



---

# TECH FEASIBILITY

---

A large, faint watermark in the center of the page, showing a pair of blue headphones with a green plant growing from the center, similar to the logo in the top right.

Kenzie Norris  
Krystian Bednarz  
Sam Asher  
Preston Lee

Sponsor: Dr. Okim Kang  
Organizer: Michael Leverington

Mentor: Daniel Kramer

---

# Table of Contents

<b>1. Introduction</b>	<b>1</b>
<b>2. Technological Challenges</b>	<b>2</b>
<b>3. Technological Analysis</b>	<b>3</b>
<b>3.1: Backend Host</b>	<b>3</b>
3.1.1: Criteria Descriptions	4
3.1.2: Proposed Technologies	4
3.1.3: Analysis	5
3.1.4: Analysis Table	6
3.1.5: Proving Feasibility	7
<b>3.2: Website Programming Language</b>	<b>7</b>
3.2.1: Criteria Descriptions	7
3.2.2: Proposed Technologies	8
3.2.3: Analysis	9
3.2.4: Analysis Table	12
3.2.5: Proving Feasibility	12
<b>3.3: Game Application Programming Language</b>	<b>13</b>
3.3.1: Criteria Descriptions	13
3.3.2: Proposed Technologies	14
3.3.3: Analysis	15
3.3.4: Analysis Table	17
3.3.5: Proving Feasibility	17
<b>4. Technological Integration</b>	<b>18</b>
<b>4.1: The Connected Product</b>	<b>18</b>
<b>5. Conclusion</b>	<b>19</b>

# 1. Introduction

For decades, the US has been the #1 study abroad country for students across the world. One of the major things apart from coming to the US to study abroad is the learning of English as the country's main language. However, one of the main issues with English is evidently how complicated it is from an outside perspective, and as young adults and older, learning new languages gets harder and harder the older you get. Websites like Duolingo are effective for learning a language's vocabulary and some of its grammar, but don't truly help you prepare for real conversations. Certain key factors to communication like context are lost when practicing flashcards of simple vocabulary.

Our clients, Okim Kang and Kevin Hirshi, are both a part of the NAU department of English. For years they have been working with different teams of developers to develop new tools and applications to help aid transfer students adjust and master the English language. They want to help students transferring to the US learn the language faster, with the use of context, a key element that is missing from modern day language learning applications. On top of that, they want to help transfer students master the proper pronunciation, another thing missing from those same language learning sites. What our goal is to take the years of research Kevin and Okim have compiled, and use it to create a gamified program that focuses on more than just vocabulary and grammar learning; teaching students context and helping them master their pronunciation.

One of the main goals of the project is to create a system that helps provide the User with the tools to help build their contextual awareness, as well as master their pronunciation of the English language. The website itself will have multiple game elements, including different styles of practicing words, as well as letting the user look up any word in what will become our Language Library, and get that word in context to help that User understand where the word is used. A more open-world style game will eventually be implemented that will allow the user to see objects and interact with them, before those words will go into the User's personal library of words they can reference later.

To bring this idea to life, the entire system needs to be broken down into smaller pieces, and those smaller to see where we as a team can tackle them, and succeed in bringing this application to life.

## 2. Technological Challenges

When creating such an application, a few things need to be taken into account. In this section we will discuss the main challenges that come to fruition when bringing this website to life. These main challenges we've broken into four sections.

### Backend

The Backend Host for the website application will need to store multiple items in order to properly assist the website and subsequent gamified elements. First and foremost user information needs to be stored and secured, this includes:

1. Username and Password,
2. The User Data, statistics of the User's Practice
3. A Personal Notebook of Words
4. Users' Activity and Progress

On top of this, the backend needs to host all of the data provided by the client. This includes a set of text and audio files, which are connected using what's called a textgrid file. The Database should store these accordingly and allow the user to access each of these connected pieces.

