Final Report - Public Health App

Summary

The purpose of our Public Health website is to connect users with healthcare resources based on their needs and location. The user's needs are determined by their health determinants– factors affecting one's health. These include access to housing and transportation, food and clean water availability, health literacy, and mental healthcare access. Thus, our customers, Dr. Rob Carpenter and Jonathan Thomas of the Texas A&M School of Medicine, asked us to provide a way for them to define new surveys, let users fill them, assess users' health determinants based on their responses, and provide a list of recommendations based on their needs and location.

Users are first required to sign up by providing basic details such as zip code, name, age, and email address. The current service is restricted to users above the age of eighteen. Users are presented with a customized survey created by our clients that asks them a series of questions to determine which health determinants they require. We then create a list of general local resources based on our database and provide links to relevant search results from Google Maps and 211texas.org website, based on the user's needs and location. It's important to note that the part we generate includes local resources near the user which may not be listed in general search engines and should be gathered and added to the database. For example, if a user suffers from low food security, the website would direct them to a list of food banks and food distribution warehouses in their area generated by Google Maps and 211texas.org resources, and provide them with a general list of resources derived from the app's database. So far, we only created a list for resources near the Bryan/College Station area, but our customers may create additional lists for other areas. Our stakeholders are our customers (Dr. Rob Carpenter and Jonathan Thomas) and the users of the website. Our intended users include patients, doctors, and others lacking health-related needs.

User Stories

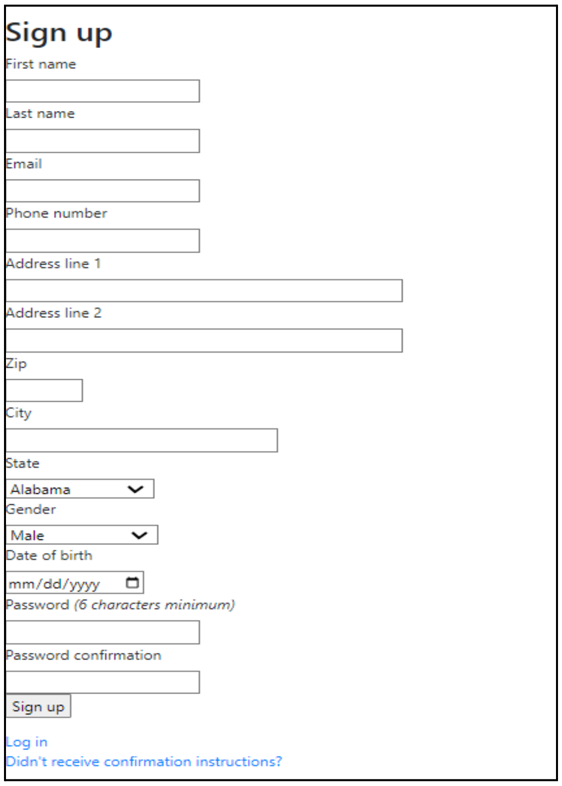
1. **App signup:** As a patient, I want to be able to create an account, so that the application can gather the required information (name, email, address, etc.).

- Story Points: 17 - This is a fundamental feature of any application, and it has a medium complexity involving a user interface for form input, data validation, and database interaction.
  (2 points) Sign up - Defining the user model
  (2 points) Sign up - Develop the Front end interface for signup.
  (2 points) Sign up - User's email is confirmed upon signup.

(3 points) Sign up - Design cucumber tests for sign-up fields
(2 points) Sign up - Adding 50 states to the "states" field
(2 points) Sign up - Adding in the Gender Identity Form Question
(4 points) Sign up - Validate user input for all signup fields.

- Implementation Status: Complete - The team developed a signup page that collects the user's name, email, address, etc, and stores the data securely in our database.

- Changes: A gender identity question, login link, and email verification link were added later on on the signup page.



