# Student Career Recommendation System Using Content-Based Filtering Method

1st Adib Hakimi Abdul Rashid
*Northport (Malaysia) Bhd*
*Jalan Pelabuhan, Pelabuhan Utara,*
42000 Port Klang, Malaysia
adibhakimi16@gmail.com

2nd,* Masurah Mohamad
*Faculty of Computer and Mathematical Sciences*
*Universiti Teknologi MARA Perak Branch*
Tapah Campus, Perak, Malaysia
masur480@uitm.edu.my

3rd Suraya Masrom
*Faculty of Computer and Mathematical Sciences*
*Universiti Teknologi MARA Perak Branch*
Tapah Campus, Perak, Malaysia
suray078@uitm.edu.my

4th Ali Selamat
*Malaysia Japan International Institute of Technology (MJIIT)*
Universiti Teknologi Malaysia
Jalan Sultan Yahya Petra, Kuala Lumpur, Malaysia
aselamat@utm.my

2022 3rd International Conference on Artificial Intelligence and Data Sciences (AiDAS) | 978-1-6654-9164-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/AIDAS56890.2022.9918766

***Abstract*—The most frequent difficulty that graduates face is finding a suitable profession. Finding a job after graduation might be challenging for students who are unsure about their future goals. The goal of this project is to create a system that only recommends careers for students studying computer science. Web scraping was used to extract the career information from the JobStreet website. The recommendation is made using a content-based filtering technique that compares one thing to another based on the user's preferences. The user can easily make recommendations using the system's user-friendly interface and straightforward instructions. Using a specialised functionality tester, this system was put through its paces and more career opportunities will be offered to career vacancy websites in the future. In addition, the system will be more sophisticated when it comes to narrowing down the pool of potential careers that the user can choose from, and it will be directly linked to career page websites to guarantee that all open positions are still open for the user to apply for.**

***Index Terms*—Career, content-based filtering, recommendation, computer science.**

## I. Introduction

Finding suitable work after graduation is the biggest problem that students have. When someone is looking for a job, it can be difficult because they are unsure of their goals and the direction they want to go in. Artificial intelligence, computer systems and networks, security, database systems, and human-computer interaction are just a few of the many areas of study in computer science that offer students a variety of career options, including software development, database administration, computer hardware engineering, systems analysis, and many more. Recommendation systems have become a common tool for internet users and give them the assistance they need in locating information as a result of the growing volume of material available online. Given the availability of this information online, job searchers require almost immediate access to pertinent career prospects. However, for many applicants, sorting through thousands of posts to choose a few that are pertinent might be a tiresome procedure [1].

Although there are many career recommendation systems on the internet, those recommendation systems are not specifically for the computer science area. If graduates in computer science use those systems, they may get a recommendation for a career that is not related to their studies. Besides, most career recommendation systems on the internet may recommend a career that is not available or offered in Malaysia since some of the systems are being developed by outsider developers. Furthermore, dynamic career recommendation systems are difficult to obtain on the internet and most of the systems do provide a personality test that recommends a career solely based on the user's personality and interests.

To overcome these issues, this project is developed to help the computer science students in choosing a career that suits their interests and qualifications especially in computer science's field. Moreover, this project is not only focusing on finding a suitable career for individuals, but it will also recommending the highest-rated career in the scope of Computer Science in Malaysia specifically.

The following sections explain the literature review, methodology, the proposed system and also the results discussion.

## II. Literature Review

This section provides the reviews of previous researches relating to the recommendation system. It also compares various techniques used in the development of a recommendation system with a few existing systems.

### A. Recommendation System

A recommendation system is a program that utilizes efficient algorithms to predict users' interest in the purchase of an item [2] and bridges the gap between data collection

and analysis by filtering all available data to show only the information that is most useful to the user [3], [4]. In 2019, Mohamed stated that many items can be purchased online, and customers are increasingly demanding that a large number of items available on websites be filtered so those unique items can be identified more conveniently based on their actual interests [5].

To create an efficient recommendation engine, recommendation systems use various forms of filtering, such as collaborative [6], content-based [7], and hybrid [8]. If people purchase an item from the website, they will be shown additional items based on the quality and specifications of the item that they purchased. Besides, the recommendation system gives the benefit in revenue, client satisfaction, personalization, discovery, and provide reports [5].

