CS 476 - Team Badgers
Technological Feasibility

Northern Arizona University

October 2, 2020

Sponsor: Dave Hopfensperger & Glenn Austin

Team Mentor: Sambashiva Kethireddy

Members:

Abdulrahman Alamoudi

Tristan Marcus

Logan Ritter

Robel Tegegne

Yuanbo Xu

Overview

The purpose of this document is to outline our problem, express our tech challenges, and search for existing technologies that will help us complete the project. Additionally, show how we plan to implement these found technologies.

Table of Contents

1. Introduction

Since the beginning of March 2020, the world has been dealing with the SARS-Cov-2 pandemic, better known as Covid-19 or Coronavirus. The virus has affected many people's lives causing a massive change in work environments. Most people, even those who have not caught the disease themselves, are being impacted. Millions of people are out of jobs due to the inability to attend work because the business is not considered essential and the state government is requiring the business to be closed. There are not a lot of places that are capable of migrating their employees to a virtual workspace, but even those who can are stumbling upon a new set of issues.

The majority of companies that were able to operate virtually are finding that many employees are struggling with motivation, recognition, and many other things that they use to find daily in the office. Since virtual employees are now abruptly working on a laptop in their home “office”, they lack the social interaction that was constantly present in the workplace of old. In most cases productivity is down due to the lack of oversight, depression is up due to the lack of social interaction, and motivation is down due to the lack of recognition.

Team Badgers has been challenged by State Farm to develop a solution to the common issues of the virtual workplace. State Farm was one of the many companies that was able to transition their business to a virtual workplace. The one thing that they could not bring with them entirely is the social environment and the overflow of recognition that they had in the office.

Our solution for this web application is a gamified badging system where employees can obtain badges and points for a wide variety of skills and exceptional work. For example, if an employee does an outstanding job on a quarterly long project, his manager can reward him with a badge. Employees will also be able to obtain badges for other accomplishments such as, completing a training course, learning a new skill, reaching a certain year mark with the company, and much more. We want these badges to be plastered everywhere so coworkers can see the accomplishments of their peers.

For us to deliver State Farm the quality project they are expecting we must combine our individual skills and knowledge of technologies. There are a few things we need to decide on after exploring our options.

The purpose of this document is to compare our options and decide on a technology for each of the following.

1. Front End Framework - used to create and design the customer facing web application.
2. Back End Language - used to drive all the operations and processes of the application.
3. Database - used to store and retrieve necessary information.
4. Graphic Design - used to create the badges.
5. Version Control - used by the team to collectively work on the project and keep our files organized.

2. Technological Challenges

While developing this project, we anticipate that we will run into a variety of issues, problems and bugs. To avoid issues that may cause major setbacks, we list and describe a few of the major aspects of our project. If not discussed or decided prior to beginning our project, we believe these topics could be the source of many problems. Here we will discuss why each topic is important and how we plan to use each.

Front End Framework

Developing a web application requires a solid implementation of UI/UX. The front end framework will supply our team with a method to flesh out an elegant and responsive GUI alongside providing a streamlined experience for the user. In looking for a framework, we must determine whether it is suitable for our team during development and provide the tools/features necessary for our target audience.

Back-End Language

The back-end development is a necessary part of web application development. The back-end languages support the inner workings of the application as a whole. Choosing a powerful yet flexible and easy to use back-end stack will allow for a solid implementation of our project. As a team, we will choose one or more languages to suit our Skills / Knowledge Badging system.

Database

In this project the database would be required for the web application. This is required to give the user the best experience. The database is needed to be able to work in real time. The key highlights of the database for this web application are to allow easy interaction between the database and the users through the web application.

Graphic Design

For this project we will be responsible for designing and creating the physical badges themselves. We will need an application built for graphic design that will allow us to build these virtual badges. We do not need a whole lot so we just need a free application, that is easy to use and is very flexible. When we turn this project over to State Farm they will implement more of their own badges but we would like to at least provide them with a minimum of 20 basic badges so they at least have a theme to follow if they choose.

