

The University of Akron

IdeaExchange@UAkron

---

Williams Honors College, Honors Research  
Projects

The Dr. Gary B. and Pamela S. Williams Honors  
College

---

Spring 2022

## Resume Parser and Job Search

Stephan Gunawardana  
srg97@uakron.edu

Follow this and additional works at: [https://ideaexchange.uakron.edu/honors\\_research\\_projects](https://ideaexchange.uakron.edu/honors_research_projects)



Part of the [Databases and Information Systems Commons](#)

Please take a moment to share how this work helps you [through this survey](#). Your feedback will be important as we plan further development of our repository.

---

### Recommended Citation

Gunawardana, Stephan, "Resume Parser and Job Search" (2022). *Williams Honors College, Honors Research Projects*. 1478.

[https://ideaexchange.uakron.edu/honors\\_research\\_projects/1478](https://ideaexchange.uakron.edu/honors_research_projects/1478)

This Dissertation/Thesis is brought to you for free and open access by The Dr. Gary B. and Pamela S. Williams Honors College at IdeaExchange@UAkron, the institutional repository of The University of Akron in Akron, Ohio, USA. It has been accepted for inclusion in Williams Honors College, Honors Research Projects by an authorized administrator of IdeaExchange@UAkron. For more information, please contact [mjon@uakron.edu](mailto:mjon@uakron.edu), [uapress@uakron.edu](mailto:uapress@uakron.edu).

Resume Parser and Job Search

Stephan Gunawardana

Department of Computer Science

**Honors Research Project**

Submitted to

*The Williams Honors College  
The University of Akron*

Approved:


 Date:

4/14/2022

Honors Project Sponsor (signed)

Yincai Xiao

Honors Project Sponsor (printed)

 Date:

4/13/2022

Reader (signed)

Zhong-Hui Duan

Reader (printed)

 Date:

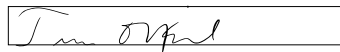
4/14/2022

Reader (signed)

En Cheng

Reader (printed)

Accepted:

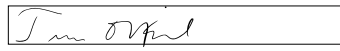
 Date:

4/14/2022

Honors Department Advisor (signed)

Timothy O'Neil

Honors Department Advisor (printed)

 Date:

4/14/2022

Department Chair (signed)

Timothy O'Neil

Department Chair (printed)

**Resume Parser and Job Search**

**Stephan Gunawardana**

**Williams Honors College**

**The University of Akron Honors Research Project**

**GitHub Link: <https://github.com/sgun7/HonorsProject>**

# Table of Contents

Abstract .....	4
Preface .....	5
Flow of Website .....	8
Preparation.....	9
Breakdown of website .....	12
-Main Idea.....	12
-Home Page.....	13
-User Authentication.....	14
-Resume Upload and Parsing.....	17
-Viewer.....	21
-User Page.....	22
Challenges .....	24
Conclusion.....	25

## **Abstract**

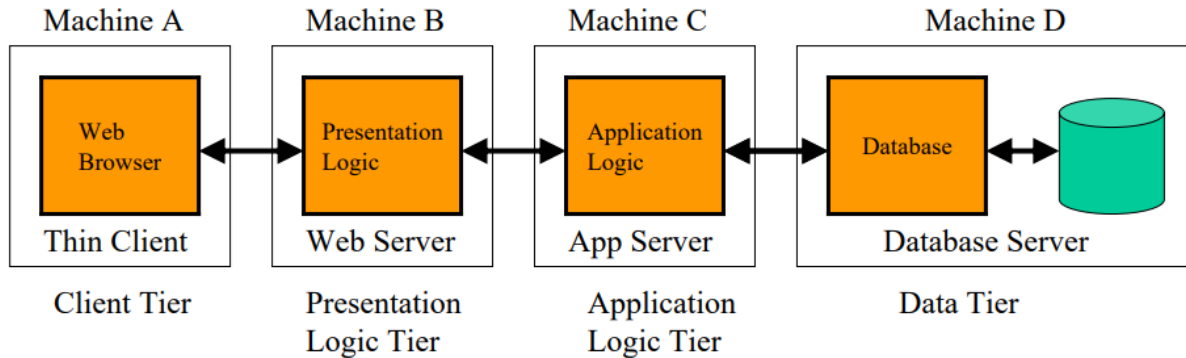
The basic goal of this project was to create a four-tier web application which would allow users to upload their personal resumes to the website, and it ultimately would parse the resume and look for keywords. Then using these keywords, it would use a job API to look for these values and ultimately display job information to the user based on their skills. While applications exist which would look for jobs based on skills, it was mostly manually inputted skills. However, with my application you can simply upload your resume to the website, and it will basically take care of this step for you. It must be noted that this is also a four-tier web application therefore I have also built-in user-authentication with sign up and sign in. Therefore, each user will have their own profile, which is protected by a high level of authentication. Lastly, the user accounts will also be supported by a modern backend database therefore all the values such as your skills, username, and preferred job offers will be saved uniquely for each user. There was a lot of learning included within this project, while I was not new to web-development this project required a lot more experience than what I had at the beginning of the project. This project implements the use of Vue for the frontend and Firebase/Firestore/Google storage for the backend.

## Preface

There are several goals I wanted to achieve with this project. The main one would be to ultimately learn something new and apply this a real-life application. Another one of my goals which ultimately aligns with the purpose of the honors project was to implement a set of learnings I have grasped over the years through classes related to my major. Lastly, I wanted to create a project which would not simply be a copy of something which is already created, instead something new which has the potential to be helpful for various users.

