Technology Specification: Exogredient

Team A

Jason Nguyen (Team Leader) - 014232658
Eleazar Gomez - 012465490
David Nguyen - 015633187
Abe Malla - 016069649
Charles Kwak - 015368728
Hovhannes Babuchyan - 017228651

November 12, 2019

Table of Contents

Summary

Development

Operating System	Windows
Backend IDE	Visual Studio Community
Frontend IDE	Visual Studio Code
Messaging Platform	Discord
Programming Languages	C#, JavaScript
Development Tools	Live Server (VSCode Extension)
Version Control	Git with Github

Technology

Target Browser	Google Chrome, FireFox, Safari
Backend Server	ASP.Net Core
Frontend Framework	Vue.js
Web Server	Microsoft IIS
Database	MySQL Community Server 8 & MySQL Workbench 8
Database CMS	XtraDB
Reverse Proxy	Nginx Reverse Proxy

APIs

Messaging & Phone API	Twilio
Computer Vision API	Google Cloud Vision API
Web Mapping API	Google Map API
Places Information API	Google Places API

Deployment

Cloud Provider	Google Cloud Services
Certificate Authority	LetsEncrypt
DNS Provider	NameCheap

Purpose

Analyze and layout the technologies that developers will use, as well as the versioning rules for the mentioned technology. This document also specifies the development and deployment environment.

Rules For Versioning

For libraries we will use LTS. Update the libraries when LTS is updated and it's a minor update. This will be performed every 3 months. Major updates will be performed every 6 months. Cascading updates will be ignored until further discussion with developers.

Development

Operating System

Top Options For Operating System
Value: The Operating System is an essential part of our system because it determines the tool sets and environments available to us. We want to choose one that the team members are all familiar with to reduce the time learning and researching.
Windows 10 **Pros**: 1. Provides access to IIS, unlike alternatives. 2. Provides the largest collection of products. **Cons**: 1. Windows CMD lacks in comparison to the Unix based CLIs of Linux and MacOS. 2. Tends to have more security issues. 3. Performance starts to lack when projects grow to a large size. **References**: 1. https://blog.soshace.com/en/programming-en/operating-systems-showdown-windows-vs-macos-vs-linux-for-web-development/ Linux **Pros**:

1. Offers the most customizable environment when compared to the other two.
2. Unix based command line is comparably better than Windows CMD or Powershell.
3. Provides similar environment to web servers since most of them run Linux.

Cons:

1. General lack of support.
2. Steep learning curve for those who have not used it.
3. Lacks popular software

References:

1. https://blog.soshace.com/en/programming-en/operating-systems-showdown-windows-vs-macos-vs-linux-for-web-development/

MacOS

Pros:

1. Able to run both Linux and Windows in VMs.
2. Unix based command line is comparably better than Windows CMD or Powershell.
3. Offers OS exclusive tools that are nice for front-end work.

Cons:

1. Only runs on Apple Computers.

References:

1. https://blog.soshace.com/en/programming-en/operating-systems-showdown-windows-vs-macos-vs-linux-for-web-development/

Chosen Technology: **Windows 10**

Windows is the has been the most used Operating System across the world for a decent amount of time now, so it's no surprise that the entire development team is firmly comfortable with using it. Using technologies familiar to the team will help cut down the time spent on configuration and dealing with non-project specific issues. Since Windows offers the largest amounts of tools already used by the team, it should help with maintaining focus on developing the actual product.

Minimum Hardware Specifications

- 8GB RAM
- 2 GHz CPU with 2 or more cores
- 100 GB of Hard drive space.

The minimum hardware requirements for all the tools that we are planning to use were gathered and the most demanding requirements of those tools determined our minimum

specifications for our development machines. The largest minimum requirement for RAM is held by mySQL, which requires 4 GB of ram. To avoid memory problems, we chose 8 GB of RAM as our minimum requirement. A majority of the tools that are planned to be used only have a minimum requirement of a cpu clocked at <2 Ghz. As for storage, 100 GB was chosen to be the minimum after adding all the storage space needed for the tools to be used. The space needed for additional features that may come with some tools was also included in our decision for minimum disk space.

IDE (C# / Backend)

Top Options For Backend IDE
Value: We want to choose an IDE that comes with a large amount of toolsets and features. We are prioritizing a more extensive development environment over a lightweight one.
Visual Studio Community 2019 **Pros:** 1. Contains IntelliSense, a code completion and code suggestion feature. 2. Developed by Microsoft who also developed the .NET framework. Therefore The .NET framework is well supported for development. 3. Developers have access to many extensions that allows the IDE to have additional functionality. 4. Includes debugger for the C# language. 5. Large amount of documentation. **Cons:** 1. Only available for Windows and Mac computers 2. Can be more resource hungry than other IDEs. 3. Takes a bit long to start the application. **References:** 1. https://visualstudio.microsoft.com/vs/features/net-development/ 2. https://www.jetbrains.com/rider/compare/rider-vs-visual-studio/
JetBrains Rider (V2019.2) **Pros:** 1. Supports the .NET framework. 2. Is Cross-platform (available for Windows, Mac, and Linux) 3. More integrated support for other VCS (Mercurial, SVN, and Perforce) **Cons:** 1. No community edition (costs money, but there is a 30 day free trial). 2. Not as mature as Visual Studio. May have more bugs.

References:

1. https://www.jetbrains.com/rider/compare/rider-vs-visual-studio/

MonoDevelop (V7.6.9.22)

Pros:

1. Lightweight. Has much faster startup time compared with Visual Studio.
2. Available across platforms.
3. Free.

Cons:

1. Does not have a mature extension functionality as Visual Studio.
2. Has less features than both IDEs mentioned above.

References:

1. https://www.monodevelop.com/documentation/feature-list/

Chosen Technology: **Visual Studio Community 2019**

We chose Visual Studio mainly because it was developed by Microsoft. Since we are developing using the .NET framework we can trust there is a high amount of support for the framework.

IDE/Editor (HTML, JAVASCRIPT / Frontend)

Top Options For Frontend Editor
Value: we are looking for a lightweight editor with a large library of extension support.

Visual Studio Code V1.39

Pros:

1. Has extension feature. Can install plugins to add new features.
2. Free.
3. Available across platform.
4. Contains IntelliSense which allows for code completion and code suggestions.
5. Lightweight IDE and not resource hungry.
6. Git integration built-in.

Cons:

1. Can have a slower launch time that other editors.
2. Based on electron, which can cause performance issues sometimes.

References:

1. https://www.slant.co/options/5982/~visual-studio-code-review

---

Atom version V1.41.0

Pros:

1. Has extension feature. Can install packages to add new features.
2. Free.
3. Available across platform.
4. Git integration built-in.
5. Simple UI / friendly for beginners.

Cons:

1. Has less extensions / packages compared to Visual Studio Code.
2. Not as fast as Visual Studio Code.

References:

1. https://atom.io/packages
2. https://marketplace.visualstudio.com/search?target=VSCode&category=All%20categories&sortBy=Installs
3. https://www.makeuseof.com/tag/visual-studio-code-vs-atom/

Notepad++ V7.8.1

Pros:

1. Very lightweight and fast. Fastest among other IDE/editors mentioned above.
2. Smallest install size among other IDE/editors mentioned.
3. Simple UI. Very beginner friendly.

Cons:

1. Only available on Windows computers.
2. Very basic compared to other mentioned IDE/editors.

References:

1. https://www.capterra.com/p/185965/Notepad/
2. https://en.wikipedia.org/wiki/Notepad%2B%2B

Chosen Technology: **Visual Studio Code**

The main reason for choosing Visual Studio Code is because we determined that it has the best balance in terms of speed and features compared to the other mentioned editors. It also has one of the largest communities so we have access to more support and extensions compared to other tools.

Browser

Top Options For Web Browser

Value: We want a browser that is popular, simple, and fast. We would also prefer a browser that would not crash entirely if a tab crashes. It has to be safe and secure and have cross platform features.

