A Job Finder Chatbot-Based Web Platform: A Use Case for Software Engineers

Panagiotis Fotiadis, Georgia M. Kapitsaki[a] and Maria Papoutsoglou[b]
Department of Computer Science, University of Cyprus, Aglantzia, Cyprus

Keywords: Job Adverts, Chatbot, Hard Skills, Soft Skills, Software Engineering Positions, GitHub.

Abstract: Finding a job requires browsing through a vast number of position openings, usually at various online sites. This process becomes more complicated for users when they are considering different potential job locations. Even though existing tools that automate the process exist (e.g. sonara.ai, dream.jobs), they lack the element of interactivity with the user and the integration of external resources that contain information on the skills of the user. With the aim to bridge the gap between job seekers and potential employers by matching resumes and job listings more efficiently and effectively, in this work we are introducing an AI-driven chatbot-based web platform to assist job seeking. In the initial implementation, we are considering the job seeking needs of software engineers but more disciplines can be easily added. We have integrated the developer's CV and the user's GitHub account, and are describing the design and implementation process of the web platform, while a small scale user evaluation has also been performed. We argue that the chatbot can be used as a starting point for similar job seeking assistants.

1 INTRODUCTION

There is a noticeable disconnect in the current job market, where employers struggle to find candidates with the right mix of skills and competences, while job seekers often find it challenging to locate opportunities that match their specific skill set, experience and career aspirations (Aluas et al., 2024). The essence of personalized job recommendations lies in their ability to significantly increase employment chances and simplify the job seeking process. By delivering tailored job suggestions that align closely with an individual's skills, experience, and career aspirations, these systems transform the task of job hunting into a more manageable and targeted endeavor. The utilization of Artificial Intelligence (AI) and machine learning technologies presents an opportunity to significantly improve the accuracy and relevance of job recommendations (Le Barbanchon et al., 2023). Even though existing commercial tools that offer such recommendations exist (e.g. sonara.ai, dream.jobs), they lack the element of interactivity with the user and the integration of external resources relevant to the user skills, which are useful for specific disciplines, such as the development activity of the user for software engineering positions.

In this work, we aim to create a chatbot-based web platform that efficiently matches job seekers with relevant opportunities based on a detailed analysis of their skills (hard and soft skills) and preferences. The chatbot aims to gather user information in a conversational manner, making the process of adding preferences and skills more engaging. The main contribution of the work lies in enhancing the efficiency of the job search and recruitment process by demonstrating how AI-driven chatbots can be utilized to personalize job seeking. By analyzing a large number of job listing from online sites, as well as developers' CVs and GitHub activities, the platform offers at the current state of implementation personalized job recommendations to software engineers, thereby increasing the chances of successful employment. Nevertheless, more disciplines will be added in the future. The job listings are collected from *ergodotisi.com* for Cyprus and *glassdor.com* for United States. The matching of CV and job listings is based on the user's previous years of experience, soft skills, hard skills, education, and other personal preferences, such as location and company. We have evaluated the chatbot via a small-scale user study focusing on assessing its usability and effectiveness in matching jobs and resumes, and

[a] https://orcid.org/0000-0003-3742-7123
[b] https://orcid.org/0000-0003-0658-5065

Fotiadis, P., Kapitsaki, G. M. and Papoutsoglou, M.
A Job Finder Chatbot-Based Web Platform: A Use Case for Software Engineers.
DOI: 10.5220/0013749000003985

In *Proceedings of the 21st International Conference on Web Information Systems and Technologies (WEBIST 2025)*, pages 197-204
ISBN: 978-989-758-772-6; ISSN: 2184-3252

the overall user satisfaction (Chowdhary and Chowdhary, 2020).

Chatbot Source Code Availability. The job finder chatbot is available for replication purposes online in a GitHub repository: `https://github.com/CS-UCY-SEIT-lab/Job-Finder-Chatbot`.

