



WPI

USDA Foodborne Illness Outbreak Detector

A Major Qualifying Project
submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
In partial fulfillment of the requirements
for the Degree of Bachelor of Science

Submitted By:
Isabel Alvarado Blanco Uribe - Computer Science
John Carroll - Computer Science
David Leandres - Computer Science
Cole Noreika - Computer Science
Nick Vachon - Computer Science

Date:

2022-03-23

Report Submitted to:

Professor Elke Rundensteiner

Worcester Polytechnic Institute

Abstract

By collecting data from social media, news media, and government reports our project aims to develop an analytic big data tool to mitigate food safety risk by leveraging machine learning models to collect relevant data on foodborne illness outbreaks. Our application also serves as a web-based visual analytics tool, allowing users to explore results for early warning and discovery.

Executive Summary

This project dealt specifically with the identification, collection, and analysis of large data sources in order to create a system that is capable of displaying relevant data pertaining to foodborne illness. The eventual goal of this system is to provide a framework to potentially predict such outbreaks using deep learning models and alert users within a certain area that foodborne illness has been detected.

We began the project by addressing the requirements of such an application. We would need to identify sources of relevant data, as well as determine whether we were able to collect and use such data. We chose to utilize a web crawler to collect this data from two specific sources to start, the USDA official website as well as iWasPoisoned.com, a page designated for reporting food poisoning. We came to find that we would need specific permissions to use data from both of these sources that extended the scope of this project, so we chose to look elsewhere for data collection. Using our Twitter API key, we drafted multiple versions of Twitter scrapers to collect Tweets relevant to food poisoning, as well as other relevant data attached to the Tweet for analysis such as text, geolocation, and user id.

The next phase of this process was to design a database schema to hold all of our data. After collecting our initial elements of data, we observed that our primary source of data was going to be Tweets, and there would need to be a focus on geolocation data for these Tweets. For this reason, our schema is structured around storing all of the attributes of each Tweet.

The final phase of this project was a method of presenting our data collection. We chose to design the front end of our application through ReactJS and deploy it using Apache Web Server on our WPI machine. Of the several data visualization tools that our application offered, the tracker map relied on using the county of specific food poisoning instances in order to

display data on a choropleth map of the United States. Although county was not an attribute we could scrape from Twitter, we did have access to a set of three bouncing coordinates, which enabled us to average the coordinates for a centered latitude and longitude and use an API call to convert these coordinates to counties for each piece of relevant data that would be displayed on our map.

In the end, we have a thorough Twitter scraper and model to collect and analyze Tweets to find relevant data for cases of foodborne illness. This data is stored within our database, which is utilized by our website deployed at usda-foodpoisoning.wpi.edu. Our application features a Timeline of historical instances of foodborne illness outbreaks, infographics that provide insight into the size and common keywords found in our dataset, as well as a choropleth map of the United States that will show the number of Tweets collected per county that have been identified by our model to contain a relevant report of foodborne illness.

Table of Contents

Abstract	1
Executive Summary	2
Table of Contents	4
Introduction	6
1.1 Project Goal	6
1.2 Objectives	6
Background	7
2.1 Food Poisoning	7
2.2 Prior Work	8
2.2.1 iWasPoisoned.com	9
2.3 GeoTagging	11
Methodology	12
3.1 Database	12
3.2 Data Collection	16
3.2.1 Twitter API	19
3.2.2 Twitter API experiments	21
3.3 Geotagging	21
3.4 Data Processing	23
3.3.1 Preprocessing	23
3.3.2 Post Processing	24
3.5 Backend Pipeline Development	25
3.6 Front End	26
3.6.1 Tools	27
React JS	27
Bootstrap 4	28
3.6.2 Website Design	28
Home Page	29
Explore Page	32
About Page	32
Meet the Team	32
Our Partners	33
Project Breakdown	33
Tracker Page	33

3.6.3 Design Process	35
3.6.4 React Components	37
Navigation and Footer Bars	37
Home Page Widgets	37
Home Page Infographics and Not Available Icon	38
Word Cloud	39
Dropdown Buttons	40
3.6.5 Pages	41
Tracker Page	41
Admin Page	42
Home Page	43
Learn Page	44
About Page	44
3.6.6 Deployment	45
GitHub Pages	45
Apache Web Server	45
4. Findings	46
4.1 User Study	46
4.1.1 Use for User Study	46
4.1.2 Breakdown of Study Process	46
4.1.3 Results of the Study	50
Future Work	54
5.1 Map / Tracker Tool	54
Appendices	57
Appendix A: Figures	57
Appendix B: IRB Approval - IRB-22-0410	74
Appendix C: User Study Survey	75
Appendix D: User Study Relevant Responses	83
Appendix E: Repository	90
Appendix F: Graphs	91

1. Introduction

1.1 Project Goal

The goal of this project is to continue the development of an analytic big data tool to mitigate food safety risks by leveraging machine learning models to make predictions. We collect and analyze data collected from social media, news media, and government reports to predict potential foodborne illness outbreaks before traditional outlets such as the Center for Disease Control (CDC). This information is disseminated through our website and potentially informs those at risk. Our website also serves as a web-based visual analytics tool, allowing users to explore results for early foodborne illnesses outbreaks.

1.2 Objectives

The overall project objective is in the process of being constructed through multiple stages. In the early stage, we explored how food borne illness outbreaks often start through observing their symptoms, infection patterns, and common foods related to them. In addition, our team worked on a machine learning modal that would identify social media posts related to food borne illness, and collect it for later use.

Using identifiable keywords, our system was modified to better organize collected posts into groups related to multiple categories. These include symptoms, foods, dining establishments, and geographical locations. By uploading dissected versions of these posts to a data server, our team was capable of accessing information about potential food borne outbreaks on a daily basis.

Next, our team needed a platform to display, and visualize the data. Our team began working on a website that would provide the tools necessary for a user to view our data. Through this platform, they can also learn more about food poisoning, how it starts, and look at its history through our interactive timeline tool. Included on the site are areas where we would provide the next stage's tools.

In the following stage, our team would create tools that would help visualize the daily potential outbreak data. This would give the user a way of seeing current food borne illnesses, and allow them to make their own educated assumptions about regions of dense outbreaks through our heat map tool. In addition, we created a word cloud that would provide a fun, and visual way of recognizing the relevance of certain locations, foods, and symptom keywords throughout time.

In future stages, we hope that teams will take this framework, and use it to create a system that can predict food borne illness outbreaks before they happen. We estimate this is possible through social media post pattern recognition, as well as using concurrent outbreaks patterns throughout history.

2. Background

2.1 Food Poisoning

Throughout history, foodborne disease outbreaks have devastated thousands of communities throughout the country. Contaminated fresh produce has led to illness incidents, including sickness, hospitalizations, and even death (CDC, 2013). The appearances of these outbreaks are often undetectable under normal circumstances, however, the rapid growth in

technology in recent years has made it possible to detect these outbreaks early enough to stop them from worsening. Data analysis tools can observe social media posts related to foodborne illness, and use them to catch potential outbreaks before they can happen. In addition, this information can be organized and presented in a way that can be understood by the general population, in order to inform them about the potential dangers of foodborne outbreaks.

2.2 Prior Work

Our goal is to communicate food poisoning outbreaks to the public before the CDC officially reports them. To do so, the project is divided into five different objectives. The first objective is to crawl social media, news media, and the CDC websites to collect real-time data on food-borne illnesses. The second objective is to design a data model to process the collected information and identify relevant information. The third objective is to analyze the data to discover patterns of events related to food poisoning. The fourth objective is to design machine learning models based on the data we analyzed. The final objective is to design and implement a web-based visual analytics tool to communicate our findings and our data to the public. Further information on the objectives can be seen in Figure 1.

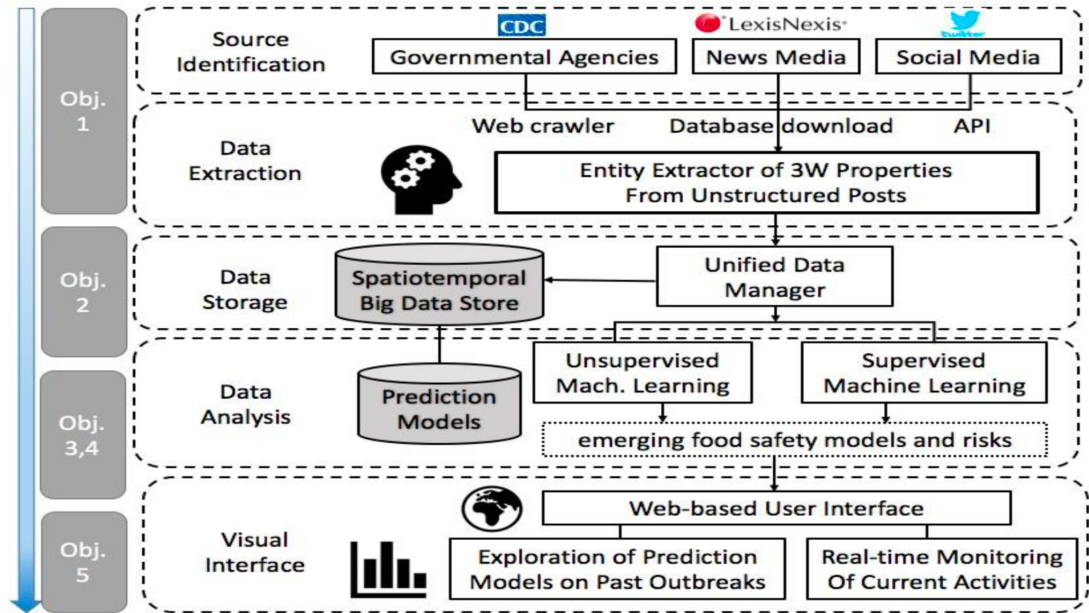


Figure 1: Depicts the initial project scope and objectives defined by the previous groups working on the project.

The machine learning model was developed prior to our team taking over the project.. It utilizes RoBERTa, a model optimized for use with BERTs (Bidirectional Encoder Representations from Transformers), which is a data structure made up of strings. The model’s training was outsourced to a crowdsourcing website, allowing users to identify the food, location, symptom, and foodborne illness keywords of tweets which were previously collected. Some minor changes were made to the model to accomodate for dependency differences in the server environment, but it was largely left unchanged.

2.2.1 iWasPoisoned.com

One site which was a great resource for our work was iWasPoisoned.com. On this site, users can view and submit reports about food poisoning they have experienced. The simplicity of

the UI and report system along with the relevance of the stored data made iWasPoisoned useful in multiple sections of our project development. One example is the map utility for displaying reports, as seen in Figure 2. This map groups several reports in the same general area to display hotspots, which is similar to the gradient map we plan on creating. The reports themselves were also extremely useful to collect as data, as explained further in the Data Collection section.

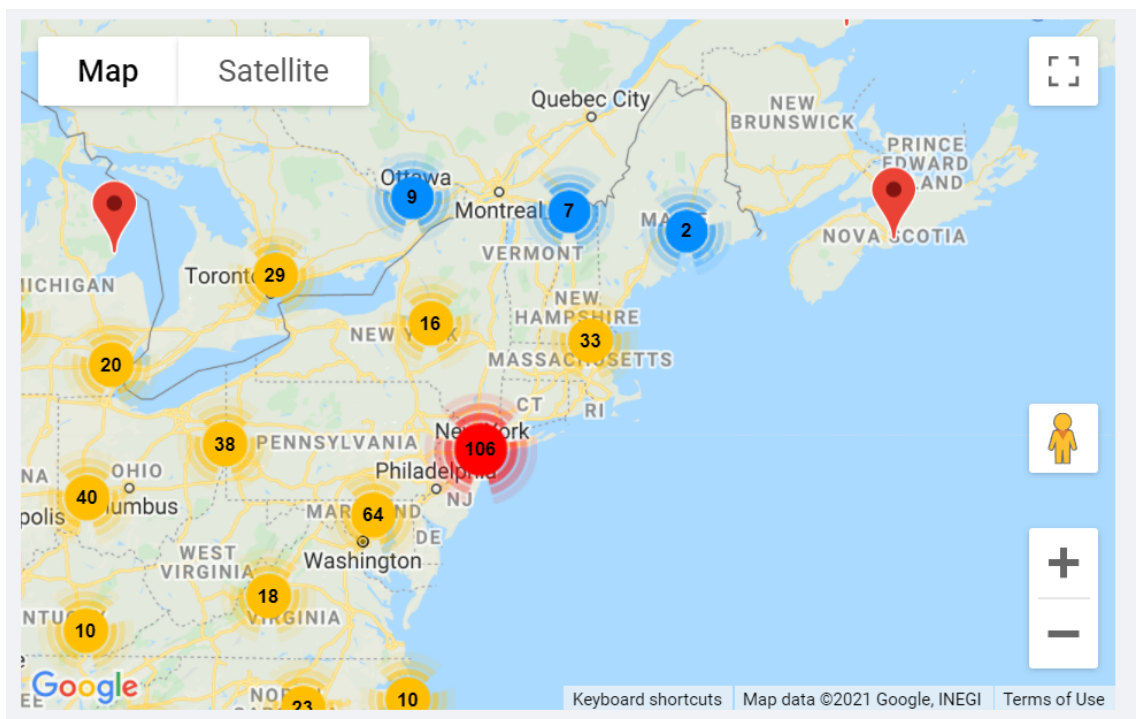


Figure 2: This is a map from iWasPoisoned.com and serves to display reports aggregated into hotspot zones across the United States.

2.3 GeoTagging

Former literature on the topic was split into two categories, text-based and network-based. Text-based methods sought to find the location of the tweet by assuming that language has a geographical bias and that by identifying it you can likely identify where the user is. The issue is however that this method requires highly complex algorithms and it has been found that the larger the data the worse the performance. In the researchers' implementation, they used a logistic regression model to predict the location, and features for each user were weighted using tf-idf, followed by per-user l2 normalization (Rahimi 3). As for network-based geolocation methods, these involve analyzing the user's network of different tweets and friends/followers online. Prior literature used only bi-directional @ mentions in tweets to signify offline relationships, however, researchers found these to be too sparse so they used unidirectional @ mentions in order to create edges in the graph/network (Rahimi 4). These researchers found that the most accurate results came from hybridization of the two models, where the more accurate methods hold higher weight in the prediction than less accurate methods (Rahimi 5).

Other researchers sought to “automatically identify ‘location indicative words’ (LIWs), that is words that implicitly or explicitly encode an association with a particular location” (Bo 1046). In order to create a map to base their geolocation, researchers opted to go with a city-based representation as opposed to points or clustered grids. They acknowledged that it makes their methodology insufficient for rural areas but argue that it is still better than points as they're too specific, and grids become too inaccurate in diverse densely populated cities. In making their data set, researchers used langid.py to filter for English and Twitter tokenizers.

Then they removed all users with less than 10 tweets and any duplicate tweets by said users. LIWs had the following properties: “1. High Term Frequency (TF): there should be a reasonable expectation of observing it for a given user; 2. High Inverse City Frequency (ICF): the term should occur in tweets associated with a relatively small number of cities” (Bo 1051). They used Placemaker as a benchmark which was a geolocating software provided by Yahoo, however, it seems to be depreciated. In conclusion, the researchers found that there was a trade-off between geographical coverage and accuracy; the wider the area trying to be predicted was inversely correlated with accuracy. In addition to this, it should be noted that only geotagged tweets were used to make these predictions and there could be textual differences between geo and non-geotagged tweets.

3. Methodology

3.1 Database

When deciding on a database program we were faced with several options. The first of which was to decide between a relational and non-relational database. There are several different factors to consider when choosing between the two models. The older, more common type of database, is a relational database management system (RDBMS), also known as SQL databases. For RDBMS databases, data is stored in rows and columns inside separate tables. Each table has primary and foreign keys. Primary keys are unique identifiers in a particular table. Foreign keys are the same data in another table that allow for relationships to be established and related data from multiple tables to be queried. These established relationships ensure data integrity through what is known as referential integrity. Any time a primary key is going to be deleted from a

primary table, reference constraints require that all related records are deleted. This is important as it prevents records from being orphaned in secondary tables with no reference to them in the primary table. Data is then accessed and modified using a structured query language (SQL), which leverages the primary and foreign key relationship to relate data from several tables. These databases have high transaction ability capabilities and excel in use cases where trends are being extrapolated from the data.