Google Chrome (latest version)

Pros:

1. Multiprocess Architecture: Every tab, window, and plug-in runs in its own environment.
    a. Entire browser won't crash because of a tab.
    b. Fast.
2. Safe and Secure.
    a. Automatically update when update and internet connection is available.
    b. "Safe Browsing" option to block dangerous/malicious website.
    c. Cleanup tool for Windows to clean some malware.
    d. Lock icon displayed left side of url to show secureness of website.
3. Powerful Developer Console.
    a. Can monitor network, view source code, and choose colors to adjust the design online.

Cons:

1. Privacy Concerns with Google Tracking.
2. Limited Customization and Options.

References:

1. https://www.pcworld.com/article/150585/google_chrome.html
2. https://www.webnots.com/advantages-and-disadvantages-of-google-chrome/

Safari (latest version)

Pros:

1. Page-load times is impressive.
2. Automatically embedded security features.
3. Automatically cleans up unnecessary junk.

Cons:

1. Lack of customization.
2. Difficulty of deleting cookies on exit automatically.
3. Updates are rare.
4. Some websites do not support Safari.
5. Although pop-ups are rare, it is hard to turn off pop-ups.

References:

1. https://www.online-sciences.com/computer/the-advantages-and-disadvantages-of-safari-web-browser/

Mozilla Firefox (latest version)

Pros:

1. Automatically embedded security features.
    a. Counter threats like spyware, malware, and some viruses from visiting a site or downloads.
    b. Prevent some pop-ups.
2. User experience similar to Chrome and Safari.
    a. Tabbed browsing.
    b. Restoration of tabs on reboot if desired.

References:

1. https://brandongaille.com/8-pros-and-cons-of-firefox/
2. https://www.pcmag.com/review/241850/mozilla-firefox

Values

	User Base	High Memory Usage	Clean and Simple (easy to use)	Security
Google Chrome	44.5%	Yes	Yes	Yes
Safari	25.4%	No	Yes	Yes
Mozilla Firefox	7.4%	Yes	Yes	Yes

Continued

	Cross-platform	Speed
Google Chrome	Good platform features	Fast
Safari	Provides features but compatibility issues	Fast but slow at times (compatibility)
Mozilla Firefox	Provides features but compatibility issues	Fast but slow at times (compatibility)

References:

1. https://www.zdnet.com/article/chrome-is-the-most-popular-web-browser-of-all/
2. https://cloudacademy.com/blog/google-vision-vs-amazon-rekognition/
3. https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/home
4. https://brandongaille.com/8-pros-and-cons-of-firefox/
5. https://www.pcmag.com/review/241850/mozilla-firefox
6. https://www.online-sciences.com/computer/the-advantages-and-disadvantages-of-safari-web-browser/

7. https://www.pcworld.com/article/150585/google_chrome.html
8. https://www.webnots.com/advantages-and-disadvantages-of-google-chrome/

Chosen Technology: **Google Chrome**

Most of the top options fit our desire for a browser, however we chose Google Chrome. Chrome's multi-process architecture means it is less prone to crashing due to a single tab or website. It covers the safe and secure part with the tools listed above. It is fast all around and also provides a powerful developer console. There are a few cons associated with it, like the limited ability for customization, high memory/CPU usage, and privacy concerns with Google tracking. These cons negatively affect the user experiences but there is still a large user base for this application. Chrome provides a fast and consistent speed on various platforms. On the other hand, some websites are not supported by Safari. Chrome also auto updates when there is an internet connection, while Safari rarely has updates. Firefox is also fast but it has compatibility issues with some platform, and it also uses a lot of memory. After comparing the pros and cons for each browser, we will go with Chrome because it most aligns with our goal.

Messaging Platform

Messaging Platform
Value: We want to team chat tool that offers the most relevant features for a free plan.
Discord **Pros:** 1. Has free voice channels. 2. Message history is not limited on free plan. 3. Free screen sharing. 4. Has push to talk feature.

Cons:

1. Can monitor other applications. Privacy concerns.
2. Does not natively have feature to record calls.

References:

1. https://slack.com/pricing
2. https://zapier.com/blog/slack-vs-discord/
3. https://www.guidingtech.com/skype-vs-discord-comparison/
4. https://support.discordapp.com/hc/en-us/articles/223657667-Group-Chat-and-Calls

---

Skype

Pros:

1. Oldest of all the messaging platform. Because it is more mature it may be more reliable.
2. Skype will let users upload larger files compared to Discord.
3. Has feature to record calls.
4. Has higher group call members than Discord.
5. Has integration with Microsoft Office Suite.

Cons:

1. Lacking integration on other apps (Google drive).
2. Lacking in server chat features other platforms have. (private servers, threads, different channels etc.).

References:

1. https://www.guidingtech.com/skype-vs-discord-comparison/
2. https://blogs.skype.com/news/2019/04/04/call-up-to-50-people-at-once-with-skype/

Slack

Pros:

1. Integrated with many other apps (Google drive).
2. Has self-message features. Convenient place to save notes or attachments.
    a. Note: many of the cons only applies to the free plan. The paid plan resolves most of the limitations.

Cons:

1. Videos and voice calls are limited to 2 members on the free plan.
2. Voice calls on channels require paid subscription.
3. Message history is limited to 10,000 on free plan.
4. Screen sharing requires paid plan.
5. Does not have push to talk.

References:

1. https://slack.com/pricing

Chosen Technology: **Discord**

We chose Discord for our messaging platform. One of the most important reasons why was due to the free voice channel feature. Although Slack has more native integration we decided that the ability to extend Discord with bots will be adequate for our team.

Web Server

Web Server
Value: We want a web server that is efficient, easy to learn, and has a lot of online resources.

Apache V2.4.41

Pros:

1. Strong security.
2. Open sourced, strong community and lots of online resources.
3. Most widely used web server.

Cons:

1. Process based server, can cause overhead.
2. Configuring Apache could lead to serious security concerns.
3. Serving static content is not as fast as Nginx or IIS.

References:

1. https://apachebooster.com/blog/about-apache-web-server-its-advantages-and-disadvantages/
2. https://code.tutsplus.com/tutorials/apache-vs-nginx-pros-cons-for-wordpress--cms-28540
3. https://www.digitalocean.com/community/tutorials/apache-vs-nginx-practical-considerations

---

Microsoft IIS V10.0.17763.1

Pros:

1. Strong security.
2. Efficient resource management.
3. Lots of learning resources online.

Cons:

1. Can only run on Windows OS.
2. Can only be configured through a UI.
3. Lacks flexibility in some areas.

References:

1. https://hilton.org.uk/iis-asp-perlscript-ado
2. https://stackshare.io/microsoft-iis

Nginx V1.17.5

Pros:

1. High performance.
2. Event based model can handle more with less overhead.
3. Serves static content very fast.

Cons:

1. 3rd party support and extensions isn't as strong as Apache.
2. Less mature than other web servers.
3. No out-of-box support for dynamic content - must be rendered with an external tool/library.

References:

1. https://code.tutsplus.com/tutorials/apache-vs-nginx-pros-cons-for-wordpress--cms-28540
2. https://www.digitalocean.com/community/tutorials/apache-vs-nginx-practical-considerations

Chosen Technology: **Microsoft IIS**

We chose Microsoft IIS for our web server. IIS is one of the most widely used web servers and has fantastic community support. This also means IIS is very stable and capable, which is important for Exogredient. Furthermore, it has strong security and resource management.

Version Control

Top Options For Version Control Software and Hosting platform
Value: A critical need for our application is the ability to keep track of source code changes from contributors and manage these changes.

Git V2.23.0 with Github

Pros:

1. Has many GUI clients for beginner users.
2. Very popular. Not hard to find support if help is needed.
3. Available across platform.
4. Extensive documentation.
5. Faster than competitors.
6. Free.

Cons:

1. Has a bit of a learning curve: there are many different types of commands with many different options. It also takes some time to also learn best practices.

