ASL Search IW Report

(aslsearch.herokuapp.com)

1. Background

1.1 Introduction

American Sign Language, or ASL, is a visual language expressed through facial and hand movements, the primary language of many deaf and hard-of-hearing North Americans.[1] I was first exposed to ASL when I took the class LIN 205: American Deaf Culture in the Fall of 2020. The class was taught by Professor Noah Buchholz from the linguistics department, who is pioneering one of the first comprehensive university ASL programs here at Princeton. Most of the curriculum we used in LIN 205 was developed by Professor Buchholz himself, who is even in the process of writing his own textbook. This is because it was often hard to rely on outside resources for ASL instruction. As students, if we wanted to learn additional ASL vocabulary for something like a presentation, one online ASL dictionary might suggest one ASL sign for a given English word, while another dictionary suggests a completely different sign. As a result, would usually end up checking new signs I found online with Professor Buchholz before using them.

1.2 Motivation

The problem with existing ASL material comes down to variation. Only 10% of Deaf people are Native users of ASL, meaning they were taught ASL formally as a first language. However, 90% of Deaf people are born to hearing families, and thus take many paths in finding their way to ASL, whether they first learn phrases through friends, online sources, caretakers,

[1] National Institute on Deafness and Other Communication Disorders

etc. Online dictionaries thus contain many variations of signs for a given English word, often without providing sufficient context for their usage, leaving students unsure as to which sign they should use.

My app, ASL Search, remedies this issue for Professor Buchholz by providing a custom platform for the Princeton ASL program that allows him to upload his own ASL content in the form of YouTube videos, as well as accompanying context to guide his students to the correct sign for a specific scenario. The platform enables Professor Buchholz to create his own online English to ASL dictionary to accompany the textbook he is creating, giving him more ASL content he can feel confident referring his students to. The app thus consists of a user interface for students to look up English words and find their ASL translations, as well as an administrator interface for Professor Buchholz and his colleagues to manage content.

1.3 Related Work

Here is an overview of the features that existing online ASL dictionaries have and lack:

Dictionary Name	**ASL Gloss**	**Part of Speech**	**Additional Information**	**Advanced Searching**	**Autoplay / Looping**
HandSpeak[2]	✓	X	✓	X	X
Signing Savvy[3]*	✓	X	✓	✓	X
SignASL[4]	✓	X	X	X	X
Lifeprint[5]	✓	X	✓	X	X

[2] https://www.handspeak.com/
[3] https://www.signingsavvy.com/
[4] https://www.signasl.org/
[5] https://www.lifeprint.com/

Spread the Sign[6]	X	X	X	X	✓
ASLsearch	✓	✓	✓	Coming Soon!	Coming Soon!

*includes more features with a paid account

1.3.1 Contextual Information

- **ASL Gloss:** the syntax for writing ASL signs in English, capturing its grammar.
  - Ex: DOG, MIND-OPEN, TO-LEARN++, OPEN-DOOR
- **Part of Speech**: essential for unique ASL grammar that doesn't exist in English.
  - Ex: noun, adjective, progressive verb/gerund, noun-verb pairing
- **Additional Information**: can include example sentences, regional variants, context, etc.
  - Ex: HE MIND-OPEN. Frequently used in Philadelphia. Raise eyebrows as well.

As you can see from the table, existing online dictionaries vary in the amount and types of contextual information included. ASL search currently remedies these issues by organizing the page of an individual word such that students can easily discern the appropriate sign for a given situation using a combination of the contextual clues listed above. ASL gloss is something any student of ASL will be familiar with and can use to choose the proper sign for an English word, since this relationship is not always 1-1. For example, the sign for THANK-YOU is a single sign, while the English phrase "thank you" is two words. Part of speech also varies greatly between ASL and English, for example, in addition to nouns and verbs ASL grammar also includes noun-verb pairings, which are signs that can indicate nouns or verbs based on their repetition. The sign for the verb OPEN is also the sign for the verb OPEN-DOOR, which depicts opening a door. However, if one depicts closing the door again, this becomes the sign for the

[6] https://www.spreadthesign.com/en.us/search/

noun “DOOR.” Finally, additional information can communicate anything from when a sign is primarily used in a certain region (many unique signs originate from Philadelphia, for example) to instructions such as raising your eyebrows when asking a question (facial expressions are considered a “non-manual marker” of sign language).

1.3.2 Advanced Technological Capabilities

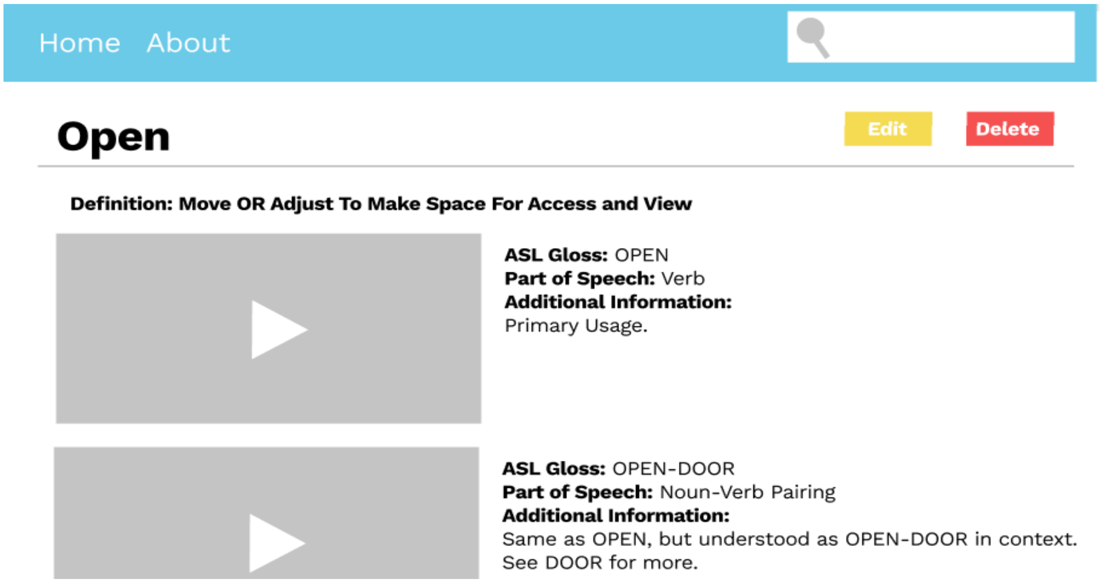
- **Advanced searching**: searching that rivals online English dictionaries
  - Ex: show results in a dropdown, ordered not only alphabetically but by frequency of lookup, ability to search for similar words, ability to search for ASL gloss in addition to English word
- **Video features**: allows for increased ease of viewing signed videos
  - Ex: videos start automatically on screen focus (autoplay), video loops repeatedly instead of suggesting more videos, remove YouTube header from videos

In addition to contextual information, existing online dictionaries also contain different levels of technological capabilities. While most online English dictionaries contain all of these features, the lack of technological resources dedicated to ASL materials creates a gap in the quality of online English to ASL dictionaries. For example, when searching for a word, results should be displayed in a dropdown rather than on a separate page. These results should then be ordered not just alphabetically, but by a lookup frequency metric, in order to suggest more commonly searched-for words first. Also, users should not only be able to look up the exact English word or a part of it (like “op” or “open”) but also other similar forms (like “to open” or “opens”). Finally, it would be helpful for signed videos to autoplay on focus, so the user does not have to press play on the desired video, and for videos to loop repeatedly instead of stopping after playing once, since they are often very short and need to be viewed more than once. While ASL search currently remedies the contextual deficiencies of existing dictionaries, it will also soon address these advanced technological features as well.

2. Approach

2.1 Gathering Requirements

I first had to understand the relationship between ASL and English, which is not 1-1 as displayed in Section 1.3.1, and determine how to capture the complex grammar of ASL in my app. I then started drawing up Figma prototypes to design the page for a single word, meeting with Professor Buchholz regularly to iterate on my designs and settling on the layout below.



Open

Definition: Move OR Adjust To Make Space For Access and View

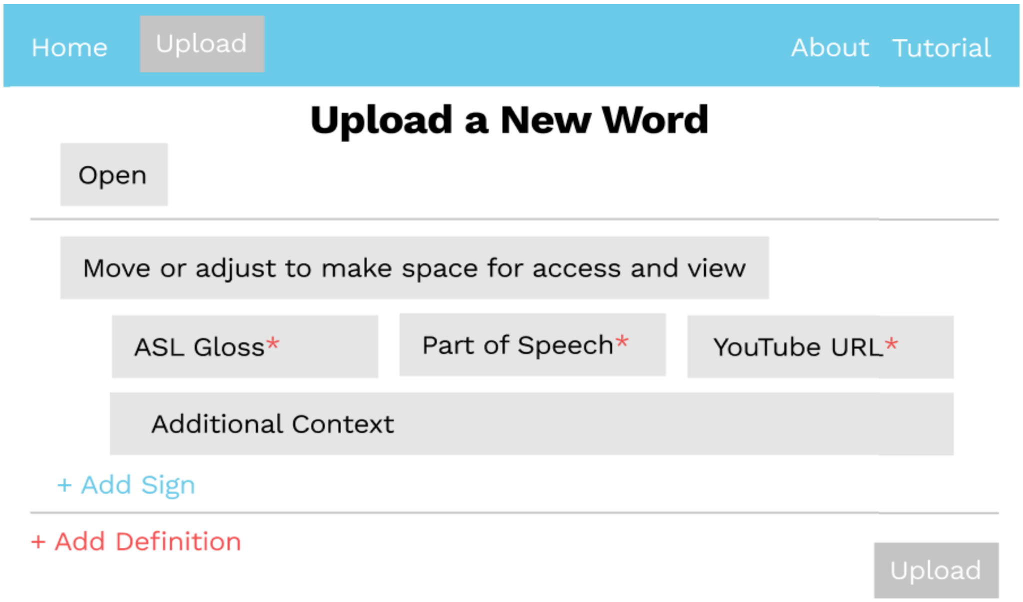
ASL Gloss: OPEN
Part of Speech: Verb
Additional Information:
Primary Usage.

