CREATING AN EQUITABLE OPTION FOR HIGH SCHOOL INTERNSHIP OPPORTUNITIES: AN ONLINE PLATFORM FOR EFFICIENTLY CONNECTING STUDENTS AND EMPLOYERS

Eiffel Vuong[1], Kevin Jones[2]

[1]Portola High School, 1001 Cadence, Irvine, CA 92618
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

Internships are nearly impossible to find as a highschooler outside of paying for one or having familial connections [1]. A reason for this is that there are only substandard public resources for finding these internships as a highschooler [2]. So, by creating an equitable option through an online website, highschoolers would be able to find internships much better. Within the proposed website would include ways to search/filter for internships, have employers create accounts and create internship posts. Students would also need to quickly be able to browse internship options, which was fixed through adding a list of random internships of different varieties that can be scrolled through on the main page [3]. With this ease of access to obtaining an internship, many highschoolers would be able to find what interests them easily and help them find experiences that are more worthwhile beyond just the skills they learn such as helping with college applications or getting a job [4].

KEYWORDS

Highschool internships, internship platform, Firebase database

1. INTRODUCTION

In Indeed's website, its main purpose is to create a place where employers can create job posts for people to apply to. However, internships are sometimes also shown as posts but normally are only in certain industries with nearly no diversity. My project would try to be more focused on the specific varieties of internships in order to have something for each field of interest instead of having limited options. In SimplyHired's website, they had many internships and gave details about each one. A problem with it however was the fact that the internships searched under highschool internships all ended up coming out to college internships. My project fixes this since the website was created with the intention of only serving highschool students, so most internships wouldn't be able to get confused like SimplyHired's website. In internship.com's website, they also had many internships, however, many of them were very outdated or old. My website already fixes this problem with it's automatic deletion of a post system that is set by the employer.

David C. Wyld et al. (Eds): SIPM, ITCA, CMIT, FCST, CoNeCo, SAIM, ICITE, ACSIT, SNLP - 2023
pp. 41-50, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130905

An online job post website that employers can post internship opportunities for highschool students [5]. By creating an online job post hub, students would be able to access resources to obtain an internship equally. As an online website that has no restrictions, it inherently removes all inequality as anybody can access the resource and helps everyone find a high quality opportunity. I believe it is effective as a solution since this model has already succeeded in creating an equitable method for people looking to find jobs. However the only difference would be that instead of creating the website with the purpose to help people find jobs, it would be created with the intention to find students internships. In addition, the method proposed allows students to find exactly what they would be looking for with ease. Without the creation of this resource, students would've most likely had to be limited to the small selection schools offer or not have gotten an internship altogether. By creating a place for businesses of any type to create internship opportunities for highschoolers, not only can students find a large variety of job types on the site but the employers also find people that they may want to hire in the future. I believe that by essentially creating an online job hub for student internships, students would be more likely to use it rather than public resources that require going through a third-party such as through a counselor or family since it allows the student to be as specific as they'd want when finding the perfect internship for themself.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Find Keywords within Internship Posts

Students will need to be able to search for exactly the types of internships they would want and also make sure that the internship location is close enough to them. I would need to make a way to find keywords within internship posts done by the employers in order to properly filter through the entire list. To do this, I will create a search function that will reference the locations and titles of each internship post in order to try and find words that are the same between the student's search query and the internship.

2.2. The Way to Post Intership Opportunities

Employers will need to be able to post internship opportunities. I could use a form within the website that employers could fill out with the required information in order to create the post. By having the employer just fill out some simple queries such as the internship name, its location, time period, etc, it would allow them to be able to quickly and easily post internship opportunities. Also, I could include ways for the employer to only have the internship opportunity up for a certain amount of time or for a certain amount of people.

2.3. Quickly See a Variety of Internship Opportunities

Students will need to be able to quickly see a variety of internship opportunities and detailed information about them. I could create a chart on the front page of the website that would display random internships of all types. I would have one half be filled with the basic information of each internship and the other would be empty, waiting to be filled with the detailed information of each internship once the student might decide to click on one. I could also use this space as a way for students to search and find opportunities they may like by creating ways to scroll through multiple pages of internship opportunities.