The remainder of the text is structured as follows. Section 2 presents relevant related work, while Section 3 describes the design choices and the implementation tools of the chatbot, including use demonstration. The user evaluation is presented in section 4, while section 5 discusses the main findings. Finally, section 6 concludes the work.

2 RELATED WORK

Chatbots. Chatbots function effectively across various fields, from customer support to personal assistance, making them a crucial asset in numerous applications (Nagarhalli et al., 2020). In customer service, they can offer 24/7 assistance, handling inquiries and resolving issues promptly (e.g. *chatbot.com*). In e-commerce, chatbots enhance the shopping experience by offering personalized recommendations and support (e.g. *maisieai.com*). In healthcare, they can triage symptoms and provide health-related information (Shinde et al., 2021), whereas they have also been used for career counseling purposes (Suresh et al., 2021). In software engineering, they have been used to assist in understanding and using Open Source Software licenses (Shittas et al., 2025).

Skills, Job Matching and Recommendations. Prior works have focused on extracting skills from job advertisements, using both automatic and manual techniques. A pre-trained LinearSVC model was used for identifying skills in advertisements in German (Grüger and Schneider, 2019), while manual analysis was used for finding soft skills in job advertisements for software positions in Cyprus (Kapitsaki et al., 2024). Earlier research has also focused on gathering software developers' prior experience from different sites in one location, in order to assist employers (Greene and Fischer, 2016; Kapitsaki and Foutros, 2017).

In job matching, an algorithm considering seeker's preferences has been introduced by Zhou et al. (Zhou et al., 2019). Zero-Shot Learning and specific pretrained models (e.g. all-MiniLM-L12-v2) were used in order to align job descriptions with candidate profiles (Kurek et al., 2024). Job advertisements from Avenga were employed for this purpose, while the used dataset contains also previous mappings manually created by recruiters. When it comes to personalized job recommendations, using the largest online job board in Sweden, it was found that recommending a vacancy to a job seeker increases the seeker's possibility to work at that specific place (Le Barbanchon et al., 2023).

Table 1: Comparison of current work with online job seeking systems.

Features	sonara.ai	simplify.jobs	careerflow.ai	dream.jobs	Our work
Chatbot functionality		✓			✓
Analyze resume/CV	✓		✓	✓	✓
Large variety of jobs in database		✓			✓
Considers hard skills	✓	✓	✓	✓	✓
Considers soft skills				✓	✓
Use of GitHub					✓

Concerning online tools, the recent escalation in the development of AI made it possible to create tools that help, personalize, and boost the process of finding the best-fitting jobs for the seekers. *Sonara*,[1] *Simplify*,[2] *Careerflow*[3] and *Dream.jobs*[4] are main such tools that offer different subscription plans. *Sonara* and *Careerflow* work by analyzing user's CV for years of experience and hard skills and based on that information filters the job listings that are posted on the side of the employer. *Simplify* uses a chatbot to collect information such as hard skills, location, salary, and preferred company. This system has a lot of companies uploading their job listings directly to them. *Dream.jobs* also has a small number of job listings, but the system's CV analyzer takes into consideration hard skills and soft skills. Table 1 provides a comparison of the features of the above systems and the current work.

3 DESIGN AND IMPLEMENTATION

3.1 Platform Overview

The implemented job finder chatbot-based web platform aims to match the abilities of a software engineer based on their CV and their GitHub activity with available job advertisements in online sites. The plat-

[1] https://www.sonara.ai/
[2] https://simplify.jobs/
[3] https://www.careerflow.ai/
[4] https://dream.jobs/

form initially analyzes users' CV and GitHub profile in order to extract hard skills and soft skills. GitHub was chosen, as it summarizes well developer's prior experience in OSS projects and has been used also in prior works for this purpose, functioning as an up-to-date CV for developers (Greene and Fischer, 2016; Kapitsaki and Foutros, 2017). Then, with the use of the chatbot, the users provide information on the desired job location, company, type of employment (full or part time), their years of experience and any additional hard skills and soft skills they want to add. Finally, using the collected information the platform filters the available job listings that are collected and analyzed from *glassdoor.com* and *ergodotisi.com*, in order to provide to users best matches.