ASL Gloss: OPEN-DOOR
Part of Speech: Noun-Verb Pairing
Additional Information:
Same as OPEN, but understood as OPEN-DOOR in context.
See DOOR for more.

Further notes from these meetings are contained in Section 7.1 of the Appendix. I finalized a layout representing the one-to-many relationships between English words and their definitions, and between English definitions and ASL signs. Each individual English word page is sectioned by its definitions, which can each contain one or more signs consisting of a YouTube video and accompanying contextual information that would be helpful for students. This format helps incorporate complex grammar like noun-verb pairings mentioned in Section 1.3.1, as ASL Gloss and Part of Speech provide key information while the Additional Information section allows Professor Buchholz to add further details for clarification.

2.2 Implementation

Then, while building the app, I had to continue changing this design as I understood the constraints of what would be too complex to implement. For example, I initially wanted to make the upload process a single page consisting of fields for English words, definitions, and ASL signs all on one page.



The page would include Add Definition and Add Sign buttons that trigger expandable input forms to handle any number of inputs. I realized this would not only be more difficult to implement, taking up important time during the development process, but also more confusing to the user, as Professor Buchholz needed further explanation to understand how the form would expand/collapse when I showed him this design. This made me realize a multi-step process would be both easier to implement and more user-friendly, resulting in the final upload design seen in Section 3.2.3. I continually tested my app with Professor Buchholz and a few of his students to iterate on my implementation and eliminate points of confusion.

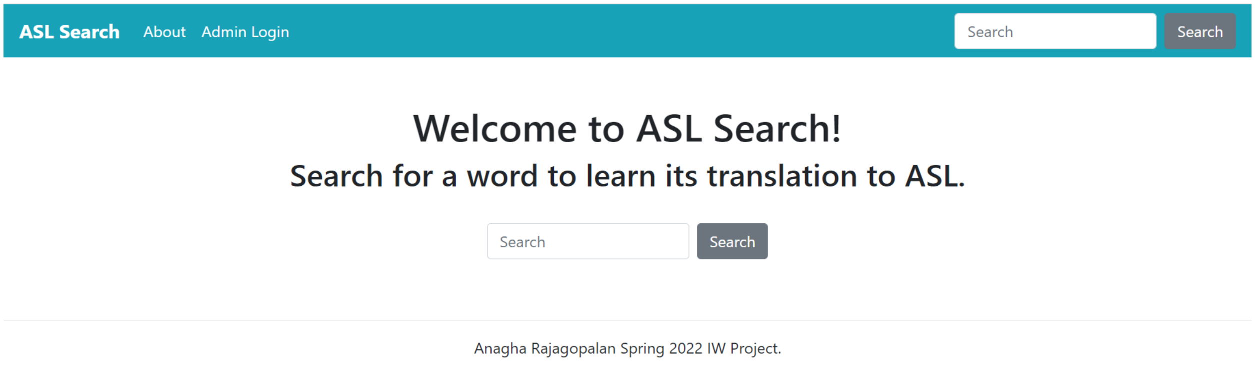
3. Feature Overview

To follow along, navigate to the app on any web browser at aslsearch.herokuapp.com.

3.1 User Interface

3.1.1 Pages

The user interface consists of the Homepage, About page, Search Results, and individual Word Pages. All pages contain a navigation bar with links to the Hompage, About page, and Admin Login, as well as a footer statement. The Homepage and navbar both contain search bars while the about page describes the problem ASL Search is solving.



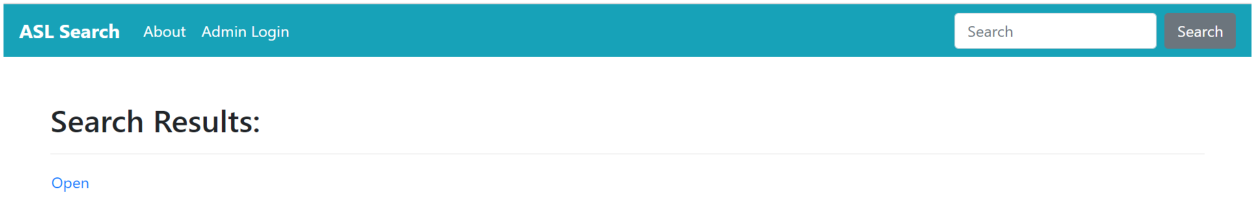
Welcome to ASL Search!

Search for a word to learn its translation to ASL.

Anagha Rajagopalan Spring 2022 IW Project.

3.1.2 Searching

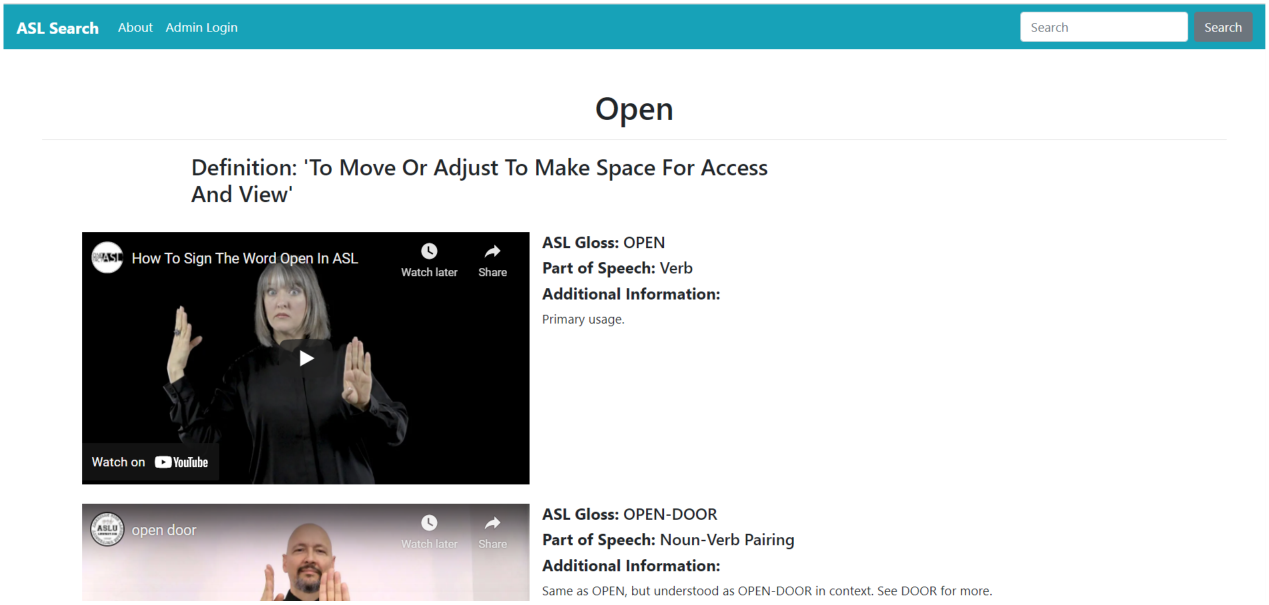
Users can search for an English word like “Open” from either the navbar or the Homepage, then click on it in the Search Results to navigate to its individual Word Page, where they can find its definitions and signed ASL videos.



Search Results:

Open

As described in Section 2.1, each English word can have multiple definitions and multiple associated signs. The user can discern which ASL sign is useful to them by looking at the information on the right, then play the video on the left to see the sign in action. The app currently contains random videos found on YouTube, but eventually Professor Buchholz his own video content as he produces it for his upcoming textbook. You may notice the implemented Word Page matches the final Figma design closely and did not undergo major changes.



Open

Definition: 'To Move Or Adjust To Make Space For Access And View'

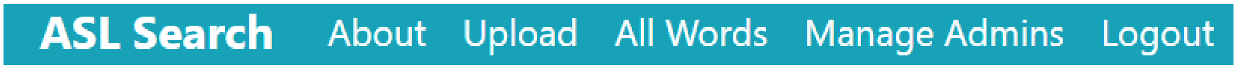
ASL Gloss: OPEN
Part of Speech: Verb
Additional Information:
Primary usage.

ASL Gloss: OPEN-DOOR
Part of Speech: Noun-Verb Pairing
Additional Information:
Same as OPEN, but understood as OPEN-DOOR in context. See DOOR for more.