As mentioned within the abstract this would ultimately be a four-tier web application which would allow users to parse their resume and do an automatic job search based on the skills of that person's resume. I have personally gone through the job-search process for a college student, and it tends to be a very long and tedious process. With the use of top job searching websites, most websites I did research on seems to require that the user manually inputs keywords/phrases. While this can be seen as a trivial task it can somewhat be difficult if you have a lot of skills and your job preferences are large. Therefore, with this application the whole job search process will be simplified and allow users to focus on the real goals of jobs such as the interview.

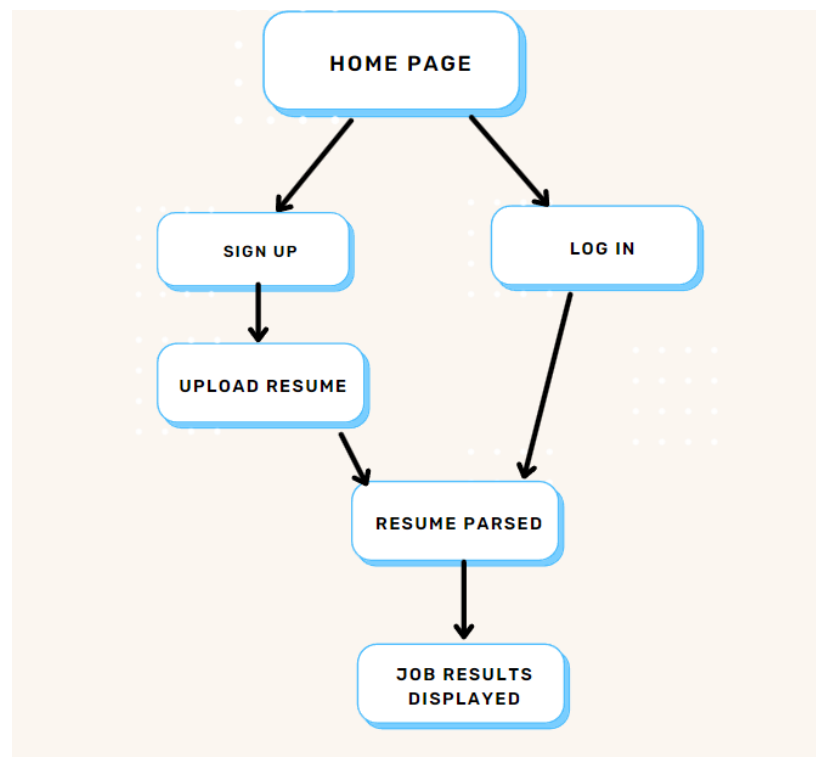
## 4-Tier Application Architecture



(Figure 1)

Another point of focus for this project was the four-tier web application. The details of my websites relation to this will be displayed later within the paper, however I wanted to discuss the idea behind it and how it ultimately forms the backbone for my website. I first learned about four tier web applications in ISP class during one of the semesters. The basic idea behind the four-tier web application would be that a normal web application being broken down into four main sections. Referring to figure 1 the first would be the client tier in which is the topmost layer, this is most likely the user interface and is the presentation of the website. Next, the logic tier is the middle section in which it coordinates commands, makes logical decisions, and acts as a connection between the two surrounding layers. This layer can be broken down into two sections within a four-tier web application. Lastly would be the data tier in which the data is stored and retrieved from. One key component to understand with this diagram is the arrows moving back and forth, there is no direct flow of data or logic on a four-tier web application, in fact data can be received or sent out from all tiers. Lastly, one of the main

reasons I chose this type of architecture is the fact that encapsulating code within certain sections allows the developers to benefit greatly. For example, detecting bugs when the code is broken down into several sections becomes somewhat of a trivial task simply because it becomes very evident where the bug is occurring. Also, if we are creating a large codebase, it can be somewhat difficult to update or change sections of the code if everything was bunched together, however with this architecture changes are very efficient and effective. This type of architecture is also very common within the modern world, therefore this sum of all the reasons why I chose this architecture for my own website.



(Figure 2)



## Flow of Website

Figure 2 highlights the main flow of the website. Each box representing a main feature included within the program comprised into a singular term. The arrows represent the flow of data or routes throughout the website. It must be noted that there will be other accessible assets to the website however this diagram will highlight only the main ones. Also, each box within the diagram will be explored thoroughly later within the paper, however this section will primarily focus on the flow of the main website.

First, we start of at the main home page, this is where every user will be routed to despite their account status. Therefore, all users will see the same homepage. It will contain a sign-up notification button which will allow users to sign up for the website. Coincidentally, it will also contain a log in button for previous users and this will ultimately split the path for the flow of the diagram. Next, the user will be taken to a register page if they chose to sign up for the website. This webpage will contain an input for username, password, and bio which will be updated within the backend-database. The same thing will occur for the log in feature however instead of updating the backend it will perform a check on the backend to validate if the user has typed in the correct username and password. It must be also noted that this will use modern validation/security protocols to maintain reliability and security of the website. Also, the sign up will contain a one extra step when compared to the log in. Which would be the upload resume page, which will simply contain the upload drag and drop feature added for the file. One thing to note from this page is that this will only upload the resume to our database and will not actually parse the resume. We are under the assumption that the logged in user already has uploaded the resume.