3. SOLUTION

The website's main functions are split into three major components: an internship search service, an employer account system, and an internship creation/authentication system. All the information such as the internships or employer data is stored within the Firebase. I used Firebase as it was the most easily accessible in terms of free databases and due to its ease of use with requests. The website is designed to be as simple as possible where students can just load the page and understand how to navigate it. The front page will display a table with two columns and a search bar, operating similarly to some search engines or Indeed job searching. The user may scroll down to keep viewing more internships in the first column and may choose to click on one of them to open up more information in the second column. If they use the search bar, it will narrow down the amount of internships shown down to only those that fit their filter. They may also click on the more information button in the left column for each internship opportunity in order to take the student to another page where more information about the internship is displayed. Employers may also create their own accounts in order to create jobs by switching on the top navigation bar to their respective pages. Both the job creation and account creation forms utilize forms in order to get the information necessary to create both. The information then is sent to the Firebase where the information is held until requested by the website for either displaying the internship information or for creating a post which requires having an employer account [6].

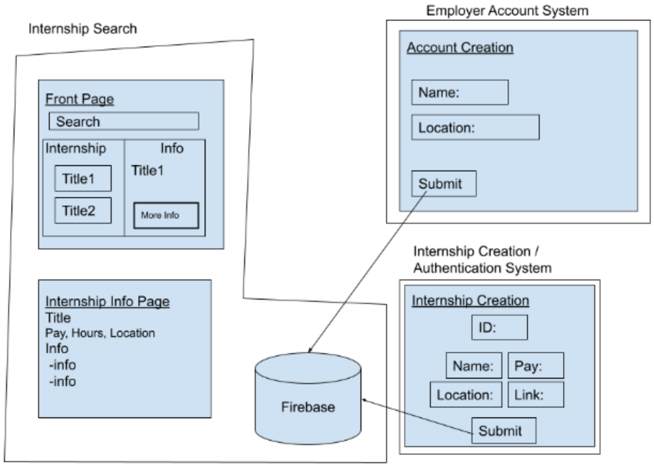
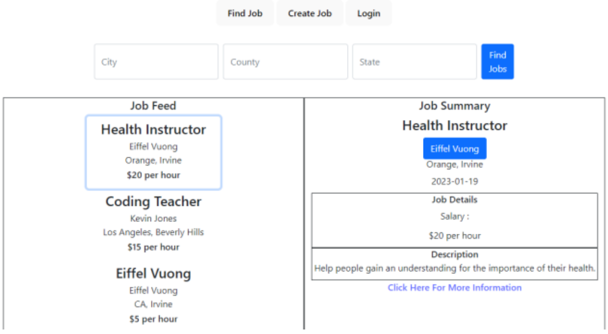


Figure 1. Overview of the solution

An important part to the website's functionality is the internship search service [15]. The service itself is the website's most used and important system since it is the primary way for the users[students] to find internships and apply to them. This search service relies on the usage of a filtering function that relies on using a model that checks internships for keywords that might narrow down the search.



Job Feed

Health Instructor
Eiffel Vuong
Orange, Irvine
$20 per hour

Coding Teacher
Kevin Jones
Los Angeles, Beverly Hills
$15 per hour

Eiffel Vuong
Eiffel Vuong
CA, Irvine
$5 per hour

Job Summary

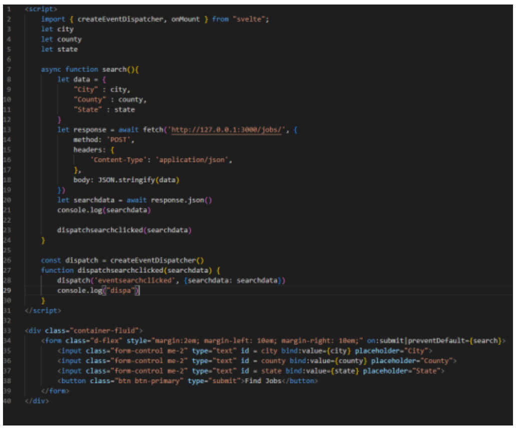
Health Instructor
Eiffel Vuong
Orange, Irvine
2023-01-19

Job Details
Salary :
$20 per hour

Description
Help people gain an understanding for the importance of their health.

