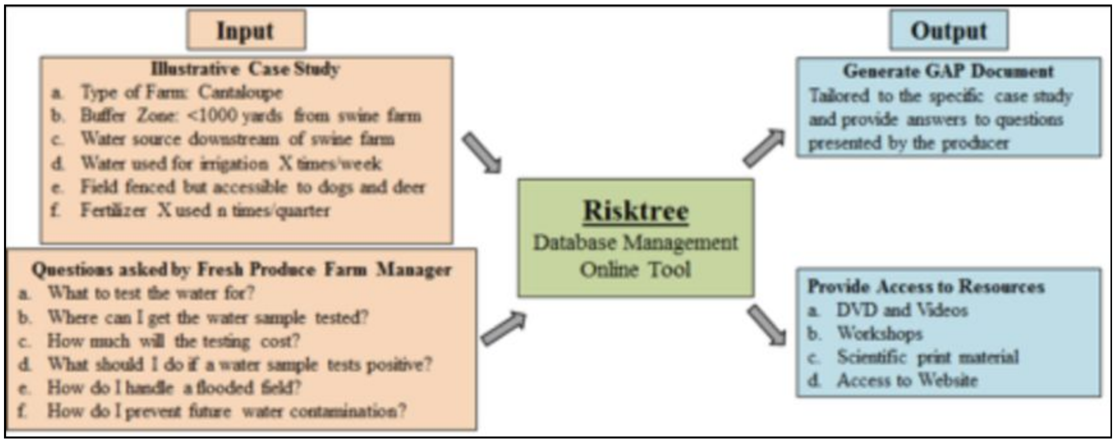

Input
Illustrative Case Study
a. Type of Farm: Cantaloupe
b. Buffer Zone: <1000 yards from swine farm
c. Water source downstream of swine farm
d. Water used for irrigation X times/week
e. Field fenced but accessible to dogs and deer
f. Fertilizer X used n times/quarter
Questions asked by Fresh Produce Farm Manager
a. What to test the water for?
b. Where can I get the water sample tested?
c. How much will the testing cost?
d. What should I do if a water sample tests positive?
e. How do I handle a flooded field?
f. How do I prevent future water contamination?
Risktree
Database Management Online Tool
Output
Generate GAP Document
Tailored to the specific case study and provide answers to questions presented by the producer
Provide Access to Resources
a. DVD and Videos
b. Workshops
c. Scientific print material
d. Access to Website

Figure 1: Input-output model

Requirements

Our client, Minh Duong, had many requirements for our website. First and foremost, he wanted the website to be mobile friendly and easy to interface with so that the website could provide information quickly to farmers. It was also extremely important to make the website customizable for each farmer. This means that based on certain characteristics of the farm, the type of crops grown, the type of livestock on the farm, and the size of the farm for example, different information and resources will be presented to the user. Another feature that Minh wanted was the ability to have admin privileges. These privileges will allow any user with admin access to add different resources and information to the database. We were also tasked with providing a live feed of recent food recalls in the United States. Further, Minh wanted to develop opt-in services such as text or email alerts that provide reminders, and new resource/information alerts depending on the farm's characteristics, as well as provide a way for Extension Agents or Food Safety Virginia Tech Faculty to connect and engage with users.

Design

Frontend design

We used a combination of the following languages:

- HTML
- CSS
- Javascript

The main component of the front end design is react. React is a JavaScript library for building user interfaces. It is maintained by Facebook and a community of individual developers and companies.

Since one of the major requirements is to make the website mobile friendly, we are accomplishing this task through the use of CSS to change the HTML components in the desktop design so that they are displayed suitably for mobile users.

Database design with MongoDB

MongoDB can be integrated easily with node JS and react JS and is a good design choice for our website. MongoDB is a cross-platform document-oriented database program. MongoDB uses JSON-like documents with schema. It is supported by MongoDB Inc. and licensed under the Server Side Public License (SSPL). The biggest difference between MongoDB and more conventional database management software such as SQL is that it uses collections rather than tables. Collections are analogous to tables in relational databases. If a collection does not exist, MongoDB creates the collection when you first store data for that collection. The models for our database can be found under models in the src directory. These models define the structure of entities used in our backend.

For our current project our schemas have all been defined under **models in the root directory**. An example is:

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;

const foodSchema = new Schema({
   name: {
      type: String,
      unique: true
   },
        foodType: {
            type: String
        },
        references: {
            type: [String]
        },
    });

    module.exports = Food = mongoose.model('foods', foodSchema);
```
As you can see from the above example it is extremely easy to define the nature of what the data would look like within the database. It is also possible to add constraints such as unique and add foreign keys using mongoDB.

The required CRUD operations to operate the backend can be found under **routes/api.** An example of the functions used for CRUD operations with explanations is:

First we need to import the following modules:
```
const express = require('express');
const router = express.Router();
```
The model for the entity needs to be imported at the beginning of the file:
```
const Food = require('../../models/Food');

//Import routes
//All routes in this file start as /food
//Look at server.js base route is set there

//Get all foods in the database
```
This function is run when a web administrator makes a get food request from the URL: base_url+'/food'. It will return a list of all available foods in the database:
```
router.get('/', async (req, res) => {
    try {
        const foods = await Food.find();
        res.json(foods);
    } catch(err) {
        res.json({message: err});
    }
});

//Add new food item to database
```

```
//Requires a JSON file input
//ex: {"name": 'Tomatoes', "type": 'produce'}
router.post('/add', async (req,res) => {
    const foodItem = new Food({
        name: req.body.name,
        foodType: req.body.foodType,
        references: req.body.references
    });
    try {
    const savedFoodItem = await foodItem.save();
    res.json(savedFoodItem);
    } catch(err) {
        res.json({message: err});
    }
});
```

This function is similar to the previous function. However instead of getting the list of all foods in our database, we filter the results by the name of the food. When a user makes a get request through ‘base_url’+/api/food/name/:foodname, ‘foodname’ is passed as a parameter for querying the results. This parameter can be extracted using req.params.foodName

```
//Get specific food item
router.get('/name/:foodName', async (req, res) => {
    try {
        const food = await Food.find({name : req.params.foodName});
        res.json(food);
    }
    catch(err) {
        res.json({message: err});
    }
});


//Get specific food item by ID
router.get('/id/:foodID', async (req, res) => {
    try {
        const food = await Food.findById(req.params.foodID);
        res.json(food);
    }
    catch(err) {
```

```
    res.json({message: err});
  }
});

//Remove a specific food item
router.delete('/delete/:foodName', async (req, res) => {
  try {
    const removedFood =  await Food.remove({name : req.params.foodName});
    res.json(removedFood);
  }
  catch (err) {
    res.json({ message: err });
  }
});
```

This function is used to update an existing entry within our database. The parameter for the name of the food is passed in a way similar to the previous function. However instead of running ‘get’ we use ‘update’ within the body of the function.

```
//Update a specific food item
//Only the type of food can be updated since food name is unique
//Requires JSON input {"type" : "produce"}
router.patch('/update/:foodName', async (req, res) => {
  try {
    const updatedFood =  await Food.update({name : req.params.foodName},
     {
      $set: {
       foodType: req.body.foodType,
       references: req.body.references
      }
     });
    res.json(updatedFood);
  }
  catch (err) {
    res.json({ message: err });
  }
});

