The STEM Academy Data Solution

Technology Review

Shannon Ernst, Kyle Nichols, Javier Franco

14 November 2016

CS 461 Fall 2016

Group 48

——————————————— ◆ ———————————————

Abstract

The STEM Academy Data Solution will manage a database of survey questions. Its purpose is to automate the process of aggregating survey data and generating reports on this data. For this technology review, split our product into subcomponents and compared three different alternatives for those subcomponents. We analyzed these alternatives and came to a decision on which technology we would use for each.

1 INTRODUCTION

Data drives much change in today's world with the advent of data technology. The STEM Academy in Corvallis, Oregon has been collecting data on their various camps and programs by paper surveys. This data has often gone untabulated due to the shear amount of surveys. They now wish to move to a digital variety. The STEM Academy Data Solution is comprised of three fundamental parts: survey generation, data storage and data analysis. At first glance, an obvious solution to the problem would be to use a piece of preexisting survey software such as Survey Monkey [**?**] or Qualtrics[**?**], both of which allow a user at a basic level to create surveys with various question types, distribute the surveys via a web link and view their collected data according to the survey. The primary drawback to both these solutions is that the data would no longer be controlled by the STEM Academy and would be soley reliant on these programs. A secondary drawback is the cost of these services which work on a premium basis[**?**]. Finally, the learning curve for the software is steep and contains many features which the STEM Academy does not require, making the software unmanageable for the STEM Academy. The STEM Academy needs a solution which will allow them to maintain control of their data and meet their specific needs without additional fluff. As the STEM Academy is largely nontechnical, the software needs to be incredibly simple to use and require no maintainence upon delivery. The following report will examine the techonologies which can be used to construct the solution. Shannon Ernst authored Section 2: Survey Generation. Kyle Nichols authored Section 3: Database Management, Section 4: Survey Distribution, Section 5: Printable Surveys and Section 6: Decision on Database Management, Survey Distribution and Printable Surveys. Javier Franco authored Section 7: Search Form, Section 8: Printing a Report, and Section 9: Saving Report Results.

2 SURVEY GENERATION

In generating surveys, the user should be able to add/remove questions, adjust their ordering at any time, and control how and when they are displayed. There are a multitude of ways to accomplish this and the results are as follows.

2.1 Question and Survey Creation: Logic and Storage

The primary issue of survey generation is figuring out how to extract the information from the user that needs to be asked in a question and produce a digital form variant of it. If we were hard coding the surveys and questions, we would just create an html form with supporting logic and set it up so once a participant has filled it out it would post the responses to the data base. However, our questions and survey generation need to be dynamic as the questions, ordering and content are likely to change over time and according to the camp or program.

2.1.1 Option 1: FluidSurveys API

In order to avoid reinventing the wheel, an API could be used to help generate surveys. FluidSurveys is an API provided by Survey Monkey to generate questions and surveys. The surveys, questions and responses will be stored via FluidSurveys. The API returns JSON objects which could be used to generate the user interface of the project[**?**]. As this is an API, the requesting of information from the user can be severly masked as they could simply provide a list of questions, types, and answers and a survey will be generated with a publically distributed link.

2.1.2 Option 2: Qualtrics API

The Qualtrics API provides a variety of survey services depending on which of their platforms you want to use. For this project we would use there Survey Platform which allows for essentially everything but the creation of surveys[**?**].

You can only use the API if you have access to a license, which OSU does have[**?**]. You would construct your survey using their tool and then use the API to access, update, distribute and view responses of the survey. They provide more export components and transformation functions. All of their RESTful functions return JSON [**?**].

2.1.3 Option 3: From Scratch with JSON

One thing that most survey APIs have in common is that they store the construction of their survey and questions in a JSON structure. This will allow it to be interpretted in multiple different ways as it is just holding data. For example, a question JSON may be represented as such:

```
{
    "id": "ajfoijfe24",
    "type": "multi",
    "question": "What is your favorite animal?",
    "answers": "dog, cat, tiger, dolphin"
}
```

This JSON for questions could then be included in a JSON for a survey which would be constructed similarly. Then when the survey is being built the user could simply select what type of question they want to insert and the JSON would continue to be constructed. The general set up of the JSON would also make it easier to store and allow for a variety of display interpretation.