3.2 Sources of Job Listings

The selection of *glassdoor.com* and *ergodotisi.com* as primary sources for the data collection was influenced by their structural similarities and extensive job listings. Both platforms present job advertisements in a concise list format, with each listing linked to a separate URL detailing the full job description. *Glassdoor.com*, being one of the leading job advertisement websites in the United States, and *ergodotisi.com*, holding a similar standing in a European country (Cyprus), were good candidates for compiling a comprehensive and inclusive dataset of job posts. This way two geographical areas were added in the platform, whereas additional sources for other locations can be added. Data are collected at regular intervals to ensure the platform considers recent job posts.

Before settling for those two sources, other platforms, such as *LinkedIn* and *Monster* were considered and examined. However, they were both excluded for the following reasons. *LinkedIn* features a sophisticated search engine and complex website structure, but extracting information is not allowed. *Monster* was found to have a limited number of listings for software engineering positions that were targeted for the initial implementation of the chatbot-based web platform.

3.3 Data Collection and Extraction

Concerning the data collection process, Selenium[5] was used to automate the browsing process to collect job listing data from the online platforms. BeautifulSoup4[6] was used alongside Selenium to extract specific content from web pages after they were loaded and interacted with by Selenium, taking into consideration the job listing structure from the sites. Lightcast[7] offers a taxonomy in hard and soft skills consisting of more than 33,000 skills accessible via the Lightcast's Open Skills API. This service was preferred over creating a skills taxonomy from scratch or employing other Machine Learning or Natural Language Processing (NLP) techniques to extract skills, as Lightcast updates these data constantly, ensuring that any new hard skill or technology that emerges is added in the database. The Lightcast API is then used to return the relevant soft and hard skills found within each job post description (using as input the job post text), so that these can be subsequently matched against the user's skills. The final structure of the data collected from job posts are stored in JSON (JavaScript Object Notation) format so that they can subsequently be matched against the developer profile and preferences (note however, that job postings from *ergodotisi.com* are used only for matching purposes and are not further stored).

From the side of the developer, data are collected: 1) from the user (via the interaction with the chatbot), 2) the uploaded CV and 3) the user's GitHub profile. Uploading a CV generally allows a job matching system to analyze comprehensive details about a user's educational background, skills, and accomplishments, and is the traditional approach used also in the online systems (e.g. *sonara*). Integrating GitHub data enables the system to assess in addition the user's coding projects, contributions to public repositories, and technical skills, which are often not fully captured in a traditional CV. The system can thus, identify not only the programming languages that the user knows but also their recent activity and expertise level in using those languages in real-world projects. The information collected from the user via the chatbot are specifically: location(s) of desired job positions, job type, preferred company, years of experience, education information, hard skills, and soft skills. These data are also stored in a JSON schema as job listings data.

The data retrieved and used for the job listings and the user are all listed in Table 2. Concerning the user, fields beginning with *info* are provided by the user (via the chatbot), while the rest are retrieved from the developer's CV and GitHub profile (programming languages captured in hard skills and user's location are extracted from GitHub, while hard skills and soft skills are retrieved from the user's CV). For the employment type, we are using the following options: 1) full or part time, 2) hybrid, remote or flexible, 3) permanent or temporary.

[5]https://www.selenium.dev/

[6]https://pypi.org/project/beautifulsoup4/

[7]https://lightcast.io/

Table 2: Data used for the job listing and the user.