module.exports = router;
```

The API makes use of JSON input to query the backend and returns an output as a JSON file. It is also possible to use the initial HTTP request to get the needed input parameters for creating a query.

router.get('/name/:foodName', async (req, res)

The above example shows how to pass an input parameter to the GET request through the URL. This is only recommended for GET requests. For POST requests please use JSON body like:

foodType: req.body.foodType

The ER diagram in Figure 2 represents our backend.

We have 5 schemas with fields as follow:

1. User:
    a. **ID:** Auto generated field that acts as the primary key
    b. **Admin**: Boolean field that indicates whether the user is an administrator or not
    c. **Username**: Unique char field that is used to store the name of a user
    d. **Password:** Char field that is used to authenticate a user during login.
    e. **First Name:** Non-required field that is used to customize user profiles
    f. **Last Name:** Non-required field that is used to customize user profiles
    g. **Farm Name:** Non-required field that is used to customize user profiles
2. Food:
    a. **ID:** Unique integer field that is used to store the ID of a food item
    b. **References:** String field that points to a link that contains the relevant safety information regarding a particular food item that is provided to us by the client
    c. **Food Type:** Char field with the type of the food item
    d. **Food Name:** Char field with the name of the food item
3. Psurvey:
    a. **ID:** Unique primary key that points to the ID of this survey table
    b. **Rarely raw:** Boolean field extracted from survey provided by the user
    c. **Raw commodity**: Boolean field extracted from survey provided by the user
4. PSR Survey:
    a. **ID**: Unique integer field that is used to store the ID of a piece of information
    b. **Grows produce:** Boolean field extracted from survey provided by the user
    c. **Less than 25k:** Boolean field extracted from survey provided by the user
    d. **Less than 500K:** Boolean field extracted from survey provided by the user
    e. **Reduce pathogens**: Boolean field extracted from survey provided by the user
    f. **Commodity rare raw**: Boolean field extracted from survey provided by the user
5. PCR Survey:
    a. **id**: Unique integer primary key for table

b. **manHumanConsumption:** Boolean field extracted from the survey provided by the user
c. **packHumanConsumption:** Boolean field extracted from the survey provided by the user
d. **processFood:** Boolean field extracted from the survey provided by the user
e. **packOffFarm:** Boolean field extracted from the survey provided by the user
f. **majority Famers:** Boolean field extracted from the survey provided by the user

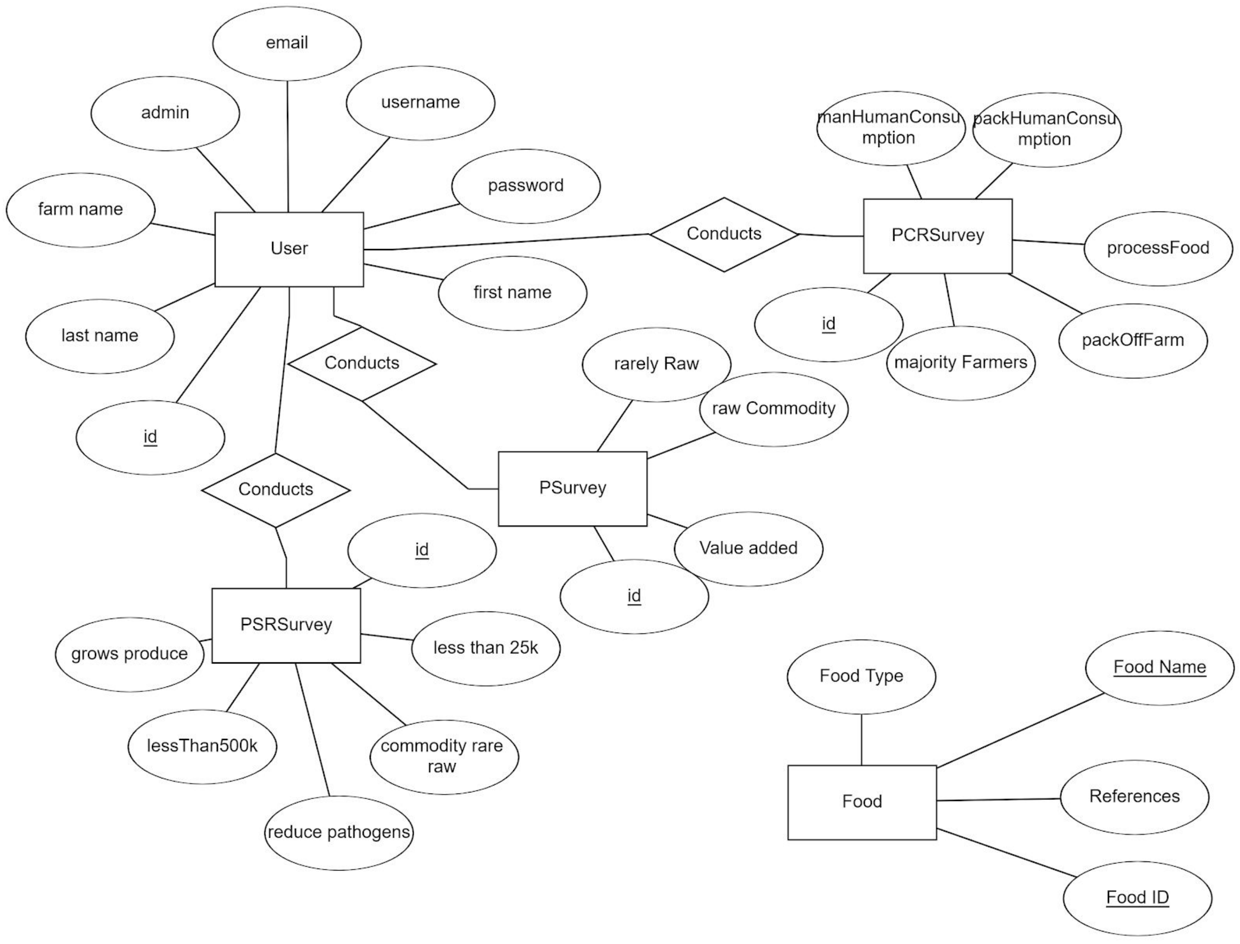
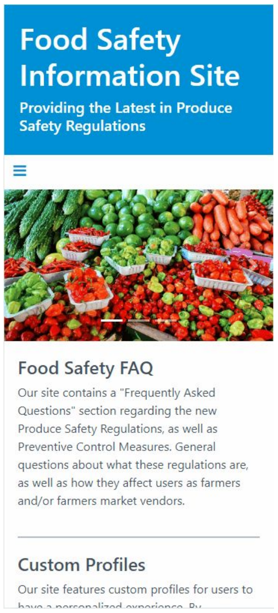


Figure 2: Database design

Implementation

Home Page

The home page is the most important page of the website. It is the main point of entry and needs to list all the features of the website in a user friendly manner. The following screen dumps are a side by side comparison of the home page when viewed through a desktop vs. a mobile device like a smartphone.



Food Safety Information Site

Providing the Latest in Produce Safety Regulations

Food Safety FAQ

Our site contains a "Frequently Asked Questions" section regarding the new Produce Safety Regulations, as well as Preventive Control Measures. General questions about what these regulations are, as well as how they affect users as farmers and/or farmers market vendors.

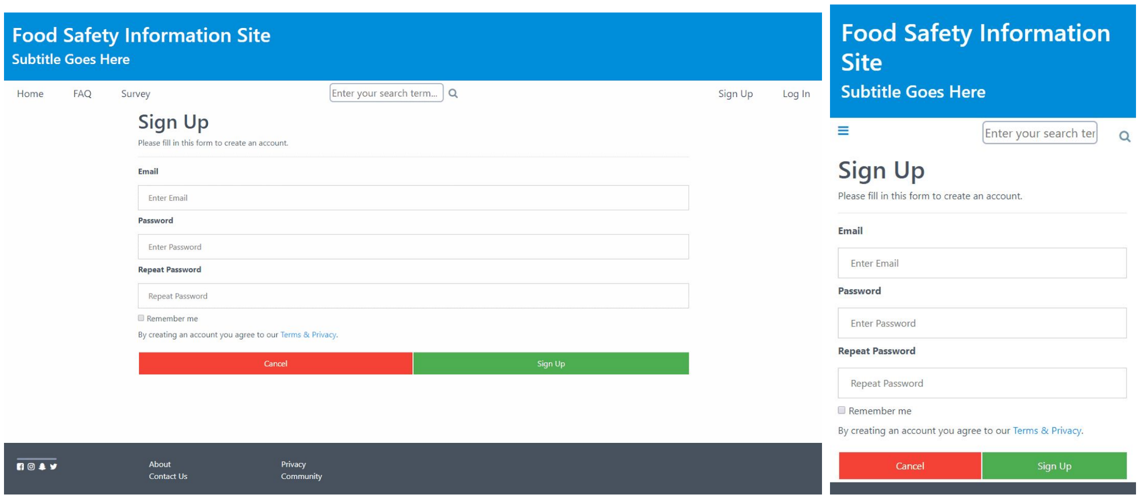
Custom Profiles

Our site features custom profiles for users to

Figure 3: Home screen

Sign In/Sign Up Pages

As seen in Figures 4 and 5, the sign in and sign up pages are used for user registration and sign in for the website. Users will be redirected to a survey after registration. These responses will be stored for future use.


Food Safety Information Site
Subtitle Goes Here

Home FAQ Survey Enter your search term... Sign Up Log In

Sign Up

Please fill in this form to create an account.

Email: Enter Email

Password: Enter Password

Repeat Password: Repeat Password

☐ Remember me

By creating an account you agree to our Terms & Privacy.

Cancel | Sign Up

About, Contact Us, Privacy, Community

Food Safety Information Site
Subtitle Goes Here

Enter your search ter

Sign Up

Please fill in this form to create an account.

Email: Enter Email

Password: Enter Password

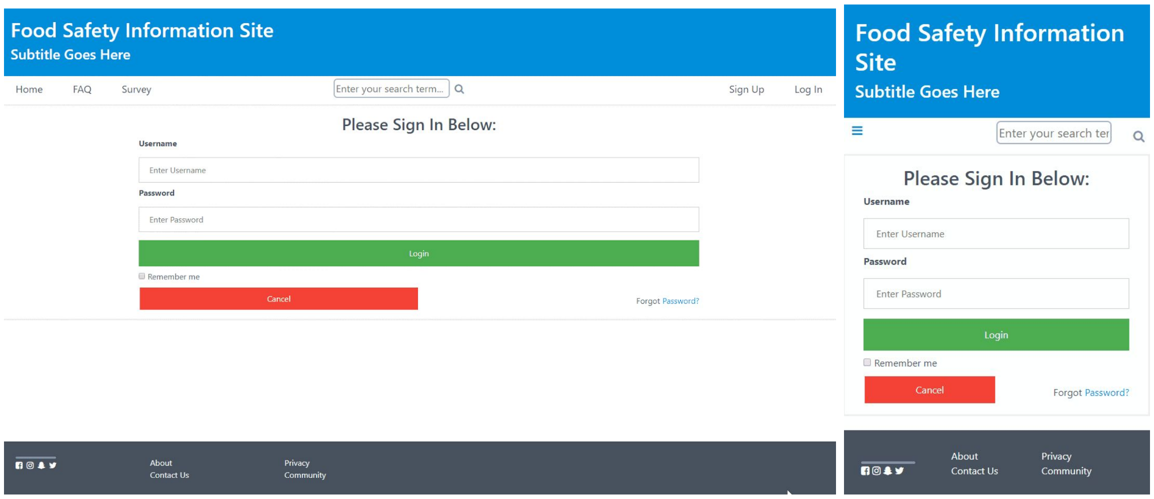
Repeat Password: Repeat Password

☐ Remember me

By creating an account you agree to our Terms & Privacy.

Cancel | Sign Up

Figure 4: Sign up screen


Food Safety Information Site
Subtitle Goes Here

Home FAQ Survey Enter your search term... Sign Up Log In

Please Sign In Below:

Username: Enter Username

Password: Enter Password

Login

☐ Remember me

Cancel | Forgot Password?

About, Contact Us, Privacy, Community

Food Safety Information Site
Subtitle Goes Here

Enter your search ter

Please Sign In Below:

Username: Enter Username

Password: Enter Password

Login

☐ Remember me

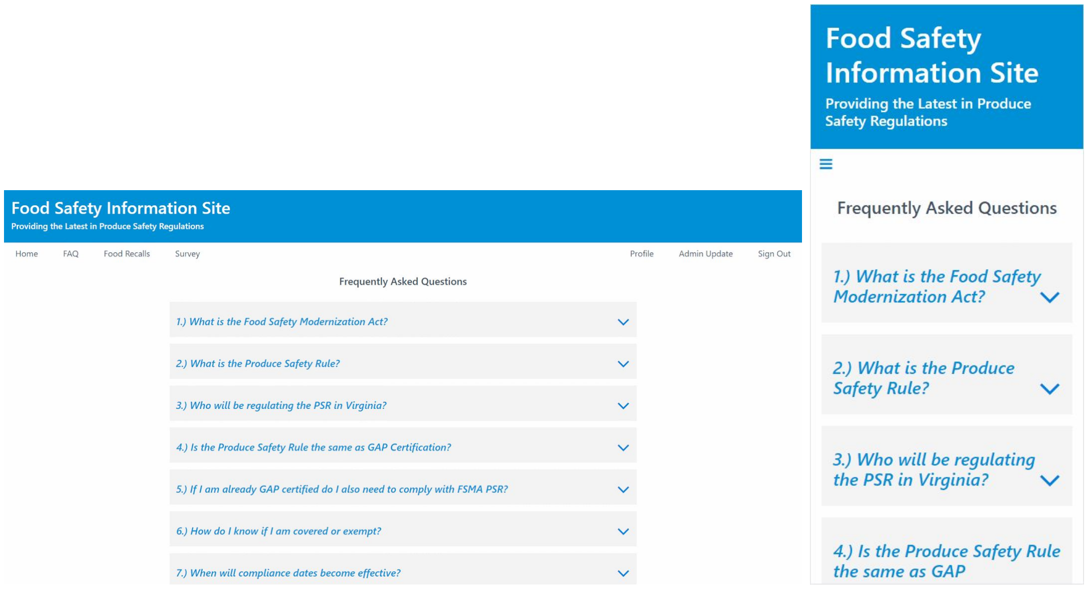
Cancel | Forgot Password?

About, Contact Us, Privacy, Community

Figure 5: Sign in screen

FAQ Page

This page contains a list of frequently asked questions that provide information to our users. The information within this page is related to produce safety rules and other farming regulations. The questions within the FAQ page and the answers to them can be changed by the administrator.



Food Safety Information Site

Providing the Latest in Produce Safety Regulations

Home FAQ Food Recalls Survey Profile Admin Update Sign Out

Frequently Asked Questions

1.) What is the Food Safety Modernization Act?

2.) What is the Produce Safety Rule?

3.) Who will be regulating the PSR in Virginia?

4.) Is the Produce Safety Rule the same as GAP Certification?

5.) If I am already GAP certified do I also need to comply with FSMA PSR?

6.) How do I know if I am covered or exempt?

7.) When will compliance dates become effective?

Food Safety Information Site

Providing the Latest in Produce Safety Regulations

Frequently Asked Questions

1.) What is the Food Safety Modernization Act?

2.) What is the Produce Safety Rule?

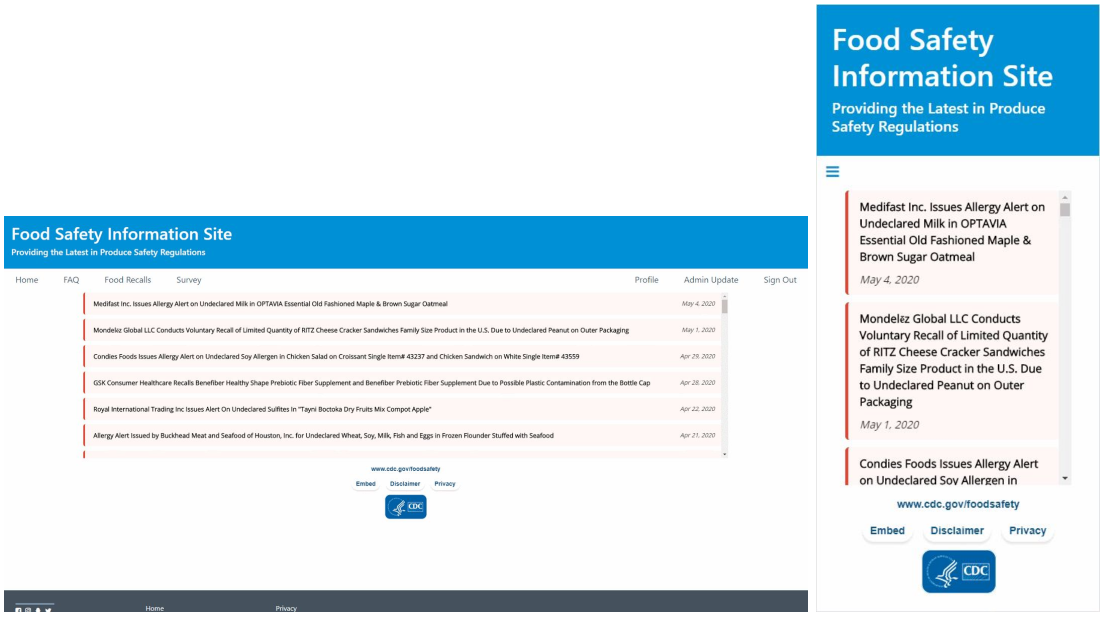
3.) Who will be regulating the PSR in Virginia?

4.) Is the Produce Safety Rule the same as GAP

Figure 6: FAQ screen

Food Recalls Page

This page contains a live scrollable feed directly from the Center for Disease Control's (CDC) web page. From this webpage we have scraped and imported their feed in relation to food recalls across the world. This will keep user's up-to-date on changes in the food industry related to allergens, recalls, and safety.



Food Safety Information Site

Providing the Latest in Produce Safety Regulations

Home FAQ Food Recalls Survey Profile Admin Update Sign Out

Medifast Inc. Issues Allergy Alert on Undeclared Milk in OPTAVIA Essential Old Fashioned Maple & Brown Sugar Oatmeal *May 4, 2020*

Mondelēz Global LLC Conducts Voluntary Recall of Limited Quantity of RITZ Cheese Cracker Sandwiches Family Size Product in the U.S. Due to Undeclared Peanut on Outer Packaging *May 1, 2020*