Sign up

First name

Last name

Email

Phone number

Address line 1

Address line 2

Zip

City

State
Alabama

Gender
Male

Date of birth
mm/dd/yyyy

Password *(6 characters minimum)*

Password confirmation

Sign up

Log in
Didn't receive confirmation instructions?

App Signup

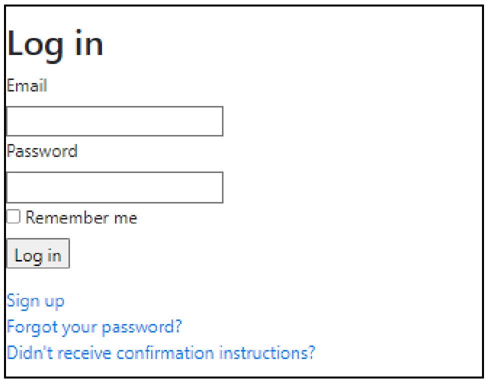
2. **App login**: As a patient, I want to log into the application so I can access the health surveys, gain access to resources and share the results of the surveys with my doctor.

- Story Points: 12 points - It's less complex than the signup process because it involves fewer fields and fewer data validation. Following is the breakdown of story points:
(2 points) Log in - Rspec test case for invalid login credentials
(2 points) Log in - Workflow/Design Diagram for login feature
(3 points) Log in - Rspec test for correct login credentials
(2 points) Log in - Cucumber test case correct & incorrect login credentials
(2 points) Log in - Forgot password functionality

(1 points) Direct authenticated users to the home page (with surveys, results), and non authenticated users to the login page.

- Implementation Status: Complete - The team created a login page where users can enter their email and password to access the application.

- Changes: None - This is a basic feature that didn't require much change during the project other than remembering registered users and the email verification link.