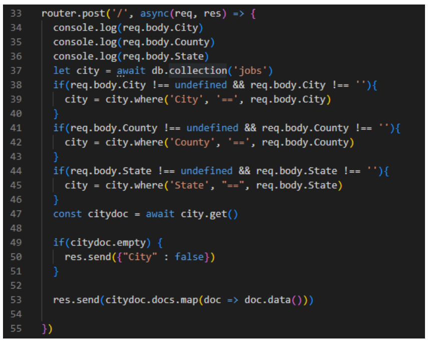
Click Here For More Information

Figure 2. Search job page



```
<script>
    import { createEventDispatcher, onMount } from "svelte";
    let city
    let county
    let state

    async function search(){
        let data = {
            "City" : city,
            "County" : county,
            "State" : state
        }
        let response = await fetch('http://127.0.0.1:3000/jobs/', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json',
            },
            body: JSON.stringify(data)
        })
        let searchdata = await response.json()
        console.log(searchdata)

        dispatchsearchclicked(searchdata)
    }

    const dispatch = createEventDispatcher()
    function dispatchsearchclicked(searchdata) {
        dispatch('eventsearchclicked', {searchdata: searchdata})
        console.log("dispa")
    }
</script>

<div class="container-fluid">
    <form class="d-flex" style="margin:2em; margin-left: 10em; margin-right: 10em;" on:submit|preventDefault={search}>
        <input class="form-control me-2" type="text" id = city bind:value={city} placeholder="City">
        <input class="form-control me-2" type="text" id = county bind:value={county} placeholder="County">
        <input class="form-control me-2" type="text" id = state bind:value={state} placeholder="State">
        <button class="btn btn-primary" type="submit">Find Jobs</button>
    </form>
</div>
```

Figure 3. Screenshot of code 1



```
router.post('/', async(req, res) => {
    console.log(req.body.City)
    console.log(req.body.County)
    console.log(req.body.State)
    let city = await db.collection('jobs')
    if(req.body.City !== undefined && req.body.City !== ''){
        city = city.where('City', '==', req.body.City)
    }
    if(req.body.County !== undefined && req.body.County !== ''){
        city = city.where('County', '==', req.body.County)
    }
    if(req.body.State !== undefined && req.body.State !== ''){
        city = city.where('State', "==", req.body.State)
    }
    const citydoc = await city.get()

    if(citydoc.empty) {
        res.send({"City" : false})
    }

    res.send(citydoc.docs.map(doc => doc.data()))

})
```

Figure 4. Screenshot of code 2

The main component of the front page, the filtering service, is best represented through this function [7]. The svelte page itself is called when someone places something inside one of the inputs, whether it be a city, county, or state and presses the submit button. When pressed, the search function runs and sends a post request to the server where the data is put into a json file. This is done through the variable ‘response’ which holds the post request. In the response, the fetch request with the link allows the data to be sent to the server where it can be managed along with the database data. The other important variable is the ‘data’ variable which holds the user’s search queries, and are sent through along with the post request by being put into the body of the request. In the server, the data sent is then cross referenced with the data in the firebase where it checks whether the search data matches any internship data held within the database, doing this for each piece of information sent by the user. The website waits for the server to send the data back and once it has been retrieved, the data is then sent to the main svelte page by then dispatching the data through the event. With the new search results, the main page displays the user's filtered internships.

Another component is the employer account system. Without the account system, employers would not be able to create accounts in order to publish internships opportunities for students. The system itself relies only on the form that the employer fills out with their data and the backend receiving the data to send to the firebase in order to be stored.