Condies Foods Issues Allergy Alert on Undeclared Soy Allergen in Chicken Salad on Croissant Single Item# 43237 and Chicken Sandwich on White Single Item# 43559 *Apr 29, 2020*

GSK Consumer Healthcare Recalls Benefiber Healthy Shape Prebiotic Fiber Supplement and Benefiber Prebiotic Fiber Supplement Due to Possible Plastic Contamination from the Bottle Cap *Apr 28, 2020*

Royal International Trading Inc Issues Alert On Undeclared Sulfites In "Tayni Boctoka Dry Fruits Mix Compot Apple" *Apr 22, 2020*

Allergy Alert Issued by Buckhead Meat and Seafood of Houston, Inc. for Undeclared Wheat, Soy, Milk, Fish and Eggs in Frozen Flounder Stuffed with Seafood *Apr 21, 2020*

www.cdc.gov/foodsafety

Embed Disclaimer Privacy

Home Privacy

Food Safety Information Site

Providing the Latest in Produce Safety Regulations

Medifast Inc. Issues Allergy Alert on Undeclared Milk in OPTAVIA Essential Old Fashioned Maple & Brown Sugar Oatmeal

May 4, 2020

Mondelēz Global LLC Conducts Voluntary Recall of Limited Quantity of RITZ Cheese Cracker Sandwiches Family Size Product in the U.S. Due to Undeclared Peanut on Outer Packaging

May 1, 2020

Condies Foods Issues Allergy Alert on Undeclared Soy Allergen in

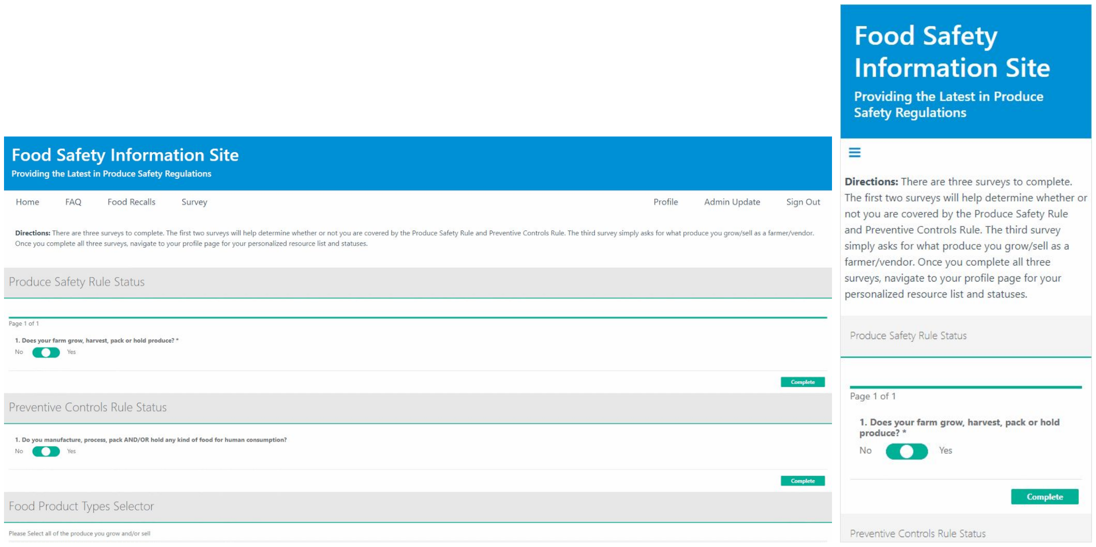
www.cdc.gov/foodsafety

Embed Disclaimer Privacy

Figure 7: Food recalls screen

Questionnaire Page

This page contains 2 surveys that indicate to the user whether they are covered by, or exempt from, rules based on their response to the survey. The determination is based on a table from our client.



Food Safety Information Site

Providing the Latest in Produce Safety Regulations

Directions: There are three surveys to complete. The first two surveys will help determine whether or not you are covered by the Produce Safety Rule and Preventive Controls Rule. The third survey simply asks for what produce you grow/sell as a farmer/vendor. Once you complete all three surveys, navigate to your profile page for your personalized resource list and statuses.

Produce Safety Rule Status

Page 1 of 1

1. Does your farm grow, harvest, pack or hold produce? *

Preventive Controls Rule Status

1. Do you manufacture, process, pack AND/OR hold any kind of food for human consumption?

Food Product Types Selector

Please Select all of the produce you grow and/or sell

Food Safety Information Site

Providing the Latest in Produce Safety Regulations

Directions: There are three surveys to complete. The first two surveys will help determine whether or not you are covered by the Produce Safety Rule and Preventive Controls Rule. The third survey simply asks for what produce you grow/sell as a farmer/vendor. Once you complete all three surveys, navigate to your profile page for your personalized resource list and statuses.

Produce Safety Rule Status

Page 1 of 1

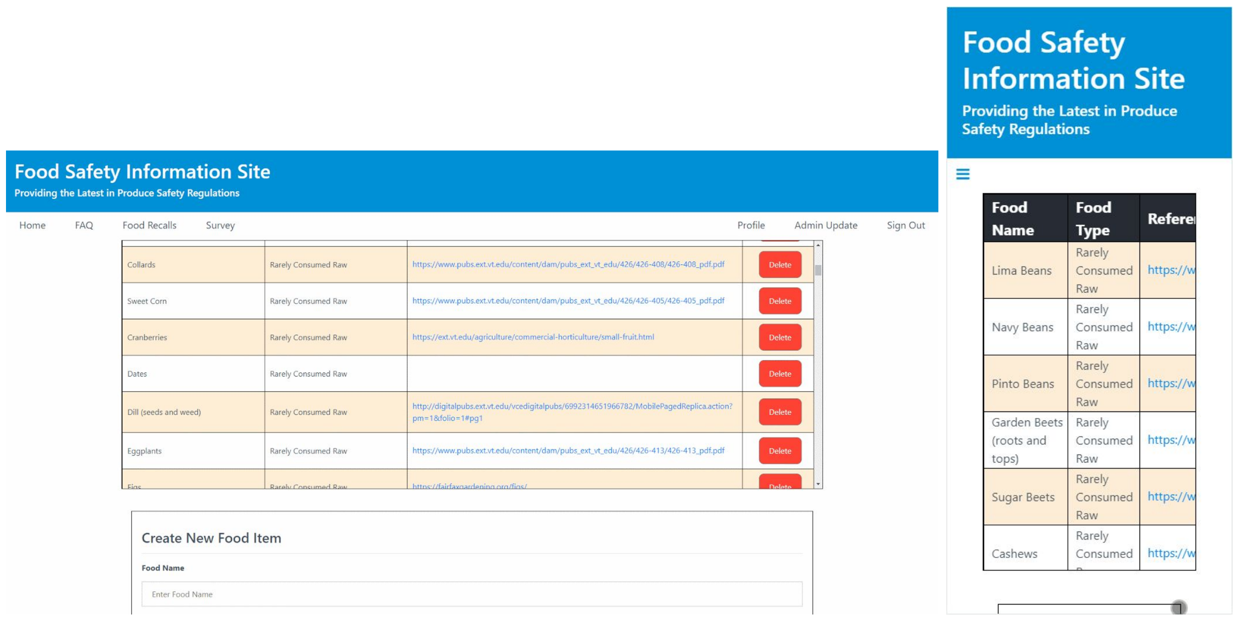
1. Does your farm grow, harvest, pack or hold produce? *

Preventive Controls Rule Status

Figure 8: Questionnaire Screen

Admin Update Page

As seen below in Figure 9, the admin update page contains the 'Food' table from our database which shows all foods grown in Virginia and North Carolina that our client is providing resources for. This table includes food name, food types, and the food's respective reference links to essential resources. This page also contains a form where the admin profile can add new items to this database table and also delete certain entries.


Food Safety Information Site

Providing the Latest in Produce Safety Regulations

Home FAQ Food Recalls Survey Profile Admin Update Sign Out

Collards	Rarely Consumed Raw	https://www.pubs.ext.vt.edu/content/dam/pubs_ext_vt_edu/426/426-408/426-408_pdf.pdf	Delete
Sweet Corn	Rarely Consumed Raw	https://www.pubs.ext.vt.edu/content/dam/pubs_ext_vt_edu/426/426-405/426-405_pdf.pdf	Delete
Cranberries	Rarely Consumed Raw	https://ext.vt.edu/agriculture/commercial-horticulture/small-fruit.html	Delete
Dates	Rarely Consumed Raw		Delete
Dill (seeds and weed)	Rarely Consumed Raw	http://digitalpubs.ext.vt.edu/vcedigitalpubs/6992314651966782/MobilePagedReplica.action?pm=1&folio=1#pg1	Delete
Eggplants	Rarely Consumed Raw	https://www.pubs.ext.vt.edu/content/dam/pubs_ext_vt_edu/426/426-413/426-413_pdf.pdf	Delete
Figs	Rarely Consumed Raw	https://fairfaxgardening.org/figs/	Delete

Create New Food Item

Food Name

Enter Food Name

Food Safety Information Site

Providing the Latest in Produce Safety Regulations

Food Name	Food Type	Refere
Lima Beans	Rarely Consumed Raw	https://w
Navy Beans	Rarely Consumed Raw	https://w
Pinto Beans	Rarely Consumed Raw	https://w
Garden Beets (roots and tops)	Rarely Consumed Raw	https://w
Sugar Beets	Rarely Consumed Raw	https://w
Cashews	Rarely Consumed	https://w

Figure 9: Admin update screen

Testing/Evaluation

We tested the user interface with the latest version of our website. We needed to make sure that everything on the site was intuitive and easy to use even for those without a technical background. To test usability, we created benchmark tasks and asked a person with little to minimum technical background to complete these tasks. These tasks can be seen in Table 1. The format of the table was adapted from *The UX Book* (Hartson 385).

Work Role	UX Goal	UX Measure	Measuring Instrument	UX Metric	Baseline Level	Target Level	Observed Results
Someone with a non-technical background	Ease of use for new user	Initial user performance	Find out what the Food Safety Modernization Act is	Average number of errors	< 1	< 1	<1
Someone with a non-technical background	Ease of use for new user	Initial user performance	Access the Preventative Controls Rule Status Survey	Average time on task	10 seconds	8 seconds	4 seconds

Someone with a non-technical background	Ease of use for new user	Initial user performance	Fill out a registration form and create a new account	Average time on task	45 seconds	30 seconds	20 seconds
Someone with a non-technical background	Initial customer satisfacti on	First Impression	Questionnaire	Average rating across user questions	6/10	7/10	7.5/10

Table 1: UX target table

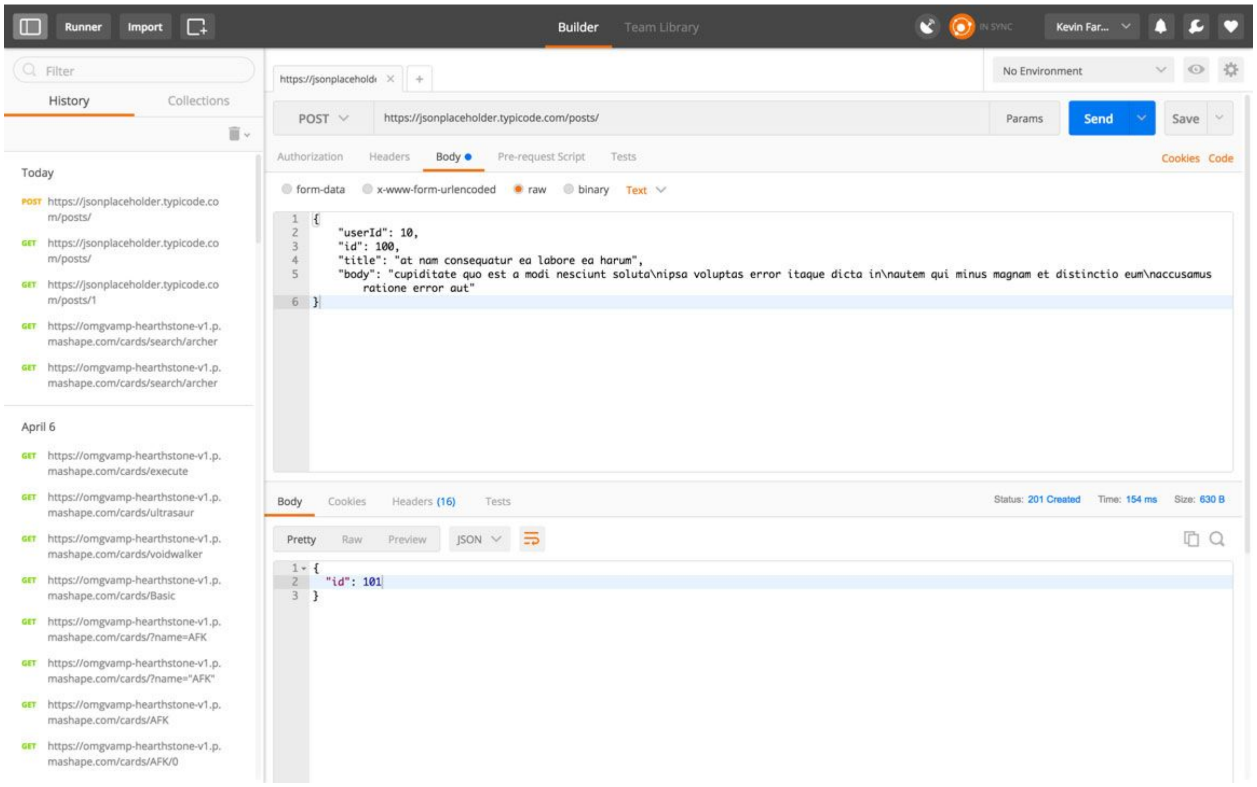
The questionnaire as seen in Table 2, also adapted from *The UX Book* (Hartson 446), was used for the fourth benchmark task. It asked the user to rate various topics with respect to the website.

1. Terminology reflects application purpose	[distantly]	0 1 2 3 4 5 6 7 8 9 10	[closely]
2.. Informative feedback	[never]	0 1 2 3 4 5 6 7 8 9 10	[always]
3.. Intuitive layout	[never]	0 1 2 3 4 5 6 7 8 9 10	[always]
4. Sequence of displays	[confusing]	0 1 2 3 4 5 6 7 8 9 10	[clear]

Table 2: Questionnaire used to measure first impression

Due to COVID-19 and the restriction of not being able to interact with people outside of our households, it was difficult to administer this benchmark to a large number of users. As such, we have currently only administered this evaluation to two people. The results from one of the testers can be seen in the last column of Table 1.

For testing the backend in addition to utilizing the actual website we used a combination of the software programs, Postman and Compass. The UI of Postman can be seen in Figure 10.



POST https://jsonplaceholder.typicode.com/posts/