### Website Application

The Website itself needs to be a User interactive site that has a friendly UI and user friendly navigation throughout its multiple features. These include:

- The Profile Page: A home page for the user to display their statistics, progress, and give them daily missions if they so wish to complete them
- The Library: A search feature that is open to anyone regardless of owning an account, that will allow a User to search a word and get results with text providing context, and audio results as well
- The Game Page: A page that provides different options for practicing a word of the users choice, a pronunciation game, a context text-based game, and a sentence building game to help build more than just the vocabulary
- The Interactive Forum: A system to allow the administrative users to speak with the generic users of the site. This would build a community that is integral to sites like this

All of which need to present itself as a fast and professional looking platform that is not only inviting to users but gives off a sense of this being a professional application.

## **Game Application**

The game application will be built into the web application, and will provide multiple options in terms of minigames for the user to help hone in their skills. A few of these minigames would include, a dictionary game, a context selection game, a pronunciation game, and a few others all with the idea to help promote practice and spaced repetition. These games will help add to a user's statistics which will be able to be viewed on the profile page. This will need to communicate with not only the library but also need to communicate with the users current statistics.

## **3. Technological Analysis**

Our project consists of multiple factors that relate to individual technologies, each with specific roles tailored to the completion of the project overall. Each technology has multiple options for development based on our prior research, where the relevancy of each option was determined based on their own potential usefulness for the project. How well each option would fit our desires for the project is graded based on previously agreed upon criteria, designed to accurately score how well each option would work for our project needs. Our scoring system consists of grading each option with each criteria, receiving a score out of 5 for each criteria. The combined score from each criteria for every option determines how well of a fit each option would have if used in our project development. Total scores are then compared between each option, and the option with the highest score is chosen for use in our project.

### **3.1: Backend Host**

The Backend Host component plays a crucial role in storing and securing user game stats, as well as hosting and connecting large amounts of text and mp4 files. It's essential that we select a technology that can scale up as needed, while also offering cost efficiency, ease of use, and robust security features. To that end, we have identified several cutting-edge technologies, and we will assess them meticulously based on their ability to meet our requirements.

### 3.1.1: Criteria Descriptions

- Scalability
  - This criterion measures the ability of the hosting platform to scale up or down easily as the demand for the application changes. This is important because as the project gains traction, more users will be accessing it, and the platform needs to be able to handle the increased traffic.
- Cost
  - This criterion measures the cost of using each hosting platform. It includes both the upfront cost of getting started with the platform as well as any ongoing costs for usage and maintenance. This is important because the project is starting as a small prototype, and cost is a factor that needs to be considered before committing to a platform.
- Ease of use
  - This criterion measures how easy it is to set up and use each hosting platform. It includes factors such as the availability of documentation and support, as well as the ease of integration with other technologies. This is important because the team is likely to be small and may not have dedicated DevOps resources to manage the backend hosting platform.
- Security
  - This criterion measures the level of security provided by each hosting platform. It includes features such as encryption, access control, and threat monitoring. This is important because the platform will be handling sensitive user data, and security is critical to ensure the privacy and protection of user information.

### 3.1.2: Proposed Technologies

- Azure Cosmos DB
  - Azure Cosmos DB is a globally distributed, multi-model database service designed for low latency, high availability, and elastic scalability of data.
- Amazon Web Services (AWS)
  - Amazon Web Services is a cloud computing platform that provides a wide range of services, including hosting and storage.
- Google Cloud Platform

- Google Cloud Platform is a suite of cloud computing services that provide scalable and reliable infrastructure, storage, and data analytics tools for building and deploying applications on the web.

### 3.1.3: Analysis

We will evaluate each proposed technology based on the criteria described above, giving a score out of 5 for each criterion.

- Azure Cosmos DB
  - Scalability- 4.5/5
    - Azure Cosmos DB provides global distribution and automatic scaling, making it a highly scalable option for handling large amounts of data.
  - Cost- 3.5/5
    - Pricing can be complex, but it offers a pay-as-you-go model that allows for flexibility and cost control.
  - Ease of Use- 3/5
    - While Azure Cosmos DB has a user-friendly interface, the complex pricing and setup can make it challenging to use at times.
  - Security- 4.5/5
    - Azure Cosmos DB offers comprehensive security features, including encryption, role-based access control, and network security policies.
- AWS
  - Scalability- 5/5
    - AWS offers a range of scalable database solutions, including Amazon RDS and Amazon DynamoDB, that can handle massive amounts of data and requests.
  - Cost- 4/5
    - AWS offers flexible payment options, including pay-as-you-go and reserved capacity, making it a cost-effective solution for many organizations.
  - Ease of Use- 4/5
    - AWS has user-friendly tools like the Amazon RDS Console and Amazon DynamoDB Console, as well as comprehensive documentation and support.
  - Security- 5/5