2.1.4 Discussion and Recommendation

The primary issue at hand is cost and security of data. The STEM Academy wants their data kept in house meaning that farming out survey generation and data collection to a third party API is not ideal. While both FluidSurveys and Qualtrics offer many aspects which allow for the simpler creation and construction of surveys, due to having to rely on them to host the surveys is a deal breaker. In addition to losing direct control over the surveys and data, neither API is fully open source and some of their key feature require payments in order to get the API keys [**?**] [**?**]. Both however show clear structuring of JSON elements which are both very similar to one another. This gives a strong lead to simply construct the questions and surveys based on a similar JSON schema. The benefit to constructing our own JSON schema is that we can store the surveys and questions in our database without having to pay premiums for the API keys. We can also strip out a good portion of the functionality that we would never use. Therefore, constructing our own API from scratch is the best option.

2.2 User Interface of Creating Survey

The primary issue of the user interface of creating surveys is, regardless of what is used to actually generate the survey, the survey creator needs to have a sense of what their creating. They should be able to quickly and accurately create their questions and adjust the formatting of their survey. The fundamentals of the survey is creating a dynamic form.

2.2.1 Option 1: From Scratch with Backbone

Backbone is a javascript framework which is base on model-view-presenter which is the parent process of model-view-controller. It is closely linked to UnderscoreJS for page templating services. Backbone does not do any form of data binding so it is faster for it with the trade off of if you wan the binding you have to write it yourself. This also means

that Backbone relies on direct DOM manipulation to enact changes [**?**]. Backbone could be useful as it is lightweight, well documented and fast for large systems.

2.2.2   Option 2: From Scratch with React

React is a javascript library intended to focus primarily on the view portion of model-view-controller using a virtual DOM to selectively update portions of it. It generates the markup via javascript so you simply need to include the link to your javascript in the html [**?**]. React is based on components, and sometimes states, but is primarily just javascript. Essentially, the javascript chooses how best to update the view in the most efficient way as data changes. React can also be rendered on the server meaning it can load faster than client based rendering. One problem with React being just the view component of model-view-controller is that it does not have any support inherently for AJAX, event systems, data layer or promises. This means that it either needs to be paired with a framework or many things need to be constructed seperately [**?**].

2.2.3   Option 3: From Scratch with AngularJS

AngularJS is a javascript framework which is based on model-view-controller. It has a two way binding system that binds the html and javascript together so that elements are able to dynamically appear with greater ease by continually updating the DOM [**?**]. By setting up templates of questions in html, these html snippets can be included multiple times in the survey. These templates would be repositories for inputs or sections of forms. Upon being submitted the information would be used to construct JSON objects which would represent the format of the survey. AngularJS is constructed with testabilitity and modularity in mind however is has a somewhat complex system of directives, controllers, factories, scopes and services which makes debugging difficult. It also requires it's own digest cycle to bind the data and the html which can make the timing of events difficult to track in debugging [**?**].

2.2.4   Discussion and Recommendation

The UI for creating the survey is going to require that templates are used. This is due to the nature of providing question types such as multiple choice, matrix and text entry which are going to be fundamentally different in their html representation. Using a model- view set up will allow for us to dynamically add these templates as the user adds them to the survey. As the user fills in required fields for these templates to construct their questions, we are going to want the data to be captured. This would best be done with the two way data binding of AngularJS however the bigger the survey, the slower it will be due to this massive amount of binding. Backbone or React would be a better option if we are looking for speed however the survey creation is not going to be largely event driven eliminating the primary function of Backbone. As Backbone and React are both fairly light weight it is likely that there would be more overhead in generating code to get the functionality we desire. It is recommended that AngularJS is used for survey generation UI as it will allow for the developers to more easily seperate out concerns, create more html templates which will be specific to each question type and also dynamically store the information provided by the user via two way data binding.

2.3   Creation of User Interface for Survey Taker

A participant needs to be able to take a survey and the their view will be generated based on the survey constructed by the user. This means that the participants survey view will be generated by predetermined schema, not a hard coded form. This dynamic quailty is the primary issue that needs to be addressed as most modern surveys allow for display

logic and control flow. One of our requirements is usability which means the user should not have to manually skip over static elements. The survey must be reactive to input and based on what the user designated in the survey generation step.