```
{
    "userId": 10,
    "id": 100,
    "title": "at nam consequatur ea labore ea harum",
    "body": "cupiditate quo est a modi nesciunt soluta\nipsa voluptas error itaque dicta in\nautem qui minus magnam et distinctio eum\naccusamus
        ratione error aut"
}
```

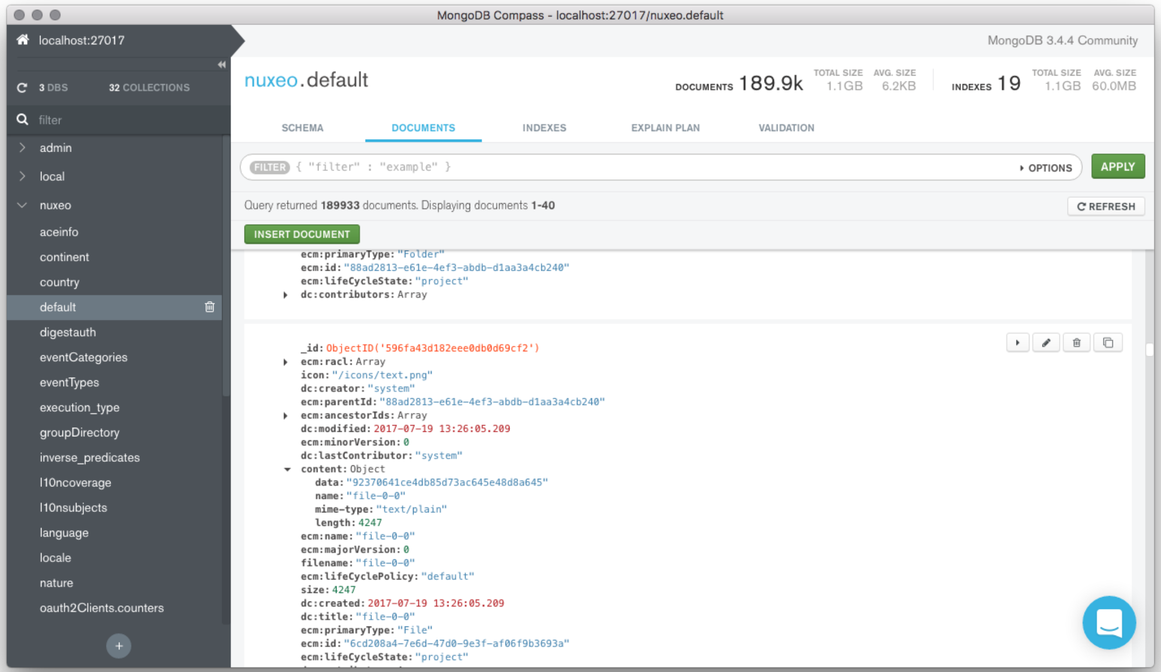
Status: 201 Created Time: 154 ms Size: 630 B