References:

1. https://git-scm.com/download/gui/windows
2. https://git-scm.com/doc

Mercurial V5.1.2 with OSDN

Pros:

1. Has lower learning curve compared to Git.

Cons:

1. Less hosting choices than Git.
2. Since Git is more popular, Git will be easier and faster to get help online if needed.

References:

1. https://www.mercurial-scm.org/quickstart

SVN V1.13.0 with ProjectLocker

Pros:

1. Is more centralized than Git. This can give the project manager more control.
2. Available across platform.

Cons:

1. Less hosting choices that Git.
2. Since Git is more popular its easier and faster to get help online if needed.
3. Most actions in Git are much faster. (see link under Git)
4. Community isn't as large as Git.

References:

1. https://subversion.apache.org/features.html
2. https://git-scm.com/about/small-and-fast
3. https://www.whoishostingthis.com/compare/subversion/

Chosen Technology: **Git(v2.23.0) with Github**

One of the main reasons we chose Git with Github is due to the popularity of the platform. We can be sure of reliability in terms of seeking support and extensive documentation. We also believe that the tool will continue remaining popular, so additional practice with it will help us in the future.

Database

Top Options For Database
Value: A critical need for our application is the ability to store, read, update, and delete the data that gets passed through our application from a reliable and scalable data store.
Microsoft SQL Server 2019 & SQL Server Management Studio 18 **Pros**: 1. Free version. 2. Visual tool SSMS 18. 3. Easily migrate to Microsoft Azure cloud without changing code. 4. AI for data analysis and prediction. 5. SSMS supports 32 bit and 64 bit machines. 6. Available on Windows and Linux (Red Hat and Ubuntu), as well as Docker. 7. Can connect to a Linux SQL Server instance from windows SSMS (VM). **Cons**: 1. Unavailable on macOS. 2. SSMS unavailable on Linux. **References**: 1. https://www.microsoft.com/en-us/sql-server/default.aspx 2. https://www.techopedia.com/definition/1243/sql-server 3. https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-overview?view=sql-server-ver15 4. https://www.microsoft.com/en-us/sql-server/sql-server-2019#Install 5. https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-manage-ssms?view=sql-server-linux-2017
MySQL Community Server 8 & MySQL Workbench 8 **Pros**: 1. Free version. 2. MySQL Workbench visual tool available on Linux (Ubuntu, Red Hat, Fedora), Windows, and macOS.

    a. Includes data modeler for ER diagrams.
    b. Supports forward and reverse engineering.
    c. Point and click database documentation with DBDoc.
    d. Syntax highlighting.
    e. Auto-complete.
    f. SQL snippet re-use.
    g. Export results to CSV.
    h. Visual performance dashboard with one click optimization.
    i. Database migration to Microsoft SQL Server.
3. Available for 32 bit and 64 bit machines.
4. Available for 32 bit and 64 bit machines.
5. Available for Linux (Ubuntu, Debian, Red Hat, Fedora), Windows, macOS, Oracle Solaris, and FreeBSD.
6. SQL and NoSQL support.
7. Triggers.

References:

1. https://dev.mysql.com/downloads/mysql/
2. https://www.mysql.com/products/workbench/
3. https://www.mysql.com/products/workbench/dev/
4. https://www.mysql.com/products/workbench/design/
5. https://www.mysql.com/products/community/

Google Firebase

Pros:

1. Real time.
2. Accessible through the web on all operating systems.
3. Can build a serverless application.
4. Authentication code is provided.
5. Cloud functions.
6. Optimized for offline use.
    a. Uses local cache when offline to serve and store changes. Synchronized when back online.

Cons:

1. Only one database per project for the free version.
2. 1 GB per month for the free version.
    a. For 25$ / month, only increases to 2.5 GB.
2. Only 125K invocations / month for cloud functions.

References:

1. https://firebase.google.com/
2. https://firebase.google.com/pricing

3. https://firebase.google.com/products/realtime-database/

Chosen Technology: **MySQL Community Server 8 & MySQL Workbench 8**

MySQL is supported on a larger amount of operating systems than MSSQL and the visual tool MySQL Workbench provides a larger amount of services than SSMS. MySQL supports SQL the same as MSSQL, but has the added feature of supporting NoSQL which we may use for logging and archiving. The feature of exporting data to CSV also expedites the archiving process. Google is optimized for being real-time and supporting PWAs, but the limitations of the free version will likely prove to be too costly. Also, MySQL's support for SQL allows for easier learnability for our development team based on experience rather than Google Firebase's strictly non-relational nature.

Development Tools

Top Options For Development Tools
Value: As a team we are need of tools that can speed up development, specifically a tool that can serve up our html in a browser and reload whenever we save a file rather than having to refresh every time.

Live Server V5.6.1 (VS Code Extension)

Pros:

1. Free.
2. Built for VS Code, button integrated into bottom right of editor to launch the extension.
3. No installing from command line or editing settings.
4. Live browser reload.
5. Customizable port.
6. Customizable browser.
7. Supports debugger for chrome.

References:

1. https://github.com/ritwickdey/vscode-live-server/blob/master/docs/settings.md

---

lite-server V2.5.4

Pros:

1. Free.
2. Live browser reload.
3. Customizable port.
4. Customizable browser.

Cons:

1. Installed from the command line with npm.
2. Requires editing setting json files.
3. Launched from the command line.

References:

1. https://www.freecodecamp.org/news/how-you-can-use-lite-server-for-a-simple-development-web-server-33ea527013c9/

---

browser-sync V2.26.7

Pros:

1. Free.
2. Live browser reload.

Cons:

1. Installed from the command line with npm.
2. Requires editing setting json files.
3. Launched from the command line.

References:

1. https://www.browsersync.io/

Chosen Technology: **Live Server V5.6.1 (VS Code Extension)**

Live Server incorporates into Visual Studio Code, our editor of choice, unlike the other options. It also has the live browser reload feature, the main reason for these tools to be used as it will speed up development. Also, the fact that it requires no additional setup other than an install allows for easy learnability for our development team.

Backend Framework

Backend Framework
Value: We want a backend framework that is fast, scalable, and easy to develop with.
NodeJS V10.16.3 **Pros:** 1. Async. 2. Great native and 3rd party library support. 3. TypeScript is more familiar to the development team and has type checking. **Cons:** 1. Single threaded. 2. Not good for CPU intensive work. 3. Callback hell. **References:** 1. https://www.netguru.com/blog/pros-cons-use-node.js-backend 2. https://www.educba.com/node-js-vs-asp-net/ 3. https://medium.com/@guillaumejacquart/experience-on-working-with-asp-net-core-and-nodejs-5e6c6351fc1f
ASP.Net Core V4.7.2 **Pros:** 1. Good framework support. 2. C# is easier to debug than JS. 3. Multi-threaded and async. **Cons:** 1. HTML control is limited. 2. Memory leaking problems and poor garbage collection. 3. Complicated page lifecycles. **References:** 1. https://www.educba.com/node-js-vs-asp-net/ 2. https://medium.com/@guillaumejacquart/experience-on-working-with-asp-net-core-and-nodejs-5e6c6351fc1f
Ringo JS V1.2.1

Pros:

1. Many out of the box libraries.
2. Package manager for downloading and managing external packages.
3. Async.

Cons:

1. Old JavaScript engine.
2. Not reactive.
3. Not many 3rd party libraries compared to other frameworks.

References:

1. https://www.simplytechnologies.net/blog/2018/6/19/nodejs-pros-and-cons-of-server-side-javascript

Chosen Technology: **ASP.Net Core V4.7.2**

We chose ASP.Net Core for our backend framework. ASP.Net is a stable technology backed by strong corporate support. This framework is used by a lot of high profile web apps too. Its programming language, C#, is also widely recognized as a performant language that has clear syntax.

Front-end Framework

Top Options For Front-End Framework
Value: Due to the limited amount of time allowed to learn new technologies, we need to choose a Front-End Framework that is easy and quick to learn while can support our needs.