*1) Collaborative Filtering:* Collaborative filtering is a recommendation system approach that focused on the assumption that [6] an item will be recommended [9] similarity among a group of users or items to predict missing ratings then make appropriate recommendations. According to Geetha [8], collaborative filtering is based on user similarity rather than item similarity, thus it generates more powerful recommendations.

Collaborative filtering can be divided into several methods is Memory-based Collaborative filtering, Model-based Collaborative filtering, and Hybrid Collaborative filtering [6]. Memory-based algorithms use the similarity of other users or items to the target user or item to predict new ratings based on the available data (that is stored in the memory), Model-based Collaborative filtering algorithms use a range of techniques to identify patterns in the data and learn a model for predicting new ratings on the training set [10], and Hybrid collaborative filtering is a recommendation system methodology that incorporates collaborative filtering with other approaches, such as content-based, to balance the advantages and disadvantages of each process and increase the efficiency of recommendation systems [11].

*2) Content-based Filtering:* Content-based filtering is based on an item's description [12] and a preference profile of the user [5]. From all of the information provided by the users, these systems will create a list of item profiles for them [9]. In simple word, this method will compare previously rated items and suggests the best match [8]. In Content-based filtering, there are three process steps [5]. First is the content analyzer. To move item representation from the original data space to the target one, feature extraction methods are used to evaluate data items. The second is profile learner, This model creates a user profile by gathering data on user preferences and attempting to generalize it [13]. The third, filtering component, this approach uses user data profile to discover matching items related to the item list, but in new item and show this new item to the user [14]. Some of the popular method that being applied in content-based filtering is term frequency-inverse document frequency (TF*IDF) and vector space model.

*3) Hybrid filtering:* This approach combines collaborative filtering techniques with content-based filtering techniques [5], [8], [9]. Priyanga and Kamal [12], claim that some of the most common problems in recommendation systems, such as cold start and sparsity, may be addressed with these techniques. Weighted hybridization incorporates the results of many separate recommendation approaches. Depending on the case, the hybrid solution uses a number of recommendation approaches [9]. A feature combination is a recommendation algorithm that incorporates features from multiple recommendation data sources into a single algorithm [15]. Cascade recommendation consists of a set of guidelines, which are defined by the others [16]. Feature augmentation is where the output of one technique is used as an input element for another. A model learned by one recommendation serves as feedback to another in a meta-level context [5].

### *B. Existing career recommendation system*

There have been numerous career recommendation systems developed in recent years. This project is similar to three other career recommendation systems.

*1) Recommendation System of Information Technology Jobs using Collaborative Filtering Method Based on LinkedIn Skills Endorsement:* This system is developed for Informatics Engineering students where a classification of IT skills is needed to determine the best career path. 400 data from IT experts will be used as a guide for students and will be accessed from their LinkedIn accounts. The K-Means Clustering algorithm is used to decide how feasible it is to use data from IT professionals as a reference. Then, based on the proximity of student skills to information technology career fields, classification is calculated using the K-NN algorithm's Collaborative Filtering method. The result of this system will show a list of recommended careers based on the student's skills [6].

*2) PCRS: Personalized Career-Path Recommendation System for Engineering Students:* This system examines the personal and academic characteristics of high school students in order to assist them in choosing an academic field and a potential career. PCRS is concentrating on engineering higher education in Palestine because it is one of the most common faculties among high school students. Artificial intelligence is used in this method to assess the best fit between the engineering discipline, the student's personality, and academic profile. PCRS is built on the fuzzy intelligence of N-layered architecture and creates a structure based on academic achievement, personality style, and extracurricular skills. The relationship between personality type and engineering discipline was developed using a sample of 1250 engineering students from An-Najah National University who were enrolled in seven engineering disciplines. The device is demonstrated in a mobile application that is tested by a group of 177 engineers. The sample consisted of graduate or undergraduate engineers who were satisfied with their engineering disciplines. According to the author, social-economic factors such as employment rates, economic situation, and parents' context may improve the recommendations, particularly in developing countries like Palestine [17].

*3) Generating unified candidate skill graph for career path recommendation:* This system is very basic and easy to use

since it uses an Open IE pipeline to extract education and experiences from a user's profile in pdf format or from other public data sources. However, the system simply shows the skill graph and does not create a list of recommendations, which would be more direct and beneficial to the user [18].

## III. Methodology

The proposed system is developed using HTML and PHP, and the database is stored using MySQL. It is constructed based on the framework development process as depicted in Figure 1 and consisted of five main phases. The phases are data extraction, data cleaning, web-based system development, user-profile creation and career recommendation.

