-

CONNECTED to RESEARCH

Benny Zhao, Chris Nguyen, David Winkler

Team 29, *The Majestic Turtles*

-

Abstract

Our goal is to create a system that improves the way researchers research funding opportunities. the system will pull data from funding sources, add opportunity data to a database, and present that data to users. The system is being developed as part of the *Connected to Research* project, with collaboration from researchers at Pacific Northwest National Laboratory.

We have identified 4 areas where we will have a choice between multiple technologies:

1. A way to pull data from funding opportunity sources.
2. A database system to store that data.
3. A way to use the database to present information to interested parties.
4. An agreed-upon language to increase maintainability.

This document will describe each of those parts and the technology our team can use to complete them.

-

Methods of Retrieving Data (Task I.a or I.f):

HTML Method: To extract opportunity data, we could search through the website's HTML files. This would involve using web scraping. We would need to implement a script that searches through the HTML document and extracts data from the website using tags. This would involve iterating through each opportunity on the website, reading through the HTML file until we reach an identified tag, and then associating the data from that field and inserting it into our database. For example, on FedBizOps.gov, the solicitation number of an opportunity is listed in a div as follows:

<div id="dnf_class_values_procurement_notice__solicitation_number__widget"...

This approach has many flaws. First of all, it would be very inefficient. We would need to load many webpages and generate many requests. The script we develop would also need to read through whole HTML files and match patterns. If the website we pull data from ever decides to change their website format, we would need to update our script. Clearly, this option should only be our last resort.

Pros: All of the information available to the site users will be accessible through our script.

Cons: It is very inefficient and susceptible to change. We have no guarantee that the HTML design will stay the same for long. The scripts will need to load many webpages and read through them all to extract the relevant data. This will involve many requests and a lot of processing. We will need to study the HTML layout of the site and investigate ways of iterating through the website using our chosen programming language.

XML Method: Another method would be reading through the XML file of each resource. Both FedBizOps.gov and grants.gov publish XML documents that contain information about funding opportunities. Using these files, we can use a script that automatically downloads the file and extracts data from it, pushing the formatted data to our database. For example, the solicitation number of an opportunity is listed in an XML file as follows:

<SOLNBR> ...

This approach is similar to the HTML method. However, this method is much preferred. Using this method we will have access to all available information in a clear and legible interface. Since the XML file is designed to contain only relevant information, we can pull data from it significantly faster than we could from any HTML page. The XML document data tags will likely not change for a long time, and if they do, such a change is likely to be announced. Unlike a change to the website HTML form. This way, we will be able to predict and prepare for changes before they occur.

As for getting the document, there are two types of XML documents. There is a weekly document, which contains all of the current funding opportunities, and a nightly document, which contains new and updated opportunities. If we use the weekly file, the one file will take longer to parse, but will likely be faster than parsing nightly files. However, the data gleaned from it will become out-of-date faster. If we use the nightly files, our data will be more accurate, but the files will need to be parsed daily.

Grants.gov publishes their entire database as a daily XML file. We could choose to parse the file daily, or on some other schedule.

-

Pros: XML files are designed for this. Both PHP and Javascript have supported methods for parsing an XML file. If necessary, there are many open source examples that we can adapt for our purposes. The XML files are an officially supported method to pull data from our funding sources. The fields in the XML files are not likely to change, even if the website design changes.

Cons: we will need to develop and maintain the scripts ourselves. We will need to inspect the XML files to determine what fields to extract. The XML parsing scripts will need to be scheduled, and we will need to decide when they should be run to keep our database up-to-date. We will need to determine which XML files to parse, and at what times.

FBOpen Method: 18f is a US government agency dedicated to making digital products for government organizations. One of their products is called FBOpen, an open-source project that allows users to search for funding opportunity data. Though it's named for FedBizOpps, it also allows users to search Grants.gov, with plans to incorporate more sources in the future.

Using this API, we can make use of an official API created by the government to access government data. We can use this API to avoid duplicating code and reinventing the wheel, and to improve the maintainability of the system.