Vue.js V1.2.1

Pros:

1. Vue.js has the smallest API surface area and is easy to start learning and using it.
2. All documentation is up-to-date and well-written.
3. UI developers can create flexible components and reuse them later even in other projects.
4. It has browser devtools extensions that allow to change separate components manually when needed.

Cons:

1. Too much flexibility could lead to irregularities in code.
2. Since Vue.js is relatively new, it is mostly supported by individual developers.

References:

1. https://existek.com/blog/top-front-end-frameworks-2019/

React V16.10.2

Pros:

1. Virtual Dom improves the experience of the user and the developer.
2. Since React uses isolated components, reusing the components save time.
3. One-direction data flow makes the code stable.
4. An open-source library with a variety of tools.

Cons:

1. It lacks documentation because of its high pace development.
2. The learning time is relatively higher than Vue.js.

References:

1. https://existek.com/blog/top-front-end-frameworks-2019/

Angular V8.0.0

Pros:

1. The component-based architecture allows creating components and reusing them.
2. TypeScript is the core language.
3. Angular Universal Support helps render the apps on a server.
4. Angular ecosystem is maintained by Google's long term support.
5. Angular Resources incorporates the UI environments, IDEs, analytics tools, facilities for ASP.NET, etc.

Cons:

1. Angular complexity makes it difficult to manage the components.
2. The learning difficulty is high as junior developers have to cover lots of materials such as components, modules, dependency injection, and etc.
3. CLI documentation is poorly described and hard to find out more on GitHub or other forums.

References:

1. https://existek.com/blog/top-front-end-frameworks-2019/

Chosen Technology: **Vue.js 1.2.1**

We chose Vue.js over React and Angular because:

1. Vue.js has well-written and up-to-date documentation while React and Angular are lacking in that aspect. This is crucial to our group since we are new to all of these, hence we need very good documentation for learning and utilizing the technology.

2. Vue.js is easy to learn and start using in a relatively short time compared to React and Angular. Although React and Angular might have more powerful, useful tools and functions, Vue.js is more suitable to our group since we need a technology that we can learn as fast as possible and start using it for implementation right away.

APIs

Top Options For Image Recognition
Value: We are mainly using it to take pictures of text so then a recommended list of ingredient names can be generated. We mainly want something that is accurate.
Google Cloud Vision API (latest version) **Pros**: 1. Free forever with (1000 units per month per account) for each functionality. 2. Accurately is very high. 3. Good at designing sophisticated large scale systems. a. Can handle huge sets of data. b. Large-scale data processing capacity (iterate through the data fast). 4. Google has the resources available to invest in and improve this technology. 5. Google has a huge dataset available for user. 6. Experienced users can adapt this api to the environment better, improving the accuracy and speed. 7. Optical Character Recognition (OCR) a. Moderate content. b. Detect the language and identify what is written. c. Deals with detected thing, syntactically and semantically. 8. Detected 125 labels (6.25 per image, on average) 93.6% of labels were relevant. a. Overall 125 labels - 93.9% relevance b. Small 28 labels - 96.4% relevance c. Middle 20 labels - 75% relevance d. Large 37 labels - 94% relevance e. Huge 34 labels - 100% relevance **Cons**: 1. JPEG with lower resolution (1 MB) negatively affect results. 2. There are some instances where it would either provide results with no labels above 70% confidence or results with high confidence but wrong labels. 3. Worse results when rotational invariance is introduced. a. Perform worse when images are rotated up to 90 degrees. **References**: 1. https://kaptur.co/google-cloud-vision-api-and-its-impact-on-the-image-recognition-industry/ 2. https://www.cognitiveclouds.com/insights/all-you-need-to-know-about-google-cloud-vision-api/ 3. https://cloudacademy.com/blog/google-vision-vs-amazon-rekognition/

4. https://medium.com/datadriveninvestor/4-simple-steps-in-building-ocr-1f41c66099c1

Amazon Rekognition (latest version)

Pros:

1. AWS free (12 months, up to 5000 procesconsed images per month per account) for rekognition functionality.
    a. Cheaper than Google Cloud for normal usage.
2. Object Detection (OCR).
    a. Syntactically and semantically.
3. Detected 129 labels (6.45 per image, on average) 89% of labels were relevant
    a. Overall 111 labels - 87.3% relevance
    b. Small 35 labels - 88.5% relevance
    c. Middle 10 labels - 100% relevance
    d. Large 31 labels - 86% relevance
    e. Huge 35 labels - 82.8% relevance
    f. Despite a lower relevance rate, always managed to detect at least one relevant label for each image.
4. Remain relevant when rotational invariance is introduced.

Cons:

1. JPEG with lower resolution (1 MB) negatively affect results.

References:

1. https://cloudacademy.com/blog/google-vision-vs-amazon-rekognition/
2. https://medium.com/datadriveninvestor/4-simple-steps-in-building-ocr-1f41c66099c1

Microsoft Azure Computer Vision API (latest version)

Pros:

1. OCR to identify text in different languages and identify what is written.
    a. Moderate content in images.
2. Use Computer Vision containers to recognize printed and handwritten text.

Cons:

1. Bad when rotational invariance is introduced.

References:

1. https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/home
2. https://codematters.online/computer-vision-api/
3. https://medium.com/datadriveninvestor/4-simple-steps-in-building-ocr-1f41c66099c1

Image type supported and Image Size Limits

Google	Supports most image formats used on the Web.	20 MB/Image
Amazon	Limited to JPG and PNG formats.	5 MB/Image, 15Mb/Image from S3
Microsoft	Supports only JPEG, PNG, GIF, BMP.	4MB/Image

Performance Testing (MacBook Pro, Krakow, 1000 tiles, 10 at a time)

	Average	Minimum	Maximum	90th percentile
Google	1.23s	0.69s	1.68s	1.42s
Amazon	2.42s	1.03s	3.73s	3.21s
Microsoft	1.11s	0.65s	5.07s	1.5s

References:

1. https://blog.filestack.com/thoughts-and-knowledge/comparing-google-vision-microsoft-cognitive-amazon-rekognition-clarifai/
2. https://cloudacademy.com/blog/google-vision-vs-amazon-rekognition/
3. https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/home

Chosen Technology: **Google Cloud Vision API**

We are going with Google Cloud Vision API. All of the top options have pretty high accuracy. They all have the OCR feature (optical character recognition), which allows for the recognition of text printed by a computer. There is a difference in performance testing though. Google has a more consistent ranging time for its performance. Nothing goes over 2 seconds compared to the other options (look at the chart above). When talking about average amount of labels and relevance of labels, I couldn't find the information for Microsoft but I could for Google and Amazon. Google detects about 6.25 per image with a 93.6% relevancy and Amazon is about 6.45 per image with a 89% relevancy. Google does have some problems with rotational invariance and accuracy/confidence but it is always improving because they have the resources to invest in this technology. Google is also good with large scale systems because they can handle large datasets. It also states that accuracy can improve once we get more used to the technology and adapt it better to our environment. It was a hard choice, but we chose Google because of its accuracy and its potential to improve.

Top Options For Web Mapping Service
Value: We are mainly using it to get distance between user's location and stores. We will also be using it to give the user a way to get directions to the store.
Google Map API (latest version) **Pros**: 1. Provide basics of mapping, pins, reviews. 2. Geolocation, autocomplete boxes, traffic, transit, and more. a. Multiple transportation modes and schedule for some. 3. Provide street view perspective. 4. Robust tool that can be used to create custom map, searchable map. 5. Mashup with different APIs. 6. Does not charge for services at low usage levels. 7. Huge number of online resources available (documentation). 8. Good code quality. a. Custom markers. b. Add geoJSONs. c. Layer interaction. d. Popups. 9. Can handle a huge number of markers more efficiently. 10. Skybox's real-time micro-satellite service for large amount of data. **Cons**: 1. Limited accuracy. a. May produce route that does not reach destination. b. Not up-to-the-minute information. c. Remote locations may not be in google maps. 2. Forced to use default Google base layer (although customizable). **References**: 1. https://www.techwalla.com/articles/disadvantages-advantages-of-using-the-google-maps-website 2. https://medium.com/@helenflam/why-when-and-how-to-use-the-google-map-api-f5dfa35986dc 3. https://www.codementor.io/victorgerardtemprano/google-maps-api-or-leaflet--what-s-best-for-your-project-faaev60vm 4. https://www.creativebloq.com/web-design/leaflet-google-maps-121413738
Leaflet (latest version) **Pros**: 1. Provide basics of mapping.