```
{
  "id": 101
}
```

Runner
Import
Builder
Team Library
IN SYNC
Kevin Far...
Filter
History
Collections
Today
POST https://jsonplaceholder.typicode.com/posts/
GET https://jsonplaceholder.typicode.com/posts/
GET https://jsonplaceholder.typicode.com/posts/1
GET https://omgvamp-hearthstone-v1.p.mashape.com/cards/search/archer
GET https://omgvamp-hearthstone-v1.p.mashape.com/cards/search/archer
April 6
GET https://omgvamp-hearthstone-v1.p.mashape.com/cards/execute
GET https://omgvamp-hearthstone-v1.p.mashape.com/cards/ultrasaur
GET https://omgvamp-hearthstone-v1.p.mashape.com/cards/voidwalker
GET https://omgvamp-hearthstone-v1.p.mashape.com/cards/Basic
GET https://omgvamp-hearthstone-v1.p.mashape.com/cards/?name=AFK
GET https://omgvamp-hearthstone-v1.p.mashape.com/cards/?name="AFK"
GET https://omgvamp-hearthstone-v1.p.mashape.com/cards/AFK
GET https://omgvamp-hearthstone-v1.p.mashape.com/cards/AFK/0
No Environment
Params
Send
Save
Authorization
Headers
Body
Pre-request Script
Tests
Cookies
Code
form-data
x-www-form-urlencoded
raw
binary
Text
Body
Cookies
Headers (16)
Tests
Pretty
Raw
Preview
JSON

Figure 10: Postman UI screen

Postman allows the user to send POST, GET, and PATCH requests directly to the server. It returns the response as required. This application made it extremely easy to test our API by looking at the results returned. It is also great that Postman allows the user to customize the body of the JSON file that is sent as the request. To see whether the changes made by these requests are permanent we utilized Compass. The UI for Compass is shown in Figure 11.



```
{ "filter" : "example" }
```

Query returned **189933** documents. Displaying documents **1-40**

```
ecm:primaryType: "Folder"
ecm:id: "88ad2813-e61e-4ef3-abdb-d1aa3a4cb240"
ecm:lifeCycleState: "project"
dc:contributors: Array