The final section to the flow diagram starts out where the resume is parsed, this will not be found within a separate page instead the user will be able to simply upload the resume, and this will be displayed as a background function. The details of the parsing will be explained later within the paper. Next, the data or keywords which are pulled from the resume will be uploaded to the database for each user, therefore each user will contain a set of keywords which will play a key role searching for the appropriate job results. The last destination of the flow diagram will be a physical page which displays the job results, in a table manner. This diagram will be pulled from an API call. The keywords parsed from the resume will provide the route for each API call which will be displayed in an appropriate manner. This will be the most used page on the website most likely because it displays important data required to accomplish the main goal of the website. Also, the movement from one page to another will be very fast and only be limited by human error, simply because we are using modern frameworks and database for this project. Another thing to note would be the fact that lots of these pages will be safeguarded with route protection based on the current user's login credentials. Therefore, you will not be able to skip into pages which are guarded by the user protection

### **Preparation**

There were several steps taken before preparing for this project. This was not a simple idea which was easily thought of overnight or over a short period of time. In fact, this project started out as a completely different project. However, after discussing several ideas with my sponsor for the project we were able to finally decide on a proper idea. When I started this project, I knew I wanted to work with some type of four-tier web

application because I was passionate about this project and wanted to learn more. It was evident from several classes I took that there was a lot more for me to learn and understand about this type of work. Therefore, all these indications lead me to pursue this current project.

As mentioned previously there were several iterations of this project throughout the process of creation. There were three main iterations of this project which eventually progressed into the result. First, it started out as a vague idea of a website which serves to be a large social media, which provides to make a social space where users can communicate about their work preferences or work experiences to help with the hiring process. There were two main problems with this idea. First, this type of idea was already heavily populated, there are several websites like this in the real world such as LinkedIn and Glassdoor. The other problem would simply be that I would not be really adding anything unique from the other existing websites, therefore there would be no enticement for the users to switch platforms because it would be providing the exact same benefits. After this idea, I decided to pursue an idea which would serve as more of a website which allows users to learn and understand the properties and tools used within frontend development. This would simply be a website which would allow users to complete certain frontend tasks, such as creating a simple CSS model to represent the powers and usability of CSS. There were also several problems with this idea in my opinion. There was no reason to implement a backend database for this project, therefore it would ultimately not have been a four-tier web application. Also, this would turn my project more into a research project instead of a project which is focused on becoming a real-world product which could be used by day-to-day people. After several

weeks of discussing ideas with my sponsor, I was able to finally come up with my final project. This as described above would be a project which allowed users to upload their own resumes to the website and it would automatically parse it and do a job search based on the keywords found within the resume. This covered several problems, first it allowed the use of a backend database since the users needed profiles to store their own resumes. Also, there were none to a few websites which offered this feature therefore the market for this type of product was not populated at all. Lastly, this allowed me to add my own personal touch to the website, instead of using an already existing idea.

The last thing I wanted to discuss within this section would be the specific technologies I decided to use within project to gain the most efficient and proficient methods. First for the frontend, I chose to use Vue. My three choices were Angular, Vue, and React, while the other two options were viable, I had the most experience in Vue, therefore it was the obvious choice. While I did have experience this did not mean I did not have to learn anything new. In fact, I spent the first couple weeks of the project simply reviewing what I had learned in previous classes and learning new material. My experience with languages such as JavaScript expanded a lot over the period of this project. There were also several small libraries which are referenced within the frontend which are included within the headers of each “.vue” file which are used within my project but would be too trivial to list each of them. Some notable libraries used would be first “Vuetify”. Vuetify makes component creation much similar because it already contains pre-loaded components within the libraries therefore this saves countless hours of coding CSS, which is not the focus of my website. Another resource used

would be the “Coolers” website which provides users with custom color pallets which are popular for modern day websites. This allowed me to choose a style for the website which would give it the modern looking style. Lastly, I also used the “JSON” JavaScript library which became a key component in my project simply because lots of the data being read in from the API was created as a JSON, therefore this library allowed me to parse this data properly and ultimately display in the frontend. For the backend, as mentioned previously I ended up choosing Firebase. There are a couple reasons I chose to use Firebase for this project. Like my frontend’s choice, I had already previous experience using Firebase, but for a much smaller and easier project. Firebase also contains modern day user-authentication and protection overall over the database; therefore, it provides the best protection without needing implementation by developers to increase security. Also, paired with this the cost of the website using firebase for my database and storage was very minimal. Lastly, the data visibility of using Firebase is easy to understand from the developer’s perspective which makes the whole process of this project much easier. Those roughly cover most of the main technologies used, each technology will be expanded on within the detailed discussion further down the page.