Name	Data type
Job listings data (J)	
URL	string
location	string
company name	string
employment type	string array
years of experience	string
hard skills	string array
soft skills	string array
User data (U)	
ID	string
username	string
info_location	string array
info_job type	string array
info_company	string array
info_years of experience	string array
info_education level	string array
info_education type	string array
info_hard skills	string array
info_soft skills	string array
cv_hard skills	string array
cv_soft skills	string array
github_hard skills	string array
github_location	string

3.4 Skills Matching

When all data from the job adverts and the user are available, skill matching is performed using the two sets of data: one set comprises of the hard or soft skills of the user (U) and the second set contains the respective hard or soft skills from the job adverts (J). We decided to perform a match on skills on keywords level, as the extended taxonomy of Lightcast covers a very large number of skills. For this purpose, we are relying on Jaccard similarity:

$$J(J,U) = \frac{|\,J \cap U\,|}{|\,J \cup U\,|} \tag{1}$$

The two sets are considered relevant for the user, and hence the respective job advert, if there is a value of 0.2 or higher for either the hard skills or the soft skills matching. We have opted for this threshold after experimentation with the matching process in the sources of job adverts, but the threshold can be updated (e.g. if more sources are added).

3.5 Implementation Tools and Chatbot Flow

The Rasa[8] framework was used at the core of the implementation as it was adopted as the chatbot implementation framework. It was chosen due to its popularity and since it has an Open Source edition. Rasa uses a number of steps for understanding user's intent and performing the appropriate action. It combines tokenization, feature extraction, and sophisticated classification and decision-making mechanisms, to empower the application to handle user interactions effectively, recognizing intents and extracting entities, with the above features:

1. *User input:* The job finder chatbot collects detailed user inputs across several categories critical to job matching, including geographic location, job type preference (e.g. full time), years of experience, educational background, soft skills and hard skills, as detailed in subsection 3.3. It engages users in a comprehensive dialogue, ensuring that all relevant information is gathered to support personalized job recommendations.
2. *NLP analysis:* The job finder chatbot successfully interprets and processes user responses, such as interpreting "Alabama" as a location and recognizing "5 years" as the level of experience. The NLP component of Rasa has been tuned in the framework of the current work with relevant configuration files offered to understand and extract meaningful information from user inputs.
3. *User feedback mechanism:* Each stage of the interaction with the chatbot allows for user confirmation or correction, which is crucial for maintaining the accuracy of user data.
4. *Data integration:* The job finder chatbot-based web platform is an integrated platform that takes user data (e.g. skills and experience) from various sources (chatbot, CV, GitHub) and maps them to the available job postings (then shows to the user links to job listings tailored to the user's profile). The backend of the web platform is leveraging the collected data to fetch relevant job opportunities from external job sites.

For the chatbot functionality, Rasa relies on a number of key configuration files, including *domain.yml*, *nlu.yml*, and *stories.yml*. Understanding user's intent is vital for the above process, and specifically items 2 and 3. The *nlu.yml* file contains lookup tables with expected words that the user can give and variations of answers that the user might give using

[8]https://rasa.com/

the lookup tables' words. This file is used to create the natural language model which the chatbot is based on, recognizing the user's answers and deciding on the chatbot's story flow. A subset of the possible user intents when collecting soft skills is presented below, with the text in square brackets (*[]*) referring to the parameter from the user provided text and the text in parenthesis (*()*) referring to the entity's name, i.e. *(soft_skills)* in the example intent:

```
- intent: collect_soft_skills
  examples: |
    - I am good at [teamwork](soft_skills)
    - [leadership](soft_skills)
    - [fast learner](soft_skills) and
    [leadership](soft_skills)
    - [fast learner](soft_skills), [leadership]
    (soft_skills) and [teamwork](soft_skills)
    - I excel in [communication](soft_skills)
    and [team building](soft_skills).
    - My strengths include [critical thinking]
    (soft_skills) and [problem solving](soft_
    skills).
    - I possess strong [analytical skills]
    (soft_skills) and [innovation](soft_skills)
    capabilities.
    - [Self-motivation](soft_skills) is one of
    my core qualities.
    - I have experience with [virtual teams]
    (soft_skills) and [knowledge transfer]
    (soft_skills).
    - [Consulting](soft_skills),
    [report writing](soft_skills), and
    [budgeting](soft_skills) are my expertise
    areas.
    - I am known for my [reliability]
    (soft_skills) and [ability to meet
    deadlines](soft_skills).
    - Proficient in [quality assurance]
    (soft_skills) and [progress reporting]
    (soft_skills).
    [...]
```