3.2 Administrator Interface

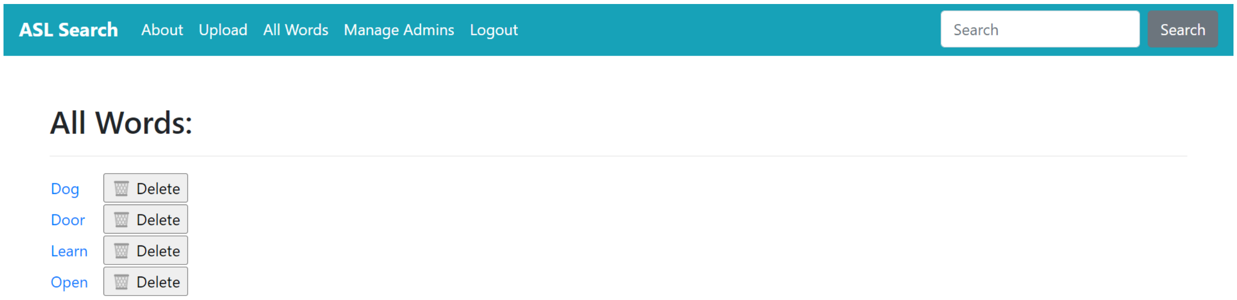
3.2.1 Login/Logout with CAS Authentication

The “Admin Login” link in the navigation bar redirects the user to the CAS authentication system for Princeton students and staff. While anyone with a Princeton netid may CAS authenticate, non-administrators will be directed to a rejection page and prompted to return to the general user interface Homepage. For netids granted administrator privileges, logging in will redirect to the admin interface Hompage and reveal new links in the navigation bar in place of Admin Login: Upload, All Words, Manage Admins, and Logout. Clicking “Logout” will remove the admin’s ticket from the session and redirect them to the Homepage.



3.2.2 Managing Dictionary Content

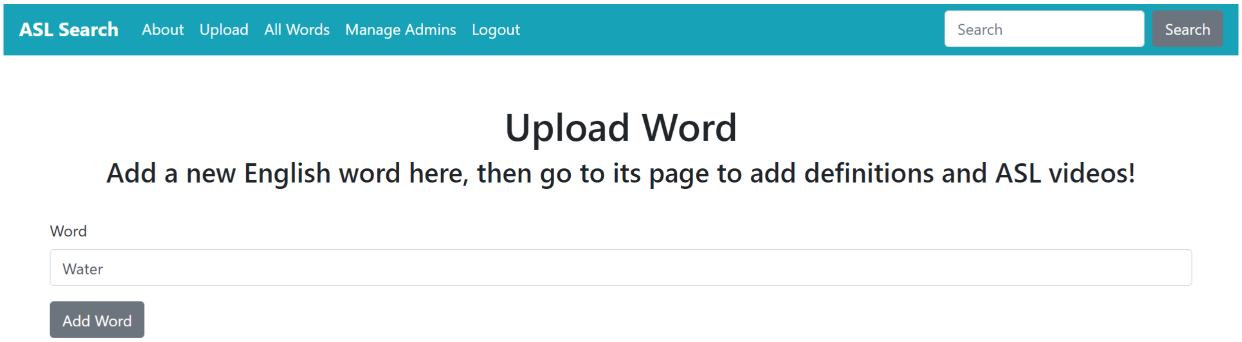
The All Words page displays the English words that have been added to the database so far, from where admins may delete words.



All Words:

Dog	Delete
Door	Delete
Learn	Delete
Open	Delete

To add a new word, admins can click on Upload and enter an English word. The field validates that only English words are being added, preventing against invalid entries, cross-site scripting, etc.

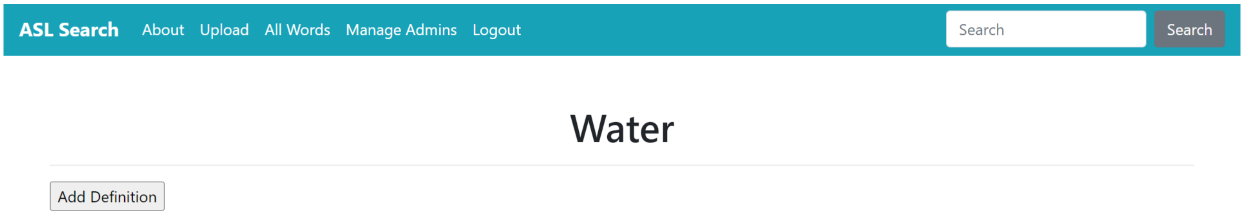


Upload Word

Add a new English word here, then go to its page to add definitions and ASL videos!

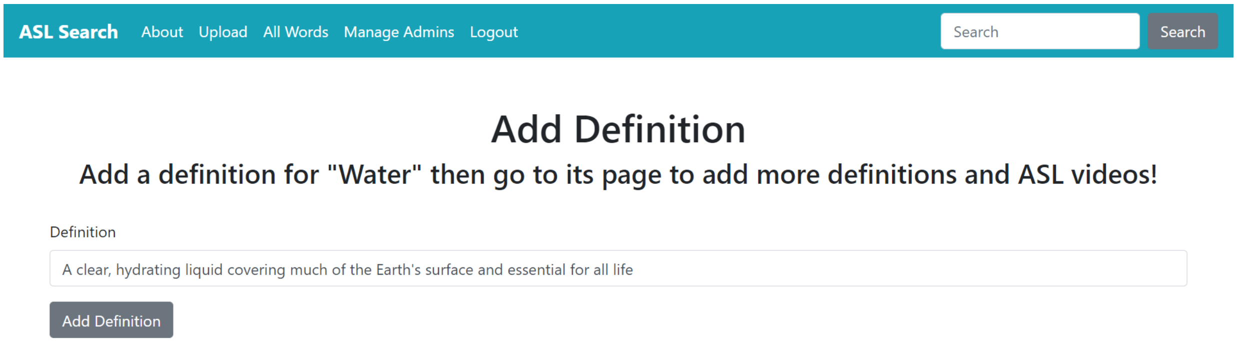
Word: Water

If an administrator tries to add a word that already exists, they will be redirected to the Upload page to add a different word. Otherwise, upon successfully submitting, they will be redirected to the new Word Page that was created for the provided word.



Water

There, the administrator can add a definition for the word by clicking on “Add Definition” and filling out the field on the next page.



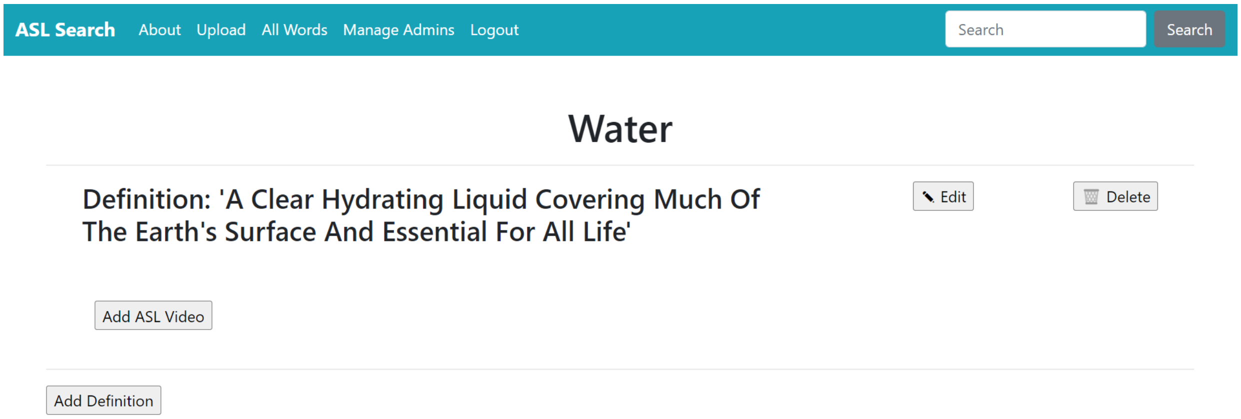
Add Definition

Add a definition for "Water" then go to its page to add more definitions and ASL videos!

Definition

A clear, hydrating liquid covering much of the Earth's surface and essential for all life

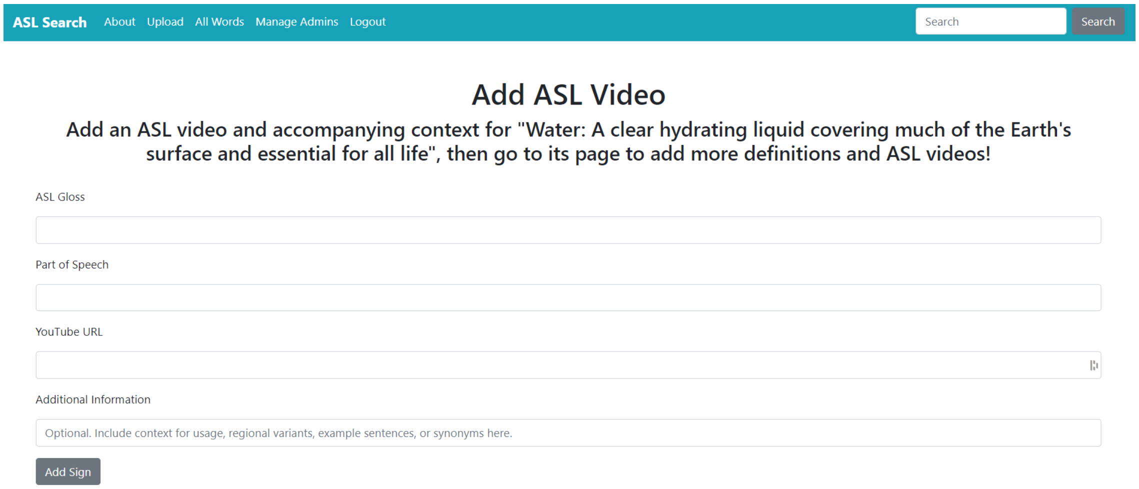
Adding a definition will cause Edit and Delete buttons to appear, allowing admins to modify or remove the definition that was just added. It also prompts a new “Add ASL Video” button, using which admins can add corresponding signs for the definition. The admin also still has the option to add another definition, which will create a new section that can then contain its own ASL signs.