- AWS provides strong security features, including encryption at rest and in transit, IAM roles and permissions, and advanced security tools like AWS CloudTrail and AWS GuardDuty.
- Google Cloud Platform
  - Scalability- 4.5/5
    - Google Cloud Platform provides horizontal scaling with Cloud Spanner and Cloud Datastore, making it highly scalable.
  - Cost- 4/5
    - Pricing is generally competitive, and there are a range of pricing models to choose from, including pay-as-you-go and committed use discounts.
  - Ease of Use- 4/5
    - Google Cloud Platform offers a user-friendly interface with tools like the Google Cloud Console and Datastore API that simplify database management and development.
  - Security- 4.5/5
    - Google Cloud Platform provides a range of security features, including encryption, IAM roles and permissions, and security logging and monitoring.

### 3.1.4: Analysis Table

Figure 1 contains the overall scores from each hosting service, based on the criteria they have been judged by.

	Azure Cosmos DB	AWS	Google Cloud Platform
Scalability	4.5	5	4.5
Cost	3.5	4	4
Ease of Use	3	4	5
Security	4.5	5	4.5
Total	15.5/20	18/20	17/20

*Figure 1: Backend Host Programming Language Table*



Based on these ratings, AWS is the top performer, with a total score of 18/20. It offers the best combination of scalability, cost-effectiveness, ease of use, and security per the requirements.

### **3.1.5: Proving Feasibility**

Based on the requirements for storing large amounts of data, scalability, global availability, and compatibility with MySQL, this table recommends using AWS for our cloud hosting needs.

AWS offers a range of scalable database solutions, including Amazon RDS and Amazon DynamoDB, that can handle massive amounts of data and requests. Additionally, AWS has a global infrastructure that can ensure reliable performance and availability around the world, including in China. AWS also offers compatibility with MySQL, making it easy to migrate existing data and applications.

Overall, AWS provides a combination of features that align with our needs and can help ensure reliable and cost-effective performance as it scales and expands.

## **3.2: Website Programming Language**

A main component of our capstone project revolves around a website. The programming language used must contain specific features to allow the site to both connect to other code components and provide necessary features to suit our programming desires. Prior understanding of each language is also important, as being able to write code sooner is beneficial for our project development timeframe.

### **3.2.1: Criteria Descriptions**

- Familiarity
  - Less time spent figuring out how to code in a specific language means more time spent programming overall. How well our team already understands chosen languages is important to identify, as our prior understanding of the language will boost our productivity throughout the project.
- Compatibility
  - Other components outside of the website can be coded in a variety of possible languages, and whether or not the website programming language can support sending data to such components is necessary to identify. The

less compatibility issues with the website programming language, the easier it will be to send important information between the site to other components.

- Features
  - Programming problems can be solved in multiple ways, and how many ways each problem can be solved is based on the features the language has. A programming language with more features means more options to solve programming problems, so languages with more features are more favorable. In addition, our site requires key components to function properly, so identifying what a language is capable of in this aspect is necessary.

### 3.2.2: Proposed Technologies

- HTML
  - HTML is the most common programming language when it comes to website development. Our first introduction to the programming language came from various sources, whether it came from a website programming course we've already taken, or from our own programming endeavors outside of academia. Created by Tim Berners-Lee in 1993, HTML has been the dominant website programming platform ever since. Considering how often HTML is used in most websites, and how familiar it is to our group overall, HTML is a viable candidate for website development in our case.
- React (Javascript)
  - React is a JavaScript-based programming language library designed by Jordan Walke in 2010. The library was not available until 2013 however, since Walke designed the language library to incorporate Facebook ads, and Mark Zuckerberg (CEO of Facebook) had then decided to make the library open to the public. Our team's first contact with React came from our own research in possible website programming languages. Considering the low learning curve, declarative nature allowing for easier implementation of interfaces, and syntax facilitation, React is a suitable contender for our website programming needs.
- PHP
  - PHP was designed by Rasmus Lerdorf in 1994, designed towards a more general purpose scope geared towards website development. This language has seen some use in our team initially, in addition to further research