_id: ObjectID('596fa43d182eee0db0d69cf2')
ecm:racl: Array
icon: "/icons/text.png"
dc:creator: "system"
ecm:parentId: "88ad2813-e61e-4ef3-abdb-d1aa3a4cb240"
ecm:ancestorIds: Array
dc:modified: 2017-07-19 13:26:05.209
ecm:minorVersion: 0
dc:lastContributor: "system"
content: Object
  data: "92370641ce4db85d73ac645e48d8a645"
  name: "file-0-0"
  mime-type: "text/plain"
  length: 4247
ecm:name: "file-0-0"
ecm:majorVersion: 0
filename: "file-0-0"
ecm:lifeCyclePolicy: "default"
size: 4247
dc:created: 2017-07-19 13:26:05.209
dc:title: "file-0-0"
ecm:primaryType: "File"
ecm:id: "6cd208a4-7e6d-47d0-9e3f-af06f9b3693a"
ecm:lifeCycleState: "project"
```

Figure 11: MongoDB compass screen

This application is great as it allowed our team to connect to our mongoDB database and see what changes were made. This software combination facilitated testing our backend.

<u>User's Manual</u>

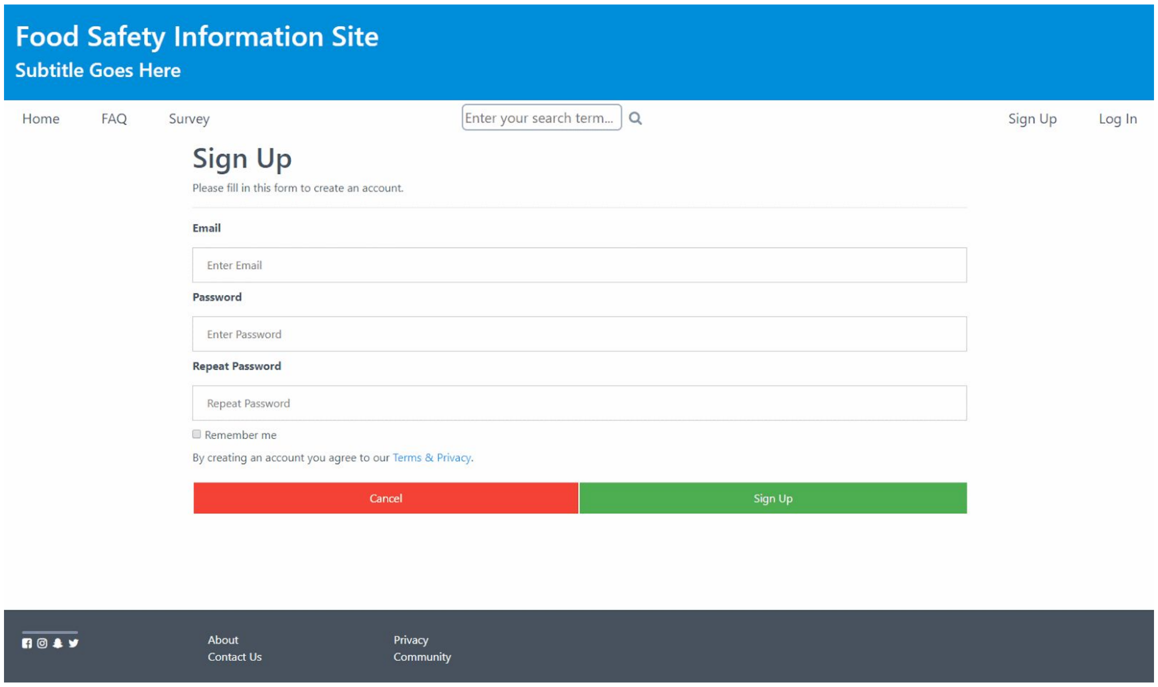
Home Page

Upon entering the site, you are taken to the home page as seen in Figure 8. The page includes a site header and a navigation bar at the top which can be used to navigate to different pages on the site. The header and navigation bar will be standard on every page. The navigation bar includes clickable tabs to return to the home page, access the frequently asked questions, be taken to the food safety surveys, search the website, go to the sign up page, and be taken to the login screen. The final version of the home page can be seen below in Figure 12.

Figure 12: Home page

Sign Up Page

After clicking on the sign up tab in the navigation bar, you will be taken to the sign up page. See Figure 13.


Food Safety Information Site
Subtitle Goes Here

Home FAQ Survey Enter your search term... Sign Up Log In

Sign Up

Please fill in this form to create an account.

Email

Enter Email

Password

Enter Password

Repeat Password

Repeat Password

Remember me

By creating an account you agree to our Terms & Privacy.

Cancel Sign Up

About
Contact Us
Privacy
Community

Figure 13: Sign up page

The sign up page asks users to enter information to create an account. As seen in Figure 13, the sign up page is a form that takes several inputs. We require users to enter an email address, create a password, and re-enter the password in the correct fields. Users can also select the “Remember me” checkbox if they want to be automatically logged in for future visits to the website. Once all entry boxes have been filled, the user can click on the green “Sign Up” button to create their account. If a user decides that they no longer want to create an account they can click on the red “Cancel” button at any time to stop the account creation process and be taken back to the homepage.

Sign In Page

If a user clicks on the “Log In” tab in the navigation bar, they will be taken to the sign in page. See Figure 14.

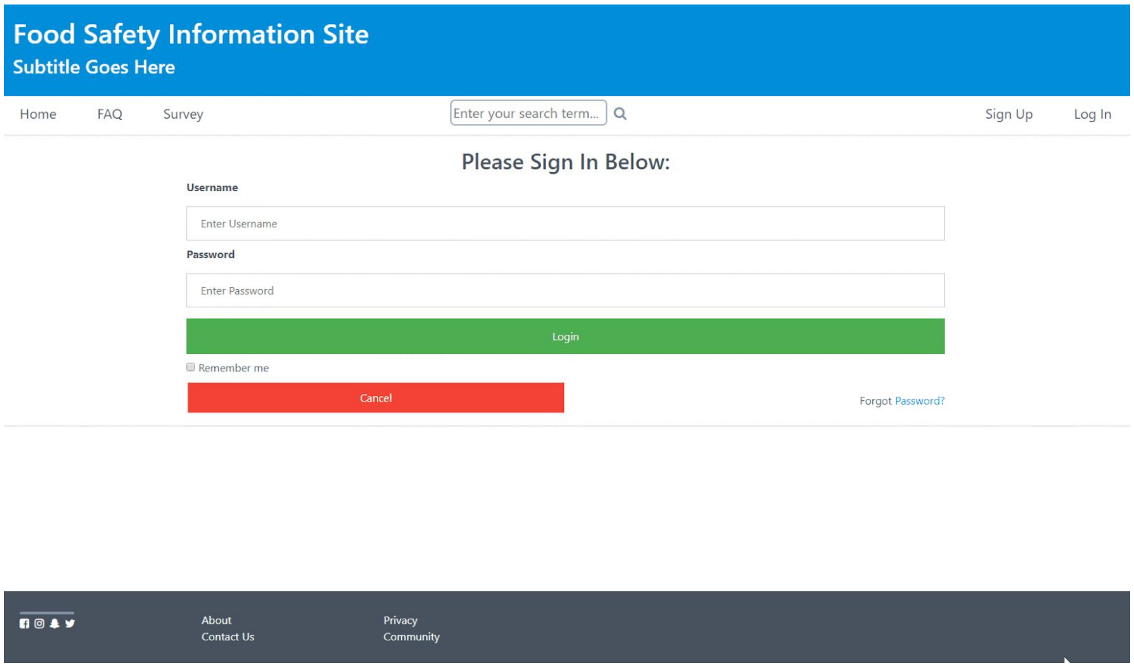
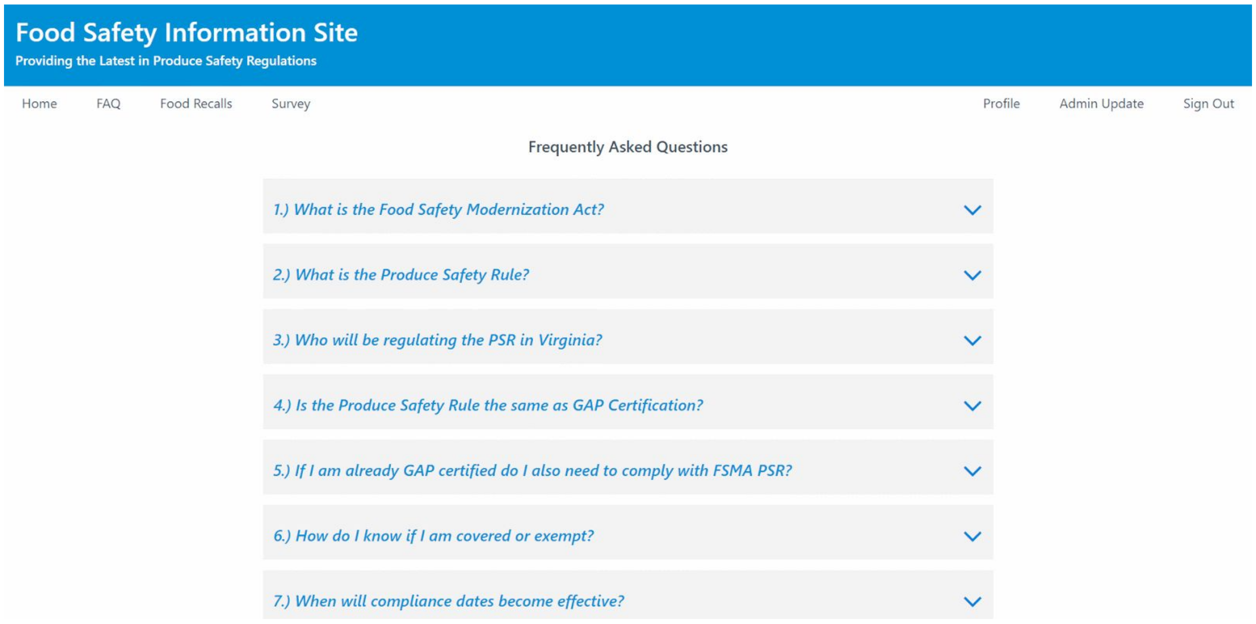


Figure 14: Sign in page

The sign in page contains a form that asks the user for a username/email address and a password. Once a user has entered these fields they can click the green “Login” button to be signed in. If the user enters incorrect information they will be shown an error informing them that the login attempt failed. Similar to the sign up page, there is a “Remember Me” checkbox and a “Cancel” button with the same functionalities. In addition, this form also includes a “Forgot Password?” link that users can click when they are unable to remember their password. Once clicked, this should take users through additional steps to reset their passwords, though this has not been implemented yet.

FAQ Page

Upon clicking the “FAQ” tab in the navigation bar, users are taken to the frequently asked questions page as seen in Figure 15.


Food Safety Information Site
Providing the Latest in Produce Safety Regulations
Home
FAQ
Food Recalls
Survey
Profile
Admin Update
Sign Out

Frequently Asked Questions

1.) What is the Food Safety Modernization Act?

2.) What is the Produce Safety Rule?

3.) Who will be regulating the PSR in Virginia?

4.) Is the Produce Safety Rule the same as GAP Certification?

5.) If I am already GAP certified do I also need to comply with FSMA PSR?

6.) How do I know if I am covered or exempt?

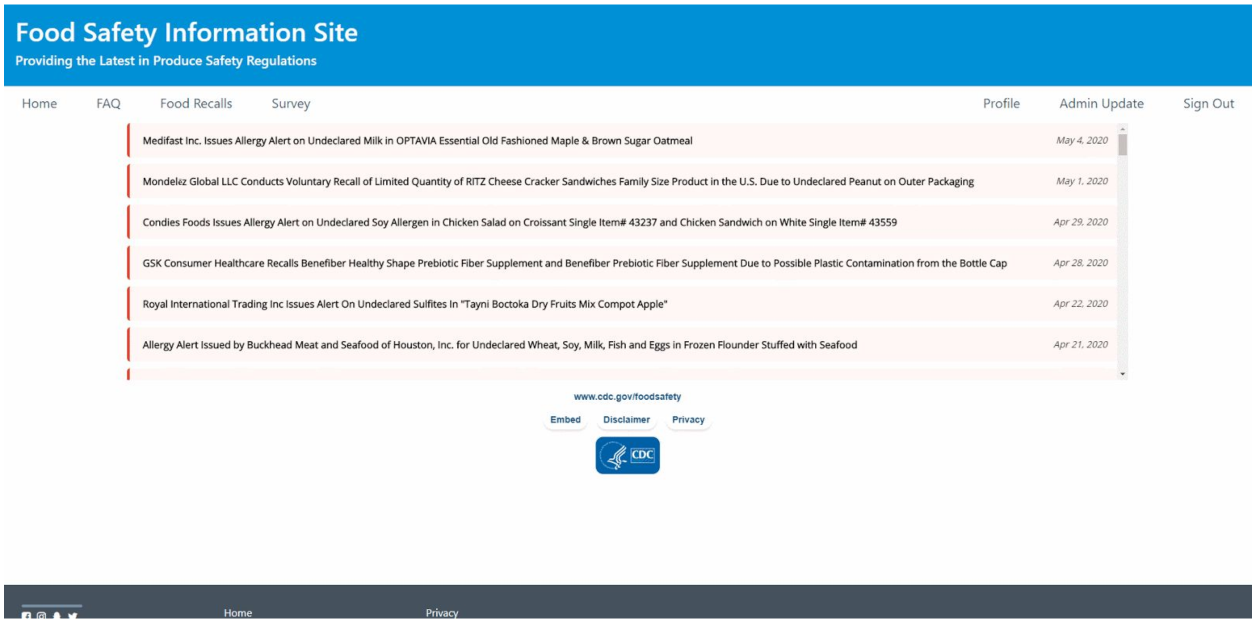
7.) When will compliance dates become effective?

Figure 15: FAQ page

This page includes the answer to thirteen common questions pertaining to food safety. If a user clicks on a question, the answer will appear in the form of a drop down menu. Users can open as many or as few drop downs as they want.

Food Recalls Page

The food recalls page is accessed by clicking on the “food recalls” tab in the navigation bar. A user is not required to be signed in to view this feature. A scrollable feed will be shown on this page that is directly pulled from the Center for Disease Control’s (CDC) website. This feed provides up-to-date concerns and recalls on all food across the globe. Each feed item is clickable to learn more about the recall associated and will take the user to the article.


Food Safety Information Site

Providing the Latest in Produce Safety Regulations

Home FAQ Food Recalls Survey Profile Admin Update Sign Out

Medifast Inc. Issues Allergy Alert on Undeclared Milk in OPTAVIA Essential Old Fashioned Maple & Brown Sugar Oatmeal	May 4, 2020
Mondelēz Global LLC Conducts Voluntary Recall of Limited Quantity of RITZ Cheese Cracker Sandwiches Family Size Product in the U.S. Due to Undeclared Peanut on Outer Packaging	May 1, 2020
Condies Foods Issues Allergy Alert on Undeclared Soy Allergen in Chicken Salad on Croissant Single Item# 43237 and Chicken Sandwich on White Single Item# 43559	Apr 29, 2020
GSK Consumer Healthcare Recalls Benefiber Healthy Shape Prebiotic Fiber Supplement and Benefiber Prebiotic Fiber Supplement Due to Possible Plastic Contamination from the Bottle Cap	Apr 28, 2020
Royal International Trading Inc Issues Alert On Undeclared Sulfites In "Tayni Boctoka Dry Fruits Mix Compot Apple"	Apr 22, 2020
Allergy Alert Issued by Buckhead Meat and Seafood of Houston, Inc. for Undeclared Wheat, Soy, Milk, Fish and Eggs in Frozen Flounder Stuffed with Seafood	Apr 21, 2020

www.cdc.gov/foodsafety

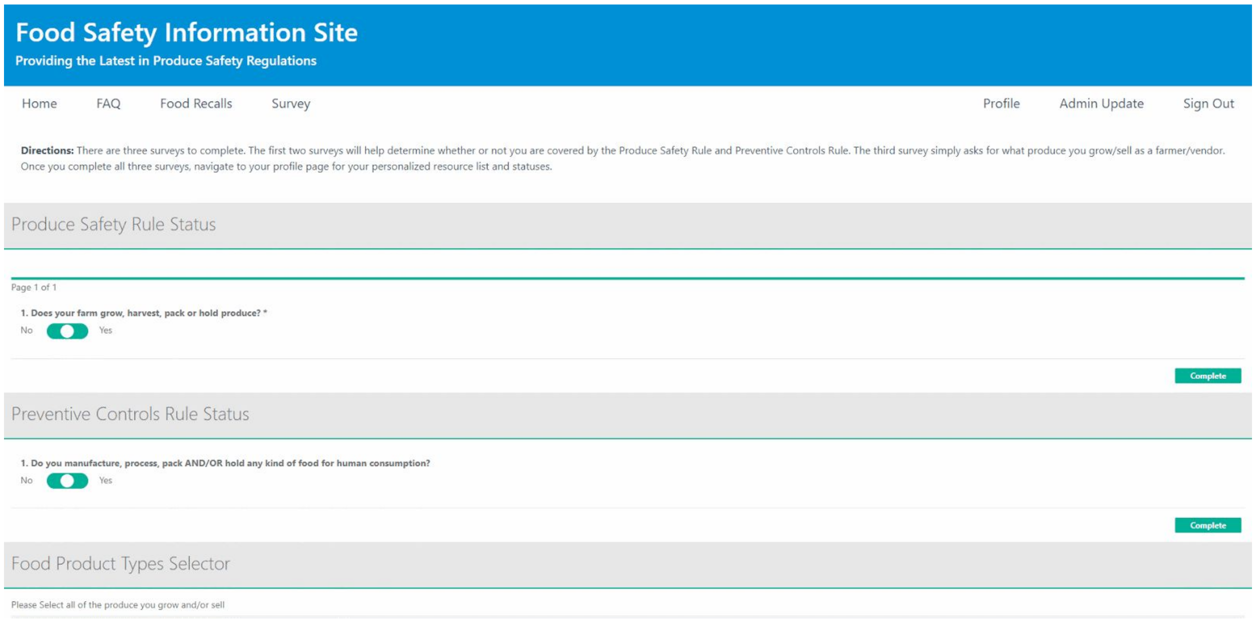
Embed Disclaimer Privacy

Home Privacy

Figure 16: Food recalls page

Questionnaire Page

The questionnaire page is accessed by clicking on the “Survey” tab in the navigation bar. A user is required to be signed in to use this feature. Two surveys will be shown on this page. The first survey tells a user if they are covered or exempt from the Product Safety Rule. Users will click to select “yes” or “no” for each question in the survey. Instant results for coverage or exemption will be returned according to the answers the user selects. If a user would like to save the results of their survey to their profile, they can click the green “complete” button. The other survey regarding the Preventative Controls Rule has similar usage.


Food Safety Information Site

Providing the Latest in Produce Safety Regulations

Directions: There are three surveys to complete. The first two surveys will help determine whether or not you are covered by the Produce Safety Rule and Preventive Controls Rule. The third survey simply asks for what produce you grow/sell as a farmer/vendor. Once you complete all three surveys, navigate to your profile page for your personalized resource list and statuses.

Produce Safety Rule Status

Page 1 of 1

1. Does your farm grow, harvest, pack or hold produce? *

No Yes

Preventive Controls Rule Status

1. Do you manufacture, process, pack AND/OR hold any kind of food for human consumption?

No Yes

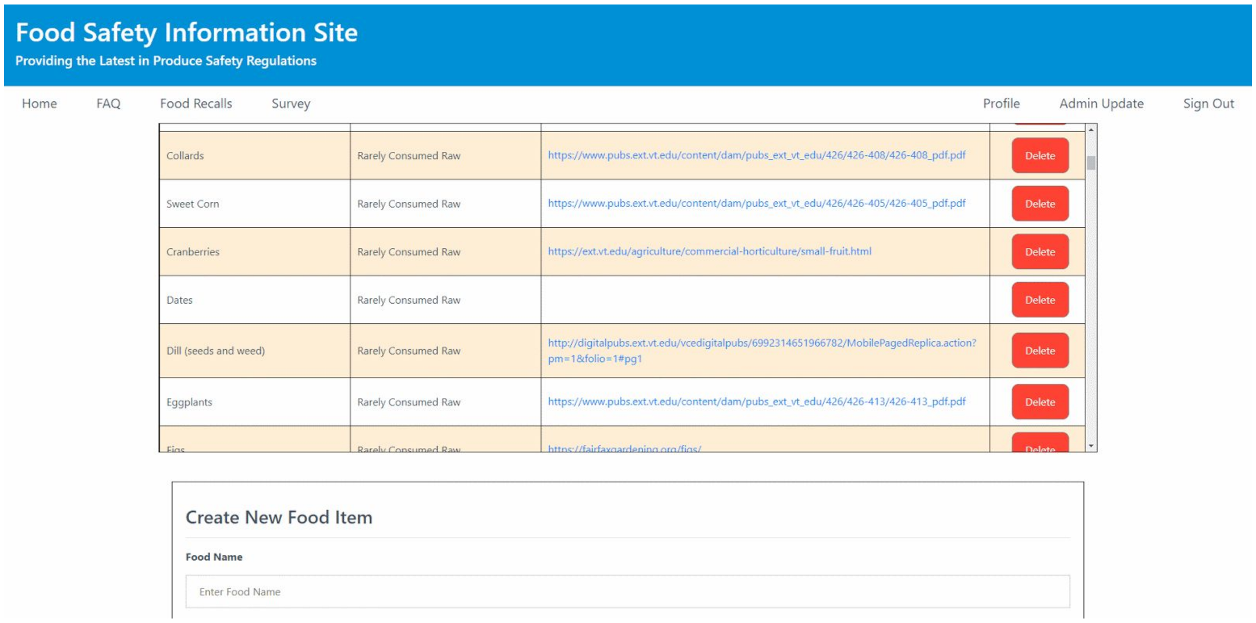
Food Product Types Selector

Please Select all of the produce you grow and/or sell

Figure 17: Questionnaire page

Admin Update Page

The admin update page is seen in the navigation bar when the current user is logged into the admin account associated with our application. This feature allows the admin to view, add, and delete the food items that are in our foods database. This database contains the produce and reference links that are delivered to certain users who receive data from survey completion. The user will see the populated data table with each row containing a delete button which will remove that given entry from the database. Underneath the table is a form used to add new items into the database. The admin must fill in the ‘Food Name’ and ‘Food Type’ fields and can have zero or more entries for the ‘Food’s Reference Links” field by adding or removing entries using the buttons. Once the admin user has filled out all relevant fields they can click submit and the table will be updated along with the database.


Food Safety Information Site
Providing the Latest in Produce Safety Regulations
Home
FAQ
Food Recalls
Survey
Profile
Admin Update
Sign Out

Collards	Rarely Consumed Raw	https://www.pubs.ext.vt.edu/content/dam/pubs_ext_vt_edu/426/426-408/426-408_pdf.pdf	Delete
Sweet Corn	Rarely Consumed Raw	https://www.pubs.ext.vt.edu/content/dam/pubs_ext_vt_edu/426/426-405/426-405_pdf.pdf	Delete
Cranberries	Rarely Consumed Raw	https://ext.vt.edu/agriculture/commercial-horticulture/small-fruit.html	Delete
Dates	Rarely Consumed Raw		Delete
Dill (seeds and weed)	Rarely Consumed Raw	http://digitalpubs.ext.vt.edu/vcedigitalpubs/6992314651966782/MobilePagedReplica.action?pm=1&folio=1#pg1	Delete
Eggplants	Rarely Consumed Raw	https://www.pubs.ext.vt.edu/content/dam/pubs_ext_vt_edu/426/426-413/426-413_pdf.pdf	Delete
Figs	Rarely Consumed Raw	https://fairfaxgardening.org/figs/	Delete

Create New Food Item

Food Name

Enter Food Name

Figure 18: Admin update page

<u>Developer's Manual</u>

Project Methodology

In order to properly understand our project, the following section is dedicated to listing the project goals, the types of users that will use our website, and details regarding how to implement services for those users to achieve their goals.

For our website system, we have a few types of users, each with similar, and also different, goals when accessing the site. The main groups of people that will use it are the following:

1. Farmers
2. Farmers Market Vendors
3. Web Administrator(s)

Each of these types of users have goals that our system must support, that are listed below:

- Farmers - These types of users are one of the two which are directly affected by the Produce Safety rules and regulations. They are the ones who grow the produce and crops that are pertinent to farmers market supply, and must comply with these regulations in order to continue to do so. Their goals consist of the following:
    - Access information regarding Produce Safety Rules, Preventive Control Rules, and other related regulations

    - Understand whether or not their farm and farming practices comply with produce and safety regulations
    - Understand, if they do not comply, how to change their practices and farms to follow the produce and safety rules
- Farmers Market Vendors - These types of users are one of the two which are directly affected by the Produce Safety rules and regulations. They are the ones distributing produce and crops, and must follow guidelines in order to continue distribution in farmers markets. Their goals consist of the following:
    - Access information regarding Produce Safety Rules, Preventive Control Rules, and other related regulations
    - Understand whether or not their supply and distribution methods comply with produce and safety regulations
    - Understand, if they do not comply, how to change their practices and supply to follow the produce and safety rules
- Web Administrator(s) - These types of users are in charge of maintaining the website system, and need back-end administrative rights in order to keep up with the system's information, as well as provide the proper, updated news and information on the laws and regulations that the other types of users need. Their goals consist of the following:
    - Better inform users of any changes to the produce and safety guidelines
    - Answer any common questions users may have regarding the new produce and safety regulations
    - Provide users with resources that will help them comply with the new produce and safety regulations

For each type of user that will access our site, their goals need to be broken down into smaller tasks, so that the flow of their usage can be easily read, in order to help determine implementation:

Farmers

- Access information regarding Produce Safety Rules, Preventive Control Rules, and other related regulations
- As seen in Figure 19, whether a user visits the FAQ page, or searches about the information requested, the same workflow diagram can be generated to show how the user can achieve the goal.

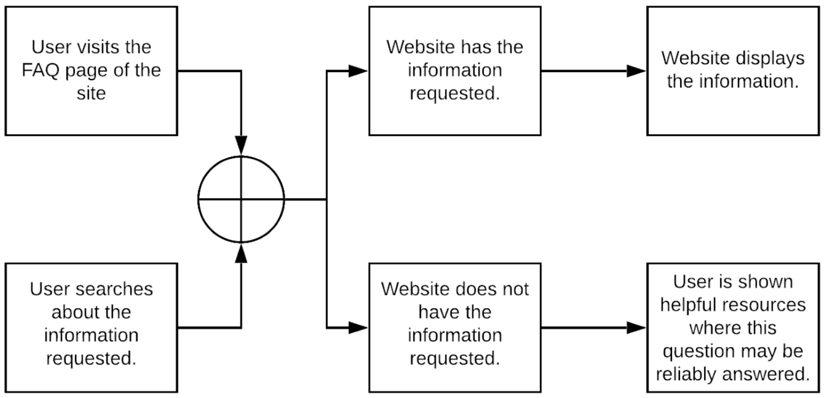

User visits the FAQ page of the site
User searches about the information requested.
Website has the information requested.
Website displays the information.
Website does not have the information requested.
User is shown helpful resources where this question may be reliably answered.

Figure 19: Farmer user access information goal diagram

- Understand whether or not their farm and farming practices comply with produce and safety regulations