Subsequently, there is another type of database known as a non-relational database, or a no SQL database. These databases were initially invented to combat the issue of large data sets and the lack of scalability present in RDBMS. However, there is not a single type of non-relational database; multiple exist such as document databases, key-value databases, column-oriented databases, and graph databases. Upon closer inspection, only column-oriented and document databases were found to be relevant to our use case. Column-oriented databases originally looked promising, as it is similar to RDBMS where data is stored in rows and columns. However, there was a lack of consistency in writing to the database which could skew our findings and predictions so it was disregarded. The other viable option is a document database that stores data in documents like JSON, BSON, YAML, XML, and plain text. In particular, document databases setup using MongoDB was of interest to us as their product is one of the only ones to support ACID transactions. ACID stands for atomicity, consistency, isolation, durability which are all standards of high data integrity. In addition to this MongoDB even has its own proprietary query language called MQL, which offers much of the same capabilities as normal SQL.

Given that MongoDB has almost parallel capabilities to the most common open-source RDBMS PostgreSQL, we decided to first try MongoDB. It offered greater flexibility in how we wanted to structure our data and horizontal scaling, unlike PostgreSQL. The first attempts to implement MongoDB to create a non-relational document database were successful in initializing a local database. However, when trying to set permissions for remote access there were several issues that became apparent. The most pressing being authentication issues stemming from obtaining a .pem file that contains both the TLS/SSL certificate and key. Given these issues, we had to weigh if the added functionality was worth dealing with the issues. Ultimately as a team, we decided that using PostgreSQL would be fine, as the data for the back end of the website should all be structured prior to being added. In addition, storage size constraint concerns were dampened as the data we will store will have undergone filtering through machine learning models. Integration with a variety of analytics tools is also available as PostgreSQL is one of the most widely used RDBMS. Lastly, it's easier to maintain high data integrity and consistency which is critical when making future outbreak predictions.

In the interest of retaining as much information as possible, we decided to keep all the columns from the prior collected data as a basis for our table. From there additional columns were added to store the additional data our work on the project produces. Specifically, we built a geo-tagging tool for the tweets that necessitated a profile location, state, street address, zip code, and county code columns as seen in figure 3. Additionally, our data pipeline parses out the individual food and symptom tokens, so these received their own columns as well. As for the front end, we decided it would be best to connect them to views as they update dynamically from the main tweets table. These views for the website data visualization tools are also depicted in figure 3.

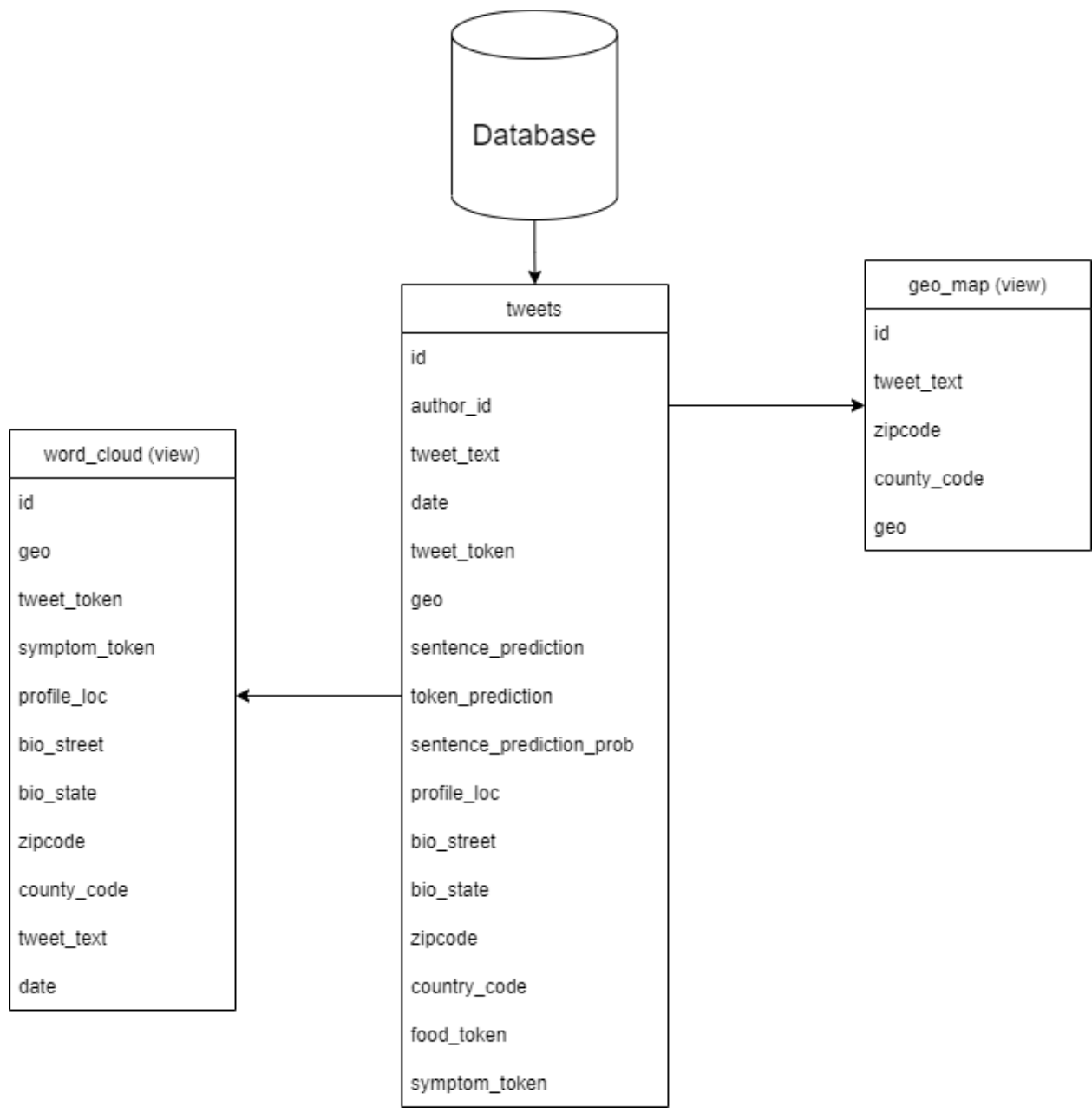


Figure 3: This diagram depicts the database schema design.

3.2 Data Collection

The main focus of our work this term was collecting and incorporating data from sites other than Twitter. This was done by creating a web-crawling program that was capable of locating food illness outbreak-related data on different websites, and understanding how to divide that data into predetermined properties to export as a CSV. To choose the websites for the crawler, we had to think about data other than tweets that could be useful for us. Because we plan to both have a live map of reports and display predictions based on historical data, it was important to find sources of data that were either rapidly updated or preserved data over a long period of time.

For the first criteria, a source of rapid and up-to-date data, we found the perfect answer in the form of iWasPoisoned.com. This site allows visitors to submit reports about food poisoning they have experienced. The report format is simple and collects for symptoms, locations, specific foods or ingredients, and a description. Reports are listed in the reverse order they arrive, so newer ones are always present on the home page. Configuring the crawler to work with [iWasPoisoned](http://iWasPoisoned.com) was fairly straightforward; the basic logic involved locating the HTML element containing the reports and iterating through each of the reports (also HTML elements) contained within to extract the report elements. The page number in the URL is then incremented and reloaded, repeating this process on the second, third, and so on, pages. On [iWasPoisoned](http://iWasPoisoned.com), the time of reports is given as “X hours ago,” or “Y days ago.” The crawler had a function that would iterate through these strings and convert the time to seconds, which could then be subtracted from the current time to get a timestamp. This can be seen in the below example.

```

TIMEFRAME_TO_MULTIPLIER = {"hour": 3600,
                             "hours": 3600,
                             "day": 86400,
                             "days": 86400}

# Extract the age of the report (x hours ago, y minutes ago, etc), grab current time, determine time report
was posted

currTime = datetime.now().timestamp()

age = report.find_element_by_xpath("div[@class='float-left report-title']/h3[@class='h6 card-subtitle
mb-2 text-muted']").text.split()

delta, timeframe = age[0], age[1]

deltaSeconds = int(delta) * TIMEFRAME_TO_MULTIPLIER[timeframe]

timestamp = datetime.fromtimestamp(currTime - deltaSeconds)

```

Figure 4: Crawler Iteration Code

Our second goal, an accurate source of historical food outbreak data, was found on the CDC website. The data on this site was officially reported food-related outbreaks, from the CDC themselves. This data was tab-separated by time frame and was archived back to 2006. The crawler configuration for this data was a bit more complicated. It begins by locating the HTML element containing links to the reports, then iterating and storing said links. We initially were not aware that there was a page specifically for foodborne outbreaks, so we had to program the crawler to differentiate between them and other unrelated outbreaks. To do this, the crawler would search for a banner image that is present on every report page. This banner was differently titled on each type of outbreak, and we could use this to filter out the non-foodborne reports.

```

type = crawler.find_element_by_xpath("//img[contains( @ title, 'Banner')]").get_attribute("title")

if type == FOOD_SAFETY_BANNER_STRING:
    container = crawler.find_element_by_xpath("//main[@aria-label='Main Content Area']")

    title =
container.find_element_by_xpath("div[@class='row']/div/div[@class='syndicate']/h1").get_attribute("innerHTML")

    date =
crawler.find_element_by_xpath("/html/body/div[6]/main/div[3]/div/div[3]/div[1]/div/div[2]/div/p").get_attribute(
"innerHTML")

reportSummaries.append([title, date])

```

Figure 5: Differentiation Code

While this method was effective, upon finding the correct page we were able to make a few changes to have the crawler simply iterate through the tabs to collect all the reports, much quicker than with the image-checking.

After the crawler was up and running, we needed a way to represent both types of data within a CSV or the database schema. We attempted to make it as simple as possible, with both reports being stored together to make it easier to move to the database later. The first attribute was source type, with 1 being an iWasPoisoned report and 2 being a CDC outbreak. The next two attributes are the relevant foods or ingredients the data is about, and any relevant text in the

data (i.e. report description). The location of the report is also stored, as well as the time the report was made.

3.2.1 Twitter API

On the data collection end, one of the very useful tools that we have access to is the use of the Twitter API, used to scrape for Tweets that fulfill certain keywords and time periods. Using this data, we can apply the models to scrape through all tweets containing relevant keywords and filter them to find data that shows discovery or prediction of reported foodborne illness outbreaks. There are two primary parts of the code that are utilized in the collection of our data. Below is the code that is used to collect all of the data from each Tweet. We are collecting account IDs, time of posting, geotagging, Tweet ID, language, as well as metrics, and text. All of these different properties can be used to sort, filter, and apply learning models to our dataset.

```
# 1. Author ID
author_id = tweet['author_id']

# 2. Time created
created_at = dateutil.parser.parse(tweet['created_at'])

# 3. Geolocation
if ('geo' in tweet):
    geo = tweet['geo']['place_id']
else:
    geo = " "
```

4. Tweet ID

```
tweet_id = tweet['id']
```

5. Language

```
lang = tweet['lang']
```

6. Tweet metrics

```
retweet_count = tweet['public_metrics']['retweet_count']
```

```
reply_count = tweet['public_metrics']['reply_count']
```

```
like_count = tweet['public_metrics']['like_count']
```

```
quote_count = tweet['public_metrics']['quote_count']
```

7. source

```
source = tweet['source']
```

8. Tweet text

```
text = tweet['text']
```

Figure 6: Tweet parts Code

Next, we move on to the filtering portion of the code. Here, we take a set of exclusive keywords that must be contained within the tweet, as well as a time range to select the tweets between. The maximum number of tweets currently supported by the API is 500 per call, so we have set that to the maximum value in order to collect as much data as possible.

```
keyword = '(egg OR hard boiled OR #egg OR #hardboiled) lang:en -is:retweet' #has:geo
```

```
start_list = ['2020-01-01T17:00:43.000Z'] #ISO8
#           '2021-03-21T00:00:00.000Z']

end_list = ['2020-06-01T23:00:00.000Z']
#           '2021-03-31T00:00:00.000Z']

max_results = 500
```

Figure 7: Keyword Code

3.2.2 Twitter API experiments

The next step of using the Twitter API was to begin running experiments to learn more about the datasets being collected and how we could modify our collection methods to produce the most meaningful dataset. To begin, we started by working with known food-bourne illness outbreaks that had extreme numbers of cases (close to or over 1000 reported cases). From this, we set the range of collected tweets from 10 days prior to the first reported case up until the date of the final reported case. Lastly, we looked through the data reported on the CDC website about the outbreak and used that accordingly to generate keywords that would provide the most data.

3.3 Geotagging

Another feature of the Twitter API is that you can collect and filter tweets by those that contain Geotagging information. One of the goals for our final website is to display a saturation map of the United States that would report on data regarding food-borne illness. Unfortunately, only about 5% of tweets collected through the API contain Geolocation information, so we have been looking into other methods of collecting location information. Initially, we had intentions of

using an open-source tool called Carmen, however, there were issues in attempting to implement Carmen into the data pipeline. Carmen was created a few years ago and was originally created to work with version 1 of the Twitter API, and updating it to work with the new version two of the Twitter API proved to be tedious. The way data is structured when it is queried has changed from one API version to the next, the geotag for tweets is no longer stored in the same JSON object as other Tweet attributes. Additionally, there are other profile attributes, like profile location or description/biography, that Carmen would leverage that have also been moved. Documentation for Carmen was sparse as well so modifying the tool seemed unfeasible and reformatting the data into the correct JSON format did not seem reasonable. The data needed for Carmen would require two separate queries and 3 different JSON files.

We decided to do a literature review of researchers attempting to do similar work in geo-locating social media posts. In doing so we identified several alternative methods to be able to geotag a tweet, and ordered them by the most likely to be accurate. Currently, the first alternative to a Tweet-specific geotag is to run a second Twitter API query in order to get the user profile data. Then we check if there is a profile location and if there is we use that as the location of the Tweet. Then we check the profile description to try and tokenize the description, checking each token to see if it is a location indicative word. In particular, our geolocating program looks for street addresses, state names and abbreviations, and for zip codes.

We then had intentions to do the following; however, with the rate limit imposed by the Twitter API, it just became unrealistic as we were limited to 300 queries per 15 minutes. Subsequently, check the users' other tweets and see if they happen to have a geotag. If they have 10 geotagged tweets then we consider using them as a location alternative if multiple of them are in the same location. Lastly, we check the profile location of the specific user's mentions because

we assume that typically people only mention people they know in their tweets. We account for Tweet mentions of people they don't know by excluding Twitter users that are verified or have large followings as these are likely to be public figures.

3.4 Data Processing

3.3.1 Preprocessing

We also added a few additional preprocessing functions to the data collection. One of these functions handled the conversion of the JSON geo-tag data returned from the Twitter API into the equivalent latitude/longitude pair. We did not immediately realize that this should be done in preprocessing because the coordinate-lookup dictionary is stored apart from the individual tweets in the API response. Once we discovered this data we were able to add code to look up the "place" value in the coordinate dictionary and save the values in their own parameters in the database.

This was further expanded upon with the development of a geolocation predictor. Once the coordinates are gathered for the already geotagged tweets, the author IDs of the remaining tweets are collected. From here, additional API calls are performed to get the author's profile information. As discussed previously in the report a variety of methods are used to try and predict a user's location, such as substring search in their biography or inspection of their set location.