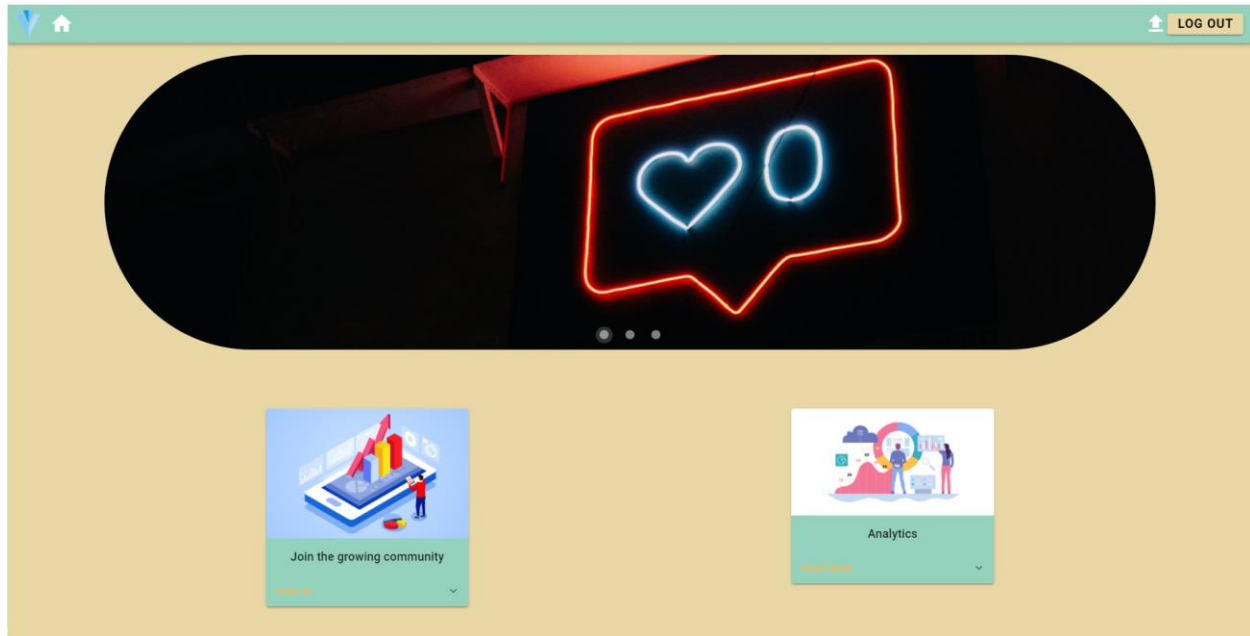
### **Breakdown of Website**

#### Main Idea

This section of the paper will highlight the main piece of written content about the project. As far breakdown of this section, I will break each subsection into the title of each page of the website. However, this does not mean all that this section will contain will be limited to the frontend. Instead, I will highlight the frontend and backend code

involved within each page. This also means that it will contain snippets of code to understand each process.

### Home Page



(Figure 3)

This page is main landing page of the website, and is the place that all users, member or non-members, will be taken to when first accessing the website. There is not much involvement within the homepage with the backend, therefore most of the code for this section will be frontend focused. We will focus on Figure 3 for this representation. First, would be the carousel component which is shown first underneath the navigation bar. I chose to add this to give an aesthetic look to the website, since this webpage will be visible to all users. A carousel component basically takes in images as vue props and creates a slider which will go to the next image based on a customized timer. Not visible within the picture, but it also contains a hover-over CSS effect which will essentially active when a mouse-hover occurs over the object, and it will trigger the

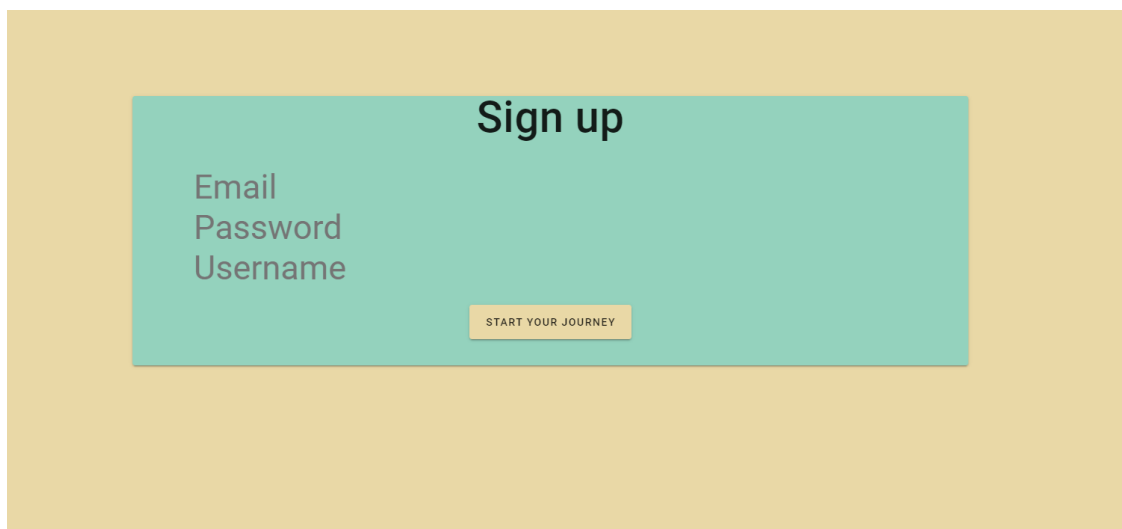
image to move-over right away without waiting for the timer to trigger. Also, there are two cards found in figure 3 which also contain various CSS properties. For example, it contains the hover over effect, shadow-effect, and flexbox sizing as well. These will help direct users to the correct page based on the button they clicked.

```
<v-flex xs12 md4>
  <image-card-1 title="Join the growing community" action="Sign Up" paragraph="We are a network of people who want to make connections and expand our knowledge of the j
Click the button above to sign up for our program" :imageSource="require('@assets/growth.png')"></image-card-1>
</v-flex>
<v-flex xs12 md4>
  <image-card-1 title="Analytics" action="Read More" paragraph="We are constantly creating data and updating our database with the most recent analytics" :imageSource="
```

(Figure 4)

Lastly, these highlight one of the key features of the using vue components, and that would be the use of props. As shown within figure 4 the component “image-card-1” is only one component, however it is displayed on the frontend and two different components with different functions. Therefore, I created a component which can take in multiple props, such as “action”, “paragraph”, “title”, and “imageSource” as seen in Figure 4. Thus, I only need to write the code once for the card, but the props passed into the component should make each instance unique based on what is passed.