The *domain.yml* file contains intents and entities names that can be extracted from user's input. In our case, the entities correspond to the user data that can be collected from the chatbot as presented in Table 2 (starting with *info*). *Stories.yml* file defines a diagram of the chat flow and how the chatbot must proceed after each question is answered and what information it should expect from the user to provide. We are relying on a main story flow, which contains a large number of actions and intents, in order to collect all required user data sequentially. All configuration files are available on the GitHub repository of the platform.

For the main implementation of the platform, MongoDB was also used for data storage, Docker was used for making the chatbot deployment easier, and Flask was used for the server-side logic and database interactions. We also employed JSON Web Tokens (JWT) that are defined by the open standard RFC 7519 for securely transferring information, while we relied on React JS for frontend development purposes.The interconnection between the main tools is depicted in Figure 1.

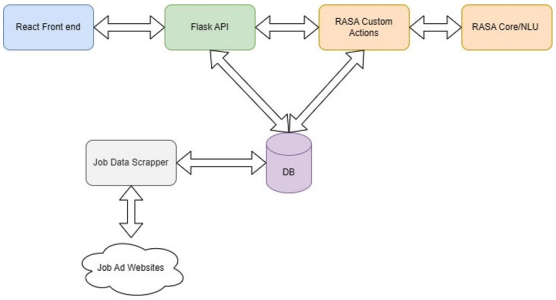

Figure 1: Interaction of implementation tools.

3.6 Use Demonstration

Basic interactions with the chatbot-based web platform are depicted in Figure 2 and Figure 3. When the user is logged in the system, the available page layout allows the user to refresh the current chat (by clicking "Refresh Chat"), create a new chat (by clicking "New Chat"), upload his/her CV (using "Resume/CV Details"), trigger the GitHub analysis (by giving his/her GitHub username) and navigate to old chats. An example of the respective analysis results after the user has uploaded his/her CV and has triggered the GitHub analysis is shown in Figure 2. When the user adds his/her GitHub username, the backend uses GitHub's API to detect all programming languages that are in the repositories the user has contributed to, and to also expert the user's location if this information is available in the user profile on GitHub. The text of the CV is also analyzed to detect hard and soft skills (relying as aforementioned on the taxonomy provided by Lightcast). Those data are used along with the data collected through the chatbot conversation in order to find the most fitting job adverts for the user.

Concerning the chatbot, user's temporary data that are stored during a chatbot use are deleted every time the user logs in or clicks on "Refresh Chat" option. An example interaction flow with a user is depicted in Figure 3. The user is interested in job posts in Alabama, whereas Larnaca is detected as user's location from GitHub, so both locations are used for the job listings recommendations. The user is interested in full type employment, has 5 years of experience and is not interested in specific companies. The user then provides information on his/her education level, hard and soft skills, and at the end the chatbot recommends a specific job post to the user.

4 USER EVALUATION

Survey Creation. The chatbot-based web platform for job recommendations has been evaluated with the