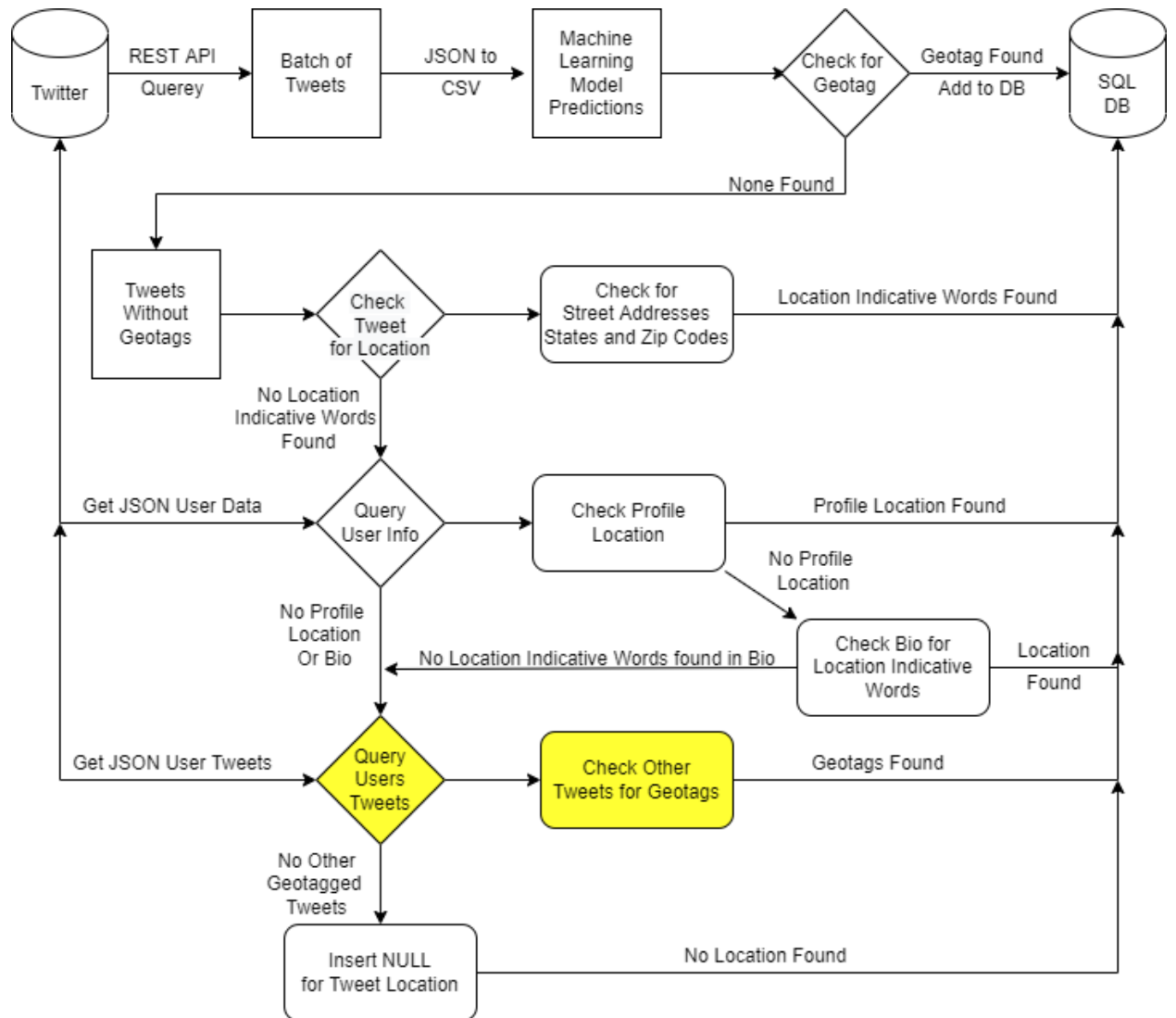


Figure 8: This is the data pipeline that shows how data collected from sources like Twitter is processed and geo-tagged before being added to the database. The sections in yellow were removed due to rate limit restrictions on the API, however, would still be beneficial in the future.

3.3.2 Post Processing

Additional post-processing was added to the model as well. For starters, we included the prediction confidence in the model output, allowing for the related tweets to be easily isolated

even as we experimented with different cutoffs for the confidence value. Functionality to parse and divide the token predictions into their own database fields was also added. This division allows for the tweets pertaining to specific foods or symptoms to be easily queried from the database, which is essential in creating dynamic data visualizations for the front-end. At this point, we discovered that the token prediction was case-sensitive, and would also sometimes include non-alphabetic characters. This caused issues with query counts being artificially inflated, as well as some queries not returning all instances of the desired token if there were unusual characters. However, this was easily remedied by filtering the token predictions to alphabetic characters, converting them all to lowercase, and removing any duplicates present afterward.

For the final processing, the code to inspect the authors' accounts, as discussed in the *Data Sources / Collection* section, is run on the data, adding its prediction on the account type to each entry. This is crucial as it allows us to distinguish between tweets which we should use to make predictions about foodborne illness outbreaks, and tweets that we should use to verify past predictions, and display as statistics in our front-end visualizations.

3.5 Backend Pipeline Development

Developing the server-sided pipeline was a major focus throughout this term. This process began by simply consolidating all the project code files into a single environment, and ensuring that they could run in conjunction with one another. This turned out to be a pretty painless process, only requiring a few changes to dependency versions and function inputs and outputs between files. With everything finally working synchronously, we decided it would also

be an ideal time to clean up the codebase: function descriptions were added, confusing code was commented, and any obvious optimizations were made. These changes all contributed to making the backend more understandable for all group members, as well as allowing for potential future groups to easily continue the project's work from where we left off.

The pipeline additionally includes automation functionality. A config file was created, containing values for all the input parameters. This includes the list of keywords to use in the Twitter API, the timestamp of the latest collected tweet, batch and epoch sizes for the prediction model, etc. Crucially, this config file also contains a value called “collection_rate,” which indicates how long the server should wait in between collecting data for Twitter. After concluding processing and storage on a set of collected Twitter data, the server will set up a Linux “cron” command using the collection_rate value and the intro pipeline function. This command schedules a job to run the pipeline, after the specified number of seconds. Because it is a scheduled job, and not hanging the program such as with a sleep call, the server is able to handle any calls from people using the website without issue.

Lastly, we set up a barebones command-line interface for interacting with the server. From this CLI you are able to alter the config values, immediately run the entire model, or run individual components and inspect the outcomes.

3.6 Front End

3.6.1 Tools

React JS

When designing an application of this scale that contains so many components that will need to work seamlessly together, we decided that building this application on the ReactJS framework would be in our best interest. There are several reasons for this decision, many of which had to do with the efficiency and flexibility of the framework. To begin, React JS is a JavaScript library that was created and maintained by Facebook, or now Meta. It enables users to build easily scalable front-ends for web applications with a flexible and completely open-source JavaScript library. ReactJS is also one of the most popular front-end frameworks used by Fortune 500 companies, so it would also serve as a great learning experience to apply our prior knowledge and learn this framework. It is important to break down all of the benefits that this framework made to our project, and how it can continue to support the future of this project as well. To begin, the flexibility of ReactJS was one of the very attractive features of the framework. It is modular in structure and comes very familiar to students like us that have a strong background in object-oriented programming. In addition to this, a project that has several individually operating components translates easily to such a framework. The next benefit that we observed was the speed. React allows users to run individual components of the application on both client-side and server-side in real-time, which significantly decreases the development time needed for a front-end application. In addition to this, components are all built individually, meaning that flaws to one component will not break the entire application. This makes both debugging and development a much smoother process. Performance is another huge benefit to ReactJS. The virtual DOM and server-side rendering is done by react-enabled applications, regardless of their complexity, to compile very quickly. To add to this, ReactJS is very

user-friendly and offers countless benefits to both the developer and end-user from a usability standpoint. Another benefit to React that we appreciated but did not get to fully take advantage of for our project was its flexibility for mobile app development. Despite the little focus we put on mobile development, our application runs fairly well on mobile due to the flexibility provided right out of the box from ReactJS.

Bootstrap 4

As seen in all of the design and front-end prototyping of our project, our proposed website is very detailed and dynamic. Due to this, developing the website from scratch would take an unreasonable amount of time, and we would need to utilize many of the modern JavaScript frameworks and libraries to assist us in our development process. For this reason, we looked into Bootstrap 4, a free and open-source tool collection that is used to build dynamic and responsive websites and web apps. This framework is commonly utilized for making mobile websites and having diverse browser compatibility. Bootstrap also supports a variety of visualization libraries, such as D3, and would significantly improve the development of our website (Kumar, 2021).

3.6.2 Website Design

In order to visualize our results in the most effective way, our team conducted research on how to best design the website application. Our goal was for the user to observe, understand, and learn from our data in the most intuitive way possible. Therefore, we decided to divide our website into four different sections: the home page, the about page, the dashboard, and the explore page.

Home Page

Our goal for the home page is to inform the user on what our project is, what we have done, and why it matters. Therefore, we decided to have a data visualization on the home page displaying the data we have analyzed and a section that features the different sections of our website. The visualization that we display is dependent on the Twitter data that we collect. If the data collection allows for us to display geographical location, we would like to have a saturated graph on the home page. If not, we plan on having a timeline that will provide a glimpse into what our Twitter data offers in terms of food poisoning outbreaks throughout time.

The timeline is unique in that it is capable of showing the element of time within our research. We achieved this by displaying a box containing the outbreak, and the duration of the outbreak over time determines the box's width. The outbreak's location on the timeline is determined by its starting date, as seen in Figure 5 below.

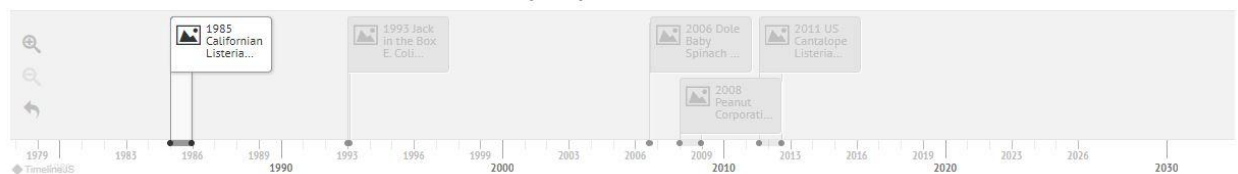


Figure 9 - Timeline example obtained from our live MQP Site.

Hovering over an event on the timeline will display a pop-up with a breakdown of the event. If a breakdown is not available, the function should instead display the news article or tweet that is related to this outbreak or event. Lastly, clicking an event will cause its information to slide onto the top of the screen above the timeline, allowing the user to observe more data than what is provided in the popup.



Figure 10: Timeline slide in information example obtained from our live MQP Site.

Another visualization that we want to display on the home page is infographics. We plan on designing and programming two dynamic infographics on the home page. The first one addresses how much twitter data we have collected. The second one visualizes the relevancy of food poisoning displayed by the number of recent food poisoning outbreaks in the United States. We would gather this data from our database and from crawling the CDC website. Further down on the home page, we plan on displaying static informative infographics about food poisoning. We would display three different types of static infographics, one addressing what food poisoning is, another displaying the different severities of food poisoning symptoms, and another one advising different steps to take to prevent food poisoning. All this information would be gathered from reliable sources such as the Mayo Clinic and CDC websites.

The use of infographics would allow users to quickly process the information, since people process visuals all at once, unlike written text which is processed in a linear way (Smiciklas). With the constant shortening of peoples' attention spans, especially in younger generations, displaying information that is quickly processed is one of our top priorities. For the first infographic, we will show the progression of where food poisoning comes from to someone experiencing symptoms of food poisoning. For the second infographic, we want to display the spectrum of food poisoning symptoms using a color spectrum of yellow to red. We picked two colors instead of a rainbow spectrum because rainbow maps are known for being bad for accurately visualizing data (Moreland). For the third infographic, we want to display, in steps, what people can do to prevent food poisoning. The steps would be shown in progressive order of someone getting to a restaurant to return home with leftovers.

On the home page, we would also be directing users to the different features the website has to offer, such as the about page, the tracker, and the explore page. The buttons with action items will incentivize users to keep on exploring the website, as displayed in Figure 7.

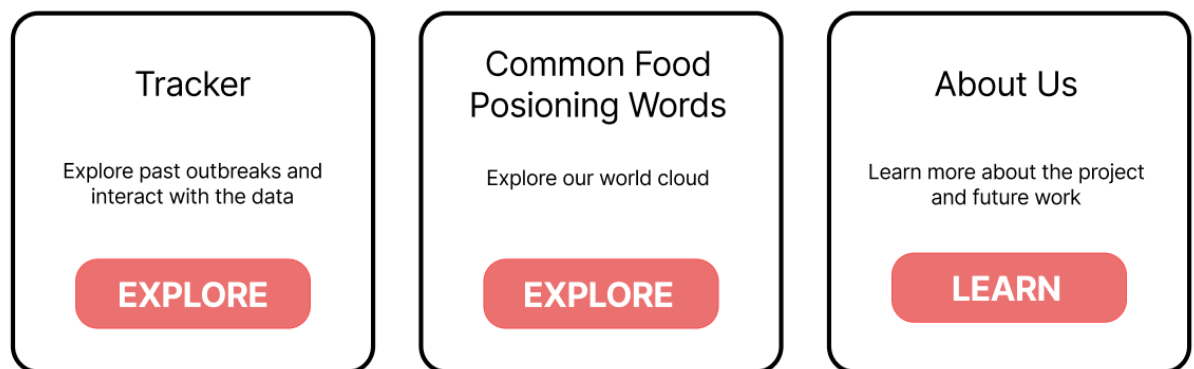


Figure 11: Buttons that lead to the different pages on our home page.

Explore Page

Another section of our website is the explore page. The goal for this section is to allow users to explore the data that we have made available to them through visualizations. In order to do so, we have decided to include a word cloud where users are able to pick the food poisoning outbreak they want to explore and see the most common words that were used on Twitter. Below the word cloud, users can learn more about what food poisoning is through infographics. These visualizations would be more informative than the ones on the home page, where we could inform users of interesting findings that we have found while working on this project and the relationship between Twitter and food poisoning outbreaks.

About Page

The About page originally displayed information directly related to foodborne illnesses, alongside historical facts and symptom data. However, the Home page already achieved this, and so we assigned the About page to another task. Information regarding our team, the timeline of work progression, and sponsor/partner descriptions were important to include. The recreation of the About page is now divided into three sections.

Meet the Team

This section is an area where the user can see the faces of those who contributed to the project. Under each name, the user can find the degree, major, and university of the individual. Project contributors come from three main groups, being the undergraduate MQP team, graduate students alongside the project advisor, and UIUC representatives.

Our Partners

This section is dedicated to informing the user of the project sponsor, and our UIUC partner's involvement in the project.

Project Breakdown

Lastly, it was important to provide a means of explaining the progress of our project, and how it worked. This section provides an image and a breakdown of the different elements in the project. Also included, is a flow of how everything works together to accomplish the task of the assignment.

Tracker Page

The remaining tools will appear on the Tracker Page. Here, the user is able to observe the tracked data from our sources. This is the page where maps, charts, and outbreak descriptions will go. The tracker's main purpose is to help the viewer visualize outbreak information, using our heat map tool. This tool will display a saturation map. The darker an area on the map, the more cases that have been detected by our system in that area.

A saturation map is a tool that utilizes location data in collected sources and uses to visualize occurrences by location. Since people by default have a dark-is-more bias, where darker colors are perceived to have higher quantities in data, the darker areas in the map represent densely populated areas of interest (Sibrel). The "interest" can be determined by the currently observed dataset the user is viewing. In our map, a viewer can click a county in the United States. Then, an event window will pop up, displaying the location, and the number of outbreak cases in that country.

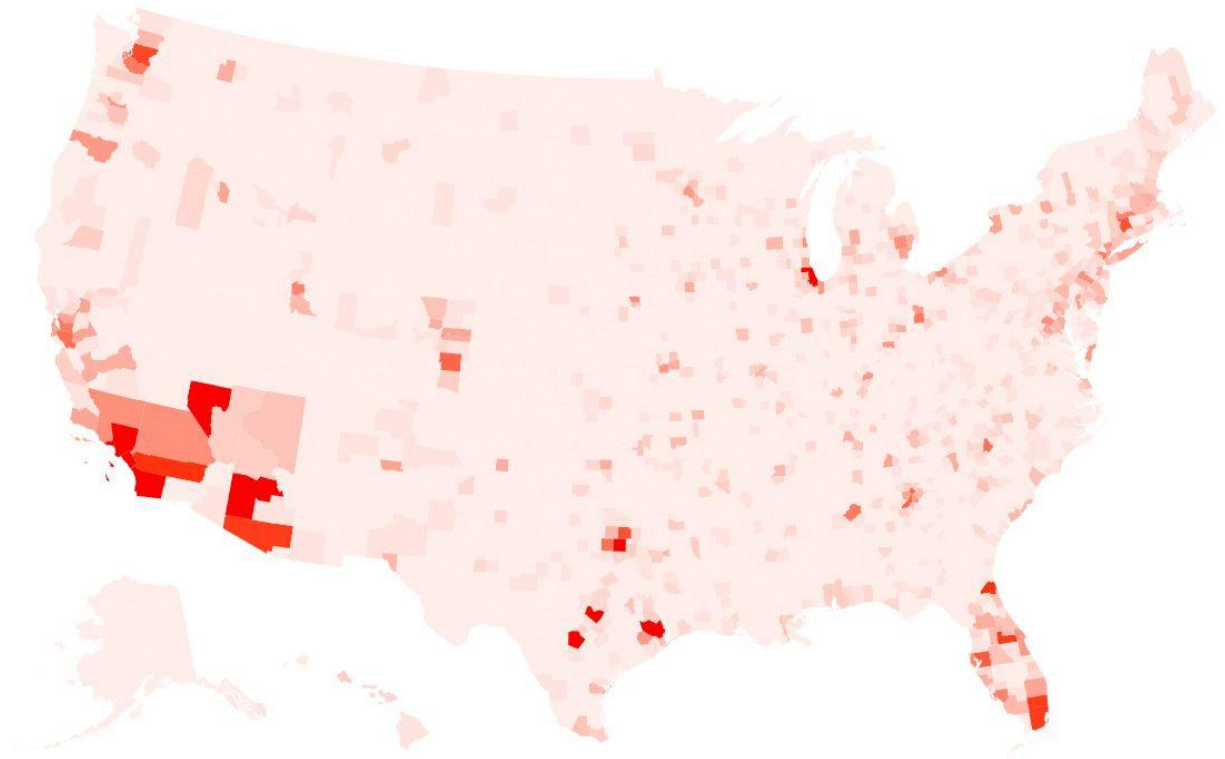


Figure 12: Heat map of user cases found by our system based on tweet information.