The FBOpen implementation is open source and can be modified to suit our needs. According to its website, it is a "RESTful search API, hosted by the awesome api.data.gov, and backed by the search index server, Solr." The API can be used to query formatted data for all opportunities, or to retrieve information for a particular opportunity. If we choose to, we could store only the opportunity numbers and tags in our database, and retrieve all other information directly from FBOpen. That would give users the most up-to-date information about the funding opportunities.

Pros: FBOpen is developed by a department of the US government. FBOpen queries the FedBizOps and Grants.gov sources directly. Using FBOpen, our database might only need to store opportunity IDs, and associated data can be obtained through FBOpen. The project claims it will add support for more funding sources. FBOpen is free and open source, so it can be modified to suit our needs.

Cons: it might be better suited for individual queries, not to populate our whole database. We are limited to 10,000 queries per day, though we are encouraged to request more. Development is ongoing, and some features are not yet supported. Other features may change in the future. Our scripts may need to be updated when the FBOpen API changes.

Our Selection: We plan to use the FBOpen API. We will consider the XML parsing option as a way to populate our database initially. Parsing XML will probably be faster and more straightforward than querying every entry through FBOpen. Afterwards, we will keep our database up-to-date using the FBOpen API. Using this approach, we will be able to reduce the size of our database while also improving reliability. There may be a speed penalty due to the time it takes to query the FBOpen API. We believe that the penalty will be offset by the storage saved and increased efficiency of the component.

-

Determine the database system to use (Task I.e):

MySQL: This is an open-source relational database management system. The software's source code was developed and distributed by a Swedish company called MySQL AB under the terms of GNU General Public License. MySQL is now owned by Oracle Corporation. This was the first database management system that we were going to use. The reason for this is because we were all familiar with MySQL and it is open source. MySQL is supported on most operating systems such as Windows, Max OS X, and Linux. We will not be using MySQL due to our client's request of PostgreSQL.

Pros: MySQL is well known, so it is simple to manage our data. Scaling the project with MySQL would not be a problem due to simplicity which can lead to more functionality and/or automation testing. Doing simple read heavy operation is an ideal use of MySQL compared to other database management systems due to simplicity of MySQL calls rather than doing the same operation with complex calls that would slow down performance. MySQL would be useful in cases where there is a continuation of the project. In the most recent versions of MySQL, there are JSON features that can be used. This is a benefit because this project may require JSON support. MySQL also can be run from using the command line.

Cons: A disadvantage of MySQL would be that it does not provide as much features or potential functionality of PostgreSQL. The main disadvantage of MySQL is that it does not scale well with the project's performance as the project scales in size and functionality. This project may potentially deal with a huge volume of data where there may be numerous operations waiting to be processed. The concern is with poor performance as the project demands greater and greater volume of data and operations.

Microsoft SQL Server: This is a relational database management system developed by Microsoft. The most recent release being Microsoft SQL Server 2014 version 12, and ongoing with Microsoft SQL Server 2016 version 13 being released in 2016.

Pros: Microsoft SQL Server is known to have high performance since its 2012 product due to its support for performance optimization. Manageability is relatively simple to perform giving us flexibility. It has the ability of scale with the project's database as well as having a hybrid of a disaster recovery and backup tools.

Cons: Microsoft SQL Server is not open source which means we must buy a license in order to use it and it is expensive. A big problem with this database system is that it is not portable which means it can not be used in any operating system but Windows. Another disadvantage is that Microsoft SQL Server is graphical user interface (GUI) based. The reason why being completely GUI based is a disadvantage is because it does not perform well with low-bandwidth or high-latency connections. External language binding is difficult with Microsoft SQL Server as it may have different requirements such as creating classes to store the data you are querying or install extra drivers.

PostgreSQL: This is an object-relational database management system. PostgreSQL is developed by the PostgreSQL Global Development Group, a diverse group of many companies and individual contributors. This is free and open source software with a permissive free-software license. Our sponsor requires our team to use PostgreSQL for the database hosting. A great reason to use PostgreSQL is because it supports standard SQL and our sponsor is familiar with it.

-

Pros: PostgreSQL supports advanced SQL functionality as well as support for a variety of data types, such as arrays and user defined types. PostgreSQL also does performance optimization, like Microsoft SQL Server. PostgreSQL can be driven completely from the command line, like MySQL. A big advantage of PostgreSQL compared to MySQL and Microsoft SQL Server is its external language binding. Its external API, "libpq" is easily implemented to connect and be used from programming environments. PostgreSQL provides absolute data integrity and reliability which is crucial for this project in protecting users' data and information.