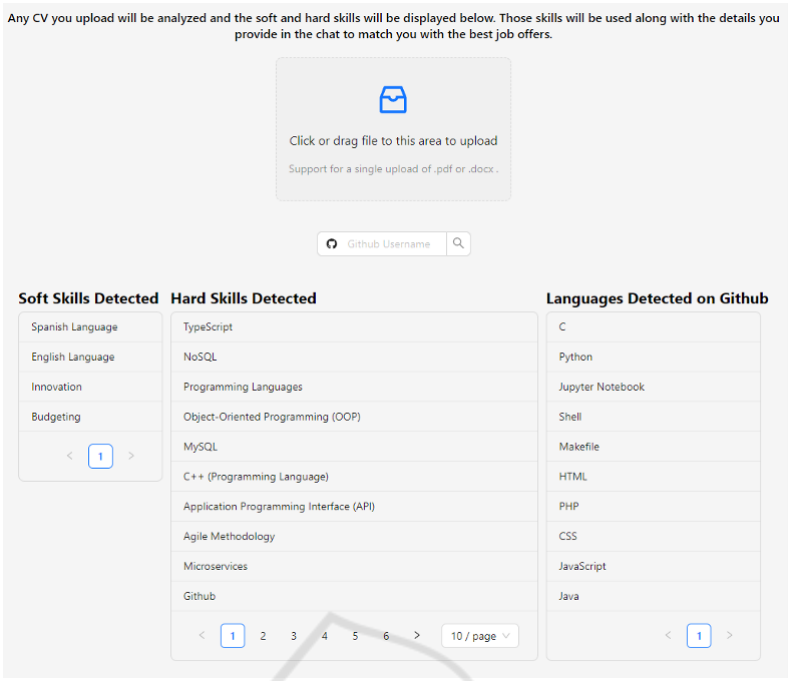

Figure 2: CV analysis in job finder chatbot.

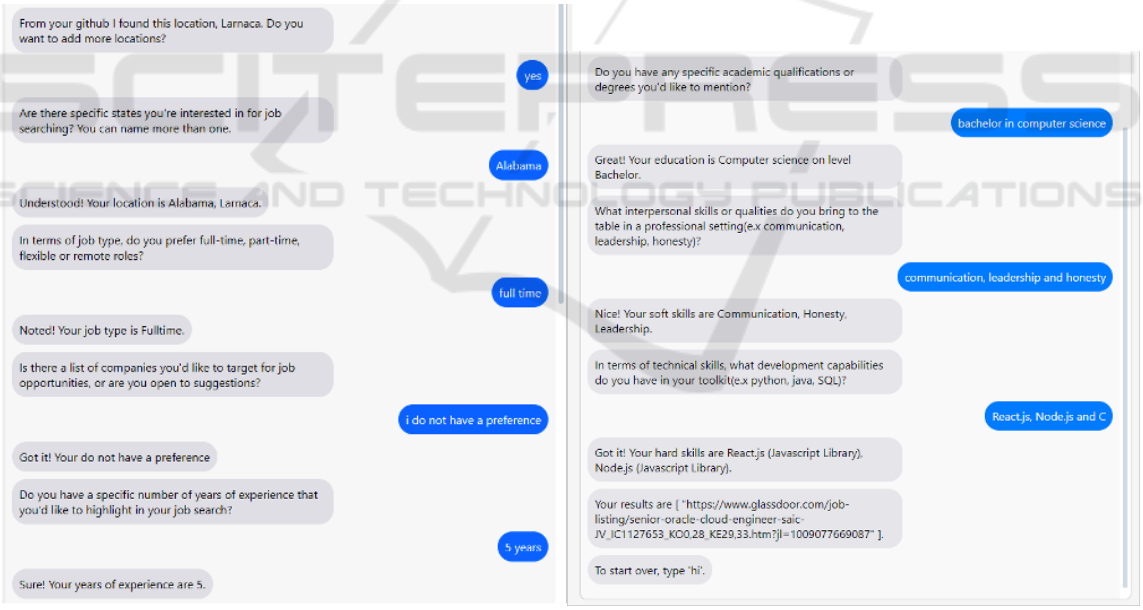

Figure 3: Interaction example with the job finder chatbot.

participation of users from the University of Cyprus. The user evaluation aimed to gather feedback on the usability, effectiveness, and overall user experience with the web platform. The questionnaire consisted of 15 questions that cover the main parts presented in Table 3, including prior experience with job search and the feedback from the interaction with the implemented platform. Questions used are mainly multiple choice, 5-Likert case and open ended questions. Potential participants within the Computer Science Department of University of Cyprus were contacted via email communication. Participants were informed that no personal data would be collected in the survey, that their replies would be used solely for research purposes and would not trace back to the participant in any way. In order to answer the questionnaire, par-

Table 3: Main parts of user evaluation questionnaire.