2. Clear standard Javascript-plugin-style documentation and a lot of online resources.
3. Good code quality.
    a. Custom markers.
    b. Add geoJSONs.
    c. Layer interaction.
    d. Popups.
4. Open-sourced.
    a. Adaptable and accessible.
    b. Checked on developers often.
    c. Control.
    d. Growing collection of plugins (frequent updates).
    e. Ability to change and design aspects of map without fear of licensing or unexpected change of data.
5. Javascript library.
    a. Multiple types of base layers.

Cons:

1. Relies on 3rd party services for some features.
    a. Geolocation.
    b. May be harder to implement for newer developers.

References:

1. https://www.codementor.io/victorgerardtemprano/google-maps-api-or-leaflet--what-s-best-for-your-project-faaev60vm
2. https://www.creativebloq.com/web-design/leaflet-google-maps-121413738

Mapbox (latest version)

Pros:

1. Standardized data flow for large organized projects.
    a. Easy access to data.
    b. Great for loading large amount of data.
2. Easy to customize map tiles.
3. Very fast loading due to custom tile generation.
4. A lot of documentation available online.
5. Good for interactivity of application, but it can be hard to do compared to other mapping APIs.

Cons:

1. Complex for simple projects or maps.
    a. Hard to learn compared to other mapping APIs.
2. Not a free service.
3. Strict data standardization can cause extra difficulties.

4. Available documents are not very detailed and can still be hard to understand and learn.

References:

1. https://www.codementor.io/victorgerardtemprano/pros-and-cons-of-using-mapbox-for-your-project-dx04pfgxw

Chosen Technology: **Google Map API**

We are going with Google Map API. All of the top options cover the requirements for the value I mentioned, but Google Map has extra functionality that we like. For the options, it is mostly between Google and Leaflet because Mapbox is more complex for simple projects or maps. It is also not a free service and has a steep learning curve. Now between Google Map and Leaflet, both can handle all basics of mapping quite well and they are both good quality code. They also have a lot of online resources available for new developers. The difference is that Leaflet is a open-sourced, developer-driven API. It is adaptable, accessible, and ever-growing. The thing is that services like geolocation and some other things are from 3rd parties. Google on the other hand provides geolocation itself and also have a street view perspective to make it easier to find stores. Google can handle a huge number of markers more efficiently and have a real-time micro-satellite service for large amount of data. For these reasons, we chose Google Map because it aligns with our goal of helping the user calculate the distance and find the store.

Top Options For Getting Places Information
Value: We are mainly using it to get details of stores (owner information, address, etc). That means store owners have to have their store in the database. That is why we want something that is mostly accurate and easy to sign up for and use.

Google Places API (latest version)

Pros:

1. Huge worldwide set of business listings.
2. Can filter businesses by name and by type.
3. Features
    a. Place search, details, photos.
    b. Place autocomplete.
    c. Query autocomplete.
4. Practical and easy to manage.
5. Easy to edit listing and manage information (customization).
6. Signing up is simple.

References:

1. https://www.quora.com/What-are-the-pros-and-cons-of-each-%E2%80%98places%E2%80%99-API-ie-Foursquare-versus-Google
2. https://developers.google.com/places/web-service/intro
3. https://support.g)oogle.com/google-ads/answer/143059?hl=en
4. https://docs.google.com/document/d/1BZeDYv-ec1DxGdzcTEdecgiGSrrtyyLMJ_9ilLBl71U/edit

---

Factual Global Places (latest version

Pros:

1. Over 130 million place records (19.4 million from the United States) and over 52 countries.
2. 25 core attributes (Name, address, geocode, website, email, etc.).
3. Existence score to filter out closed, duplicate, and junky records.
4. Factual is an agile-based. It also uses data processing pipeline to build, update, and maintain the dataset.
5. Factual uses a machine learning technology that processes out duplicate and differing data from the dataset.
6. Viewing is free.

Cons:

1. Download cost money (around $0.10 to $1 USD).

References:

1. https://www.factual.com/data-set/global-places/
2. https://streetfightmag.com/2012/02/13/comparing-the-pros-and-cons-of-5-top-location-apis/

Foursquare Places API (latest version)

Pros:

1. Huge database.

2. Database system is created many developers and it is focused towards check-ins feature.
3. Can filter businesses by name and by type.

Cons:

1. User-generated.
    a. It would only the most data where user base is big.
    b. It may contain duplicate or places that aren't really business (ex: individual's home, parks, etc).
2. Limits on access (500 requests per hour per user. 5000 requests per hour for venues or userless requests).

References:

1. https://www.quora.com/What-are-the-pros-and-cons-of-each-%E2%80%98places%E2%80%99-API-ie-Foursquare-versus-Google
2. https://streetfightmag.com/2012/02/13/comparing-the-pros-and-cons-of-5-top-location-apis/

Chosen Technology: **Google Places API**

We are going with Google Places API. All of the top options have huge a huge database. Out of the 3, Foursquare is not really what we are looking for. The database is user-generated, as in anyone can add information into the list. This can lead to duplication of businesses and existence of non-business listings. On the other hand, Factual and Google has what we are looking for. Factual has an existence score to filter out closed, duplicate, and junky places. They also use an agile-based data processing pipeline that builds, updates, and maintains the dataset so integrity isn't compromised.Viewing information is free but it cost about $0.10 to $1 USD to download. Google also provides a service with accurate information and is free for $200 USD worth each month. We chose Google Places because of the various reasons mentioned and it's sign up process and use it simple for users.

Top Options For SMS Messaging And Phone Calls
Value: We are mainly using it to message and call users for verification. Basically we need something that is easy to use for communication through SMS and calls.

Twilio (latest version)

Pros:

1. Easy to build and customize own applications.
2. Easy to integrate communication functions in existing website or apps easily.
3. Utilizes over 1000 mobile carriers in over 150 countries for voice and SMS services.
4. High level of dependability on website.
5. Constantly evaluating carriers to rule out fraudulent and bad ones. Actively monitor utility and adjust routes.
    a. Results in best routine and best carriers to handle user's data.
6. 24/7 support but more expensive plans are prioritized.
7. Incoming/outgoing calls are charged on a per minute basis.

References:

1. https://getvoip.com/blog/2016/06/01/twilio-alternatives/

Bandwidth (latest version)

Pros:

1. Good reputation.
    a. One of the original carriers for Twilio.
    b. Provider behind Google and Skype communication services.
    c. First telecommunications provider to their own API service that is specific to voice and message functionalities.
2. Removed middle man.
    a. Cheap while providing good services. ($0.35 USD per month for dedicated number compared to Twilio $1).
    b. No worries about partnering with other providers/carriers, while maintaining a robust network throughout the United States and Canada.
3. 24/7 support through online tickets or phone. Emergency service number too.

Cons:

1. Does not sell or host short code numbers.

References:

1. https://getvoip.com/blog/2016/06/01/twilio-alternatives/

Nexmo (latest version)

Pros:

1. SMS numbers available in over 35 countries and voices in over 90 countries.
2. Work with direct carrier network so numbers and service is to continue to advance and grow in many markets.
3. Adaptive routing. Messages sent with best route in mind, ensuring best possible time.

4. Flexible pricing (per second instead of per minute).
5. 24/7 support through email.
    a. For urgent issues, guaranteed response time of 2 hours during the week and 4 hours during the weekend.
    b. Minor problems guaranteed response time of 6 hours.
    c. Upgrade available ($5000 USD a month) for support through phone and chat. Guaranteed response time of 30 minutes for urgent issues and 1 hour for minor issues.
6. No charge for incoming SMS.
7. Shared short code number for free.