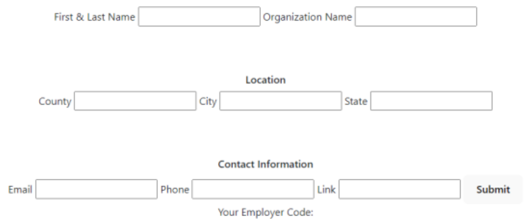
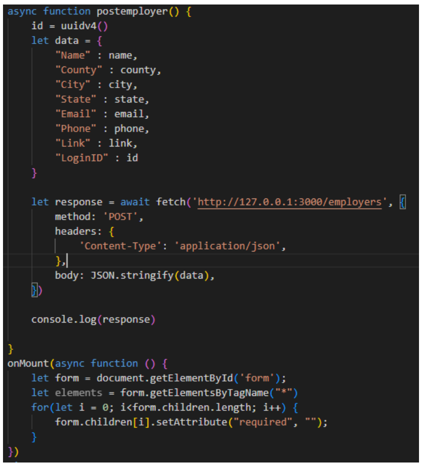


Figure 5. The form that the employer fills out



```
async function postemployer() {
    id = uuidv4()
    let data = {
        "Name" : name,
        "County" : county,
        "City" : city,
        "State" : state,
        "Email" : email,
        "Phone" : phone,
        "Link" : link,
        "LoginID" : id
    }

    let response = await fetch('http://127.0.0.1:3000/employers', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
        },
        body: JSON.stringify(data),
    })

    console.log(response)

}
onMount(async function () {
    let form = document.getElementById('form');
    let elements = form.getElementsByTagName("*")
    for(let i = 0; i<form.children.length; i++) {
        form.children[i].setAttribute("required", "");
    }
})
```

Figure 6. Screenshot of code 3

The form's data that the employers fill out in order to create an account is sent through the function above. The function uses the data of the form through binding each value of the form to their respective variable then creating an array with each value and putting it into the 'data' variable. The function runs when the employer presses the submit button, which then initiates the 'response' variable that posts to the database with the data just retrieved from the form. On the server side, the data from the front end is received which triggers the router to go to the post function. Since there isn't any parameters within the route, it gets sent to the one shown in the code where the data is then added to the firebase by adding it directly into the employers collection of the database. After adding the data the server sends a response to confirm the data has been added and adds along the employers' specific code that they can use in order to create internships through the internship creating system.

The final component is the internship creation system. The system's purpose is to allow employers to be able to create internship opportunities for students to see and apply to [14]. The creation relies on the employer filling out a form and having the data being sent to the server side in order to be processed then sent back to the front end in order to be displayed on the front page for students to see.

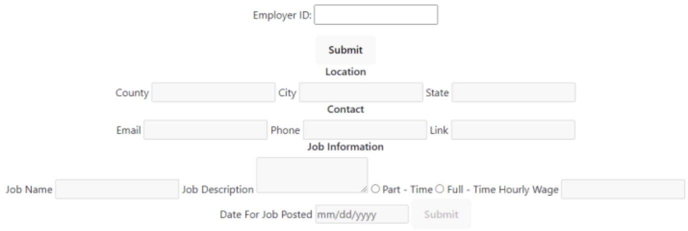
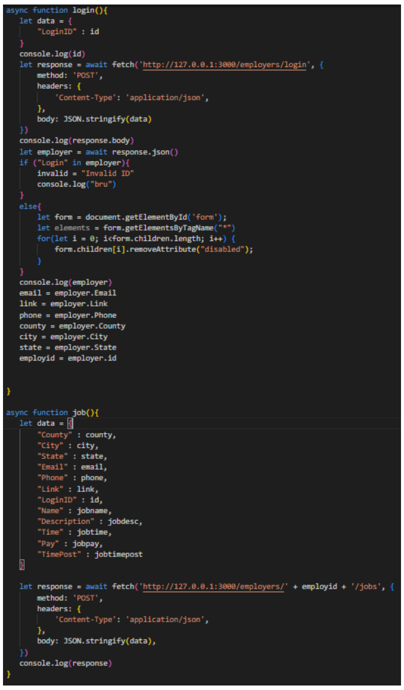
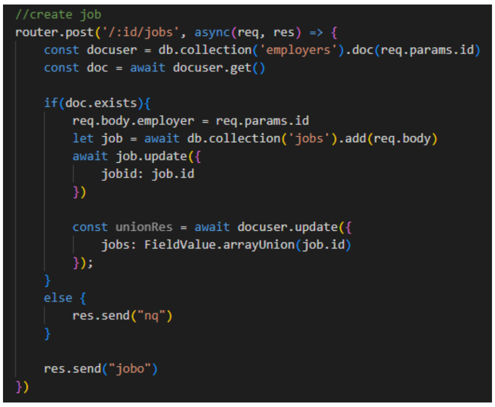


