# GreenAZ: Technological Feasibility

Ryan Demboski, Justin Eggan, Jack Gilliam

CS-476, Northern Arizona University

Michael Leverington

November 11, 2022

## Contents

The state of Arizona is currently one of the lowest ranking of the fifty continental states for recycling waste and waste management. According to a study performed by *LawnStarter,* Arizona currently sits at the 47th worst state for recycling - with the Ball Corporation citing that the state has an 18 percent recycling rate. This problem only contributes to the massive problem in the country and world when it comes to the handling of waste and, more importantly - recycling. While some Arizona municipalities have been able to invest millions of dollars into infrastructure to avoid filling new landfills, smaller municipalities cannot meet this.

This is where our project comes in. Our project is a joint effort with the Arizona Recycling Potential (ARP) team, who has talked to the Arizona Board of Regents about receiving a $1,600,000 grant for the overall project. This project will be created in four total phases over the course of 3 years. Phase 1 will be the collection of data for recyclable materials in Arizona, with phase 2 being the creation of the Arizona Recycling Potential model. Our project will be Phase 3 of the project, which will have us create a web-based visualization system for the Arizona Recycling Potential model and team. Supporting us in the development of Phase 3 is our project sponsor, Doctor Richard Rushforth. He is an assistant research professor in the School of Informatics, Computing, and Cyber Systems here at Northern Arizona University. His research focuses on big data modeling of food, energy, and water systems to further the understanding of complex, coupled natural-human systems.

Our vision of the Arizona Recycling Potential Visualization product is a web-based system that utilizes GIS (Geographic Information System) tools to allow Arizona Recycling Potential Model team users to upload data in order to see on-the-fly calculations and data overtop of a map. This map would be broken down into multiple levels, going as small as the county level and as far as an overview of the country as a whole. The visualization software is a web-based system, also allows people not affiliated with the Arizona Recycling Potential team to view the data. The goal of the visualization system is to aid in using Arizona Recycling Potential data by helping the team visualize their data in a way more convenient than a spreadsheet.

## Technical Challenges

The Arizona Recycling Potential project is one that has potential to be vital to cleaning up the state of Arizona, but it comes with its own challenges. We have broken the challenges down into a few separate categories; frontend, visualization, GIS tools, and backend. Each of these challenges required research into what would be the best solution.

## Frontend

We need a way to present our web application online. We need a tool that is able to communicate with the backend in order to access our data and display it in our online web application. Since we are primarily using Javascript, we need a tool that can take in formatted data and present it online.

There are several characteristics we are looking for with the front end tool we choose. One desired characteristic is that the tool must work efficiently and responsively. Our web application is going to be loading many different data sets, and we will need to be able to process and display these quickly. We also need a tool that can handle a dynamic interface. Having a tool that is able to be dynamic will allow us to easily move through the various segments of our web application seamlessly without having to open up many separate tabs. Another characteristic that we are looking for is that the tool we choose is well documented and can be easily implemented with the rest of the technology we choose to use.

Our client recommended that our team checks out the ‘MEAN’ stack for creating the web application. The ‘A’ in ‘MEAN’ stands for Angular. Angular is an open source JavaScript framework used for the front end development of web applications. It is managed by a team of Google developers. It is the most popular client-side framework for developing high performing web applications. Angular offers clean code development and many libraries which help create robust template solutions. Angular is part of the MEAN stack and is easily integrated with other tools we are using.

After researching Angular, we looked for potential alternatives in case there was a better option for us to use. The biggest competitor and definitely the other most popular option is React. React is a frontend JavaScript library that is used to build user interfaces from reusable UI components. React uses server-side rendering to provide flexible and performance-based solutions. React is straightforward and easy for developers to learn, making it a popular choice for developers to build fast and scalable applications.

Another popular alternative I found and researched was Vue. Vue is very beginner friendly yet offers many of the same features as React and Angular. The ease of use in turn, provides a difficult testing environment and makes it tough to fix errors in the code.