### User Authentication



The image shows a user authentication form titled "Sign up". The form is set against a light yellow background. It features a teal-colored container with the following elements:

- Title:** "Sign up" in bold black text at the top right of the teal container.
- Input Fields:** Three stacked input fields on the left side of the teal container, labeled "Email", "Password", and "Username" in a light gray font.
- Button:** A yellow button with the text "START YOUR JOURNEY" in black, located at the bottom center of the teal container.

(Figure 5)

The user authentication section is a key feature of the website, because it will allow users to log in/sign up to ultimately save their data. This will contain frontend and backend use. As seen within figure 5 the sign-up card will contain a box for email, username and password. In similar fashion the log in page will only contain the email and password to log in. The frontend functionality is simple, it just saves each input box value as v-model and directly imports it to firebase through the firebase API. This API was assessable from the frontend due to several imports included within the main webpage.

```
const dbStore = firebase.firestore();
firebase
  .auth()
  .createUserWithEmailAndPassword(this.email, this.password)
  .then(cred => {
    return dbStore.collection('users').doc(cred.user.uid).set({
      username: this.username
    })
  })
  .catch(err => alert(err.message))
  this.$router.push('/');
```

(Figure 6)

Figure 6 displays the initial connection to the firebase API. The “.auth()” command will be directed towards the firebase author authentication database. This is where all the encryption occurs for each user account. However, within the “.then()” statement we also store the user information within “firestore database” which is different from the authentication database. The reason for this will be in the next paragraph.



## Authentication

Users Sign-in method Templates Usage

Search by email address, phone number, or user UID					Add user		
Identifier	Providers	Created ↓	Signed In	User UID			
tester123@gmail.com	📧	Mar 31, 2022	Mar 31, 2022	S27Q5sZDy0ZVMR6X6HdHRRb0...			
bestgamer123@gmail.com	📧	Mar 28, 2022	Apr 1, 2022	pNezSFMhclMb7q0zk1pG11xwib2			
test1235@gmail.com	📧	Mar 22, 2022	Mar 28, 2022	qZBOVjVxq6PAx1hkjpSuphu83Rf2			
mytest123@gmail.com	📧	Mar 12, 2022	Mar 28, 2022	D6AgAtLoxNOLFPQT9ccNJ5Vs7V...			
newtest123@gmail.com	📧	Mar 8, 2022	Mar 8, 2022	bPPhuPPD5eYFLUzcF9OyNkLCBbr2			
23@gmail.com	📧	Mar 6, 2022	Mar 6, 2022	RISFQ9qkfUgTbevpcG02PAxelMr2			
Rows per page: 50 1 - 6 of 6							

(Figure 7)

Figure 7 highlights the main authentication database; this will update real-time as users sign into my website. This saves presets as email, creation date, and user UID. The user UID is very important because it is a random unique key generated for each user through firebase, this allows me to refer to each user uniquely within the frontend. As mentioned previously Firebase does not allow developers to save unique saved values such as usernames within the authentication database, instead we must use the Firestore Database to save unique values.

honors-prod-20f03	users	pNezSFMhclMb7q0zk1pG11xwib2
+ Start collection	+ Add document	+ Start collection
users	D6AgAtLoxNOLFPQT9ccNJ5Vs7VG2 S27Q5sZDy0ZVMR6X6HdHRRb0m4o2 pNezSFMhclMb7q0zk1pG11xwib2 qZBOVjVxq6PAx1hkjpSuphu83Rf2	+ Add field skills: ['C++', 'Python', 'C#', 'F...] username: "sgun7"

(Figure 8)

This is evident within figure 8, as it shows the usernames being stored with each unique user. The user's documents column in this are organized by firebase UID, therefore no two users will be the same. This allows us to query each field for each unique firebase

UID. The code for the query will be shown on a different section to highlight the features of that.

### Resume Upload and Parsing

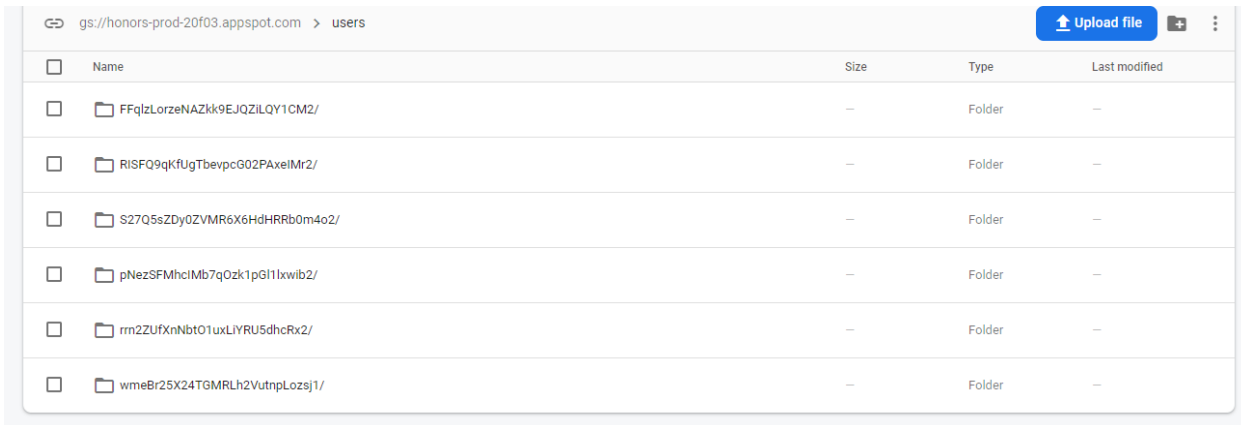


(Figure 9)