Version Control

Version control is one crucial part of software development. It is the place where our codes will be stored and managed. Our chosen version control should support issue tracking, continuous integration and is suitable for our team based on cost and limit of contributors for a repository. Version control with an easy interface, free and easier to learn/use is our primary choice.

3.Technological Analysis

Developing a badging system for State Farm will require Team Badgers to analyze the proposed technological issues at hand. Our team has determined these critical issues to be choosing a proper front-end framework, back-end language, database, graphic creation software and version control system to support our development. For each issue, sufficient alternatives will be compared based on desired characteristics. Upon deciding on an alternative, feasibility for this chosen technology will be proven based on the requirements specific to our badging system.

3.1 Front-End Framework

Introduction

This project will be developed as a web application. A necessary component will be a front end framework that uses JavaScript. The requirement for this application is that it remains a single page application (SPA). This requirement is the main reason React, Angular, and VueJS were chosen for research.

Desired Characteristics

The desired characteristics of our chosen framework is its ease of use, followed by functionality, followed by performance. Our project will be focusing on quick and decisive implementation, therefore ease of use is a high priority. Functionality plays a role in our project but most modern frameworks have features that support our specific projects requirements. Performance will be a tie breaker considering this is a single page web text and image based web app, not large enterprise software. The characteristics for each framework will be weighed and used as a metric for our decision.

Alternatives

The front end framework options that are most suitable for this project are React, Angular, and VueJS. These are the most popular frameworks used this year for all web applications and there is ample reason behind that. Each one of these frameworks happen to provide all the tools necessary for a successful implementation of a web application. For our specific project, a thorough examination of these frameworks is required in order to best meet our specific needs.

Analysis

React

Developed by Facebook, React is a JavaScript library rather than a framework, but has all the necessary tools to be used as one. This library makes use of JavaScript combined with HTML logic. It is easy to learn and is the most minimal in design compared to Angular and VueJS. Performance compared to the other frameworks shows that React is the fastest due to rendering optimization tools.

Ease of use: Medium

Functionality: Minimal

Performance: Fast

Angular

Developed by Google, Angular is a Javascript framework with a large amount of features using TypeScript. It is the more complex framework compared to React and VueJs, but proves to be a more extensive option. Performance compared to the other frameworks tends to be on the slower side but isn't inherently slow by nature.

Ease of use: Complex

Functionality: Extensive

Performance: Relatively Slow

VueJS

Developed as an open source project, this community driven framework offers a medium amount of built in features that uses regular JavaScript. This framework is the easiest to learn but is constantly being updated due to it being open source. Performance compared to the other frameworks shows Vue to be faster than Angular, but not as fast as React.

Ease of use: Simple

Functionality: Medium

Performance: Medium

Decision

After reviewing the framework options, Team Badgers has decided to choose React. Looking at the ease of use for all frameworks, VueJS leads, but React is not far behind. Most team members have experience with React so ease of use plays less of a factor. As for functionality, React is behind Angular, but sufficient enough for our development purposes. Performance shows that React ranks higher than the other frameworks due to its internal optimization tools. Looking at total scores, React leads slightly ahead of VueJS, leading us to default to React due to past experience.

Option	**React**	**Angular**	**VueJS**
Ease of use	5	3	5
Functionality	4	5	4
Performance	5	3	4

Proving Feasibility

In relation to our project, React as a framework is perfect for several reasons. Its component based structure will allow us to create a single page application with little struggle. Additionally, React offers a wide enough variety of tools to support our functionality as an internal social media. Integration of badges on the front end will be simple and modular due to the embedded tools in React's components. Also, performance optimization tools are already implemented into React's libraries allowing for more development time rather than optimization. React also natively supports NodeJS giving our team an advantage in integrating our project's front and back end.