conducted to determine candidacy as our website programming language. PHP is HTML based, which is beneficial considering the common usage of HTML throughout most websites. A key factor of PHP involves code execution occurring server side, where the generated HTML is then run client side. Malicious users would have a difficult time determining the content of the code we write due to this factor, decreasing the possibility for online threats to the site as a whole.

### 3.2.3: Analysis

- HTML
  - Familiarity - 5/5
    - All members of the team have a firm grasp on HTML as a whole. We have all coded in it before, and can easily start website development with little research throughout the programming process.
    - Most sites use some amount of HTML, so any research required for website development would be quick and effective. Any possible desires for the site and how to accomplish them can be easily found and executed.
  - Compatibility - 4/5
    - Programming languages used for other technologies in our project might have compatibility issues if our site is HTML based. While HTML is not completely compatible with other languages on its own, HTML does allow for specific frameworks to be established to mediate this problem.
    - While using a framework would aid HTML compatibility, having to use a framework at all appears to be an extra step that could be completely avoided through using other languages.
  - Features - 3/5
    - HTML is unique in that it is forgiving when it comes to programming mistakes. This could be viewed as a positive, until seemingly perfect code breaks the site due to some unknown and unlisted error. Writing one section of the site with a mistake could break other sections, resulting in the tedious process of reading through everything to find what's broken.

- Broken yet still functional code can be remedied through third party applications, but this is also an additional step that can easily be circumvented through usage of a different language in the first place.
- HTML is simplistic in nature, so more convoluted programming solutions could result in obscenely long stretches of code. More code to write for a task hurts overall productivity.
- React (Javascript)
  - Familiarity - 3/5
    - While React is Javascript based, a language our team is already familiar with, none of our team has ever really used the library in any significant manner. A fair amount of research into how to code in this language would need to be conducted before any real progress can be made.
    - Our research describes a low learning curve for this language library. How to accomplish more involved programming practices, such as data transference from the site to outer applications for example, might be less applicable to this sentiment. Surface level programming endeavors might be easily understood, but the more niche applications that require more nuance would still take more time.
  - Compatibility - 4/5
    - React is relatively lenient when it comes to the programming language the backend host is coded in. Data sent from the server can easily be interpreted and redefined for usage in the site when necessary.
    - Connections between a React website and other outer components would require additional code that's unnecessary when considering other programming options. While it is possible to develop such connections, it would require more code overall compared to the other website programming language options.
  - Features - 5/5
    - React as a library is tailored towards web application development. Considering the scope of the project as a whole, React is designed to work with exactly what we intend to develop as a large scale web application.

- There are extensions that can improve programming with the React library as a whole. There is a Google Chrome extension that can better aid with React debugging, for example, that would make figuring out problems in our code easier when compared to other programming language possibilities.
- PHP
  - Familiarity - 4/5
    - PHP is HTML based, and HTML is well understood by our team. The syntax remains more or less the same, so PHP should be fairly simple to implement into our website without need of too much research alongside development.
  - Compatibility - 4/5
    - A PHP based website can obtain data from outside programs, but how well this process is done is based on implementation. This means that the possibility is there, but additional code would need to be developed accurately to maintain effective data transfer.
    - Connecting Python with PHP, for example, would work best when considering the two as “two separate languages.” This involves designing a more generalized API for data transfer. Determining transferred data would then need to be done within the site itself. This means that data can be sent between technologies fairly well, but would require more work to accurately identify what was being sent in the first place.
    - PHP supports database management, meaning that adding and removing data from the backend host would be a seamless process.
  - Features - 5/5
    - PHP allows for integration with multiple programming languages, such as HTML, XMP, and Javascript.
    - PHP offers error reporting, better allowing our team to know what’s broken in the site throughout development.
    - Websites coded in PHP load quickly, as PHP is a high performance programming language. Longer stretches of code requiring higher complexity would load faster when compared to other languages we have researched.