Some other ideas of tools that can be implemented on this page came up during development. In future stages, the following could be great additions that would allow users to have more flexibility when observing data.

1. Time manipulation: The user can change the time span in which they wish to observe collected data. The visualizations should change as the time period changes. This functionality is to be built into the searching feature within the tracker page.
2. Prediction model (May or may not be implemented): The user will be able to observe our system's predictions with this tool. The prediction should use the keyword to calculate its prediction.

3. Line Graph: The line graph tool will show change over time. The user is able to switch between case count/outbreak count over the span of time they selected.
4. Bar graph: This is meant to show data in relation to other schema types. For example, one may be able to observe the number of cases with salmonella caused by chicken.

3.6.3 Design Process

Before designing, we explored existing visualization options and analyzed how effective they were. The main websites that we referred to were the CDC website and <http://matters.mhtc.org>. When analyzing the examples, we referred to Steven's Psychological Power Law and Cleveland & McGill's experiment results. Steven's Psychological Power Law indicated that people perceive saturation and length better than area and depth. This led to us picking a timeline and a saturation map for our main visualizations. Cleveland & McGill's experiment ranked the visualizations where people were able to compare quantities the best. The best visualization to do so was the bar graph, which we decided to incorporate into our website. We pointed out the elements that we thought were effective and ineffective, as can be seen in Figure 8.

3.6.4 React Components

As previously discussed, one of the several benefits to using ReactJS is that the developer can separate out the pieces of their application into individual, reusable components. This saves time, prevents repetitive code, and allows for easier collaboration when developing the website. Our project used several components, each of which served a unique purpose in the usability of our application.

Navigation and Footer Bars

Our first two components are the Navbar and Footer bars, which were reused across all the pages and served as the primary navigation tools around the application. Both of these elements utilized a reactive, easily understandable design that allows the user to navigate between pages from either the top of the screen or the bottom, adding to the overall usability of the page.

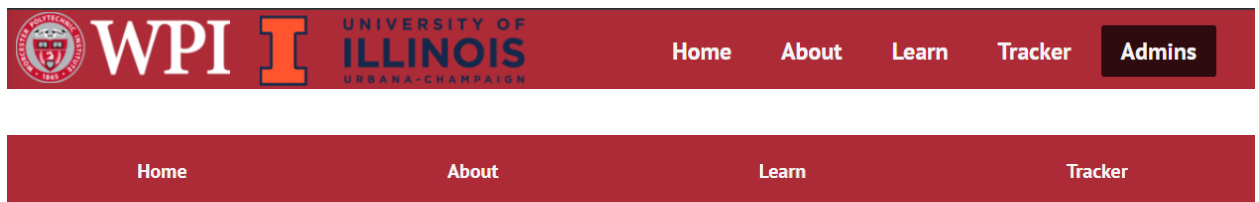


Figure 14: The navigation bar (above) and bottom banner (below) from the website.

Home Page Widgets

Another component that we included on our home page was three different reactive widgets that gave a brief explanation of the different pages that our application offered, as well

as a navigation link. By including images in the widgets, users are able to get a quick grasp of what the website offers and what features the user can interact with on the different pages.



Figure 15: Three widgets located on the home page.

Home Page Infographics and Not Available Icon

Another component that we included on our home page was the two infographics that feature quick facts about our project. On the left hand side, we have an infographic that shows how many tweets have been analyzed by our machine learning model to identify food poisoning outbreaks. Currently, this infographic is connected to the backend, running a count query to the database to see how many tweets are currently stored in our tweets table.

On the right hand side, the infographic shows how many current food poisoning outbreaks have been identified by the CDC. The right infographic is not connected to CDC data right now, thus featuring an icon that the user can hover over it that will display a “This feature is currently not available” message. The component consists of an image, a number that visibly goes up to the provided number, and text explaining the number. If the data is not available, it will display the Not Available component.

The Not Available component is there to inform users that the information that they are seeing is not connected to our database. This indicates the features of our website that would be recommended for future work. It uses the MUI Tooltip and Icon Button React libraries. We decided to use an external information icon to incentivise users to hover over it.



Figure 16: The two infographics located on the home page featuring the Not Available component.

Word Cloud

The word cloud that is in the website is meant to allow users to explore the data in a different way. They can see which are the most common symptoms, locations, and ingredients from the tweets that were collected. The user would be able to select what type of data they want to see by using a dropdown menu. Since we are collecting a large magnitude of tweets, we need to set a timeframe and a limit of what tweets we are going to count and display on the screen.

Currently, the word cloud is connected to a local server that connects to the database. There is filtering and modifications that are done on the front end to make the data usable for the word cloud.



Figure 17: A sample of the word cloud.

Dropdown Buttons

Lastly, there is a dropdown component that is added to the Navbar when the width of the page is not wide enough to house all of the different buttons. This provides a dropdown for the user to select a page and also contributes to the functionality of our application on mobile as well. These four components work together to provide optimal ease of access to the user while also providing a description of each page so that the user is able to navigate to the correct place.



Figure 18: The dropdown buttons, displaying what it looks like when you click on them

3.6.5 Pages

Tracker Page

The tracker page of our application utilizes a visually appealing and interactive method to display the data that was collected and curated on the back end. To begin, the map is generated using a JavaScript library called react-simple-maps. This library allowed for many modifications to the data being provided, although it had limited customization for the map itself. The data was then pipelined from our database into a format that could be read by the map. This map broke up the United States based on county, so we needed to have data on the total cases and specific Tweets that are written from a given county. After this, a gradient is set to determine the darkness of the color for each county based on the data provided. At first a linear gradient was used, and the map looked much more concerning than the reality of the number of cases that were reported. To alleviate this potential concern to the user, we chose to utilize a gradient that is exponential and leans towards the lighter colors, and counties will only display a deep color if there is a significant increase in reported cases over the other counties. The map features zooming and panning functionality, and a dynamic bar on the top that will display county-specific data when the user clicks on a county. Unfortunately, the functionality of this map was limited by the capabilities of the JavaScript library, and a new map framework was drafted and discussed in the Future Work section.

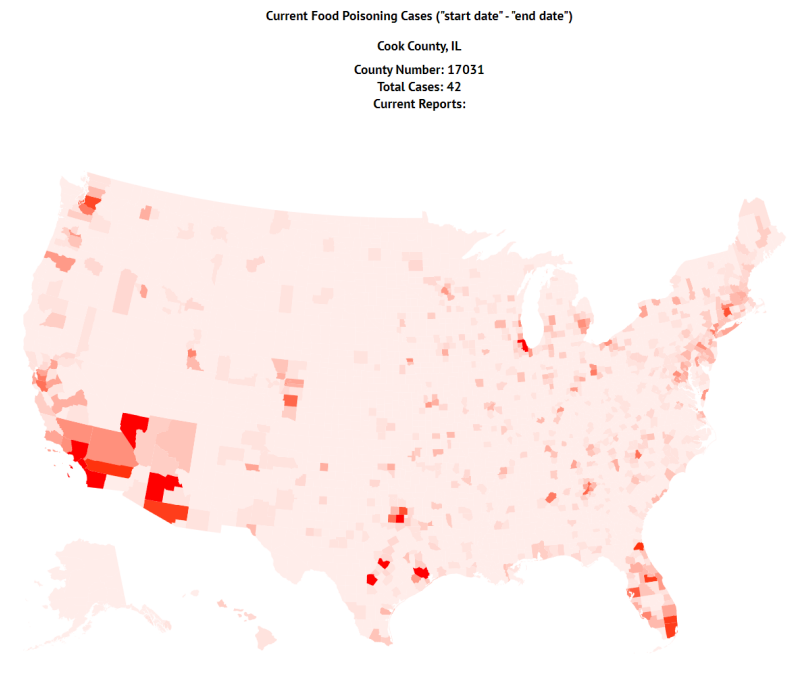


Figure 19: The map from the tracker page on our website.

Admin Page

The admin page is where the collaborators developing the website can have direct control of specific elements within the backend of the site.

Figure 20: The admin page from our website.

Home Page

The home page consists of many components. The most noticeable one is the Timeline visualization, and right below are the two infographics that display quick facts about food poisoning and the project. Then, there are three main infographics to teach the user more about food poisoning. These are made with components but they are not reused throughout the website. For the layout of the components, the flex command was used in CSS, which adapts for different screen sizes.

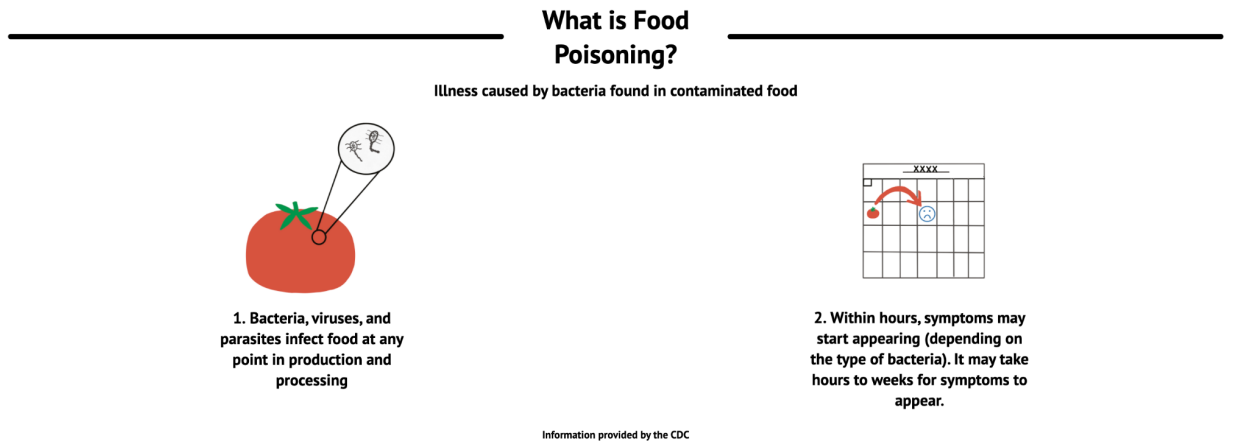


Figure 21: Food Poisoning infographic #1

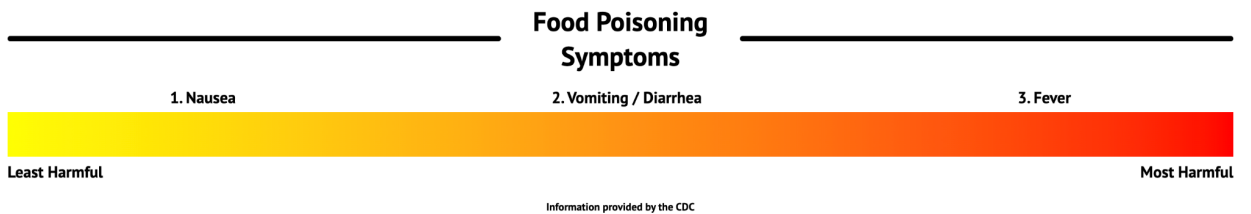


Figure 22: Food Poisoning infographic #2

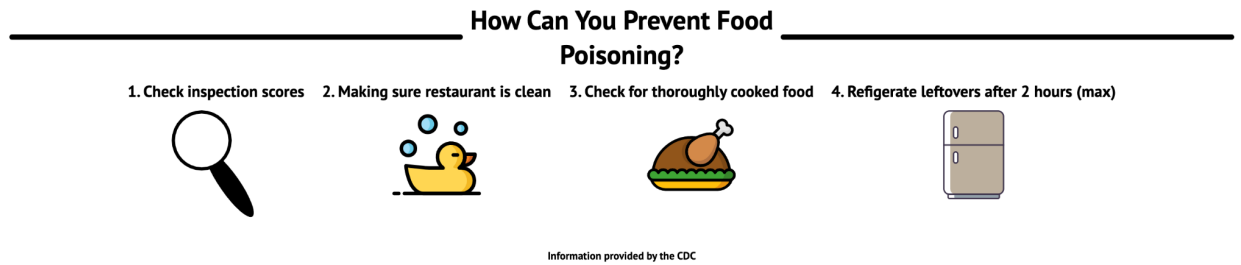


Figure 23: Food Poisoning infographic #3

Learn Page

The learn page consists of multiple components. It has four dropdown components and one WordCloud component. In the Learn.js file, the useEffect() function is used to update the information provided to the WordCloud component when the selection from one of the dropdown components changes. This file also fetched information from the remote Node.js server hosted on DigitalOcean that makes API calls to the database. It is also in this file that the words fetched from the database are filtered, counted, and scaled. In order to scale the words, the maximum count is preserved and then each count is divided by the maximum value. If a word's count divided by the maximum value is less than 0.0007 then the word is not included in the word cloud to prevent word crowding.

About Page

The About Page provides the user with a place to better understand the ins and outs of project development, as well as team/sponsor information. The flow of the page inspires the user to explore it as they please, by providing buttons that transit the user to the desired section.

3.6.6 Deployment

As with every web application, there needs to be a method of deployment so that the application is visible on a URL and can be viewed by any user on the internet. To start the development of our project before we had any allocated server space or URL we thought that utilizing GitHub and GitHub pages would be our best option.

GitHub Pages

GitHub Pages allows a Git repository to have a main (master) branch that is published with a live URL. GitHub also supports HTTPS, custom page URLs, 404 pages, as well as various other visibility settings. Having a shared, personalized repository allows us to develop the website on our local machines and have a live branch that we can utilize for development purposes at all times. It also provides access to all of the source code for when we publish the website and move its contents to the server. Privacy was also a concern that was looked into, as some of the data we may be storing and utilizing may need to be kept private. Since our Git repository is currently set to private, only the developers will have access to the source code of the website, and we will need to ensure that we are using the proper privacy and visibility settings on our published page. GitHub pages also worked flawlessly with React, and by adding a few scripts to our package.json file we could run npm scripts straight from our development console that would deploy our current branch onto the 'github-pages' branch and be visible on our live URL.

Apache Web Server

Of course, GitHub Pages would not serve as a permanent home to host this application. Once we were able to set up our WPI server and move our project code to this machine, it was

time for a new method of deployment. We were given the url usda-foodpoisoning.wpi.edu, and using Apache Web Server we were able to host a build of our application on this link. Within our project repository on the server, the user can run ‘npm build’ to compile and build our app, which will be housed within a build folder in the repository. This is where Apache Web Server pulls from, and the live link will be updated.

4. Findings

4.1 User Study

4.1.1 Use for User Study

Our user study was designed with the intent of collecting user feedback on our website. This feedback will help our team understand how to better design tools in our website for ease of access and usability. Information provided in both short answer responses, as well as multiple-choice questions, will help our team find the elements of our project that need additional work before its final release.

4.1.2 Breakdown of Study Process

The users will be expected to navigate our website in the order provided on our survey form. First, the user will be expected to navigate through the “Home” page of our site. They are first asked if they used the Timeline tool at the top of the page, and any suggestions they may have towards this tool.