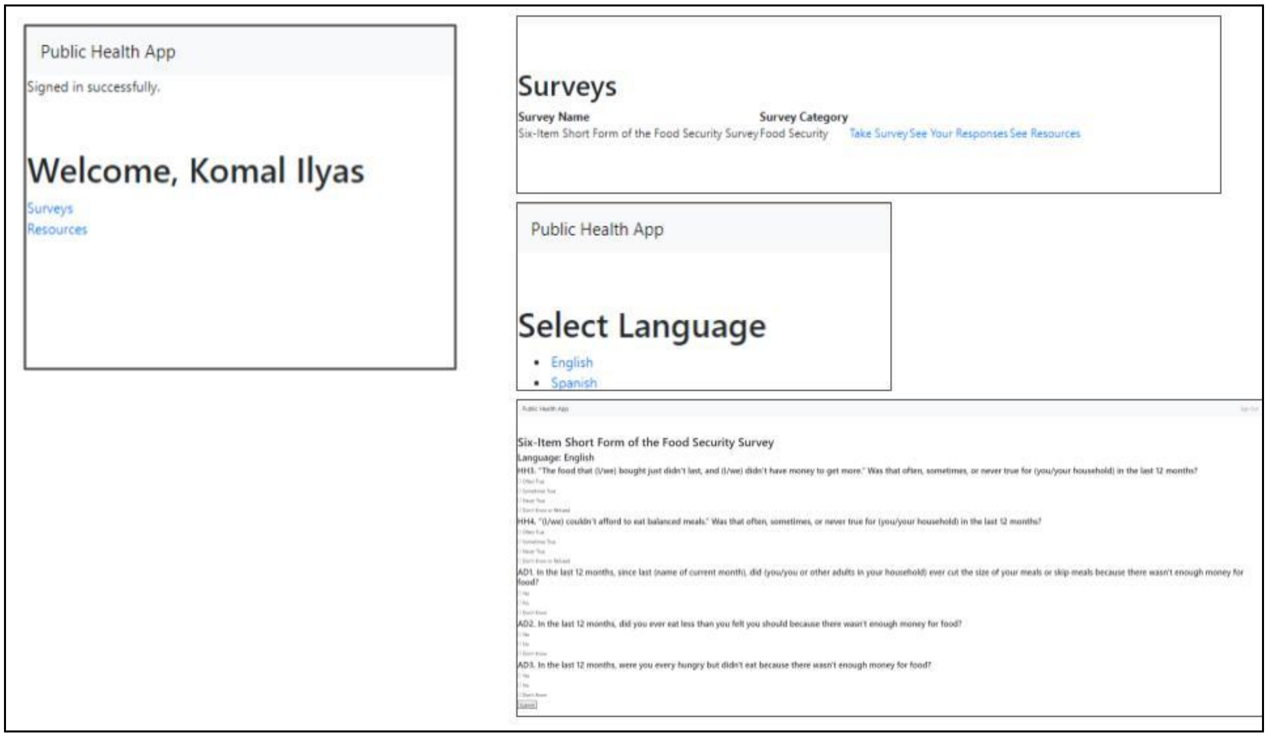
App Login

3. **Survey**: As a patient I want to complete the necessary health surveys so that I can be directed to the resources I may need.

- Story Points: 19 points - This involves a fair amount of complexity, as it requires developing the survey itself, creating an interface for it, storing responses, and handling any logic tied to the responses.
  (3 points) Survey - Created a database and seed correctly to store survey_name, survey category, and languages.
  (4 points) Survey - Create survey controller to route users
  (2 points) Survey - Design Diagram for the survey
  (3 points) Shifting development DB to PostgreSQL
  (1 point) Seeding the questions database
  (2 points) Querying questions with given survey and language
  (2 points) Question - Resolve the Failed Test Cases/Add appropriate Test Cases for Question page as required
  (2 points) Questions - Ability for conditional questions to appear/disappear depending on a user's previous answer

- Implementation Status: Complete - The team developed a survey form that adjusts according to the user's responses, stores responses to questions in the survey, and redirects to the results of the survey

- Changes: The original story was modified to allow for dynamic adjustments of the survey based on user input


Public Health App

Signed in successfully.

Welcome, Komal Ilyas

Surveys

Resources

Surveys

Survey Name	Survey Category	
Six-Item Short Form of the Food Security Survey	Food Security	Take Survey See Your Responses See Resources

Public Health App

Select Language

- English
- Spanish

Public Health App

Six-Item Short Form of the Food Security Survey

Language: English

HH3. "The food that (I/we) bought just didn't last, and (I/we) didn't have money to get more." Was that often, sometimes, or never true for (you/your household) in the last 12 months?

- Often True
- Sometimes True
- Never True
- Don't Know or Refused

HH4. "(I/we) couldn't afford to eat balanced meals." Was that often, sometimes, or never true for (you/your household) in the last 12 months?

- Often True
- Sometimes True
- Never True
- Don't Know or Refused

AD1. In the last 12 months, since last (name of current month), did (you/you or other adults in your household) ever cut the size of your meals or skip meals because there wasn't enough money for food?

- Yes
- No
- Don't Know

AD2. In the last 12 months, did you ever eat less than you felt you should because there wasn't enough money for food?

- Yes
- No
- Don't Know

AD3. In the last 12 months, were you every hungry but didn't eat because there wasn't enough money for food?