3.2 Back-End Language

Introduction

This project will be developed as a web application. The back-end development needs appropriate languages. Choosing an appropriate back-end language will make the developing process much easier and the product more perfect. For this reason, we will compare several popular back-end languages to each other and choose the most suitable one as our back-end development language.

Desired Characteristics

The desired characteristics of our chosen language is Readability followed by ease of use, followed by performance. Readability will consider whether the syntax of code is strict and code is close to the languages we use in our life. Almost any language options we have chosen will get our job done. So the readability of the language becomes the most important factor in analysis as it is a team project and we need to be able to understand each other's work without requiring a long explanation. The ease of use is the second important factor because our project will be focusing on quick and decisive implementation. Functionality plays a role in our back end but as mentioned before, any language options we choose will get our job done, so it becomes the last to consider.

Alternatives

We select python, Java, PHP and NodeJS for analysis. These are popular back-end languages used this year for all web applications. All of these back-end languages would work but choosing an appropriate one based on our specific needs will make our product more perfect.

Analysis

Python

Popular among the scientific community, Python is an open source, and high-level programming language which boasts of code readability, concise code and comprehensive standard library. It also features a dynamic type system and automatic memory management. Python takes less time than Java to develop due to its high-level data types and its dynamic typing programs.

Ease of use: Medium

Functionality: Extensive

Readability: High

Java

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. The long commercial life and wide adoption of Java has created a robust ecosystem of documentation, libraries and frameworks many of which are aimed at e-commerce, security, and complex transactional architectures. Java programs are expected to run faster than python programs.

Ease of use: Complex

Functionality: Extensive

Readability: Medium

PHP

PHP is a general-purpose programming language with a vast ecosystem of developers, frameworks, and libraries. PHP doesn't have rules like compiled languages or strict standards as seen with Python, but rather guidelines available from the developer community. Due to ease of programming and popularity, many big players are using PHP. Although PHP loses to python for both readability and maintainability. Python supports structured exception handling while most PHP functions do not use exceptions for reporting errors.

Ease of use: Simple

Functionality: Medium

Readability: low

NodeJS

Node.js is an open-source, cross-platform, back-end, JavaScript runtime environment that executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to produce dynamic web page content server-side before the page is sent to the user's web browser.

Ease of use: Medium

Functionality: Extensive

Readability: Medium

Decision

After reviewing the framework options, Team Badgers has decided to choose python and NodeJs as our back-end languages. Python is very functional and easy to use, it is very suitable for quick deployment. NodeJs uses Javascript in the back end, which can maintain consistency with the front end. It is also easy to use. Another important reason for choosing python and Node js is their readability since it is a key factor to consider when working as a team.

Option	**Python**	**Java**	**PHP**	**NodeJS**
Ease of use	4	2	5	4
Functionality	5	5	3	4
Readability	5	3	2	4

Proving Feasibility

For all the reasons mentioned above, we choose NodeJS since it can handle the concurrent request handling efficiently better than others including Python. If our badging system receives many requests at the same time, the incoming requests will get lined up and are executed quickly and systematically. The Node.js is also known to be highly extensible, which means that we can customize and further extend Node.js as per their requirements. Although JavaScript does not have a well equipped and robust library system in comparison to other programming languages, we could solve this problem by selecting python as a supplement since python has lots of libraries. Also all the members of our team are familiar with python and Javascript.

3.3 Database

Introduction

The system of our web application requests a database. The issue would fundamentally be the database communicates between the data set and the users through a web application In the real-time data sync. A significant test related to this will be running constant questions to this information base when performing searches for information, users, groups and etc.

Desired Characteristics

The desired characteristics or the ideal qualities we need in a database are easily accessible, easy to use and syncing the real-time data. Also, providing all the tools that are needed for this application. Also, it needs to support the framework that we will use. Some of these requirements are available in a lot of databases. However, that does not mean any database tool can be the best fit for our web application. This is because we have specific requirements to create a web application page that is needed to fit with the database for our project. Some providers that we need in the database are supposed to provide authentication analytics to manage real time data, push notifications file storage and web hosting. Moreover, it needs to work with the framework we will use in this project.