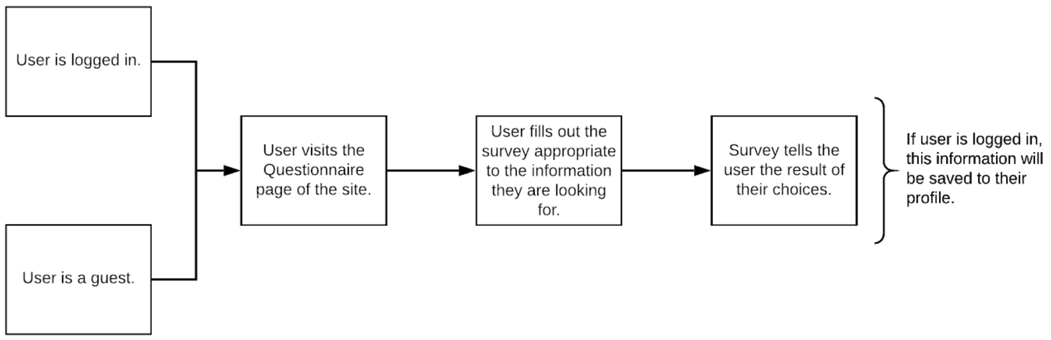

User is logged in.
User is a guest.
User visits the Questionnaire page of the site.
User fills out the survey appropriate to the information they are looking for.
Survey tells the user the result of their choices.
If user is logged in, this information will be saved to their profile.

Figure 20: Farmer user assess farm goal diagram

- Understand, if they do not comply, how to change their practices and farms to follow the produce and safety rules

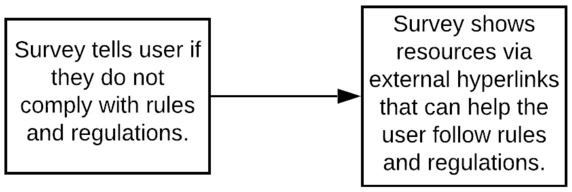

Survey tells user if they do not comply with rules and regulations.
Survey shows resources via external hyperlinks that can help the user follow rules and regulations.

Figure 21: Farmer user change farm goal diagram

Farmers Market Vendors

- Access information regarding Produce Safety Rules, Preventive Control Rules, and other related regulations
- As seen in Figure 22, whether a user visits the FAQ page, or searches about the information requested, the same workflow diagram can be generated to show how the user can achieve the goal.

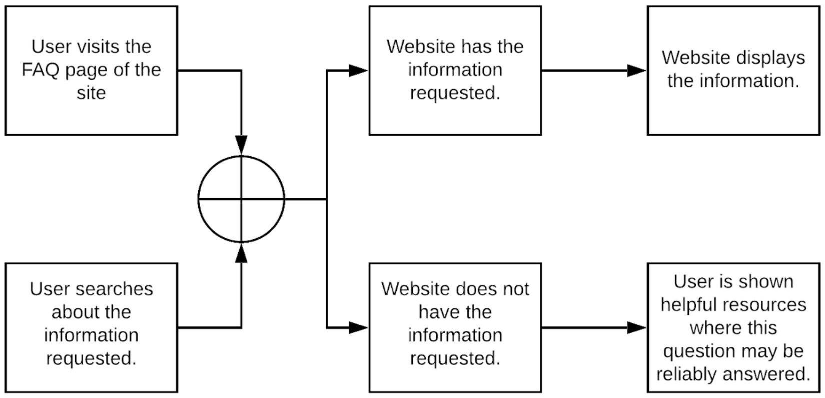

User visits the FAQ page of the site
User searches about the information requested.
Website has the information requested.
Website does not have the information requested.
Website displays the information.
User is shown helpful resources where this question may be reliably answered.

Figure 22: Vendor user access information goal diagram

- Understand whether or not their supply and distribution methods comply with produce and safety regulations

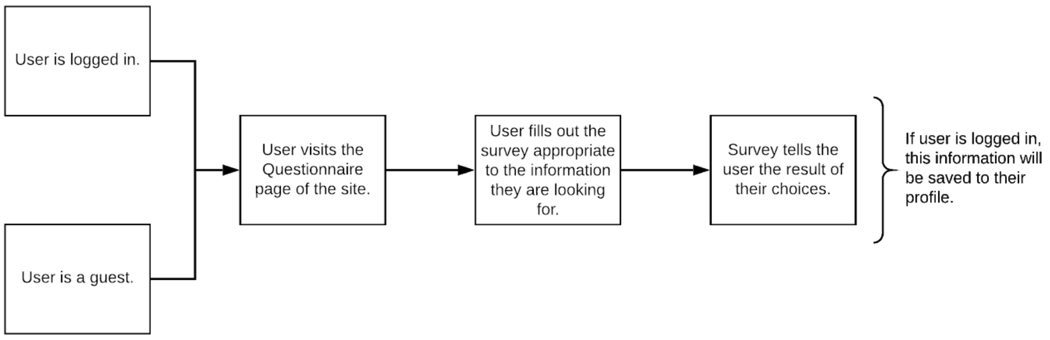

User is logged in.
User is a guest.
User visits the Questionnaire page of the site.
User fills out the survey appropriate to the information they are looking for.
Survey tells the user the result of their choices.
If user is logged in, this information will be saved to their profile.

Figure 23: Vendor user assess supply goal diagram

- Understand, if they do not comply, how to change their practices and supply to follow the produce and safety rules

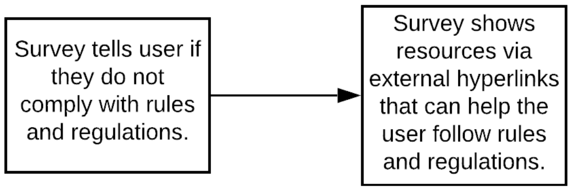

Survey tells user if they do not comply with rules and regulations.
Survey shows resources via external hyperlinks that can help the user follow rules and regulations.

Figure 24: Vendor user change supply goal diagram

Web Administrator(s)

- Better inform users of any changes to the produce and safety guidelines

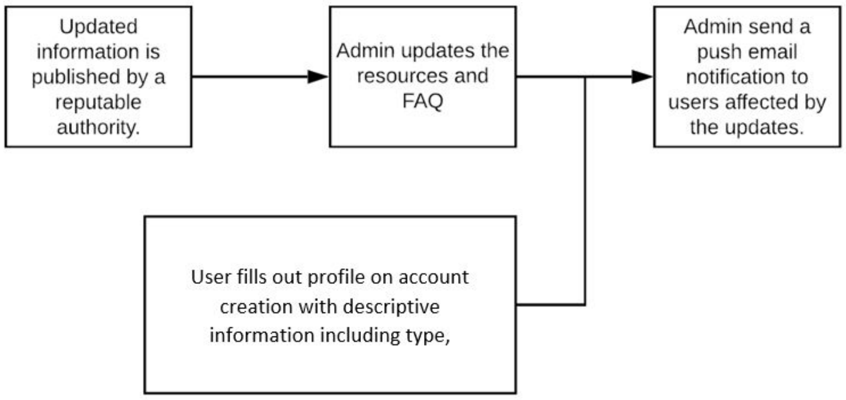

Updated information is published by a reputable authority.
Admin updates the resources and FAQ
Admin send a push email notification to users affected by the updates.
User fills out profile on account creation with descriptive information including type,

Figure 25: Web Admin user inform changes goal diagram

- Answer any common questions users may have regarding the new produce and safety regulations

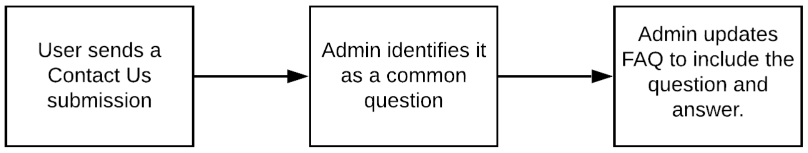

User sends a Contact Us submission
Admin identifies it as a common question
Admin updates FAQ to include the question and answer.

Figure 26: Web Admin user FAQ goal diagram

- Provide users with resources that will help them comply with the new produce and safety regulations

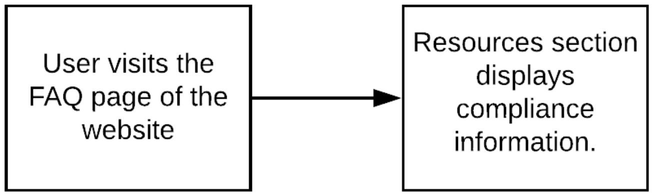

User visits the FAQ page of the website
Resources section displays compliance information.

Figure 27: Web Admin user resources goal diagram

With an understanding of the smaller tasks each user will need to complete in order to achieve their goal(s), we can then describe an implementation-based service for each type of user and their task(s):

Farmers:

1.) Task: Access information regarding Produce Safety Rules, Preventive Control Rules, and other related regulations

The goal of this website is to provide Produce Safety Rules, Preventive Control Rules, and other regulations to the Farmers (users) in order to meet Virginia state requirements to sell produce/products at certain places. For this we need input files from the FDA and Virginia State regulations from our clients data set. From there depending on the user's survey results and what laws they are covered/not covered by we produce a user output of information we have in the database. If we do not have the proper supplemental information we will output links to better answer the user's questions/concerns. For this we are building a React website environment where the user will be requesting information and/or taking surveys to receive personal status information. We will need functions that grab the proper information stored in our database, as well as handle the cases when we do not have all the information requested and we can grab external links when appropriate.

2.) Task: Understand whether or not their farm and farming practices comply with produce and safety regulations

This task determines the status of a farmer and his/her farm. In order to receive certain selling benefits and access to sell at farmers markets throughout the state of Virginia they have to meet certain local, state, and federal legislation for farm safety. We have created an input survey within our website that analyzes the characteristics about a user's farms and produce, which determines a user's farm status using the proper Virginia Safety Regulations. The users' answers to this survey are processed by our database and backend functions that produce an output that directs the farmer to their own personal results with external links and supplemental information provided by Virginia and the FDA.

3.) Task: Understand, if they do not comply, how to change their practices and farms to follow the produce and safety rules

The next task is knowing when a particular farmer (user) does not meet the produce safety regulations. Here our input again is the survey the user has taken to provide our website with the proper information on their farm and how they grow their produce and raise their livestock. If our survey determines that they do not comply with certain safety regulations we need to provide them with the proper information on how they can adjust to meet these regulations and therefore resume selling their product through local vendors. The output will be provided by us through our database to provide the user tailored information on how they can adjust their farming practices to meet the required legislation and safety rules required for their type of farm. The database both provides specific information on all types of produce and livestock as well as farm size regulations which have been given to us from our client and will be scraped from an FDA website to be stored. The whole idea is that with quickly changing rules and regulations we want to keep our farmers up-to-date so that they aren't wasting time adjusting their business model and can get back to growing and selling their produce and livestock.

Farmers Market Vendors

1.) Task: Access information regarding Produce Safety Rules, Preventive Control Rules, and other related regulations

Access to this information as mentioned in previous tasks relates to our database system and how we plan to manage it. We will be using SQL and a Relational Database Management System to keep track of users and the proper information required for farmers market vendors. Using the information provided from our users in the survey, we are able to extract, using our database tables, the proper supplemental information tailored to individual vendors. Vendors will have user access to the same regulations as farmers because they need to know what farmers they can allow to sell at their locations.

2.) Task: Understand whether or not their supply and distribution methods comply with produce and safety regulations

Like farmers, vendors also need to know whether they are meeting the proper regulations. Regulating how we supply and distribute locally sourced food is just as important as how we grow it. Vendors are responsible too so they also need to receive information from our database that lets them know if they are following the proper safety practices. Our database will give tailored information to vendors as well, based on their survey results. We are able to differentiate famers from vendors and therefore have the proper information to make sure both types of users are following the proper regulations.

3.) Task: Understand, if they do not comply, how to change their practices and supply to follow the produce and safety rules

And lastly, just like farmers again, if our vendor users do not meet the proper requirements we want to provide them with the proper changes in practices so that they can become covered by Virginia and FDA safety regulations which gives them coverage. From our database, if our system detects that the vendor survey results in non-compliance, our system will provide them with best practices on supply and distribution so that they can adjust. The whole idea is that with quickly changing rules and regulations we want to keep our vendors up-to-date so that they aren't wasting time adjusting their business model and can get back to selling and distributing effectively throughout Virginia without confusion.

Web Administrator(s)

1.) Task: Better inform users of any changes to the produce and safety guidelines

The admin user is responsible for updating the database and supplemental information with the most recent produce and safety guidelines for Virginia farmers and vendors. Their input into the database is what keeps our users informed and focused on providing safe food to the people of Virginia. When there are changes in guidelines and legislation our system admin will quickly update any change information in our database and send notifications to all affected user's to make sure they are quickly made aware. Our team has in place functions that are triggered when an admin changes information in the database so we can efficiently inform our users in real time.

2.) Task: Answer any common questions users may have regarding the new produce and safety regulations

The admin is responsible for helping users through a vastly changing and fluid environment. Regulations are always changing, especially at the local level. Common questions can easily be addressed in the "frequently asked questions" page in our site. It is designed by our admin to hit all the most important and pertinent information for ALL farmers and vendors. We made sure to have this information easily accessible. There is no need to even have a user account to view this information because of its importance.

3.) Task: Provide users with resources that will help them comply with the new produce and safety regulations

The admin for our system is someone closely involved in the farming industry and the study of improved safety and techniques for having safe farming. Therefore, they have the accessibility and knowledge to provide the most valuable resources to help our users stay up-to-date and comply with new product and safety regulations as they are released. Having the real time update to database changes allows our admin to easily update changes without having to worry about essential users not receiving the proper information or safety updates. Our system is designed for seamless access to the most pertinent information for farmers and vendors to easily maintain their safety measures and continue selling and producing.

Finally, with all the information from above, we can then create a set of workflows of our system to ensure that all user goals are covered when accessing our website.
Persona of user(s) wanting to use the food safety site:

1) Unregistered users: want information on food
2) Registered users (typically farmers and farmers market vendors): Want a more personalized experience in addition to the features that unregistered users can use.
3) Admin: (Like our client) who can change the backend and which users of the website can access certain parts of the website.

Set of goals and workflows for the users described above:

1) All users: Find safety information about a particular food item.
    a) Get search input
    b) Search through database based on search input
    c) Display search results
    d) Wait for user input to select the food item
    e) Redirect user to information based on user input

2) Unregistered user: Register for website
    a) Display registration form
    b) Wait for user submission
    c) Save form information
    d) Give user option to complete survey to personalize profile

3) Registered users: Login to website.
    a) Give users the option to login using Google account or an already registered user account on our website
    b) Authenticate user credentials
    c) If successful, redirect the user to the home page and log the user in. If unsuccessful, send error message
4) All users: Show survey results. (On whether they comply with results or not)
    a) Display survey form with user input.
    b) Save user input on submission.
    c) Use input to compute results using our database.
    d) Display results based on computations.

5) Registered users: Save survey results
    a) Save any previously completed surveys to database
    b) Display survey results in profile.
6) Admin: Change database