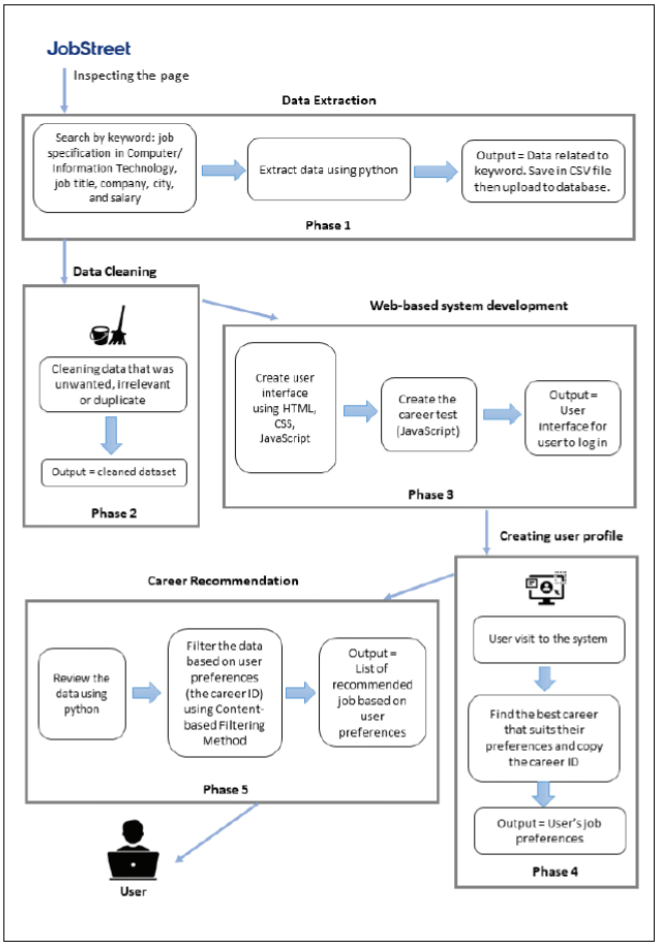


Fig. 1. System architecture

In phase 1, web scraping is used to collect information about available careers from careers websites on the internet, with a focus on Malaysia. The data from the career websites such as Jobstreet will be scraped using Python. Jupyter Notebook software is used to apply the Python language. The Python language makes use of two libraries: i. BeautifulSoup - a Python library that allows you to parse HTML and XML files. It generates parse trees, which are useful for quickly extracting data. ii. Pandas - an open-source data manipulation and analysis package. It's used to extract data and save it in the format that you want. After extracting the data, the dataset will then be saved in the database and utilized anytime the user logs into the online system.

The data cleaning procedure is done in phase 2. Data cleaning is the process of removing undesirable, irrelevant, or redundant data. Microsoft Excel will be used to clean the data. Any errors or null values in each column and row will be corrected and replaced. The data cleaning process guarantees that the data that will be used is clean and correct by identifying errors in the data. In order to avoid an error when processing the data, the inaccurate or defective data must be deleted or rectified.

In phase 3, Visual Studio Code is used to create the web-based system, which is written in HTML and CSS. Visual Studio Code has all of the functionality that a programmer would expect from a code editor. In phase 4, the user must explore each category in the Computer Science area and read the overview of each path. The five pathways that the user may be interested in are multimedia, development, data analysis, security, and networking. After determining the path they believe is ideal for them, users should review the available career list and filter it based on the characteristics of careers they prefer. Then they must select a preferred career and copy the career ID. The user must then navigate to the recommendation page and enter the career ID into the provided input box.

In phase 5, the content-based filtering method filters the user's preferred career based on the career ID that they enter, and the system will compare it to the available careers in the system, resulting in a list of available Computer Science careers that are suited for the user's interests and preferences. This method is entirely based on comparing user preferences to product attributes. The products that have the most features in common with the user's interests are the ones that are recommended. In order to apply the content-based filtering method, Python language is needed. Jupyter Notebook is one of the most popular software that is suitable to develop the recommendation system using python language. Three steps to apply the content-based filtering to recommend a list of careers which are data review using Python, implement TF-IDF content-based filtering method and vector space model and embed python code to visualize the obtained results.

The following subsections explain the implementation of TF*IDF and vector space model in constructing the recommendation engine.