MAY 17, 2017 – OCTOBER 4, 2017

2017 IMPORTED MARADOL PAPAYA SALMONELLA OUTBREAK

In 2017, the CDC, public health and FDA investigated a widespread outbreak of Salmonella infections related to imported Maradol papayas. The agencies were able to track the papayas back to the Carica de Campeche farm in Mexico. The outbreak was one of four related to these papayas. It resulted in 220 infections with 68 hospitalizations and one death.

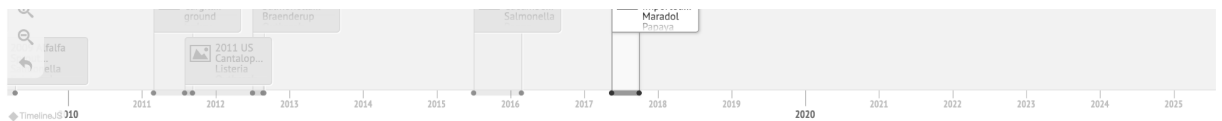

 2015-2016 CUCUMBER SALMONELLA POONNA OUTBREAK

Food Navigator



Figure 24: Timeline Tool

Next, the user is asked to proceed through the Home page and is asked about our infographics.



So far we have analyzed ...



100

Food Poisoning Tweets



50

Current Food Poisoning Outbreaks

Tracker



Explore past outbreaks and interact with the data

EXPLORE

Common Food Poisoning Words



Explore our world cloud

EXPLORE

About Us



Learn more about the project and future work

LEARN

Figure 25: Tweet and Outbreak counters.

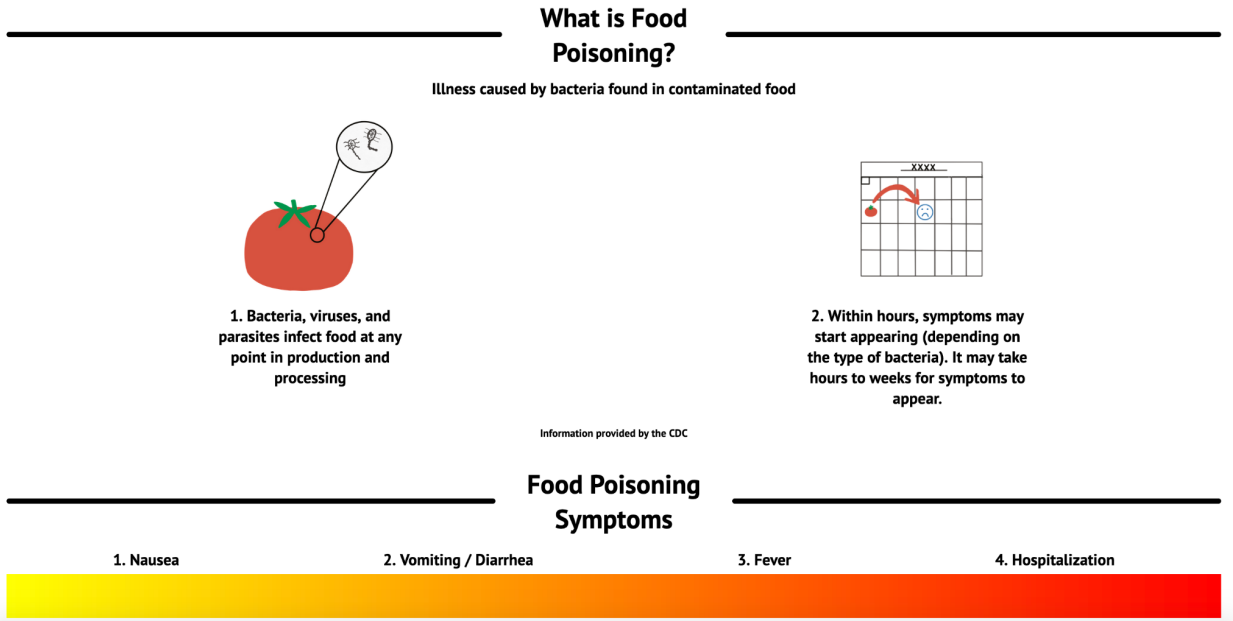


Figure 26: Infographics on food poisoning facts.

The user is then guided to the next page, the “About” page. Here they are asked if they tried using the tabs to navigate this page and if they learned additional info that they were looking for in regards to the website.

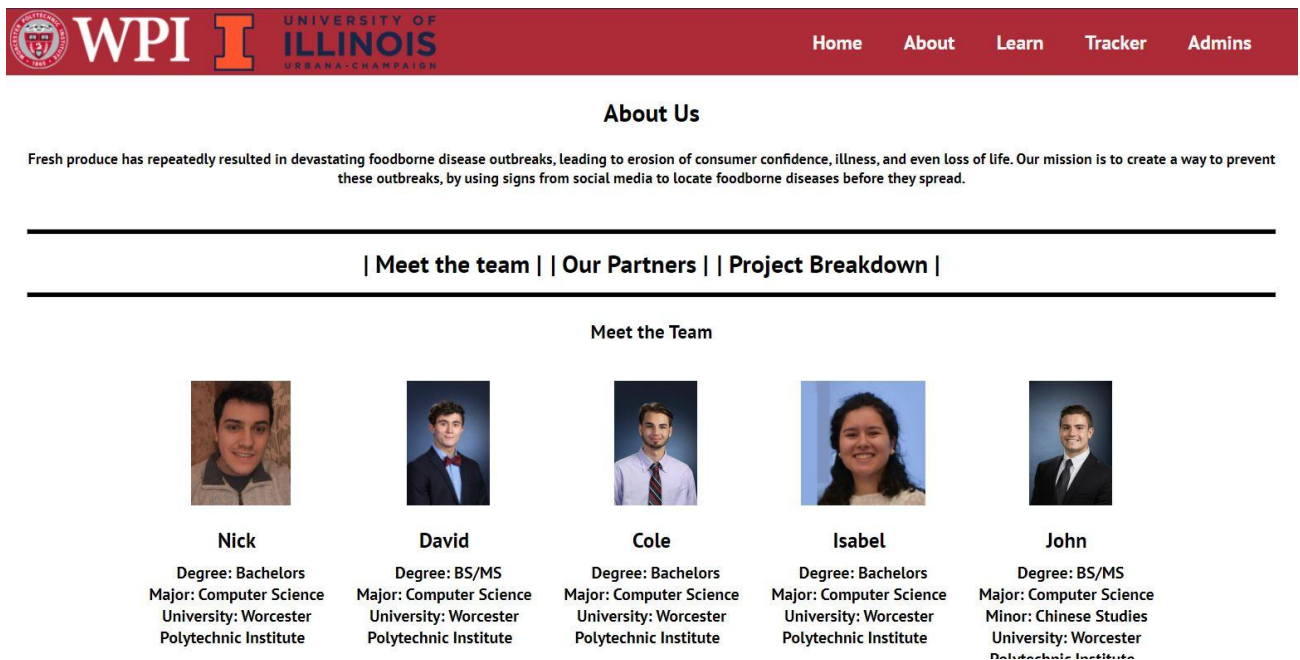


Figure 27: Meet the team section from our website.

Our Partners



The University of Illinois team provided essential research towards the early stages of machine learning development. Their team explored the process of using the deep learning model to analyze data. They provided the framework for the machine learning algorithms that pulled tweet data. The collaboration with UIUC allowed our team to have a head start in this USDA sponsored project.



The USDA is the sponsor of this MQP. They tasked our team with creating the startup of what would later become a food poisoning prediction system.

Figure 28: Partner Descriptions

Then, the user is guided to the next page, the “Learn” page. Here they are asked to interact with our Word Cloud tool. The user is expected to provide information about the usability of the tool, visualizations they liked/disliked, etc...

The screenshot shows a navigation bar at the top with the WPI and University of Illinois logos on the left and links for Home, About, Learn, Tracker, and Admins on the right. Below the navigation bar is a section titled "Explore past outbreaks" with a subtitle "Select the year and type of words to explore the most common words used in tweets relating to food borne illness". There are four red buttons with dropdown arrows: "INGREDIENTS", "JANUARY", "2022", and "NEW YORK". Below these buttons is a word cloud of food items in red text, including beef, artichoke, anchovy, pork, apple, asparagus, berry, chicken, bacon, anchovies, aspicate, avocado, and strawberries.

WPI UNIVERSITY OF ILLINOIS URBANA - CHAMPAIGN

Home About Learn Tracker Admins

Explore past outbreaks

Select the year and type of words to explore the most common words used in tweets relating to food borne illness

INGREDIENTS JANUARY 2022 NEW YORK

*beef
artichoke
anchovy
pork
apple
asparagus berry
chicken bacon
anchovies
aspicate avocado
strawberries*

Figure 29: Example of the Word Cloud tool found on the Learn page.

Lastly, and most importantly is the final page, the “Tracker” page. Here the user is expected to interact with our heat map tool. They are asked questions regarding the

visualizations of the data, usability of the tool, and to report any confusion they may have when using the tool.

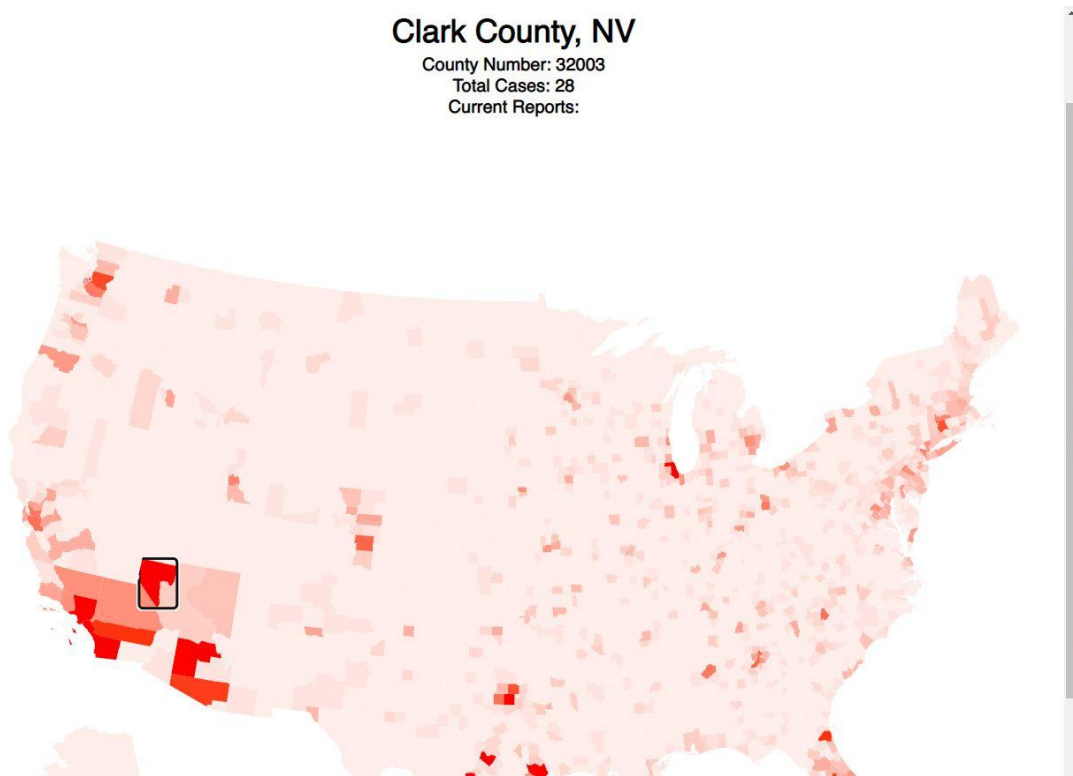


Figure 30: Example of the Map Tool found on the Tracker page.

In general, this user study is a straightforward, website walkthrough, that will help our team gain valuable insight towards improving our project.

4.1.3 Results of the Study

The study ended its collection of data after only five users were able to fill it out. While the study consisted of a minimal quantity of users, the information gathered greatly helped in the advancements of the project. The Consent Form as well as individual user responses can be found in Appendices C and D for reference.

Going in order through the form, we start with questions regarding the Home Page. The first set of responses were aimed towards the Timeline. The users were prompted if there was anything that would make the tool better, and in general, the majority of users were happy with the Timeline. Only minor remarks on small visuals were made, and our team decided it was good enough to keep our focus on other elements. Next were comments about the infographics on the home page, seen below.

Did the infographics help teach you about food born illness?
5 responses



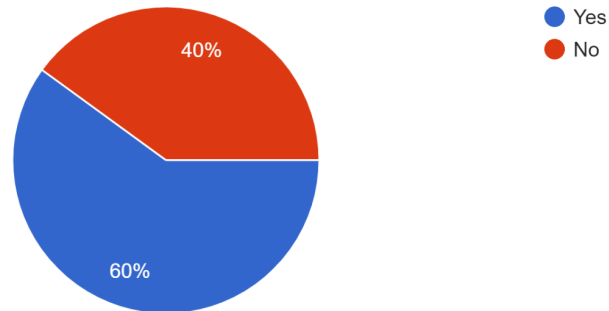
Graph 1: Infographics results.

In summary, it seems the infographics did not provide substantial information to the user. Many seemed to feel the information was common knowledge, with some even commenting further that the information provided was incorrect. These graphics were not changed before the final submission, but should be a focus point in future stages.

Next, the form shifts to questions regarding the About Page. The first question here refers to the navigation tabs specific to this page and whether they used them to navigate the page.

Did you use the tabs "Meet the team" "Our Partners" and "Project breakdown" to navigate the About Page?

5 responses

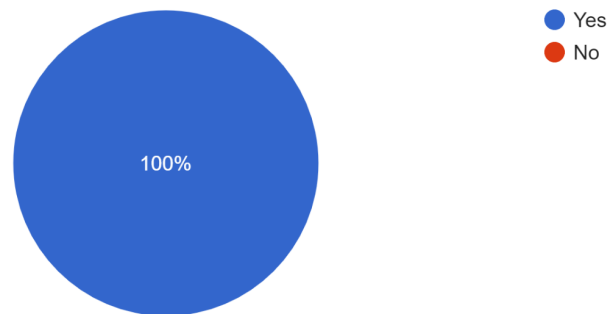


Graph 2: Tab results.

It seems the majority did use the tabs to navigate the page. One comment stating “Make the tab buttons turn a certain color when hovering over them. It’ll make it feel like they do something.”, was a good comment. We proceeded by fulfilling this easy change, and found it added a sense of use to the buttons. They now highlight red when hovering over. Lastly, comments regarding the page as a whole revealed a complaint about the description of the project. It was originally a copy and paste of the original project description, and one of our local tests suggested we write our own, so that was also changed as a result of the study.

Next is the Learn Page. From the data we can see that all users think the Word Cloud is a fun, interactive tool to explore and play around with.

Does the Learn Page provide an engaging experience for learning?
5 responses



Graph 3: Learn results.

Suggestions regarding the Word Cloud were given however. This form was filled out by many of the users before the tool was fully implemented. As a result, many of the suggestions were about adding more to the tabs, and data as a whole. Luckily, as the tool approached its final stage, we can say we have fulfilled those requests. The tool properly functions with more data, and tab items.

Lastly, we asked about the Tracker Page, and more specifically about our Map Tool. It seemed evident from the data that the users were pleased with this tool, proven by the results of the first question.

Does the Tracker Page provide an easy way to identify areas of potential food borne illness outbreaks?

5 responses

YEs the map is very interactive

yes

Yes!

Yes.

Yes! Very cool!

In further questions, we received great feedback on how the visibility and usability of the tool could be improved from the following comments.

Could anything be improved to the design and/or usability of the map?

5 responses

Just that a legend with what the color codes mean

add number of cases when one moves to a specific area

I feel it's a little odd that clicking on an area displays the results at the top, so perhaps a window with the results would work better?

No

Move the results of each area somewhere more readable.

These were wonderful results for this tool. In the weeks following these responses, our back end team worked on a new Beta version of the map tool. This new tool, which can be implemented in future stages, can be found in the future work section below. It covers all of the comments above, and more, thereby fulfilling the study requirements.

Overall, our team was pleased to see that the overall goal and functionality of our pages and tools were met. The suggestions given provided us with an important reminder that external feedback is crucial towards the perfection and finalization of these projects.