In order to compare the viability of each front end web application development tool, we did online research about when each option works best. We found through our research that when it comes to speed, all of the options are capable of being very quick and responsive, what it really comes down to is how the tool is used, and the code is structured. Research on implementation showed that the difficulty of implementing these tools depends on the other technologies in play. The dynamic characteristics of each alternative are all present for what we need to accomplish with them. Finally, online research showed that each alternative varied in its difficulty level for programmers with little to no experience using them.

| | Angular | React | Vue |
|---|---|---|---|
| Speed | 5 | 5 | 5 |
| Implementation | 5 | 3 | 3 |
| Dynamic | 5 | 5 | 5 |
| Ease of Use | 2 | 5 | 5 |

We have decided to choose Angular as the tool we are going to use when developing the front end of our web application. Despite ranking the lowest out of all other alternatives in the Ease of Use Category, Angular has many benefits we are looking for. Angular is most easily integrated with the other technologies we are going to use, making it a desirable choice for us. Because Angular is capable of being quick and dynamic, it will serve as a great way for our team to display our web application in a modern and professional way. In case we run into problems with Angular down the road, we have chosen React as a backup option, since it has all of the desired features we need, only it is more difficult to integrate than Angular is.

In order to prove that Angular is a viable tool for our group to use, we plan to implement some testing to ensure we are making the right choice. One test we are going to implement is creating a dynamic website that the user can navigate through quickly, easily, and without opening too many extra tabs. We will create a mock-up layout using Angular so we can test the navigation aspects of the web application. Another test we will implement is the ability to take in data from our backend technologies and send it to the graphing software we choose. This is a critical test to complete since it will show that all of our selected technologies work together.

## **Javascript Visualization**

We need a javascript library to create charts and graphs of our data. We need to visualize data in an appealing way that is easy to understand.

In order for the tool we choose to be the best option for us, there are some desirable characteristics we need to consider. The first characteristic we would like is that it is easy to implement into our stack. Since this tool will be used exclusively with the frontend of our application, the tool only really needs to integrate well with whatever tool we use for the structure of our frontend. Another characteristic we are looking for is that the tool we choose is able to create many different charts and graphs. This is important because we want to display our data in variable ways. Also, a characteristic we value immensely is the price of the tool we use. Since we are a group of college students and not a business, we are looking for something that is free to use. Finally, we want to use a tool that is popular so that documentation and other helpful resources can be found and easily accessed. Documentation is important because it will help us understand how to use the tool during development.

Chart.js is a free JavaScript library for making HTML-based charts. It is one of the simplest visualization libraries for JavaScript, and comes with many standard types of charts.

HighCharts.js is a paid JavaScript library that allows developers to create charts for web and mobile platforms. Since it is built on JavaScript and TypeScript, it can be easily implemented into whatever stack the developer is using.

D3.js is a free JavaScript library used for manipulating documents based on data. It has been regarded as a great way to create web based charts and graphs for a long time now and hence has wide popularity. However, due to its complexity and high level of customization, this tool is generally used when the user wants to create a new, never-before-seen type of graph. Other tools are more commonly used when it comes to creating standard graph types.

In order to choose one of these alternatives over the others, we did some online research into each of them, trying to compare them based on a few different criteria. The first was the price. Most tools researched were free, while one of the other tools we found required a subscription fee. Another criterion we looked for was the ease of use when it comes to implementing this tool with other technologies, particularly the ones we are going to use. This is important because we want to be able to seamlessly use this tool with the other tools we are already using. We also researched what each alternative was capable of creating when it came to graphs and charts. We found that some tools focused on just the basic, standard charts, while others took it a little further and allowed users to produce creative and innovative graphs.

| | Chart.js | HighCharts.js | D3 |
|---|---|---|---|
| Cost | 5 | 1 | 5 |
| Implementation | 5 | 5 | 3 |
| Customization | 3 | 4 | 5 |
| Documentation | 5 | 5 | 5 |

We have decided to choose Chart.js as the tool we are going to use in our Capstone project. Chart.js was chosen by us for several reasons. The first reason is that it is free and easy to use for beginners. Since none of us have any experience creating graphs, having a tool that will make it easy for us is a must. On top of that, the documentation online is plentiful and easy to read, so if we run into problems using Chart.js, we can easily figure it out. Even though other tools have more customization options, this tool will provide us with everything we need and can be easily implemented into the stack we are going to use.

In order to show that this tool is viable for us to use in our Capstone project, we will conduct a series of tests. The first test we will perform is to create each different type of graph that comes with Chart.js on the frontend of our web application. This will show us that everything is working soundly together and that the tools we have chosen do not have issues with each other. The next test we will perform is to create charts with some data that resembles the actual data our application will be using. This will show that we are able to convert the data into a neat web based visualization.