Alternatives

Our choices for databases in this project are Firebase, Docker and MongoDB. This is because these are the most popular databases used for developing web applications in the last five years. Each one of these databases provides the necessary tools to develop a successful web application in which our team will analyse. Some of the providers we want are easy-to-use, ready-made API that is easy to synchronize across devices.

Analysis

Firebase

The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync data between web application users in real time. Cloud Firestore enables the storage, sync, and query app data at global scale. Firebase is a Backend-as-a-Service — BaaS — that started as a YC11 startup and grew up into a next-generation app development platform on Google Cloud Platform. Firebase is your server, your API and your datastore, all written so generically that you can modify it to suit most needs.

Ease of use: Easy

Accessibility: Simple

Real time Sync: Fast

Docker

Docker is extraordinary for running information bases in an improving climate. It can be used for databases of small, non-basic undertakings which run on a solitary worker. It's an incredible device, however programmers presumably do not need to bother with the upsides it can give for this situation.

Ease of use: Medium

Accessibility: Simple

Real time Sync: Fast

MongoDB

MongoDB is an article situated, straightforward, dynamic, and versatile NoSQL information base. It depends on the NoSQL report store model. The information objects are put away as independent reports inside an assortment — rather than putting away the information into the segments and lines of a customary social data set.

Ease of use: Medium

Accessibility: Medium

Real time Sync: Fast

Decision

After reviewing the database options, Team Badgers has decided to choose Firebase as our best fit for this project. This is because it's the easiest and the most accessible compared to the other options.

Option	**Firebase**	**Docker**	**MongoDB**
Ease of use	4	3	2
Accessibility	4	3	3
Real time Sync	4	4	4

Proving Feasibility

For all the reasons mentioned above we believe Firebase is the perfect database we could use for this web application. This is because Firebase is easier to use than both Docker and MongoDB. Firebase is more understandable and more clear to find what you are looking for than the other database tools. Firebase provides authentication analytics to manage real time data, push notifications file storage and web hosting. Moreover, Firebase has service functions that can call with auth context from the front-end, so it can easily tell the back-end code if a user logged in.

Another reason to choose Firebase is that it has a background conductivity database service called cloud fire store that captures all the changes in the database. We can use it to upload and import files through cloud storage. Even if the network link or connection is poor, we would be able to retry the process right where it was left off, saving time and bandwidth for us. Also, almost all the members of the group are familiar with Firebase more than the other two database tools.

3.4 Graphic Design

Introduction

After meeting with our sponsor from State Farm, we learned that designing the badges was one of the requirements. Our team has little to no experience at all with graphic design which leads this to becoming a challenge. We would like to provide State Farm with 20+ badges at the very minimum. We may also use said graphic design application to create logos or images, such as our team logo to implement on the single page application itself. In order to give these badges a good look, we are going to need a solid graphic design program or technology. All we need is a simple application that will allow us to create images and icons without too many problems. We are willing to learn new software to accomplish these tasks but we are hoping to find an application that does not require too much of a learning curve to be successful.

Desired Characteristics

The key features we want in a graphic design application are ease of use, flexibility, and free or cheap. We need the application to be easy to use as none of us have any significant experience with graphic design, yet we still want to deliver quality badges to our clients. Next, we need the application to be flexible, in other terms it needs to allow us to do what we want to the badges without having to go around a ton of roadblocks (for example, editing a badge in PowerPoint would not be ideal because there are limitations to what you can do to an image). Lastly, we need the application to be free or very cheap.

Alternatives

After some research, We have decided that some of the more suitable graphic design options that we found are Adobe Photoshop, Gimp, and Inkscape. These are some of the top graphic design applications out there that won't require a large amount of expenses. All of these graphic design applications would most likely get the job done but a more in depth comparison between them will be conducted below to see what application fits our teams needs the best.