- PHP frameworks are available to make programming in PHP easier. Overall website maintenance is better achieved through the possible frameworks PHP is compatible with.

### 3.2.4: Analysis Table

Figure 2 contains the overall scores from each language option, based on the criteria they have been judged by.

	HTML	React	PHP
Familiarity	5	3	4
Compatibility	4	4	4
Features	3	5	5
Total	12/15	12/15	13/15

*Figure 2: Website Programming Language Table*

HTML scores high in most categories other than the features HTML has for website code development. React scored lower in familiarity, where our team would require more research to get started with React. PHP scored well in all categories, where our team understands the language as a whole, recognizes how well a PHP site can connect with outer programs, and the presence of multiple readily available features for our team to use.

### 3.2.5: Proving Feasibility

PHP has proven to work best as our website programming language, based on what our team both expects and needs the site to accomplish. General website development, as well as graphic design, will be developed through basic means until we have a solid base to work with. We would then test the modification, sending, and receiving of data from the backend host database, to determine what measures need to be taken to store important data in a secure manner.

### 3.3: Game Application Programming Language

One of the linchpins of our capstone project is the gamified aspect of the software. It is an aspect that we are excited to work on, and that we have a lot of ideas for. In order to implement this we must choose a programming language that will give us access to tools that will make creating games easier, as well as offer compatibility with the other technologies that we selected. This is a large part of the final product, so it is important that whatever we choose is efficient and easy to work with.

#### 3.3.1: Criteria Descriptions

- Familiarity
  - Familiarity of the whole team with the language will gain us a lot of time since we will not have to learn something entirely new. A deeper knowledge of a language also has the potential advantage of allowing us to solve problems easier and produce better end results. It would allow for better distribution of the workload, because we all would share a common understanding of what others have written, and pick up where they left off.
- Available Resources
  - Many programming languages include extensive libraries or predefined functionalities that can make the programming process much smoother. The more of these resources that we can take advantage of, the less time we will have to spend writing code to support functions, and instead be able to work on and add new ones. This will give us a game with a bigger variety of interesting features.
- Versatility/Flexibility
  - Gamified features often contain a multitude of components that our chosen language needs to be able to handle. It should be flexible enough to allow us to implement all the features that we really want to have without too much hassle. Whether that comes from prebuilt libraries and tools or the higher-level functions of the language itself, it's important that rigidity does not become a major roadblock for us during development, as we will not easily be able to switch to a different language.
- Compatibility
  - Crucially, whatever programming language we choose to write the game portion of our application in, it needs to be compatible with the other technology that we want to use. Our games are useless if we can't insert

them into the website itself, or if they're unable to communicate effectively with the database. Seamless integration would be the ideal scenario, but since that is not always possible then we will aim for a solution with the least number of potential complications.

### 3.3.2: Proposed Technologies

- Python
  - Python is a high-level, interpreted programming language, known for its simplicity, readability, and versatility. First released in 1991, it has since become known as one of the most popular programming languages in the world. It is designed to be easy to learn and use, making it a popular choice for beginning programmers, but also offers a lot of flexibility and functionality for more advanced users who are able to take advantage of it. It also features a large and supportive online community, with many resources available for learning and troubleshooting, as well as a rich library of modules and tools. It is commonly used in web development, game development, data analysis, and visualization. Python's ease of use and high versatility make it a great choice for a variety of programming projects.
- Javascript
  - JavaScript is yet another high-level, interpreted programming language, commonly used for creating dynamic web content and applications. JavaScript is primarily used for client-side scripting. It is a great tool for creating interactive features on websites. JavaScript can also be used for server-side programming using platforms such as Node.js. JavaScript has a syntax like C and Java, making it relatively easy to learn for those who are already familiar with those languages. JavaScript is one of the go to choices for web development, but also is versatile enough to be used to complete a wide range of tasks.
- C
  - C is a general-purpose, low-level, procedural programming language. Because of its low level it provides a lot of control over the computer's hardware and memory, and it is especially well known for its efficiency, speed, and ability to handle low-level tasks, due in part to the fact that it is compiled into machine code. As stated above, C has a syntax similar to that of Java, but it has fewer high-level features and requires much more manual memory management. This simplicity gives C an advantage when it comes



to programming things such as operating systems, device drivers, and embedded systems. It is also widely used in the development of applications that require high performance, such as video games and image processing. C also has a large community like Python, who have provided a multitude of libraries and tools to make development easier. Where C lacks in ease of use, it makes up for in performance and control.