*1) The implementation of TF*IDF:* The number of times a word appears in a document is known as TF (term frequency). Meanwhile IDF is the inverse document frequency (IDF) of a word that will measure of its importance in the whole corpus. The TF*IDF technique is used to weigh a term in a document and give its relevance depending on how many times it appears in the document. The phrase is rarer and more essential if the TF*IDF score (weight) is greater, and vice versa. Figure 2 demonstrates the formulation of TF*IDF equation (EQ 3) which is derived from equation 1 (EQ 1), equation 2 (EQ 2).

$$TF(t) = (Number\ of\ times\ term\ t\ appears\ in\ a\ document) / (Total\ number\ of, terms\ in\ the\ document). \quad \text{(EQ 1)}$$

$$IDF(t) = log\_e\ (Total\ number\ of\ documenT / Number\ of\ documents\ with\ term\ t\ in\ it). \quad \text{(EQ 2)}$$

$$\boldsymbol{W_{x,y} = tf_{x,y} \times log\left(\frac{N}{df_x}\right)} \quad \text{(EQ 3)}$$

$tf_{x,y} = frequency\ of\ x\ in\ y$
$df_x = number\ of\ documents\ containing\ x$
$N = total\ number\ of\ documents$

***TF – IDF***
*Term x within document y*

Fig. 2. TF*IDF equation

*2) Vector space model:* Using vector space model, each product is kept as a vector of its characteristics in an n-dimensional space, and the angles between the vectors are computed to indicate their similarity. Following that, user profile vectors are built based on the career ID that the user has chosen, and the similarity between an item and a user is calculated in the same way. Cosine similarity is a measure of how similar two vectors are. It can be used on objects in a dataset to show their closeness to one another using keywords. To determine the distance between two vectors, divide the dot product by the magnitude value (A and B). To obtain the data into the format required for cosine similarity, Pandas and NumPy in the scikit learn library is utilized. After all of the information is gathered, the system will generate a list of open careers in Malaysia that match their personality, interests, and preferences.

*3) The Python codes for content-based filtering process:* The Anaconda is utilized as the local host to start the Jupyter Notebook to conduct content-based filtering. Jupyter Notebook will create the recommendation process in Python. The function used for implementing the content-based filtering is depicted in Figure 3.

| Code | Function |
|---|---|
| TfidfVectorizer | To calculate the TF-IDF score word-by-word for each document's description. |
| tfidf_matrix | the matrix with each word and its TF-IDF score in relation to each document |
| cosine_similarities | To compute the similarity of each item in the dataset to every other item |
| cosine_similarities[idx][i], job_df['id'][i]) for i in similar_indices | The data was classified into different groups based on their similarity to item i and the values were saved in the results. |
| recommend(item_id=8011, num=10) | Input the career id and the number of recommendations desired, and the function will gather the results[] matching to that career id, displaying the user's recommendations on the screen. |

Fig. 3. Python codes for content-based filtering process

## A. Overall steps of the proposed system

Figure 4 depicts the whole steps of developing the project. The first step is to use web scraping to obtain data from the Jobstreet website. To scrape the data, the BeautifulSoup and Pandas libraries are implemented in Python. After the data has been cleaned and stored in the database, the user interface is created so that the user may view a list of possible careers.

The user must then select one career that matches their preferences and enter the career ID into the recommendation page to get the system to recommend another 10 related careers for the user to choose from. This is where the content-based filtering technique comes into play. The TF-IDF algorithm and cosine similarity are used to filter career information that is comparable to the user preferences. After receiving the list of recommended careers, the user can return to the career list page to see if the career is still available.