2.3.1   Option 1: Angular Schema Form

Angular Schema Form generates a form based on JSON schema in a Bootstrap 3 style. It is open source and has many add ons for various styles, question types and features [**?**]. It posts JSON information back as it relies on AngularJS for two way data binding. The library is currently undergoing refactoring[**?**]. This would also require that AngularJS is used which has its own limitations (see Section 2.2.3).

2.3.2   Option 2: From Scrath with React

React is a javascript library intended to focus primarily on the view portion of model-view-controller using a virtual DOM to selectively update portions of it. It generates the markup via javascript so you simply need to include the link to your javascript in the html [**?**]. React is based on components, and sometimes states, but is primarily just javascript. Essentially, the javascript chooses how best to update the view in the most efficient way as data changes. React can also be rendered on the server meaning it can load faster than client based rendering. One problem with React being just the view component of model-view-controller is that it does not have any support inherently for AJAX, event systems, data layer or promises. This means that it either needs to be paired with a framework or many things need to be constructed seperately [**?**].

2.3.3   Option 3: JSON Form

JSON Form is an open source javascript library which creates an html form based on a JSON schema. The html includes client side validation where available. Upon completion of the form it returns javascript data structure model which matches the form. It is reliant upon jQuery and UnderscoreJS. The library has not been updated in three years [**?**].

2.3.4   Discussion and Recommendation

Largely the generation of what the survey looks like will be determined in how the survey creation is stored. React would be a good option as it adjusts the view based on provided data meaning there could be more complicated control flow and display logic. The Angular Schema Form would allow for dynamic data binding and also dynamic control flow. Both of these require that the developers do more pre and post processing to ensure the form comes out correctly. The JSON Form library is set and would not require any additional set up, merely read the JSON. However it would not allow for dynamic control flow. It is recommended that React be used as it is lightweight and will allow for dynamic control flow. Though it will take more intial set up from JSON it will still be faster that Angular Schema which is also undergoing a refactoring leading to some instability of features.

3   DATABASE MANAGEMENT

For the database, we need to be able to set up the database (including the tables, relationships, etc.) and store it at a location accessible to our clients.

3.1 Microsoft Access

Microsoft Office contains Microsoft Access as a tool for creating custom SQL databases accessible from a browser [?]. It stores those databases on its own Microsoft Azure SQL Database service [?]. The advantage of using Microsoft Office software is that anyone with an Oregon State email has access to the Microsoft Office suite, which includes our clients [?]. Microsoft Access, in particular, can be installed on institutional computers by contacting the department's computer administrator, though there may still be a cost involved in this as well [?]. This is also on institutional computers only, which limits the access we have to Microsoft Access for developing the database. Microsoft Access also utilizes SharePoint to allow for sharing of database among multiple users and database permissions [?]. This would be really useful, as our clients can share access and permissions among any employees of STEM Academy that would need it, and since we will likely also need access and permissions while developing the project, we can use our own Oregon State email accounts for access.

3.2 phpMyAdmin

phpMyAdmin is a free tool written in PHP for accessing MySQL databases, specifically over the web [?]. Since it is written in PHP, if we develop our system as a web application, it will become easier to write the system using PHP and use PHP to generate queries to the database. phpMyAdmin utilizes a web interface for setting up and editing its database, though this may see limited use as our clients will likely be using our custom web interface for dealing with the database. phpMyAdmin is also very well documented, which makes for ease of development [?]. Another strong advantage is that phpMyAdmin is free, cutting costs for our client, though a separate web server will also be necessary.

3.3 MongoDB

MongoDB is an open-source document-oriented database software built on NoSQL [?]. MongoDB is also document-oriented, which means that it organizes and manages data as documents [?]. This type of structure and orientation fits well with the storing of surveys, which are inherently documents, however, this does not work for storing data on students and their responses, which may require a more traditional relational database system to connect the students with their survey responses. As open source software, using the bare bones software for MongoDB is free of cost, which reduces or eliminates the cost for our clients to uphold the database. [?]. However, this is for the bare bones software, and if there are additional features needed by our clients that MongoDB provides, that will require additional cost [?].

4 SURVEY DISTRIBUTION

Surveys must be distributable in URLs or other formats so that survey takers can easily access the survey from their own computer. The most useful format would be through email.