Analysis

Adobe Photoshop

Developed by Adobe, Photoshop is the number one most recognized and used graphic design application. It is very easy to use and due to its popularity there are thousands of tutorials and videos on how to use it. Although our team does not have a lot of experience with graphic design, all of our experience comes in photoshop. Some pros with photoshop are, it has the best user interface compared to our other options, it is also very flexible and easy to use. The largest con with photoshop is its price coming in at $20.99 a month.

Ease of use: Very easy

Flexibility: High

Cost: $20.99/month or $4.20/month per person on the team.

Gimp

Developed as an open source project, Gimp is one of the most popular free to use graphic design applications. Some pros of gimp included its ease of use, its flexibility and built in tools, and most importantly, it's free. Some cons of gimp include, lack of updates due to it being an open source project and the user interface is not the best.

Ease of use: Easy and comes with tutorials.

Flexibility: High

Cost: FREE

Inkscape

Developed as another open source project, Inkscape is a popular free to use graphic design application. Unlike Gimp, Inkscape receives regular updates by a designated team. It also has a great user interface much like photoshop making it very easy to use. Since it is open source, users are able to change the code to make it work better for them. Some cons of Inkscape are that it has a few bugs when it comes to editing text. It also does not work very well on a Mac OS system which may be an issue for some team members. Lastly it has been reported that it is a bit slower than other applications.

Ease of use: Very Easy

Flexibility: Very High

Cost: FREE

Decision

After reviewing the graphic decision options, Team Badgers has decided to choose Gimp. If price were not a factor we would have most likely gone with photoshop. Since we do not have a budget, we have gone with one of our free options. We chose gimp over Inkscape due to its interface being more familiar to some team members as well as its performance being great.

Option	**Photoshop**	**Gimp**	**Inkscape**
Ease of use	5	5	4
Flexibility	4	4	3
Cost	1	5	4

Proving Feasibility

For all the reasons mentioned above we believe gimp will allow us to successfully create badges to deliver to State Farm. Gimp will allow us the freedom and flexibility to design these badges with ease and since it is a free application the team will not have to spend any money from their own pocket. With the easy to learn and easy to navigate user interface, we will use Gimp to create 20+ badges all around the same State Farm theme. Since it is also open source we will be able to modify any code to better fit our needs throughout the designing of these badges. We may also use Gimp for other designs across our front end single page application such as logos, sprites, images, and more. Using this application we will be able to save a large amount of time by bypassing the learning curve of some other extensive graphic design applications while still being able to come up with quality images, badges, and logos.

3.5 Version Control

Introduction

Version control allows you to manage changes of files, or code over time. Version control helps members of a team to be up-to-date with the recent version of code. It also keeps track of changes made to a file or code with more detailed information such as by who, when and to which code. With that being said, choosing a definitive Version control application is in our team's best interest.

Desired Characteristics

The key features desired for our project are cost, flexibility and ease of use including concurrent development, tracking changes, recovery, and team collaboration. We are looking for alternatives that can do the above points with less or free of charge. We will also take into consideration how easy to learn the version control is. Ease of deployment will also play a factor. Comparing all the alternative version control will help us to pick the best version control for our team. The characteristics for each version control will be weighed and used as a metric for our decision making.

Alternatives

After some research, We have decided that some of the most suitable version control options that we came up for are Github, BitBucket and GitLab. These are the most widely used version controls available at relatively less or free charge. Three of the version controls mentioned above can accomplish our desired job. However, we will take some desired characteristics to weigh them and decide the best that fits our project specification and requirement.

Analysis

GitHub

Github is the most widely used version control with over 200 supported languages. It provides hosting for software development and version control using Git. It also offers distributed source code management. Its core features include access control, bug tracking, feature requests, task management, continuous integration and wikis for every project. Github also focussed more on open source projects and public code. Github pages is a feature that makes it easy to publish and host web pages within Github. Additionally, it is easier to integrate with third party APIs. Syntax highlighting is offered in Github which BitBucket lacks. It also offers unlimited private and public repositories.