As mentioned previously the resume will need to be parsed therefore, we must have some type of upload feature for the user. As far as design goes, I decided to go with a modern day drop design for files. Therefore, when a user clicks on the blue box, they will be directly given access to their file explorer to update their own resumes. One unique thing which occurs within the background is that we store the resume in the backend so it can be referenced for each unique user. Firebase also offers a storage branch which perfectly provides us with an opportunity to directly store each file in an organized manner.

```
const storage = firebase.storage().ref('users/' + user.uid + '/resume.md');  
const storageRef = storage.put(file);
```

(Figure 10)



(Figure 11)

Figure 10 displays the main part of the code which will store the file within the appropriate folder in the storage. First it accesses a large file containing all users titled, “/users”. Then it will obtain the current “user.uid”, it is not as simple as referring to it directly as shown in Figure 10. This will be run by a separate firebase function to obtain the current and updated UID value. Then finally the resume is titled “Resume.md” and stored as a file. Then finally, we run a “storage.put()” to send the object to the database. Also, this will trigger an alert which lets the users know that the file was updated successfully. Figure 11 is a representation of what it looks like within the firebase UI. Just to clarify this would be within the “users” folder containing sub-folders which are appended directly to the UID of the user. Therefore, since these are unique and will never be the same, this provides us with a way to make sure there are never any conflicts and users are not mixing up their own resumes with other users on the same database.

The next section is another key part of my website, this would be the parsing done within my website. As mentioned previously we store the user's resume file within our database, so we can parse it. Another thing to note would be that the website only accepts ".md" files for a resume. There are several reasons why I chose this, I will cover more within the challenges section on why I decided to choose these types of files over things such as PDF's or word Documents. There are also several benefits for ".md" files, they are very suitable with all forms of code and could be directly used within the code base. It is also supported within GitHub, which greatly aids in displaying the data in an organized manner. However, file formats such as PDF's and Word are not supported by GitHub and can lead to confusion.

## OBJECTIVE:

To obtain a Computer Science Co-op Work Assignment and/or Internship

## EDUCATION

The University of Akron, Akron, Ohio 8/2017-Present  
 Currently Pursuing a Computer Science Degree May 2022  
 Overall Grade Point Average 3.80/4.0  
 Dean's List: Fall 2017, Spring 2017, Fall 2018, Spring-Fall 2019, Spring-Fall 2020-2021  
 The University of Akron Honors Scholarship

## Cuyahoga Valley Christian Academy Aug 2012-2017

High School Weighted GPA-4.12  
 Various AP classes taken in High School  
 Member of the School of Engineering|

## Skills:

C++,Python,C#,Frontend

## Hardware

Ability to put basic computer parts together to build the mainframe of a computer. Familiar with various computer parts and other components. Also, have the ability to put together FPV drones.

Software

(Figure 12)

Figure 15 is a common example of a resume file which a user may upload to the website to get parsed. From a visual perspective it might seem as if it is just bolded

text followed by normal text. However, since the file is a “.md” file there are certain characters in front of headers of “.md” files and normal text in “.md” files as well. Most resumes are organized in a way they are titled by headers followed by more precise language such as details about each section. Therefore, my parser will approach it in an organized way such as described above. After it establishes this, it will parse the headers and input them into essentially a JSON dictionary in which the header values are the keys, and the small writings are the key values for each JSON object. While, this part sounds simple, it was difficult to do this with code directly imported from a “.md” file. For example, there were key things to look at such as getting rid of commas or other variables which would affect the query. After spending a lot of time working with this problem and eventually solving it, the whole “.md” file was given to the developer as a JSON object, with specific keys as headers. Therefore, after this is complete, I do a query on a keyword search for the skills of the resume and then I will essentially get the values for those skills into my database. One thing I wanted to expand on would be the keyword search for the skills. In the future, I would like to use a common database of keywords to search directly for skills in the resume to make it more usable with all formats of resumes.

```
var user = firebase.auth().currentUser;
var newRef = firebase.firestore().collection('users').doc(user.uid);

for (let i = 0; i < this.skillObj.length; i++)
{
  newRef.update({
    "skills": firebase.firestore.FieldValue.arrayUnion(this.skillObj[i])
  });
}
```

(Figure 13)

Figure 16 displays this therefore, since one user can have multiple skills, it would not make sense to append to one variable within the Firestore. Thus, we first get the users UID once again to append to unique users' folder. Then, we simply append an array of skills to the user to pull later within the website.

### Viewer

This is a simple section, so after the user has uploaded their resume, they will receive a confirmation alert letting them know that the resume has been updated and added to the backend. However, I also provided a feature for the user to review the resume, to provide a resource for the user to validate their inputs.



(Figure 14)

Now there is a box button, as highlighted within the figure 17, for the viewers to access this page. After the viewer clicks this button, they are guided to the viewer page, which contains simple logic.