### 3.3.3: Analysis

- Python
  - Familiarity - 5/5
    - Our entire team has worked with Python before, and has a strong grasp of its concepts and utilities.
    - Python is easy to learn and use, so any gaps in knowledge can be quickly filled if need be.
  - Available Resources - 5/5
    - Python has a large online community that has created a vast number of tools and libraries that are freely available, such as TensorFlow, Numpy, Keras, PyTorch, etc. All are well documented and easy to integrate into projects. This would boost our team's productivity immensely.
  - Versatility/Flexibility- 4/5
    - Although like most languages it excels at certain tasks, Python is a highly versatile language that allows for a multitude of different applications. Python is useful for things such as automation, scripting, data analysis, and machine learning. Unfortunately, while it is still capable, it is not the strongest of the options that we have selected for web development purposes.
  - Compatibility - 3/5
    - Python is not only available on a wide variety of platforms, but is also compatible with all kinds of technology, such as machine learning libraries, database technologies, and web development frameworks. Python supports procedure-oriented as well as object-oriented programming. Again however, Python is a bit more difficult to integrate with HTML than our other options.
- JavaScript
  - Familiarity - 4/5

- Some, but not all team members have used JavaScript in the past. We all share a good working knowledge of Java, which is similar in some aspects.
  - Available Resources - 5/5
    - JavaScript, like Python, has a large online community that has come together to create a huge kit of tools and libraries that are oftentimes freely available to developers. The majority are well documented and could prove to be very useful.
  - Versatility/Flexibility - 5/5
    - JavaScript's main strength lies in web development, but it can also easily be used for other purposes such as game development, desktop app development, and even server-side development.
  - Compatibility - 5/5
    - JavaScript is often used together with HTML and CSS to create dynamic and interactive web pages. JavaScript also works well with JSON, as well as various web API's. JavaScript has many tutorials for integrating databases that use SQL and AWS.
- C
  - Familiarity - 2/5
    - Every member of our team has some experience in C programming, but it is not any of our strong suits. Of the languages discussed so far this one is the one that we are least familiar with.
  - Available Resources - 3/5
    - C gives us access to many libraries that include some very essential services, such as input/output operations, string handling, and memory management. While there are several tools available, they are often more barebones and more difficult to implement than libraries from languages such as python.
  - Versatility/Flexibility - 4/5
    - C is a low-level language, giving us a lot of control over the specifics of what we program. This level of depth can be helpful, but also may hurt us if we spend too much time coding the small details. While it is nice to have a stronger control over the finer details of our system, it will probably not be necessary for our purposes.
  - Compatibility - 4/5

- C is compatible with a wide range of operating systems, including Linux, Windows, and macOS. It can be used for database management, game development, web development, and even machine learning.

### 3.3.4: Analysis Table

Figure 3 contains the results of scoring each language option, based on the criteria outlined above. Each language's score is out of a total of 5 for each category, giving them a total out of 20.

	Python	JavaScript	C
Familiarity	5	4	2
Available Resources	5	5	3
Versatility / Flexibility	4	5	4
Compatibility	3	5	4
Total	17/20	19/20	13/20

*Figure 3: Game Application Programming Language Table*

Python and JavaScript both scored very high in all categories. JavaScript emerged as the winner based on score, with Python's comparatively worse compatibility with the other technology that we have chosen being a deciding factor. The familiarity criteria is what hurt C's score the most, as well as a lower rating in available resources, as the libraries are relatively harder to implement and less expansive than the other languages.