- Yes
- No
- Don't Know

Submit

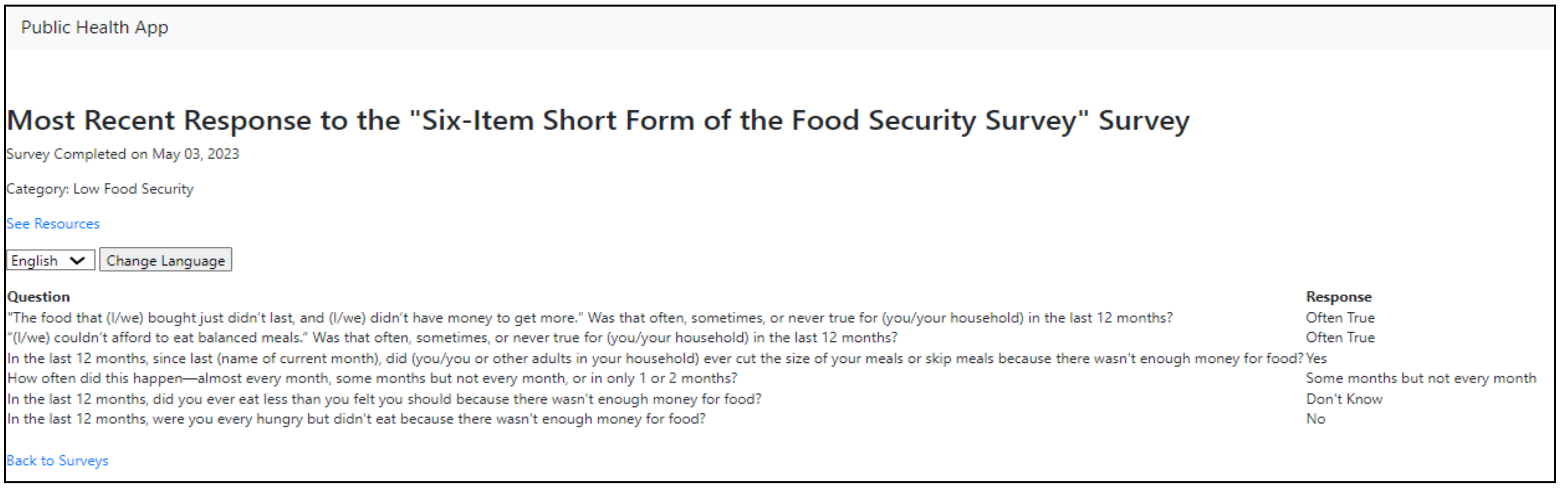
App Survey Screenshots

4. **Access survey results:** As a patient I want to be able to access my submitted survey results so that I can track my progress and determine what is affecting my health.

   Story Points: 17 points
   (3 points) Creating a results table and connecting survey results with responses table
   (2 points) Creating the Survey Category Table, and connecting with Survey Results, displaying in Responses
   (4 points) Responses table - creating and querying (creating the responses table, seeding responses, querying responses with survey, language, and user, and implementing a view to show the queried results)
   (2 points) Survey Results Category Validation
   (2 points) Responses score tests
   (4 points) Survey responses validation tests and redirection to response
   (2 points) Response Scores Test

   Implementation Status: Complete - The team developed a feature that allows users to view their survey results and identify the category based on survey results.

Changes: Survey results show the results based on the latest responses submitted by users.



Public Health App

Most Recent Response to the "Six-Item Short Form of the Food Security Survey" Survey

Survey Completed on May 03, 2023

Category: Low Food Security

See Resources

English | Change Language

Question	Response
"The food that (I/we) bought just didn't last, and (I/we) didn't have money to get more." Was that often, sometimes, or never true for (you/your household) in the last 12 months?	Often True
"(I/we) couldn't afford to eat balanced meals." Was that often, sometimes, or never true for (you/your household) in the last 12 months?	Often True
In the last 12 months, since last (name of current month), did (you/you or other adults in your household) ever cut the size of your meals or skip meals because there wasn't enough money for food?	Yes
How often did this happen—almost every month, some months but not every month, or in only 1 or 2 months?	Some months but not every month
In the last 12 months, did you ever eat less than you felt you should because there wasn't enough money for food?	Don't Know
In the last 12 months, were you every hungry but didn't eat because there wasn't enough money for food?	No