Ease of use: Medium

Flexibility: High

Cost: Free

BitBucket

BitBucket is a web-based version control repository hosting service that supports both Git and Mercurial. BitBuckets focus is on enterprise software and users who prefer a private repository over public or open-source. It is also easier to integrate Trello for tracking purposes.

Ease of use: Medium

Flexibility: Low

Cost: Free

GitLab

GitLab is a web-based DevOps lifecycle tool that uses Git-repository manager. It provides issue-tracking, continuous integration, wiki, and development pipeline features. It also provides unlimited private repositories like Github and BitBucket. However, its main focus is DevOps.

Ease of use: Medium

Flexibility: Low

Cost: Free

Decision

After reviewing the version control options, Team Badgers has decided to choose Github. Github is free plus the storage cap is higher than both BitBucket and GitLab. Besides that, we chose Github because it offers Github pages where we can deploy and host our project easier. Additionally, most of our team members are familiar with Github and its interface.

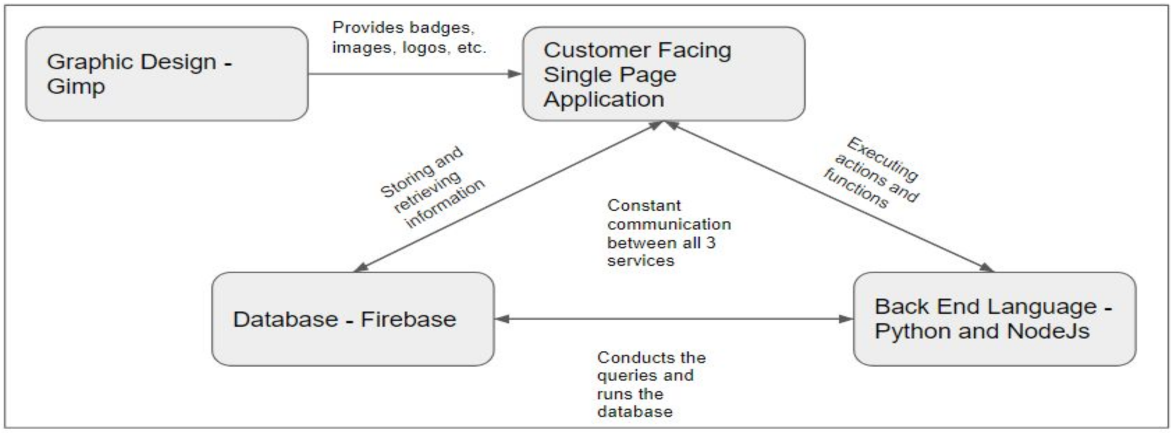
Option	**Github**	**BitBucket**	**GitLab**
Ease of use	4	3	3
Flexibility	5	4	4
Cost	5	5	5

Proving Feasibility

For all the reasons mentioned above we believe Github will allow us to successfully manage our code. Github's easier deployment methods are a large factor to choose it. It is also a widely used version control with the biggest community. Github's desktop applications also makes it easier to use and maintain our project. It is also the most preferred and used among our team members for its simplicity.

4. Technological Integration

Now that we have discussed all the foreseeable issues we may have and we have made the software decisions for this project, we must bring it all together. Our clients, State Farm, have asked us to create this project as a single page application. We will be using React to build a customer facing single page application. This front end page will have to work alongside with the backend languages, Python and NodeJS, and the database, Firebase, to perform the necessary actions. We will be using Python to write any actions or functions on our application. The back end language will also speak to the database to store and retrieve any information a user could need. We will need to store things such as, login information, user's current badges, etc. We will need a location to store all our progress and we decided to go with GitHub as we are all very familiar with its functionality. Lastly, we will use Gimp to create numerous graphics for our application. The number one purpose for Gimp will be to design the badges that will be given to users. We may also need to use Gimp to create any visuals that we may need on our front facing application. We must use Gimp instead of finding images because like many other companies, our client does not want to have to deal with any trademark, or copyright issues. With all these technologies that we have chosen, we believe we are well equipped to design and build our single page application and deliver the product our sponsors are expecting.