Water

Definition: 'A Clear Hydrating Liquid Covering Much Of The Earth's Surface And Essential For All Life'

Clicking “Add ASL Video” takes the admin to a form where they can add YouTube videos of signs with accompanying context.



Add ASL Video

Add an ASL video and accompanying context for "Water: A clear hydrating liquid covering much of the Earth's surface and essential for all life", then go to its page to add more definitions and ASL videos!

ASL Gloss

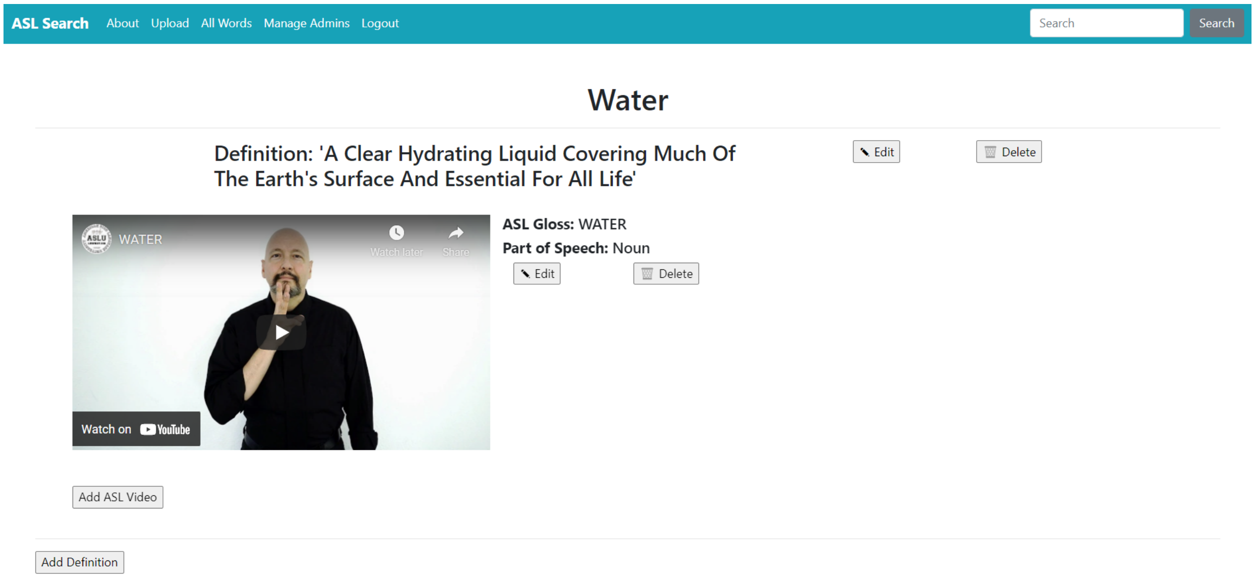
Part of Speech

YouTube URL

Additional Information

Optional. Include context for usage, regional variants, example sentences, or synonyms here.

All of the input fields in the upload process, from the English word to the definition to the information for the ASL sign, are required except for the optional Additional Information field, which is indicated by the placeholder. Furthermore, the Youtube URL field validates that a YouTube link is being added by checking it against a regex expression. The final result is a word page that looks similar to the general user view, but for Edit, Delete, Add Definition, and Add ASL Video buttons.



Water

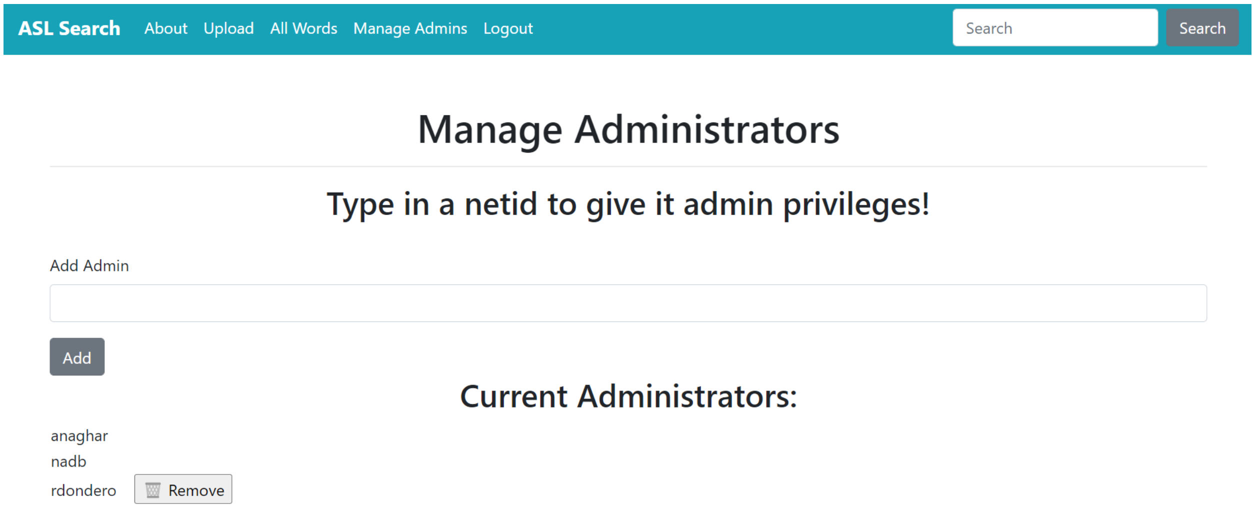
Definition: 'A Clear Hydrating Liquid Covering Much Of The Earth's Surface And Essential For All Life'

ASL Gloss: WATER

Part of Speech: Noun

3.2.3 Managing Administrators

Finally, administrators can also view, add, and remove other administrators with the Manage Admins tab, excluding "superadmins" Professor Buchholz and myself. Adding an existing netid will redirect, but otherwise, the Add button will grant admin privileges to the supplied netid.



Manage Administrators

Type in a netid to give it admin privileges!

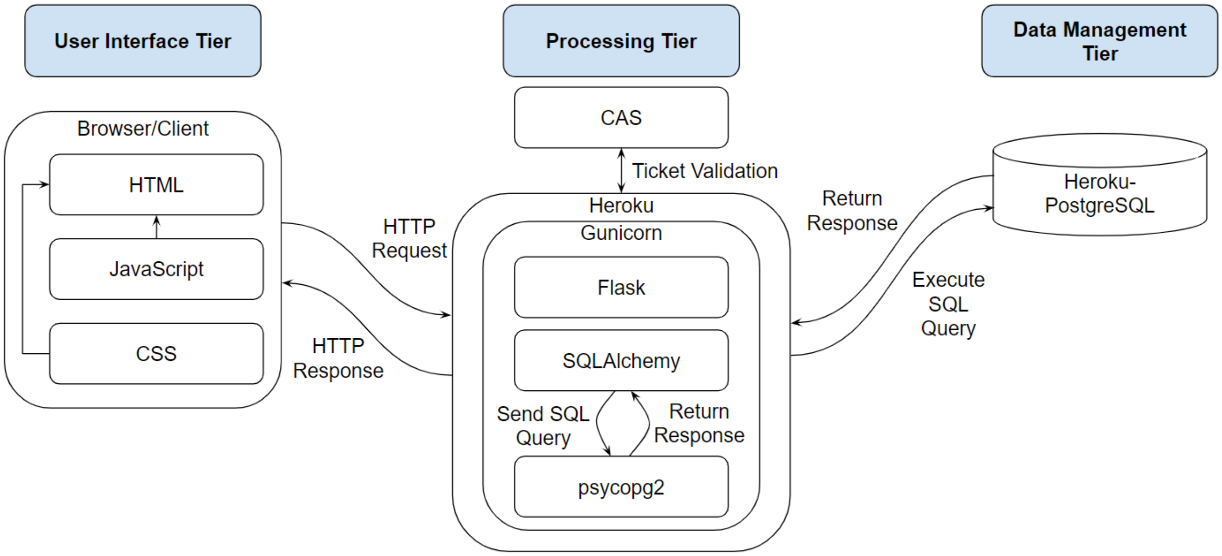
Add Admin

Current Administrators:

anaghar
nadb
rdondero

4. Implementation

4.1 Three-Tier Architecture



In the user interface tier, the user interacts with the web application by accessing aslsearch.herkouapp.com through a web browser. Navigating to different pages of the app sends HTTP requests to the Heroku web server: GET requests for obtaining data from the processing tier and returning HTTP responses rendering combinations of HTML, CSS, and/or Javascript, as well as POST requests for sending data to the processing tier and updating the data management tier. Heroku is the cloud platform hosting the web app in the processing tier, running a Gunicorn process as the WSGI HTTP server. Flask is the server-side web framework the app is implemented in, with Central Authentication (CAS) ticket validation used for administrator authentication. SQLAlchemy is the object-relational mapper used to interact with the data management tier with Python code, facilitated by the psycopg2 database driver. SQLAlchemy was used to create the database models/schema for the Heroku-PostgreSQL database and to update the database.