## GIS Tools

As our project is a visualization system that aims to generate a map with data on it, the most obvious hurdle is that of picking the correct GIS tool to use for our project. GIS tools are applications that allow the user to create maps of areas while also utilizing some kind of visual editor to allow the user to mark specific areas or add tags in the creation of said specific areas.

The desired characteristics of the GIS tool we will be using for the visualization system are simple. The first is that the tool must allow us to create maps at county, state, and ideally country levels. The second is that these maps must be able to have zoom functionalities and be able to have zones that can be selected. These zones should be able to house separate groups of data. For integration purposes, the ability to integrate with Javascript and the MapBOX tools are preferable. Finally, a low cost of a license is almost a must.

The ability to create maps at varying levels is the cornerstone of this project. Without the ability to create these maps, our visualization system cannot meet the requirements given to us by the Arizona Recycling Potential team and Doctor Richard Rushforth. As a sub-requirement of this main desired characteristic, these maps should have the basic functionality of a zoom feature - being able to zoom in from a state-level view to a county-level view is a characteristic that our visualization project needs.

The next desired characteristic is the zones housing groups of data. The data that will be provided to us will have different sets of data that relate to the Arizona Recycling Potential Model. This data is data such as recycling waste by area - such as by county or municipality, recycling rates by area, and so on. ~~H~~ousing this data in set zones allows the Arizona Recycling Potential team and any viewers of the visualization product to see the data for specific areas in a convenient area.

The final two desired characteristics are quite simple. As we are making a web-based visualization system rather than an application, we would like the tool to have some integration with javascript to ease development. It should also have integration with MapBOX, a tool that allows further editing of GIS maps. The final desired characteristic is the barrier to entry. As this product will be used further down the line and *not* solely by our Capstone team, we would like to keep license costs down to a minimum. Therefore, we prevent grant money from being used solely on licenses.

As we have many desired characteristics for our visualization system, we have been forced to look into multiple different GIS tools to use to create the visualization system. The initial GIS systems that we researched were ArcGIS, Maptitude, and QGIS. These three GIS systems were the ones that jumped out at us the most after research using our desired characteristics as search criteria. Each of them would be a good tool to use in the creation of our visualization system, but ~~t~~o keep development simple, we had to eliminate the tools down to one.

ArcGIS is the most obvious choice of GIS Tool for our project and was the first one that jumped out at us during the initial review of GIS Tools. The first reason for this is that one of our team members has had previous experience using ArcGIS at a professional level due to their work experience in the GIS field. Additionally, ArcGIS is one of the most used GIS tools at a professional level due to its high degree of documentation and vast library support for other tools and programming languages. The ability to create maps at the varying levels desired for our project is one of ArcGIS's main features. These maps have zoom functionalities integrated directly into them, and the zone tagging feature will allow us to house our groups of data. ArcGIS is able to integrate with MapBOX, and has a well-documented Javascript API available for developer use. The characteristic that ArcGIS fails to meet is the price - at a staggering $100 per basic student license, it breaks our desire for a cheap GIS tool.

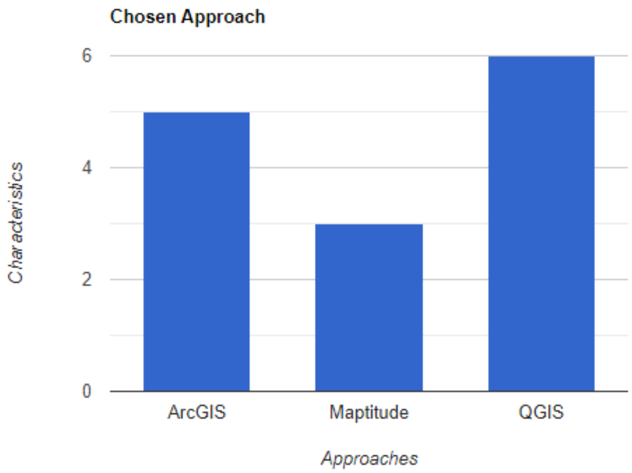
Maptitude is a GIS tool that came up in our research of GIS Tools. It's a tool that boasts having ready-to-use maps out of the box. The tool has built-in data search functionalities, a tagging and note feature that allows users to store data based on geographic locations, and a powerful Python 3 and C# library. However, these libraries are not part of our specifications - we are searching preferably for a Javascript library. Additionally, Maptitude does not integrate with MapBOX - instead is often referred to as a competitor or alternative to the program due to having most MapBOX tools integrated into it. While Maptitude is a powerful tool for development, its lack of MapBOX integration, javascript API, and high price tag - an astounding $695 per year per user - means that it fails some of our most desired specifications.