Section	# questions	Question examples
Background, job search behavior and preferences	4	What is your current employment status? (*multiple choice*)
Previous experience with job finder tools	4	In which industry/field are you seeking employment? (*open*) How many days per week do you dedicate to looking for job opportunities? (*open*)
Willingness to use AI-driven job finder tools	2	Would you use an AI chatbot that finds you personalized job listings based on the details given on the chatbot, GitHub and CV that you provide? (*multiple choice*)
Feedback on chatbot usage (after interacting with the chatbot)	5	JobFinder chatbot was easy to use. (*5-Likert scale*) I trust the job positions recommended to me by the chatbot. (*5-Likert scale*)

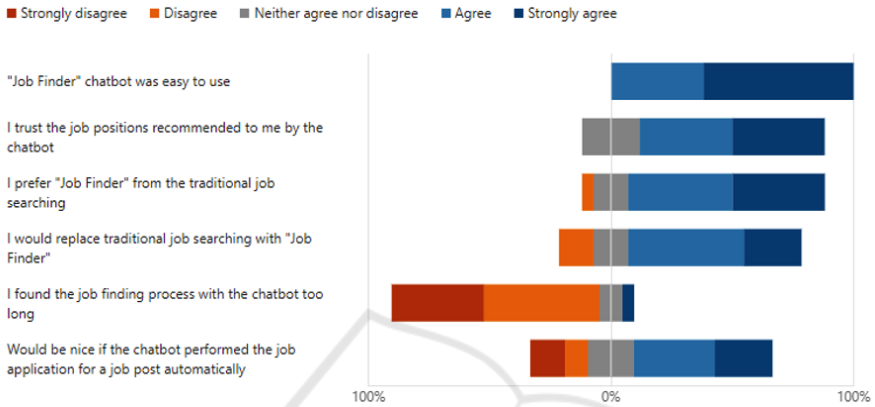

Figure 4: Users' replies to main questions of evaluation.

ticipants gave their consent to the above.

Results. 21 users participated in the evaluation in total. 13 (61.9%) of them were employed, while 3 (14.3%) were seeking employment and 5 (23.8%) were students. Data concerning the hours users spent on job search were collected (second section of the questionnaire), with an average value of 5.4 hours per week indicated by the users (standard deviation: 3.56 hours) and with some participants spending even 10 hours on this task weekly.

Concerning the experience with using the chatbot-based web platform, most participants found it easy to use: 13 (61.9%) participants agreed strongly and 8 (38.1%) agreed. Replies' distribution to some of the main questions on the chatbot interaction are shown in Figure 4. The chatbot was generally well-received, particularly in terms of usability, trustworthiness in job recommendations and the length of the job search process. An important question was whether users would prefer this chatbot-based approach over the traditional job search and most of the users indicated a preference for the chatbot: 15 (71.4%) participants agreed or agreed strongly. Adding more features, such as automatic job application by the chatbot on behalf of the user, were also considered useful. Such features were among the participants suggestions for improving the chatbot.

5 DISCUSSION

We have gathered overall positive feedback from the users, which allows also to transfer the current implementation to other disciplines beyond software engineering. This would require considering potentially also other sources of information for skills acquisition besides GitHub that is tailored to software development. The main matching performed based on the chatbot interaction with the user does not require any adaptations for this transfer. Concerning the data sources used, the accuracy of the recommendations provided by the chatbot depends heavily on the comprehensiveness and reliability of the information it receives from the users. However, we argue that it is vital to use more than one sources of information. If the chatbot relies solely on conversational data provided during interactions with the user and does not have access to additional detailed sources like CVs, the matching might not be as accurate.