4.2 User Interface and Processing Tier - Templates and Routes

- **/:** used to generate the Homepage with the templates base.html and index.html
- **/about:** used to generate the About page with the templates base.html and about.html
- **/admins:** used to view and add administrators with the templates base.html and admins.html, only accessible by administrators
- **/<int:adminid>/delete:** used to remove the provided administrator
- **/words:** used to query the Words table for a match with user input from the search bar and displays the results with the templates base.html and words.html, administrators can additionally delete words from this route, also used by the All Words link to display all of the English words added to the Words table so far
- **/wordpage/<string:title>:** used to display the individual word page for the provided English word with the templates base.html and wordpage.html, displaying its definitions and signs by querying the Words, Defs, and Signs tables, administrators can additionally edit and delete definitions and signs from this route
- **/uploadword:** used to add an English word with the templates base.html and uploadword.html, generating an upload form that adds a row to the Words database with the user input on successful submission (unless the user submits a non-English word or a word that already exists) and redirects to the newly created word page

➢ **/<string:word>/delete:** used to delete the provided English word, i.e. removes the corresponding row in the Words table as well as all of its associated definitions in the Defs table and all of its definitions' associated sign in the Signs table

➢ **/<string:word>/uploaddef**: used to add an English definition with the templates base.html and uploaddef.html, generating an upload form that adds a row to the Defs database with the user input on successful submission (unless the user provides an empty input) and relates it to the provided word

➢ **/<string:word>/<int:defid>/edit:** used to edit an English definition with the templates base.html and uploaddef.html, prepopulating the Add Definition form with the existing definition and, on successful submission, changing the corresponding definition field of the provided definition in the Defs table

➢ **/<string:word>/<int:defid>/delete:** used to delete the provided English definition, i.e. removes the corresponding row in the Defs table as well as all of its associated ASL signs in the Signs table

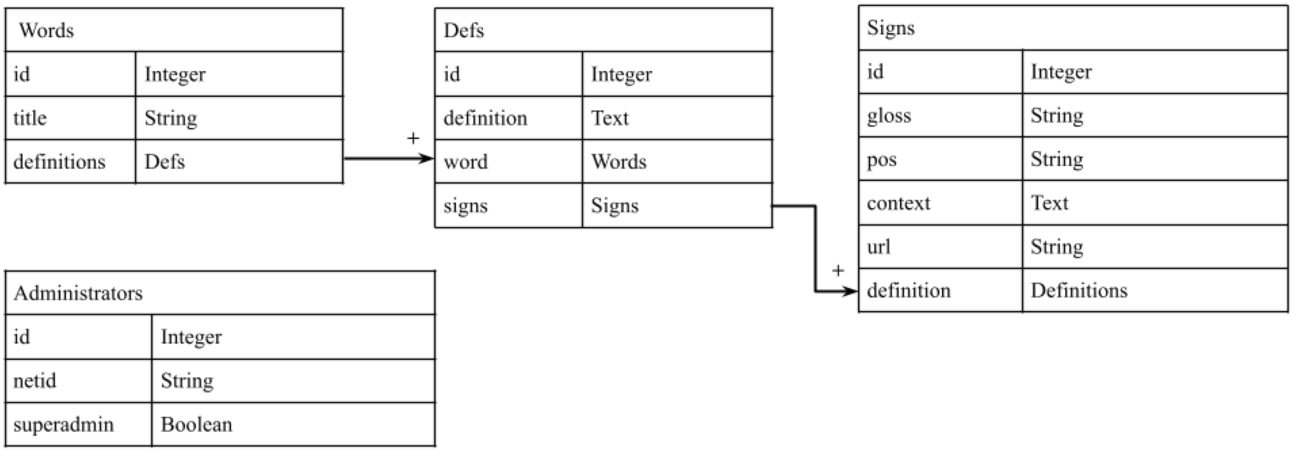
➢ **/<string:word>/<int:defid>/uploadsign:** used to add an ASL sign with the templates base.html and uploadsign.html, generating an upload form that adds a row to the Signs database with the user input on successful submission (unless the user provides empty input for required fields or a non-YouTube link for the URL field) and relates it to the provided word and definition

➢ **/<string:word>/<int:defid>/<int:signid>/edit:** used to edit the provided ASL sign with the templates base.html and uploadsign.html, prepopulating the Add Sign form with the existing inputs and, on successful submission, changing the corresponding fields of the provided sign in the Signs table

- **/<string:word>/<int:defid>/<int:signid>/delete:** used to delete the provided ASL sign, i.e. removes the corresponding row in the Signs table
- **/login:** used to log in the user through a call to the CAS API, checks if the supplied netid has administrator privileges, and, if so, redirects to the administrator interface homepage
- **/logout:** used to log admins out of the administrator interface
- **/logout_callback:** used to perform necessary actions after successful logout, removing admin ticket and redirecting the user

4.3 Data Management Tier - SQLAlchemy and PostgreSQL Schema

The ASL Search database relies on SQLAlchemy schema which implements the postgreSQL database in the data management tier. Essentially, this is the conceptual schema of how a program will view the database in SQLAlchemy, though it actually creates an underlying postgreSQL database.



Words	
id	Integer
title	String
definitions	Defs

Defs	
id	Integer
definition	Text
word	Words
signs	Signs

Signs	
id	Integer
gloss	String
pos	String
context	Text
url	String
definition	Definitions

Administrators	
id	Integer
netid	String
superadmin	Boolean

4.3.1 Words

- **id (Integer):** unique numerical id for every word (primary key)
- **title (String):** English word to be looked up in the dictionary with its own word page, must be a unique word or will throw an Exception
- **definitions** (**relationship field** — one-to-many relationship with Defs implemented by a SQLAlchemy relational table): all the definitions of a given English word

4.3.2 Defs

- **id (Integer):** unique numerical id for every definition (primary key)
- **definition (Text):** English definition to be displayed on an individual word page, uses Text type to handle longer inputs
- **word** (**relationship field** — back-populate relationship with Words, each definition maps to exactly one word): the associated English word being defined by a given definition
- **signs** (**relationship field** — one-to-many relationship with Signs implemented by a SQLAlchemy relational table): all the ASL signs of a given English definition

4.3.3 Signs

- **id (Integer):** unique numerical id for every sign (primary key)
- **gloss (String):** the ASL gloss for a given ASL sign
- **pos (String):** the part of speech of a given ASL sign
- **context (Text):** additional information for a given ASL sign such as examples, regional variants, etc., uses Text type to handle longer inputs
- **url (String):** the YouTube URL for the video depicting a given ASL sign

- **definition** (**relationship field** — back-populate relationship with Defs, each sign maps to exactly one definition): the English definition associated with a given ASL sign

4.3.4 Administrators

- **id (Integer):** unique numerical id for every administrator (primary key)
- **netid (String):** the Princeton netid of a given administrator
- **superadmin** (**Boolean**): whether or not a given administrator is a "super" administrator, meaning they cannot be removed from the Manage Admins page in the user interface

4.3.5 PostgreSQL Schema

Words	
id	INT (primary key)
title	VARCHAR(255)

Words_to_Defs	
word_id	INT (primary key)
def_id	INT (primary key)

Defs	
id	INT (primary key)
definition	TEXT

Defs_to_Signs	
def_id	INT (primary key)
sign_id	INT (primary key)

Signs	
id	INT (primary key)[1]
gloss	VARCHAR(255)
pos	VARCHAR(255)
context	TEXT
url	VARCHAR(255)

Administrators[2]	
id	INT (primary key)
netid	VARCHAR(255)
superadmin	BOOLEAN

This is the underlying postgreSQL database implemented with SQLAlchemy. The one-to-many relationships between Words and Defs, as well as Defs and Signs, are implemented by intermediate tables mapping primary keys to foreign keys, allowing word to have multiple definitions and definitions to have multiple signs.

4.4 Interesting Design Problem

4.4.1 Upload Process

Add mentioned in Section 2.2, deciding how to implement the Upload process of adding English words, which can have multiple definitions, which can have multiple corresponding ASL signs, was a difficult process. I began by sketching out different designs and meeting with Professor Buchholz to get his feedback, then changed the design even further as I began implementation. Details and a screenshot of an original Figma prototype are contained in Section 2.2, but overall, this was one of the biggest challenges and points of learning during the project. Through this process, I learned the pros and cons of consolidating features for ease of access versus spreading them out for a more digestible format.

4.4.2 WTForms

Another design choice I made was in the actual tool I used to implement the forms. I was used to working with HTML <form> tags which allow you to add <input> fields and a <submit> button in order to receive user input and handle it on the python end, however, I found it difficult to handle multiple actions beyond just redirecting on submit. For example, when adding a sign, I needed to not only add the sign to the Signs table in my database, but also relate it to the definition it was added under, as well as redirect back to the Word Page. I learned about WTForms from an online blog and decided I could leverage its validate_on_submit() function in my routes.py file for handling multiple complex actions at once. I could also easily raise validation errors for the input fields by using WTForms built-in validators as well as design my own custom validation, like checking for English words in the Upload Word form (using the pyenchant library) and for YouTube URLs in the Upload Sign form (using regex) which is described further in Section 4.5.1. Overall, it proved to have many beneficial features that I could use in the implementation of the upload forms.