Back to Surveys

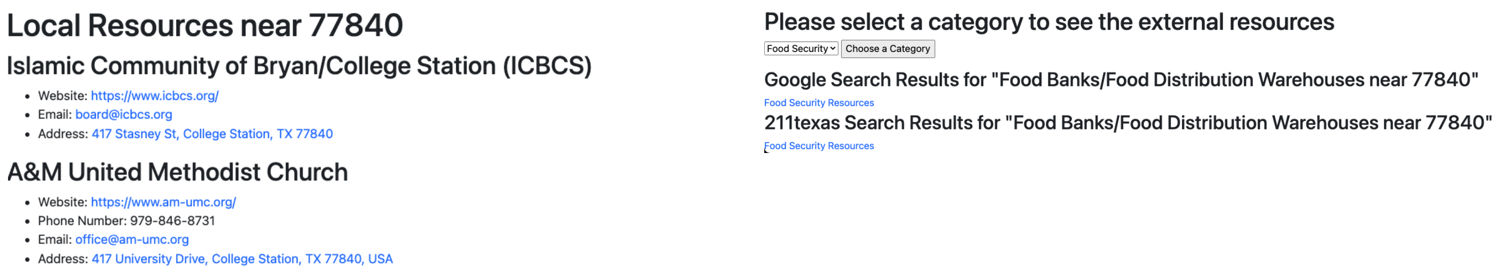
App Survey Results

5. **Access health resources**: As a patient, I want to be directed to health resources after completing the survey so that I can improve my health.

   Story Points: 6 points

- This involves creating a list of resources based on category, linking them to survey results, and presenting them in a user-friendly way.
  (1 points) Getting Resources from Texas 211
  (1 points) Getting Resources from Google and displaying in View
  (1 points) Creating a Table for local resources and gathering local BCS Resources
  (3 points) Creating a View to show all resources

- Implementation Status: Complete - The team developed a feature that presents users with a list of resources based on their survey results.
- Changes: The story was expanded to include personalized resource recommendations and an embedded link to Google search based on zip code



Local Resources near 77840

Islamic Community of Bryan/College Station (ICBCS)

- Website: https://www.icbcs.org/
- Email: board@icbcs.org
- Address: 417 Stasney St, College Station, TX 77840

A&M United Methodist Church

- Website: https://www.am-umc.org/
- Phone Number: 979-846-8731
- Email: office@am-umc.org
- Address: 417 University Drive, College Station, TX 77840, USA

Please select a category to see the external resources

Food Security | Choose a Category

Google Search Results for "Food Banks/Food Distribution Warehouses near 77840"

Food Security Resources

211texas Search Results for "Food Banks/Food Distribution Warehouses near 77840"

Food Security Resources

Health Resources

Legacy Project Discussion

We had a team that previously had worked on the same project but their implementation used Node.js for backend development and React.js for frontend development. For the database, the application used MongoDB, while Jest and Postman were used for testing. After looking at the previous team's implementation of the Public Health App, unfortunately, we discovered quite a few issues:

- The previous team utilized a different tech stack, rather than Ruby on Rails, and Cucumber which have been covered in this class.
- They failed to provide sufficient tests for the application, thus tests must be written for different functionalities using jest/postman, and the data in the database has to be repopulated for testing.
- The surveys in their app did not have a language option set up and thus a result of this was that the same surveys in different languages were considered as different surveys, this was a major logical flaw that we solved.
- The structure of their application was such that users had to manually check which surveys he/she had already taken without keeping track of previous survey results. This was a major feature that was required by the client.

After discussing with our client, Dr. Carpenter and Jonathan, and Prof. Ritchey about these issues, we decided it will be best to rebuild the application using Ruby on Rails, to incorporate improved functionalities and more rigorous testing. In addition, changing the tech stack to Ruby on Rails will ensure future development can continue on a familiar tech stack covered in this course.

Improvements to Existing Functionalities

1. We added a feature to get user information like zip code, state, etc other information from users during signup which will help recommend the resources.
2. Also users' emails will be verified on signup as well. The current application allows users to change passwords without verifying the email, thus this is a major security issue for the application. In addition, sensitive information is not encrypted in the database.
3. The Survey page currently presents the users with multiple surveys, which are duplicates of each other in other languages. We will add a language option before each type of survey.
4. The completed surveys are not being highlighted differently as completed (not obvious to a user which survey has been completed, which remains) Thus we can add functionality where the completed surveys will be highlighted as completed and will automatically show in the Survey results section. There will also be an option to edit the survey results.