### 3.3.5: Proving Feasibility

We have chosen to use JavaScript because we believe that it will be the smoothest and most straightforward programming experience. It is highly compatible with HTML and PHP which we believe will make the integration process with the website smoother and this combined with the expansive libraries and toolkits available should make the development process much more productive. As a team we do not anticipate having to

spend too much time familiarizing ourselves with it, as we each have a little bit of experience in similar languages. Overall, JavaScript seems like the best choice for our purposes.

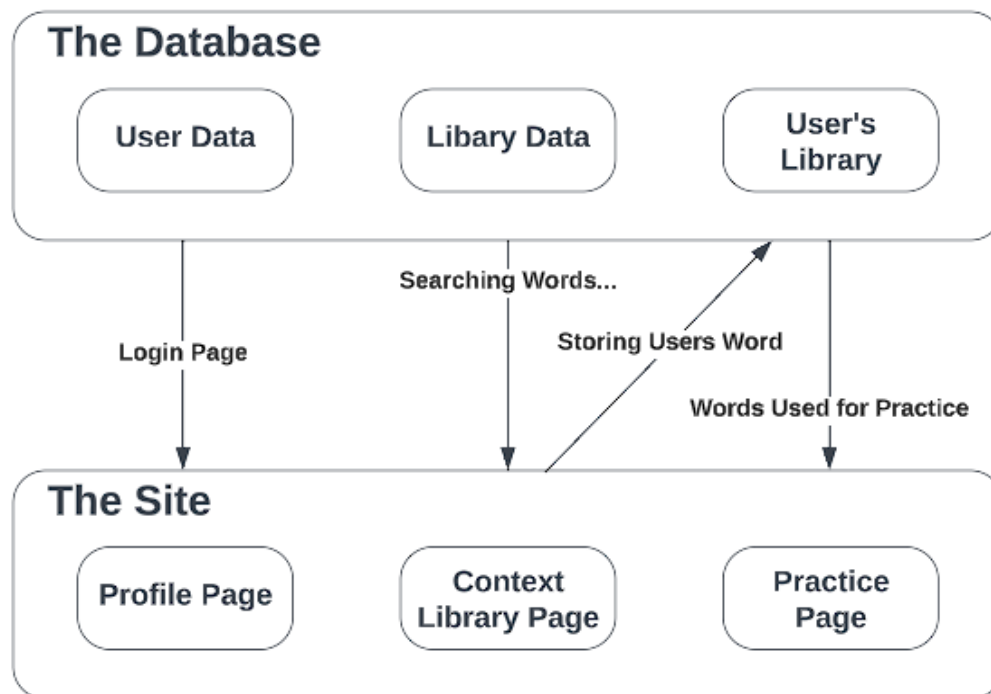
## **4. Technological Integration**

With all of the languages and sources decided, this will all eventually become a fun language learning website for english learners. This together should hopefully create a seamless experience for users as they come back to practice, or just to learn about words they may have heard out in the real world. The game elements and forums allow the user to communicate with others about their English learning experience. We've broken it into parts, but what truly matters is how everything is going to end up when put together.

### **4.1: The Connected Product**

As you'll see in the diagram below, there is a close relation to our database and the website itself. With things like the speech-to-text program pulling from the database, and transferring data back to that database based on the results. Another thing is the game pulling from that database and constantly updating itself. As the user interacts all of the different parts of the site will be fully initialized, with transfers from the database and the website keeping track of multiple subsets of data to help enhance the overall experience of the user.

Figure 5 shows how the systems interact with one another, showing all of the pulls done from the user, to the website, and the database. Each connection is vital to the overall experience of the application, and theoretically this system should provide an engaging environment to any user.



*Figure 5: The Integrated System Diagram*

## 5. Conclusion

We all know Duolingo is a language learning platform that exists, it's gamified, but has proven time and time again, vocabulary will never help you truly understand the fundamentals of a language. Spoken language and context can change the use of vocabulary and that one thing is what our Website will solve. Students across the world coming to study here in the US could learn context and pronunciation months in advance, and continue to use it as new words come up in conversation.

By using a mix of SQL, PHP and Javascript, our goal is to create an active environment for these students to help build upon their knowledge of the English language. With a focus on self discovery and self reflection, the website will be a tool for research in discovering more into the language learning spectrum. With our tools tested and ready to use, we are ready to develop this evolved language learning program.