4.1 PHP mail

This is the built-in mail functionality for PHP [?]. The `mail()` function covers all of the basic requirents of sending an email (sender, destination, message, additional headers such as CC information) [?]. Emails generated by HTML, similar to MailChimp below in allowing for customization of emails, can also be sent by the `mail()` function [?]. A simple, built-in function has the advantage of easily being used and integrated in our system, which may likely use PHP, without employing outside systems and services that come with more features than are necessary. PHP's `mail()`

function doesn't employ any encryption and privacy features or automation, however. If we used this as our way of mailing out automated emails, we would have to do extra work to encrypt it and to generate the emails on a mailing list. These are still things that can be done from scratch in the PHP code, but it is also more code that we would have to write for the system. Another downside to generating automated emails on a mailing list is that the `mail()` function opens and closes an SMTP socket on each email, which is inefficient for iterating over a mailing list [**?**].

4.2   PHPMailer

In direct contrast to the built in `mail()` function for PHP, PHPMailer is an open-source API for sending emails from PHP [**?**]. PHPMailer uses the same basic setup details for emails that the `mail()` function does, but it also includes several other additional features that the `mail()` function does not [**?**]. SMTP support is integrated in PHPMailer, meaning having a separate, local mail server is not necessary [**?**]. This feature in particular could be useful if STEM Academy does not have the resources for a local web server and only has access to their own @oregonstate.edu emails. The SMTP integration also includes authentication [**?**]. Emails in PHPMailer can use HTML for customization, but can also send alternative, non-HTML versions for non-HTML clients [**?**].

4.3   MailChimp

MailChimp is a tool used for sending out marketing emails and automated messages [**?**]. Emails can be generated using HTML and CSS, and collaboration options are built in for working on those emails as a team [**?**], [**?**]. Tools also exist for automating emails, though this is more target towards marketing and usage, and less mailing lists, which is all our clients need [**?**]. MailChimp also provides an API for use in developing [**?**]. As marketing software, MailChimp has many features and uses, many of which would be superfluous for our clients, who only need a way to provide a quick-access link to the surveys and distribute them in a way that survey takers can access them from many different locations. Also as marketing software, it's purpose is drastically different from what we would be using it for. Email lists and automated messages are provided by MailChimp for this, but it could be easier to use different tools that are more focused on just those features.

5   PRINTABLE SURVEYS

Surveys that are generated on our system must have the ability to be printed and distributed on paper instead of just online.

5.1   HTML `window.print()`

This is a built-in function for HTML which will print the web page by opening up the print dialog box for the browser being used [**?**]. The `print()` function exists as part of the Window object in HTML, which represents an open wndow in a browser [**?**]. It is supported by all browsers, though without any standard enforced between them [**?**]. The ease of this method is that it is supported by all browsers and can use their printing functionality, which might provide a wide variety of options in how it is printed. Unfortunately, this means it also not an automated method, though our clients may not need anything automated and may just want a way to easily print the surveys generated. The Window object contains many different properties and functions that could be used in other areas of our web service, though this is the only one that would apply to our specific need for printing of surveys and other pages.

5.2 CSS

Cascading Style Sheets describe how HTML web pages are display on screen or other media, such as paper [**?**]. It can even control multiple web pages at once [**?**]. Although CSS can be useful in the general development of our web service, it can also be used for formatting and displaying a web page. This is because HTML only describes content [**?**]. Since we will likely be using HTML or PHP to display content, CSS will probably be necessary to not only make our web pages user friendly in appearance, but also to ensure that they are formatted in a way that pages are easily printable. Although we may primarily use a different, defined method for printing pages, we may also use CSS in conjunction to format our pages so they look clean and formatted for printing.

5.3 css2pdf

This is a Javascript library intended to be plugged-in to a website for creating a print button on a web page that produces a PDF as embedded or downloadable (though some features are only supported by certain browsers) [**?**]. The format of the generated PDF can be set and any HTML or CSS stylings are applied to the printed selections [**?**]. This allows us to generate a button on the page without creating the code to do so from scratch and applying it to a built-in function, like `window.print()` above. It supports several HTML elements such as different block elements, tables, and lists [**?**]. It is also in process and being improved and will likely add more features and capabilities in the future, though this may not be useful as we will most likely finish developing this project at the end of the year.