Figure 7. Intership creation system



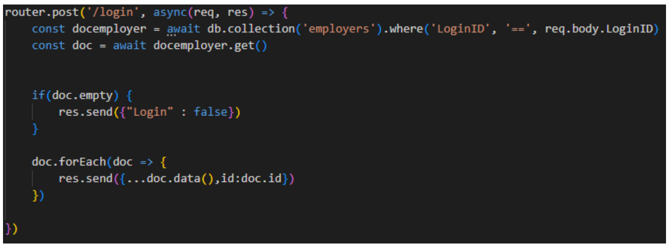
```
async function login(){
    let data = {
        "LoginID" : id
    }
    console.log(id)
    let response = await fetch('http://127.0.0.1:3000/employers/login', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
        },
        body: JSON.stringify(data)
    })
    console.log(response.body)
    let employer = await response.json()
    if ("Login" in employer){
        invalid = "Invalid ID"
        console.log("bru")
    }
    else{
        let form = document.getElementById('form');
        let elements = form.getElementsByTagName("*")
        for(let i = 0; i<form.children.length; i++) {
            form.children[i].removeAttribute("disabled");
        }
    }
    console.log(employer)
    email = employer.Email
    link = employer.Link
    phone = employer.Phone
    county = employer.County
    city = employer.City
    state = employer.State
    employid = employer.id


}

async function job(){
    let data = {
        "County" : county,
        "City" : city,
        "State" : state,
        "Email" : email,
        "Phone" : phone,
        "Link" : link,
        "LoginID" : id,
        "Name" : jobname,
        "Description" : jobdesc,
        "Time" : jobtime,
        "Pay" : jobpay,
        "TimePost" : jobtimepost
    }

    let response = await fetch('http://127.0.0.1:3000/employers/' + employid + '/jobs', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
        },
        body: JSON.stringify(data),
    })
    console.log(response)
}
```

Figure 8. Screenshot of the code 4



```
//create job
router.post('/:id/jobs', async(req, res) => {
    const docuser = db.collection('employers').doc(req.params.id)
    const doc = await docuser.get()

    if(doc.exists){
        req.body.employer = req.params.id
        let job = await db.collection('jobs').add(req.body)
        await job.update({
            jobid: job.id
        })

        const unionRes = await docuser.update({
            jobs: FieldValue.arrayUnion(job.id)
        });
    }
    else {
        res.send("nq")
    }

    res.send("jobo")
})
```

Figure 9. Screenshot of the code 5



```
router.post('/login', async(req, res) => {
    const docemployer = await db.collection('employers').where('LoginID', '==', req.body.LoginID)
    const doc = await docemployer.get()


    if(doc.empty) {
        res.send({"Login" : false})
    }

    doc.forEach(doc => {
        res.send({...doc.data(),id:doc.id})
    })

})
```

Figure 10. Screenshot of the code 6

The login and job functions are the best representations of this component. The login function serves as the authentication service and runs when an employer types in their employer code that they should've received from the employer account creation system. When the login function is called, the id the employer put in is then sent in a post request to the server. On the server side, the post route with the login parameter is called. In it, the variable 'doc' is created through finding where in the database (Firebase) that the login ID given by the employer is the same as an ID in the database [8]. If there are no matches, the server sends a response saying that the login attempt failed. However if there is a match, the server sends back the data of the matched ID. In the front end, once the response has been received and the login is not invalid, the rest of the form under the login ID gets the 'disabled' parameter removed through the variable 'form' which holds all of the elements of the creation form. By looping through each element of the form, the form becomes available to be filled in. Once the employer fills the form in and presses the submit button, the job function runs. Then, it sets the data from the form into the variable 'data' and sends a post request into the employers in order to add the internship data into the firebase.