References:

1. https://getvoip.com/blog/2016/06/01/twilio-alternatives/

Chosen Technology: **Twilio**

We are going with Twilio. All of the top options pretty much provide the same kind of services for a price. The price ranges for each. Customer support for all 3 is pretty much 24/7 with a small variation. It seems like Nexmo is more customer friendly because they offer cheaper alternatives. Price is per second instead of per minute. Their customer service seems a bit better because it is handled quickly. It does not charge user for incoming SMS and a shared short code number is provided for free. On the other hand, Bandwidth removed the middle man of network carrier which makes it the second cheapest in this list. Twilio is the most expensive but it does utilizes a large range of mobile carriers. It is also said that provider boasts a high level of dependability on website. They evaluate the carriers to rule out SIM farms and carrier filtering. We chose to go with Twilio because it offers an environment for developers and companies to create, host, and deploy communication applications that are easy to customize and integrate.

Top Options For Stemming Algorithm
Value: We are mainly using a stemmer to associate ingredient names with each other. For example: apples, apple pie, apple drink should be associated with apple.

Porter Stemmer (latest version)

Pros:

1. Most commonly used stemmer.
2. One of the most gentle stemmers.
3. Java supported.
4. Most computationally intensive of the algorithms.
5. Oldest stemming algorithm.

References:

1. https://stackoverflow.com/questions/10554052/what-are-the-major-differences-and-benefits-of-porter-and-lancaster-stemming-alg

---

Snowball Stemmer aka Porter2 (latest version)

Pros:

1. Regarded as an improved Porter Stemmer by a lot of people.
    a. Improvement on speed.

References:

1. https://stackoverflow.com/questions/10554052/what-are-the-major-differences-and-benefits-of-porter-and-lancaster-stemming-alg

---

Lancaster Stemmer aka Paice-Husk (latest version)

Pros:

1. Faster compared to Porter and Snowball.
2. Can reduce working set of words hugely.

Cons:

1. Very aggressive which can result in bad data.
2. Not as intuitive to a reader as Porter and Snowball.

References:

1. https://stackoverflow.com/questions/10554052/what-are-the-major-differences-and-benefits-of-porter-and-lancaster-stemming-alg

ChosenTechnology: **Snowball Stemmer**

We are going with Snowball Stemmer; it is also known as Porter2 and is widely known as the better version of Porter. It is slightly faster at computation than Porter and has a large community around it. Stemmer has practically most of the pros from Porter but with some extra support. On the other hand, Lancaster is a bit too aggressive and that could lead to a fault. It isn't as intuitive as Porter and Snowball. For these reasons, we are choosing to go with Snowball.

Programming Languages

Top Options For Backend Programming Languages
Value: We are mainly looking for a back-end language that is easy to deploy on a network. There should not be any or at least not much problems with deployment. It should have some security aspects.

C# (v7.3)
Pros:
1. The preferred architecture for backend languages and automation in the Windows environment.
   a. Integrate well with Windows.
   b. No special configurations necessary to get program to run in Windows.
2. .NET language with a C-style format.
3. Easy to deploy project on network.
   a. As long as server/workstation supports .NET. Should be a smooth transition from development from production.
4. Compiled language.
   a. Code stored in binary form on a public-facing server.
   b. If server is compromised, source code is not automatically compromised.
5. Large community support and a lot of documents available online.
Cons:
1. Compiled code.
   a. More difficult to work with since it has to be compiled each time something is changed.
   b. Could lead to bugs if changes isn't thoroughly tested each time.
2. The server application must be windows.
   a. To execute any .NET application a windows platform is needed.
3. Older .NET frameworks stopped being updated after a few OS upgrades.
References:
1. https://agilites.com/pros-and-cons-of-using-c-as-your-backend-programming-language.html
 	
Visual Basic.NET (v.15)
Pros:
1. About the same support for C# and Visual Basic from Microsoft.
2. Easy syntax to learn and understand (lower learning curve).
   a. Straightforward.
   b. Structured

3. Object-oriented like Java.
4. Platform independent.
    a. Can be compiled on a variety of computer platforms and be executed on different OS.
5. Often times regarded as slightly better than C# based on productivity aspects of completed software.
6. Large community support and a lot of documents available online.

Cons:

1. Third-party companies support C# more.
    a. More access to samples, snippets, blog support, etc.
2. Most schools only teach C derivatives and most developers would stick with what they start with.
3. C# viewed as the language of the future while VB is viewed as the past.

References:

1. https://www.ethany.com/choosing-between-c-or-vb-net-pros-and-cons/
2. https://www.educba.com/what-is-vb-dot-net/

F# (v. 4.7)

Pros:

1. Less code than C#.
    a. No curly braces or semicolons needed.
    b. No types need to be declared because of type inference technique. It is as type-safe as C#, or safer.
2. Simple.
3. Interactive development.
    a. Can write code and test it immediately.

Cons:

1. Less code than C# also means indentation has to be carefully done.
2. Smaller community than C# and VB.net.
3. Weaker tooling.

References:

1. https://fsharpforfunandprofit.com/posts/fvsc-sum-of-squares/

Chosen Technology: **C#**

C# fits the value we need for development. Visual Basic also provide similar functionalities as C#, but it isn't as popular. Third-party companies support C# more and most students are being taught a C language in school. The way they are supported and viewed are entirely different. F# is also a simple language to learn and use. It also provides features to help reduce code and a development tool to easily play around and

test code. The downside is, F# has a smaller community and some syntax needs to be carefully examined. We decided to go with C# because it is a derivative language of C, which we are familiar with. The language makes deployment easier and it is also the most popular of the three choices.

Top Options For Interpreted Scripting Language
Value: We need something that is fast, easy to learn, and easy to implement. Something that is popular and would continue to get better. Being popular usually also means a lot of resources available online.
ECMAScript 2016 (Javascript) **Pros**: 1. Speed. a. Very fast because it can run immediately in client-side browser. b. Unless there is an outside resource, it would be slowed down by any network calls (backend server). c. No compilation needed on client side. 2. Simplicity and flexibility. a. Learn and implementing. 3. Popularity. a. Used practically everywhere on the web. b. A lot of resources available. 4. Interoperability. a. Interacts with other languages nicely and can be used in a huge variety of applications. b. Any file extension can be inserted into web page. 5. Server load is reduced (client-side less demand on server). 6. Extended Functionality if third party add-ons is enabled. 7. Versatility (many possible actions with javascript through Node.js servers). 8. Updates are usually available every year. 9. Supported by all modern web browsers. **Cons**: 1. Client-Side Security. a. Exploitation can be an issue because code is executed on the user's computer. 2. Browser Support .

    a. It can be interpreted differently from browsers to browsers.
    b. Client-side code may produce different output (Server-side is not affected by this problem).

References:

1. https://guide.freecodecamp.org/javascript/advantages-and-disadvantages-of-javascript/
2. https://bytescout.com/blog/2016/07/javascript-vs-coffeescript-vs-typescript.html

---

TypeScript (v. 3.6.3)

Pros:

1. Completed superset of Javascript.
    a. All Javascript code works in TypeScript.
    b. May be the future path of JavaScript.
2. Support for modules.
3. Optional parameter function.
4. Strict typing (everything stays the way it is defined), strongly-typed (static typing).
    a. Makes it easier to interface with external JavaScript libraries.
5. Structural typing, type annotations, type inference.
6. Great deal of syntax taken from object-oriented programming.
    a. Interfaces.
    b. Classes.
    c. Enumerated types.
    d. Generics.
    e. Modules.
7. Precise defining through typing.
    a. Developers are notified of mistakes before they can push the code to the repository.
8. Types can make code management easier, which would potentially increase productivity.
9. Easier to code than Javascript.

Cons:

1. Overly complicated typing system.
2. Required compilation.
    a. Type checks only work though compilation.
3. False sense of security (Developers may rely on it too much).
4. No support for abstract classes.