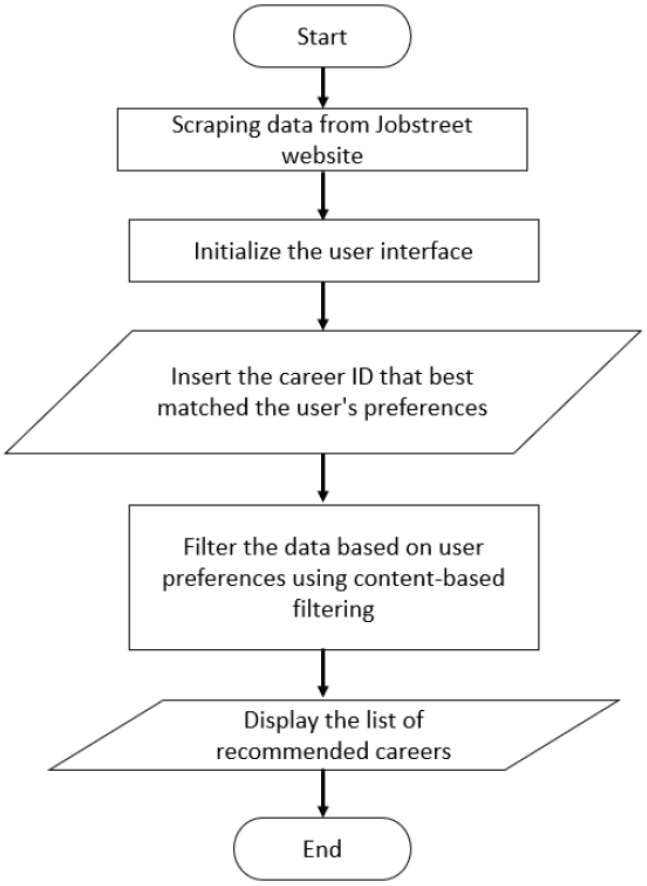


Fig. 4. Flowchart of the overall process

## IV. Results and Discussion

To test the performance of the system, black box functional testing and usability testing are being implemented. The evaluation process involved the system's interface features and

the core logic implementation which was done by the expert. Meanwhile the usability test is used to examine how well users understand the system and to verify that it is easy to use by using a system usability scale (SUS). A survey that includes 10 items and five responses that range from highly agreed to strongly disagree is carried out among 10 final year students in computer science department [19].

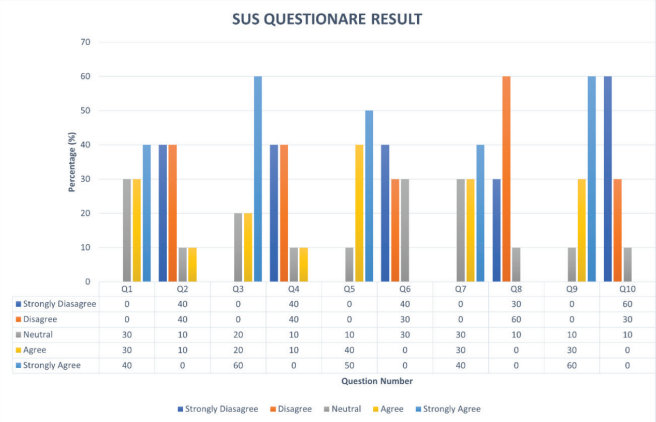


Fig. 5. SUS usability testing results

Figure 5 depicts a summary of the usability testing results. It can be seen that for Question 1 (Q1), 40% of the respondents want to use the system regularly, and none of them disagree. Another 40% of respondents were strongly disagree with the statement that the system is complex (Q2), and 40% choose to disagree as well. The system is easy to use (Q3), according to 60% of the respondents. Question four asks if the user would require the assistance of a technical person in order to operate this system, and 40% strongly disagree and 40% also disagree.

The system's various functions were nicely integrated (Q5), according to 50% of respondents and 40% of the respondents strongly disagree that there is too much inconsistency (Q6). Moreover, most users would learn to utilize this system quickly (Q7), according to 40% of the respondents. 60% of respondents disagree that the system is difficult to use (Q8). 60% of respondents said they were highly confident in their ability to use the system (Q9). Finally, 60% of respondents strongly disagree that they had to learn a lot of things before getting started with this system (Q10).

To get the SUS score, the score contributions from each item will be added. Each item's score will contribute between 0 and 4 points. For items 1, 3, 5, 7, and 9, the scale position minus 1 contributes to the score. For items 2, 4, 6, 8, and 10, the contribution is a 5 minus scale position. To get the overall value of SUS, the sum of the scores will be multiplied by 2.5. The SUS scale ranges from 0 to 100. A SUS score of 68 is considered average. If the score is less than 68, the application is regarded as below average, and there are likely major issues with it. It is considered above average if the score is more than 68. The sum of the 10 respondents' scores is 812.5, which is divided by the number of respondents to provide an average SUS score of 81.25 percent. Therefore, with an average score of more than 68, this system has a high degree of usability. Figure 6 presents the SUS questions that have been used to evaluate the system's usability.