Cons: With PostgreSQL, due to having more complex functionality than MySQL, may potentially be less performant. An example of this would be simple read operation where MySQL could use a simple SQL command to do the work while PostgreSQL uses a more complex SQL command to do the same operation. This is the reason why systems that utilizes fast read operation should not use PostgreSQL.

Our Selection: We will be using PostgreSQL. Our clients chose PostgreSQL for us. They probably chose to go with PostgreSQL because of its extended functionality and the simplicity of connecting to it from programming environments. Another reason is because PostgreSQL provides absolute data integrity and reliability which is needed for our project. Lastly PostgreSQL was requested to be implemented by the client.

-

Different way to present the data (Task III):

Bootstrap Method: Our client recommends our team to use the Bootstrap framework as a way to display the aggregated funding resources to the user. As a team, we have decided that this will be our main method to implementing a comfortable front-end user interface for researchers to search for funding opportunities based on their interests. A strong reason for our team to use Bootstrap is because it is a popular and well-developed framework, yielding a sufficient amount of documentation if we require clarifications.

This approach is feasible because there is enough documentation and there is simple bootstrap snippets of code already implemented from the previous senior design group. Our tasks is just to develop more from the last group and essentially add-on more specific features.

Pros: Has responsive design for all devices. Has a grid layout for organizing the display of different resources on a single webpage. Has pre-processor for SASS and LESS.

Cons: Only supports limited amount of browsers compared to Foundation (Bootstrap supports only Firefox, Chrome, Safari, and IE8+).

Road-Map:

- Create a bookmarking feature for user, if they are interested in saving a funding opportunity.
    - Be able to save the bookmarks possibly in a database table so that it could send sorted information back to the user in chronological view.
- Create and test different displaying options that the user could possibly interact with.
- Create a share option, if user would like to share funding opportunity (My Research Circle, Researcher, or via Email Address)

Foundation Method: Foundation is an front-end framework made by the company ZURB and is our alternative. Most likely, we won't be using Foundation as our main way to present data since our client would like us to use Bootstrap to embellish the front-end. Also the fact that there is already small snippets of existing Bootstrap code from the senior capstone group before us, it is more advantageous to continue developing using the same framework. Another reason we picked Foundation as an alternative is due to it being the second most popular and well-developed framework beside Bootstrap. Also large companies utilize Foundation, which gives the framework itself more credibility.

This approach would also be feasible since it is one of the top web frameworks next to Bootstrap being the most popular. But our focus is moreover on the Bootstrap framework. As to why this framework is chosen as our alternative is due to the sufficient amount of documentation it has, it's credibility that the company has been around for a while and it supports more specific mobile devices.

Pros: Also has a grid layout template for organizing different sections of a webpage. Supports Chrome, Firefox, Safari, IE9+, iOS, Android, Windows Phone 7+.

Cons: Only has pre-processor for SASS compared to Bootstrap.

Zimit Method: A second alternative to Bootstrap we found was Zimit, which is also a front-end framework that is open source. The reason to picking Zimit as our second alternative is for it's capability in building prototype HTML5 websites with responsive features. Zimit also is light-weight, where it can

-

compile and minify files to smaller sizes for faster loading. It also has a unified style that can assist in creating a light-weight user friendly interface. Plus the framework supports the more popular web browsers such as Chrome, Firefox, Opera, Safari and Internet Explorer.

This approach is not as feasible as the two mentioned above due to the fact that it is fairly new compared to Bootstrap and Foundation. But it is still a good alternative if we choose to write code for testing and prototyping in HTML5. The documentation is fairly limited since this is a new open-source project.

Pros: Light-weight, based on modular and scalable framework. Has design skeletons and a suite of stylized components.

Cons: Only has pre-processor for LESS and not SASS compared to Bootstrap.

Our Selection: We plan on using Bootstrap as our way to present the data to the user. The reason behind to using Bootstrap is mainly due to the popularity and sufficient level of documentation it can provide to our team. Also our client pushes us to continue developing on the existing Bootstrap code from the previous group. Nonetheless, it seems that Bootstrap supports SASS and LESS pre-processors , plus it has a well responsive design.

