

KSRE Survey App

Overview

This web app is intended to help people within KSRE track their progress when it comes to day-to-day activities such as: meeting with people (*Direct Contact*), doing interviews or podcasts(*Indirect Contact*), hosting events(*Event*), etc. Some of this information is reported manually by the people within KSRE to the local government whereas the rest of the information is used internally.

Primary point of contact for this project is Frances Graves, Director and Community Vitality Agent, K-State Research and Extension - Wildcat District: francesgraves@ksu.edu

Functionality

This app is not intended to work with other systems such as: PEARS (reporting software KSRE employees use) or anything else. All information submitted to the web app is either entered manually into another system by the person using it or viewed internal to understand their progress. The only additional features that need to be added are posting information from the web app to a database under that user. Additionally, importing the data whenever the person is logged in so that if they decide to export the data as an Excel spreadsheet, it will contain everything they've reported up until that point.

Note: A lot the more technical details such as SQL tables, JSON structure, and Authentication endpoints are listed written in the repo.

What we've accomplished

We designed a front end using React.js. The user interface was created by meeting with clients and going over needs/wants and necessary features required by the app. As it is currently, the app gathers all the information it's supposed to and stores it locally on the client. Additionally, it's designed to adapt to both the mobile and desktop views.

All pages within the app have comments that explain the pages and functions. SQL Table creation scripts and Table diagrams are included within the Github repository.

Initial wireframe: <https://app.moqups.com/VLgzgFXNMg/view/page/ae8fe8eb0>

The wireframe also includes clients notes, requirements, and general project information.

Github Repo: <https://github.com/trevorreynolds2020/ksre-survey>

What still needs to be done

Handle authentication, create a database, and pass information back and forth using a backend (tentatively AWS Lambda) approved by K-State IT (talk to: Greg Dressman dressman@ksu.edu; Don Eisele eisele@ksu.edu; Robert Wirtz rob@ksu.edu)

Notes about what needs to be done

At the end of the day, your best bet is using AWS, we think at least

We started this process using .NET however K-state IT was unable to support this because it requires too much upkeep and maintenance. Therefore, a more long-term solution will be a more lightweight backend such as AWS. This would be hosted on a K-State IT server as previously believed it would be hosted on the Extension server. We included more specific information about implementing both the authentication and database with our repo (cd backend/backend_concept).

The last communication from K-state IT was the following:

"Don, Rob and I had a short discussion following our meeting and so I wanted to get back with the team with some options. We are not able to provide VMs for the web and database tiers but we could potentially migrate a solution into our AWS environment if it was running on one of their platform offerings such as API Gateway, Lambda, etc. Here are a few links with more information.

<https://aws.amazon.com/api-gateway/>

<https://medium.com/@maeluenie/aws-lambda-function-with-dynamodb-and-api-gateway-40f4960937f0>

<https://aws.amazon.com/education/awseducate/students/>

Conversation between our Professor (Prof Bean) and IT guys(samurai guy) about authentication

Prof Bean: "How K-State is handling eid-based authentication with these services. I assume you have an already-established approach? If so, could you share or direct me to resources on it?"

IT guys: "If we can keep the web tier in Client side scripting frameworks then what we'd do is host that in the CMS and use .htaccess files to force SSO. <https://www.k-state.edu/webservices/cms/htaccess/restrict-eid-password.html>"

Us: *We hope this helps somehow*

Sidenotes

95% of the code we added is in the 'src' folder. Most additions were components, redux, and styles. We didn't mess with much outside of that.