| Scale 1 – Strongly Disagree, 2 – Disagree, 3 – Neutral, 4 – Agree, 5 – Strongly Agree | | | | | | | |
|---|---|---|---|---|---|---|---|
| No | Question | Scale | | | | | |
| 1 | I think that I would like to use this system frequently. | 1 | 2 | 3 | 4 | 5 | |
| 2 | I found the system unnecessarily complex. | 1 | 2 | 3 | 4 | 5 | |
| 3 | I thought the system was easy to use. | 1 | 2 | 3 | 4 | 5 | |
| 4 | I think that I would need the support of a technical person to be able to use this system. | 1 | 2 | 3 | 4 | 5 | |
| 5 | I found the various functions in the system were well integrated. | 1 | 2 | 3 | 4 | 5 | |
| 6 | I thought there was too much inconsistency in this system. | 1 | 2 | 3 | 4 | 5 | |
| 7 | I imagine that most people would learn to use this system very quickly. | 1 | 2 | 3 | 4 | 5 | |
| 8 | I found the system very cumbersome to use. | 1 | 2 | 3 | 4 | 5 | |
| 9 | I felt very confident using the system. | 1 | 2 | 3 | 4 | 5 | |
| 10 | I needed to learn a lot of things before I could get going with this system. | 1 | 2 | 3 | 4 | 5 | |

Fig. 6. The System Usability Scale (SUS) Questions

The following figures 7 and 8 demonstrate several screenshots of the proposed system. Figure 7 is the Career list page that displays the record of career according to the specified categories such as multimedia, development, data analysts, security and networking. Figure 8 presents the recommendation page for the user to find the suitable career that match with their requirements. The page will display only ten most similar carriers with the input that being entered by the user together with the offered salary.

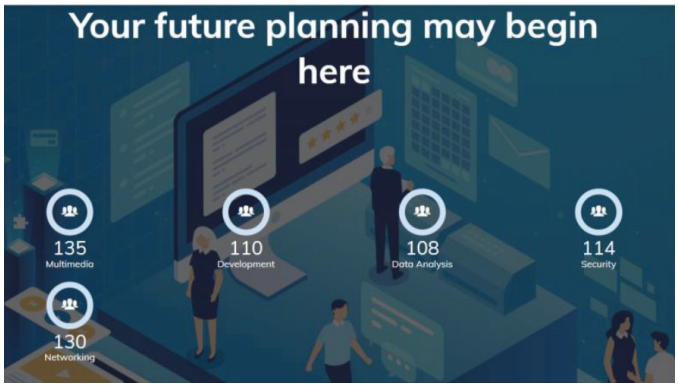


Fig. 7. Career list page

## V. Conclusion

The aim of this project is to create a career recommendation system that will assist Computer Science students in deciding their career route after graduation. After graduation, they can choose from a range of career fields, including networking, multimedia, security, and many more. Therefore, this system is focused on Computer Science topics and exclusively advises

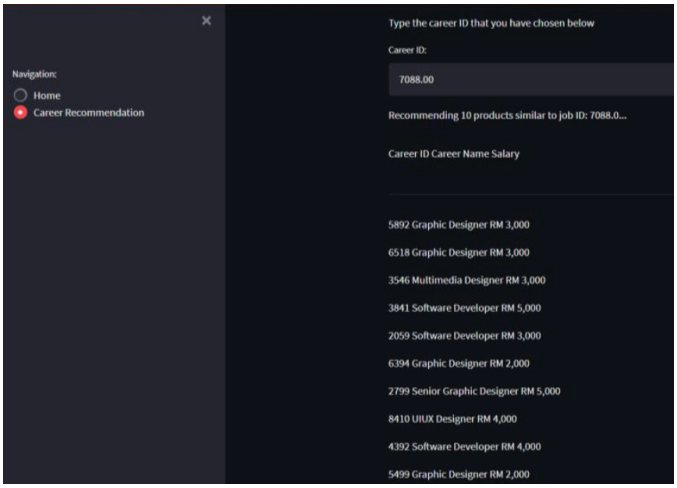


Fig. 8. Career recommendation page

Computer Science-related careers. Even though the proposed system is well-functioning and has a high degree of usability, it also has limitations. The system only recommends careers that are accessible within the specified time. Therefore, the possibility of the career has been filled up by another candidate is high. Moreover, the careers that are available in the system are not being update periodically. It relies on the developer to update the latest categories into the system's database.