-

Determine the languages to use (I.i):

PHP: PHP is a server side scripting language for delivering dynamic content in HTML pages. PHP has the advantage of having a very large codebase and many tutorials. Many open source projects are written in PHP. PHP is very widely used and supported by every major web browser and operating system.

PHP executes on the server, not in the client like JavaScript. PHP can be used to generate HTML files that service requests, but cannot be used to create interactive web pages. For that, JavaScript is necessary. Because of this limitation, PHP is usually used for server-side scripting, while JavaScript is used for interactive elements in HTML pages.

PHP was initially developed in 1995, and has become widely adopted. Due to its age and popularity, it has grown to include a wide variety of functionalities. PHP can be used to generate HTML pages, or it can be used to run scripts to process data. We can use PHP to parse the XML documents mentioned above. PHP also allows either object oriented or procedural programming. PHP can parse an XML file and insert the resulting data into a database.PHP is also compatible with Postgres, our chosen database system.

Pros: PHP has a long history. It can be used in many places. It has support for many database systems. It can be used to parse XML files or generate HTML pages. It can be used on most web browsers. It has a large codebase with many supported features.

Cons: It can't be used to create interactive web pages. PHP is designed for server-side web development, not client-side development. PHP is less often used for creating dynamic web pages than it used to be.

JavaScript: This is a high level, dynamic, untyped, and interpreted programming language most commonly used for the frontend. JavaScript is one of the three essential technologies of the World Wide Web (WWW) content production where a majority of websites utilize and support it. This is the programming language that all of the interactions with the user interface will be programmed in. We will be using JavaScript for the frontend.

Pros: Applications developed by JavaScript are reliably responsive. Task processes complete almost instantly, which helps with memory management since they don't have to be processed in the website's server and get sent back to the end user, which consumes bandwidth. JavaScript utilizes JSON which is used and seen as an object. The benefit of JSON is that it is more compact than XML and can easily be loaded into JavaScript for use. JSON's processing speed is fast and does not need parsing in comparison to XML since everything inside of it is seen as an object and can be called and used like a variable. A big advantage with using JavaScript is that there are many third party add-ons that extends JavaScript's functionality.

Cons: Programming on the client side can potentially be difficult due to the possible concurrency issues. Since this project is based on users searching the website which uses the database to obtain information, the frontend and backend must be able to work together synchronously. There are security issues with JavaScript. The issue is once JavaScript snippets are appended onto the web page they execute immediately on the client servers which can potentially be used to exploit the user's system. Rendering JavaScript may vary depending on the layout engine. This is a small issue and is becoming a smaller issue with the latest versions of JavaScript implementing a universal standard for rendering.

-

AngularJS: AngularJS is also known to be based off of JavaScript, but more importantly it is a JavaScript framework that uses directives to extend HTML and binds data with the use of expressions. AngularJS is known to be one of the more popular front-end frameworks used with HTML. It has a large community and is backed by Google, which gives the framework itself credibility. The advantages to using AngularJS is that it helps categorize the application into model-view-controller (MVC) components and the most notable feature is it's two-way data binding.

Pros: The top advantage of AngularJS is it's two-way data binding, where any changes on the view will be immediately reflected on the model and vice versa. This feature provides an instant projection of the model, where testing is simpler because it is easy to test the controller alone without the view dependency. Other pros to using AngularJS are it is fast in development once familiar with, good with applications that deal heavily with interaction of client side code and less code is needed for the same results.

Cons: The downside to AngularJS is that the basics are simple, but the learning curve gets harder. It is also hard to debug scopes within AngularJS and the directives can be hard to use.

Our Selection: We will be using PHP on the backend, and JavaScript on the front end. This choice was made for us by our clients, but we would likely have made the same decision. PHP is well-suited for server-side scripting, and JavaScript is well-suited for interactive web pages. We can use both in their separate spheres of operation to create an efficient and attractive web site. PHP and JavaScript are also compatible with each other. AngularJS was used by the previous capstone group, and we will be extending their codebase. Therefore we will likely have to make use of AngularJS in our code.