4.4.3 Login

Finally, I was also posed with the problem of Admin Login: how could I allow Professor Buchholz to log in and manage content without storing his sensitive information or making login too accessible for students? I wanted to avoid implementing my own database of usernames and passwords for administrators, as I would need to encrypt this information in order to protect the security of the web application. However, I worried that by implementing login with CAS authentication, students would feel encouraged to log in and be confused when they were unable to gain access to more features. I decided that leveraging existing CAS authentication was still the best way to proceed for security purposes, especially since Professor Buchholz already had a netid he was familiar with using. You can see from his interview notes in Section 7.2.2. that he stated he knew “the CAS drill,” and this familiarity had a positive impact.

I then debated providing him with a hidden login link that would be inaccessible from the navbar, in order to avoid exposing students to the login capabilities, however, after discussing the idea, he felt he wouldn't want to share this link with new administrators each time a colleague wanted to contribute to the app. Finally, we settled on providing login in the navbar, but adding an additional layer of checking that a user has admin privileges before revealing admin features. This could be edited from a "Manage Admins" tab, where Professor Buchholz could add and remove admins directly from the frontend of the app, meaning in the future, he would never need to modify the backend database directly. We hoped that indicating it was an "Admin" login would deter students from wasting time logging in, and I found this to be true as seen in the user interview notes from Section 7.3.2.

4.5 Testing

4.5.1 Internal Testing: Designing your code to test itself. Performed by programmers.

- **Checking function/method failures:**

  The only Exceptions encountered thus far have been SQLAlchemy integrity errors, which occur when a user tries to add a word that has already been added to the database since the title field should be unique. This Exception is currently caught by redirecting the administrator to a rejection page stating the word exists and prompting them to return to the Upload Word page to try adding a different word. In case any additional errors are encountered, the user is presented with an unexpected error message and prompted to return to the Homepage. The type of the error is then printed to the console (local) or to the Heroku logs (remote).

- **Validation**:

  All input fields except for Additional Information require some input, which is ensured by the DataRequired() WTForms validator. Furthermore, the input for uploading a new English word must consist of English words only, which is validated using the pyenchant library which checks inputs against a robust English language dictionary. The input may be multiple English words, like “Thank You,” but will throw a validation error on “Thnak You,” emojis, non-English characters, and other strange inputs. Finally, the YouTube URL field when adding a new ASL sign validates against a regex expression that matches a variety of YouTube URL sharing links.

4.5.2 External testing: Designing code or data to test your code.

- **White box external testing:**

Boundary testing was done by trying to add inputs with emojis, non-English characters, etc. I added many different English words with different numbers of definitions, and ASL signs as well as different grammar concepts. I also tried short and long inputs for the definition and Additional Information field, while length is restricted for words, ASL gloss, part of speech, and URLs. I also attempted to conduct coverage testing with the python coverage tool, but experienced database authentication issues, which I plan to tackle further ahead of Dean’s Date.

- **Black box external testing:**

I conducted use case testing by executing the cognitive walkthroughs I designed for both regular users and administrators myself, before conducting interviews with others. The functionality I had implemented worked, but user testing exposed difficulties that those unfamiliar with the app experienced and allowed me to improve the clarity of these functionalities, further described in the next section.

The use of SQLAlchemy prevents against SQL injection attacks, as no SQL code is directly written or executed by the program and SQLAlchemy handles the creation of prepared statements. The validation of English words described in the last section prevents against cross-site scripting attacks, as it requires the input to be English words and prevents the upload of attacks in the form of HTML, Javascript, or other code. This is only done for English words because they are displayed in the Search Results page and linked to individual Word Pages. All other input fields are displayed as text and cannot be made into cross-site scripting attacks, except for YouTube URLs which are also validated.

Finally, I conducted stress testing by adding a large number of random words using the python random-word library, which can be found in the stress_db.py file. Then, using the user interface, I also tried adding the longest word in the English dictionary, pneumonoultramicroscopicsilicovolcanoconiosis, uppercase and lowercase inputs, which should work since I make all inputs lowercase before adding them to the database, and multi-word inputs like “Thank You,” “Open Minded,” etc.

5. Evaluation

5.1 Administrator Evaluation - Professor Buchholz

I consulted with Professor Buchholz many times throughout the semester as referenced in earlier sections, but also conducted a formal cognitive walkthrough of the administrator interface at the end of the semester. From this interview, I made lots of wording changes in different places of the web app to eliminate confusion: changing “Login” to “Admin Login,” “Add a new Word” to “Add a new English Word,” “Add Sign” to “Add ASL Video,” and “Context,” to “Additional Information.” I also learned I had to make the YouTube validation more robust to handle different types of links that users could copy, as Professor Buchholz used a method I hadn’t considered. To fix this issue, I found a regular expression from Stack Overflow that matched with most types of YouTube sharing links and used that to validate the URL field instead of the simple one I had written. Professor Buchholz also requested that I add an All Words page, from where administrators could view and delete words from the database rather than having a delete button next to the word on its own word page. He thought the separation of functionalities made things easier to understand overall, such as the multi-page Upload process, and preferred that deletion be the same. Finally, when experimenting with the Manage Admins page, Professor Buchholz removed his own netid, which I realized shouldn’t be possible since he wouldn’t be able to log back in. I added a superadmin field to the Admins table to ensure that Professor Buchholz and myself would always be admins. These changes were all implemented ahead of the User Evaluation, so you will notice I use the updated wording in the next section.

https://drive.google.com/file/d/1PqtnOrLAYNrgA9zeSwyqnzYcTngsI4rX/view?usp=sharing

See the video linked above to watch Professor Buchholz's final comments on the app.

5.2 User Evaluation - ASL Students

At the end of the semester, I also met with three of Professor Buchholz's current ASL students and conducted cognitive walkthroughs of the user interface. A specific task list and notes from these interviews can be found in Section 7.3. Some of the most important feedback I received had to do with searching difficulties, for example, when a user searched "to open" and was unable to find the page for open, or, similarly, when a user searched "OPEN-DOOR." This alerted me to the need for creating a more robust searching functionality that can query the database for similar word forms of a given English word as well as query for the ASL gloss that appears on that word page. I also received the suggestion that the Additional Information field could be optional so that Professor Buchholz wouldn't have to fill in information for simple cases like an English word with a single definition or ASL sign, which I have since implemented. Finally, students also suggested that hyperlinking related pages inside Additional Information would be useful, such as linking the page for "Door" in the information for OPEN-DOOR rather than just stating see DOOR for more. Something that surprised me was that students seemed averse to trying the Admin Login, perhaps because it said "Admin." This was something Professor Buchholz was worried about but ended up not being a big point of confusion for users.

6. Conclusions

6.1 Future Work

ASL Search remedies many problems with existing dictionaries and can be used right now by Professor Buchholz as it is. However, there is definitely room for improvement. I plan to tackle additional improvements and new features before Dean's Date and even over the summer, ahead of the Fall semester when students may begin using the app.

6.1.1 Existing Weaknesses

The only glaring weakness in the functionality of ASL Search is its lack of advanced searching capabilities. As of right now, users must search exactly for the English word they are looking for or a matching substring of the word. This is problematic for a few reasons:

Firstly, users cannot find the word they are looking for if they search for a keyword that is similar to the word but not a substring. For example, if a user searches for "to open" or "opens," they will not find the word "Open" in the Search Results page. Fixing this would require modifying the database from Section 4.3 by adding an additional field in the Words table where Professor Buchholz can add a list of similar words to a given English word that can also be queried from the search bar.

Furthermore, as I mentioned in Section 5.2, users would like to be able to search for ASL gloss using the search bar. This addition wouldn't involve modifying the database but modifying the SQL query executed by searching to look through a combination of the Words and Signs database to find matching results.

I also later hope to add a field to the Words database containing a search frequency metric. The results in the Search Results page are currently ordered alphabetically, but in most advanced online English dictionaries, words that are more frequently searched are usually prioritized. For example, if a user were to search “op,” the word “Opens” should come before the word “Opaque,” since it is a more commonly used word, even though “Opaque” comes before “Open” alphabetically.

Finally, I would like to display search results in a dropdown rather than navigating the user to a Search Results page and requiring an additional step. This would be implemented using AJAX to update the dropdown on each user keystroke as well as Javascript to render the dropdown both in the homepage and the navigation bar. An intermediate step to this goal would be displaying search results on the Homepage itself, below the search bar but not in a dropdown, which would eliminate the added step but would not look as clean or as similar to most online English dictionaries.

6.1.2 Stretch Goals

In addition to bettering searching, I also hope to add new features that were requested by Professor Buchholz or suggested through user evaluations.

For one, Professor Buchholz mentioned that he found it annoying that he had to press play on videos in existing ASL dictionaries, and it would be much easier if videos were to autoplay. Furthermore, given that these signed videos are very short and often need to be observed more than once to be understood, he found it annoying that he had to press play again to rewatch a video instead of videos looping by themselves. Implementing video autoplay and looping would require researching YouTube embedding, which is currently done using <iframe> tags. These tags can be modified to allow for features like autoplay and looping, but since each word page can contain multiple videos, it may also require Javascript to detect which video is being focused on by the screen currently.