6 DECISION FOR DATABASE MANAGEMENT, SURVEY DISTRIBUTION AND PRINTABLE SURVEYS

Whichever software we use for the database setup and storage will need to depend on what cost Microsoft Access would have for STEM Academy. If the cost is manageable for STEM Academy, we could use Microsoft Access. However, if it is not, we may decide on phpMyAdmin. Although it would require accessing a web server that can be hosted on Oregon State's infrastructure (which is done in database classes), it is free software that can accomplish what we need without any worries of cost. For distributing the surveys, our best solution will probably be PHPMailer. It provides the functionality of the built-in `mail()` function, but with extra options that could be useful for us, including not requiring a separate web server. For printing surveys, the css2pdf library looks like a more rounded option than the default `print()` function without having to do some extra work on our part (such as creating our own print button).

7 SEARCH FORM

The goal of the search form is to be able to make queries based on camp, club, program, student, demographic information, and survey questions. The form should also allow to make queries using a combination of searh criteria. The user interface for the search form should be easy to use and should not require any knowledge of how to make a mysql query, so that users that do not have a computer science background can interact with the system. The people involved in STEM Academy do not have a computer science background for knowing how to make mysql queries which is why there is emphasis on making the product user freindly, so that it can be used by any person regardless of their background.

7.1 Microsoft Access

In Microsoft Access you can create a a search form by using a technique called query by form (QBF). First you make a form that contains empty text boxes for the search criteria and each text box represents a field in a table that is going to be queried [**?**]. Empty text boxes will be ignored [**?**]. To submit your search query you click on a command button on the form [**?**]. When the command button is clicked it uses a macro that calls a query to produce the query results.

7.2 FlySpeed SQL

FlySpeed SQL is a query tool whose objective is to make the management of data a simple process and it supports data servers that include MySQL, SQL, and PostgreSQL [**?**]. It allows users to search and change their database by using two different views which include a grid format and a custom form view [**?**]. The database connection and queries generated are saved, so that users can return and work on them at a later time [**?**]. The queries that are executed are are stored automatically[**?**]. FlySpeed SQL is a tool that can be downloaded for free. However, there is a paid desktop version that has some additional features that are not available in the free version and it costs $39 for a 1 year subscription and $99 for a lifetime subscription[**?**]. The paid version features transferring office formats, printing data, and saving reports as a PDF [**?**].

7.3 QueryTree

"QueryTree is a web based ad hoc reporting tool that works with any Microsoft SQL Server, MySQL or PostgreSQL database" [**?**]. Reports are created using a report building widget that is easy to use and QueryTree can be used in various devices [**?**]. A user first selects a table and then is able to produce queries using widgets, produce basic calculations, and create groupings [**?**]. QueryTree is not a free tool. It costs $29 per month and the tool can be used in two databses and allows up to ten users [**?**].

7.4 Discussion

The query by form technique that uses Microsoft Access is an easy method to use. A user simply enters their search criteria into input boxes and submits their query by clicking a command button which returns the query results by checking the table fields in the database using the search criteria. To create our search form we need individual input boxes for camp, club, program, and multiple input boxes for the demographic information. Furthermore, we will need a predefined number of input boxes for searching questions, but this is a problem since we do not know the number of questions that a user will use in their query. For example, if there are only two input boxes and a user wants to search for all the students that chose the color red, attended North Salem High School, and have parent's with a college education they will not be able to do their query. The user would have to manually add more input boxes for the number of questions that they are going to query and this requires knowledge of how to use Microsoft Access which makes this method not user freindly and ineffective. Another issue with this method is that we will need to figure out how to make the Microsoft Access form into a webpage for the website user interface.

Some issues with the QueryTree tool are that it has a monthly payment of $29, only ten people are allowed to use the tool, and two databases can only be connected. The number of users that are allowed to use the tool can be an issue if STEM Academy plans on having more than ten users that will be generating reports. If more than ten users will be generating reports the subscription will have to be upgraded which will cost more money per month to use the tool.

The constraint of only being able to connect two databases will not be an issue if we are planning to store all of STEM Academy's information into one database. The benefits of QueryTree are that it is simple and easy use for searching information in a database. The user simply uses widgets to create queries to search the database.

Based on the screenshots I did not like how users generate queries to search a database using the FlySpeed SQL tool. A user clicks on fields that are on the tables to generate a query and it also displays text showing the query. The tables and query text make the FlySpeed SQL tool look difficult and intimidating to use which makes the tool not user freindly. A good thing about the tool is that it can be downloaded for free which helps people to determine whether or not the tool will be useful to them.