New functionalities

1. The recommender system for the results has not been implemented. We added this functionality by using the location of the user (provided at signup), the results from the

survey, and recommending nearby resources using Google and 211texas.org, and also other local resources.
2. The ability to keep track of users' previous survey results is also being done. All user responses are stored in the Results table in our application.
3. Functionality to update user details like Address, Email, and Phone Number was added.

Team Roles

Iteration	Product Owner	Scrum Master
0	Sidharth Baveja	Chinmay Ajit Sawkar
1	Komal Ilyas	Fisher Coburn
2	Dheep Manish Dalamal	Pouyan Forghani
3	Chinmay Ajit Sawkar	Sidharth Baveja
4	Fisher Coburn	Komal Ilyas
5	Pouyan Forghani	Dheep Manish Dalamal

Each iteration we rotated who was the Product Owner and Scrum Master such that everyone gets experience in each role.

Iteration Summaries

Iteration 0

- Discussed the app structure and features that were required by the client and had lo-fi UI mockups validated by the client. We discussed the final deliverable and understood the purpose of the app and its different features.
- Created initial repository for the project in Github required for working collaboratively and set up the technical architecture Heroku website for production for deploying the website.

Iteration 1

- **Sign up -** Developed a backend user model and frontend signup page interface with input validation for the input fields.
- **Login -** Developed a frontend login page that validated passwords with the stored version and also created a Forget Password interface.
- **Survey -** Created surveys database and seeded correctly to store survey_name, survey category, and languages

Iteration 2

- **Sign up -** Email verification for new users was set up

- **Login -** Functionality to restore account using forgot password option was developed, Redirected the authenticated users to homepage, while users who are not logged in were redirected automatically to signup page
- **Questions** - Shifting development DB to PostgreSQL was required for the development database as there were differences in array formats for PostgreSQL which was being used in production. This was important as we could not know the errors before deploying to production. Created a database for storing questions and seeded it appropriately.

Iteration 3

- **Sign up -** Validation of all Signup Fields in Both Front and Backend and added Gender Identity Field to the SignUp Form
- **Survey** - Seeding the questions database, Querying questions with the given survey and language from the seeded questions database, Creating a view for user to display the queried questions and answer the question of the survey. Added the functionality where specific questions are hidden/shown based on a previous question's response.
- Responses table - Creating the responses table

Iteration 4

- **Sign up -** Added the functionality to choose a state from a dropdown list
- **Responses -** Created a table to store responses and seeding it appropriately. Added functionality to querying responses for the user for selected given survey & language. Added backend logic for calculating the score and category of the user based on the responses. Implemented a 'See Your Responses" view to show the queried results. Added functionality to validate the user responses (check if all questions are answered including conditionally viewed questions) and provide an appropriate confirmation message

Iteration 5

- **Survey and Responses -** Added additional test cases and resolved previous errors in tests for survey responses validation, redirection to responses, generating responses score. Added Survey Results Category Validation.
- **Resources Tab -** Creating a View to show all resources Creating Table for local resources and gathering local BCS Resources. Getting Resources from Google and displaying in View. Getting Resources from texas 211.

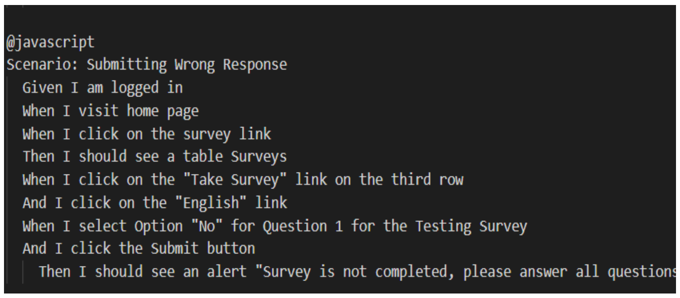
Customer Meetings

Iteration 0 Meeting: The first meeting took place on February 6th, 2023 at 8:30 AM with Jonathan on Zoom. During this meeting, we discussed the previous team's implementation of the project and what the goals for the project are in this semester. The major goal was to implement the resources section of the application to direct patients to resources they may need after they complete a survey.