Finally, QGIS is a GIS tool that was recommended to use by our client, Doctor Richard Rushforth. He initially recommended the tool to us due to its power and, most importantly, the cost. QGIS is an open-source and free tool designed for public use without a license requirement.

The tool has an impressive library of tools, and APIs for a variety of programming languages - but most importantly, a JavaScript library. The tool allows for the creation of maps with zoom functionalities, and has a tagging system that allows us to store data. In addition, it has a well-documented integration with MapBOX that allows us to use both without the need for scrambling for documentation.

With each of these GIS Tools in mind, we had to analyze and evaluate the tools to come to a concrete conclusion for which to use for our Visualization System. For this purpose, the desired characteristics were weighed heaviest with all the possible tools we could choose. Then, we looked at the documentation for each tool - how feasible would it be for us to program using the tools while also allowing further teams using our code to adjust it? By weighing the desired characteristics for each tool, the approaches ordered themselves to QGIS by meeting all desired characteristics, ArcGIS for meeting 5 of the 6, and finally, Maptitude, which only met the creation characteristics but none of the other ones. Next, by weighing the documentation for the libraries and tools themselves, the approaches would be ordered as ArcGIS, QGIS, and Maptitude. The level of documentation for ArcGIS is staggering - every little tool available in the application is documented, with subsections for each. QGIS, as it is a community-driven open source program, has less documentation than ArcGIS, but its tools are documented on a main hub. Finally, Maptitude has *some* degrees of documentation, but at only a page of "official" documentation, it falls short compared to the rest.

Our chosen approach for this project will be using the QGIS application and its associated programming API. While the documentation of the ArcGIS programming



API is astounding and would certainly help future admins restructuring our visualization system, the fact that QGIS met all of our desired characteristics was the most important to us. The figure to the right shows how QGIS meets all six of our desired characteristics, ArcGIS meeting five, and Maptitude meeting a disheartening three of them. While each tool is powerful in its own right, QGIS meeting all six was the ultimate decision breaker for us.

In order to prove the feasibility of QGIS in our capstone project, we have some specific demos in mind for the Technical Prototypes demo. The first is obvious - making a map that can house data in specific areas. The second demo would be being able to pull data from a *file*, rather than hardcoding the data and putting that data in a tag. These two demos allow us to prove that QGIS would be feasible to use while not simply just finishing the product.

## Backend

Every web application that relies on displaying dynamic content and data to the user must have a backend system. A backend system connects a web server to a database system to dynamically add, delete, or view data on a database directly from the frontend interface of the web application. In our application specifically, the backend will be responsible for importing recyclable material datasets into the database, to then be displayed on the website through various frontend technologies and techniques. Eventually, it should be able to allow the user to upload new datasets directly from the frontend into the database to ensure the user never has to interact directly with the actual database itself. The site should be fully user operable from the frontend alone.

One characteristic of the backend that we would like to have is the ability to visualize the datasets of recyclable materials available in each community of Arizona. This is a use case that cannot be done without a database connected to a web server, because the data needs to be able to be dynamically inserted or deleted. It needs to be viewable from the frontend but also modified from the frontend in a way that it makes permanent changes to the server, instead of being changed locally on the client machine. With a database system in place we will be able to upload several datasets and have them display however we like, and we can permanently delete or make changes to the data from the frontend as needed. Each data change will be immediately viewable on every machine connected to the application.

However, the database is only one important component of the backend system we need for this project. Another characteristic we would like is the ability to run the web server via a technology that meshes well with our frontend languages of choice. Specifically, we want backend technologies that use the same language throughout the entire web stack. If we go with Javascript frameworks for our frontend and data visualization, then we will greatly benefit from using backend frameworks that also use javascript to keep things simple and much easier to maintain in the future, for new capstone teams that expand on this project further. A simple web stack makes development easier and increases the potential to optimize the application as much as possible.

A natural question to ask ourselves is if there are any alternatives to this approach of a web server, database, and javascript backend frameworks. Of course, there are many alternatives in terms of which specific technologies we use, but the concept of backend development stays the same regardless of which technologies we use. Every web application needs a server, a database, and a backend framework to connect all the pieces together. From this perspective, there is no real wiggle room to find an alternative approach. We could try to find different technologies or methods of running our server and database, but our planned web stack really does perform at its optimal level when the most popular technologies of this approach are integrated together to create a primarily javascript oriented visualization system. We find that our plan allows us to accomplish the project requirements, while having easy to maintain source code for future use.