Another feature suggested by users was the ability to add hyperlinks in the Additional Information field when adding an ASL sign. This would allow students to easily navigate to the page for a related word to the word page they are currently on, rather than having to search for the word using the search bar.

6.1.3 Further Testing

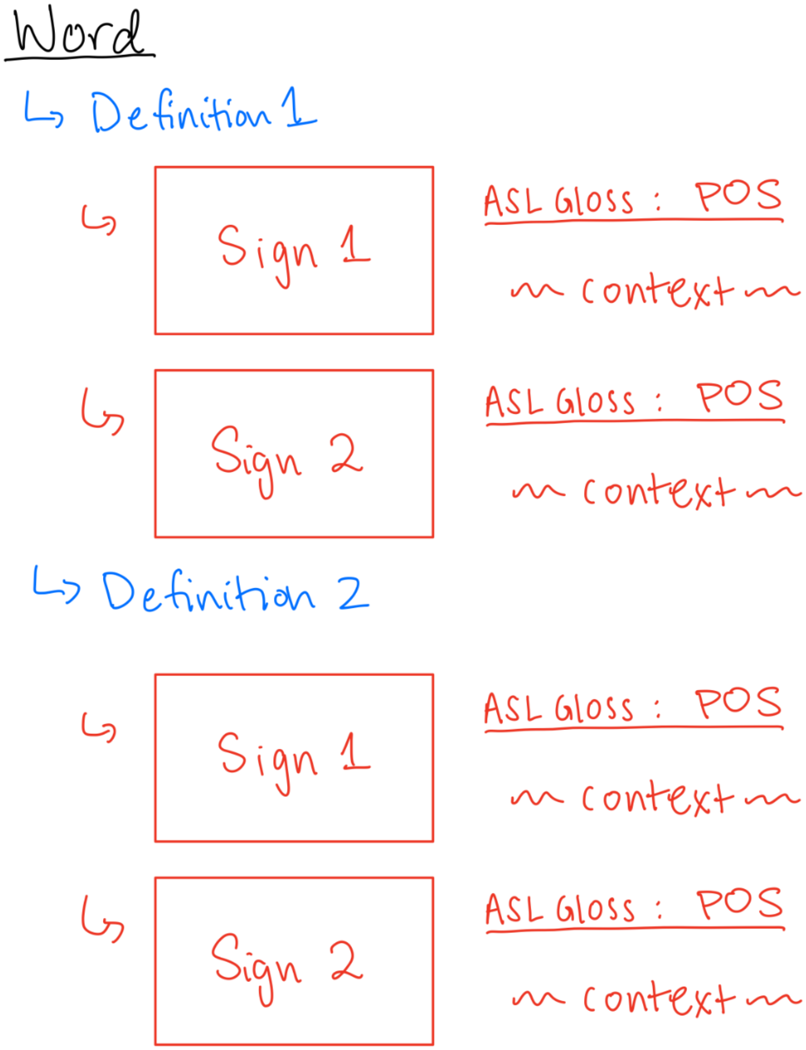
I was also unable to resolve a database authentication issue that prevented me from doing coverage testing on my codebase. I plan to do this prior to Dean's Date in order to ensure all of my code is reachable and necessary, and remove anything that is not encountered.

6.2 Acknowledgements

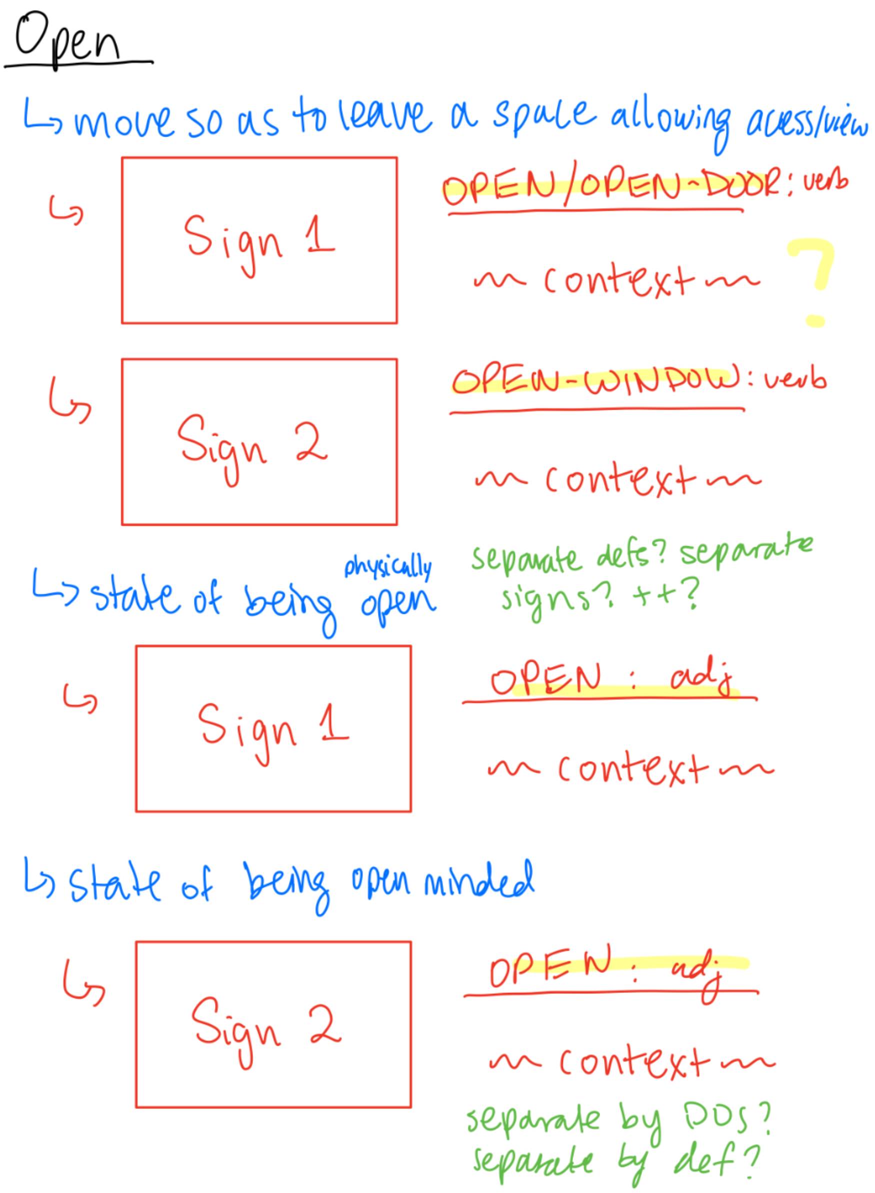
Overall, I learned from this project the importance of dedicating technological resources to ASL instruction and providing customizable platforms to give instructors the ability to make their own content and feel secure in the materials to which they refer their students. I would like to express my immense gratitude towards Professor Dondero, who met with me weekly during an already busy semester to advise me on this project, Professor Buchholz for his consistent and invaluable input, and the students and peers who helped test and improve ASL Search over the semester.

7. Appendix

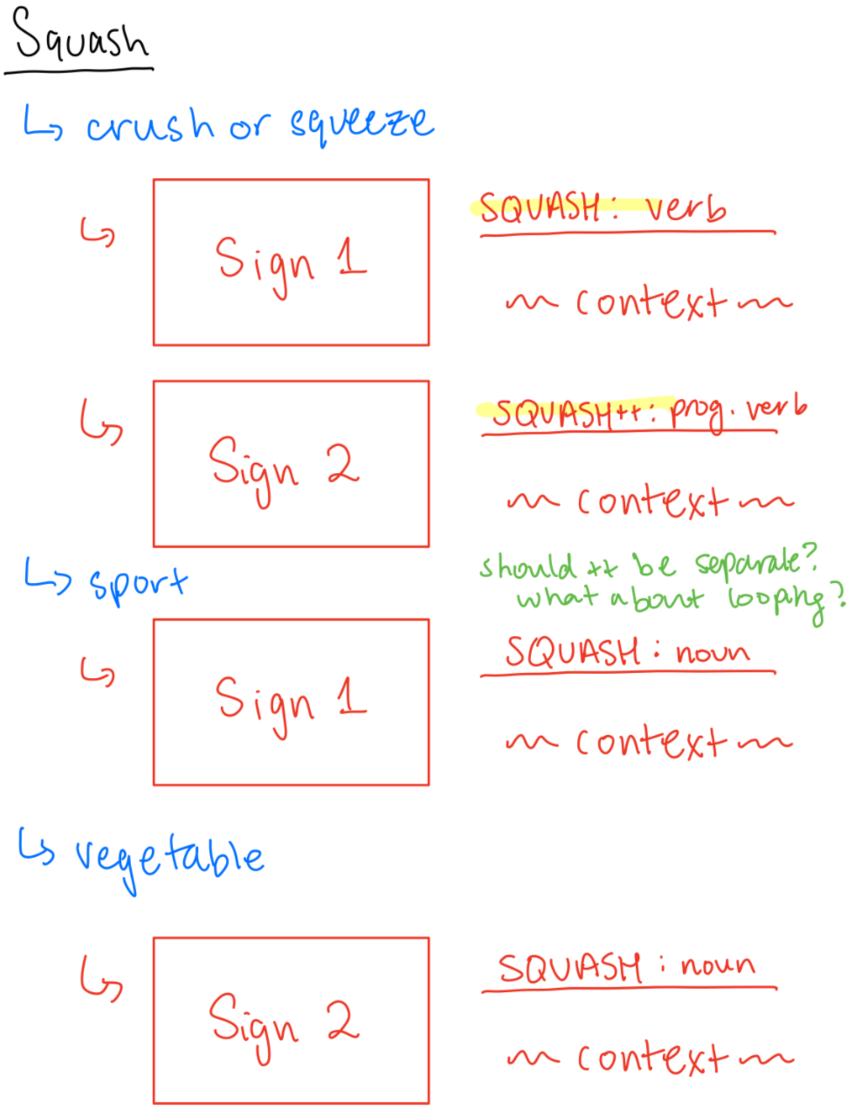
7.1 Initial Learning Notes



I met with Professor Buchholz regularly at the beginning of the semester to understand the relationship between English and ASL, and to determine the best possible structure for the word page. Here you can see the initial concept for the word page drawn on my iPad, which I made during one of our meetings. We realized that this format would allow for the word page to capture multiple sign variations for a single English word.