Therefore, a few improvements are suggested such as the system should be linked directly with the JobStreet website to ensure that the list of careers is still available to be applied by the candidate. Besides, the filtering function in the career section list also need to be update periodically to ease the user in finding the career that meets with their preferences.

## REFERENCES

[1] W. Shalaby, B. E. Alaila, M. Korayem, L. Pournajaf, K. Aljadda, S. Quinn, and W. Zadrozny, "Help me find a job: A graph-based approach for job recommendation at scale. ," *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017*, vol. January, 2017.

[2] K. Samundeeswary and V. Krishnamurthy, "Comparative study of recommender systems built using various methods of collaborative filtering algorithm." *ICCIDS 2017 - International Conference on Computational Intelligence in Data Science, Proceedings*, vol. January, 2018.

[3] P. Aggarwal, V. Tomar, and A. Kathuria, "Comparing Content Based and Collaborative Filtering in Recommender Systems." *International Journal of New Technology and Research*, vol. 3, 2017.

[4] S. Okyay and S. Aygun, "İşbirlikçi Filtreleme için Pearson Korelasyonu Üzerine Statik ve Dinamik Önem Ağırlıklandırma Çarpanları Çalışması." *European Journal of Science and Technology*, vol. November, 2020.

[5] M. H. Mohamed, M. H. Khafagy, and M. H. Ibrahim, "Recommender Systems Challenges and Solutions Survey." *Proceedings of 2019 International Conference on Innovative Trends in Computer Engineering, ITCE 2019*, 2019.

[6] L. D. Kumalasari and A. Susanto, "Recommendation System of Information Technology Jobs using Collaborative Filtering Method Based on LinkedIn Skills Endorsement." *Sisforma*, vol. 6, 2020.

[7] J. Son and S. B. Kim, "Content-based filtering for recommendation systems using multiattribute networks." *Expert Systems with Applications*, vol. 89, 2017.

[8] G. Geetha, M. Safa, C. Fancy, and D. Saranya, "A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System." *Journal of Physics: Conference Series*, vol. 1000, 2018.

[9] S. Dwivedi and V. S. K. Roshni, "Recommender system for big data in education." *Proceedings - 2017 5th National Conference on E-Learning and E-Learning Technologies, ELELTECH 2017*, vol. 2, 2017.

[10] M. Jalili, S. Ahmadian, M. Izadi, P. Moradi, and M. Salehi, "Evaluating Collaborative Filtering Recommender Algorithms: A Survey." *IEEE Access*, vol. 6, 2018.

[11] R. Xiong, J. Wang, N. Zhang, and Y. Ma, "Deep hybrid collaborative filtering for Web service recommendation." *Expert Systems with Applications*, vol. 110, 2018.

[12] P. Priyanga and A. R. N. B. Kamal, "Methods of Mining the Data from Big Data and Social Networks Based on Recommender System." *International Journal of Advanced Networking Applications*, vol. 8, 2017.

[13] S. Bansal, A. Srivastava, and A. Arora, "Topic Modeling Driven Content Based Jobs Recommendation Engine for Recruitment Industry." *Procedia Computer Science*, vol. 112, 2017.

[14] D. Das, L. Sahoo, and S. Datta, "A Survey on Recommendation System." *International Journal of Computer Applications*, vol. 160, 2017.

[15] A. Ochirbat, T. K. Shih, C. Chootong, W. Sommool, W. K. T. M. Gunarathne, H. H. Wang, and Z. H. Ma, "Hybrid occupation recommendation for adolescents on interest, profile, and behavior." *Telematics and Informatics*, vol. 35, 2018.

[16] F. Wójcik and M. Górnik, "Improvement of e-commerce recommendation systems with deep hybrid collaborative filtering with content: A case study." *Econometrics*, vol. 24, 2020.

[17] M. Qamhieh, H. Sammaneh, and M. N. Demaidi, "PCRS: Personalized Career-Path Recommender System for Engineering Students." *IEEE Access*, vol. 8, 2020.

[18] A. Gugnani, R. Kasireddy, V. K., and K. Ponnalagu, "Generating unified candidate skill graph for career path recommendation." *IEEE International Conference on Data Mining Workshops, ICDMW*, vol. September, 2018.

[19] J. Brooke, "System usability scale (SUS): a quick-and-dirty method of system evaluation user information." *Reading, UK: Digital equipment co ltd*, vol. 43, 1986.