7.5 Recommendation

The tool that I decided to choose for making the search form is QueryTree. The reason why I picked it is because it is user freindly unlike FlySpeed SQL. In addition, it will not have any issues generating multiple queries based on survey questions. I think that the usability of the tool outweights the costs instead of having to use two other free technologies that have issues.

8 PRINTING A REPORT

Reports generated using queries based on camp, program, club, demographics, student, and survey questions should be printable at all times. This will allow STEM Academy to present these results to their sponsors and on their applications to help them obtain more funding for their organization. When a report is printed it should only print out the report results and it should exclude parts of the webpage that are not related to the report. For example if a user printed a report it should not display the title of the webpage and the navigation bar. A print icon should be visible near the report to make it easy to print. The print options displayed should allow a user to choose the orientation, number of copies, margins, printer, etc.

8.1 JavaScript with CSS

First I will discuss Javascript with a CSS style sheet. When a user clicks on on a print button or hyperlink using the window.print() method it prints out the entire webpage[**?**]. To resolve the issue a separate CSS style sheet is created to exclude unwanted parts of a webpage before printing a document [**?**].

8.2 JQuery Print Preview plugin

"The jQuery Print Preview plugin is designed to provide visitors with a preview of the print version of a website" [**?**]. "Unlike traditional print previews this plugin brings all content and print styles within a modal window" [**?**]. The jQuery Print Preview plugin consists of a source code and a demo. In the demo when a user clicks on the print icon or print this page hyperlink it displays a modal window that covers the entire page with the print version of the webpage. At the top of the modal window there are two icons for exiting the modal window and for displaying the print dialogue box which allows the user to choose their print options. The modal window forces the user to interact with it before returning to the main webpage.

8.3   Smart widget

Smart widget makes printing more efficient by removing unnecessary things from a webpage which include headers, footers, and advertisements before showing the print dialogue box [?].The smart widget tool has extra features that include a print preview, edit option for removing additional content, and saving a page as a PDF [?]. The website that provides the tool has the code available for incorporating the smart widget into a HTML webpage and it is only one line of code.

8.4   Discussion

The Print Preview Plugin can be adapted to help print a user's reports. The modal window that appears when one of the print links is clicked seems redundant. The reason why it is redundant is becuase the print dialogue box displays the print version and it also shows what is going to be printed on each page.

The smart widget tool is easy to incorporate into an HTML webpage. It also saves time by not having to write the code that excludes certain parts of a webpage. If we incorporate this tool into an HTML webpage we must verify that it is not removing content from the report that is going to be printed. If no content is removed from the report then this tool will be considered the best option.

To create a print version of a webpage using JavaScript and a CSS style sheet is not hard to implement if you have experience in web development. By making our own CSS style sheet for the print version of the webpage it ensures that the report form will be printed to our liking.

8.5   Recommendation

The tool that will be selected for printing a report is JavaScript with a CSS style sheet. The reason why I chose this tool is because we will be creating the CSS style sheet which will guarantee that only the report form is going to be printed. This tool is not redundant like the jQuery Print Preview plugin since the print icon or link will only display the print dialogue box.

9   SAVING REPORT RESULTS

Saving reports into a database will allow administrators to edit the title and captions of the data. It will also allow them to print their reports whenever they need them instead of having to generate another report and trying to recall the queries they used to obtain the same results. Saving reports should be an easy process for an administrator since one of our objectives is to make the user interface user freindly.

9.1   Izenda

There are two ways that Izenda stores reports which include a file system mode and a database mode [?]. In the file system mode reports are stored in a default folder that is named Reports [?]. In database mode reports are stored in a default database table called IzendaAdHocReports [?]. Some benefits of using the filesystem mode over the database mode include quicker access to stored reports, customization of security setup, and customization of how reports are distributed and stored [?].