4. EXPERIMENT

To ensure that the internship search engine website is fully functional and user-friendly, an experiment was conducted involving15 participants. This sample size was chosen as it is reasonable enough to account for any variability. The participants were asked to navigate through the webpage and test its features for at least ten minutes. The features tested included the application process, the search function, and the events page. After testing, participants were provided with a survey link on Google Forms [9]. This allowed the participants to rate the functionality and convenience of the application on a scale of 1 to 10. An optional free-response section was located at the bottom of the survey, which allowed participants to share any additional thoughts.

Based on the table and chart below, the overall feedback for both the functionality and convenience of the website was positive. The functionality had an average rating of 8.13, with a highest rating of 10 and a lowest rating of 5. Similarly, convenience had an average rating of 7.67, with a maximum rating of 10 and a minimum rating of 5. Although the ratings for functionality were higher than convenience, the feedback from the optional free-response section suggested that there were some issues with the events page. It seems that a few participants had trouble accessing or viewing the events page. Despite this, most participants were able to use the application without any issues.

The results of the experiment indicated that the application was successful in its primary purpose of searching for internships. The vast majority of participants rated the functionality as 5 or higher out of 10, which falls within expectations. The features were also tested and revised multiple times before the experiment was conducted. Based on the positive feedback for the convenience and intuitiveness of the application's interface, it seems that the implementation of the features contributed directly towards the website's primary purpose. However, there was feedback from one participant indicating that the interface could use more decoration to be more visually appealing to users. After considering this feedback and identifying the specific parts of the interface that did not work, improvements were brainstormed to make the interface more user-friendly.

Overall, the experiment provided valuable feedback on the functionality and convenience of the internship search engine website [10]. The positive feedback from most participants suggests that the website is an effective tool for finding internships. The feedback from the optional free-response section will also help the developers to address the issues encountered by some participants and improve the website's functionality and convenience even further.

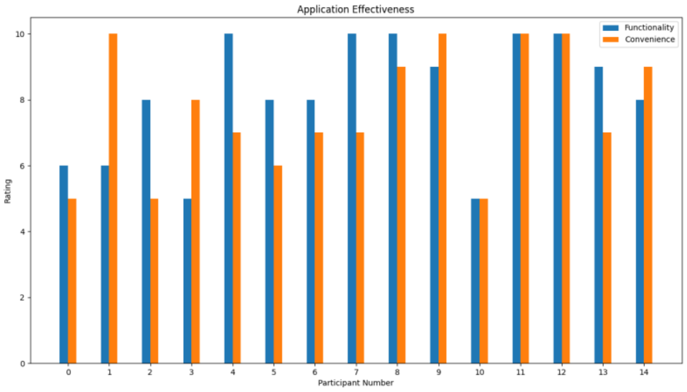
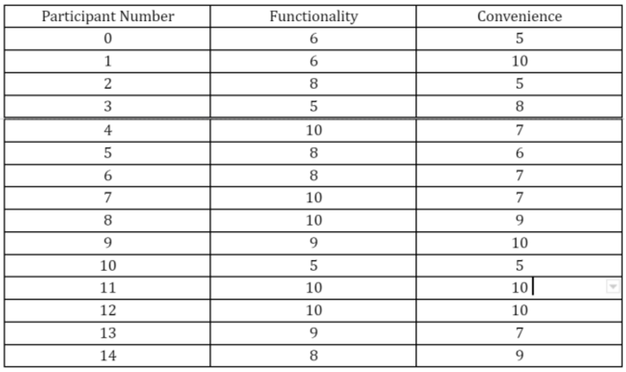


Figure 11. Application Effectiveness



Participant Number	Functionality	Convenience
0	6	5
1	6	10
2	8	5
3	5	8
4	10	7
5	8	6
6	8	7
7	10	7
8	10	9
9	9	10
10	5	5
11	10	10
12	10	10
13	9	7
14	8	9

Figure 12. Table of experiment

5. Related Work

The website Indeed also has an option for finding high school internships [11]. Although they don’t have a specific section for it, people submit job posts but under the title they specify that the post is meant for high school students looking for internships. Although there are a lot of internship posts, many are specified only for the food industry, making the options extremely limited. Without a specific filter for high school internships, employers are forced to specify in their titles and descriptions. With my project, everything is designed specifically for finding internships so employers or students don’t have to worry about parsing through any other job posts. Also, with my website, more internships would be from other industries besides just the food, allowing a greater range of types of internships for students to find.