Iteration 1 Meeting: This meeting took place on February 20th, 2023 with both Jonathan and Dr. Carpenter where we demoed the deployed application with signup and login functionality.

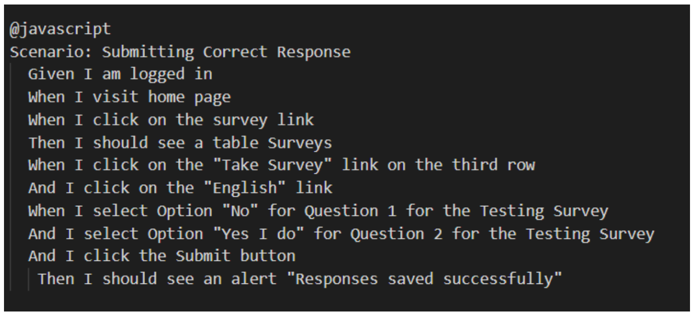
Iteration 2 Meeting: Both Dr. Carpenter and Jonathan were unable to make our bi-weekly meeting (scheduled for March 6th, 2023), so for this iteration, we recorded a video demo of email verification upon user signup and a basic survey page.
Iteration 3 Meeting: We met with Dr. Carpenter on Mar 20, 2023, over Zoom. Since the previous week was spring break the meeting was brief. We showed the progress of the app and discussed the plan for this iteration. The client asked to add an extra feature of conditional questions (some questions are to be displayed only if previous questions in the survey are answered positively). This was implemented in this iteration as well.
Iteration 4 Meeting: We met with Dr. Carpenter and Jonathan on April 3rd, 2023 over Zoom where we showed how we added some surveys to the deployed application. In addition, we added the feature to hide/show survey questions based on a previous question's response.
Iteration 5 Meeting: We met with Dr. Carpenter and Jonathan on April 17th, 2023 over Zoom where we demonstrated the survey submission functionality. Also, we showed the functionality where a user can view their most recent response to a survey, and we also compute what "category" a survey response belongs to (High Food Security, low food security).
Final Meeting: We met with Dr. Carpenter and Jonathan on May 1st, 2023 over Zoom. In the final Zoom meeting, we demoed the resources page of the application which shows local resources, resources found on Google, and 211texas.org based on the user's zip code.

BDD/TDD



```
@javascript
Scenario: Submitting Wrong Response
  Given I am logged in
  When I visit home page
  When I click on the survey link
  Then I should see a table Surveys
  When I click on the "Take Survey" link on the third row
  And I click on the "English" link
  When I select Option "No" for Question 1 for the Testing Survey
  And I click the Submit button
    Then I should see an alert "Survey is not completed, please answer all questions
```

Sad path



```
@javascript
Scenario: Submitting Correct Response
  Given I am logged in
  When I visit home page
  When I click on the survey link
  Then I should see a table Surveys
  When I click on the "Take Survey" link on the third row
  And I click on the "English" link
  When I select Option "No" for Question 1 for the Testing Survey
  And I select Option "Yes I do" for Question 2 for the Testing Survey
  And I click the Submit button
  Then I should see an alert "Responses saved successfully"
```

Happy path

We defined 31 cucumber scenarios and 18 Rspec tests combining both BDD and TDD approaches. We have all the test cases working correctly. The scenarios defined in the cucumber test cases were based on discussions with clients. After which we worked to complete those test cases and write step definitions in the backend to emulate the behavior of users using gems like capybara gem which was used for testing JavaScript functionality. Given below is an example of test cases we define for a bad path and a good path of the user for testing the functionality of validating the responses:
We believe that this approach is important for getting a direction of work aligned with what the client wants and also helps figure out if something is not working before we push our code to production.