To delve more specifically into our web stack of choice, let us introduce you to the MEAN stack. In this method of developing full stack web applications, the acronym MEAN stands for MongoDB, Express.js, Angular/React, and Node.js. Each of these technologies does something completely different and helps form our application into a whole. As far as the backend system is concerned, we only care about MongoDB, Express.JS, and Node.js. Notice the js as a common theme. This means these backend technologies utilize javascript during runtime, which is the same language that the frontend frameworks such as Angular and React are built from. This makes it very easy to connect all of these pieces together. Programmers who know Javascript can understand our entire source code with just a little practice of React syntax. To start from the top, Node.js is used to start up a server in a javascript runtime environment. It has support for HTTP so it can create web servers, which is what we will use it for. Next, Express.js is our backend javascript framework of choice, which handles the routing of our frontend and talks to the node server and our database. Speaking of databases, MongoDB is our database and the final piece of the backend system. It's a NoSQL database management system, which means it relies on JSON (Javascript Object Notation) objects to collect and store data, and return data back to the server, instead of using SQL queries to get data. With this approach we can continue to use javascript rather than introducing another language such as SQL in our source code, in order to keep the entire application consistent, maintainable and well optimized.

The MEAN stack is an incredibly popular and very common methodology of developing web applications in 2022. It just so happens that it meshes very well with our project requirements and we are very confident that we can deliver the exact product that is asked of us using these technologies. The efficiency in which these exact technologies communicate with each other in just one language means there is very little learning curve, small potential for issues, and keeps our application light on resources (which is a requirement for us as we are going live with this project on a small server that has a limited capacity for resources). It is also capable of handling dynamically changing datasets, such as user uploading/deleting data, or grabbing the data from a live server in the form of calling REST APIs through Express.js. How far or above and beyond we have time to achieve in this project will easily be able to be handled by our chosen technologies in this backend system. They are professional grade technologies used by millions of developers, and it's the perfect choice for our effort to make a web application that strives to visualize a greener Arizona.

## Conclusion

In conclusion, the Arizona Recycling Potential (ARP) team has entrusted us with creating a visualization system to aid in the use of the Arizona Recycling Potential Model. The Arizona Recycling Potential Model seeks to help clean the state of Arizona due to it being one of the lowest ranked states for recycling waste. While some parts of the state have started trying to alleviate this issue, this visualization system will be a web-based system that displays a map that houses the requisite data and displays it through varying means. Our job, as a capstone team, is to produce this visualization system and allow the Arizona Recycling Potential team to input their data into the system for further use in their project with the Arizona Board of Regents in order to help clean up the state of Arizona.

The frontend of our project will be a website that houses the information for the project and has the visualizer front-and-center. As the website needs to integrate the visualizer while also housing the data, the approach we have chosen for the frontend is using Angular, which has a lot of our desires but lacks proper ease of use.

For chart visualization, we decided upon using chart.js, a Javascript library that allows for the making of HTML-based charts. These tests will be done by making simple graphs and charts on our frontend. The map visualization itself will be built upon a GIS tool, which we have decided will be QGIS. This is because it meets all of our specifications and desired qualities, such as the ability to make an interactive map that can have marked zones that contain the information. The GIS tool's feasibility will be proven by simple tests that test the interactivity and data importing functions.

For the backend of our project, we have decided upon using the MEAN stack. This is an incredibly powerful and popular way of developing web applications, and meets the requirements we have set for our project. It is capable of handling all that we need it to, including dynamically changing datasets, such as user uploading/deleting data, or grabbing the data from a live server in the form of calling REST APIs through Express.js.

Overall, our project’s feasibility is at a point that we are confident in the development of Phase 3 of the Arizona Recycling Potential Team’s project. The next steps for our project include proving the feasibility of our project before starting proper development on the GreenAZ website. If we were to come to a hurdle during development, we’ve made sure to choose approaches that are already well documented so we can hopefully clear the hurdles fast and continue going forward without any major setbacks. We also chose these well documented approaches in order to prepare any future team that handles our code, since - if all goes according to plan - the GreenAZ Recycling Potential Visualization project will be used in years to come.