9.2 Access

To save a report in Access you just click the Microsoft Office Button and then click save [**?**]. If the report does not have a name it will require a name in the Report Name box field [**?**]

9.3 QueryTree

"QueryTree is a web based ad hoc reporting tool that works with any Microsoft SQL Server, MySQL or PostgreSQL database" [**?**]. Reports are created using a report building widget that is easy to use and QueryTree can be used in various devices [**?**]. The QueryTree user interface is simple and easy to use. A user first selects a table and then is able to produce queries using widgets, produce basic calculations, and create groupings [**?**]. QueryTree is not a free tool. It costs $29 per month and the tool can be used in two databses and allows up to ten users [**?**].

9.4 Discussion

Saving reports in Access is easy, but we will have to generate reports using Access. This will be a problem because we need to generate reports in a webpage for our user interface and it must also be easy to use. By choosing to use Access to save reports we will not be able to satisfy our requirements because the user will require knowledge on how to use Access and we will fail on having a webpage form for generating reports.

The two different options for storing a report using Izenda is a nice feature. The only problem is that it requires you to install the Izenda tool in order to able to use the two methods. I searched for a price in the Izenda website, but was unable to find any prices. However, you can send your contact information which also includes your company name, number of employees, and estimated number of end users to get a price for using their tool [**?**]. The information that Izenda requires to obtain a price makes me assume that the tool is expensive and could potentially cost thousands of dollars to use. One of our goals is to use tools that will minimize the cost of creating our product for STEM Academy. For this reason Izenda is not a good option for storing reports into a database.

The QueryTree tool has the ability to save reports which helps satisfy the requirement of saving reports. In addition, it will help satisfy the report generation requirement due to its user freindly interface for generating reports. The only issue about the tool is the monthly fee that is required to use it. If we decide to use this tool we have to certain that it will be used, so that money is not wasted on purchasing a tool that will not end up being incorporated into our project. An issue that we will have to resolve is figuring out have to incorporate into our website for our user interface.

9.5 Recommendation

The best option for saving reports is the QueryTree tool. It not only saves reports it also generates reports using a user freindly interface that uses widgets. By choosing this tool I will be able to satisfy two requirements that I am responsible for which includes printing reports and creating a search form. Eventhough the tool has a monthly fee it is far more usable in our project comapred to the other two technologies.

REFERENCES

[1] 'Survey Monkey Homepage' [Online] https://www.surveymonkey.com/ [Accessed: Nov 13, 2016]

[2] 'Qualtrics Homepage' [Online] https://www.qualtrics.com/ [Accessed: Nov 13, 2016]

[3] 'Survey Monkey Plans and Pricing' [Online] https://www.surveymonkey.com/pricing [Accessed: Nov 13, 2016]

[4] 'Survey JSON' [Online] http://docs.fluidsurveys.com/fluidsurveys/api/json.html [Accessed: Nov 13, 2016]

[5]   'The Qualtrics Developer Hub' [Online] https://api.qualtrics.com/ [Accessed: Nov 13, 2016]

[6]   'Oregon State University: Qualtrics' [Online] http://main.oregonstate.edu/qualtrics [Accessed: Nov 13, 2016]

[7]   I. Herskovits, 'AngularJS vs. Backbone.js,' in Back&, 2015. [Online] http://blog.backand.com/angular-vs-backbone/ [Accessed: Nov 13, 2016]

[8]   'React: A javascript library for building user interfaces' [Online] https://facebook.github.io/react/ [Accessed: Nov 13 2016]

[9]   A. Ray, 'ReactJS for Stupid People,' in Andrew Rays Blog, 2014. [Online] http://blog.andrewray.me/reactjs-for-stupid-people/ [Accessed: Nov 13, 2016]

[10] 'AngularJS Home Page' [Online] https://angularjs.org/ [Accessed: Nov 13, 2016]

[11] K. Sanket, 'Why I Ditched Angular for React,' in Six Revisions, 2015. [Online] http://sixrevisions.com/javascript/why-i-ditched-angular-for-react/ [Accessed: Nov 13, 2016]

[12] 'Angular Schema Form' [Online] http://schemaform.io/ [Accessed: Nov 13, 2016]

[13] 'github: angular-shema-form' [Online] https://github.com/json-schema-form/angular-schema-form/blob/master/docs/index.md [Accessed: Nov 13, 2016]

[14] 'github: jsonform' [Online] https://github.com/joshfire/jsonform [Accessed: Nov 13, 2016]

[15] 'Database Software and Applications — Microsoft Access.' [Online] Available: https://products.office.com/en-us/access. [Accessed: Nov 13, 2016].