Graphic Design - Gimp
Provides badges, images, logos, etc.
Customer Facing Single Page Application
Storing and retrieving information
Executing actions and functions
Constant communication between all 3 services
Database - Firebase
Back End Language - Python and NodeJs
Conducts the queries and runs the database

All held in our version control - GitHub

5. Conclusion

We aim to build a secure internal web application where employees can easily recognize the skills and achievements of others while proudly displaying their honors or awards. When building a web application such as ours, there are many problems that come along in the development process. Among these problems are making sure our front-end and back-end are sufficiently integrated, making sure we have a database to store our information, having flexible and powerful version control, and finally being able to design badges using an easy to use graphical creation software.

The technical objectives covered in this document include the analysis of several web technologies leading our team to decide on which is suitable for implementation. Our front-end framework will be React due to its ease of use and our team's prior experience. This gives our badging system the needed flexibility and modularity of UI/UX due to Reacts component based architecture. Coupled with that are our back-end languages being NodeJS and Python (supplementary). These languages correspond to State Farm's back end stack allowing for a smooth transition when State Farm picks up the software. Our database of choice will be Firebase considering its speed and simplicity advantages. As for the badges themselves, we need to design some styles of badges, and this work will be done by using Gimp. Also our project will have a complete professionally-documented codebase, we plan to use Github as our version control and repository.

As a team, we will be moving on to prototyping our Skills/Knowledge Badging web application using our well researched and hand picked software stack. After prototyping and fleshing out a proper basis for development, we will begin implementation. Our main goal in development is to have extensive abilities when it comes to giving and receiving badges, but we would also like to consider modularity as a whole. The project will be implemented in a manner for ease of change and adaptation. As a team we believe this will lead to State Farm having an easy to use and upgradable product.

	Choice	Feasibility	Confidence (1-10)
Front-End Framework	React	Easy to use, high functionality and performance	9
Back-End Language	Python with NodeJS	Easy to use, highly readable and functional	9
Database	Firebase	Real time sync, easy to use and access	7
Graphic Design	Gimp	Free, easy to use, highly flexible	8
Version Control	GitHub	Free, flexible, widely used.	9

6. References

Front-End Framework:

1. https://www.codeinwp.com/blog/angular-vs-vue-vs-react/
2. https://medium.com/techmagic/reactjs-vs-angular5-vs-vue-js-what-to-choose-in-2018-b91e028fa91d
3. https://vuejs.org/v2/guide/comparison.html

Back-End Language:

1. https://medium.com/@MarutiTech/how-to-choose-the-perfect-back-end-technology-5674db9c0192
2. https://www.activestate.com/blog/python-vs-java-duck-typing-parsing-whitespace-and-other-cool-differences/
3. https://en.wikipedia.org/wiki/Node.js

Database:

1. https://firebase.google.com/products/realtime-database
2. https://docs.docker.com/get-started/overview/
3. https://www.mongodb.com/why-use-mongodb

Graphic design:

1. https://helpx.adobe.com/creative-cloud/how-to/photoshop-illustrator-indesign-overview.html
2. https://www.adamenfroy.com/best-graphic-design-software
3. Photoshop vs. Illustrator vs. InDesign | Adobe Creative Cloud tutorials

Version Control:

1. https://www.upguard.com/blog/bitbucket-vs-github
2. https://about.gitlab.com/pricing/
3. https://www.elegantthemes.com/blog/wordpress/github-vs-bitbucket#:~:text=If%20you%20boil%20it%20down%20to%20the%20most,tends%20to%20have%20mostly%20enterprise%20and%20business%20users