References:

1. https://stxnext.com/blog/2019/08/30/typescript-pros-cons-javascript/
2. https://www.geeksforgeeks.org/difference-between-typescript-and-javascript/
3. https://bytescout.com/blog/2016/07/javascript-vs-coffeescript-vs-typescript.html

CoffeeScript (v. 2.4.1)

Pros:

1. Popular.
2. Similar syntax to Javascript but it is a bit more clean and concise.
3. Easy to read and learn.
4. Variable scoping is better than JavaSCript.
    a. Variables declared for user in output.
5. Aproximately 55% needed code than Javascript for same functionality.
6. More limited syntax making it easier to manage in bigger projects.
7. “It’s just JavaScript”. Code compiles one-to-one into equivalent Javascript.
    a. No interpretation at runtime.
    b. Use Javascript libraries easily.
    c. Compiled output is easy to read because it is neatly printed.
8. Often times run as fast or even faster than Javascript.

Cons:

1. Can be difficult to compile.
    a. Syntax mistakes can give it some problems.
2. Not as a big community as Javascript.
3. Section logic might not work like it should.
    a. May have to spend time to study and learn about the underlying javascript code.

Referferences:

1. https://coffeescript.org/
2. https://bytescout.com/blog/2016/07/javascript-vs-coffeescript-vs-typescript.html

Chosen Technology: **ECMAScript 2016**

We are going with ECMAScript 2016. ECMAScript is basically the parent of the other two alternatives. It would just be better to work with ECMAScript then use the other two if needed. It practically covers all of the value we mentioned above. It is also the most popular. There are a lot of resources available. Being easy and having a lot of resources can save us a lot of time, which is very important for this project.

Top Options For Markup Language
Value: We are looking for something that is popular and have a lot of resources available online. Something that is easy to learn, simple to use, and be flexible.

Something with something with good structure and support. Most importantly, something that is consistent between browsers and platforms.

HTML5

Pros:

1. Support web applications that deals with user interactions, local data, and servers.
    a. Effective and easy.
2. Constantly growing user base.
3. Simple and neat code.
    a. Easy to write code that are descriptive and clear at the same time.
    b. Semantic code (separation of meaning from style and content).
4. Consistency.
    a. HTML code structure for creating web page are consistent from one page to another. This makes it easier for developers/designers to immediately understand it.
5. Page layout.
    a. Offer old elements like Paragraph, Div, Span, and Heading, but also new ones like Area, Article, Header, Footer, and Aside.
    b. Allows for more possibility of structural construction, dependant on purpose.
6. Forms.
    a. HTML can create forms without the need for Javascript code.
    b. Forms are also formatted nicely, creating a better user experience.
    c. Formats/functions include options for different types of text inputs, search, and fields (dependant on purpose).
7. Geolocation support for any compatible browser-based application.
8. Mobile.
    a. A lot of tools available for creating mobile sites and apps.
    b. Mobile browsers support makes designing and constructing easier.
    c. From desktop to mobile and vice versa, not a lot of tweaking and revisions needed.
9. Open and accessible content.

Cons:

1. Only modern browser support HTML5.
    a. There's also a lack of compatibility with Internet Explorer.
2. Even though it is stable, HTML5 is still considered a work in progress.
    a. Not expected to be a finished language for a few more years. During this time, changes could be made at any time.
3. Media Licensing Issues.

    a. To be compatible with most browsers, the media had to be first compressed in multiple formats. This can be a tedious process and is a pain to do.
4. Multiple device functionality.
    a. It may not look the same from devices to devices. It may also not render correctly on older devices/ newer devices.
    b. It may also cost a lot of resources to develop and test from one device to another (cost of development and test environment).
5. Less available cache space/storage for instant recovery functionality (offline/online mode).
6. May experience slower speed at times because data and screen components may be needed for each pulled down page.

References:

1. https://www.lucidsoftech.com/blog/html-5-pros-cons
2. https://www.educba.com/html5-vs-html4/

HTML4.1

Pros:

1. More Stable.
    a. Standard for developing browser applications.
    b. More than 10 years of developer user.
2. Compatibility with browsers is better than HTML5.

Cons:

1. Less features compared to HTML5.
    a. Fewer available tags.
2. Worse at error handling.

References:

1. https://www.educba.com/html5-vs-html4/

Chosen Technology: **HTML5**

We chose to go with HTML5 because it is the latest version. Browsers are always updating so it would be smart to use the latest technology to work with them. HTML5 has new features that HTML4.1 does not. The main con of HTML5 was that it is still a work in progress, so there may be some inconsistencies. Even though that may be a problem, the user base is continuing to grow. It is just better to work with it because simple, effective, and improving.

Top Options For Style Sheet Language

Value: We need a style sheet language that can be responsive and consistent across browsers and different screen sizes.

CSS

Pros:

1. Most used styling language.
2. Lost of online resources.
3. Able to generate CSS code online, which cuts development time significantly.
4. Many styling libraries to choose from.

Cons:

1. Can be difficult to style certain elements in a certain way.
2. Hard to learn for a beginner; there are many gotchas.
3. Fragmentation across browsers, which makes compatibility tests harder.

References:

1. https://connectusfund.org/6-advantages-and-disadvantages-of-cascading-style-sheets
2. https://www.webdeveloper.com/d/262910-css-and-its-advantages-and-disadvantages

---

Sass

Pros:

1. Better nesting.
2. Can write CSS equivalent code quicker than in CSS.
3. Compatible with every CSS version, which means any CSS libraries work too.

Cons:

1. Must be preprocessed before running.
2. Features are slowly getting ported over to CSS.
3. May cause loss of the browser's built-in element inspector.

References:

1. https://www.javatpoint.com/sass-advantages-and-disadvantages
2. https://blog.udemy.com/less-vs-sass/

Less.js

Pros:

1. Valid CSS code works.
2. Better nesting.
3. Better media queries.
4. CSS libraries can work with Less.js too.

Cons:

1. Must be preprocessed before running.
2. Requires a special project structure.

3. Features are slowly getting ported over to CSS.

References:

1. https://amasty.com/blog/magento-2-less-vs-css-starter-guide/
2. https://blog.udemy.com/less-vs-sass/

Chosen Technology: **CSS**

We have chosen CSS as our style sheet language for many reasons. CSS is the most widely used styling language, therefore it has a tremendous amount of resources online. CSS also has many libraries available to it, which can cut down on development time. Lastly, there are many CSS code generators online, which can remove a lot of the headache from CSS when trying to style components to work across browsers and screen sizes.

Deployment

Browser support desktop: Chrome, Mozilla, Safari
Browser support mobile: Chrome, Mozilla, Safari
Mobile OS's: Android / iOS
Cloud Computing/Hosting Platform: Google Cloud
Web Server: Microsoft IIS (V10.0.17763.1)
Certificate Authority: LetsEncrypt

Cloud Provider

Top Options For Cloud Provider
Value: Cloud services provide the resources and environment necessary to host our product. These resources include hardware and software that aid with the deployment process of our product.

Google Cloud Services

Pros:

1. Offers access to Google's wide array of tools, such as the Google Vision API.
2. VMs running on compute engine come with persistent storage.
3. Google's Kubernetes Engine (GKE) is considered the best and easiest to use.
4. Heavy focus on security.

Cons:

1. Offers the least amount of services compared to the other two alternatives.
2. No backup or archiving options.

References:

1. https://www.datamation.com/cloud-computing/aws-vs-azure-vs-google-cloud-comparison.html
2. https://www.computerworld.com/article/3429365/aws-vs-azure-vs-google-whats-the-best-cloud-platform-for-enterprise.html

---

Amazon Web Services (AWS)

Pros:

1. Offers the most amount of services, due to being around the longest.

Cons:

1. Pricing can be hard to determine due to many hidden fees.
2. Large amount of services tends to be overwhelming for newer users.

References:

1. https://www.datamation.com/cloud-computing/aws-vs-azure-vs-google-cloud-comparison.html
2. https://www.computerworld.com/article/3429365/aws-vs-azure-vs-google-whats-the-best-cloud-platform-for-enterprise.html

---

Microsoft Azure

Pros:

1. Built to integrate well with Microsoft tools such as .NET and SQL Server.
2. Supports a wider range of options for app deployment.