[16] 'Microsoft Software — Information Services — Oregon State University.' [Online]. Available: http://is.oregonstate.edu/service/software/microsoft-software. [Accessed: Nov 13, 2016].

[17] 'phpMyAdmin.' [Online]. Available: https://www.phpmyadmin.net/. [Accessed: Nov 13, 2016].

[18] 'MongoDB for GIANT Ideas — MongoDB.' [Online]. Available: https://www.mongodb.com/. [Accessed: Nov 13, 2016].

[19] Lijin Joseji, '11 OPEN NoQL Document Oriented Databases,' July 23, 2012. [Online]. Available: https://dzone.com/articles/11-open-nosql-document. [Accessed: Nov 13, 2016].

[20] 'MongoDB Cost Breakdown,' Dec 15, 2011. [Online]. Available: https://www.compose.com/articles/mongodb-cost-breakdown/. [Access: Nov 13, 2016].

[21] 'PHP: mail - Manual.' [Online]. Available: http://php.net/manual/en/function.mail.php. [Accessed: Nov 13, 2016].

[22] 'PHPMailer/PHPMailer: The classic email sending library for PHP.' [Online]. Available: https://github.com/PHPMailer/PHPMailer. [Accessed: Nov 13, 2016].

[23] 'About — MailChimp.' [Online]. Available: https://mailchimp.com/about/. [Accessed: Nov 13, 2016].

[24] 'Features — MailChimp.' [Online]. Available: https://mailchimp.com/features/. [Accessed: Nov 13, 2016].

[25] 'HTML Email Basics — Email Design Reference.' [Online]. Available: http://templates.mailchimp.com/getting-started/html-email-basics/?_ga=1.68323481.962129787.1479082297. [Accessed: Nov 13, 2016].

[26] 'Window print() Method.' [Online]. Available: http://www.w3schools.com/jsref/met_win_print.asp. [Accessed: Nov 13, 2016].

[27] 'Window Object.' [Online]. Available: http://www.w3schools.com/jsref/obj_window.asp. [Accessed: Nov 13, 2016].

[28] 'CSS Introduction.' [Online]. Available: http://www.w3schools.com/css/css_intro.asp. [Accessed: Nov 13, 2016].

[29] 'Xportability/css-to-pdf: Convert any HTML page or region to PDF - supports CSS, SVG, embedded XML objects, and more...' [Online]. Available: https://github.com/Xportability/css-to-pdf. [Accessed: Nov 13, 2016].

[30] 'css2pdf@cloudformatter.' [Online]. Available: http://www.cloudformatter.com/CSS2Pdf. [Accessed: Nov 13, 2016].

[31] Adams, Joseph. "Storing Reports." *IZENDA. N.p., 20 Feb. 2015. Web. 14 Nov. 2016.*

[32] *"Create a Simple Report."* Microsoft. N.p., n.d. Web. 14 Nov. 2016.

[33] "FlySpeed SQL Query." *Active Database Software. N.p., n.d. Web. 12 Nov. 2016.*

[34] *"FlySpeed SQL Query Purchase."* Active Database Software. N.p., n.d. Web. 14 Nov. 2016.

[35] Deering, Sam. "10 JQuery Print Page Options." *sitepoint. N.p., 13 Mar. 2016. Web. 11 Nov. 2016.*

[36] *"How to Print a Specific Part of a HTML Page Using CSS."* ARCLAB. N.p., n.d. Web. 11 Nov. 2016.

[37] "How to Use the Query by Form (QBF) Technique in Microsoft Access." *Microsoft. N.p., 27 Mar. 2007. Web. 11 Nov. 2016.*

[38] *"QueryTree."* Capterra. N.p., n.d. Web. 12 Nov. 2016.

[39] *Storing Report Definitions. Developer Express Inc, n.d. Web. 14 Nov. 2016.*

[40] *Tabona, Andrew. "Top 10 Free Database Tools for Sys Admins."* TalkTechToMe. N.p., 15 July 2015. Web. 11 Nov. 2016.

[41] "Web Page Print and Pdf Button." *100widgets.com. N.p., n.d. Web. 13 Nov. 2016*

[42] *"Izenda Offers a Compelling BI and Reporting Framework at an Exceptional Value.."* IZENDA. N.p., n.d. Web. 14 Nov. 2016.