**OBJECTIVE:**  
To obtain a Computer Science Co-op Work Assignment and/or Internship

**EDUCATION**  
The University of Akron, Akron, Ohio 8/2017-Present  
Currently Pursuing a Computer Science Degree May 2022  
Overall Grade Point Average 3.80/4.0  
Dean's List: Fall 2017, Spring 2017, Fall 2018, Spring-Fall 2019, Spring-Fall 2020-2021  
The University of Akron Honors Scholarship

**Cuyahoga Valley Christian Academy Aug 2012-2017**  
High School Weighted GPA-4.12  
Various AP classes taken in High School  
Member of the School of Engineering

**Skills:**  
English, Art, Graphics

**Hardware**  
Ability to put basic computer parts together to build the mainframe of a computer. Familiar with various computer parts and other components. Also, have the ability to put together FPV drones.

**Software**  
Familiar with basic Microsoft products such as Microsoft Word, Excel, PowerPoint. Also work with some CAD programs such as SolidWorks and OnShape. Familiar with VideoPad, Lightworks, Adobe Photoshop, and

(Figure 15)

I used a library called “vue-markdown” which is used to convert the “.md” file directly to html. Therefore, with vue being a dynamic framework, this directly allows me to display HTML this will be shown directly as shown in figure 18.

### User page

The last page is also a key page for the website. This will most likely be the page users spend most of their time on clicking other various job links and looking through each job descriptions and names. When the user first access this page, in the background the frontend will do a query for the skills uploaded from their resume. In fact, there is a safeguard inputted in which if the user has not updated their skills they will get a message letting them know that there is no skills on their account and require an upload of resume. When the skills are present a button will appear which will allow users to click to generate their table of results of jobs to look for. I decided to go with a public job API from “Adzuna” which provides me with around certain number of queries

per day. If I wanted to make this website eventually go public, I would need to increase this query amount to allocate for the number of users, however since this is mostly for display the default amount is fine.

```
getResults (element) {
  axios
    .get(`https://api.adzuna.com/v1/api/jobs/us/search/1?app_id=f5c4a827&app_key=b6043bd309878d16edbb90773fddd666&results_per`
    .then(response => {
      var obj = JSON.parse(JSON.stringify(response));
      this.mainData = obj.data.results;
      //this.results = this.mainData
      this.results.push(...this.mainData);
    })
    this.showTable = true;
```

(Figure 16)

Figure 19 highlights some the key code which is used for the API call. I used Axios as my main caller simply because it is used a lot within the industry, and it provides a safe and easy protocol for API calls. First, I get the URL for the API call with the custom constraints such as pulling 4 results per skills and limiting the job search to only US. Each skill is directly inputted into the URL as a keyword for the API call to get the response. Then the response is taken in as an object and converted into a JSON. Lastly, these results are pushed into another array to store the data. The reason for this second array is simply because users can have very varying skills, for example you could be could in English and be skilled in coding therefore it will make sure to query for each of those skills and append to separate arrays. Therefore, if you have four skills appended to the backend, then each skill will generate four jobs for a total of sixteen jobs.