Configuration management

For every feature, there was a branch created for that feature; this branch was based off the main branch (which is currently deployed to Heroku). Then once the feature was completed, the developer of the feature would create a pull request on GitHub; the pull request required at least two review approvals before it could be merged into the main branch.

Release Process

The planning stage involved a detailed definition of the scope for each release, which included identifying the specific features and fixes to be included. Our development phase saw our team working on creating new features, rectifying bugs, and effecting necessary changes, with each feature being developed on its own branch that originated from the main branch. Our release strategy was dynamic, with deployments to Heroku happening immediately upon the merging of completed features into the main branch, rather than following a rigid release schedule. Once the app is deployed, you should test it thoroughly to ensure everything is working as expected. If there are any issues, you can roll back to a previous version. Heroku provides a user-friendly dashboard for managing releases and rollbacks. You can find more information on our wiki page. In the release process, we switched our database schema to Postgres for our development and testing environments. In the process of deploying our application to Heroku, we encountered the following issues:

- We experienced database incompatibility issues between SQLite and PostgreSQL, particularly in the handling of arrays. Our app was developed with SQLite, but Heroku uses PostgreSQL. Differences in array handling between these two systems led to deployment errors.
- During the development of user authentication, we had many redundant authentication classes that weren't used in local deployment. These extra classes complicated the deployment process on Heroku and reduce code coverage as well.

Issues with any other tools

Our team encountered a few obstacles with the tools we used during development. These obstacles came from a few different places, including Heroku, Github, Cloud9 IDE, and MySQL. As we were deployed on Heroku, we had their default gems loaded in as part of our project. One gem that came with this default set was ZenTest. This gem had some capability issues with other gems that were going to be used for testing, causing us to have to find alternatives to the way we were implementing those tests. We did utilize GitHub for the version control. While it didn’t have any issues with how it ran, it was a bit of a challenge initially for some team members to get acclimated with the role it played in a large development environment setting. As we progressed through the project this got better as everyone familiarize themselves with it. As we set up with cloud9 IDE, it got challenging to run the web app during the dev process. This challenge was fixed by setting up a local IDE to run it smoother. Lastly, we initially used SQLite for our development database but after issues with array formats, we made the switch to PostgreSQL instead and finished our development with PostgreSQL.

Specific Tools/Gems Utilized

Our team used several helpful tools and Gems while developing our web-based application on Ruby on Rails. Let me give you a rundown of some of the most important ones: First, there's Devise - this Gem made it easy for us to build a secure user authentication and authorization system without having to spend valuable time coding from scratch. To test and preview the emails our app sent during development, we used MailCatcher. With MailCatcher, we could examine email messages in a local web interface without actually sending them to real email addresses. This was useful in ensuring our app's email functionality was working properly before we deployed it. We also migrated from SQLite to Postgres for our development and testing environments, to match the database management system we used in production. Postgres gave us robust functionality and performance, and by using it throughout development and testing, we could catch any potential issues earlier in the process and avoid unexpected problems in production. Please click here to check our wiki page on how to shift to Postgres in development. To track our test suite's code coverage, we utilized SimpleCov. This handy tool allowed us to see which parts of our codebase needed testing, so we could improve our testing efforts and ensure that the app was thoroughly tested. Overall, these tools and Gems were crucial in helping us develop a high-quality and reliable web-based application.

Discuss how to deploy Application to Heroku

The guide shared here presents the necessary steps to deploy updates to the application.

Relevant Links

Github: https://github.com/sidbav/public-health-app-v2
Pivotal: https://www.pivotaltracker.com/n/projects/2630179
Heroku: https://healthapp.herokuapp.com/
Presentation and Demo Video: https://youtu.be/kLNxoFKhSWE