Together, Professor Buchholz and I had to determine how complex ASL grammar forms like noun-verb pairings would be represented by walking through examples like the word "Open." You can see my questions in green: should the verb open and the adjective open (both having to do with physical openness) be under separate definitions? Should we separate signs by part of speech in addition to the definition? How should we handle progressive verbs (indicated by the ++), which in ASL are verb signs repeated twice, when the eventual plan is for videos to loop?



Another helpful example I walked through with Professor Buccholz was the word "Squash," which has multiple meanings and was also used to decipher the issue of progressive verbs. Since implementing video looping would repeat the sign constantly, we decided to indicate that a verb should be signed twice for the progressive verb in the "context" section, which later became “Additional Information.” By going through many examples like this, we could be sure that the fields we chose were sufficient context for any sign, as part of speech and ASL gloss provided key information, and Professor Buchholz could always provide further details for more complex grammar forms in Additional Information.

7.2 Admin Evaluation Notes

Important notes that were either implemented or included in Future Work are bolded.

7.2.1 Admin Evaluation Task List

- ➢ Navigate to ASL Search using the link aslsearch.herokuapp.com (Homepage)
- ➢ What would be your first step to start managing dictionary content? (Login)
- ➢ Add the English word “Water” to the dictionary
- ➢ Add a definition for the word “Water”
- ➢ Using a provided sample YouTube video, add an ASL sign for WATER
- ➢ Edit the sign and then the definition you just added
- ➢ Delete the sign and then the definition you just added
- ➢ Add a new administrator, then delete them

7.2.2 Admin Evaluation Notes

- ➢ Thought login made sense and “knows the CAS drill,” but was worried students would try to log in and **suggested renaming “Login” to “Admin Login”**
- ➢ Noted that logging in updated the navbar, expected clicking “Upload” would allow him to add videos

- Pleasant surprise to see that “Upload” was just for the initial word and that the upload process was broken down, found it more digestible than our earlier plans for a single upload page
- Thought **“Add a new Word” should be “Add a new English Word” instead**, even though he knows this from our discussions other instructors might need clarification that they are being prompted to add an English word, not an ASL word
- Didn’t expect immediate redirect to a newly created word page on submitting the upload form, another pleasant surprise as he expected a more overwhelming next step with “many boxes to fill in”
- Clicked “Add Definition” and found functionality was as expected
- Was **confused about “Add Sign” and would prefer “Add ASL Video”** to be clear that this is the step where the video is actually added
- Likes that adding definitions and signs takes place from the word page, helps him know he’s in the right place and adding information for the right word
- When adding the YouTube video, clicked on “Share” then “Copy” (produces a www.youtu.be.com link instead of www.youtube.com) instead of copying URL from the URL bar so the **YouTube URL field validation didn’t work - needs to be more robust to handle different kinds of YouTube linked**
- Thought **“Context” should be “Additional Information,”** to clarify that it could include example sentences, regional info, etc. - maybe **add placeholder**
- After adding the sign, thought the resulting word page was really cool!

- ➢ Tried editing the definition and the additional information for the sign, thought both made sense and the placement of the buttons made it clear what was being edited
- ➢ Also thought deletion hierarchy made sense - deleting a definition deletes its signs too
- ➢ Thought it would be helpful to have an **“All Words” page to view and delete words from there** rather than from the word page - thought it was too much management from one page
- ➢ Found it easy to add and remove admins, but accidentally deleted his own netid, then couldn’t log back in after logging out! - **need to add a superadmin for admins who can’t be removed**
- ➢ Overall very impressed and excited

7.3 User Evaluation Notes

Important notes that were either implemented or included in Future Work are bolded.

7.3.1 User Evaluation Task List

1. Navigate to ASL Search using the link aslsearch.herokuapp.com (Homepage)
2. What would you do to learn more about the app? (About page)
3. You would like to sign that it’s hot and you want to open a door, but forgot the sign for OPEN-DOOR. What would you do? (Searching)
    a. Search from Hompage vs. Navbar, search keyword, finding the right page
4. From the word page for OPEN, would you know which sign to choose?
    a. Context clues used, thoughts on the adequacy of provided information

5. You would like to sign that you’ve been learning ASL for a year, but forgot the sign for LEARN. What would you do?
6. From the word page for LEARN, would you know how to use the sign in the video?

7.3.2 User Evaluation Feedback

- **Student 1**
  - Homepage made sense, liked the search bar in two places
  - Navigated to About page for more information
  - Didn’t think of clicking “ASL Search” to navigate back to Homepage
  - Searched “open,” made sense that OPEN, OPEN-DOOR, and OPEN-WINDOW are all together
  - Thought it was clear that she should sign LEARN twice for LEARN++
  - Didn’t think it was necessary to log in, assumed it was for admins only
- **Student 2**
  - Liked Homepage, navigated to About page for more information, clicked “ASL Search” to navigate back
  - Searched “to open” first, then tried “open” - said this was a common problem in existing dictionaries and **would like to be able to look up similar word forms**
  - Liked the inclusion of noun-verb pairings in the word page and the “see DOOR” or “see WINDOW” for more, but **would expect some pointer or link directly from this page as well**

- Thought the context was all important information and covered everything she needed to know like regional variants
- Thought that **additional info could be optional** for more simple cases, for example, when an English word has one definition or sign
- Understood the difference between the gerund/progressive verb and the regular verb form of LEARN from Additional Information
- Didn’t think to try logging in and expected to be redirected, when prompted to the rejection page was as expected

➢ **Student 3**

- Went to About to learn more, then clicked on ASL Search to go back
- Looked up “OPEN-DOOR,” then tried “open” - his second thought was that he might find OPEN-DOOR in the page for OPEN, but he **assumed he could search for any related ASL gloss and the word page should come up**
- Would also like to a see **hyperlink to DOOR in the info for OPEN-DOOR**
- Searched “learn,” thought it was clear to sign twice for LEARN++
- Expected that admin login was for Professor Buchholz to fix things and that he would not be allowed to log in

Bibliography

"American Sign Language." National Institute of Deafness and Other Communication Disorders. U.S. Department of Health and Human Services. https://www.nidcd.nih.gov/health/american-sign-language.

"American Sign Language ASL Dictionary." Sign ASL. https://www.signasl.org/.

"ASL American Sign Language." • ASL • American Sign Language. https://www.lifeprint.com/.

"COS 333, Spring 2022." Princeton University. The Trustees of Princeton University. https://www.cs.princeton.edu/courses/archive/spring22/cos333/index.html.

"Flask-Bootstrap¶." Flask-Bootstrap - Flask-Bootstrap 3.3.7.1 documentation. https://pythonhosted.org/Flask-Bootstrap/index.html.

"Handspeak® Sign Language Online:" Handspeak. 2022. https://www.handspeak.com/.

"Learn Python Programming." Learn Python Programming - Python Tutorial. https://pythonbasics.org/.

Ogundepo, Odunayo. "Create a Web App and Deploy It to the Cloud in 20 Minutes with Python." Arctype Blog, September 27, 2021. https://arctype.com/blog/postgres-heroku/.

Real Python. "Deploying a Python Flask Example Application Using Heroku." Real Python, June 4, 2021. https://realpython.com/flask-by-example-part-1-project-setup/.

"Sign Language Dictionary." SpreadTheSign. http://www.spreadthesign.com/en.us/search/.

Signing Savvy, LLC. "ASL Sign Language Video Dictionary." Signing Savvy. https://www.signingsavvy.com/.

"Sqlalchemy¶." Flask. https://flask-sqlalchemy.palletsprojects.com/en/2.x/.

Wang, Xue. "Focus on deploying a simple Flask Application into Heroku, interacting with PostgreSQL and Troubleshooting." Towards Data Science. https://towardsdatascience.com/deploy-a-micro-flask-application-into-heroku-with-postgresql-database-d95fd0c19408

"Welcome to Flask¶." Welcome to Flask - Flask Documentation (2.1.x). https://flask.palletsprojects.com/en/2.1.x/.

"WTFORMS." WTForms. https://wtforms.readthedocs.io/en/stable/.