Cons:

1. Known for being difficult to configure.
2. Microsoft has had the most number of outages out of the 3, so it has the least dependable service.
3. Some users have reported that documentation is lacking.

References:

1. https://www.datamation.com/cloud-computing/aws-vs-azure-vs-google-cloud-comparison.html

2. https://www.computerworld.com/article/3429365/aws-vs-azure-vs-google-whats-the-best-cloud-platform-for-enterprise.html

Chosen Technology: **Google Cloud Services**

All three products offer more than enough services needed to deploy and host our product. Due to lack of funds, cost played a big role in our decision. Google's platform includes access to many Google APIs we plan on using like Google Vision, Places, and Maps. Though it has the least amount of overall features offered between the 3 platforms, it has no complaints of being hard to setup and maintain like Azure and AWS.

Certificate Authority

Top Options For Certificate Authority
Value: A certificate authority is a provider of SSL certificates, which are necessary to secure a website with the HTTPS protocol. The added security allows for your website to process sensitive data with clients.
LetsEncrypt **Pros**: 1. Free domain validation(DV) and Subject Alternative Name(SAN) certificates provided. 2. Offers an installer, Cerbot, which streamlines the process of acquiring a certificate. 3. Supported on all popular browsers. **Cons**: 1. Only offers 2 certificates types: DV and SAN. **References**: 1. https://premium.wpmudev.org/blog/ssl-certificate-authorities-reviewed/
Comodo **Pros**:

1. Offers multiple types of certificates including Domain Validation, Extended Validation, Wildcard, and Organization Validation.
2. Every certificate comes with a warranty.
3. More premium certificates come with the option of ECC 256-bit encryption.

Cons:

1. Has a cost above $0.

References:

1. https://premium.wpmudev.org/blog/ssl-certificate-authorities-reviewed/

Symantec

Pros:

1. Offers DSA encryption for some its more premium certificates.
2. All certificates come with a daily malware scan of your website.

Con:

1. Very high cost for most of its certificates.

References:

1. https://premium.wpmudev.org/blog/ssl-certificate-authorities-reviewed/

Chosen Technology: **LetsEncrypt**

We chose LetsEncrypt for our certificate authority since it meets all the needs of our product and it comes without a cost. We do not intend to include any functionality involving payment processing, so it is unnecessary to purchase a certificate with a warranty.

DNS Provider

Top Options For DNS Provider
Value: A DNS provider owns servers that map the IP addresses of websites to their domain names.

NameCheap

Pros:

1. Provides a FreeDNS service if you already own a hostname.
2. Offers PremiumDNS which provides speed and security upgrades for $4 a year.
3. Automatically carries out website backups every two days.

Cons:

1. Lower hardware resources are provided for new users.

References:

1. https://www.namecheap.com/domains/freedns/
2. https://cdnify.com/blog/10-best-free-dns-hosting-providers/

Godaddy

Pros:
1. Hosting plans include unlimited bandwidth, Microsoft Office 365, and email functionalities starting at $3 a year.
2. Premium hosting plans cost $7 a year and offer unlimited storage and DDoS protection.

Cons:
1. Must sign up for 3 years to get rates mentioned above.
2. Customer support only available on working hours on weekdays.

References:
1. https://webhostingmedia.net/godaddy-pros-and-cons/

Hurricane Electric Internet Services

Pros:
1. Free DNS hosting offered with servers all over the world.
2. Offers multiple domains for 1 account.

Con:
1. Dated interface can make configuration and navigation of the product very difficult.

References:
1. https://dns.he.net/

Chosen Technology: **Namecheap**

All three DNS providers offer roughly the same service at a very low cost. Namecheap was chosen because it supports DynamicDNS and multiple DNS record types at the free tier. Though Godaddy offers similar services, their pricing model requires a 3-year commitment for cheaper yearly prices. Namecheap provides us with the freedom of choosing a free DNS host and upgrading when other features are necessary.

Proxy Server

Top Options For Reverse Proxy Server
Value: A reverse proxy server is meant to aid the proper flow of network traffic. A reverse proxy is typically located inside a network and filters all requests such that clients never have direct contact with the main web server.

Nginx Reverse Proxy

Pros:

1. Offers better performance when compared to alternatives due to being built to specifically be either a web server or a reverse proxy.
2. Uses fewer hardware resources than alternatives.
3. Able to process up to 500 million HTTP requests in a day.

Cons:

1. Not much community support.

References:

1. https://www.keycdn.com/support/nginx-vs-apache
2. https://www.linuxnix.com/which-reverse-proxy-is-good/

Pound

Pros:

1. Compatible with all types of web servers.
2. Detects failing backend servers and automatically load balances to avoid non-functioning servers.
3. Created with a focus on security.
4. Able to decrypt HTTPS requests.

Cons:

1. Tends to use more resources than other alternatives

References:

1. https://www.linuxnix.com/which-reverse-proxy-is-good/

HAProxy

Pros:

1. More efficient load-balancer when compared to other alternatives.
2. Handles HTTPS requests more efficiently than alternatives.

Cons:

1. Load balancing is its main strength and it tends to lack in other aspects.

References:

1. https://alternativeto.net/software/haproxy/

Chosen Technology: **Nginx Reverse Proxy**

All three products offer decent options for load balancing and network security, however Nginx manages to do all of the same things as the alternatives while using less resources. Even at max load, Nginx reverse proxy servers only use about 15MB of RAM compared to 400MB of RAM that's used by Pound. HAProxy seems to handle requests faster in some scenarios however, its main function is load balancing so it lacks in areas, like SSL management.

Database Cluster Management System

Top Options For Database Cluster Management System
Value: Clusters are a form of High Availability(HA) solutions and they serve the purpose of ensuring a database can remain functional for long periods of time. High availability solutions typically involve storing data records in a redundant fashion in order to preserve functionality without having a single point of failure.
Galera **Pros**: 1. Synchronous replication allows for loss of nodes without service interruption. 2. Automatically handles partitioned nodes. **Cons**: 1. Lacks a built-in load balancer. 2. Sometimes background threads handling transactions can cause a crash. 3. Lacks in performance when compared to the alternatives. **References**: 1. https://www.percona.com/pdf-viewer/viewer.html?file=https://www.percona.com/sites/default/files/presentations/Percona%20XtraDB%20Cluster%20vs%20Galera%20Cluster%20vs%20MySQL%20Group%20Replication.pdf Percona XtraDB **Pros**: 1. Offers State Snapshot Transfer(SST) which automatically replicates new nodes with existing ones. 2. Comes with ProxySQL load balancer. 3. Automatically handles partitioned nodes in the cluster. 4. Fully compatible with all popular cloud providers. 5. Auto bootstrap is available in the case that all nodes lose power. **Cons**: 1. Has a limited maximum transactions size. 2. Sometimes background threads handling transactions can cause a crash. **References**: 1. https://www.percona.com/software/mysql-database/percona-xtradb-cluster 2. https://dzone.com/articles/choosing-mysql-high-availability-solutions MySQL Group Replication **Pros**:

1. Offers the largest pool of OS support.
2. Offers the most feature-filled self-testing suite out of the three.

Cons:

1. Doesn't automatically handle partitioned nodes correctly.
2. Lack of partitioned node management often leads to split brain.

References:

1. https://www.percona.com/pdf-viewer/viewer.html?file=https://www.percona.com/sites/default/files/presentations/Percona%20XtraDB%20Cluster%20vs%20Galera%20Cluster%20vs%20MySQL%20Group%20Replication.pdf
2. https://dzone.com/articles/choosing-mysql-high-availability-solutions

Chosen Technology: **XtraDB**

We decided to choose Percona XtraDB for our database cluster management system. XtraDB is compatible with all the technology we are using. It is also very reliable and has advanced mechanisms to replicate data efficiently and accurately. Furthermore, it comes with ProxySQL load balancer, which other systems lack.