5. Future Work

5.1 Map / Tracker Tool

One of the limitations that we faced during this project was the expandability and utility that the tracker map was able to provide. Although it had plenty of features for displaying data,

there was little utility to customize the map itself and improve the user experience. The majority of these limitations came from the JavaScript library that was originally used for the map, react-simple-maps. For this reason, towards the end of our development process we decided to draft a new map tool using MapBoxGL, a far more flexible mapping library that runs on OpenGL and provides fluid, feature rich mapping tools. Although this portion of the project was not developed to a point where it could replace the original map, the beta code for this component is included in the repository under the directory `‘/src/components/map-beta’`.

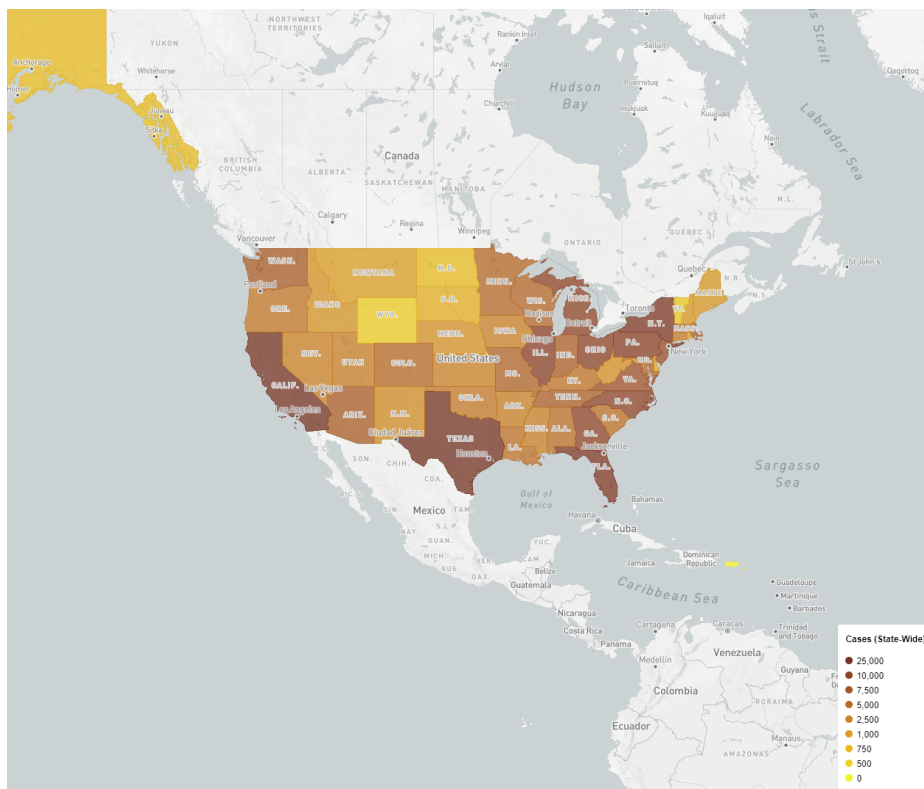


Figure 31: Example of the Map Tool we think could be easily implemented in the future.

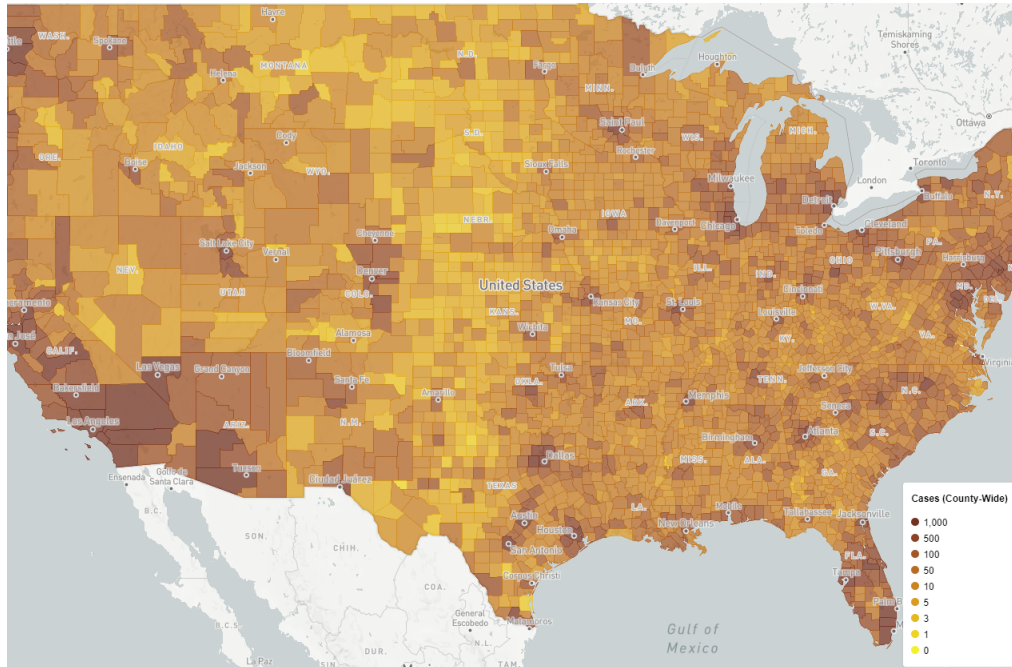


Figure 32: Another example of the Map Tool we think could be easily implemented in the future.

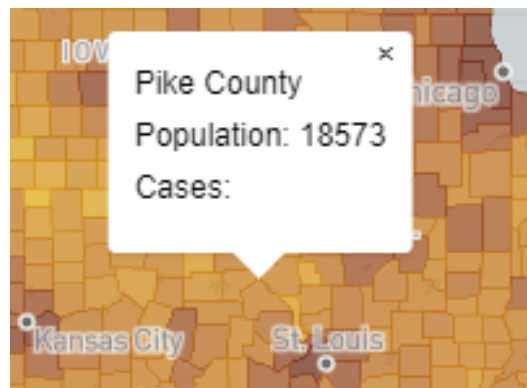


Figure 33: Example of the future Map Tool display feature.

6. Appendices

Appendix A: Figures

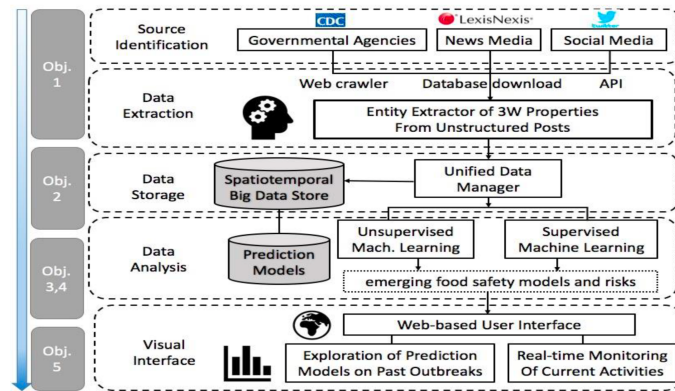


Figure 1: Depicts the initial project scope and objectives defined by the previous groups working on the project.

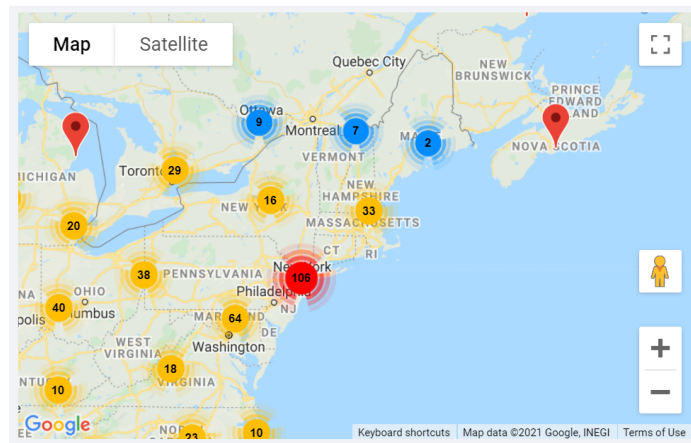


Figure 2: This is a map from iWasPoisoned.com and serves to display reports aggregated into hotspot zones across the United States.

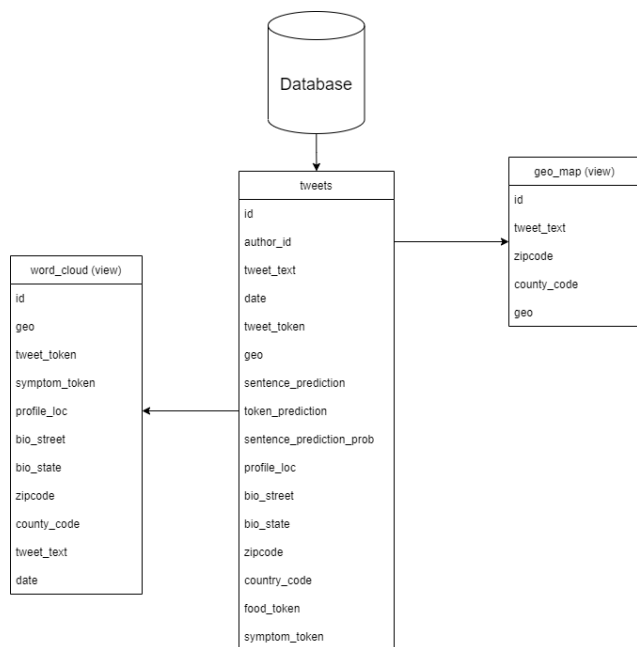


Figure 3: This diagram depicts the database schema design.

```

TIMEFRAME_TO_MULTIPLIER = {"hour": 3600,
                             "hours": 3600,
                             "day": 86400,
                             "days": 86400}

# Extract the age of the report (x hours ago, y minutes ago, etc), grab current time, determine time report
was posted

currTime = datetime.now().timestamp()

age = report.find_element_by_xpath("div[@class='float-left report-title']/h3[@class='h6 card-subtitle
mb-2 text-muted']").text.split()

```

```
delta, timeframe = age[0], age[1]

deltaSeconds = int(delta) * TIMEFRAME_TO_MULTIPLIER[timeframe]

timestamp = datetime.fromtimestamp(currTime - deltaSeconds)
```

Figure 4: Crawler Iteration Code

```
type = crawler.find_element_by_xpath("//img[contains(@title, 'Banner')]").get_attribute("title")

if type == FOOD_SAFETY_BANNER_STRING:

    container = crawler.find_element_by_xpath("//main[@aria-label='Main Content Area']")

    title =

container.find_element_by_xpath("div[@class='row']/div/div[@class='syndicate']/h1").get_attribute("innerHTML")

    date =

crawler.find_element_by_xpath("/html/body/div[6]/main/div[3]/div/div[3]/div[1]/div/div[2]/div/p").get_attribute(
"innerHTML")
```

```
reportSummaries.append([title, date])
```

Figure 5: Differentiation Code

```
# 1. Author ID  
author_id = tweet['author_id']  
  
# 2. Time created  
created_at = dateutil.parser.parse(tweet['created_at'])  
  
# 3. Geolocation  
if ('geo' in tweet):  
    geo = tweet['geo']['place_id']  
else:  
    geo = " "
```

4. Tweet ID

```
tweet_id = tweet['id']
```

5. Language

```
lang = tweet['lang']
```

6. Tweet metrics

```
retweet_count = tweet['public_metrics']['retweet_count']
```

```
reply_count = tweet['public_metrics']['reply_count']
```

```
like_count = tweet['public_metrics']['like_count']
```

```
quote_count = tweet['public_metrics']['quote_count']
```

7. source

```
source = tweet['source']
```

8. Tweet text

```
text = tweet['text']
```

Figure 6: Tweet parts Code

```
keyword = '(egg OR hard boiled OR #egg OR #hardboiled) lang:en -is:retweet' #has:geo
```

```
start_list = ['2020-01-01T17:00:43.000Z'] #ISO8
```

```
# '2021-03-21T00:00:00.000Z']
```

```

end_list = ['2020-06-01T23:00:00.000Z']
#         '2021-03-31T00:00:00.000Z']
max_results = 500

```

Figure 7: Keyword Code

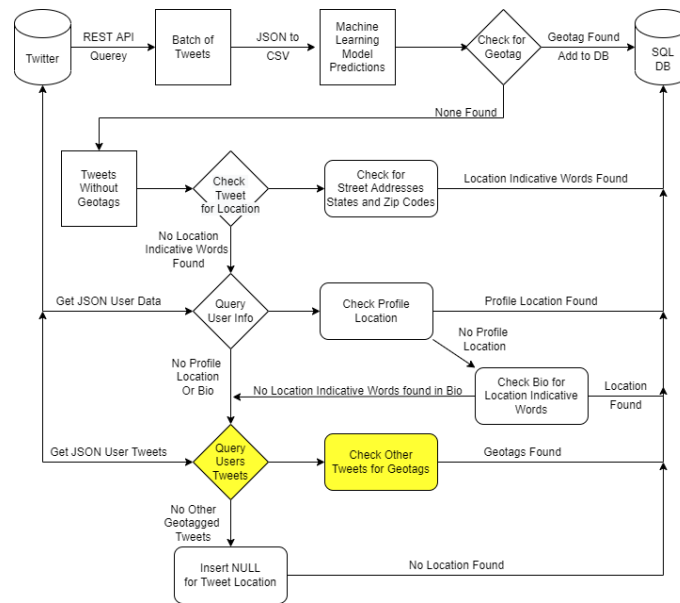


Figure 8: This is the data pipeline that shows how data collected from sources like Twitter is processed and geo-tagged before being added to the database. The sections in yellow were removed due to rate limit restrictions on the API, however, would still be beneficial in the future.



Figure 9 - Timeline example obtained from our live MQP Site.



Figure 10: Timeline slide in information example obtained from our live MQP Site.

Tracker

Explore past outbreaks and interact with the data

EXPLORE

Common Food Positioning Words

Explore our world cloud

EXPLORE

About Us

Learn more about the project and future work

LEARN

Figure 11: Buttons that lead to the different pages on our home page.

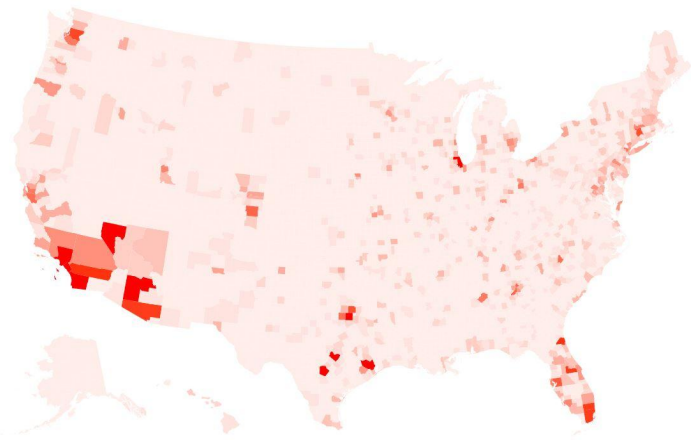


Figure 12: Heat map of user cases found by our system based on tweet information.

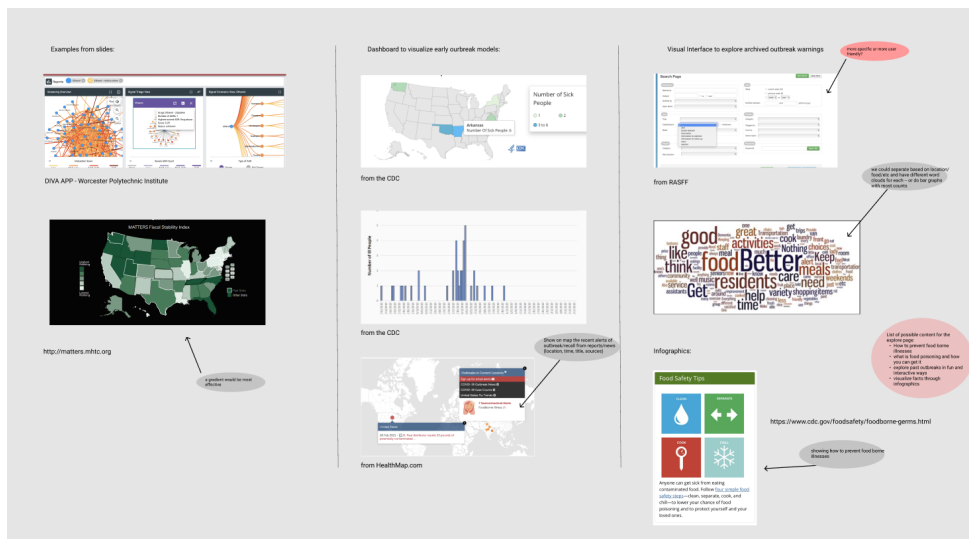


Figure 13: Drawing board for website visualizations.