Limitations. In terms of data collection for job adverts, we are limited by the terms of service and the structure of respective sites, and for this reason it is not feasible to use any site for job adverts collection. Concerning the software development activity, we are relying on GitHub to gather relevant data but this may not reflect the user's activity on all skills level, as it refers mainly to programming language skills. More-

over, the evaluation has been limited to the population of University of Cyprus, so main results might differ if a wider population of software engineers employed in the industry had tested the chatbot.

6 CONCLUSIONS

In this work, we have presented a job finder chatbot-based web platform, that assists users to get personalized recommendations for their software development job search, considering their preference and skills, and using different data sources (CV and GitHub profile). The chatbot can be used as a starting point for the development of similar interactive job seeking systems, and its initial small-scale user evaluation is promising in this respect. Future work will expand the chatbot's algorithms and databases to support a broader range of job categories beyond software engineering, while new sources of job adverts covering more countries will be added. Moreover, participants from the software industry will also be recruited for evaluation purposes. An extended evaluation will allow to draw more conclusions on its comparison with traditional job searching techniques.

REFERENCES

Aluas, M., Angelis, L., Arapakis, I., Arvanitou, E., Georgiou, K., Gogos, A., Jahn, M., Kehagias, D., Kordoni, V., Macaluso, S., et al. (2024). Skillab: Skills matter. In *2024 50th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 491–498. IEEE.

Chowdhary, K. and Chowdhary, K. (2020). Natural language processing. *Fundamentals of artificial intelligence*, pages 603–649.

Greene, G. J. and Fischer, B. (2016). Cvexplorer: Identifying candidate developers by mining and exploring their open source contributions. In *Proceedings of the 31st IEEE/ACM international conference on automated software engineering*, pages 804–809.

Grüger, J. and Schneider, G. J. (2019). Automated analysis of job requirements for computer scientists in online job advertisements. In *WEBIST*, pages 226–233.

Kapitsaki, G., Chatzivasili, L., Papoutsoglou, M., and Galster, M. (2024). An exploratory study on soft skills present in software positions in cyprus: a quasi-replication study. In *Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 200–211.

Kapitsaki, G. M. and Foutros, P. (2017). Dear developers, your expertise in one place. In *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 371–374. IEEE.

Kurek, J., Latkowski, T., Bukowski, M., Świderski, B., Łępicki, M., Baranik, G., Nowak, B., Zakowicz, R., and Dobrakowski, Ł. (2024). Zero-shot recommendation ai models for efficient job–candidate matching in recruitment process. *Applied Sciences*, 14(6):2601.

Le Barbanchon, T., Hensvik, L., and Rathelot, R. (2023). How can ai improve search and matching? evidence from 59 million personalized job recommendations. *Evidence from*, 59.

Nagarhalli, T. P., Vaze, V., and Rana, N. (2020). A review of current trends in the development of chatbot systems. In *2020 6th International conference on advanced computing and communication systems (ICACCS)*, pages 706–710. IEEE.

Shinde, N. V., Akhade, A., Bagad, P., Bhavsar, H., Wagh, S., and Kamble, A. (2021). Healthcare chatbot system using artificial intelligence. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1–8. IEEE.

Shittas, G., Kapitsaki, G. M., and Papoutsoglou, M. (2025). Licy: A chatbot assistant to better understand and select open source software licenses. In Mannion, M., Männistö, T., and Maciaszek, L. A., editors, *Proceedings of the 20th International Conference on Evaluation of Novel Approaches to Software Engineering, ENASE 2025, Porto, Portugal, April 4-6, 2025*, pages 573–581. SCITEPRESS.

Suresh, N., Mukabe, N., Hashiyana, V., Limbo, A., and Hauwanga, A. (2021). Career counseling chatbot on facebook messenger using ai. In *Proceedings of the International Conference on Data Science, Machine Learning and Artificial Intelligence*, pages 65–73.

Zhou, Q., Liao, F., Chen, C., and Ge, L. (2019). Job recommendation algorithm for graduates based on personalized preference. *CCF Transactions on Pervasive Computing and Interaction*, 1:260–274.