Search			
Display_name	Title	Location	Link
INA Solution	C Developer	New York City, New York	<a href="https://www.adzuna.com/land/ad/2993921288?se=IGv1rHm67BGpVzDifHSYRA&amp;utm_medium=api&amp;utm_source=f5c4a827&amp;v=0A5DFAC45CE946AF009147E1DC4F01018A752F9C">https://www.adzuna.com/land/ad/2993921288?se=IGv1rHm67BGpVzDifHSYRA&amp;utm_medium=api&amp;utm_source=f5c4a827&amp;v=0A5DFAC45CE946AF009147E1DC4F01018A752F9C</a>
Gennex Resourcing	C Programmer	New York City, New York	<a href="https://www.adzuna.com/land/ad/2726476930?se=IGv1rHm67BGpVzDifHSYRA&amp;utm_medium=api&amp;utm_source=f5c4a827&amp;v=8FF5C442728075F1C052013A5E4F063633D42469">https://www.adzuna.com/land/ad/2726476930?se=IGv1rHm67BGpVzDifHSYRA&amp;utm_medium=api&amp;utm_source=f5c4a827&amp;v=8FF5C442728075F1C052013A5E4F063633D42469</a>
May Street Inc	C Developer	New York City, New York	<a href="https://www.adzuna.com/land/ad/2993925066?se=IGv1rHm67BGpVzDifHSYRA&amp;utm_medium=api&amp;utm_source=f5c4a827&amp;v=E2C018A4A43D88447327557CCB116BE45EC01397">https://www.adzuna.com/land/ad/2993925066?se=IGv1rHm67BGpVzDifHSYRA&amp;utm_medium=api&amp;utm_source=f5c4a827&amp;v=E2C018A4A43D88447327557CCB116BE45EC01397</a>
Modis	C Developer	New York City, New York	<a href="https://www.adzuna.com/land/ad/2993928196?se=IGv1rHm67BGpVzDifHSYRA&amp;utm_medium=api&amp;utm_source=f5c4a827&amp;v=C170A18F292FB23268FFBCE7BC96A4CE2997B689">https://www.adzuna.com/land/ad/2993928196?se=IGv1rHm67BGpVzDifHSYRA&amp;utm_medium=api&amp;utm_source=f5c4a827&amp;v=C170A18F292FB23268FFBCE7BC96A4CE2997B689</a>
TapeDeck	Senior Frontend Engineer - REMOTE	US	<a href="https://www.adzuna.com/land/ad/2853145326?se=EH_2rHm67BGpVzDifHSYRA&amp;utm_medium=api&amp;utm_source=f5c4a827&amp;v=636CA96DB360DCE0700EB86FBE8732A2BE4910A4">https://www.adzuna.com/land/ad/2853145326?se=EH_2rHm67BGpVzDifHSYRA&amp;utm_medium=api&amp;utm_source=f5c4a827&amp;v=636CA96DB360DCE0700EB86FBE8732A2BE4910A4</a>
TapeDeck	Lead Frontend Engineer - REMOTE	Portland, Multnomah County	<a href="https://www.adzuna.com/land/ad/2852657079?se=EH_2rHm67BGpVzDifHSYRA&amp;utm_medium=api&amp;utm_source=f5c4a827&amp;v=CAB8265866F4AE6C88B96DB4676D68CBE7B94AEA">https://www.adzuna.com/land/ad/2852657079?se=EH_2rHm67BGpVzDifHSYRA&amp;utm_medium=api&amp;utm_source=f5c4a827&amp;v=CAB8265866F4AE6C88B96DB4676D68CBE7B94AEA</a>
NerdWallet, Inc.	Frontend Architect	California, Campbell County	<a href="https://www.adzuna.com/land/ad/3031211156?se=EH_2rHm67BGpVzDifHSYRA&amp;utm_medium=api&amp;utm_source=f5c4a827&amp;v=27457A4E451AB46F12264294D7A3A9ED859FB588">https://www.adzuna.com/land/ad/3031211156?se=EH_2rHm67BGpVzDifHSYRA&amp;utm_medium=api&amp;utm_source=f5c4a827&amp;v=27457A4E451AB46F12264294D7A3A9ED859FB588</a>
Nexon America Inc	Frontend Engineer	El Segundo, Los Angeles County	<a href="https://www.adzuna.com/land/ad/2993919302?se=EH_2rHm67BGpVzDifHSYRA&amp;utm_medium=api&amp;utm_source=f5c4a827&amp;v=22D3695E767372806FCBC193586AF841CE145F45">https://www.adzuna.com/land/ad/2993919302?se=EH_2rHm67BGpVzDifHSYRA&amp;utm_medium=api&amp;utm_source=f5c4a827&amp;v=22D3695E767372806FCBC193586AF841CE145F45</a>
Morris Heights	Chairperson, Department of	Baltimore County,	<a href="https://www.adzuna.com/land/ad/3035509430?se=ckDsrHm67BGpVzDifHSYRA&amp;utm_medium=api&amp;utm_source=f5c4a827&amp;v=22D3695E767372806FCBC193586AF841CE145F45">https://www.adzuna.com/land/ad/3035509430?se=ckDsrHm67BGpVzDifHSYRA&amp;utm_medium=api&amp;utm_source=f5c4a827&amp;v=22D3695E767372806FCBC193586AF841CE145F45</a>

(Figure 17)

Figure 20 highlights the table which will be displayed for each user. It will contain the display name, title, location, and link for each job which was queried. From the diagram above it seems the users was more skilled towards computers and software; therefore, they got a lot of results of C developers, frontend work, etc. The users will also able also query the data within the table if the resulting table is large.

## Challenges

There were also several challenges which I faced while working on this project. Without going over each detail of the solution and problem, I will briefly highlight them. First was the challenge of simply understanding the material and working with the backend database. As mentioned previously, I was somewhat new to four tier applications, therefore I needed to watch several tutorials and spent weeks studying the material to get grasp of what is going on. After getting over this initial bump, I faced the problem of figuring out the purpose for this website, as mentioned previously the original idea started out as a tutorial website, however it transformed into resume parser

website. This was mostly due to constant meeting with Professor Xiao and discussing the next steps for this process. There were also several coding problems I ran into, but it would be too tedious to mention here. The next challenge was the parsing of resumes, originally, I started with the use of PDF's however it became apparent that PDF parsing was not the best option and would most likely lead to confusions within the API call. Therefore, ".md" file was chosen for this matter. The last challenge was the organization of the backend. This was solved simply by appending things users had within their own profiles all back to the user UID thus creating a unique value for all things which belong to that specific user.

## **Conclusion**

I think the project turned out great there, however I do believe there is room for improvement. Therefore, there are several constraints I wanted to advance and deliver if I continue working on this project. One thing I would like to improve on would be the testing of the website. There are several testing capabilities with frontend work, however due to the time constraints I was not able to accomplish this. While the website works soundly, and I was able to test multiple edge cases there is still possibilities of faults. Another thing I wanted to improve on would also be the diversity of the parser, I eventually want to be able to accept any file type and broaden the keyword search of the parser. This project allowed me to combine my knowledge I have learned throughout the years in college to create a product which can be used in the real world. This project had helped me grow as a developer and truly reflects a large portion of my learning from Akron. This project also provides me with an opportunity to display to potential employers in the future.

## References

- [1] "Introduction - Vue.js." - *Vue.js*, <https://v2.vuejs.org/v2/guide/>.
  
- [2] "Job Search - Find Every Job, Everywhere with Adzuna." Adzuna, 12 Apr. 2000, <https://www.adzuna.com/>.
  
- [3] "Firebase Documentation." Google, Google, 2000, <https://firebase.google.com/docs>.
  
- [4] NGINX, Patrick Nommensen of. "Time to Move to a Four-Tier Application Architecture." NGINX, 10 Sept. 2018, <https://www.nginx.com/blog/time-to-move-to-a-four-tier-application-architecture/>.