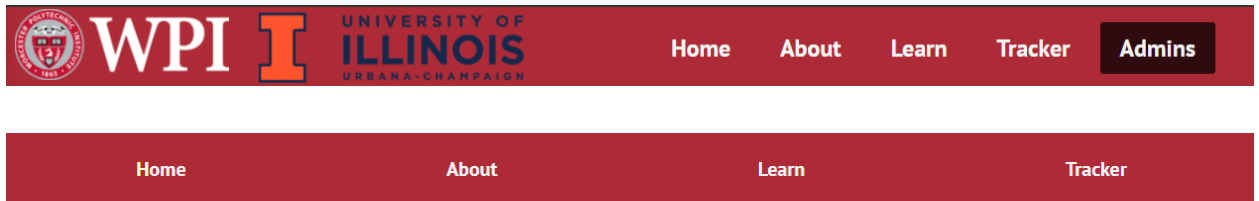


Figure 14: The navigation bar (above) and bottom banner (below) from the website.



Figure 15: Three widgets located on the home page.

So far we have analyzed ...

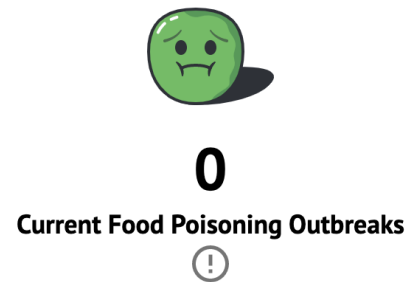


Figure 16: The two infographics located on the home page featuring the Not Available component.

*potato oatmeal
 grilled and almond meat the
 dairy milk chicken takis
 black of food iced chocolate
 banana hot taco spicy with had
 boba fried ice coffee raw chips
 fruit cheese
 fast a pizza burger
 greasy vanilla brown
 cinnamon
 eggnog*

Figure 17: A sample of the word cloud.

INGREDIENTS JANUARY 2022 NEW YORK

INGREDIENTS
SYMPTOMS
COMPANY

*potato oatmeal
 grilled and almond meat the
 dairy milk chicken takis
 black of food iced chocolate
 banana hot taco spicy with had
 boba fried ice coffee raw chips
 fruit cheese
 fast a pizza burger
 greasy vanilla brown
 cinnamon
 eggnog*

Figure 18: The dropdown buttons, displaying what it looks like when you click on them.

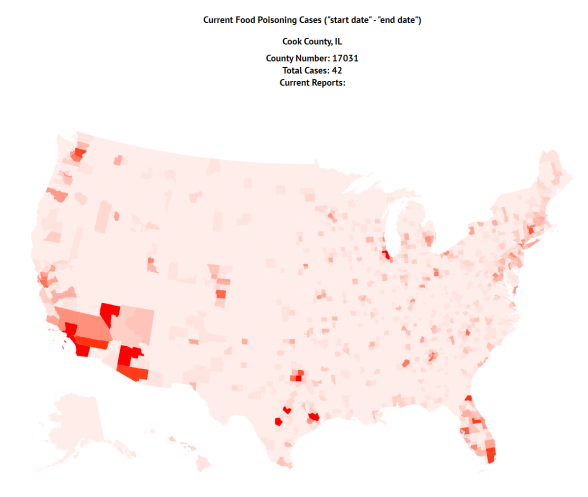
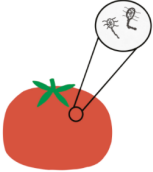


Figure 19: The map from the tracker page on our website.

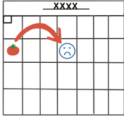
Figure 20: The admin page from our website.

What is Food Poisoning?

Illness caused by bacteria found in contaminated food



1. Bacteria, viruses, and parasites infect food at any point in production and processing



2. Within hours, symptoms may start appearing (depending on the type of bacteria). It may take hours to weeks for symptoms to appear.

Information provided by the CDC

Figure 21: Food Poisoning infographic #1

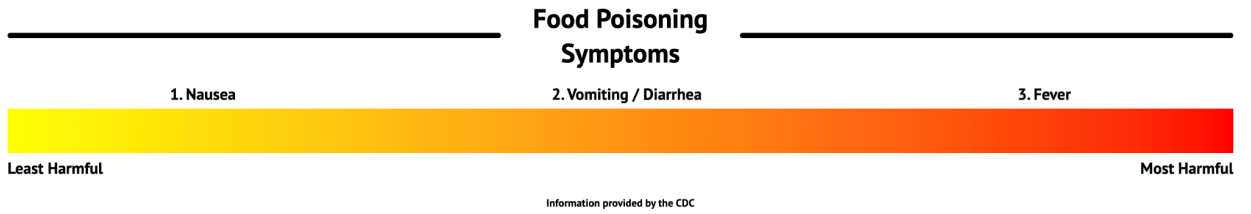


Figure 22: Food Poisoning infographic #2

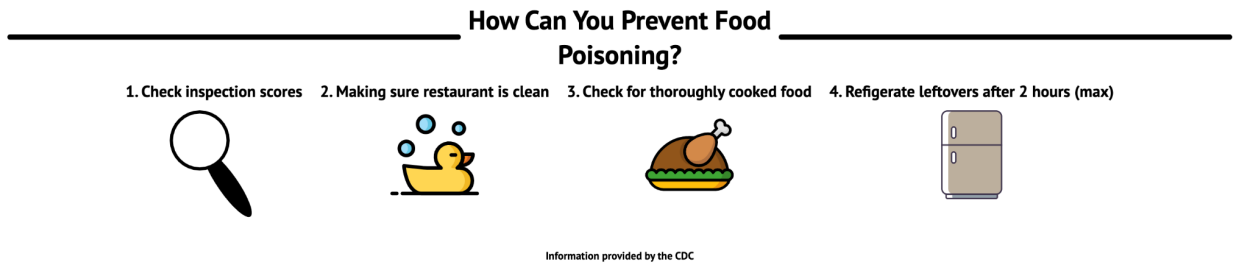
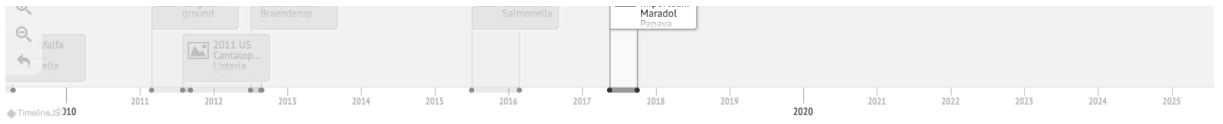


Figure 23: Food Poisoning infographic #3



Figure 24: Timeline Tool



So far we have analyzed ...



100

Food Poisoning Tweets



50

Current Food Poisoning Outbreaks

Tracker

Explore past outbreaks and interact with the data

EXPLORE

Common Food Poisoning Words

salmonella
anchovy
anchovy chicken
apple berry
pork
honey
anchovy
spaghetti sauce
spicy beef

Explore our world cloud

EXPLORE

About Us

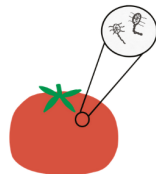
Learn more about the project and future work

LEARN

Figure 25: Tweet and Outbreak counters.

What is Food Poisoning?

Illness caused by bacteria found in contaminated food



1. Bacteria, viruses, and parasites infect food at any point in production and processing



2. Within hours, symptoms may start appearing (depending on the type of bacteria). It may take hours to weeks for symptoms to appear.

Information provided by the CDC

Food Poisoning Symptoms

1. Nausea

2. Vomiting / Diarrhea

3. Fever

4. Hospitalization



Figure 26: Infographics on food poisoning facts.

About Us

Fresh produce has repeatedly resulted in devastating foodborne disease outbreaks, leading to erosion of consumer confidence, illness, and even loss of life. Our mission is to create a way to prevent these outbreaks, by using signs from social media to locate foodborne diseases before they spread.

| [Meet the team](#) | | [Our Partners](#) | | [Project Breakdown](#) |

Meet the Team



Nick

Degree: Bachelors
 Major: Computer Science
 University: Worcester
 Polytechnic Institute



David

Degree: BS/MS
 Major: Computer Science
 University: Worcester
 Polytechnic Institute



Cole

Degree: Bachelors
 Major: Computer Science
 University: Worcester
 Polytechnic Institute



Isabel

Degree: Bachelors
 Major: Computer Science
 University: Worcester
 Polytechnic Institute



John

Degree: BS/MS
 Major: Computer Science
 Minor: Chinese Studies
 University: Worcester
 Polytechnic Institute

Figure 27: Meet the team section from our website.

Our Partners



The University of Illinois team provided essential research towards the early stages of machine learning development. Their team explored the process of using the deep learning model to analyze data. They provided the framework for the machine learning algorithms that pulled tweet data. The collaboration with UIUC allowed our team to have a head start in this USDA sponsored project.



The USDA is the sponsor of this MQP. They tasked our team with creating the startup of what would later become a food poisoning prediction system.

Figure 28: Partner Descriptions

Explore past outbreaks

Select the year and type of words to explore the most common words used in tweets relating to food borne illnesses



Figure 29: Example of the Word Cloud tool found on the Learn page.

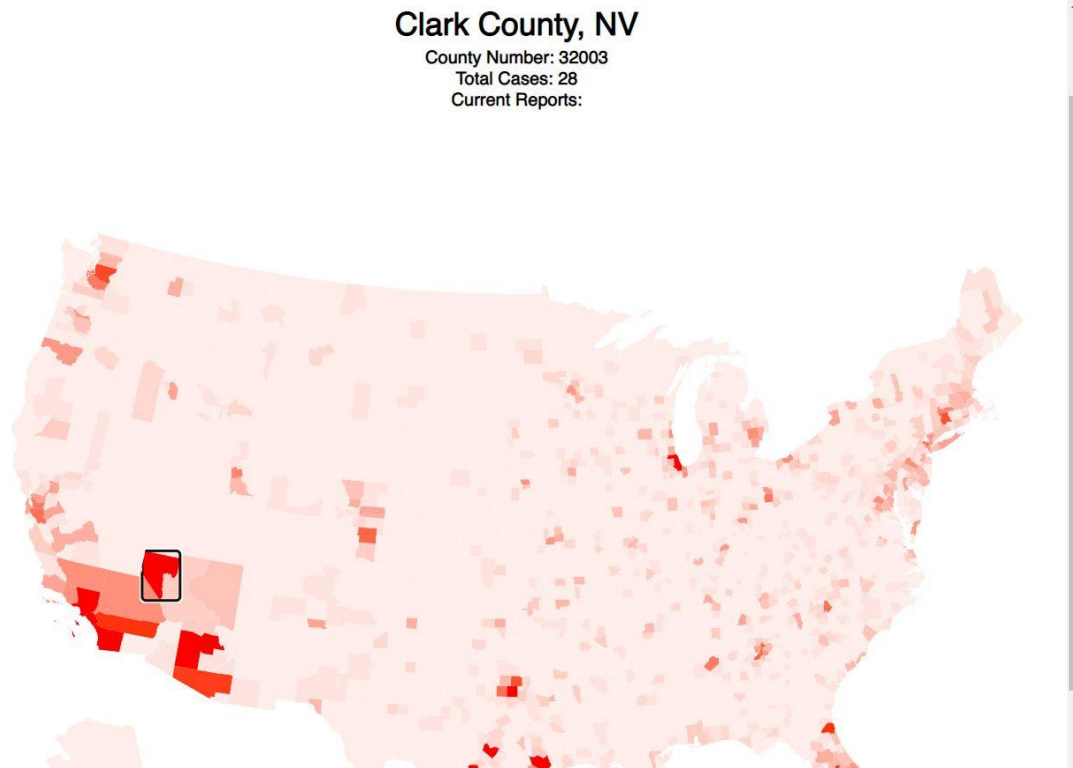


Figure 30: Example of the Map Tool found on the Tracker page.

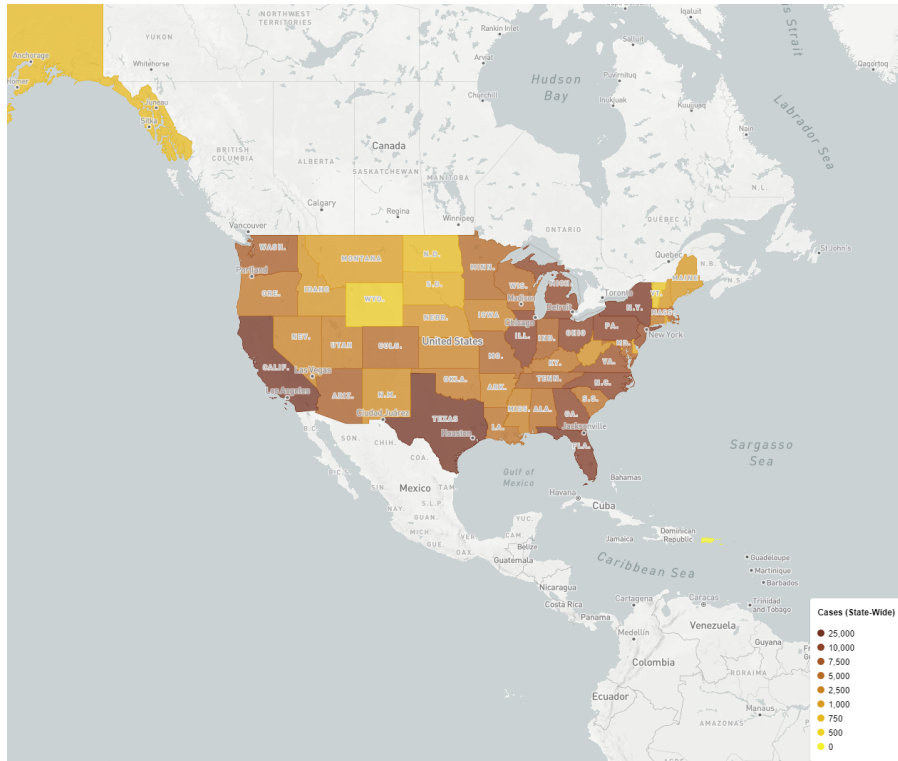


Figure 31: Example of the Map Tool we think could be easily implemented in the future.

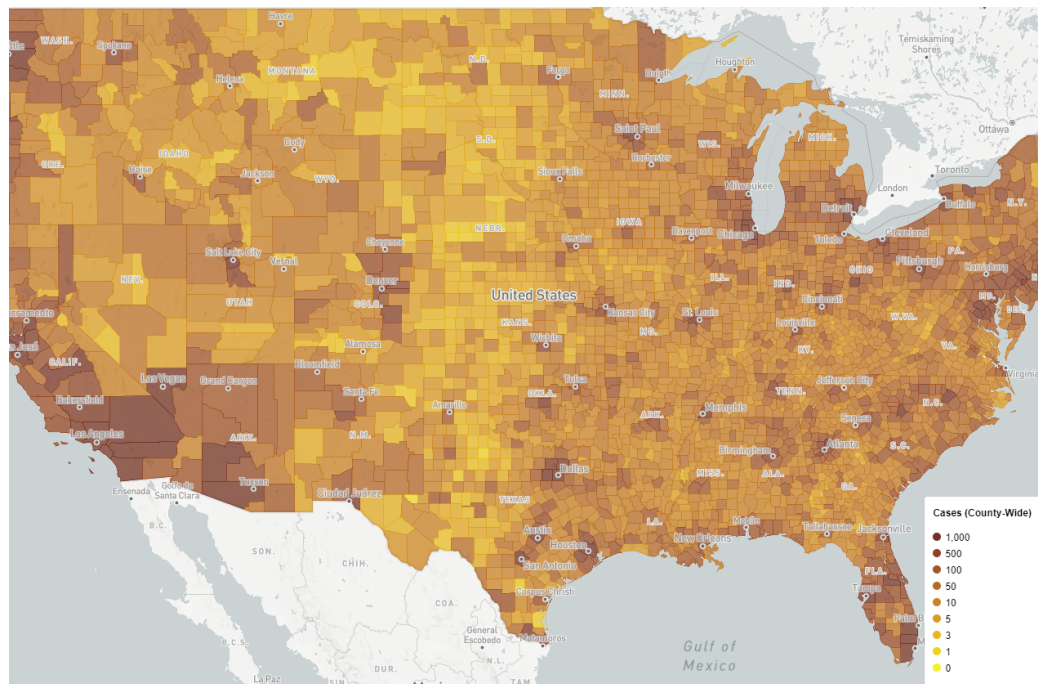


Figure 32: Another example of the Map Tool we think could be easily implemented in the future.