    a) Create a separate portal for administrators within navigation bar
    b) Display the names of tables within the database, e.g.: Users, Food, etc.
    c) Show search bar after user clicks on table
    d) Wait for search input
    e) Search for item within the clicked table and display to user
    f) Give the admin the option to delete or edit items
    g) Save changes to database
7) Admin: Create a new item within database
    a) Create a separate portal for administrators within navigation bar
    b) Display the names of tables within the database.
    c) Show search bar as well as create new item button after user clicks on table
    d) If a user clicks on create, create a separate form with the needed fields within the table for the user to fill.
    e) Wait for user submission.
    f) Add entry to the clicked table and save.
8) Registered Users: Change profile information
    a) Display current profile information within the user profile.
    b) Give the user the option to edit their profile within their user profile.
    c) If a user clicks on edit profile, redirect them to a separate form with their current profile information.
    d) Wait for the user to make changes to form and click on submit.
    e) Update the information in the user table after a user clicks submit.
    f) Display the new information on the user profile tab.

Overview

This project system is meant to be a hosted website, using HTML, CSS, and Bootstrap for the aesthetics and front-end development, on a Node.js and React.js development environment for the backend and server-side development. The website is hosted a Ubuntu 18 server.

Getting Started

Before cloning the repository, you will need the following downloaded on your computer:

- The most up-to-date LTS version of Node.js (v12.16.1 as of 4/2/2020)
    - https://nodejs.org/en/download/
- Git for Desktop
    - https://desktop.github.com/
- MongoDB and MongoDB Atlas
    - https://www.mongodb.com/download-center/community
    - https://www.mongodb.com/download-center/compass

Once you have all of the prerequisite software, type out the following commands in your terminal of choice:

1. git clone git@code.vt.edu:tim98/cs4624-foodsafetysite.git
2. In the project root folder of the repository, run the command "npm install" in order to install the dependencies of the back-end code.
3. Inside of the project root folder there is a folder labeled "client". Navigate to the client folder, and run the command "npm install" in order to install the dependencies of the front-end code.

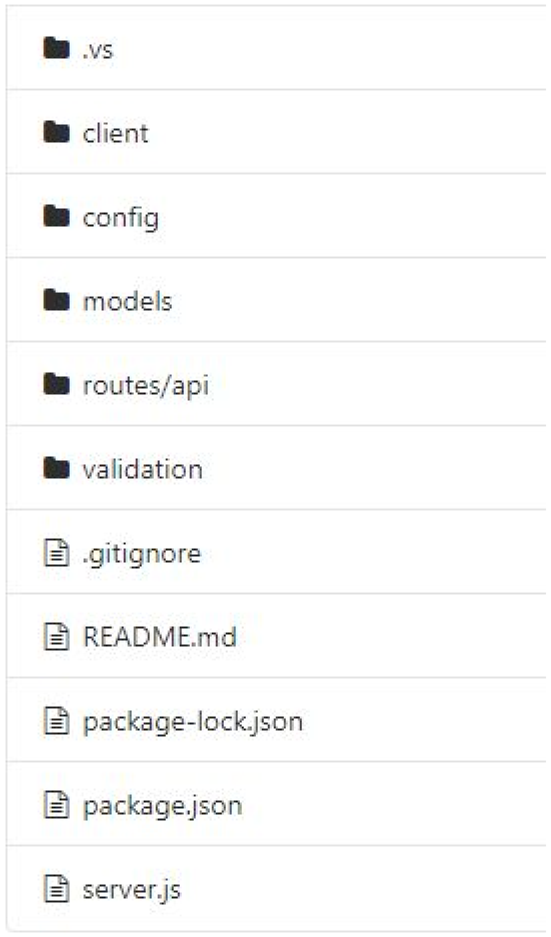
The first command will clone the repository containing all the files associated with this project. The second command will install all the necessary node_modules for running this development environment's server-side code. The third command will install all the necessary node_modules for running this development environment's client-side code. All node modules used for this website can be viewed in the package.json files inside of the available repository.

Running the Development Environment

Now that the repository is cloned, and all necessary node modules are installed, navigate to the project root folder and type the command "npm run dev" in your terminal of choice to start up the development environment. The web application will be available to view on your default browser, when navigated to the following hyperlink: http://localhost:3000.

Repository File Structure

The repository containing the files for this project has the file structure shown in Figure 28.



.vs
client
config
models
routes/api
validation
.gitignore
README.md
package-lock.json
package.json
server.js

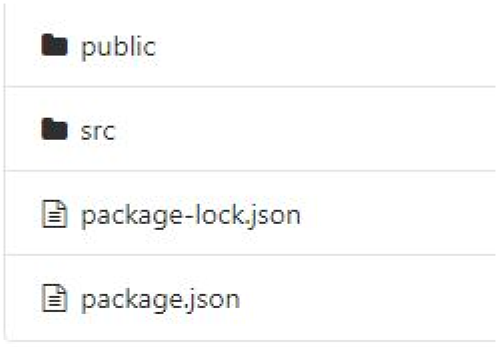
Figure 28: Repository file structure

The important directories and files to consider are detailed below:

1. Package.json - This file is responsible for maintaining the list of node_modules and dependencies that the server-side of the project relies on to function and operate properly.
2. Server.js - This file is responsible for setting up the Node.js and Express server, as well as the routes for basic access for CRUD operations on the databases. This file should be edited for adding new API routes for the website.
3. Config - This folder contains configuration files for connecting the database, as well as setting up the account system with passport.
4. Models - This folder contains the schema for the various databases used in this project, containing the structure of the databases, as well as MongoDB connections to said databases. This folder should be modified for adding/editing/deleting database schema for the back-end.

5. Validation - This folder contains validators for when users register and sign in to the website, ensuring that they comply with certain rule sets for creating and logging into an account.
6. Routes/API - This folder contains the files with the HTTP requests that the back-end will use to access the CRUD operations on the databases. This folder should be modified for creating/editing/deleting HTTP requests to the databases.
7. Client - This folder contains source code for rendering the web application, front-end components, and back-end functionality. When adding/editing/deleting pages and other components rendered onto the website, this folder should contain the files that need to be changed to do so.

Within the client folder, is where the folders for the front-end development of the website are located. This folder breaks down into the file structure shown in Figure 29:



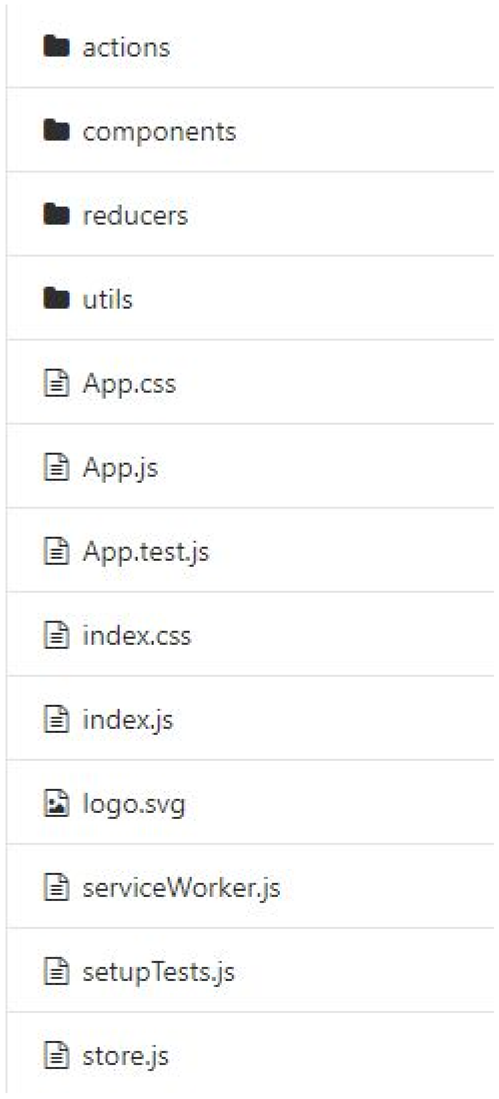
```
public
src
package-lock.json
package.json
```

Figure 29: "client" folder file structure

The important directories and files to consider are detailed below.

1. Src - This folder contains source code for rendering the web application, front-end components, and the connectors for connecting front-end to back-end (redux).
2. Public - This folder contains the entry point for all components of the src folder to render into, as well as files containing universal styling and operations for the whole website. When adding/editing/deleting CSS and JavaScript features that are universal within the website, this folder should contain the files that need to be changed to do so.
3. Package.json - This file is responsible for maintaining the list of node_modules and dependencies that the client-side of the project relies on to function and operate properly.

Within the src folder is where most changes, maintenance, etc. will be done on the project for the front-end and connecting it to the back-end. This folder breaks down into the file structure shown in Figure 30.



actions
components
reducers
utils
App.css
App.js
App.test.js
index.css
index.js
logo.svg
serviceWorker.js
setupTests.js
store.js

Figure 30: "src" folder file structure

The important directories and files to consider are detailed below:

1. Components - This folder contains the individual files that are responsible for creating the various pages of the website. When adding/editing/deleting pages of the website, this is the folder in which those changes should be made.
2. App.js - This file is the main React.js file for rendering the various pages of the website, depending on what page users want to access. It is the file that sets the routing for the various pages within the components folder, and will render the specified component based on the route selected by the user. When adding/editing/deleting pages of the website, the router in this file needs to be updated.
3. Actions - This folder contains the redux functions for referencing the back-end HTTP request calls that do CRUD operations on the databases: the redux functions that actually

connect the front-end to the back-end. This folder should be modified when a new connection from a back-end API call needs to be made with a front-end component.
4. Reducers - This folder contains the redux configuration files for pushing certain information from the back-end to the front-end page components within the components folder. This folder should be modified when a new connection from a back-end API call needs to be made with a front-end component.

Within the components folder, is where changes/maintenance to the front-end design/page layouts of the website will be done. This folder contains various JavaScript and CSS files that create each page of the website, as well as the navigation and footer bars at the top and bottom of each page. Each file is labeled for what page it is responsible for, and can be changed for page edits and maintenance.

Ubuntu Server

To host our site we used Phusion Passenger source edition with Nginx. Instructions that were followed can be found at https://www.phusionpassenger.com/library/walkthroughs/deploy/nodejs/ownserver/nginx/oss/bionic/install_passenger.html.

Our website is currently being hosted on a CS virtual machine provided to us by Professor Edward Fox. In order to access this VM and make changes to the website's production deployment, you will need to download MobaXTerm and SSH in with the following login credentials:

- DNS Name: foodsafety.cs.vt.edu
- IP Address: 128.173.237.145
- User: fox
- Password: This will be provided when beginning development.

Once you have SSHed into the VM, you will be able to configure the Nginx web server, as well as the production deployment of the website repository. The repository folder can be found at /var/html/.

In order to host the databases that we needed for the backend, we needed to create a MongoDB account. The login for the email that was used for this account is below:

- Email Address: foodsafetysiteCS4624@gmail.com
- Password: This will be provided when beginning development.

When asked to login on MongoDB, use the “Log In with Google” option and enter the above information to reference the database cluster that contains the needed databases.

Lessons Learned

Timeline/Schedule

Due to COVID-19 we experienced issues with maintaining our original schedule and timeline. Not only was spring break extended one week, but being unable to meet in person caused some delays in our schedule. To counteract this, we have prioritized what was most important to the client first, and then if the time allows, we would implement extra features. Due to time constraints, we were unable to implement the opt in services/real time notifications that Minh originally wanted. Another reason that our timeline shifted was because we had issues getting the necessary data from our client, which resulted in the backend team being behind schedule. This project taught us that it is not always possible to follow the original schedule completely, and to be flexible and adapt to any challenges that we face.

Problems & Solutions

- **Problem:** Establishing a communication channel between the team and client during unforeseen circumstances
- **Solution:** Reach out to the client and team members as fast as possible and find a common communication channel like Slack that everyone is willing to use.
- **Problem:** Explaining the requirements on the form of data needed for the implementation to the client.
- **Solution:** Give the client a clear example of an abstract form of data that the client can understand to replicate for requirements.
- **Problem:** Keeping the entire team updated on the progress of the current project.
- **Solution:** Maintain a shared repository using a site such as Github and post regular changes to the repository and to the common communication channel.
- **Problem:** Getting the requirements on the project prior to design without having to modify the foundation as we go.
- **Solution:** An abstract idea of the project is not sufficient before creating a prototype. It is important to question the client during any confusion and get a comprehensive understanding of the requirements.
- **Problem:** Working on a shared repository at the same time prior to agreement on selected features.

- **Solution:** Use different branches for features and merge to the main branch after fixing any bugs.

Acknowledgements

We would like to thank our client, Minh Duong (**minhd16@vt.edu**) who is a doctoral student in Food, Science and Technology for allowing us to work on, and aiding us in, this project. His assistance in explaining the scope and requirements of our project was invaluable. Finally, we would like to thank Dr. Fox (fox@vt.edu) for his guidance throughout CS 4624 and helping us understand the steps needed to carry out a project successfully and assisting us when any problems arose.

References

Hartson, Rex, and Pardha S. Pyla. *The Ux Book: Process and Guidelines for Ensuring a Quality User Experience*. Burlington: Elsevier Science, 2012, http://www.sciencedirect.com/science/article/pii/B9780123852410000117. Accessed 4 May 2020.

“React Top-Level API.” *React*, 2020, reactjs.org/docs/react-api.html. Accessed 4 May 2020.

“Survey Library Overview, SurveyJS Library Documentation.” *SurveyJS*, 2020, surveyjs.io/Documentation/Library. Accessed 4 May 2020.

Vermaak, Stuart. “Online Survey Software: Qualtrics Survey Solutions.” *Is My Farm Operation Covered or Exempt from the Produce Safety Rule?*, Virginia Cooperative Extension, 2020, vce.az1.qualtrics.com/jfe/form/SV_emnhR0UpFpiVvlr. Accessed 4 May 2020.