Another website that helps students find internships is SimplyHired [12]. The website includes a wide range of options for choosing recent internships. However, many of them aren't geared towards high school students and are actually meant for college students. This would be fixed in my website since it's specifically geared towards creating high school internships, so the mistake of mixing high school and college internships together. This website ignores the difference between the two and just places both when searching for high school internships. This can be seen with how each internship includes salaries and high pays, which are normally only shown with college internships.

The website internships.com includes many highschool internships as a main part of their website [13]. They have a wide range of options and include all the information necessary to understand what the internship is about. However, one limitation is that most of the internships are outdated and many if not all are over 30 days old, showing how the posts may not even be available anymore. In my website, each internship post has a set timer that the employer sets themself for when the post goes down, preventing this problem of posts that are so old that students wouldn't want to apply to them due to the fear that the internships are no longer available.

6. CONCLUSIONS

One limitation is the fact that there isn't any way to directly apply to internships on the website. Currently, internship posts only include information on the internship and the contact information of the employer in order to contact them with the student's interest. I believe that this should be fixed through creating the form within the website and making it possible for employers to see on the website who has applied to their internship and their data. I also believe on top of this, the ability to edit and delete posts at any time should be implemented as it would allow for more flexibility with the employer's posts and allow them to delete them once they have enough applicants.

Although there are many components of this project that can be improved and added, I believe that the core components itself can make the project good enough to be implemented almost anywhere. Through this project, I believe high school students will now all be able to equally get internship opportunities.

REFERENCES

[1] Coco, Malcolm. "Internships: A try before you buy arrangement." SAM Advanced Management Journal 65.2 (2000): 41.

[2] Murillo, Marco A., Karen Hunter Quartz, and Jaime Del Razo. "High school internships: Utilizing a community cultural wealth framework to support career preparation and college-going among low-income students of color." Journal of Education for Students Placed at Risk (JESPAR) 22.4 (2017): 237-252.

[3] Knouse, Stephen B., John R. Tanner, and Elizabeth W. Harris. "The relation of college internships, college performance, and subsequent job opportunity." Journal of employment Counseling 36.1 (1999): 35-43.

[4] Ryken, Amy E. "The holding power of internships: Analyzing retention in a school-to-career program." The Community College Enterprise 10.2 (2004): 37.

[5] Faberman, R. Jason, and Marianna Kudlyak. "What does online job search tell us about the labor market." Economic perspectives 40.1 (2016): 1-15.

[6] Khawas, Chunnu, and Pritam Shah. "Application of firebase in android app development-a study." International Journal of Computer Applications 179.46 (2018): 49-53.

[7] Thomas, Kurt, et al. "Design and evaluation of a real-time url spam filtering service." 2011 IEEE symposium on security and privacy. IEEE, 2011.

[8] Thomas, Kurt, et al. "Design and evaluation of a real-time url spam filtering service." 2011 IEEE symposium on security and privacy. IEEE, 2011.

[9] Vasantha Raju, N., and N. S. Harinarayana. "Online survey tools: A case study of Google Forms." National conference on scientific, computational & information research trends in engineering, GSSS-IETW, Mysore. 2016.

[10] Hergert, Michael. "Student perceptions of the value of internships in business education." American Journal of Business Education (AJBE) 2.8 (2009): 9-14.

[11] Walmsley, Andreas, Rhodri Thomas, and Stephanie Jameson. "Internships in SMEs and career intentions." Journal of Education and Work 25.2 (2012): 185-204.

[12] Diamond, Michael, and Angela Mattia. "Data visualization: An exploratory study into the software tools used by businesses." Journal of Instructional Pedagogies 18 (2017).

[13] Schawbel, D. "Millennial Branding and Internships. com Release First Ever Study on High School Careers." News release, February 3 (2014).

[14] Van Over, David, and Byron Dangerfield. "Student internship in information systems: Creating opportunities and solutions." Journal of Information Systems Education 5.4 (1993): 34.

[15] Huppertz, T., P. F. Fox, and A. L. Kelly. "The caseins: Structure, stability, and functionality." Proteins in food processing. Woodhead Publishing, 2018. 49-92.