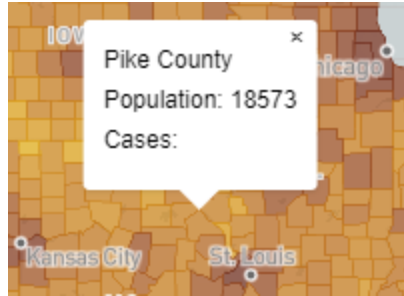


Figure 33: Example of the future Map Tool display feature.

Appendix B: IRB Approval - IRB-22-0410

WORCESTER POLYTECHNIC INSTITUTE

100 INSTITUTE ROAD, WORCESTER MA 01609 USA

Institutional Review Board

FWA #00030698 - HHS #00007374

Notification of IRB Approval

Date: 25-Feb-2022

PI: Elke Rundensteiner A

Protocol Number: IRB-22-0410

Protocol Title: Food Poisoning Alerts MQP

Approved Study Personnel: Vachon, Nicholas R~Rundensteiner, Elke A~Leandres,
David G~Alvarado Blanco Uribe, Isabel C~Carroll, John
C~Noreika, Cole G~

Effective Date: 25-Feb-2022

Exemption Category: 3

Sponsor*:

The WPI Institutional Review Board (IRB) has reviewed the materials submitted with regard to the above-mentioned protocol. We have determined that this research is exempt from further IRB review under 45 CFR § 46.104 (d). For a detailed description of the categories of exempt research, please refer to the [IRB website](#).

The study is approved indefinitely unless terminated sooner (in writing) by yourself or the WPI IRB. Amendments or changes to the research that might alter this specific approval must be submitted to the WPI IRB for review and may require a full IRB application in order for the research to continue. You are also required to report any adverse events with regard to your study subjects or their data.

Changes to the research which might affect its exempt status must be submitted to the WPI IRB for review and approval before such changes are put into practice. A full IRB application may be required in order for the research to continue.

Please contact the IRB at irb@wpi.edu if you have any questions.

Appendix C: User Study Survey

Consent Form!

In order for us to use the information you provide in this survey, you must consent to the following. The website contains information about food poisoning illness throughout the country, which some viewers may find disturbing. Any surveys who answer no, will be discarded, and we will not view the information.

* Required

Purpose of the study

This study is intended to obtain information from everyday users to help improve the usability of our website. By having real site testers, our team can gain a better understanding of how people with no information about the website will interact with the elements.

Procedures to be followed

The user will be expected to spend 10-20 minutes exploring our website, and providing feedback in the indicated fields. As the user moves along, they should note anything that confuses them, and include it in their feedback. By following the provided survey, the user will have gone through every element of the website by the time the complete the form.

Risks to study participants

This study does not impose any risk on the user.

Benefits to research participants and others

We hope the user is able to gain additional understanding of foodborne illness throughout our country.

Record keeping and confidentiality

Records of your participation in this study will be held confidential so far as permitted by law. However, the study investigators, and under certain circumstances, the Worcester Polytechnic Institute Institutional Review Board (WPI IRB) will be able to inspect and have access to confidential data that identify you by name. Any publication or presentation of the data will not identify you.

Compensation or treatment in the event of injury

There are no risks of physical injury in this study, and as a result, no compensation will be provided in the result of injury related to this study. You do not give up any of your legal rights by signing this statement.

**For more information about this research or about the rights of research participants,
or in case of research-related injury, contact me or the IRB**

Contact information of the Student Investigator: Nicholas Vachon, Tel. 6034977617, Email: nrvachon@wpi.edu

Contact information of the IRB Manager: Ruth McKeogh, Tel. 508 831-6699, Email: irb@wpi.edu

Contact information the Human Protection Administrator (Gabriel Johnson, Tel. 508-831-4989, Email: gjohnson@wpi.edu).

Your participation in this research is voluntary

Your refusal to participate will not result in any penalty to you or any loss of benefits to which you may otherwise be entitled. You may decide to stop participating in the research at any time without penalty or loss of other benefits. The project investigators retain the right to cancel or postpone the experimental procedures at any time they see fit.

By signing below,

you acknowledge that you have been informed about and consent to be a participant in the study described above. Make sure that your questions are answered to your satisfaction before signing. You are entitled to retain a copy of this consent agreement.

1. Please print your full name below *

2. Please select todays date *

_____ *Example: January 7, 2019*

3. Please approve your consent *

Mark only one oval.

Yes

No

Food
Poisoning
Alerts
MQP

Hello and thank you for taking the time to test our site! Please use the following link to access the Food Poisoning Alerts MQP website.

<https://usda-foodpoisoning.wpi.edu/>

Then, observe each section of the site in the order provided below, and leave any thoughts, questions, or suggestions about each section in the provided text field. Short, to the point remarks are okay!

Home Page

The Home Page was designed with the intent to catch a user's interest in the subject matter, and encourage them to explore the rest of the website.

4. Did you explore the Timeline at the top of the page?

Mark only one oval.

Yes

No

5. Is there anything about the Timeline that was confusing, or could be improved?

6. Did the infographics help teach you about food born illness?

Mark only one oval.

Yes

No

Other: _____

Food
Poisoning
Alerts
MQP

Hello and thank you for taking the time to test our site! Please use the following link to access the Food Poisoning Alerts MQP website.

<https://usda-foodpoisoning.wpi.edu/>

Then, observe each section of the site in the order provided below, and leave any thoughts, questions, or suggestions about each section in the provided text field. Short, to the point remarks are okay!

About Page

The About Page was designed to inform the user about individual, specific elements of the project. Here, you should be able to see who worked on the project, why the project exists, and how our team went about designing it.

7. Does the About section inform you about specific elements that were not provided on the Home Page?

8. Did you use the tabs "Meet the team" "Our Partners" and "Project breakdown" to navigate the About Page?

Mark only one oval.

- Yes
- No
- Other: _____

9. Do you have any suggestions to make the About Page better?



Hello and thank you for taking the time to test our site! Please use the following link to access the Food Poisoning Alerts MQP website.

<https://usda-foodpoisoning.wpi.edu/>

Then, observe each section of the site in the order provided below, and leave any thoughts, questions, or suggestions about each section in the provided text field. Short, to the point remarks are okay!

Learn Page

The Learn Page was included to offer a fun, interactive way for the user to learn more about food poisoning outbreaks.

10. Does the Learn Page provide an engaging experience for learning?

Mark only one oval.

Yes

No

11. Was the word cloud an easy tool to learn and interact with? Is it clear what information you are obtaining through each tab?

12. Do the different colors within the word cloud provide any value? Can you tell what different colors mean?

13. What did you take away from using the word cloud?

14. Do you have any suggestions to make the Learn Page better?

Food
Poisoning
Alerts
MQP

Hello and thank you for taking the time to test our site! Please use the following link to access the Food Poisoning Alerts MQP website.

<https://usda-foodpoisoning.wpi.edu/>

Then, observe each section of the site in the order provided below, and leave any thoughts, questions, or suggestions about each section in the provided text field. Short, to the point remarks are okay!

Tracker Page

The Tracker Page is where our main food poisoning tracking tool can be found. Here the user should be able to explore outbreaks throughout the country, that our system is currently detecting.

15. Does the Tracker Page provide an easy way to identify areas of potential food borne illness outbreaks?

16. Could anything be improved to the design and/or usability of the map?

17. Do you understand the information being displayed when clicking on a county?

Mark only one oval.

- Yes
- No
- Other: _____

18. Using the tool, can you find the county with the most foodborne illness cases? (List the county below)

19. Is there anything else you would expect to see on the Tracker Page, that would help inform you of outbreaks throughout the country?

Please leave any final suggestions you may have!

20. Please leave any final thoughts, comments, or suggestions about the website here!

Appendix D: User Study Relevant Responses

User 1:

Is there anything about the Timeline that was confusing, or could be improved?

Some of the icons on the timeline could be shown in more density dense format .

Was the word cloud an easy tool to learn and interact with? Is it clear what information you are obtaining through each tab?

YEs just the color could be a bit brighter

Do the different colors within the word cloud provide any value? Can you tell what different colors mean?

YEs , the words highlighted show the major ones

What did you take away from using the word cloud?

Great number of associated terminologies

Does the Tracker Page provide an easy way to identify areas of potential food borne illness outbreaks?

YEs the map is very interactive

Could anything be improved to the design and/or usability of the map?

Just that a legend with what the color codes mean

Using the tool, can you find the county with the most foodborne illness cases? (List the county below)

Los Angeles County, CA

Is there anything else you would expect to see on the Tracker Page, that would help inform you of outbreaks throughout the country?

The legend defining what color stands for around what (just like the geographical maps have)

Please leave any final suggestions you may have!

Please leave any final thoughts, comments, or suggestions about the website here!

The website is great just needs more color (just not red)

User 2:

Is there anything about the Timeline that was confusing, or could be improved?

remove the photo icon

Did the infographics help teach you about food born illness?

Yes

No

Other:

foodborne illness is not only caused by bacteria. There are many other source of contamination such as virus, fungus, and chemicals. Use the CDC's official definition would be better.

Does the About section inform you about specific elements that were not provided on the Home Page?

The project description needs to be rephrased so that layman can understand easily, instead of copy and paste from the narrative by saying obj1, obj2.. I also suggest remove the figure (it's not supposed to be released to the public..)

Did you use the tabs "Meet the team" "Our Partners" and "Project breakdown" to navigate the About Page?

Yes

No

Other: _____

Do you have any suggestions to make the About Page better?

make the team description consistent

Was the word cloud an easy tool to learn and interact with? Is it clear what information you are obtaining through each tab?

yes, the display of words in the word cloud could be improved

Do the different colors within the word cloud provide any value? Can you tell what different colors mean?

no, the size of the word tells the significance itself

What did you take away from using the word cloud?

most common ones

Does the Tracker Page provide an easy way to identify areas of potential food borne illness outbreaks?

yes

Could anything be improved to the design and/or usability of the map?

add number of cases when one moves to a specific area

Using the tool, can you find the county with the most foodborne illness cases? (List the county below)

not easily

Is there anything else you would expect to see on the Tracker Page, that would help inform you of outbreaks throughout the country?

the source of information updating on side of the map

Please leave any final suggestions you may have!

Please leave any final thoughts, comments, or suggestions about the website here!

the design could be improved such as the font, size, and layout of words and figures

User 3:

Is there anything about the Timeline that was confusing, or could be improved?

The timeline was a well made, and informative tool.

Do you have any suggestions to make the About Page better?

Make the tab buttons turn a certain color when hovering over them. It'll make it feel like they do something.

Was the word cloud an easy tool to learn and interact with? Is it clear what information you are obtaining through each tab?

There isn't a whole lot to it it seems. Not very much information to choose from.

Do the different colors within the word cloud provide any value? Can you tell what different colors mean?

There were only two colors, red and black. Their significance is unknown.

What did you take away from using the word cloud?

Specific relations of food born illness to symptoms, foods and locations.

Could anything be improved to the design and/or usability of the map?

I feel it's a little odd that clicking on an area displays the results at the top, so perhaps a window with the results would work better?

Using the tool, can you find the county with the most foodborne illness cases? (List the county below)

Yes! It looked to be Los Angeles County, CA. However all of the high counting counties are the same shade of red, so it is not immediately obvious.

Is there anything else you would expect to see on the Tracker Page, that would help inform you of outbreaks throughout the country?

Maybe more details about that area's risk other than just case count?

Please leave any final suggestions you may have!

Please leave any final thoughts, comments, or suggestions about the website here!

Overall a good site.

User 4:

Do the different colors within the word cloud provide any value? Can you tell what different colors mean?

Sort of? They don't seem to mean much

What did you take away from using the word cloud?

How important each item is towards food poisoning.

Is there anything else you would expect to see on the Tracker Page, that would help inform you of outbreaks throughout the country?

More info per area.

User 5:

Is there anything about the Timeline that was confusing, or could be improved?

No, it seems well made and easy to use.

Does the About section inform you about specific elements that were not provided on the Home Page?

Yes! Much more about the project itself.

Was the word cloud an easy tool to learn and interact with? Is it clear what information you are obtaining through each tab?

It is a little scarce right now. Not a lot of data.

Do the different colors within the word cloud provide any value? Can you tell what different colors mean?

I like the colors, but I'm not sure how they represent differently.

What did you take away from using the word cloud?

The foods, items, or locations that are the most relevant to food poisoning outbreaks.

Do you have any suggestions to make the Learn Page better?

Add more! There is almost no flexibility with the one word cloud thing, and there is nothing on the rest of the page!

Does the Tracker Page provide an easy way to identify areas of potential food borne illness outbreaks?

Yes! Very cool!

Could anything be improved to the design and/or usability of the map?

Move the results of each area somewhere more readable.

Using the tool, can you find the county with the most foodborne illness cases? (List the county below)

Many of the most infected areas are the same color, but after clicking through a bit, I found it. Los Angeles County, CA

Is there anything else you would expect to see on the Tracker Page, that would help inform you of outbreaks throughout the country?

I think the map is a cool tool! I'm not exactly sure what else I would expect from something like this.

Please leave any final suggestions you may have!

Please leave any final thoughts, comments, or suggestions about the website here!

Nice Job!

Appendix E: Repository

For access to the repository, please download it from the following Drive link.

<https://drive.google.com/file/d/1ikBZ6Nxc5orTUEx-8vnjdf8duZdS222e/view?usp=sharin>

g

The results of the code can be accessed at the following domain.

<https://usda-foodpoisoning.wpi.edu/>

Appendix F: Graphs

Did the infographics help teach you about food born illness?

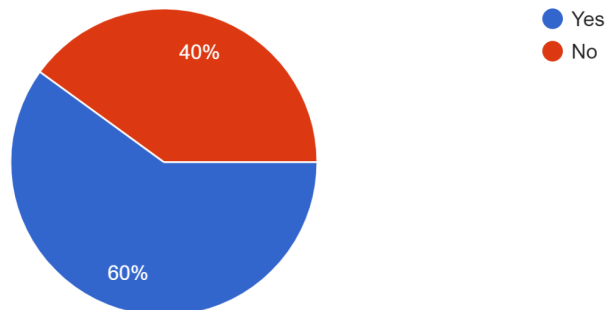
5 responses



Graph 1: Infographics results.

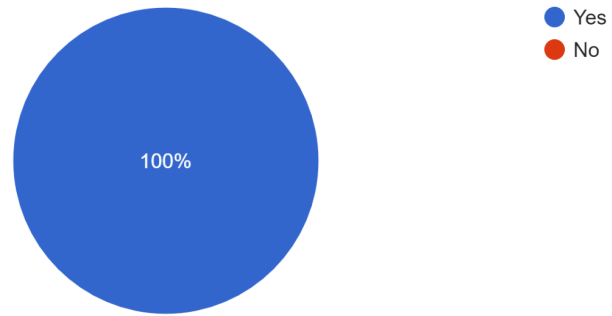
Did you use the tabs "Meet the team" "Our Partners" and "Project breakdown" to navigate the About Page?

5 responses



Graph 2: Tab results.

Does the Learn Page provide an engaging experience for learning?
5 responses



Graph 3: Learn results.

References

Rahimi, A., Vu, D., Cohn, T., & Baldwin, T. (2015). Exploiting text and network context for geolocation of social media users. *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.
<https://doi.org/10.3115/v1/n15-1153>

Bo, Han, et al. "Geolocation Prediction in Social Media Data by Finding Location Indicative Words." *Proceedings of COLING 2012: Technical Papers*, 2012, pp. 1045–1062.,
<https://aclanthology.org/C12-1064.pdf>.

Moreland, K. (2016). Why we use bad color maps and what you can do about it. *Electronic Imaging*, 2016(16), 1-6.

Sibrel, S. C., Rathore, R., Lessard, L., & Schloss, K. B. (2020). The relation between color and spatial structure for interpreting colormap data visualizations. *Journal of vision*, 20(12), 7-7.

Smiciklas, M. (2014). Infographics. *Communication and influence through images*. Saint Petersburg, Piter Publ.

Dharmendra Kumar (06 Oct 2021). **Bootstrap 4 |Introduction**.
<https://www.geeksforgeeks.org/bootstrap-4-introduction/>